A Statistical Learning Approach for the Design of Polycrystalline Materials

Veera Sundararaghavan[1] **and Nicholas Zabaras**[2]*

[1] *Department of Aerospace Engineering, University of Michigan, Ann Arbor, MI 48109, USA*

[2] *Materials Process Design and Control Laboratory, Sibley School of Mechanical and Aerospace Engineering, Cornell University, Ithaca, NY 14853-3801, USA*

Received 15 February 2008; revised 29 July 2008; accepted 18 August 2008
DOI:10.1002/sam.10017
Published online 16 January 2009 in Wiley InterScience (www.interscience.wiley.com).

Abstract: Important physical properties such as yield strength, elastic modulus, and thermal conductivity depend on the material microstructure. Realization of optimal microstructures is important for hardware components in aerospace applications where there is a need to optimize material properties for improved performance. Microstructures can be tailored through controlled deformation or heat treatment. However, identification of the optimal processing path is a non-trivial (and non-unique) problem. Data-mining techniques are eminently suitable for process design since optimal processing paths can be selected based on available information from a large database-relating processes, properties, and microstructures. In this paper, the problem of designing processing stages that lead to a desired microstructure or material property is addressed by mining over a database of microstructural signatures. A hierarchical X-means classifier is designed to match crystallographic orientation features to a class of microstructural signatures within a database. Instead of the conventional distortion minimization algorithm of k-means, X-means maximizes a Bayesian information measure for calculating cluster centers which allows automatic detection of number of classes. Using the microstructural database, an adaptive data-compression technique based on proper orthogonal decomposition (POD) has been designed to accelerate materials design. In this technique, reduced modes selected adaptively from the database are used to speed up auxiliary microstructure optimization algorithms built over the database. © 2009 Wiley Periodicals, Inc. Statistical Analysis and Data Mining 1: 306–321, 2009

Keywords: microstructure database; materials-by-design; X-means classification; adaptive proper orthogonal decomposition

1. INTRODUCTION

Many engineering materials are polycrystalline in nature. While we commonly work with metallic components ranging from few centimeters to several meters in size, material properties such as elastic modulus, strength, and thermal conductivity are dependent on the substructure of the material at lengths of few hundred microns and below called the 'microstructure'. As shown in Fig. 1, the microstructure of metallic materials are composed of an aggregate of minuscule grains (or crystals) of various shapes and sizes. Each grain is distinguished by its crystallographic orientation, which quantifies the spatial orientation of its atomic lattice (as shown in Fig. 1). The large variety of microstructural arrangements that is possible can be classified according to either topological descriptors such as grain sizes and shapes, or through orientation descriptors such as the probability density function for orientations: the orientation distribution function (ODF). Microstructural feature–property relationships are quite complex and data-mining techniques are well suited to identify such relationships [1,2].

Microstructures can be tailored so that desired properties can be achieved through controlled deformation or thermal treatment. We are interested in the problem of 'processing path design' to realize microstructures with optimal properties. Processing path design involves identification of an optimal deformation sequence that takes the microstructure of a raw material to a desired microstructure in the final component. Because of non-uniqueness in processing path solution (different processing paths leading to similar microstructural features) and complex nature of the microstructure–property–process relationships, this problem cannot be addressed solely using conventional optimization schemes. The use of data-mining strategies comes natural to such problems. Of particular interest is the development of a database that can accommodate unknown microstructures into newly formed classes without user intervention.

Correspondence to: Nicholas Zabaras (zabaras@cornell.edu)

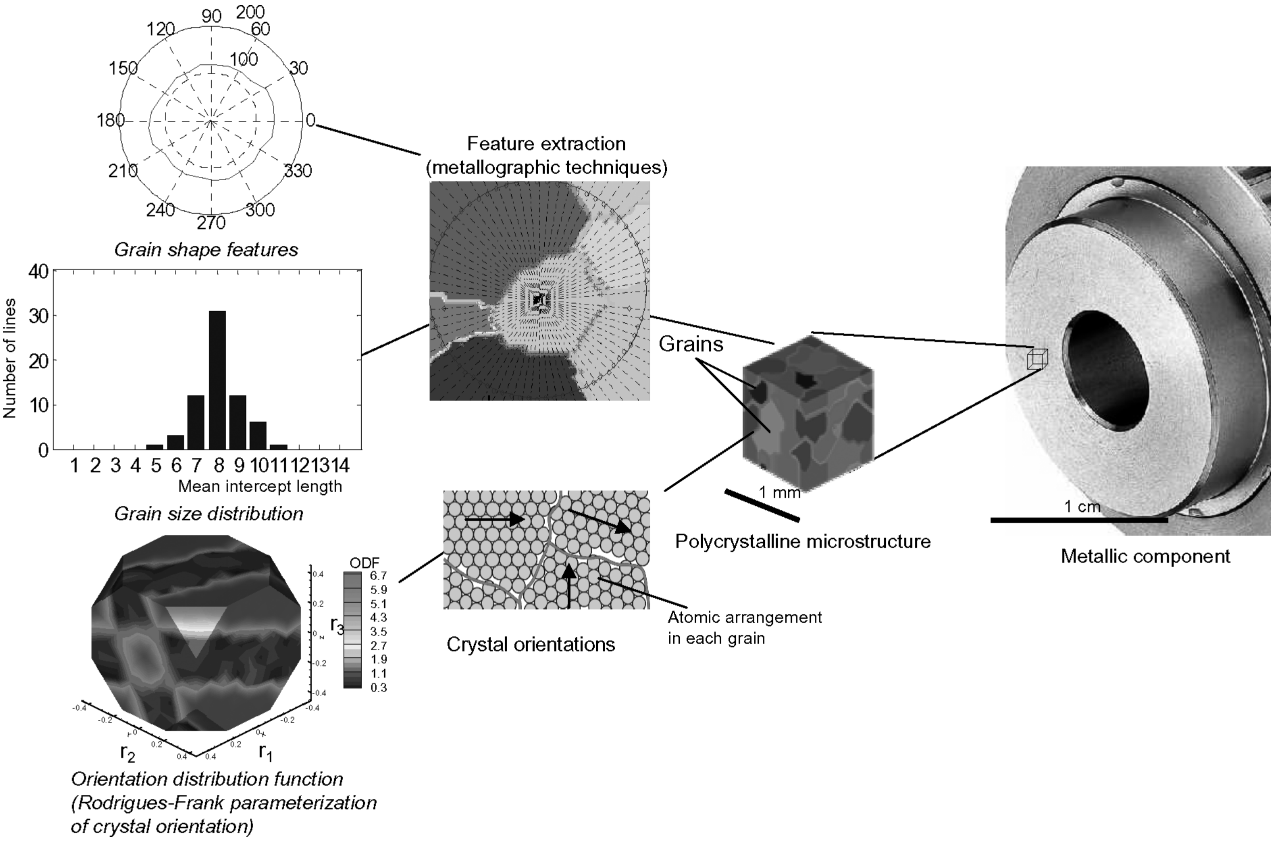

Fig. 1 Microstructure of a metallic part is polycrystalline (i.e. an aggregate of grains). Features of a polycrystalline microstructure include grain sizes/shapes (topological features) and crystal orientation (related to atomic arrangement in each crystal). These features can be quantified using rose of intersections (grain shape feature), intercept histograms (grain size measure), or through ODF (obtained from diffraction measurements). Orientation distribution function provides a color map of volume fraction of crystals associated with each crystal orientation (r_1, r_2, r_3). The orientation space is constrained to account for crystal and sample symmetries, and results in a truncated cube as shown above.

Attractiveness of k-means [3] algorithm for unsupervised classification lies in its simplicity, and in its local-minimum convergence properties. The drawback of the method is that the number of classes k has to be supplied by the user. In the case of a microstructure database, it is often impossible to anticipate the number of classes due to the complexity of microstructural features and a large variety of microstructures. We employ a modified version of k-means, called the X-means algorithm [4], that estimates the number of classes automatically by maximizing an information criterion instead of minimizing a distortion measure as in k-means. X-means involves an iterative procedure wherein after a run of k-means algorithm, local decisions are made about which classes must be split to better fit the data. The splitting decision is done by computing the Bayesian information criterion (BIC) [5].

Multi-level classification is a necessity for microstructure databases since microstructural features are of different dimensions, but need to be weighted equally for classification. For example, a 10×1 grain shape feature vector and a 1000×1 ODF cannot be represented within a single feature vector since the ODF information would be weighted 100 times more than the grain shape information in the adopted distance measure. Hierarchical classification involves classification with a single feature at each level leading to unbiased classification. A hierarchical classifier of this type is shown in Fig. 2. Here, microstructures are classified based on the grain shape measure in the first level of classification, followed by grain size measure in the second level. The hierarchical classifier is designed so that newer features can be introduced to create another level in the hierarchy whenever necessary. In this paper, we develop class hierarchies of orientation distribution function based on features in the form of pole density functions over prominent fiber families [6,7] in the fundamental region as shown in Fig. 3.

