Blind Identification of Manufacturing Variation Patterns by Combining Source Separation Criteria

Xuemei SHAN
Department of Industrial Engineering and Management Sciences
Northwestern University
Evanston, IL 60208-3119
(xuemeishan@discover.com)

Daniel W. APLEY
Department of Industrial Engineering and Management Sciences
Northwestern University
Evanston, IL 60208-3119
(aley@northwestern.edu)

Blind source separation recently has been investigated for blindly identifying variation patterns in multivariate manufacturing data, to aid in tracking down and eliminating root causes of manufacturing variation. Many different criteria can be used in blind separation algorithms, the performance and applicability of which depend on conditions that generally are not known a priori. We present a method for automatically combining the different criteria so as to directly minimize the mean squared estimation error. The resulting algorithm is more effective and more robust than counterparts that use other means of combining the criteria.

KEY WORDS: Blind source separation; Factor rotation; Manufacturing variation reduction; Multivariate statistical process control; Principal components analysis.

1. INTRODUCTION

Large amounts of multivariate measurement and inspection data are routinely available in many manufacturing processes. This creates tremendous potential for extracting information regarding the nature of process variability, to aid in root cause identification and variation reduction efforts. Toward this end, let \( \mathbf{x} = [x_1, x_2, \ldots, x_n] \) denote a vector of \( n \) measured product/process variables, and let \( \{x_i, i = 1, 2, \ldots, N\} \) denote a sample of \( N \) observations of \( \mathbf{x} \). To represent the effects of major variation sources on the measurement data, we use the model

\[
\mathbf{x}_i = \mathbf{Cv}_i + \mathbf{w}_i, \quad i = 1, 2, \ldots, N, \tag{1}
\]

where \( \mathbf{C} = [c_{ij}] \) is an \( n \times p \) unknown constant matrix, \( \mathbf{v}_i = [v_{i1}, v_{i2}, \ldots, v_{ip}]^T \) is a random vector representing \( p \) independent variation sources (typically \( p \ll n \)), and \( \mathbf{w}_i = [w_{i1}, w_{i2}, \ldots, w_{in}]^T \) is a mean-0 random noise vector. The variation pattern vector \( \mathbf{c}_j, j = 1, 2, \ldots, p \), characterizes the spatial nature of the \( j \)th variation source, and \( \{\mathbf{v}_{ij}, i = 1, 2, \ldots, N\} \) characterizes its temporal nature over the sample. Together, \( c_{ij} \) represents the effect of the \( j \)th variation source on the \( i \)th multivariate observation. Throughout the article, we drop the observation index \( i \) when convenient. In the context of statistical process control, one might view the \( \mathbf{C} \) and \( \mathbf{w} \) terms in \( (1) \) as the systematic assignable causes of variability and the common causes of variability. Because the focus is on variation, rather than on mean shifts sustained across the sample, we assume that the sample mean is subtracted from the data. Thus in the model \( (1) \), \( \mathbf{x} \) and \( \mathbf{v} \) represent deviations from their means. We assume that the covariance matrix of \( \mathbf{w} \) is \( \Sigma_w = \sigma^2 \mathbf{I} \) for some scalar \( \sigma \). In situations in which this assumption is unreasonable, such as when the elements of \( \mathbf{x} \) are on entirely different scales, the data should be transformed using methods described by Apley and Lee (2003) before applying the procedure described in this article.

The model \( (1) \) is quite broadly applicable in manufacturing. Apley and Shi (2001), Apley and Lee (2003), and Lee and Apley (2004) used the model to represent variation patterns in automobile body assembly and crankshaft manufacturing. In one application, for example, \( \mathbf{x} \) represented \( n \approx 150 \) dimensional features distributed across an automobile, and samples consisted of \( N \approx 200 \) separate automobiles measured through an in-process laser-optical metrology station. One of the variation sources (say \( \mathbf{v}_1 \)) was due to a worn locating element in the assembly tooling. The resulting pattern vector \( \mathbf{c}_1 \) represented a translation of a particular subassembly with respect to the rest of the automobile, and \( \mathbf{v}_1 \) represented the amount of translation for automobile number \( i \) in the sample. Other tooling- and stamping-related problems resulted in additional variation sources that were similarly characterized as translations or rotations of automobile panels.

A large body of related previous work (e.g., Ceglarek and Shi 1996; Barton and Gonzalez-Barreto 1996; Apley and Shi 1998; Ding, Shi, and Ceglarek 2002) has assumed the linear model \( (1) \) but focused on the situation in which one is able to preidentify a set of potential variation sources offline and premodel (analytically, or perhaps empirically) their effects to obtain a known potential \( \mathbf{C} \) matrix. In this case the online task reduces to detecting which columns of \( \mathbf{C} \) are present as pattern vectors in the current sample of data. This approach obviously is limited to processes in which the knowledge of the potential variation sources and the physics of the process is sufficiently detailed to allow accurate premodeling of the sources.

The objective of this work, like that of Apley and Shi (2001), Apley and Lee (2003), and Lee and Apley (2004), is quite different. We require neither preidentification nor premodeling of the variation sources. Given a sample of measurement data \( \{\mathbf{x}_i, i = 1, 2, \ldots, N\} \), the objective is to blindly estimate the
number of sources $p$, the $\mathbf{C}$ matrix composed of the variation pattern vectors, and the temporal history $\{v_i, i = 1, 2, \ldots, N\}$ of the sources. For $j = 1, 2, \ldots, p$, the estimates of $\mathbf{c}_j$ and $\{v_{ij}, i = 1, 2, \ldots, N\}$ then can be graphically illustrated to provide a process engineer with insight into the spatial and temporal nature of each major source of process variability. This aids in identifying and subsequently eliminating (or controlling) their root causes, thereby reducing process variability.

With this objective, Apyle and Shi (2001), Apyle and Lee (2003), and Lee and Apyle (2004) all used what can be considered forms of factor analysis, with principal components analysis (PCA) as the first step. Note that the model (1) is essentially a linear orthogonal factor analysis model. It follows from the model and assumptions that $\mathbf{C}$ is of the form $\mathbf{Z}_p(\mathbf{A}_p - \sigma^2 \mathbf{I})^{1/2} \mathbf{Q}$ for some $p \times p$ orthogonal rotation matrix $\mathbf{Q}$ (Apyle and Lee 2003). Here $\mathbf{A}_p$ is a $p \times p$ diagonal matrix containing the $p$ largest eigenvalues of $\mathbf{\Sigma}_x$ (the covariance matrix of $x$), and $\mathbf{Z}_p$ is an $n \times p$ matrix, the columns of which are the eigenvectors corresponding to the $p$ largest eigenvalues. Consequently, after PCA is performed on the sample covariance matrix $\mathbf{\Sigma}_x$, to obtain $\mathbf{Z}_p$ and $\mathbf{A}_p$, the problem of blindly estimating $\mathbf{C}$ reduces to estimating $\mathbf{Q}$ and then taking $\hat{\mathbf{C}} = \mathbf{Z}_p(\mathbf{A}_p - \sigma^2 \mathbf{I})^{1/2} \mathbf{Q}$. Throughout the article, the """" symbol denotes an estimate of a quantity. Traditional factor rotation methods select $\mathbf{Q}$ to provide the simplest interpretation of the factors (also known as variation sources). Using the varimax criterion (Jackson 1980, 1981; Johnson and Wichern 1998), for example, one attempts to rotate the factors so that each factor affects a disjoint subset of measured features. This implicitly assumes a structure for $\mathbf{C}$ that has little physical justification in manufacturing applications. Apyle and Shi (2001) assumed a somewhat less restrictive structure for $\mathbf{C}$ that would have broader applicability in manufacturing. The only statistical information that traditional factor analysis methods and Apyle and Shi (2001) take into account when estimating $\mathbf{Q}$ is the covariance of the data.

