Classifying the Dimensional Variation in Additive Manufactured Parts From Laser-Scanned Three-Dimensional Point Cloud Data Using Machine Learning Approaches

The objective of this work is to develop and apply a spectral graph theoretic approach for differentiating between (classifying) additive manufactured (AM) parts contingent on the severity of their dimensional variation from laser-scanned coordinate measurements (3D point cloud). The novelty of the approach is in invoking spectral graph Laplacian eigenvalues as an extracted feature from the laser-scanned 3D point cloud data in conjunction with various machine learning techniques. The outcome is a new method that classifies the dimensional variation of an AM part by sampling less than 5% of the 2 million 3D point cloud data acquired (per part). This is a practically important result, because it reduces the measurement burden for postprocess quality assurance in AM—parts can be laser-scanned and their dimensional variation quickly assessed on the shop floor. To realize the research objective, the procedure is as follows. Test parts are made using the fused filament fabrication (FFF) polymer AM process. The FFF process conditions are varied per a phased design of experiments plan to produce parts with distinctive dimensional variations. Subsequently, each test part is laser scanned and 3D point cloud data are acquired. To classify the dimensional variation among parts, Laplacian eigenvalues are extracted from the 3D point cloud data and used as features within different machine learning approaches. Six machine learning approaches are juxtaposed: sparse representation, $k$-nearest neighbors, naive Bayes, support vector machine, and decision tree. Of these, the sparse representation technique provides the highest classification accuracy (F-score $>97\%$). [DOI: 10.1115/1.4036641]

Keywords: additive manufacturing (AM), fused filament fabrication (FFF), dimensional variation, spectral graph theory, Laplacian eigenvalues, machine learning, sparse representation

1 Introduction

1.1 Motivation. Additive manufacturing (AM) offers significant advantages over traditional subtractive and formative manufacturing processes, such as being able to create intricate geometries, eliminating specialized tooling, and reducing material waste [1]. Nevertheless, the lack of part consistency and process repeatability stymies the wider adoption of AM processes [2]. The difficulty in classifying the dimensional variations in AM parts using traditional statistical measurements motivates this research [3]. This experimental observation is exemplified in Fig. 1, which shows the flooded contour plot of a test part printed using FFF polymer AM process. The parts shown in Fig. 1 are made from acrylonitrile butadiene styrene (ABS) polymer material under distinctive FFF process conditions (different infill percentage ($I_i$)). A flooded contour plot translates the dimensional variations of the part to the corresponding spatial locations in terms of hues or colors. The flooded contour plot in Fig. 1 was reconstructed from the laser-scanned data for a test artifact (called circle–square–diamond, see Sec. 3.1) used in the current work. Each plot in Fig. 1 is constituted from close to 2 million 3D point cloud obtained from a laser scanner.

In Fig. 1, the dark colored areas indicate positive deviations, i.e., the part dimensions are larger than the computer-aided design (CAD) design; light colors represent negative deviations. This deviation of the printed part from the CAD model is termed as a dimensional variation. The dimensional variation is further quantified in this work under two statistical parameters called root-mean-square (RMS) deviation and average in-tolerance percentage. These terms are mathematically defined in Sec. 3.3. Smaller RMS deviation translates to smaller (better) dimensional variation; the opposite holds for average in-tolerance percentage. Traditional geometric dimensioning and tolerancing (GD&T) parameters can also be used, but are eschewed in this work. This is because the test part is not a functional component nor is it a part of an assembly. Hence, there is a danger of over tolerancing and assigning features out of functional context with GD&T.
Recently published works in the area have resorted to RMS for assessing the dimensional variation of AM parts [4,5]. Moreover, the difficulty in translating GD&T parameters to the AM domain remains a vexing problem [6,7].

It is visually apparent from Fig. 1 that as the infill percentage \( I_p \) increases, the shape deviation of the printed part progressively deteriorates. Particularly, for the part with \( I_p = 100\% \) shown in Figs. 1(d1) and 1(d2), considerable warping is evident on the edges and top areas. However, it was observed that there is limited ability to differentiate (or classify) between the four parts shown in Fig. 1 with conventional statistical measurements, namely, RMS deviation and average in-tolerance percentage extracted from the point cloud data. The statistical analysis results, as summarized in Sec. 3.4, show that parameters such as RMS deviation and in-tolerance percentage are ill-suited to capture dimensional variations as a function of process parameters; the regression \( R^2 < 35\% \). This limitation was also shown in the authors’ previous work [3].

1.2 Objective and Salient Aspects. To address the above shortcomings, the objective of this work is to differentiate (classify, categorize) AM parts from one another with respect to the severity of their dimensional variations based on sparse sampling from a large laser-scanned dataset of part coordinates (\( \sim 2 \times 10^6 \) data points are acquired for each part). To realize this objective, spectral graph theoretic Laplacian eigenvalues extracted from the laser-scanned data of AM parts are invoked within a machine learning framework, e.g., sparse representation-based classification. From a methodological standpoint, this work is the first to suggest the use of spectral graph theoretic Laplacian eigenvalues as features for classifying AM parts per their severity of dimensional variation in a machine learning framework. This contribution is practically important—instead of scanning an entire part (which can be time-consuming and inefficient), a few samples can be taken and the quality in terms of severity of dimensional variation can be quickly quantified. For instance, the difference in RMS deviation between parts with 70% and 80% infill shown in Fig. 1 is less than 20 µm (\( \sim 1/10 \)th of the set layer height of 150 µm); nevertheless, the proposed approach captured these subtle differences, with accuracy approaching 95% by sampling fewer than 5% of the \( 2 \times 10^6 \) data points.

The present work significantly extends upon the authors’ prior research [3]. This previous work requires sampling of 100% of the part, viz., close to 500,000 data points, and therefore computationally intensive. Additionally, a machine learning approach to classify the part quality was not forthcoming. These shortcomings are addressed in this paper. It is noted that the test parts are made on a consumer-grade FFF printer (Makerbot2X) and relatively inexpensive laser scanner (NextEngine). The authors acknowledge that although the choice of a desktop machine constrains the experimental scope of this paper, however, the mathematical concepts are independent of the hardware and manufacturing process.

The rest of this paper is organized as follows: Section 2 reviews the related literature; Sec. 3 details the experimental studies with FFF additive manufacturing process (AM), acquisition of 3D point cloud data through laser scanning, and subsequent statistical analysis of the data; Sec. 4 elucidates the spectral graph theoretic approach to extract Laplacian eigenvalues from the 3D point cloud data; Sec. 5 illustrates the application of Laplacian eigenvalues for categorization of dimensional variation of AM parts in different supervised classification-based machine learning approaches; and Sec. 6 summarizes the findings of this work.