Although it is impossible for a finite database to contain the entire universe of microstructures and processing paths, it is a relatively easier task to identify promising processing path solutions through finite samples. Once the initial

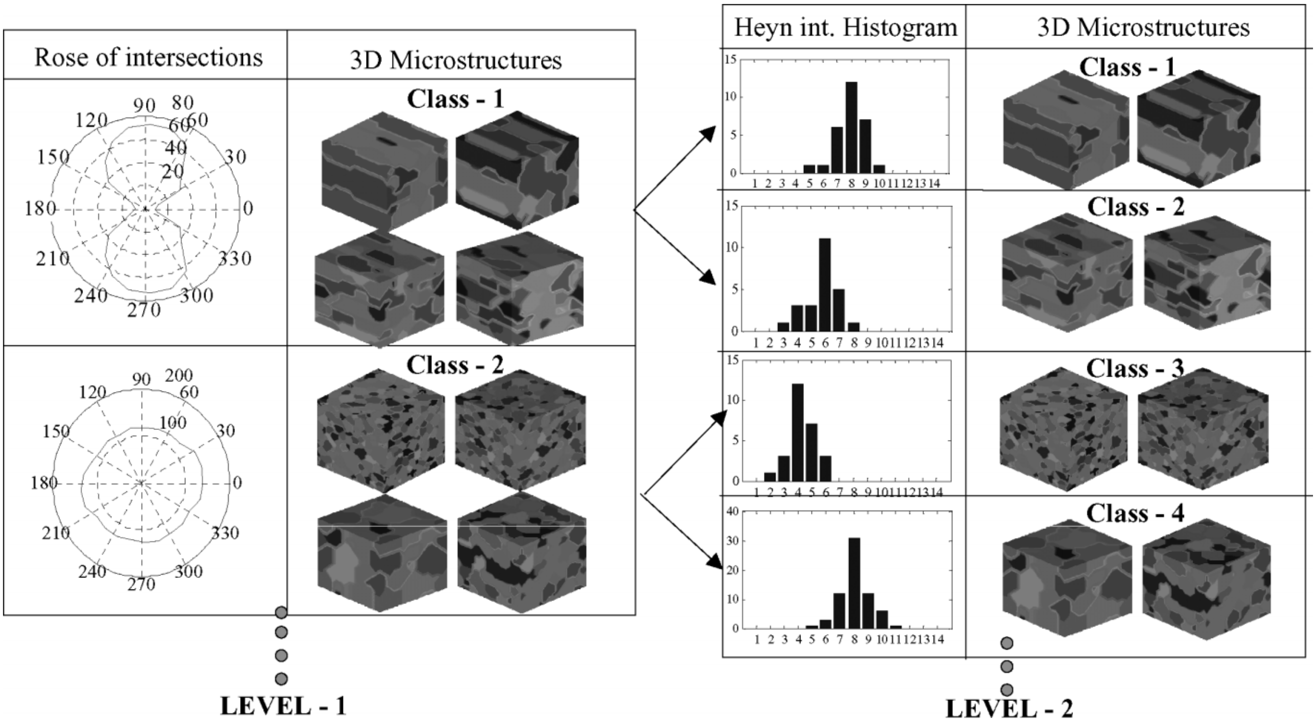

Fig. 2 The classification hierarchy based on topological descriptors. The feature vectors contain the Heyn intercept measure and the rose of intersections.

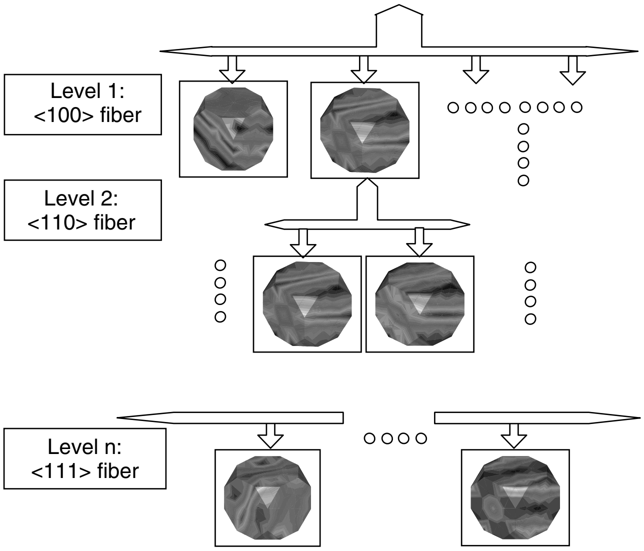

Fig. 3 The classification hierarchy for ODFs. The feature vector contains the pole density functions at different sample directions for the family of fibers specified at each classification level.

processing path is identified, local optimization methodologies can be invoked to fine-tune process parameters. An innovative use of the database to this end is its use for generating reduced-order models for accelerating local optimization problems [8,9]. Reduced-order representation of the ODF allows microstructure evolution simulations with as few as three to four unknowns compared with thousands in the case of a full-order simulation. However, one needs to select appropriate reduced-order models that represent the physical mechanism encountered in the process. Once a large database is available, reduced-order models can be generated on-the-fly using existing ODFs in the database. These reduced models can be subsequently used to significantly accelerate local optimization schemes. The process of obtaining reduced-order models is through a technique called proper orthogonal decompositon [10], a discrete version of which is called principal component analysis [11] in the image-processing community.

The rest of the paper is organized as follows. In Section 2, the X-means classifier is briefly explained followed by introduction to the concept of microstructural features in Section 3. In Section 4, a texture evolution model based on reduced representation of texture is discussed followed by the adaptive model reduction strategy in Section 5. The process design methodology is explained in Section 6. In Section 7 we present relevant applications of the methodology.

2. MICROSTRUCTURE CLASSIFICATION

A microstructure classification approach is used to create an organized database of microstructural information from which relationships between materials process and microstructures can be identified. The task of the classifier is to identify a class of microstructures (and associated process parameters) that may result in a desired property. The underlying inverse problem can be addressed by mining information over a large and comprehensive database. As described in the Introduction, microstructure classification is posed as an unsupervised classification problem. We aim to unearth the relationships between a set of microstructural data without the need for any user-defined inputs in the form of class labels. Consider a data-set (D) consisting of n features of the microstructure, $\boldsymbol{x}_i$, $i = 1, \ldots, n$, with each feature attribute being a vector of m values as $\boldsymbol{x}_i^{\mathrm{T}} = \{x_{1i}, \ldots, x_{mi}\}, \boldsymbol{x}_i \in \Re^m$. The unsupervised classification problem is posed as follows:

Find the cluster centers $\{\boldsymbol{C}^1, \boldsymbol{C}^2, \ldots, \boldsymbol{C}^k\}$ in $\Re^m$ such that the sum of the 2-norm distance squared between each feature $\boldsymbol{x}_i$ and its nearest cluster center $\boldsymbol{C}^h$ is minimized.

The clustering problem can be written using the above mentioned 'distortion measure' as the problem of finding the cluster centers $\{\boldsymbol{C}^1, \boldsymbol{C}^2, \ldots, \boldsymbol{C}^k\}$ so that the cost function (J) is minimized,

$$J(\boldsymbol{C}^1, \ldots, \boldsymbol{C}^k) = \sum_{i=1}^{n} \min_{h=1,\ldots,k} \left(\frac{1}{2} \|\boldsymbol{x}_i - \boldsymbol{C}^{h(i)}\|_2^2 \right). \quad (1)$$

The coordinate $\boldsymbol{C}^{h(i)}$ is the centroid of the cluster in which the feature $\boldsymbol{x}_i$ is located. Given a database D of n points in $\Re^m$ and cluster centers $\{\boldsymbol{C}^{1,t}, \boldsymbol{C}^{2,t}, \ldots, \boldsymbol{C}^{k,t}\}$ in $\Re^m$ at iteration t, the well-known Lloyd's algorithm computes the cluster centers, $\{\boldsymbol{C}^{1,t+1}, \ldots, \boldsymbol{C}^{k,t+1}\}$ at iteration $t + 1$ in the following two steps:

1. *Cluster Assignment*: For each data $\boldsymbol{x}_i$, assign $\boldsymbol{x}_i$ to cluster h such that center $\boldsymbol{C}^{h,t}$ is nearest to $\boldsymbol{x}_i$ in the 2-norm.
2. *Cluster Update*: Compute $\boldsymbol{C}^{h,t+1}$ as the centroid of all points $\boldsymbol{x}_i$ assigned to cluster h.

The algorithm is stopped when $\boldsymbol{C}^{h,t+1} = \boldsymbol{C}^{h,t}$, $h = 1, \ldots, k$, otherwise t is incremented by 1 and steps 1 and 2 are repeated. At the start of the algorithm, the k cluster centers, $\{\boldsymbol{C}^{1,0}, \boldsymbol{C}^{2,0}, \ldots, \boldsymbol{C}^{k,0}\}$ are randomly initialized. The cluster center solutions produced depend on these initial values, and bad initial guesses may result in suboptimal partitioning. The standard solution is to try several starting configurations. However, due to high dimensionality and variety of microstructural features, it is difficult to estimate or analyze the clustering beforehand. Hence, it becomes imperative to employ efficient techniques for discovering the actual number of classes that exist in the set of microstructures.

We employ the 'X-means' algorithm [4] for discovering the actual number of classes that exist in the set of ODFs. The Schwarz criterion, on the basis of the Bayesian information Criterion, is employed in the X-means algorithm to estimate the actual number of clusters in the data-set. Maximization of the BIC provides a balance between minimization of k-means cost function and reduction of model complexity. k-means cost function is minimum if every data point is a separate class and model complexity is minimum if all data points belong to one class. A balance between these two quantities allows selection of just the right number of classes needed for classification of a microstructural database. The Bayesian information measure for the data is obtained as follows. Given the data-set (D), the model chosen maximizes the BIC given as,

$$\mathrm{BIC} = \hat{l}(D) - \frac{p}{2}\log(n), \quad (2)$$

where, $\hat{l}(D)$ is the log-likelihood of the data taken at the maximum likelihood point, p is the number of free parameters in the model, $p = m \cdot k + k - 1 + k$, consisting of $m \cdot k$ cluster center coordinate values, $k - 1$ class probabilities (which is the probability of finding a data point in class k) and k estimates of variance in distribution of data within each class. The maximum likelihood estimate for this variance, assuming spherical-Gaussian distribution of data within a cluster (h) consisting of r_h data points ($\boldsymbol{x}_i^h, i = 1, \ldots, r_h$), is given as,

$$\hat{\sigma}_h^2 = \frac{1}{r_h - 1} \sum_{i=1}^{r_h} \|\boldsymbol{x}_i^h - \boldsymbol{C}^h\|^2. \quad (3)$$