Blind source separation (BSS) methods, some of which are referred to as independent component analysis (Cardoso and Souloumiac 1993; Comon 1994), use additional statistical information to estimate $\mathbf{Q}$. Apyle and Lee (2003) investigated BSS methods for manufacturing variation diagnosis that use either autocovariance or fourth-order cumulants as criteria for finding $\hat{\mathbf{Q}}$. The use of any particular criterion requires certain conditions regarding the source distributions to be satisfied for the sources to be separable. The condition for separating any pair of sources using the second-order (i.e., autocovariance) criterion is that they have different autocorrelation functions (equivalently, different spectra; Belouchrani, Abed-Meraim, Cardoso, and Moulines 1997), and the condition for separating any pair of sources using the fourth-order criterion is that at least one of the pair has nonzero kurtosis (Cardoso and Souloumiac 1993), which is violated if both are Gaussian. Combining the various criteria in a single algorithm relaxes the separability conditions and can provide an overall improvement in separation performance. Toward this end, Müller, Philips, and Ziehe (1999) proposed the JADE$^4D$ algorithm to combine autocovariance and fourth-order cumulants. In a certain sense, to be defined later, their algorithm places equal weight on the different criteria. Cruces-Alvarez and Cichocki (2003) suggested placing different weightings on the different criteria to achieve better performance, but provided no strategy for accomplishing this. Lee and Apyle (2004) combined autocovariance and fourth-order cumulants through a weighting strategy that places more weight on criteria that appear to contain more information relevant to separating the sources. Tichavský, Koldovský, Doron, Yeredor, and Gomez-Herrero (2006) proposed an algorithm that uses autocovariance to separate some sources and fourth-order cumulants to separate others.

The primary purpose of this work is to propose a more effective method of combining the various BSS criteria for estimating $\mathbf{Q}$. The criteria are combined by selecting their relative weighting to directly minimize an optimality index that is an aggregate measure of the mean squared error (MSE) in estimating the variation pattern vectors. We demonstrate that the proposed method generally provides more accurate blind estimation than other methods of combining BSS criteria. In particular, because of the manner in which the criteria are combined, the performance is relatively robust to violations in the conditions under which the sources can be separated when the various criteria are used individually. Robustness is critically important, because it leads toward a more broadly applicable, black-box approach that requires less user expertise to verify whether the data satisfy the various separability conditions.

The specific criteria that we combine are autocovariance, third-order cumulants, and fourth-order cumulants. Although most previous BSS work has focused on second- and/or fourth-order information, we demonstrate that including third-order cumulants improves the estimation accuracy when sources follow skewed distributions, which is quite common in manufacturing (e.g., the third source in the crankshaft example in Sec. 4).

The remainder of the article is organized as follows. In Section 2 we provide background on BSS and existing methods of combining BSS criteria. We introduce the proposed approach for optimally combining the BSS criteria to minimize the MSE in Section 3. We illustrate the approach in Section 4 with an example from crankshaft manufacturing. In Section 5 we use simulation to compare the performance of a number of BSS algorithms in various settings. In Section 6 we present some conclusions.

2. BACKGROUND ON BLIND SOURCE SEPARATION

BSS refers to a class of methods to blindly recover (or separate) the source signals $\{v_i, i = 1, 2, \ldots, N\}$ in the model (1), based on the observed data $\{x_i, i = 1, 2, \ldots, N\}$. Most BSS approaches were originally developed for sensor array signal processing applications. The separation is blind in the sense that no knowledge of $\mathbf{C}$ is available, other than what can be inferred from the same sample of data. Comprehensive tutorials on BSS have been given by Cardoso (1998), Hyvarinen and Oja (2000), and Cichocki and Amari (2002). The elements of $\mathbf{v}$ are assumed to be statistically independent of one another and are scaled to have unit variance (without loss of generality, because the columns of $\mathbf{C}$ can be rescaled accordingly). The noise $\mathbf{w}$ often is assumed to be negligible. If not, then it is assumed to be multivariate Gaussian, independent of $\mathbf{v}$, and temporally uncorrelated with covariance $\sigma^2 \mathbf{I}$. Apyle and Lee (2003) discussed
how to accommodate a more general noise covariance structure $\Sigma_w$ when an estimate is available from a sample of in-control data or gauge repeatability and reproducibility studies.

Many BSS approaches use PCA as the first step. Estimates of $p$ and $\sigma^2$ are taken to be the number of dominant eigenvalues and the average of the $n - p$ remaining eigenvalues (see Apley and Lee 2003 for details). To estimate $Q$, it is more convenient to work with the whitened data,

$$y_i = [\Lambda_p - \sigma^2I]^{-1/2}Z_p x_i, \quad i = 1, 2, \ldots, N. \quad (2)$$

If, hypothetically, the eigenstructure were estimated perfectly, then the whitened data $y$ would be a rotated version $Qv$ of the sources, plus linearly transformed noise (see Apley and Lee 2003). In light of this and the assumption that the components of $v$ are independent, one strategy is to select $Q$ so that the components of the estimated source vector $\hat{v} = \hat{Q}y$ have as little cross-autocovariance as possible. In other words, we attempt to make the estimated autocovariance matrices $\Sigma_{x, y} = \hat{Q}^2 \Sigma_{x, y} \hat{Q}$ (1 $\leq \tau \leq T$ for some selected maximum lag $T$) as close to diagonal as possible, where $\Sigma_{x, y}$ is the lag-$\tau$ sample autocovariance matrix of the whitened data $y$. Specifically, Belouchrani et al. (1997) proposed taking $\hat{Q}$ to be the orthogonal matrix $U$ that minimizes $\sum_{\tau=1}^{T} \text{off}[U^* \Sigma_{x, y} U]$, where $\text{off}[\bullet]$ denotes the sum of squares of the off-diagonal elements of a matrix. In other words, $Q$ is the orthogonal matrix that jointly approximately diagonalizes the matrix set $\{\Sigma_{x, y}: 1 \leq \tau \leq T\}$. Cardoso and Souloumiac (1993) described an algorithm for jointly approximately diagonalizing a set of symmetric matrices using a series of Givens rotations.

Although the components of $y$ are always spatially uncorrelated as a fundamental property of PCA, they are not necessarily independent. This gives rise to another strategy for estimating $Q$ (which is ineffective if the sources are Gaussian): rotate the whitened data through $\hat{v} = \hat{Q}y$ until the components are as independent as possible, using higher-order cross-cumulants to measure independence. For example, the fourth-order cumulant of the $i$th, $j$th, $k$th, and $l$th elements of a mean-zero vector $y$ ($1 \leq i, j, k, l \leq p$) is defined as $C_{i, j, k, l}(y) = E[y_1 y_2 y_3 y_4] - E[y_1 y_2]E[y_3 y_4] - E[y_1 y_3]E[y_2 y_4] - E[y_1 y_4]E[y_2 y_3]$. Cross-cumulants have $i, j, k, l$ not all equal. The sample cumulants $\hat{C}_{i, j, k, l}(y)$ can be obtained by replacing the expectations by the corresponding sample averages. Cardoso and Souloumiac (1993) have shown that minimizing the sum of the squares of a certain set of fourth-order cross-cumulants of $\hat{v}$ is equivalent to minimizing $\sum_{\tau=1}^{T} \text{off}[U^* \hat{M}(i, j) U]$. Here $\{\hat{M}(i, j): 1 \leq i, j \leq p\}$ is a set of $p \times p$ matrices constructed from the fourth-order sample cumulants of $y$, defined such that the $k$th row, $l$th column element of $\hat{M}(i, j)$ is $\hat{C}_{i, j, k, l}(y)$. Cardoso and Souloumiac (1993) also proposed an algorithm for performing the joint approximate diagonalization of the cumulant matrices.