2 Review of the Related Research

With the advent of inexpensive desktop/handheld 3D laser and structured light scanners, complex geometries can be readily transformed into CAD models [8]. Furthermore, with the considerable improvement in precision of laser scanning techniques and concomitant efficient data analysis algorithms, optical 3D point cloud acquisition is being increasingly considered for metrology purposes [9,10]. There are several review papers on the application of optical scanning in AM [9,11,12]. Raja et al. [13] studied aerospace parts printed via five different AM processes, namely, stereolithography, selective laser sintering, laser melting, material jetting, and FFF and subsequently acquired data on the geometry from a coordinate measuring machine (CMM) machine. They concluded that FFF parts routinely showed the poorest dimensional precision compared to other AM processes. The FFF literature is replete with studies on the effect of the process parameters on the surface roughness and strength of the parts [14–17], but few on dimensional variation.

Lanzotti et al. [18] studied the dimensional variation in FFF process, based on varying three process parameters: layer thickness, flow rate, and deposition speed. Test parts were built and measured using a laser scanner. All three main factors were found...
Phase (replicates).

The NAS 979 standard artifact was first used in the AM context material defects of a component. In order to overcome this con-

To be statistically significant; pertinently, the interaction effects were found to be insignificant.

In the context of using point cloud data for assessment of geometric integrity, Kainat et al. [19] developed a method for post-

metric integrity, Kainat et al. [19] developed a method for post-

Infill percentage (at four levels) and extruder temperature (at five levels), experiment protocol has two input variables, namely, infill percentage (If) and extruder temperature (Te). To assess AM process performance [26]. GD&T characteristics, such as squareness, size (length, breadth, and thickness), and circularity, were used to compare the quality of parts produced using electron beam melting and direct metal laser sintering metal AM processes. For the experimental tests reported in the current work, the NAS 979 standard test artifact was simplified so that it was easier to make in terms of build time and feature complexity; this simplified design is called as circle–square–diamond, abbreviated as CSD. The CSD artifact is shown in Fig. 2. Each part takes between 35 and 50 min to build, depending on infill percentage; higher infill percentage takes longer time.

### 3.1 Design of the Circle–Square–Diamond (CSD) Test Artifact

The test artifact used for experimental studies is a simplification of the NAS 979 standard artifact. The NAS 979 artifact is traditionally used to test the accuracy of machining centers [26]. Recently, Cooke and Soons at NIST used the NAS 979 part to assess AM process performance [26]. GD&T characteristics, such as squareness, size (length, breadth, and thickness), and circularity, were used to compare the quality of parts produced using electron beam melting and direct metal laser sintering metal AM processes. For the experimental tests reported in the current work, the NAS 979 standard test artifact was simplified so that it was easier to make in terms of build time and feature complexity; this simplified design is called as circle–square–diamond, abbreviated as CSD. The CSD artifact is shown in Fig. 2. Each part takes between 35 and 50 min to build, depending on infill percentage; higher infill percentage takes longer time.

### 3.2 Design of Experiments

The circle–square–diamond (CSD) test artifacts described in Sec. 3.1 are FFF printed under varying FFF parameters, namely, infill percentage (If) and extruder temperature (Te) conditions as well as a statistical design of experiment plan. The design of experiment protocol has two input variables, namely, infill percentage (at four levels) and extruder temperature (at five levels), in a full factorial scheme (done in five phases plus confirmation runs for a total of 20 distinct treatment conditions (= 4 x 5)). Two parts were printed (left and right side of the print bed) at each experimental run. There are totally 34 experimental runs, and 68 parts printed, scanned, and used in the analysis, as depicted in

### Table 1 Printing conditions of FFF parts in each phase of the design of experiment [27]. Each print contains two parts (replicates).

<table>
<thead>
<tr>
<th>Phase</th>
<th>Print number</th>
<th>Extruder temperature (Te) °C</th>
<th>Infill percentage (If)</th>
<th>Phase</th>
<th>Print number</th>
<th>Extruder temperature (Te) °C</th>
<th>Infill percentage (If)</th>
<th>Phase</th>
<th>Print number</th>
<th>Extruder temperature (Te) °C</th>
<th>Infill percentage (If)</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>1</td>
<td>225</td>
<td>80</td>
<td>2</td>
<td>13</td>
<td>235</td>
<td>80</td>
<td>5</td>
<td>25</td>
<td>220</td>
<td>70</td>
</tr>
<tr>
<td>2</td>
<td>2</td>
<td>225</td>
<td>90</td>
<td>14</td>
<td>235</td>
<td>100</td>
<td>90</td>
<td>26</td>
<td>240</td>
<td>100</td>
<td>70</td>
</tr>
<tr>
<td>3</td>
<td>3</td>
<td>225</td>
<td>100</td>
<td>15</td>
<td>235</td>
<td>100</td>
<td>100</td>
<td>27</td>
<td>230</td>
<td>100</td>
<td>100</td>
</tr>
<tr>
<td>4</td>
<td>4</td>
<td>230</td>
<td>80</td>
<td>16</td>
<td>230</td>
<td>90</td>
<td>90</td>
<td>28</td>
<td>230</td>
<td>100</td>
<td>100</td>
</tr>
<tr>
<td>5</td>
<td>5</td>
<td>230</td>
<td>90</td>
<td>17</td>
<td>240</td>
<td>90</td>
<td>90</td>
<td>Confirmation</td>
<td>29</td>
<td>220</td>
<td>70</td>
</tr>
<tr>
<td>6</td>
<td>6</td>
<td>230</td>
<td>100</td>
<td>3</td>
<td>230</td>
<td>80</td>
<td>80</td>
<td>experiment</td>
<td>30</td>
<td>230</td>
<td>70</td>
</tr>
<tr>
<td>7</td>
<td>7</td>
<td>235</td>
<td>80</td>
<td>19</td>
<td>225</td>
<td>70</td>
<td>70</td>
<td>31</td>
<td>235</td>
<td>70</td>
<td></td>
</tr>
<tr>
<td>8</td>
<td>8</td>
<td>235</td>
<td>90</td>
<td>20</td>
<td>235</td>
<td>70</td>
<td>70</td>
<td>32</td>
<td>225</td>
<td>80</td>
<td></td>
</tr>
<tr>
<td>9</td>
<td>9</td>
<td>235</td>
<td>100</td>
<td>21</td>
<td>220</td>
<td>80</td>
<td>80</td>
<td>33</td>
<td>230</td>
<td>80</td>
<td></td>
</tr>
<tr>
<td>10</td>
<td>10</td>
<td>240</td>
<td>90</td>
<td>22</td>
<td>240</td>
<td>80</td>
<td>80</td>
<td>34</td>
<td>235</td>
<td>80</td>
<td></td>
</tr>
<tr>
<td>11</td>
<td>11</td>
<td>220</td>
<td>90</td>
<td>4</td>
<td>230</td>
<td>70</td>
<td>70</td>
<td>35</td>
<td>230</td>
<td>80</td>
<td></td>
</tr>
<tr>
<td>12</td>
<td>12</td>
<td>230</td>
<td>70</td>
<td>24</td>
<td>230</td>
<td>100</td>
<td>100</td>
<td>36</td>
<td>225</td>
<td>80</td>
<td></td>
</tr>
</tbody>
</table>
3.3 Laser Scanning of FFF Test Parts—Extracting RMS and In-Tolerance Percentage. After FFF, each part is laser scanned to obtain the RMS and in-tolerance percentage. The scanner used is the NextEngine HD laser scanner, the QA SCAN 4.0 software is used to 3D coordinate data. The scanner uses eight 10 mW Class 1M red lasers at 650 nm in two arrays of four diodes. These arrays rotate together and project lines onto the object, and two 3.0-megapixel cameras capture the curve of the lasers over the part. This desktop scanner was used in macro mode which is capable of scanning objects up to 5.1 in × 3.8 in (~130 mm × 96 mm) in a single scan at ±0.005 in (~125 μm) accuracy.