The probability of each point within the cluster (h) is given as a product of class probability (r_h/n) and the probability of the data point to be in class h as,

$$\hat{P}(\boldsymbol{x}_i^h) = \frac{r_h}{n} \frac{1}{\sqrt{2\pi\hat{\sigma}_h}^{m}} \exp\left(-\frac{1}{2\hat{\sigma}_h{}^2} \|\boldsymbol{x}_i^h - \boldsymbol{C}^h\|^2 \right). \quad (4)$$

The log-likelihood of all the data within the cluster is given as,

$$\begin{aligned} l(D_h) &= \log \prod_{i=1}^{r_h} P(\boldsymbol{x}_i^h) \\ &= \sum_{i=1}^{r_h} \left(\log\left(\frac{1}{\sqrt{2\pi}\hat{\sigma}_h{}^m} \right) - \frac{1}{2\sigma_h^2} \|\boldsymbol{x}_i^h \right. \\ &\quad \left. - \mathbf{C}^h\|^2 + \log\frac{r_h}{n} \right). \end{aligned} \quad (5)$$

Hence, at the maximum likelihood estimate, the log-likelihood of the data belonging to cluster h is given as,

$$\hat{l}(D_h) = -\frac{r_h}{2}\log(2\pi) - \frac{r_h.m}{2}\log(\hat{\sigma_h}^2) - \frac{r_h - 1}{2} + r_h\log(r_h) - r_h\log(n). \qquad (6)$$

The log-likelihood of the entire data set is the sum of the log-likelihoods of all clusters, hence, the BIC (Eq. 2) for the entire data-set can be written as,

$$\mathrm{BIC} = -\frac{n}{2}\log(2\pi) - \frac{m}{2}\sum_{h=1}^{k} r_h\log(\hat{\sigma_h}^2) - \frac{n-k}{2} + \sum_{h=1}^{k} r_h\log(r_h) - n\log(n) - \frac{(m+2)k-1}{2}\log(n). \qquad (7)$$

In the X-means algorithm, the Schwarz criterion is tested for configurations arising from various values of k, starting from two clusters to a maximum number of clusters specified by the user, and the best configuration is chosen. Convergence properties are improved in the algorithm by letting cluster centers (parent clusters) obtained from each k-means step to split further into two centers. This is performed through local k-means operation on the parent cluster using two new centers. The Schwarz criterion is then tested locally within the parent cluster. The parent cluster is retained only if the BIC degrades due to the splitting operation. Fig. 4 shows a comparison of the X-means and the k-means methodologies for a two-dimensional feature set. Fig. 4(a) shows a configuration produced by the k-means algorithm with the number of classes (k) given as 4. The configuration has converged but has not produced the distortion corresponding to the global minimum. Fig. 4(b) shows the k-means results with number of classes $k = 6$. With k-means, it is not possible to extract the true clustering in the data-set. The higher the number of classes, the lesser is the distortion, but the possibility of overfitting the data increases. Fig. 4(c) shows the cluster centers identified by the X-means algorithm. On the basis of the Schwarz criterion, X-means identified the four natural clusters in the data-set without user intervention. The X-means classifier, despite its simplicity, is found to provide remarkable efficiency when used to classify microstructural data.

3. FEATURE EXTRACTION

Polycrystalline microstructures are identified by orientational and topological attributes. During deformation processes, the overall deformation and lattice rotation in grains cause changes in these attributes. The ODF provides information about the volume fraction of any orientation. Spatial distributions in orientations of grains also varies with deformation. Mathematical representations of the set of stereological and orientational features are briefly given here:

1. *Orientation descriptors:* Orientation distribution function ($\mathcal{A}$) [12,13] is the probability density function for orientations. For an infinitesimal volume ($\mathrm{d}\boldsymbol{g}$) in the orientation space around orientation $\boldsymbol{g}$, the function $\mathcal{A}(\boldsymbol{g})\mathrm{d}\boldsymbol{g} = \mathrm{d}v$ is the volume of grains (per unit volume of microstructure) with orientation $\boldsymbol{g}$. The ODF, $\mathcal{A}$, can also be understood as the probability of one point randomly placed in the microstructure to land in a particular orientation. ODFs are thus primarily an one-point statistic and do not include spatial information about the distribution of texture.
2. *Topological (grain shape/size, grain boundary) descriptors:* The local structure of boundaries is also cast in terms of probability densities. The simplest metric employed to describe grain-boundaries is the probability function $f(\boldsymbol{n})$ described over a unit hemisphere. For an infinitesimal area ($\mathrm{d}\boldsymbol{n}$) in the unit hemisphere, the function $f(\boldsymbol{n})\mathrm{d}\boldsymbol{n} = \mathrm{d}S$ is the surface area (per unit microstructural volume) of lattice interface

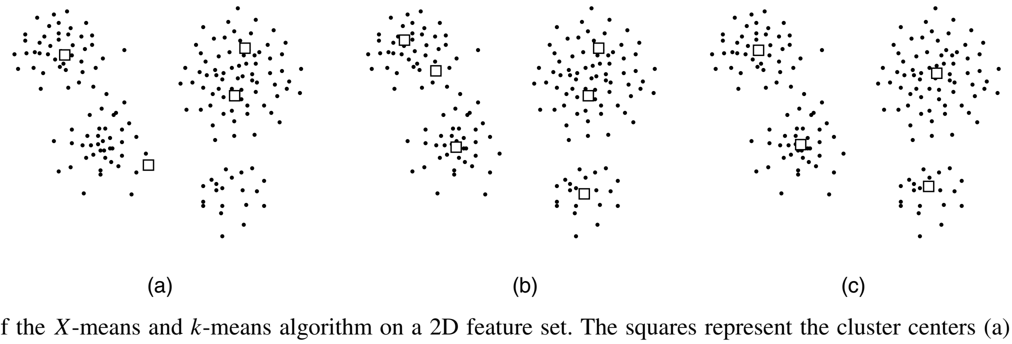

Fig. 4 Results of the X-means and k-means algorithm on a 2D feature set. The squares represent the cluster centers (a) Clustering using k-means: local optimum produced by the k-means algorithm ($k = 4$) (b) Clustering using k-means with number of classes fixed at $k = 6$ (c) four clusters identified by the X-means algorithm.

with normal $\boldsymbol{n}$. For planar images used for characterization, a simple measure called 'Saltykov rose of intersections' provides a simpler shape and size information of grains. To obtain the rose of intersections, a network of parallel equidistant lines is placed over the microstructure image at several angles and the number of grain boundary intersections with each test line is measured. The distribution of intersections with the angle of orientation of the lines is called the rose of intersection. Histograms of the intercept length distribution (mean intercept length versus number of test lines possessing the mean intercept length) can be used as a feature parameter for the grain size distribution [14,15].

3. *Combined stereological and topological descriptors:* N-point correlation functions include both orientational and topological attributes. For example, the two-point correlation function expressed as $f_2(\boldsymbol{g}, \boldsymbol{g}'|\boldsymbol{r})$ defines the probability that a randomly placed vector of length $\boldsymbol{r}$ will sample orientation $\boldsymbol{g}$ and $\boldsymbol{g}'$ at its end points, respectively. A useful lower-dimensional representation of two-point function is based on misorientation $(\Delta\boldsymbol{g})$ between the end points is called the misorientation correlation function $[f_2(\Delta\boldsymbol{g}|\boldsymbol{r})]$. More detailed descriptions of boundary texture include a consideration of lattice orientations as function of grain boundary inclinations. The boundary texture feature $f(\boldsymbol{r}, \boldsymbol{n}, \boldsymbol{r}')$ can be extracted by identifying the grain orientation on each side of a boundary, $\boldsymbol{r}$ and $\boldsymbol{r}'$ and the inclination of the boundary tangent plane, $\boldsymbol{n}$ [16].

Although both stereological and topological descriptors are important for calculating physical properties associated with a microstructure, we initially restrict ourselves to using ODF as the only descriptor in the next section. In principle, it is straightforward to include higher-order descriptors of the microstructure due to the hierarchical nature of the database. In the section on examples, we discuss several applications of classification of microstructures based on features. Significant applications include problem of identification of processing paths to realize desired microstructures and identification of multiple process solutions for achieving desired properties.

3.1. The orientation distribution function

Orientations in three dimensions are conventionally represented using the Eulerian angle representation (three angles: azimuth, tilt, and elevation). However, in our work we use the Rodrigues-Frank (RF) axis–angle parameterization of the orientation space. This is based on unique association of an orientation with a rotation axis, and an angle of rotation about the axis.

The Rodrigues' parametrization is created by scaling the axis of rotation as $\boldsymbol{r} = \boldsymbol{n}\tan(\theta/2)$. A proper rotation $\boldsymbol{R}$ relates the lattice orientation to a reference orientation. The orientation space is the region containing all possible crystal orientations in the microstructure. The fundamental region represents a region of the orientation space such that each crystal orientation is represented uniquely within the space. Fundamental region for the cubic symmetry group (includes metals such as Aluminum and Copper) results in a truncated cube. The planes that form the faces of the cube are introduced by symmetry rotations about $\langle 100 \rangle$ family of axes and the corners are truncated by planes introduced by rotations about the $\langle 111 \rangle$ axes. The ODF (represented by $\mathcal{A}$) describes the local density of crystals over this fundamental region of orientation space. The volume fraction of crystals within a part $(\Re^*)$ of the fundamental region is given by $v_{\mathrm{f}}(\Re^*) = \int_{\Re^*} \mathcal{A} \mathrm{d}v$. The use of RF space provides several advantages for the ODF classification problem. In particular, because of the symmetry of the RF space, textures take on a simple structure and most ideal orientations are present close to the boundaries of the orientation space, providing an ease of interpretation of clusters obtained from the X-means algorithm.