Although autocovariance and fourth-order cumulants are the most common BSS criteria, we also incorporate third-order cumulants in this work. Hypothetically, if we were to use third-order cross-cumulants as the only measure of independence, then we might select the orthogonal rotation matrix $Q$ to minimize the sum of the squares of the third-order cross-cumulants of $\hat{v}$: $\hat{Q}y$. The third-order cumulant of the $i$th, $j$th, and $k$th ($1 \leq i, j, k \leq p$) elements of a mean-zero vector $y$ is defined as $C_{i, j, k}(y) = E[y_i y_j y_k]$. Using the additivity and multilinearity properties of cumulants, it is straightforward to establish the equivalence of minimizing $\sum_{i \leq j < k \leq p} \sum_{i \leq j < k \leq p} \text{off}[U^* \hat{M}(i, j, k, l) U]$ over all orthogonal matrices $U$, where $\{\hat{M}(i, j, k, l): 1 \leq i < j < k \leq p\}$ denotes the set of $p \times p$ third-order cumulant matrices defined such that the $j$th row, $k$th column element of $\hat{M}(i, j, k, l)$ is $\hat{C}_{i, j, k}(y)$. Consequently, if we hypothesically were to use third-order cumulants as the only BSS criterion, then we could take $\hat{Q}$ to be the joint approximate diagonalizer of the matrix set $\{\hat{M}(i, j, k, l): 1 \leq i < j < k \leq p\}$.

In this work we combine the various BSS criteria by taking $\hat{Q}$ to be the minimizer of

$$\sum_{k=1}^{K} \alpha_k \text{off}[U^* \hat{A}_k U] \quad (3)$$

over all orthogonal matrices $U$, where $\alpha = [\alpha_1, \alpha_2, \ldots, \alpha_K]$ is a vector of weighting coefficients and $\{\hat{A}_k: 1 \leq k \leq K\}$ is the set of autocovariance matrices, third-order cumulant matrices, and fourth-order cumulant matrices. Thus $K = T + p + p^2$ is the total number of matrices to be jointly approximately diagonalized. Lee and Apley (2004) proposed the same idea, but excluded third-order cumulants and used a relatively simple method for selecting the weights. As we demonstrate in Section 5, a proper choice of weights is needed to effectively balance the criteria and avoid placing too much weight on criteria that contain little information for the data being analyzed. For example, if the sources for a particular data set happened to have little autocorrelation, then we would want to have little weight on the autocovariance matrices in (3). Placing too much weight on them would only degrade the estimation accuracy.

Our approach for selecting the weights to automatically balance the different criteria is an empirical one, based only on information contained in the data and involving no assumed models for the source distributions. By this, we mean that to implement the approach, we do not need to assume a specific distribution (e.g., uniform or Bernoulli) for the sources, or a specific model [e.g., autoregressive (AR) or moving average (MA)] for the autocorrelation, or the relative importance of higher-order cumulants versus autocorrelation information. As pointed out by an anonymous referee, certain basic distributional assumptions, such as the existence of cumulants and autocorrelation functions, clearly are required. Moreover, because our method involves combining different criteria, certain elements of the different separability conditions must be satisfied; for example, we cannot have all Gaussian sources with all the same autocorrelation function. Before we describe our proposed approach, we review the approach used by Lee and Apley (2004) for selecting the weights. They used an error matrix $E$ to represent the error between $\hat{Q}$ and $Q$, defined through

$$\hat{Q} = Q(I + E). \quad (4)$$

Adapting a result from Cardoso (1994), Lee and Apley (2004) showed that the $ith$ row, $jth$ column element of $E$ ($1 \leq i \neq j \leq p$) can be approximated as a function of $\hat{A}_k$ ($1 \leq k \leq K$),

$$E_{ij} \approx \frac{\sum_{k=1}^{K} \alpha_k d(k) - d(i) - d(j)}{\sum_{k=1}^{K} \alpha_k d(i) - d(j)^2}.$$

where $q_k$ denotes the $ith$ column of $Q$, $d(k)$ ($1 \leq k \leq K$) is a $p$-length vector comprising the diagonal elements of $Q^T \hat{A}_k Q$
and $d_j(k)$ denotes the $j$th element of $d(k)$. Because of the difficulty in selecting the weights to minimize some direct measure of $E$, Lee and Apley (2004) proposed maximizing $\sum_{i \neq j}^K \sum_{k=1}^K \alpha_k (d_j(k) - d_j(k))^2$ (i.e., the summation of the denominators of (5) over all $i \neq j$ subject to an equality constraint on the norm of the weighting vector $\alpha$, the solution to which is $\alpha_k \propto \sum_{i \neq j} (d_j(k) - d_j(k))^2$ (1 $\leq k \leq K$). They pointed out a close relationship between the term $\sum_{i \neq j} (d_j(k) - d_j(k))^2$ and the information in $\hat{A}_k$ that is pertinent to separating the sources. Although this solution is simple and intuitively appealing, it often can result in undesirable weighting. Maximizing the summation of the denominators of (5) over all $i \neq j$ can lead to some elements of $E$ being small and others being very large, which translates to some of the sources being estimated accurately and others being estimated poorly.

3. SELECTING THE WEIGHTS TO MINIMIZE THE MEAN SQUARED ERROR

Our approach to selecting the weighting vector $\alpha$ is to minimize an aggregate measure of the mean squared estimation error, which we define as

$$MSE(\alpha) = \sum_{i=1}^n \sum_{j=1}^p E(\hat{C}_{ij} - C_{ij})^2,$$  \hspace{1cm} (6)

where $C_{ij}$ is the $i$th row, $j$th column element of $C$. The MSE is a function of $\alpha$ because $\hat{C}$ depends on $\hat{Q}$, which depends on the weights chosen in (3). Because an exact analytical expression for $MSE(\alpha)$ as a function of $\alpha$ is intractable, in this section we present an approach to approximating $MSE(\alpha)$. The approach uses a first-order Taylor approximation of $\hat{C}$ combined with Markov bootstrap estimation of various quantities involved in the approximation. The Taylor approximation for $MSE(\alpha)$ is derived in Section 3.1, followed by a step-by-step illustration of the entire estimation algorithm in Section 3.2 and a description of the Markov bootstrapping procedure in Section 3.3.

3.1 Taylor Expansion of $\hat{C}$ About $C$

To derive a first-order Taylor expansion of $\hat{C}_{ij}$ about its true value $C_{ij}$, as an explicit function $\alpha$, we define $\hat{F} = \hat{F}_p(\hat{\Lambda}_p - \hat{\sigma}^2 I)^{1/2}$. Using this and (4), the estimator $\hat{C} = \hat{Z}_p(\hat{\Lambda}_p - \hat{\sigma}^2 I)^{1/2} \hat{Q}$ becomes

$$\hat{C} = \hat{F} \hat{Q}(I + E).$$

Note that $E$ depends on $\hat{Q}$, and thus it is a function of the estimated autocovariance and cumulant matrices $\{\hat{A}_k : 1 \leq k \leq K\}$, as well as the weighting vector $\alpha$. We seek a first-order Taylor expansion of $\hat{C}$ with respect to the random quantities $\hat{F}$ and $\{\hat{A}_k : 1 \leq k \leq K\}$, evaluated at their true values $F$ and $\{A_k : 1 \leq k \leq K\}$. Here $F = Z_p(A_p - \sigma^2 I)^{1/2}$ depends on the eigenvectors and eigenvalues of the true (as opposed to sample) covariance of $\mathbf{x}$, and $\{A_k : 1 \leq k \leq K\}$ denote the true autocovariance and cumulant matrices of $\mathbf{y}$ from (2), with the estimated quantities replaced by their true values. The Taylor approximation for $\hat{C}_{ij}$ about the true values is

$$\hat{C}_{ij} \approx C_{ij} + \sum_{1 \leq i \leq p} \frac{\partial \hat{C}_{ij}}{\partial F_{r,s}} |_{\hat{F} = F, \{\hat{A}_k = \hat{A}_k, k = 1, 2, \ldots, K\}} (\hat{F}_{r,s} - F_{r,s}),$$

$$+ \sum_{1 \leq i \leq p} \frac{\partial \hat{C}_{ij}}{\partial \hat{A}_{k,r,s}} |_{\hat{F} = F, \{\hat{A}_k = \hat{A}_k, k = 1, 2, \ldots, K\}} (\hat{A}_{k,r,s} - A_{k,r,s}),$$

where the last two subscripts $r$ and $s$ on $F$ and $A$ indicate the $r$th row, $s$th column elements of the matrices.