Laser scanning consists of several steps such as point cloud extraction from the CAD design, alignment of the measured scan to CAD, and the subsequent analysis, each of which has its own literature [29–31]. In practice, the laser scanning is a heuristic process that requires a carefully attuned procedure to obtain consistent results; particularly, in the way the part is aligned to the CAD model. The alignment step requires matching of at least four points from the raw point cloud data with CAD model. The following method showed the least variability. Four points each on the square and diamond portions are used to align the part as depicted in Fig. 3. Additionally, laser scanning was conducted on a sturdy vibration-free table, in a dark room, and by coating the part with a thin layer of antireflective gray modeling paint.

The point cloud datasets acquired from the 3D laser scanner contain six columns (Table 3). The first three columns are the raw scanned points’ coordinates (x, y, and z) captured by the laser scanner, and the next three columns are the deviation of these coordinates from corresponding points in the CAD design file (the CAD coordinate subtracted from the corresponding scan coordinate after fitting). The negative sign indicates that the part was smaller than the intended CAD model in a particular direction. Negative deviations were represented by blue colors in Fig. 1.

The RMS deviation measures the difference of the laser-scanned part coordinate from the part CAD model. RMS deviation is estimated as follows:

\[
\sqrt{\frac{\sum_{i=1}^{N} (t_{xi} - s_{xi})^2 + (t_{yi} - s_{yi})^2 + (t_{zi} - s_{zi})^2}{N}}, \quad \text{where } t \text{ refers to the coordinate of the tessellated CAD model and } s \text{ is the coordinate obtained from the laser scan, with the subscripts } x, y, \text{ and } z \text{ denoting the axes for a point } i; \quad N \text{ is the total number of scan points acquired; } s_{xi}, s_{yi}, \text{ and } s_{zi} \text{ are the first three columns in Table 3; } (t_{xi} - s_{xi}) \text{ is the deviation along the } x\text{-axis (fourth column in Table 3), and so on.}
\]

The average in-tolerance percentage is the proportion of points with less than 0.25 mm arithmetic average deviation from their corresponding reference point (average of last three columns in Table 3). A reference tolerance value of 0.25 mm (0.010 in) was selected because it is twice the accuracy specified for the scanner (0.125 mm, 0.005 in).

Table 2 Printing process parameters which are kept constant in the experiment

<table>
<thead>
<tr>
<th>Process parameter</th>
<th>Setting</th>
</tr>
</thead>
<tbody>
<tr>
<td>Layer height</td>
<td>0.15 mm</td>
</tr>
<tr>
<td>Infill pattern</td>
<td>Hexagon</td>
</tr>
<tr>
<td>Bed temperature</td>
<td>110°C</td>
</tr>
<tr>
<td>Raster orientation</td>
<td>(0, 90)</td>
</tr>
</tbody>
</table>

Table 1. This experimental plan is detailed by Dsouza [27]. Note that factors such as wear of nozzle diameter, vibration in the printer bed, stepper motor precision during rastering, and wear of mechanical parts are inherent noise factors, which are not accounted for in this work; Table 2 lists the constant print parameters.

There are several controllable parameters in FFF process apart from the extruder temperature (T_e) and infill percentage (I_i), such as layer height, the number of shells, build plate temperature, and chamber temperature (through an external heater). Some factors such as layer height have a significant effect on the dimensional variation, which is easy to discern and explain. For instance, increased layer height leads to poor part resolution and hence poor dimensional integrity [28].
difference between FFF parts. More evolved approaches to demarcate part dimensional variations from 3D point cloud data are needed.

4 Feature Extraction From Point Cloud Data Using Spectral Graph Theory

In this section, the difficulty in discriminating part geometric accuracy through conventional statistical analysis as shown in Table 4 is resolved using the methodology proposed herewith. In Sec. 4.1, the laser-scanned point cloud data for each part (denoted as Ψ) acquired in Sec. 3.3 are mapped into an unweighted and undirected network graph (G), and then, the Laplacian eigenvalues (λ) from the graph are extracted, respectively [3]. In Secs. 4.2 and 4.3, physical interpretations of the spectral graph theoretic approach to AM parts are tendered. The Laplacian eigenvalues are used in supervised classification-based modeling for discriminating the geometric integrity of AM parts in Sec. 5.

In the proposed graph theoretic approach, only the deviation data (last three columns in Table 3) are used to build the unweighted, undirected graph. Given a sample dataset Ψ ∈ ℝ\(^d\)\(^×\)\(k\), where \(d\) represents the number of dimensions (for this study, \(d = 3\), i.e., the deviations in the three directions), and \(k\) is the chosen sample size shown in Eq. (1). For instance, \(ψ_i\) is the \(i\)th sample in the \(x\)-axis. The sample size \(k\) in Eq. (1) should be determined in such a way that while \(k\) should be large enough, its corresponding graph is representative of the overall population of 3D point cloud data; however, it should not be too large or else the computation will become overwhelming (typically \(k = 1000\) is a reasonable choice). The procedure for setting \(k\) for a practical scenario is explained in Sec. 5.4 and illustrated in the Appendix with an example

\[
Ψ = \begin{bmatrix}
ψ_1^1 & ψ_1^2 & ψ_1^3 \\
ψ_2^1 & ψ_2^2 & ψ_2^3 \\
\vdots & \vdots & \vdots \\
ψ_k^1 & ψ_k^2 & ψ_k^3
\end{bmatrix} \tag{1}
\]

4.1 Extracting the Laplacian Eigen Spectrum From the 3D Point Cloud Data. The aim of this section is twofold: (i) to map the matrix Ψ into an unweighted and undirected graph, i.e., achieve the transform Ψ → G = (V, E), where V \(∈\) \(ℝ^d\) \(×\) \(k\) represent the vertices and edges, respectively, and (ii) subsequently extract the Laplacian eigenvalues (λ). The procedure described in Eqs. (2)–(10) is nearly identical to the one described in the authors’ recent publication [3]; it is repeated here to preserve continuity.