Feature–property relationship. Now that the microstructural feature has been mathematically described, it is useful to define the link between this microstructural feature and a macroscopic physical property. The macroscopic property is given by the average of the property $\chi(\mathbf{r}, t)$ as determined by an integral over the fundamental region:

$$\langle \chi \rangle = \int_{\mathrm{R}} \chi(\boldsymbol{r}, t) \; \mathcal{A}(\boldsymbol{r}, t) \; \mathrm{d}v. \tag{8}$$

We employ the approach in [17–19] where classification is carried out over a hierarchy of classes using lower-order microstructural features. The lower-order features for the ODF are extracted in the form of density of important orientation fibers in the fundamental region. The intensity of these fibers provide a natural link with the processes involved. Fig. 3 shows the classification scheme for textures based on pole density functions as lower-order features at various levels. As illustrated in Fig. 5, the z-axis $< 110 >$ fiber (marked as BB' in the figure) on the cubic fundamental region has different intensities in the ODFs obtained from uniaxial tension and compression processes respectively. The pole density function for this group of fibers allows us to immediately discriminate between tension and compression textures. An advantage of the clustering scheme using fiber density features is its ability to capture non-uniqueness in the process-design solutions. Through classification, identification of several processing paths that can

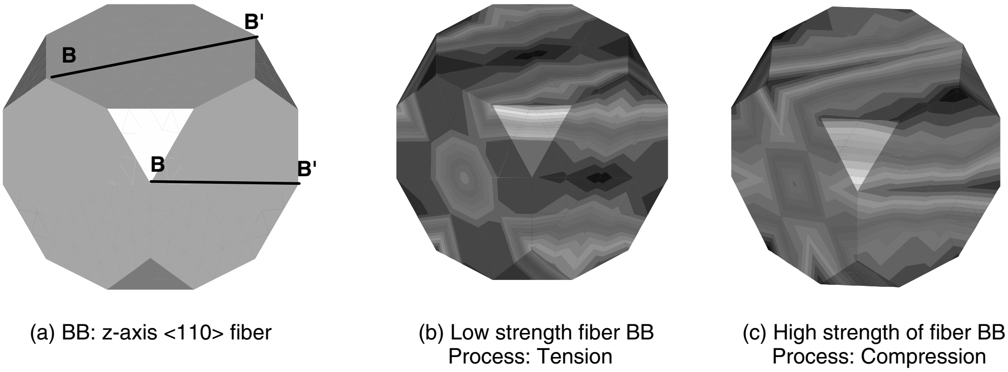
B'
B
B
B'
(a) BB: z-axis <110> fiber
(b) Low strength fiber BB
Process: Tension
(c) High strength of fiber BB
Process: Compression

Fig. 5 *Feature extraction*: (a) The orientations marked BB' are intersections of z-axis $< 110 >$ fiber with boundaries of the cubic fundamental region. (b,c) The ODFs obtained from uniaxial tension and compression processes, respectively, can be distinguished based on the intensity of z-axis $< 110 >$ fiber (BB). The pole density function for this fiber forms our feature vector for the first class level in the hierarchy.

lead to the desired ODF is made possible. The methodology for extraction of lower-order features employed is explained in Appendix A.

4. GENERATION OF MICROSTRUCTURE DATABASE

Microstructural feature evolution (in this example, the ODF evolution) under a variety of processes is simulated using physical models and is used to build the database. The evolution of the ODF is governed by the ODF conservation equation. The conventional Eulerian rate form of the conservation equation is given by [6]:

$$\frac{\partial \mathcal{A}(\boldsymbol{r},t)}{\partial t} + \nabla \mathcal{A}(\boldsymbol{r},t)\cdot \boldsymbol{v}(\boldsymbol{r},t) + \mathcal{A}(\boldsymbol{r},t)\,\nabla\cdot \boldsymbol{v}(\boldsymbol{r},t) = 0, \tag{9}$$

where $\boldsymbol{v}(\boldsymbol{r},t)$ is the Eulerian reorientation velocity. Using velocity gradient $\boldsymbol{L}$, the reorientation velocity $[\boldsymbol{v}(\boldsymbol{r},t)]$ can be evaluated through a viscoplastic constitutive relation as described in [8].

Once the underlying process (tension, compression, shear or combined process modes) of deformation is known, the microscopic velocity gradient, $\boldsymbol{L}$ can be computed as follows. $\boldsymbol{L}$ is decomposed uniquely as in Eq. (10). Each matrix in the decomposition of Eq. (10) corresponds to a given deformation process namely tension/compression (α_1), plane strain compression (α_2), shear modes ($\alpha_3, \alpha_4, \alpha_5$), and rotation modes ($\alpha_6, \alpha_7, \alpha_8$).

$$\boldsymbol{L} = \alpha_1 \begin{bmatrix} 1 & 0 & 0 \\ 0 & -0.5 & 0 \\ 0 & 0 & -0.5 \end{bmatrix} + \alpha_2 \begin{bmatrix} 0 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & -1 \end{bmatrix} + \alpha_3 \begin{bmatrix} 0 & 1 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix} + \alpha_4 \begin{bmatrix} 0 & 0 & 1 \\ 0 & 0 & 0 \\ 1 & 0 & 0 \end{bmatrix} + \alpha_5 \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 1 \\ 0 & 1 & 0 \end{bmatrix} + \alpha_6 \begin{bmatrix} 0 & -1 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix} + \alpha_7 \begin{bmatrix} 0 & 0 & -1 \\ 0 & 0 & 0 \\ 1 & 0 & 0 \end{bmatrix} + \alpha_8 \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & -1 \\ 0 & 1 & 0 \end{bmatrix}. \tag{10}$$

The ODF, $\mathcal{A}$ used for constructing the database is then computed by solving Eq. (9) using finite element techniques. The initial unprocessed microstructure is assumed to have a random texture with $\mathcal{A}(r,0) = 2.435$. The full- and reduced-order finite element methodologies for solving the partial differential equation (PDE) (Eq. 9) are described in the next section.

4.1. Full- and reduced-order approaches

Equation (9) is subject to discontinuities in the velocity divergence and its solution requires stabilized numerical methods that involve considerable computational resources due to associated mathematical and computational complexity. The proper orthogonal decomposition (POD) technique is a popular reduced-order modeling approach for decreasing the computational burden using existing information from a database. Reduced-order modeling is based on the development of a reduced set of basis functions, $\phi(\boldsymbol{r})$, to represent the associated ODF. The method of snapshots is introduced for generating the basis. It assumes that the basis, ϕ, can be expressed as a linear combination of the N

snapshots from the solution ($\mathcal{A}$) at different times of Eq. 9 as:

$$\phi_j = \sum_{i=1}^{N} w_i^j \mathcal{A}^{(i)}, \tag{11}$$

where w_i^j can be determined by the solving the eigenvalue problem:

$$\boldsymbol{C}\ \boldsymbol{W} = \boldsymbol{\Lambda}\ \boldsymbol{W}, \tag{12}$$

where, $\boldsymbol{C}$ is the spatial correlation matrix defined as,

$$\boldsymbol{C}_{i,j} = \frac{1}{N}\int_{\mathrm{R}} \mathcal{A}^{(i)}(\boldsymbol{r})\ \mathcal{A}^{(j)}(\boldsymbol{r})\ \mathrm{d}V, \tag{13}$$

and $\boldsymbol{\Lambda}$ and $\boldsymbol{W}$ are the complete eigen-description of the system. Coefficient w_i^j corresponds to the (i, j)th entry in $\boldsymbol{W}$ matrix. Once basis functions (ϕ_j) are evaluated using Eq. (11), ODF at any processing stage j can be evaluated as,

$$\mathcal{A}^{(j)}(\boldsymbol{r}, t) = \sum_{i=1}^{M^{(j)}} a_i^{(j)}(t)\phi_i^{(j)}(\boldsymbol{r}), \tag{14}$$

where $M^{(j)}$ is the number of modes used in stage j, and $a_i^{(j)}$ are the reduced-order coefficients used for representing the ODFs in stage j. Using this approximation, the weak form of the ODF conservation equation reduces to the following ordinary differential equation (ODE):

$$\dot{a} = \mathcal{R}a, \tag{15}$$

where,

$$\mathcal{R}_{i,j} = \int_{\mathrm{R}} (\nabla\phi_j \cdot \boldsymbol{v}\ \phi_i + \phi_j\ \phi_i\ \ \nabla \cdot \boldsymbol{v})\mathrm{d}v. \tag{16}$$

Equation (15) is solved over n timesteps (0 to t_n) at each stage. At any stage j, $j = 1, \ldots, p$, the initial value of a is given through the following equation,

$$a_i^{(j)}(0) = \int_{\mathcal{R}} \mathcal{A}^{(j-1)}(\boldsymbol{r}, t_n)\phi_i^{(j)}(\boldsymbol{r})\mathrm{d}v, \tag{17}$$

where, $\phi^{(j)}(\boldsymbol{r})$ is the reduced basis at stage j. The reduced basis $\phi^{(j)}(\boldsymbol{r})$ at stage j is obtained using an adaptive basis selection procedure explained in the next section. Equations (15)–(17) define the reduced-order model for the ODF conservation equation. Thus, to obtain desired ODFs (e.g. desired texture and thus desired properties), one needs to only control a small finite number of degrees of freedom (i.e. the vector $\boldsymbol{a}$).

5. ADAPTIVE REDUCED-ORDER MODEL EXTRACTION FROM DATABASE

The classification technique is database-driven and the availability of existing information can be further utilized to accelerate the texture evolution models. As discussed in the previous section, different reduced basis functions are used at each stage of a process to accurately represent the trajectory of the evolution of the ODF. These basis functions are chosen adaptively from the database during ODF evolution simulations. As an example, the process of choosing the reduced-order basis corresponding to the second processing stage in a two-stage process (compression followed by shear) is explained here. The straining rates for the two processes are first described using a vector $\boldsymbol{\alpha} = [\alpha_1, \alpha_3]$ [from Eq. (10)]. The basis for the second stage ($\phi^{(2)}(\boldsymbol{r})$) is found through the following steps [10]:

1. Select a new reduced-order basis ($\phi^{(2)}(\boldsymbol{r})$) from the existing database by searching for the closest process parameter ($\boldsymbol{\beta}$) within a user-defined tolerance limit,

$$\|\boldsymbol{\beta} - \boldsymbol{\alpha}\|_2 \leq \epsilon. \tag{18}$$

2. If $\|\boldsymbol{\beta} - \boldsymbol{\alpha}\|_2 > \epsilon$, then compute snapshots corresponding to the second process using Eq. (9) (the initial texture used is the ODF at the end of process 1) and compute the new reduced basis $[\phi^{(2)}(\boldsymbol{r})]$. Finally, update the existing database with the newly computed $\phi(\boldsymbol{r})$ and corresponding parameter $\boldsymbol{\alpha}$.