To simplify the notation and facilitate further development, define $f \equiv vec(F)$, where vec denotes the vectorization operation of a matrix. Similarly, define $a \equiv vec(A_k : k = 1, 2, \ldots, K)$, except omit the redundant elements of the symmetric $A_k$. Combining all random quantities into a single vector $r \equiv [f \ a]$', (7) can be rewritten as

$$\hat{C}_{ij} \approx C_{ij} + \left( \frac{\partial \hat{C}_{ij}}{\partial f} |_{f = r} \right)'(\hat{r} - r).$$

Substituting (8) into (6) gives

$$MSE(\alpha) = \sum_{i=1}^n \sum_{j=1}^p E(\hat{C}_{ij} - C_{ij})^2$$

$$\approx \sum_{i=1}^n \sum_{j=1}^p \left( \frac{\partial \hat{C}_{ij}}{\partial f} |_{f = r} \right)' \text{cov}(\hat{r}) \left( \frac{\partial \hat{C}_{ij}}{\partial f} |_{f = r} \right),$$

where $\text{cov}(\hat{r})$ denotes the covariance matrix of $\hat{r}$. The expression for $MSE(\alpha)$ in (9) involves two components: the partial derivative vector $\partial \hat{C}_{ij}/\partial \hat{F}$, evaluated at $\hat{F} = r$, and the covariance $\text{cov}(\hat{r})$. Note that (9) depends on the weighting coefficients $\alpha$ only through the partial derivatives $\partial \hat{C}_{ij}/\partial \hat{F}|_{\hat{r}=r}$, expressions for which are straightforward to obtain (see App. A). The remaining task is to estimate or approximate the covariance matrix $\text{cov}(\hat{r})$. As seen from (A.3) and (A.4) in Appendix A, $\partial \hat{C}_{ij}/\partial \hat{F}|_{\hat{r}=r}$ is an explicit function of $\alpha$; therefore, the MSE in (9) also is an explicit function of $\alpha$.

Because the elements of $r$ are sample autocovariances, sample cumulants, and scaled eigenvectors of the sample covariance matrix, we might consider using asymptotic results to approximate $\text{cov}(\hat{r})$. For example, Anderson (1963) derived the asymptotic distribution of sample eigenvalues and eigenvectors for Gaussian data, and Davis (1977) extended the results to independent non-Gaussian data. But the asymptotic results do not apply to autocorrelated sources. Because of this and other complications, we have found that the bootstrapping procedure for estimating $\text{cov}(\hat{r})$ that we describe in Section 3.3 is more accurate and also quite straightforward to implement.

From Appendix A, the partial derivatives in (9) are functions of the true unknown quantities $F$, $Q$, and $d(k)$ ($1 \leq k \leq K$). Similarly, $\text{cov}(\hat{r})$ depends on certain unknown quantities, such as the true $F$. Consequently, in steps S5 and S6 of the iterative algorithm described in the next section, the estimates of $F$, $Q$, and $d(k)$ from the previous iteration are substituted for their true values in the approximate expressions for the partial derivatives of $\hat{C}_{ij}$ and $\text{cov}(\hat{r})$. 

TECHNOMETRICS, AUGUST 2008, VOL. 50, NO. 3

Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.
3.2 Summary of the Optimally Weighted Blind Source Separation Algorithm

Our optimal weighting algorithm selects \( \alpha \) to minimize the approximate expression for \( \text{MSE}(\alpha) \) in (9), with the partial derivatives (expressed as explicit functions of \( \alpha \)) calculated as in Appendix A and \( \text{cov}(\hat{f}) \) estimated as described in Section 3.3. In this section we outline the steps of the algorithm. Based on a sample of measurement data \( \{x_{i,j}, i = 1,2, \ldots, N\} \), the algorithm estimates \( \mathbf{C} = [c_{1}, c_{2}, \ldots, c_{p}] \) and \( \{v_{i,j}, i = 1,2, \ldots, N; j = 1,2, \ldots, p\} \), the interpretation of which provides insight into the spatial and temporal nature of the \( p \) variation sources.

Optimally Weighted BSS (OWBSS) Algorithm.

S1: From the sample data \( \{x_{i,j}, i = 1,2, \ldots, N\} \), calculate the sample covariance matrix \( \mathbf{\hat{S}}_{x} \).

S2: Perform PCA on \( \mathbf{\hat{S}}_{x} \) to yield \( p, \mathbf{\hat{Z}}_{p}, \mathbf{\hat{\Lambda}}_{p}, \hat{\sigma}^{2}, \hat{\mathbf{f}} = \mathbf{\hat{Z}}_{p}(\mathbf{\hat{\Lambda}}_{p} - \hat{\sigma}^{2}I)^{1/2} \), and then whiten the data through \( y_{i} = (\mathbf{\hat{Z}}_{p} - \hat{\sigma}^{2}I)^{-1/2}(\mathbf{\hat{Z}}_{p})x_{i}, i = 1,2, \ldots, N \).

S3: Select the maximum lag \( T \) and calculate the autocovariance matrices \( \{\mathbf{\hat{S}}_{y,:}; \tau = 1, \ldots, T\} \), the \( p \) third-order cumulant matrices \( \{\mathbf{\hat{M}}(i,:); 1 \leq i \leq p\} \), and the \( p^{2} \) fourth-order cumulant matrices \( \{\mathbf{\hat{M}}(i,j,:); 1 \leq i,j \leq p\} \), all of which compose the matrix set \( \{\mathbf{\hat{A}}_{k}; 1 \leq k \leq K\} \).

S4: Calculate an initial estimate \( \mathbf{\hat{Q}} \) as the minimizer of \( \sum_{k=1}^{K} \text{off}[U^{\top}\mathbf{\hat{A}}_{k}U] \) (i.e., with equal weighting). This yields initial estimates \( \mathbf{\hat{C}} = \mathbf{\hat{Z}}_{p}(\mathbf{\hat{\Lambda}}_{p} - \hat{\sigma}^{2}I)^{1/2}\mathbf{\hat{Q}} \) and \( v_{i} = (\mathbf{\hat{Q}}^{\top}y_{i}, i = 1,2, \ldots, N, \) of the variation pattern vectors and source signals. Take \( \mathbf{d}(k) \) to be the diagonal elements of \( \mathbf{\hat{Q}}^{\top}\mathbf{\hat{A}}_{k}\mathbf{\hat{Q}} \) for \( k = 1,2, \ldots, K \).

S5: Calculate the partial derivatives \( \partial\mathbf{\hat{Q}}_{i,j}/\partial f_{i,j} \) and \( \text{cov}(\hat{f}) \) into (9), yielding an approximate expression for \( \text{MSE}(\alpha) \) as an explicit function of \( \alpha \) (see A.3 and A.4 in App. A).

S8: Find the optimal weight vector \( \alpha^{*} = \arg\min_{\alpha} \text{MSE}(\alpha) \) using an optimization routine (we used the Optimization Toolbox in MATLAB), where the weighting coefficients for fourth-order cumulant matrices \( \mathbf{M}(i,j) \) and \( \mathbf{\hat{M}}(i,j) \) for each \( 1 \leq i \neq j \leq p \) are constrained to be small due to the symmetry.

S9: Update \( \mathbf{\hat{Q}} \) as the minimizer of \( \sum_{k=1}^{K} \alpha^{*}_{k}\text{off}[U^{\top}\mathbf{\hat{A}}_{k}U] \) and then update \( \mathbf{\hat{C}} = \mathbf{\hat{Z}}_{p}(\mathbf{\hat{\Lambda}}_{p} - \hat{\sigma}^{2}I)^{1/2}\mathbf{\hat{Q}} \) and \( \mathbf{d}(k) = k = 1,2, \ldots, K \).