In order to obtain the graph (G), the first step is to make the pairwise Euclidean comparison matrix \(Y \in ℝ^k\times k\), which is a symmetric matrix containing the distances \(ε_{ij}\) between every pair of data points \(ψ_i, ψ_j \in ℝ^d\), where \(i, j \in (1, 2, \ldots, k)\). This is essentially the pairwise distance between each row vector of the matrix Ψ. The radial basis distance function is used to calculate the pairwise \(ε_{ij}\) (Eqs. (2) and (3)). In Eq. (2), \(σ^2\) is the total variance of the pairwise Euclidean distance matrix (Y)

\[
ε_{ij} = e^{-\frac{(ψ_i - ψ_j)^2}{σ^2}} \tag{2}
\]

\[
Y^{k×k} = [ε_{ij}] \tag{3}
\]

Table 4 P-values and model \(R^2\) from response surface analyses of RMS and in-tolerance percentage (data from 68 test parts)

<table>
<thead>
<tr>
<th>Quantifiers</th>
<th>(I_f)</th>
<th>(T_v)</th>
<th>(T_v \times T_e)</th>
<th>(I_f \times I_f)</th>
<th>(T_v \times I_f)</th>
<th>(R^2) (%)</th>
<th>(R^2)-adj. (%)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Root-mean-square deviation (RMS)</td>
<td>0.00</td>
<td>0.67</td>
<td>0.09</td>
<td>0.80</td>
<td>0.24</td>
<td>31.80</td>
<td>26.30</td>
</tr>
<tr>
<td>In-tolerance percentage</td>
<td>0.00</td>
<td>0.66</td>
<td>0.18</td>
<td>0.26</td>
<td>0.36</td>
<td>34.64</td>
<td>29.36</td>
</tr>
</tbody>
</table>

Journal of Manufacturing Science and Engineering
The similarity matrix $S$ are drawn between two nodes only if they are connected, i.e., if their internal structure. If the supposition of dimensional variation being manifested in parts differing in their infill percentages. As described previously, the similarity matrix $S$, which is the representative unweighted and undirected network graph $G$; $G$ is constituted of the from binary elements $w_{ij}$. Further theoretical discussions on the threshold are detailed in Ref. [32].

After this step, the similarity matrix $(S)$ is built by applying a threshold function $\Theta(\epsilon_{ij}) = w_{ij}$ on the pairwise comparison $\epsilon_{ij}$ as shown in Eq. (4). For this, the threshold $r$ is set as $r = (\Sigma_{i=1}^{n} \Sigma_{j=1}^{n} \epsilon_{ij})/N^2$, which is the arithmetic average of $\epsilon_{ij}$. Equation (5) is the similarity matrix $(S)$, which is the representative unweighted and undirected network graph $G$; $G$ is constituted of the from binary elements $w_{ij}$. Further theoretical discussions on the threshold are detailed in Ref. [32].

$$\Theta(\epsilon_{ij}) = w_{ij} = \begin{cases} 1, & \epsilon_{ij} \leq r \\ 0, & \epsilon_{ij} > r \end{cases}$$

(4)

$$S^{k \times k} = [w_{ij}]$$

(5)

The similarity matrix $S^{k \times k} = [w_{ij}]$ is a binary symmetric matrix which includes the structure of the mapped graph $G$. This graph has $k$ nodes, and if $w_{ij} = 1$, nodes $i$ and $j$ are connected. The binary nature of $S$ realizes an unweighted graph, while the symmetric structure of $S$ signifies an undirected graph.

The central premise is that the topology of the graph will change depending on the geometric integrity. To portray this premise, consider Fig. 6 in which a small portion (40 nodes) of the similarity matrix $S$ from Eq. (5) is depicted as a graph for two parts differing in their infill percentages. As described previously in Sec. 3.4, it was experimentally observed that as the infill percentage increases, the dimensional variation worsens (increases). If the supposition of dimensional variation being manifested in the topology of the graph holds, then the graphs reconstructed from 3D point cloud data for two infill conditions must differ in their internal structure.

In Fig. 6, the nodes of the graph (each row or column of the matrix $S$ is a node) are arranged along a unit circle, and the edges are drawn between two nodes only if they are connected, i.e., if $w_{ij} = 1$ for an element in $S$. The network graph in Fig. 6(a) is for a part with higher infill percentage ($I_f = 90\%$); Fig. 6(b) corresponds to a part with $I_f = 70\%$. From Fig. 6, it is evident that the network connectivity of the graph with lower infill ($I_f = 70\%$, Fig. 6(b)) is significantly sparser than that for the part with higher infill ($I_f = 90\%$, Fig. 6(a)). Figure 6 demonstrates that there is a link between the part dimensional variation and the network topology. The physical reasoning for this link is tendered in Sec. 4.2. Accordingly, the next aim is to extract the structural features from this graph so that the difference in topological connectivity observed in Fig. 6 for the two infill levels is quantified.

To extract the Laplacian eigenvalues ($\lambda^*$), first, the node degree of the graph $G$ is calculated, which is essentially the number of nodes directly connected to a particular node. Thereafter, the degree vector ($d_i = \in [R^{1 \times k}]$) is formed by rowwise summation of $w_{ij}$ as shown in Eq. (6); Eq. (7) then transforms the degree vector into a diagonal matrix called degree matrix $(D)$. Finally, Eqs. (8) and (9) represent the Laplacian matrix $(L)$, which is obtained by subtracting the similarity matrix $(S)$ from the degree matrix $(D)$ and the normalized Laplacian matrix $(\mathcal{L})$, respectively.

$$d_i = \sum_{j=1}^{k} w_{ij} \forall i, j \in (1...k)$$

(6)

$$D^{k \times k} = \text{diag}[d_i] = \begin{bmatrix} d_1 & \cdots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \cdots & d_k \end{bmatrix}$$

(7)

$$L^{k \times k} = D - S$$

(8)

$$\mathcal{L}^{k \times k} = D^{-1/2} \times L \times D^{-1/2}$$

(9)

$$L v = \lambda^* v$$

(10)

It is important to note that in forming the normalized Laplacian matrix $(\mathcal{L})$, $D^{1/2} = \text{diag}(1/\sqrt{d_1}, ..., 1/\sqrt{d_k})$. From the normalized Laplacian matrix, the eigenvalues are obtained using the following equation:

$$L v = \lambda^* v$$

(10)

In the above equation, $v \in [R^{k \times k}]$ are the Laplacian eigenvectors, and $\lambda^* \in [R^{k \times k}]$ are the Laplacian eigenvalues. Several topological characteristics of a network graph are manifested in these Laplacian eigenvalues [33].