REMARK 1: An initial uniform texture is assumed at the first processing stage. The reduced-order basis corresponding to the pure deformation modes (obtained from an ensemble of data corresponding to a deformation $\alpha_i = 1$, if mode i is used in the first stage) is found to be sufficient to represent the texturing in the first stage.

REMARK 2: Over large databases, the search procedure in step (1) of the adaptive reduced-order algorithm can be addressed efficiently using classification algorithms.

The difference between the reduced- and full-order control solutions depend upon the sensitivity of the desired property to the numerical error induced by the introduction of reduced basis. Selection of the threshold parameter (ϵ) plays a critical role in the adaptive basis scheme described above. Small thresholds result in more accurate solutions but are computationally expensive due to frequent basis changes. Larger thresholds involve less frequent basis changes but may result in inaccurate solutions since the basis might not model the process employed accurately. Further, the sensitivities may be inaccurate leading to divergence in the objective function. Fig. 6 shows the increase in

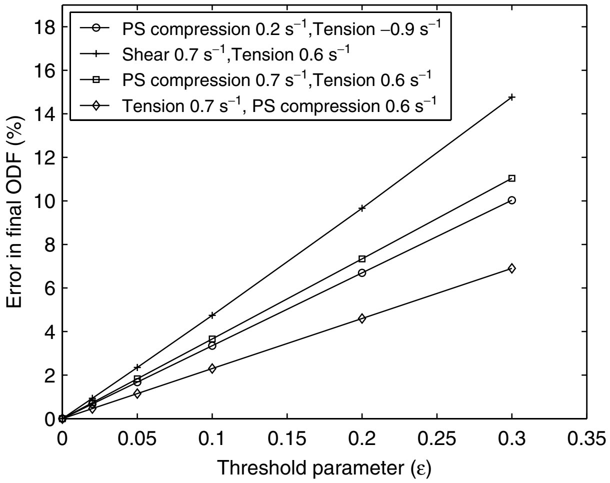

Fig. 6 Error induced due to different thresholds for the basis, the error not only depends on the threshold but also on the sequence of processing stages involved.

error caused with increasing values of ϵ used for the basis selected. The strain rate for the first stage is fixed and that of the second stage is increased which results in different values of ϵ. The ODF resulting from a basis with $\epsilon = 0$ after a time of 0.1 s is used as the reference ($\mathcal{A}^{\text{ref}}$). The error is defined as $100 \times (\|\mathcal{A}^{\text{ref}} - \mathcal{A})\|_2 / \|\mathcal{A}\|_2)$. The error not only depends on the threshold but also on the types of processing stages involved. With tension as the second processing stage, changing the first stage to shear from plane strain compression results in about 30% increase in error at the same threshold. Within a processing sequence, however, the increase in error due to change in processing parameters is small. The results also indicate that the thresholds can be varied based on the processing sequence, a tension-plane strain compression processing sequence can have twice as much threshold than the shear-tension sequence with similar errors induced by the reduced-order approximation.

Database architecture: As shown in Fig. 7, the database includes several pieces of data for each simulated process sequence. Data corresponding to each process sequence are (i) process sequence (ii) process parameters (iii) reduced ODF basis for the final stage in the process, (iv) the textures obtained from the process sequence. Final textures obtained from each process sequence are used in the classification procedure (Fig. 3) to divide the database into classes. If a user wishes to identify the processing parameters that would lead to a desired texture, first step would be to identify the class to which the final texture belongs through X-means classification. The next step is to extract a list of all process sequences from the identified class. The most promising process sequence can be chosen from this list and local optimization methodologies can be invoked to fine-tune process parameters (described in the next section). Processing parameters change during various iterations of the local optimization procedure. Basis functions in successive optimization iterations are adaptively selected from the database as described earlier in this section. Initially, about 1000 ODF samples were used to develop the database. These ODFs are computationally obtained by subjecting an initial random ODF through a series of different deformation processes. A computational approach from [8] is used for generating the database for an face centered cubic (FCC) Copper system with 12 slip systems. During optimization runs, the number of ODFs present in the database increases whenever new processing parameters (not initially present in the database) are sampled. The methodology for post-processing ODFs in the database to obtain features (density of various orientation fibers) is explained in the Appendix.

6. LOCAL OPTIMIZATION PROCEDURE FOR PROCESS DESIGN

The objective of the microstructure-sensitive design process is to control the properties in the microscale through design of appropriate deformation processes. The direct problem described in Section 4 simulates the ODF evolution given the macrovelocity gradient. The process design methodology aims to identify the macrovelocity gradient that yields a desired ODF (or desired property distribution). It is impossible for a finite database to contain the entire universe of microstructures and processing paths, it is a relatively easier task to identify promising processing path solutions through finite samples. Given a intelligent initial choice obtained from the X-means classification algorithm, gradient-based methods fine-tune the solution to obtain the right process parameters. Intelligent choice of the initial guesses can be made using prior information available in the form of a database through classification. Refer to [8,9] for complete details on the implementation of the design problem for a single stage using gradient-based approach. The calculation of gradients in the optimization framework involves reduced-order modeling of the polycrystal continuum sensitivity equation and calculation of sensitivities of the reorientation velocity through the crystal plasticity relations by design differentiation of the linking hypothesis. The following section addresses the extension of the technique to a multi-stage design process.

6.1. Multi-stage design process

Let us consider Eq. (9). We denote the sensitivity of the ODF to a small change in the process parameter ($\boldsymbol{\alpha}$) as $\overset{\circ}{\mathcal{A}} = \overset{\hat{\circ}}{\mathcal{A}}(r, t; \boldsymbol{\alpha}, \Delta\boldsymbol{\alpha})$. In the reduced-order model, we utilize the basis developed earlier for the direct problem and approximate the sensitivity fields as linear combinations of

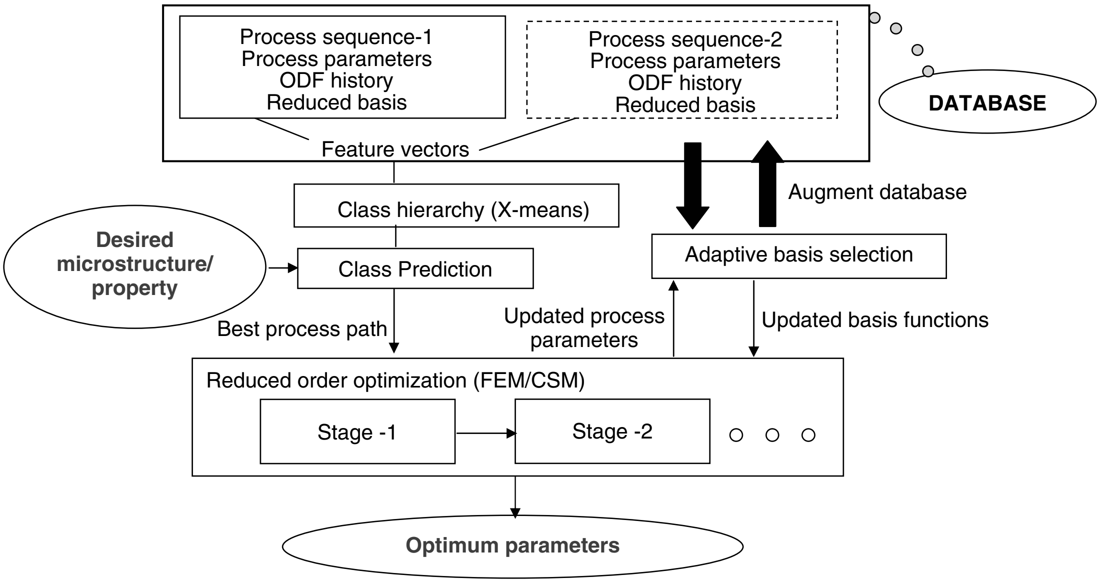

Fig. 7 Information flow from the database during a material design problem.

these basis functions. The computations for the sensitivity problem are similar to those performed for the direct analysis, and the reduced system is obtained as follows:

$$\dot{b} = \mathcal{G}b + \mathcal{H}, \tag{19}$$

where

$$\mathcal{G}_{i,j} = -\int_{\mathrm{R}} (\nabla\phi_j \cdot v\ \phi_i + \phi_j\ \phi_i\ \nabla\cdot v)\mathrm{d}v, \tag{20}$$

$$\mathcal{H}_i = -\int_{R} (\nabla\mathcal{A}\cdot \overset{\circ}{v}\ \phi_i + \mathcal{A}\ \phi_i\ \nabla\cdot \overset{\circ}{v})\mathrm{d}v. \tag{21}$$