S10: Check whether \( \mathbf{\hat{Q}} \) has converged. If it has, then terminate the algorithm with the current estimates from step S9. Otherwise, go to step S5.

We have found that the algorithm usually converges in two or three iterations of steps S5–S10 using the convergence criterion \( \|\mathbf{\hat{Q}}^{\text{new}} - \mathbf{\hat{Q}}^{\text{old}}\| < 0.01p \), where \( \|\cdot\| \) is the Frobenius norm of a matrix. In other words, we usually arrive at the final optimal weights in only one or two iterations of steps S5–S10.

3.3 Markov Bootstrap Estimation of \( \text{cov}(\hat{f}) \)

In this section we describe a Markov bootstrap procedure for estimating \( \text{cov}(\hat{f}) \) in step S6 of the OWBSS algorithm. On each bootstrap replicate, we must construct a bootstrapped sample for \( \{x_{i,j}; i = 1,2, \ldots, N\} \). Instead of directly bootstrapping the measurement data \( x \), we bootstrap the estimated variation sources \( v \) and then construct \( x \) through the model (1), with \( w \) sampled from its assumed multivariate normal distribution (mean 0 with covariance matrix \( \hat{\sigma}^{2}I \), and no temporal autocorrelation). Bootstrap sampling of the estimated variation sources is much more robust than bootstrap sampling of \( x \), because \( v \) is of much lower dimension than \( x \), and the components of \( v \) are assumed to be statistically independent. As a result, each component of \( v \) can be bootstrapped individually.

Because the sources may be temporally autocorrelated, we use the Markov bootstrap procedure of Paparoditis and Politis (2002), which we describe in Appendix B, to draw bootstrap samples of \( N \) values of \( v \) from the estimated variation sources calculated in step S9 of the OWBSS algorithm. Note that conventional bootstrapping destroys the temporal autocorrelation of the sources. Paparoditis and Politis (2002) have shown certain convergence (asymptotically in \( B \) and \( N \)) properties of the Markov bootstrap estimated quantities if the sources are truly Markov with known order. But such convergence cannot be guaranteed in the situation described in this article, because the OWBSS algorithm assumes no underlying models, Markov or otherwise, for the temporal correlation of the sources. By this, we mean that implementation of the method does not depend on a specific assumed dynamic model structure, such as AR or MA, for the sources. The sources that we use in the simulation Examples 2–5 in Section 5 (see Table 1) are all Markov of order either 0 or 1, whereas the sources in Example 1 of Section 5 are not Markov of any order. Despite this, using a Markov order \( d = 5 \) (see App. B) appears to provide reasonably effective bootstrap sampling in all of the simulation examples. Perhaps the essential underlying requirement for the Markov bootstrap procedure to be effective is that the characteristics of the source distributions can be reasonably approximated as Markov of a specified order. This is the case for the MA sources in Example 1, as it would be for many variation sources that could be expected in manufacturing processes. Keep in mind that \( \text{cov}(\hat{f}) \) is the only bootstrapped quantity used in the OWBSS algorithm, and it is used only for selecting the optimal weights.

Note that the Markov bootstrap procedure preserves the marginal distributions of the sources, which is crucial because non-Gaussian behavior is an important characteristic when considering the weighting for the third- and fourth-order cumulants. It would be difficult to preserve the marginal distributions using parametric bootstrap methods based on, for example, autoregressive moving average (ARMA) models. Another bootstrap option that would preserve marginals is the block bootstrap method of Künsch (1989), in which bootstrap samples are drawn as blocks of consecutive observations instead of as \( N \) individual observations. Block bootstrap methods may be attractive if sample sizes are sufficiently large to allow large block sizes. Considering the nature of typical applications in manufacturing variation diagnosis (e.g., typical sample sizes and whether sources may be reasonably approximated as Markov),
however, we believe that Markov bootstrapping is more suitable than block bootstrapping.

In the following bootstrapping procedure, the superscript \( b \) indicates quantities from the \( b \)th bootstrap replicate, for \( b = 1, 2, \ldots, B \), where \( B \) denotes the total number of bootstrap replicates. Repeating the following steps A1–A6 a total of \( B \) times (e.g., \( B = 10,000 \)) produces the \( B \) estimates \( \{ \hat{s}^b : b = 1, 2, \ldots, B \} \). We can then approximate \( \text{cov}(\hat{\phi}) \approx \sum_{b=1}^{B} (\hat{s}^b - \bar{s})(\hat{s}^b - \bar{s})' / (B-1) \sum_{b=1}^{B} \hat{s}^b \hat{s}^b' \), where \( \bar{s} = 1 / B \sum_{b=1}^{B} \hat{s}^b \), for use in step S6 of the OWBSS algorithm. The detailed steps involved for each replicate \( (b = 1, 2, \ldots, B) \) of the bootstrapping procedure are as follows:

A1: For each \( j = 1, 2, \ldots, p \), use the Markov bootstrap procedure described in Appendix B to obtain the bootstrapped sample \( \{ v^b_j : i = 1, 2, \ldots, N \} \) from the empirical distribution of the estimated scores \( \{ \hat{v}_i, i = 1, 2, \ldots, N \} \) from step S9 of the previous iteration of the OWBSS algorithm (or from step S4 on the first iteration).

A2: For \( i = 1, 2, \ldots, N \), calculate \( x^b_i = \hat{C}v^b_i + w^b_i \), where \( \hat{C} \) from step S9 of the previous iteration of the OWBSS algorithm (or from step S4 on the first iteration), \( v^b_i = [v^b_{i1}, v^b_{i2}, \ldots, v^b_{ip}]' \), and \( w^b_i \) is randomly generated from the multinormal distribution with mean vector \( 0 \) and covariance matrix \( \hat{\Sigma}^2 I_{nxn} \).

A3: Obtain \( \hat{A}^b_p, \hat{Z}^b_p, \) and \( \hat{\delta}^{2b} \) by performing PCA on \( \hat{Z}^b_p \), the sample covariance matrix of \( \{ x^b_i : i = 1, 2, \ldots, N \} \).

A4: Whiten the data through \( y^b_i = (\hat{A}^b_p - \hat{\delta}^{2b} I)^{-1/2} (\hat{Z}^b_p - \hat{A}^b_p G_{\hat{H}}) \) for \( i = 1, 2, \ldots, N \), where \( R^b = G_{\hat{H}} \). The singular value decomposition of \( (\hat{Z}^b_p)' / \hat{Z}^b_p \). (See the discussion in the next paragraph for why \( R^b \) is included in the whitening step.)

A5: Calculate the sample autocovariance matrices \( \{ \hat{\Sigma}^b_{xy} : 1 \leq \tau \leq T \} \), third-order cumulant matrices \( \{ M^b(i) : 1 \leq i \leq p \} \), and fourth-order cumulant matrices \( \{ M^b(i,j) : 1 \leq i,j \leq p \} \) for the whitened bootstrap data \( y^b_i = [y^b_{i1}, y^b_{i2}, \ldots, y^b_{ip}] \).

A6: Form the vector \( \hat{\phi}^b \) from the quantities calculated in step A5.