4.2 Physical Interpretation of the Spectral Graph Theoretic Procedure. Having described how to convert a 3D point cloud dataset into a network graph $G$, and subsequently, the quantification of the topology of the network $G$ in terms of the Laplacian eigenvalues $\lambda^*$ in Sec. 4.1, the aim of this section is to provide a physical interpretation of these steps in the context of dimensional variation of an AM component. This will explain why the topological connectivity of the graphs shown in Fig. 6 is distinctive for different levels of infill percentage.

The key to understanding this connection lies in further delineating the kernel function $(\Omega)$ and threshold function $(\Theta)$ in Eqs. (4) and (5), respectively. The kernel function $\Omega$ is a Gaussian radial basis function with the dispersion parameter set as the standard deviation $\sigma_3$ of the Euclidean distance matrix $(E)$, and $\Theta$ is a Heaviside hard threshold function. Equations (4) and (5) can be interpreted in terms of Fig. 7.

Figure 7(a) shows a completely connected graph, i.e., every node is connected to every other node. In physical terms, the different node shades can be taken to represent the `weight or magnitude of deviations from the ideal dimensions obtained from the 3D point cloud data. In Fig. 7, only a few nodes are shown, whereas in reality, there will be as many nodes as there are coordinate locations (rows of $\Psi$ in Eq. (1)) in the 3D point cloud data.
Initially, all nodes are connected to edges as shown in Fig. 7(a); the edges weights are estimated from Eq. (3). The threshold function $\Theta$ in Eq. (4) is used to prune certain edges based on their edge weights $w_{ij}$; the pruning procedure is depicted in Fig. 7(b). Pruning results in the final graph network shown in Fig. 7(c), which is stored in the similarity matrix $S = |w_{ij}|$ (Eq. (5)). To understand the nature of pruning, consider the following limits on the pairwise comparisons resulting from $\Theta$ in Eq. (11):

$$\lim_{|F_i - F_j| \rightarrow 0} w_{ij} = 1, \quad \text{and} \quad \lim_{|F_i - F_j| \rightarrow \infty} w_{ij} = 0$$

$$\therefore, \quad w_{ij} = \Theta(0, 1)$$

The above bound on $w_{ij}$ implies that if two nodes, say $i$ and $j$, have similar weights, then their edge weight $w_{ij}$ will approach 1, and the more different the nodes $i$ and $j$ in terms of their magnitude, the closer $w_{ij}$ to 0. Subsequently, the Heaviside step function $\Theta$ in Eq. (4) has the effect of pruning or removing those edges that connect nodes of similar magnitude. Thus, only those nodes that are dissimilar in magnitude will be joined with an edge. The pruning procedure is illustrated in Figs. 7(b) and 7(c), where a light shade node connecting to another light shade node is identified with dashed lines and is subsequently pruned (Fig. 7(c)).

From a physical perspective, this implies that if a given component has smaller dimensional variations from the CAD design blueprint, then the resulting network graph will be sparse and, consequently, have fewer edge connections. This is because much of the nodes, as in 3D point cloud deviations, will be of similar magnitude. On the other hand, if the dimensional variations are large, then the network graph will be replete with dense edge connections. Hence, the network graph for $I_f = 70\%$ (6(b)) is significantly sparser than for $I_f = 90\%$ (Fig. 6(a)); recollect that the parts with $I_f = 70\%$ had low dimensional variation than those with $I_f = 90\%$ (compare Figs. 1(a) and 1(c)). Thus, the dimensional variation of a part has been implicitly consolidated in terms of the edge density or edge connectivity of a network graph $G \equiv (V,E)$.

4.3 Justification for Using the Laplacian Eigenvalues. A justification for using the Laplacian eigen spectrum for differentiating the dimensional variation of AM parts is motivated from the area of signal processing. The rationale is similar to the discrete Fourier transform, which is essentially a decomposition of a signal (or vector) into the frequency domain via convolution of the signal with a sinusoidal basis; the Fourier decomposition of the signal is contingent on the orthonormal property of the sinusoidal basis. In a similar vein, in the spectral graph domain, because the Laplacian matrix $(\mathcal{L})$ is a real symmetric matrix, its eigenvectors $\mathbf{v}$ are an orthonormal basis, i.e., $\mathbf{v}_i \perp \mathbf{v}_j, \cdots \perp \mathbf{v}_n$; $\langle \mathbf{v}_i, \mathbf{v}_j \rangle = 0$; $\langle \mathbf{v}_i, \mathbf{v}_i \rangle = 1$ [34,35]. On account of this property, it is possible to define a graph Fourier transform $\mathcal{G}(\mathbf{\psi})$ on a signal $\mathbf{\psi} \in \mathbb{R}^N$ (consider that there is only dimensional point cloud data for simplicity). Assuming that the graph has no isolated nodes, the spectral graph Fourier transform is defined as follows [34–36]:

$$\mathcal{G}(\mathbf{\psi}) = \left[\mathbf{\psi}^T, \mathbf{\psi}_1\right], \cdots \left[\mathbf{\psi}^T, \mathbf{\psi}_n\right] = \mathbf{a}_k$$

(12)

From the above equation, the eigenvectors of the Laplacian are depicted akin to Fourier decomposition, with coefficients $a_k$ ($a_k \in \mathbb{R}$, because $\mathbf{\psi}^T, \mathbf{\psi}_k \in \mathbb{R}$). Moreover, because all the Laplacian eigenvalues are real, each of the coefficients $a_k$ may be expressed as a scalar multiple of two real numbers $p, q$, and a corresponding eigenvalue $\lambda_i$

$$\langle \mathbf{\psi}^T, \mathbf{\psi}_i \rangle = a_i = p_i \lambda_i, \quad \forall i > 1$$

In other words, the Laplacian eigenvalues capture the intrinsic patterns in the 3D point cloud data in the spectral graph domain, which is analogous to the manner in which the Fourier transform casts the temporal dynamics of a signal in frequency space. Instead of using conventional statistical features (such as mean, standard deviation, skewness, etc.) to capture the dimensional variation from the 3D point cloud deviation data (a signal in three domains), the spectral graph Laplacian can be used instead. This explains the reason for using the Laplacian eigenvalues as discriminants for part dimensional variation. One of the key propositions of this paper is to use a small subset, or the so-called sparse sample, of the complete Laplacian eigenvalues ($\lambda^*$, Eq. (10)). Section 5 details how the Laplacian eigenvalues are used for discriminating the part dimensional variation.

5 Machine Learning Classification for Assessing Dimensional Variation of AM Parts

The aim of this section is to use a subset of the Laplacian eigenvalues ($\lambda^*$, Eq. (10)) as features in a supervised classification-based machine learning framework to predict the part quality of FFF parts from the laser-scanned point cloud data as described in Sec. 3. In pursuit of this aim, two sampling strategies are developed to address the effect of part size and geometry.