For the first stage, $\overset{\circ}{\mathcal{A}}^{(1)}(r, 0) = 0$. At the end of stage j, $\overset{\circ}{\mathcal{A}}^{(j)}(r, t_n)$ is calculated from the coefficients obtained from the solution of Eq. (19) at the final time step as,

$$\overset{\circ}{\mathcal{A}}^{(j)}(r, t_n) = \sum_{i=1}^{M^{(j)}} b_i^{(j)}(t_n)\phi_i^{(j)}(r), \tag{22}$$

where $\phi^{(j)}(r)$ the reduced basis at stage j. The initial value of b for the $(j+1)$th stage is given through the following equation,

$$b_i^{(j+1)}(0) = \int_{\mathrm{R}} \overset{\circ}{\mathcal{A}}^{(j)}(r, t_n)\phi_i^{(j+1)}\mathrm{d}v, \tag{23}$$

where $\phi^{(j+1)}(r)$ is the set of reduced basis employed for the sensitivity problem at stage $j+1$. Once the sensitivity at the last time step of the final stage is found, the expectation of the sensitivity of a property to a small change in the process parameter is found as,

$$\langle\overset{\circ}{\chi}\rangle = \int_{\mathrm{R}} \chi(\boldsymbol{r}, t)\ \overset{\circ}{\mathcal{A}}^{(p)}(\boldsymbol{r}, t_n)\ \mathrm{d}v. \tag{24}$$

It is to be noted that a sensitivity problem of stage i (where process variable corresponding to stage i is perturbed) uses the stage i basis for the sensitivity problems in stages $i+1$ to p. We define the design problem of interest as the selection of the processing sequence, with stages involving tension/compression, plane strain compression, shear or rotation, and the corresponding process parameters $\boldsymbol{\alpha}$ [in Eq. (10)] that lead to a desired property Ω that is a function of the ODF. This can be stated as follows:

$$\min_{\boldsymbol{\alpha}}\ \mathcal{F}(\boldsymbol{\alpha}) = \frac{1}{N_{\mathrm{s}}}\sum_{i=1}^{N_{\mathrm{s}}}\left\{\Omega^i\left[\mathcal{A}(\boldsymbol{\alpha})\right] - \Omega^{\mathrm{desired}^i}\right\}^2, \tag{25}$$

where N_{s} is the total number of sampling points, $\Omega^{\mathrm{desired}}$ is the discrete representation of the desired microstructural property and $\boldsymbol{\alpha}$ is the design parameter involved in the iterative optimization algorithm corresponding to the process parameter (α_j) from stage $j = 1$ to p. The calculation of sensitivities of property χ to a component α_i of α require solution to sensitivity problems at $p - j + 1$ stages. The ith multi-stage sensitivity problem is driven by $\Delta\alpha_i = 10^{-2}$ with $\Delta\alpha_j = 0$ for $j \neq i$. The gradients of property (χ) with respect to α_i is calculated as,

$$\frac{\partial \chi}{\partial \alpha_i} = \frac{\overset{\circ}{\chi}\,(\boldsymbol{r}, t, \alpha_1, \ldots, \alpha_p, 0, \ldots, \Delta\alpha_i, \ldots, 0)}{\Delta\alpha_i}. \quad (26)$$

The sensitivities are then used in a gradient descent algorithm to obtain the optimum process parameters that minimize the objective function in Eq. (25). The gradient descent algorithm employed is explained in [8].

Implementation of adaptive model reduction algorithm: Let Data-set ‘A’ containing a processing sequence of tension and shear be found to result in a particular desired texture using the classifier. In the first iteration of the local optimization problem, the basis corresponding to the first two stages are obtained from data-set ‘A’. If during an intermediate iteration, the process parameters change beyond the allowed threshold (ϵ), then the database is initially queried for another data-set with process parameters matching the new process parameters. If such a data-set is not available, a new data-set is created and added to the database corresponding to the new process parameters (as indicated in Fig. 7). The basis for this data-set is used in subsequent iterations of the design problem until the process parameters once again change above the selected threshold. Using this scheme, just three modes of the basis (with three unknowns) are found to adequately represent the texturing at any stage in an optimization step, enhancing the computational efficiency of the algorithm. The success of the data-mining approach is limited to the amount of information in the database. Selection of good processing sequence solutions require a comprehensive database with data sets containing rich combination of processes. New information added to the database during the optimization process improves the possibility of identification of processing parameters and reduced basis functions directly from the database in future optimization runs.

7. APPLICATIONS IN MATERIALS DESIGN

The first problem involves designing the macrovelocity gradient to obtain a desired ODF. Given the initial processing sequence and the parameters identified by the classifier, the local optimization scheme fine-tunes the processing parameters until the error between desired texture and final processed texture is minimized. As an example, if a user wants to obtain a desired ODF shown in Fig. 8(a), the first step is to perform X-means classification to match the orientation fibers in the desired ODF to a class of ODFs in the database. From the class of ODFs, a promising process sequence is then chosen. In this example, a two-stage sequence of plane strain compression followed by compression is selected. In all examples, the time for which each deformation stage acts is fixed at 0.1 s and a random initial texture is used. The final ODF corresponding to the strain rates for the two stages, 0.65 and -0.1 s^{-1}, respectively, is shown in Fig. 8(b). By comparing Fig. 8(a) and (b), one can notice that the classifier has performed well in matching various fiber densities of the desired ODF and the ODFs in the database. The strain rates for the two processes after the adaptive reduced-order optimization procedure is obtained as 0.9472 and -0.2847 s^{-1}, respectively and the optimized ODF shown in Fig. 8(c) is obtained within an error of 1%.

The advantage of the data-mining methodology lies in the identification of multiple processing paths that lead to a desired texture. Fig. 9(b) shows a class of textures with different processing routes that can result in a desired ODF. Given the desired ODF, the classifier uses the lower-order features, namely, the pole density functions, over four levels in the class hierarchy corresponding to the fibers in the $< 110 >$, $< 100 >$, $< 111 >$, and $< 211 >$ fiber families, respectively. The orientation fibers are chosen based on their particular importance in FCC textures (refer Appendix A) and their close affiliation with the processes involved. The desired texture in Fig. 9(a) is seen to be dominated by two fibers, the z-axis $< 110 >$ fiber

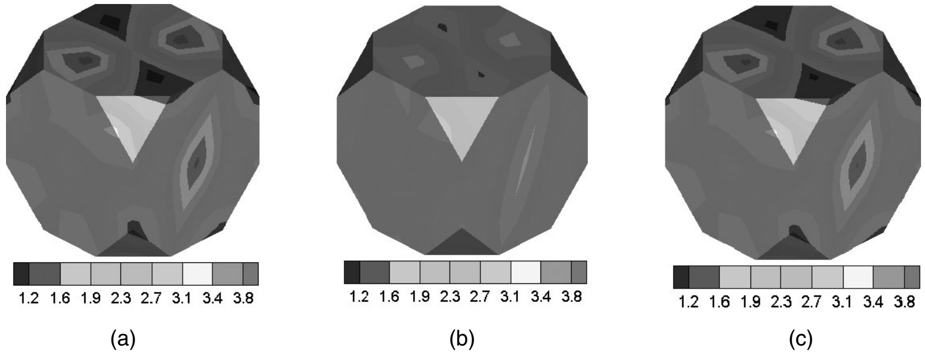

Fig. 8 Control of material texture: (a) the desired texture, (b) the initial guess identified by the classifier and (c) reduced-order optimized ODF.

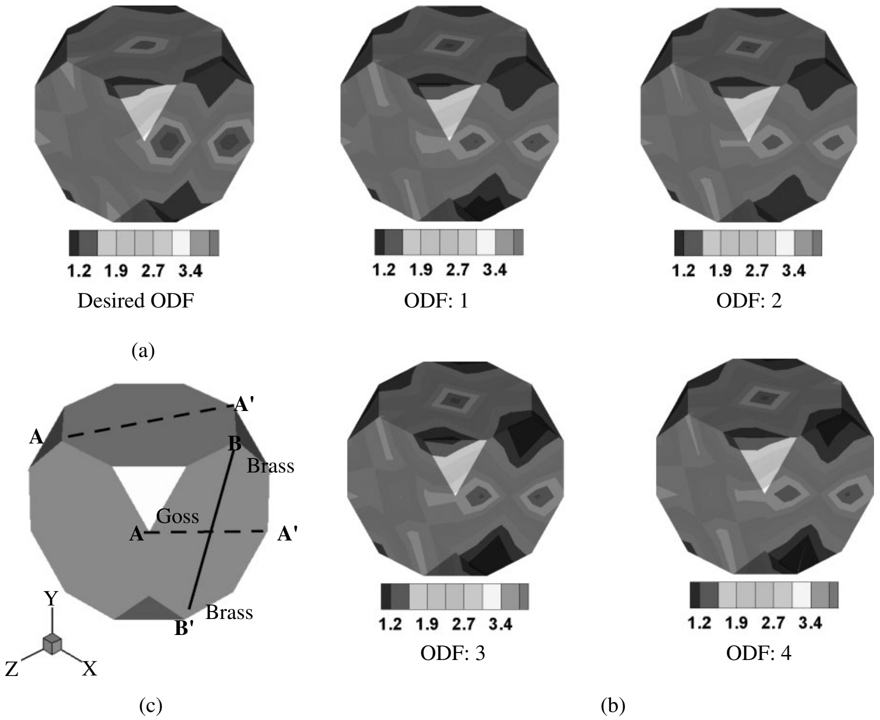

Fig. 9 (a) The desired ODF, (b) ODF: 1, 2, 3, 4 represent a class of ODFs similar to the desired ODF in their lower-order features, (c) Positions of z-axis $< 110 >$ (AA') and alpha fibers (BB') in the boundaries of the fundamental region.

Table 1. Process parameters of the ODF class in Fig.9.

ODF:	Stage 1	Stage 2	Stage 3
1	PSC (-0.677 s^{-1})	Shear (-0.165 s^{-1})	Tension (-0.881 s^{-1})
2	Tension (-0.835 s^{-1})	PSC (-0.606 s^{-1})	
3	Tension (-0.917 s^{-1})	Shear (-0.074 s^{-1})	PSC (-0.760 s^{-1})
4	Tension (-0.907 s^{-1})	PSC (-0.669 s^{-1})	Rotation (0.179 s^{-1})

and the alpha fiber (running from brass to the Goss component) shown in Fig. 9(c). ODF intensities in the alpha fiber are associated with the process of plane strain compression, although the texture shows stronger development of the brass component relative to Goss. Texturing to the z-axis $< 110 >$ fiber is normally associated with FCC metals under compression along the z-axis. From the processing sequences identified (Table 1), we see that these two processes are dominant in all the processing sequences found by the classifier.