The reason why the whitening equation in step A4 is modified to include the orthogonal matrix \( R^b \) is to avoid the inflation of \( \text{cov}(\hat{\phi}) \) that occurs when some of the \( p \) dominant eigenvalues of the sample covariance matrix \( \hat{\Sigma} \) are repeated or nearly repeated. When eigenvalues are repeated, the eigenvectors are not unique, because any set of orthonormal vectors in the eigenspace associated with repeated eigenvalues qualifies. This creates difficulties when estimating elements of \( \text{cov}(\hat{\phi}) \) that correspond to eigenvectors associated with repeated eigenvalues. The eigenvectors are inflated, because the estimated eigenvectors \( \hat{Z}^b_p \) vary widely from bootstrap replicate to bootstrap replicate due to their nonuniqueness. Therefore, to provide a more useful estimate of \( \text{cov}(\hat{\phi}) \), \( \hat{Z}^b_p \) is replaced by \( \hat{Z}^b_p R^b \) in step A4, where \( R^b \) is the orthogonal matrix that minimizes the difference between \( \hat{Z}^b \) (representing the "true" eigenvectors within the context of the bootstrapping procedure) and \( \hat{Z}^b_p R^b \). Appendix C derives the singular value decomposition solution for \( R^b \) used in step A4. The operation \( \hat{Z}^b_p R^b \) may be viewed as orthogonally transforming the bootstrap eigenvectors \( \hat{Z}^b_p \) so that they are as closely matched with \( \hat{Z}^b_p \) as possible. Note that in the hypothetical situation where \( \hat{Z}^b_p \) differs from \( \hat{Z}^b_p \) only because of nonuniqueness within the common eigenspaces for repeated eigenvalues and other estimation errors are ignored, then \( \hat{Z}^b_p \) will coincide exactly with \( \hat{Z}^b_p \) after the orthogonal rotation.

In step A4 we orthogonally transform all of the \( p \) dominant eigenvectors, including those that correspond to distinct eigenvalues. Alternatively, we might consider using statistical tests to identify groups of nearly repeated eigenvalues and then orthogonally transforming the eigenvectors only within the smaller associated eigenspaces. We recommend against this for robustness reasons. Moreover, we have found that such statistical tests are largely unnecessary; the submatrix of \( R^b \) that corresponds to distinct eigenvalues is typically quite close to the identity matrix, which leaves the eigenvectors associated with distinct eigenvalues largely unchanged.

We also point out that the rotation of the eigenvectors by \( R^b \) is done only within the bootstrapping portion of the algorithm and is not done in, for example, step S2 of the OWBSS algorithm. The nonuniqueness due to repeated eigenvalues is a problem only when attempting to bootstrap estimate \( \text{cov}(\hat{\phi}) \) and causes no other problems inherent to BSS when estimating \( C \).

4. A CRANKSHAFT MANUFACTURING EXAMPLE

In this section we illustrate the use of the algorithm for blindly identifying dimensional variation patterns in a crankshaft manufacturing process. In Section 5 we use Monte Carlo simulation to compare the performance of different algorithms for a benchmark simulation example. Figure 1 shows the same automotive crankshaft originally considered by Apley and Lee (2003), which consists of five main bearings (mains 1–5) and four pin bearings (pins 1–4). The bearing diameters are measured at three locations along each main bearing and five locations along each pin bearing, indicated by the "*" symbols in Figure 1. Therefore, the dimension of the measurement vector \( x \) on each crankshaft is \( n = 35 \). Because the sample mean of \( x \) is subtracted from the data, we can view \( x \) as the difference between the measured diameters and their mean values. Based on measurements of \( N = 247 \) crankshafts, Apley and Lee (2003) estimated that \( p = 3 \) significant variation sources were present in the sample of data.

Figures 2–4 graphically illustrate the estimated pattern vectors \( \hat{c}_1, \hat{c}_2, \) and \( \hat{c}_3 \) using the OWBSS algorithm with \( T = 10 \). Each element of a pattern vector is represented as an arrow at the location of the corresponding diameter measurement. The length of the arrow is proportional to the magnitude of the corresponding element of the pattern vector, and the direction of the arrow represents the sign (pointing out of the crankshaft for a positive element and into the crankshaft for a negative element). For visual convenience, we omitted arrows for elements that were negligibly small in magnitude. The estimated pattern vectors are quite close to the estimates of Apley and Lee (2003), which were obtained using only fourth-order cumulants. Although the estimates are quite similar for this example, the two methods may produce very different results in
general. This is evident from the performance comparison of
the JADE algorithm (which was used in Apley and Lee 2003)
and the OWBSS algorithm given in Section 5.

Apley and Lee (2003) have provided a detailed discussion
of the root causes of the three patterns. The first pattern \( \mathbf{c}_1 \),
which is shown in Figure 2, is particularly interesting, because
it accounted for 39.3\% of the total variation in all 35 mea-
sured features over the data sample. This variation pattern
affects only the main bearings and represents all bearing diame-
ters across the first four mains increasing or decreasing together
from crankshaft to crankshaft (because the arrows are all in the
same direction). The diameters on mains 2 and 3 increase or
decrease by roughly twice the amount of the diameters on mains
1 and 4 (because the arrows are roughly twice as long). After
these patterns were discussed with engineering, the suspected
root cause was identified as elastic bending of the crankshaft
during rough-cut machining, due to the large cutting forces
generated. The shaft was supported at both ends in chucks, but the
middle of the shaft was not kept from flexing under the cutting
forces. This example illustrates how graphically visualizing the
blindly identified variation patterns can support the ultimate ob-
jective of identifying and eliminating major root causes of man-
ufacturing variation.

5. PERFORMANCE COMPARISON
AND DISCUSSION

In this section we investigate the performance of different
schemes for weighting the autocovariance and higher-order cu-
mulants, in terms of the accuracy in estimating \( \mathbf{C} \). We compare
all of the approaches using the same benchmark simulation ex-
ample considered by Apley and Lee (2003) and Lee and Apley
(2004). In the example the manufactured part is a simple beam
with \( n = 20 \) dimensional measurements distributed uniformly
over its length. The part is affected by three distinct variation

![Figure 2. Estimate of (a) the first pattern vector \( \mathbf{c}_1 \) and (b) the source signal \( v_{i,1} \).](image)
patterns present simultaneously in the sample: the corresponding \( c_1 \), \( c_2 \), and \( c_3 \), as illustrated in Figure 5. If we view the part as a subcomponent of a larger assembly, then the three patterns may represent a rotation of the part about its midpoint (\( c_1 \)), an up/down translation of the part (\( c_2 \)), and a bending of the part about its midpoint (\( c_3 \)).

For this set of spatial variation patterns, we considered the five different source distributions listed in Table 1. The distributions in Table 1 were chosen not necessarily to represent any particular manufacturing process, but to illustrate differences between various algorithms and when they can be expected to perform well. For the ARMA sources, the Gaussian

Figure 3. Estimate of (a) the second pattern vector \( c_2 \) and (b) the source signal \( v_{i,2} \).

Figure 4. Estimate of (a) the third pattern vector \( c_3 \) and (b) the source signal \( v_{i,3} \).
ARMA(1, 1) model is $v_t - \phi v_{t-1} = a_t - \theta a_{t-1}$, with $a_t$ an iid Gaussian time series. The AR(1) model is a special case with $\theta = 0$, and the MA(1) model is a special case with $\phi = 0$. Example 5 considers two sources that follow skewed distributions and demonstrates the benefit of including third-order cumulants. For all examples, the sample size was $N = 250$, and the maximum autocovariance lag was $T = 10$.

### 5.1 Performance Comparison

The performance measure, denoted by $J$, was the aggregated squared estimation error $\sum_{t=1}^{T} \sum_{j=1}^{J} (\hat{C}_{ij} - C_{ij})^2$ averaged over all replicates in the Monte Carlo simulation. Although we might view the similarity between the performance measure and the MSE optimization criterion as giving an unfair advantage to the OWBSS algorithm, we believe the foregoing performance measure is the most relevant one for manufacturing variation diagnosis. The ultimate objective is to enable a process engineer to gain insight into the root causes of the variation by interpreting the columns of $\hat{C}$. Thus the difference between $\hat{C}$ and $C$ is of primary importance. In fact, the primary reason why we chose the MSE optimization criterion for the OWBSS algorithm is because it most closely reflected the foregoing performance measure, which also was used by Apley and Lee (2003) and Lee and Apley (2004).