From the empirical studies in Sec. 3, it was recognized that parts printed with 90% and 100% infill generally depict among the worse dimensional variation characteristics compared to parts with infill 70% and 80%. However, it was not possible to distinguish between parts produced at the four infill levels using statistical response surface modeling with RMS and in-tolerance percentage parameters as discriminants (output or dependent variable) as shown in Table 4. The research question addressed is twofold:

1) How to use the several hundred Laplacian eigenvalues ($\lambda^*$) to discriminate (classify) the AM part dimensional variation?
2) How to minimize the amount of 3D point cloud data that is used for the analysis?

In this section, a certain temperature ($T_c = 230^\circ$ C) is chosen, and point cloud data from eight parts printed with four different infill percentages ($I_f = 70\%, 80\%, 90\%, 100\%$) were considered as belonging to the four distinct classes contingent on their dimensional variation. This ensures that confounding of the two-factor ($I_f$ and $T_c$) effects are avoided—in other words, the classification decimation is distinct for each infill percentage setting. Accordingly, the parts with $I_f = 70\%$ are labeled as superior, $I_f = 80\%$ as acceptable, $I_f = 90\%$ as mediocre, and $I_f = 100\%$ as reject.

5.1 The Sparse Representation-Based Approach. In this section, the sparse representation-based classification approach is briefly summarized as it is found to be the classifier of choice later in Sec. 5.6 [37]. A much more detailed explanation is available in the authors’ recent work [37]. Considering the system of independent linear equations

$$\mathbf{Y} = \mathbf{A}\mathbf{\phi} + \mathbf{\eta}$$

(14)

In the above equation, $\mathbf{\phi} \in \mathbb{R}^{N \times 1}$ is a vector of unknowns, $\mathbf{Y} \in \mathbb{R}^{m \times 1}$ is a vector of measurements, $\mathbf{A} \in \mathbb{R}^{m \times N}$ contains coefficients, and $\mathbf{\eta} \in \mathbb{R}^{m \times 1}$ is the residual error. We note that Eq. (14) is an underdetermined system, i.e., $m \ll N$. In other words, the number of equations is lesser than the number of unknowns. Matrix $\mathbf{A}$ contains the training data and is called the design or information matrix. In the context of AM part 3D point cloud data, the matrix $\mathbf{A}$ contains Laplacian eigenvalues for AM parts belonging to different classes as per their dimensional variations ($\lambda^*$, Eq. (10)). In this work, parts with different infill percentages...
are considered to belong to distinct classes. Matrix $Y$, called the test vector, contains the Laplacian eigenvalues for a part whose geometric integrity is as yet unknown and must be classified. The aim is to solve for $\tilde{\omega}$, called the classification membership coefficients which will then indicate the geometric integrity of a sample part whose eigenvalues are contained in $Y$ by minimizing the residual error $\eta$. A worked example is shown in the Appendix.

Equation (15) shows the design matrix $A \in \mathbb{R}^{n \times N}$ in more detail, which includes the features, i.e., Laplacian eigenvalues ($\lambda'$), of training samples taken from different classes. We define $c \in \{1, 2, \ldots, C\}$ as the class index ($C = 4$) in this study. A $C$-class problem means that there are $C$ distinct entities that should be discriminated from each other given a set of features. In the context of this work, the $C$-class problem refers to parts with distinct dimensional variations (superior, acceptable, mediocre, and reject), which are obtained at the four different infill levels ($C = 4$). The number of columns in the design matrix is $N = n \times C$, where $n$ is the number of training sample (one or more samples taken from physical parts) available for each class. Each element of this matrix is shown by $(\tau_i)^c$ as $ith$ feature $i \in \{1, 2, \ldots, m\}$ of $jth$ sample $j \in \{1, 2, \ldots, n\}$ from class $c \in \{1, 2, \ldots, C\}$. Note that $(\tau_i)^c$ is an eigenvalue of the Laplacian matrix $\mathbf{L}$ from Eq. (10). Then, the matrix $A$ is constructed in block matrix form as $A = [A_1, A_2, \ldots, A_C]$; the expansion of which is shown below.

$$A = \begin{bmatrix}
  [A_1] \\
  (\tau_1)^1,1 \ldots (\tau_1)^h,1 \\
  \vdots \\
  (\tau_m)^1,1 \ldots (\tau_m)^h,1 \\
  [A_2] \\
  (\tau_1)^1,2 \ldots (\tau_1)^h,2 \\
  \vdots \\
  (\tau_m)^1,2 \ldots (\tau_m)^h,2 \\
  \vdots \\
  [A_C] \\
  (\tau_1)^1,C \ldots (\tau_1)^h,C \\
  \vdots \\
  (\tau_m)^1,C \ldots (\tau_m)^h,C
\end{bmatrix}$$

(15)

The classification membership coefficients are defined as

$$\tilde{\omega} \overset{\text{def}}{=} [w_1^T \ w_2^T \ \cdots \ w_C^T]^T$$

(16)

where $w_c = [w_1^c \ w_2^c \ \cdots \ w_n^c]^T \in \mathbb{R}^n$ is the vector containing the classification membership coefficients corresponding to the $c$ th class. For an ideal case, if it is presumed that observation $Y$ truly belongs to class $c$, then the corresponding $w_c$ should have nonzero elements, and consequently, all other $w_k$’s ($\forall k \in \{1, 2 \ldots, C\} \cap \{c\}$ in the matrix $\tilde{\omega}$ should contain zero elements. Based on this intuition, it follows that vector $\tilde{\omega}$ is inherently sparse; this is called the sparsity condition. This estimate of the vector $\tilde{\omega}$ under the sparsity condition for an underdetermined system of equations is called sparse estimation, and when applied to classification, it is called sparse representation-based classification. There are several sparse solution solvers to obtain $\tilde{\omega}$, such as least absolute shrinkage and selection operator (LASSO) [38], orthogonal matching pursuit [39], and Bayesian learning [40]. In this paper, the LASSO method was applied.

The approach works as follows. Suppose a new point cloud data sample of a part belonging to an unknown class $c' \in \{1, 2, \ldots, C\}$ is obtained. The corresponding eigenvalues ($\lambda'$) are extracted from the point cloud using Eq. (10), and the classification test vector $\bar{Y} \in \mathbb{R}^{n \times z}$ is created. Next, $\tilde{\omega}$ is estimated using LASSO. For each class, an indicator function $\delta_c(\tilde{\omega})$ is applied to transform the vector $\tilde{\omega}$ into $\delta_c(\tilde{\omega}) = [\delta_c(0^T), \delta_c(w_2^T), \ldots, \delta_c(w_C^T)]$ in which, except subvector $w_c$, other elements of the vector $\tilde{\omega}$ are forced to zero, and the class index ($c'$) of the vector $\bar{Y}$ can be obtained from the following equation:

$$c' = \arg\min_c (A\delta_c(\tilde{\omega}) - \bar{Y}) = \arg\min_c \|\eta\|$$

(17)

5.2 Choosing the Laplacian Eigenvalues. In the proposed Laplacian eigenvalue sparse representation algorithm, one critical question is: What and how many of the eigenvalues ($\lambda'$) should be utilized as features to ensure discrimination between classes? Particularly, a set of eigenvalues whose indices are similar for data points from the same class, while being dissimilar between data points from different classes, is desired. To choose a proper subset of eigenvalue spectra as the classification features, the concept of relative deviation in eigenvalues as described by Zhan et al. [41] is used.