7.1. Design for desired elastic modulus

The data-mining methodology can be extended to classification of ODFs based on material property distribution exhibited by the ODF. If a user wishes to identify the processing sequence and the final texture that lead to a desired texture-dependent property, the first step is to perform a X-means classification using property features [variation of the property as a function of angle from the rolling direction (RD)] to identify the class of ODFs that closely realize the property desired by the user. To perform such a classification, the properties of each ODF in the database needs to be pre-computed and stored. A variety of process sequences can be obtained from the class identified through X-means classification, from which the best process sequence may be chosen.

This example demonstrates identification of a sequence of processes in order to obtain a particular distribution of the elastic modulus of an FCC Copper polycrystal about the normal direction away from the RD. The stiffness in the crystal is given by the fourth tensor $\mathcal{C}_{ijkl}$ (values in GPa) in the crystal lattice frame for crystals with cubic symmetry [9]. The polycrystal stiffness, $\overline{\boldsymbol{C}}$, is computed through a weighted average (over $\mathcal{A}$) of the stiffness of individual crystals expressed in the sample reference frame. The elastic modulus is then computed through this polycrystal stiffness as

$$\boldsymbol{E} = \frac{1.0}{(\overline{\boldsymbol{C}})^{-1}_{(11)}}. \tag{27}$$

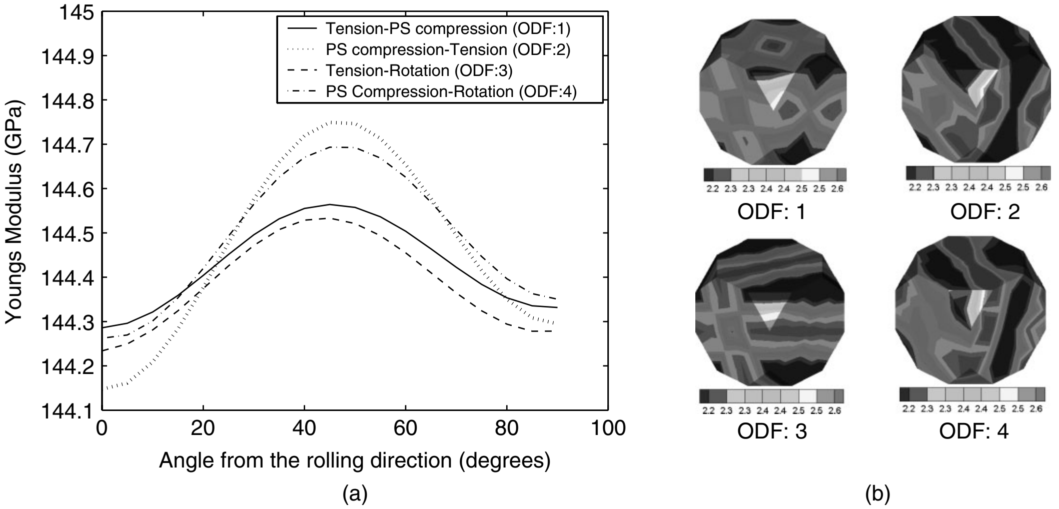

Fig. 10 (a) Classification based on property distribution: Young's modulus distribution of a class of ODFs (b) The corresponding ODFs.

Furthermore, the elastic modulus about an angle with the RD can be evaluated using the above equation, but after a coordinate transformation of $\overline{\boldsymbol{C}}$.

The classification scheme captures the non-uniqueness in process design, identifying several different textures (and processes) that might result in a desired property distribution. An example of a class of ODFs obtained from the database based on the Young's modulus property variation from RD to the transverse direction in the sample is shown in Fig. 10(b). The property distribution feature for a set of four ODFs within a class is shown in Fig. 10(a). In contrast to the texture design problem, the property design problem clearly illustrates the presence of multiple solutions. A range of different processing sequences [indicated in Fig. 10(a)] yield similar distributions of the Young's modulus. Thus, the methodology enables identification of new processes and selection of the economical process routes that leads to a desired property distribution based on available database of information.

7.2. Design for desired magnetic hysteresis loss

When a ferromagnetic material is taken through a cycle of magnetization and demagnetization in an alternating current field, energy is spent in aligning the magnetization vectors of the individual crystals along the direction of the external applied field. This alternating magnetization and demagnetization leads to a power loss in the material defined as hysteresis loss, which depends on the material microstructure. The total power loss can be expressed as a function of the external magnetization direction $\boldsymbol{h}$ by the expression [20]:

$$P(\boldsymbol{h}) = A_0 + A_1(l^2m^2 + m^2n^2 + l^2n^2) + A_2 l^2m^2n^2, \tag{28}$$

where P is expressed in watt per kilogram and l, m, and n are the direction cosines of $\boldsymbol{h}$ as represented in the crystal coordinate frame.

The coefficients A_0 and A_1 are different for the total power loss and the hysteresis loss and also depend on the frequency of magnetization. The coefficient A_2 is small and is generally ignored. For computation of magnetic hysteresis loss, the values $A_0 = 0$ and $A_1 = 6.1$ W/kg are used for computing the hysteresis loss of iron at a frequency of 30 Hz [8]. The corresponding polycrystal quantities can then be obtained by averaging over the ODF as in Eq. (8). The external magnetization direction in the sample coordinate system is taken as $(1, 0, 0)$ as shown in Fig. 11(a). To achieve a desired property distribution as shown in Fig. 11(c), we resort to the gradient-based optimization scheme with the processing sequences found using the classifier as the initial guess. A processing sequence of stage 1 of shear mode (mode 1) and stage 2 of tension mode was identified by the X-means classifier for achieving the desired property. A threshold of 0.05 is used for the selection of the adaptive basis. The final optimized process parameters were obtained as 0.9745 s^{-1} and 0.4821 s^{-1}, respectively. The hysteresis loss distribution corresponding to the optimized process parameters identified is shown along with the desired distribution in Fig. 11(c). The variation in the objective during the iteration process is shown in Fig. 11(b) shows rapid convergence to the required property from the initial guess provided by the classifier. In these demonstrative examples, simple deformation modes and moderate strains are used, hence, the resulting anisotropy in

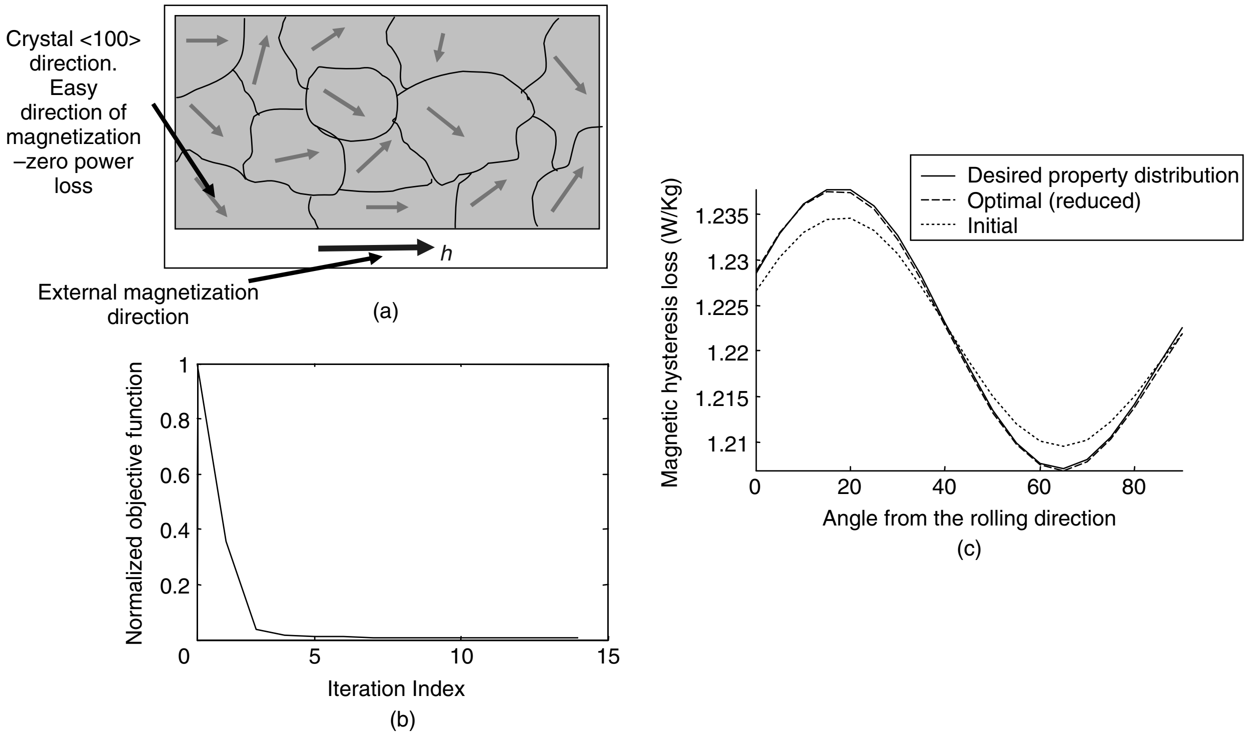

Fig. 11 Optimization of magnetic hysteresis loss distribution in the material: (a) Hysteresis losses depend on the orientation of individual crystals in the microstructure with respect to the direction of the externally applied field. (b) Variation of the objective function with iterations in the local optimization problem (c) Comparison of initial distribution identified by the classifier, the desired distribution and final optimized result.

elastic and magnetic properties seen from Figs. 11 and 10 is small. The classification methodology is general and can be extended to include complex loading histories, and also, design of several other texture-dependent properties such as thermal conductivity, the thermal expansion coefficient [12] and optical properties [13]. The proposed material design framework can be made robust by quantification of uncertainties in design solutions. Uncertainties are introduced due to changes in other features (such as grain size distribution) during deformation experiments. Quantification of such uncertainties has been a subject of our recent work in [21,22].