Table 2 displays the results of the Monte Carlo simulation comparing the OWBSS algorithm with the five methods listed in the table. The JADE algorithm is that of Cardoso and Souloumiac (1993), which essentially minimizes (3) with equal weighting coefficients and including only fourth-order cumulant matrixes. The WASOBI algorithm is that of Doron and Yeredor (2004), which considers only autocovariance and weights each individual element of the autocovariance matrixes differently, according to a weighted least squares criterion. The Lee and Apley (2004) algorithm minimizes (3) with only autocovariance and fourth-order cumulant matrixes included and with weights determined as described in Section 3. The JADE$_{PC}$ algorithm is that of Müller et al. (1999), which minimizes (3) with only autocovariance and fourth-order cumulant matrixes included and with equal weighting. Finally, the COMBI algorithm is that of Tichavský et al. (2006), which uses a hybrid combination of the WASOBI algorithm and the EFICA algorithm of Koldovský, Tichavský, and Oja (2006). The COMBI algorithm uses WASOBI to separate the sources that appear to be better separated using autocovariance and EFICA to separate the sources that appear to be better separated using fourth-order cumulants.

One conclusion that can be drawn from Table 2 is that OWBSS performs either the best, or close to the best, for each example that we considered; for example, it performs close to WASOBI for Examples 1 and 2, in which all sources are Gaussian. With Gaussian sources, third- and fourth-order cumulants provide no useful information, and the sources must be separated based only on their autocovariance. In this situation WASOBI has been shown to have certain asymptotic optimality properties (Doron and Yeredor 2004). For Examples 1 and 2, the performance of OWBSS is roughly 15% and 8% worse than that of WASOBI. Note that the performance of WASOBI and COMBI depends heavily on the maximum AR order, which is a parameter that must be specified before running the algorithm. For Examples 3–5, in which at least one source is non-Gaussian, OWBSS performs the best, sometimes substantially better. The next best algorithm in these examples is either JADE$_{TD}$ (Example 3) or JADE (Examples 4 and 5), which perform roughly 13%, 4%, and 45% worse than OWBSS in Examples 3–5.

In terms of overall effectiveness across the five examples considered, we might conclude from Table 2 that OWBSS is the most robust, with COMBI coming in second. COMBI appears to perform slightly better than OWBSS in examples for which all sources are Gaussian (7% better in Example 1 and 12% better in Example 2), but substantially worse with non-Gaussian sources (21% worse in Example 3, 100% worse in Example 4, and 195% worse in Example 5). The most likely reason why COMBI performs better than OWBSS for the pure Gaussian examples is that it relies almost exclusively on the asymptotically optimal WASOBI algorithm in these cases.

<table>
<thead>
<tr>
<th>Example</th>
<th>$v_1$ distribution</th>
<th>$v_2$ distribution</th>
<th>$v_3$ distribution</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Gaussian MA(1)</td>
<td>Gaussian MA(1)</td>
<td>Gaussian ARMA(1, 1)</td>
</tr>
<tr>
<td></td>
<td>with $\theta = 1$</td>
<td>with $\theta = -.21$</td>
<td>with $\phi = .95$, $\theta = -.84$</td>
</tr>
<tr>
<td>2</td>
<td>Gaussian AR(1)</td>
<td>Gaussian AR(1)</td>
<td>Gaussian iid</td>
</tr>
<tr>
<td></td>
<td>with $\phi = .9$</td>
<td>with $\phi = -.9$</td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>Gaussian AR(1)</td>
<td>Gaussian AR(1)</td>
<td>binary iid</td>
</tr>
<tr>
<td></td>
<td>with $\phi = .9$</td>
<td>with $\phi = -.7$</td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>Uniform(−1, 1)</td>
<td>Uniform(−1, 1)</td>
<td>binary iid</td>
</tr>
<tr>
<td></td>
<td>with $\phi = .9$</td>
<td>with $\phi = -.7$</td>
<td></td>
</tr>
<tr>
<td>5</td>
<td>Chi-squared</td>
<td>Extreme value</td>
<td>binary iid</td>
</tr>
<tr>
<td></td>
<td>distribution with</td>
<td>distribution with</td>
<td></td>
</tr>
<tr>
<td></td>
<td>4 df</td>
<td>mean 0 and standard</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>deviation 1</td>
<td></td>
</tr>
</tbody>
</table>
5.2 The Value of Adding Third-Order Cumulants

To illustrate the benefit of including third-order cumulants, we compared two versions of the OWBSS algorithm applied to Examples 4 and 5. The first version, designated OWBSS2,3,4, is the version described previously, which considers autocovariance, third-order cumulants, and fourth-order cumulants. The second version, designated OWBSS2,4, is identical, except that third-order cumulants are excluded. The results are given in Table 3.

For Example 5, in which two sources have skewed distributions, including the third-order cumulants clearly results in a significant improvement. For Example 4, in which all source distributions are symmetric, so that third-order cumulants provide no useful information, the two OWBSS versions perform roughly the same. That the OWBSS2,3,4 performance is no worse than the OWBSS2,4 performance in Example 4 and much better than it in Example 5 indicates that the OWBSS algorithm is doing a proper job of weighting the different BSS criteria. The third-order cumulants are weighted heavily in Example 5 and lightly in Example 4.

5.3 Other Issues

5.3.1 Computational Expense. The computational expense of the OWBSS is much greater than that of any of the other algorithms compared in Table 2, largely because of the bootstrap estimation of cov(Φ) in step S6. To give an idea of the computational expense, for the crankshaft manufacturing example of Section 4 with $B = 2,000$ bootstrap replicates, executing the OWBSS algorithm took roughly 100 seconds in MATLAB on a desktop computer (Pentium 4, 3.00-GHz CPU, and 1 G RAM).

The computational burden is obviously prohibitive for most real-time signal processing applications of BSS; however, it may be quite reasonable for many offline applications. For manufacturing variation diagnosis, for example, a process engineer might sit down once each day to run the BSS algorithm and diagnose the current major sources of process variation. In this case a computation time of 100 seconds or more certainly would not preclude using the OWBSS algorithm. Recall that the objective of BSS in manufacturing variation diagnosis is to graphically illustrate the estimated pattern vectors, providing a process engineer with insight into the root causes of variation. Achieving the most accurate estimation possible is well worth the additional computational expense, considering that an attempt to interpret inaccurately estimated pattern vectors is far more costly in terms of wasted time.

5.3.2 Indeterminacy in the Estimate of C. The estimate of C is clearly indeterminate regarding permutations and sign changes of its columns. Note that there is no indeterminacy in the scaling of its columns, because we have defined the sources as unit variance. The permutation and sign change indeterminacies present no problems when using BSS for diagnosing manufacturing variation, because there is no inherent ordering of the sources, and ±ci are equally interpretable characterizations of the spatial nature of the ith variation source. But they do present an obstacle in the context of the Monte Carlo simulation comparison, because we must compare each Ĉ with the true C when calculating the performance measure J. In the Monte Carlo simulation, we handled these indeterminacies in the following way. After calculating Ĉ on each Monte Carlo replicate, we found the signed permutation matrix P (i.e., a permutation matrix with nonzero elements taking values of ±1) that minimizes $\sum_{i=1}^{n} \sum_{j=1}^{p} |C_{ij} - \tilde{C}_P|_2^2$, and then replaced Ĉ with ĈP. (See Tichavský and Koldovský 2004 for an alternative method for handling a related problem.)