According to the relative deviation concept, the number of Laplacian eigenvalues to use as the feature set for the classification problem is set to $\theta$. These $\theta$ are the first $\theta/2$ and the last $\theta/2$ indices excluding the first Laplacian eigenvalue (because the first Laplacian eigenvalue is zero ($\lambda_1 = 0$)). It is assumed that $\theta$ is an even number; $A^h \in \mathbb{R}^{n \times 2\theta}$, $h \in \{1, 2, \ldots, n\}$ as shown in Eq. (18) denotes the vector of chosen eigenvalues for the $h$ th sample part. In this work, the first and last ten eigenvalues ($\theta = 20$) were chosen based on offline analysis.

$$A^h = [a_{1,1}^h, a_{1,2}^h, \ldots, a_{2,1}^h, a_{2,2}^h, \ldots, a_{10,1}^h, a_{10,2}^h]^T$$

(18)

5.3 Data Sampling. The aim of this section is to devise approaches to sample the 3D point cloud data. The choice of sampling regions on the part surface is crucial. For instance, since warpage is a common defect in FFF parts, the dimensional variation of corners can capture this effect. Likewise, deviation near a circular shape may directly correspond to the circular interpolation error in the machine. Therefore, certain regions on the part are representative of physical phenomena that may be occluded with purely random sampling. Indeed, the trivial approach of random sampling was tried; it was found that the classification accuracy ($F$-score, see Sec. 5.5) was less than 30%. Therefore, instead of instinctive random sampling, two sampling scenarios are suggested that are selective to particular regions on an AM part.

In this section, $R$ is defined as the number of areas to be sampled on a given part. Defining a value for $R$ is a heuristic, part-dependent choice that will be explained later for each proposed sampling scenario in detail (Sects. 5.3.1 and 5.3.2). The chosen eigenvalue vector ($A^h$) for area index $r \in \{1, 2, \ldots, R\}$, $h \in \{1, 2, \ldots, n\}$ sample part, and the class index $c \in \{1, 2, \ldots, C\}$ is denoted by $A^h_r \in \mathbb{R}^{n \times z}$, where $C$ is the number of classes. It is reiterated that each infill the percentage ($\bar{I}_r = 70\%, 80\%, 90\%$, and 100%) corresponds to four distinct quality classes: superior parts with 70% infill, acceptable parts with 80% infill, and so on. Hence, $C = 4; c \in \{1, 2, 3, 4\}$. Using this notation, the general structure of the training data matrix is shown in the below equation.

091005-8 / Vol. 139, September 2017 Transactions of the ASME
5.3.1 Scenario I: Feature-Based Sampling. This approach of sampling the point cloud data is contingent on the shape of the part. In the present FFF case, nine areas are used for comparison (R = 9), which is depicted in Fig. 8. For instance, areas 1–4 are chosen to capture warping, primarily caused by the different rates of cooling at different areas of a 3D printed part; areas 5–8 and area 9 are related to the dimensional variations of the cuboid and cylindrical regions, respectively.

In selecting areas for characterizing surface morphology, it is posited that the duration for which the extruder dwells above a particular area is a useful guideline. This is because the longer the extruder spends over a particular spot, the longer the temperature at that spot is liable to remain above the glass transition temperature (T_g) of the polymer (T_g for ABS is close to 105 °C; the extruder temperature is in the range of 220–240 °C). Thus, the polymer does not solidify promptly and tends to reflow when the extruder lingers over an area. This leads to poor resolution of the features and affects both the surface quality and the dimensional variation of the part.

The drawback of scenario I is that expert knowledge to define sampling regions is necessary, and this knowledge may not be readily generalizable to every part design. Therefore, it is posited that forthcoming scenario II is more generally applicable as it mainly uses height (Z-axis) to define the sampling regions.

5.3.2 Scenario II: Plane-Based Sampling. In this scenario, the part is divided contingent on the Z-direction (layer deposition direction). This sampling method considers that the integrity of a layer in the AM process is dependent on the integrity of past layers, which are related to the effect of thermal stresses and the extruder dwell time as discussed previously in Sec. 5.3.1. Additionally, as the distance from the (heated) bed increases, the cooling rate of the top layers will be different from that of the bottom layer. Due to this effect, the dimensional variation will possibly vary along the vertical print direction (Z-axis). Sampling through this scenario will therefore account for most of the physical causes of deviations.

As depicted in Fig. 9, the four horizontal faces along the height of the part are the four sampling areas (R = 4) considered, resulting in a so-called plane-based sampling approach. The proportion of a sampling area selected from different planes is adjusted to be approximately similar to allow a better classification of certain features.

5.4 Assumptions and Parameter Settings. The aim of this section is to clarify the procedure for setting the three parameters that are important for application of the algorithm, namely, parameter R that represents the number of areas for applied sampling approach, k the sample size of point clouds used in each area, and θ the number of chosen eigenvalues. In Sec. 5.2 (Eq. (18)), the choice of θ was made using the relative deviation concept suggested by Zhan et al. [41].

The sample choice of size k is heuristic and should be set large enough to make its corresponding graph be a representative of its population. Furthermore, k is directly related to computational time. Increasing k results in longer processing time. In this work, it is recommended that k = 1000; it was determined based on offline studies that such a sample size is large enough to be representative of the whole point cloud data in a selection area.

However, since one sample should be taken from each area, the number of data points needed for testing is related to the chosen scenario. For instance, if scenario I is chosen as the sampling approach, the number of points needed to build the vector of eigenvalues (Y ∈ R^{m×k}) is equal to R × k = 9000 (out of approximately 2 × 10^6 data points in each part). The classification result of both sampling scenarios is presented in Sec. 5.5. Finally, a 60-30-10 training-testing-validation schema is used. The parameter R is contingent on the part design and is set as described in Sec. 5.3: R = 10 for scenario I and R = 4 for scenario II.