8. CONCLUSIONS

The paper presents a materials design methodology using statistical learning and adaptive reduced-order model selection. In particular, we have focused on the objective of designing processes that can lead to desired texture and texture-dependent properties. because of non-uniqueness in processing path solution (different processing paths leading to similar microstructural features) and complex nature of the microstructure–process relationships, this problem cannot be addressed solely using conventional optimization schemes. The inverse problem of identifying processes corresponding to desired texture is initially solved using a hierarchical classifier algorithm over a database. The classifier matches the lower-order features of the texture in the form of the pole density function over a class hierarchy to identify the sequence of processes that lead to the desired texture, hence, identifying multiple process paths that lead to the desired texture. An unsupervised classifier based on the k-means algorithm is used for the identification of natural clusters within the database. The number of classes in the texture database is not known *apriori*, hence, a Bayesian information criterion is used to identify the number of clusters.

Use of reduced-order basis functions to solve microstructure evolution problems leads to a computationally feasible and efficient method for optimization in real-time. The method of POD provides a systematic way to obtain reduced-order models; however, the basis functions used in the optimization problem need to represent the various texture evolution paths identified from intermediate optimization iterations. This is another situation where data-mining proves useful since reduced basis functions can be obtained on-the-fly from a database. Classification techniques hold promise in relating processing, structure, and properties of materials at variety of length scales. In particular, the integration of microstructure databases with physical modeling methods and local optimization algorithms can allow development of efficient means to design advanced materials.

APPENDIX

A. FEATURE EXTRACTION: CONCEPT OF ORIENTATION FIBERS

A $\langle h \rangle$ fiber about sample axis y connects orientations that align the crystal h-axis with the sample $\boldsymbol{y}$ direction. The rotation $\boldsymbol{R}$ required to align h with y is based on a rotation of h through an angle $\phi = \cos^{-1} h \cdot y$ about axis $h \times y$. Note that h and y remain aligned even if the orientations change due to rotations about h or y-axis. These orientations define the orientation fiber. In the Euler angle space, the fibers are curves described by trigonometric functions. Orientation fibers reduce to straight lines over Rodrigues' space. The orientations along the fiber over Rodrigues space is obtained by varying the parameter λ in the following equation,

$$\boldsymbol{r} = \frac{1}{1 + h \cdot y} \left[h \times y + \lambda (h + y) \right].$$

Here, $\lambda = \quad (\phi + \overline{\phi})/2$ where ϕ and $\overline{\phi}$ are arbitrary, corresponding to rotations about h and y, respectively. Fiber textures develop as flow of crystals over the space of orientations are channeled along particular orientation fibers. The intensity of these fibers provides a natural link with the processes involved. For example, the $\langle 110 \rangle$ family of fibers appears under uniaxial compression, plane strain compression, and simple shear. FCC metals are typically associated with texturing to $\langle 111 \rangle$ and $\langle 100 \rangle$ fibers under tension and $\langle 110 \rangle$ under compression. In torsion tests, the z-axis $\langle 111 \rangle$ fibers and x-axis $\langle 110 \rangle$ are seen to predominate. The texturing of FCC metals under plane strain compression is dominated by the α fiber (ND $\langle 110 \rangle$) connecting the ideal Goss and brass orientations, and the β fiber connecting the brass, S, and copper orientations([6]).

For a particular h, the pole figure takes values $\mathcal{P}(h, y)$ at locations y on a unit sphere. The pole density function, $\mathcal{P}(h, y)$ gets contributions from orientations for which the mapping $\boldsymbol{R}$ brings $\pm h$ (or symmetric equivalent) into alignment with the sample axis(y) as, $\boldsymbol{R} \cdot h = \hat{h}, \hat{h} \parallel y$. Using the crystal symmetries in the ODF, the expression for pole density function can be obtained as [7],

$$\mathcal{P}(h, y) = \frac{1}{2} \left[\mathcal{P}^0(h, y) + \mathcal{P}^0(-h, y) \right],$$

where, $\mathcal{P}^0(h, y)$ represents a path integral given as,

$$\mathcal{P}^0(h, y) = \frac{1}{2\pi} \int_{\hat{h} \parallel y} \mathcal{A} d\theta.$$

The integration is performed over all the fibers in the fundamental region corresponding to crystal direction h and sample direction y. Given the finite element discretization of the fundamental region, integration is done by tracking the fiber through each finite element. Within a finite element, $\mathcal{A}$ is interpolated using the element shape functions and the nodal point values associated with the element. The vector of all independent nodal values is represented by A^{np}. Pole density function $[\mathcal{P}(h, y)]$ of an orientation fiber family (h) is found over a sample direction (y) using a system vector $[m(h, y)]$, computed through a vector dot product as,

$$\mathcal{P}(h, y) = m(h, y)^{\mathrm{T}} A^{np}.$$

The feature vector ($\boldsymbol{x}_i^l$) for the ith ODF in the database at level l in the classification scheme is found as follows. The level l is associated with a particular fiber family (h) and the pole density functions are calculated at various values of $y = (y_1, y_2, ..., y_m)$ as, $\boldsymbol{x}_i^l = \boldsymbol{M} A_i^{np}$ where the system matrix ($\boldsymbol{M}$) is formed as $\boldsymbol{M} = [m(h, y_1)^{\mathrm{T}}; m(h, y_2)^{\mathrm{T}}; ...; m(h, y_m)^{\mathrm{T}}]$.

The system matrix is calculated and stored beforehand for the fiber families used for classification.

ACKNOWLEDGEMENTS

This research was supported with a grant to Cornell University by the Materials Design and Surface Engineering program of the NSF (award CMMI-0757824). Additional support was provided by the Mechanical Behavior of Materials program (Dr. D. Stepp, program manager) of the Army Research Office (proposal to Cornell University No. W911NF0710519).

REFERENCES

[1] S. Malinov and W. Sha, Application of artificial neural networks for modelling correlations in titanium alloys, Mater Sci Eng A 365 (2004), 202–211.

[2] S. Kar, T. Searles, E. Lee, G. B. Viswanathan, J. Tiley, R. Banerjee, and H. L. Fraser, Modeling the tensile properties in β-processed α/β Ti alloys, Metallogr Mater Trans A 37A (2006), 559–566.

[3] R. O. Duda, P. E. Hart, and D. G. Stork, Pattern Classification, (2nd ed.), New York, John Wiley and Sons, 2001.

[4] D. Pelleg and A. W. Moore, X-means: Extending K-means with Efficient Estimation of the Number of Clusters, ICML '00: Proceedings of the Seventeenth International Conference on Machine Learning, San Francisco, CA, USA, Morgan Kaufmann Publishers Inc., 2000, 727–734.

[5] R. E. Kass and L. Wasserman, A reference Bayesian test for nested hypothesis and its relationship to the Schwarz criterion, J Am Stat Assoc 90 (1995), 773–795.

[6] A. Kumar and P. R. Dawson, Computational modeling of F.C.C. deformation textures over Rodrigues' space, Acta Mater 48 (2000), 2719–2736.

[7] N. R. Barton, D. E. Boyce, and P. R. Dawson, Pole figure inversion using finite elements over Rodrigues space, Texture Microstruct 35(2) (2002), 113–144.

[8] S. Acharjee and N. Zabaras, A proper orthogonal decomposition approach to microstructure model reduction in Rodrigues space with applications to the control of material properties, Acta Mater 51(18) (2003), 5627–5646.

[9] S. Ganapathysubramanian and N. Zabaras, Design across length scales: A reduced-order model of polycrystal plasticity for the control of microstructure-sensitive material properties, Comput Methods Appl Mech Eng 193(45–47) (2004), 5017–5034.

[10] S. S. Ravindran, Adaptive reduced-order controllers for a thermal flow system using proper orthogonal decomposition, SIAM J Sci Comput 23(6) (2002), 1924–1942.

[11] M. Turk and A. Pentland, Eigenfaces for recognition, J Cogn Neurosci 3(1) (1991), 71–86.

[12] U. F. Kocks, C. N. Tóme, and H. R. Wenk, Texture and Anisotropy: Preferred Orientations in Polycrystals and their Effect on Materials Properties, Cambridge, Cambridge University Press, 2000.

[13] H. J. Bunge, Texture Analysis in Materials Science, (English ed.), London, Butterworth's, 1983.

[14] G. F. Vander Voort, Metallography: Principles and Practice, New York, McGraw-Hill Book Company, 1984.

[15] J. Ohser and F. Múcklich, Statistical Analysis of Microstructures in Materials Science, (1st ed.), New York, John Wiley and Sons, 2000.

[16] D. P. Mika and P. R. Dawson, Polycrystal plasticity modeling of intracrystalline boundary textures, Acta Mater 47(4) (1999), 1355–1369.

[17] V. Sundararaghavan and N. Zabaras, A dynamic material library for the representation of single phase polyhedral microstructures, Acta Mater 52/14 (2004), 4111–4119.

[18] V. Sundararaghavan and N. Zabaras, Classification of three-dimensional microstructures using support vector machines, Comput Mater Sci 32 (2005), 223–239.

[19] V. Sundararaghavan and N. Zabaras, On the synergy between texture classification and deformation process sequence selection for the control of texture-dependent properties, Acta Mater 53 (2005), 1015–1027.

[20] W. B. Hutchinson and J. G. Swift, Anisotropy in some soft magnetic materials, Texture 1 (1972), 117–123.

[21] N. Zabaras, V. Sundararaghavan, and S. Sankaran, An information theoretic approach for obtaining property PDFs from macro specifications of microstructural uncertainty, TMS Lett 3(1) (2006), 1.

[22] S. Sankaran and N. Zabaras, Computing property variability of polycrystals induced by grain size and orientation uncertainties, Acta Mater, 55(7) (2007), 2279–2290.