6. CONCLUSION

Various BSS criteria, including autocovariance and higher-order cumulants, may contain useful information that can aid in blindly identifying the spatial and temporal characteristics of manufacturing variation patterns. We have presented an approach for optimally combining the different criteria to minimize an aggregate measure of the MSE in estimating the variation pattern vectors. The simulation examples demonstrate that the OWBSS algorithm effectively balances weight among the
various BSS criteria, depending on what information the data indicate is useful for separating the sources. In terms of overall estimation accuracy across the various examples considered, the OWWBSS algorithm compares favorably with existing algorithms, and its performance is relatively robust to different types of variation sources that might be encountered in manufacturing. Consequently, the OWWBSS algorithm is easy for a practitioner to use, in the sense that the algorithm need not be tuned to optimize its effectiveness for the data set at hand. The only parameter that must be chosen is the maximum autocovariance lag \( T \). For this, we recommend choosing a relatively large \( T \) (e.g., 10–20), with the expectation that the OWWBSS algorithm will place little weight on high-lag autocovariances if they appear to contain little useful information.

**ACKNOWLEDGMENTS**

This work was supported by National Science Foundation grants CMMI-0554824 and CMMI-0826081. The authors thank the anonymous referees and associate editor for their many helpful comments.

**APPENDIX A: EXPRESSIONS FOR THE PARTIAL DERIVATIVES OF \( \mathbf{C} \)**

Here we derive expressions for \( \partial \hat{C}_{ij} / \partial \hat{\mathbf{A}}_{kr} \) for a fixed \( \alpha \), for use in step 5 of the OWWBSS algorithm. Note that the elements of \( \hat{\mathbf{F}} \) are defined as the elements of \( \hat{\mathbf{F}} = \hat{\mathbf{Z}}_p(\hat{\mathbf{A}} - \hat{\sigma}^2 \mathbf{I})^{1/2} \) and of the sample autocovariance and cumulant matrices \( \{ \hat{\mathbf{A}}_k : 1 \leq k \leq K \} \). Writing the elements of \( \hat{\mathbf{C}} = \hat{\mathbf{F}} \hat{\mathbf{Q}} \) as \( \hat{C}_{ij} = \sum_{k=1}^{p} \hat{F}_{ik} \hat{Q}_{kj} \) and noting that \( \hat{\mathbf{Q}} \) is a function of \( \{ \hat{\mathbf{A}}_k : 1 \leq k \leq K \} \) [see eq. (3)], it follows that

\[
\frac{\partial \hat{C}_{ij}}{\partial \hat{F}_{kr}} = \begin{cases} Q_{ij} & \text{if } i = r \\ 0 & \text{if } i \neq r \end{cases}
\]

\[
\frac{\partial \hat{C}_{ij}}{\partial \hat{A}_{kr}} = \sum_{i=1}^{p} F_{ir} \frac{\partial \hat{Q}_{kj}}{\partial \hat{A}_{kr}} , \quad r = 1, 2, \ldots, s
\]

From (4), we have that

\[
\frac{\partial \hat{Q}_{ij}}{\partial \hat{A}_{kr}} = \sum_{k=1}^{p} Q_{hk} \frac{\partial \hat{E}_{hj}}{\partial \hat{A}_{kr}} , \quad r = 1, 2, \ldots, s
\]

Combining (A.1) and (A.2) gives

\[
\frac{\partial \hat{C}_{ij}}{\partial \hat{A}_{kr}} = \sum_{i=1}^{p} F_{ir} \sum_{h=1}^{p} Q_{hk} \frac{\partial \hat{E}_{hj}}{\partial \hat{A}_{kr}} . \quad r = 1, 2, \ldots, s
\]

Finally, the following expression for \( \partial \hat{E}_{hj} / \partial \hat{A}_{kr} \) follows from (5) and the symmetry of \( \{ \hat{\mathbf{A}}_k : 1 \leq k \leq K \}:

\[
\frac{\partial \hat{E}_{hj}}{\partial \hat{A}_{kr}} = \begin{cases} \alpha (d_{j}(k) - d_{h}(k)) (Q_{hj} Q_{kj} + Q_{kh} Q_{hj}) & \text{if } r = s, h \neq j \\ \sum_{l=1}^{p} \alpha_l (d_{j}(l) - d_{h}(l))^2 & \text{if } r \neq s, h \neq j \\ \sum_{l=1}^{p} \alpha_l (d_{j}(l) - d_{h}(l))^2 & \text{if } r = s, h \neq j \\ 0 & \text{if } h = j \end{cases}
\]

**APPENDIX B: MARKOV BOOTSTRAP METHOD**

We summarize the Markov bootstrap approach of Paparoditis and Politis (2002) for resampling from a non-iid, stationary, univariate sample of data \( \{y_i : i = 1, 2, \ldots, N\} \) to form the bootstrapped sample \( \{y'_i : i = 1, 2, \ldots, N\} \). This is used in step A1 of the algorithm given in Section 3.3. To illustrate the approach, first suppose that the conditional distribution of \( y_i+1 \) given \( y_1, y_2, \ldots, y_i \) depends only on \( y_i \), that is, a first-order Markov dependency. After generating \( y'_i \), the next observation, \( y'_{i+1} \), is drawn as \( y_i \) with probability \( K_h(y_i - y'_i) / \sum_{r=1}^{N} K_h(y_r - y'_i) \), for \( i = 1, 2, 3, \ldots, N \). We must choose the kernel function \( K_h(\cdot) \), for example, \( K_h(\cdot) = g(\cdot)/h \), where \( h \) is the smoothing bandwidth and \( g \) denotes the standard normal density function. The initial observation \( y'_1 \) can be drawn arbitrarily from the original data series. For all of the examples in this article, we used the smoothing bandwidth \( h = (\frac{1}{20})^{1/5} s \), where \( s \) denotes the sample standard deviation of \( \{y_i - \bar{y} : i = 1, 2, \ldots, N\} \), as suggested by Bowman and Azzalini (1997). To bootstrap a sample with \( d \)-order Markov dependency, the procedure is the same, except that a multivariate standard normal density function is used for \( K_h(\cdot) \). Specifically, \( y'_i \) is drawn as \( y_i \) with the probability \( K_h(u_i - u'_i) / \sum_{r=1}^{N} K_h(u_r - u'_i) \), for \( i = d + 1, 2, \ldots, N \), where \( u_i = (y_i, y_{i-1}, \ldots, y_{i-d+1})' \), and \( u'_i = (y'_i, y'_{i-1}, \ldots, y'_{i-d+1})' \). For a given data set, we could use the approach suggested by Auestad and Tjøstheim (1990) to estimate a suitable \( d \).

**APPENDIX C: THE SOLUTION FOR \( \mathbf{R}^b \) IN STEP A4 OF THE BOOTSTRAPPING ALGORITHM**

This problem can be formulated as minimizing

\[
\text{tr}(\hat{Z}_p - \hat{Z}_p R^b)' (\hat{Z}_p - \hat{Z}_p R^b)
\]

\[
\text{tr}(\hat{Z}_p R^b) + \text{tr}(\hat{Z}_p (\hat{Z}_p)' R^b) - 2 \text{tr}(\hat{Z}_p' \hat{Z}_p R^b)
\]

subject to \( R^b(R^b)' = I \), where \( \text{tr}(\cdot) \) denotes the trace operator. Because only the last term depends on \( R^b \), this is equivalent to maximizing \( \text{tr}(\hat{Z}_p R^b) \) subject to \( R^b(R^b)' = I \). Using a Lagrange multiplier formulation, we maximize \( \text{tr}(\hat{Z}_p R^b) - \frac{1}{2} \lambda (R^b(R^b)' - I) \), where \( \lambda \) is a symmetric matrix of Lagrange multipliers. Setting the derivative of this expression with respect to \( R^b \) to equal to 0, it follows that the solution must satisfy \( (\hat{Z}_p)' \hat{Z}_p = \lambda R^b \). Using the singular value decomposition \( \hat{Z}_p = G S \hat{H} \), where \( G \) and \( H \) are \( p \times p \) orthogonal matrices and \( S \) is the diagonal matrix of singular values, the optimality condition becomes \( G S \hat{H}' = \lambda \). It can be verified that \( \lambda = G S \lambda' \) and \( R^b = \hat{H} \). Thus, a solution.

[Received May 2006. Revised March 2008.]

**REFERENCES**