5.5 Procedure Used for Evaluation of the Classification Accuracy. Each of the sampling scenarios described in Secs. 5.3.1 and 5.3.2 are evaluated by comparing F-score extracted from a confusion matrix (see Table 6). The reason for choosing F-score (Eq. (20)) as the classification performance measurement is that it reflects both recall (complementary to type II statistical error, failing to detect the error) and precision (related to type I statistical error, false positive error) at the same time. The F-score ranges between 0% and 100% (higher is better); a higher F-score indicates higher classification recall and precision (Eq. (20)). It is important to note that F-score does not exist (NaN) where all elements of a column or a row in the confusion matrix are zero. In this work, the F-score is calculated from the total confusion matrix (Table 6), which in turn is aggregated by summing up the
confusion matrices from all replicates (in this study, a ten-fold replication procedure is used)

\[
F_{\text{Score}} = 2 \times \frac{\text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}} \quad (20)
\]

A limitation of this work is that only four classes are defined because the controlled experiments were restricted to four levels of infill percentage. In practice, there might be infinite classes as defined by the user; the higher the granularity in classes, the higher the requirement for test samples.

5.6 Application and Validation of the Approach. The aim of this section is to show that spectral graph theoretic Laplacian eigenvalues are effective as discriminating features for supervised machine learning-based classification of dimensional variation in AM parts. Accordingly, the results are juxtaposed with different popular classification approaches, namely \[32,42–44\]: (1) two global classifiers, namely, sparse representation-based classification (SRC) and neural network (NN); and (2) four round robin classifiers, namely, k-nearest neighbors (kNN), naïve Bayes (NB), support vector machine (SVM), and decision tree (Tree).

Table 5 reports the performance of these classifiers with different sampling scenarios and classifiers in different sample sizes \(k\). The numbers in parentheses are the standard deviations from replicating each algorithm ten times.

Table 6 Confusion matrices for classification of the FFF parts using conventional statistics in sparse representation-based classification

<table>
<thead>
<tr>
<th>Confusion matrix</th>
<th>Superior (I_f = 70%)</th>
<th>Acceptable (I_f = 80%)</th>
<th>Mediocre (I_f = 90%)</th>
<th>Reject (I_f = 100%)</th>
<th>Recall rate (%) (sensitivity)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Actual (%)</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Superior (I_f = 70%)</td>
<td>100.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>100.0</td>
</tr>
<tr>
<td>Acceptable (I_f = 80%)</td>
<td>100.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
</tr>
<tr>
<td>Mediocre (I_f = 90%)</td>
<td>100.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
</tr>
<tr>
<td>Reject (I_f = 100%)</td>
<td>100.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
</tr>
<tr>
<td>Precision (%)</td>
<td>25.0</td>
<td>NaN</td>
<td>NaN</td>
<td>NaN</td>
<td>(F = \text{NaN})</td>
</tr>
</tbody>
</table>

Scenario II Sampling

| Actual (%)       |                     |                     |                     |                     |                     |
| Superior \(I_f = 70\%\) | 39.9                | 2.1                 | 1.4                 | 2.5                 | 93.9                |
| Acceptable \(I_f = 80\%\) | 0.4                 | 98.5                | 0.0                 | 1.1                 | 98.5               |
| Mediocre \(I_f = 90\%\) | 1.3                 | 0.0                 | 96.0                | 2.7                 | 96.0               |
| Reject \(I_f = 100\%\) | 2.1                 | 2.1                 | 0.0                 | 95.7                | 95.7               |
| Precision (%)    | 96.1                | 95.8                | 98.5                | 93.8                | \(F = 96.04\)     |
size) from each area, and both the statistical features and Laplacian eigenvalues are estimated from the (identical) selected points in each area.

Juxtaposing the contents of Tables 6 and 7, it is evident that the sparse representation-based classification approach with graph theoretic invariants outperforms the situation where statistical features are used. Indeed, with the statistical features, the manner in which the data are sampled has an effect; the recall rate for sampling scenario I with statistical features is vanishingly small. Sparse representation is used as a benchmark as it outperforms rest of the classification approaches studied in this paper; similar inferior results \( (F\text{-score < 90\%}) \) were obtained with other methods. This section thus demonstrates that the spectral graph invariants are preferable over conventional statistical features. These results are in line with the authors’ recent findings that sparse representation is relatively computationally efficient with accuracy comparable to popular classifiers, such as SVM and NN [37].

6 Conclusions and Future Directions

The novel contribution of this work is in using spectral graph theoretic Laplacian eigenvalues as extracted features from laser-scanned 3D point cloud data followed by a supervised classification framework for classification of AM parts contingent on the severity of their dimensional variation. This contribution is practically important—instead of scanning an entire part, which can be time-consuming and inefficient, a few samples can be taken and the part quality can be quickly ascertained. Specific contributions are as follows:

1. To overcome the difficulty in differentiating the dimensional variation of AM parts from laser-scanned 3D point cloud data, a novel approach integrating the domains of spectral graph theory and machine learning was proposed. The approach first maps a selective sample of the scan data as an unweighted and undirected network graph. Two scenarios were tailored for sampling of data points, and different sample sizes were tested. Thereafter, the Laplacian eigenvalues of the network graph are used in a supervised classification framework for discriminating dimensional variation in AM parts. The classification accuracy in terms of \( F\text{-score} \) was compared with six different approaches, such as sparse representation, neural network, \( K\)-means, naïve Bayes, decision tree, and support vector machines.

2. It is demonstrated that the spectral graph theoretic features in a sparse representation-based classification algorithm are comparable (and better in most cases) in \( F\text{-score} \) among five other popular classifiers. In contrast to the other classifiers, sparse representation-based classification does not assume a priori distribution structure. The classification accuracy of the proposed approach exceeded 95% \( (F\text{-score}) \) and required about 10,000 sample points from a total of over \( 2 \times 10^9 \) coordinate points. This entails a substantial reduction in computation burden for classification from a large 3D point cloud dataset.

3. Apart from product quality assurance, the approach can lead to diagnosis of the manufacturing process in case of a fault. Traditionally, a two-class accept versus reject demarcation is popular in quality control, akin to a control chart. However, with this traditional approach, there is little scope to know which samples can be corrected rather than being discarded, and the degree and location of the correction to be applied. The proposed approach and sampling schemes described in Sec. 5.3 are together capable of providing this granular level of resolution.

The disadvantage of this approach is that it is a dictionary-based, supervised learning method, and hence requires a priori training data related to the part. Hence, the results from one part design cannot be translated to another. Furthermore, this work did not assess the part functional integrity in terms of the geometric dimensions and tolerance (GD&T); GD&T along with the current approach can lead to a deeper diagnosis of the process performance. The authors will attempt to carry out these GD&T studies in their forthcoming research.

Acknowledgment

One of the authors (ZIK) acknowledges the support of the National Science Foundation (NSF) for this work via grant CMMI 1436592.

Appendix: Application Example

In this example, we briefly demonstrate, step by step, the approach for classification of FFF printed parts with respect to their dimensional variation. The parameters and assumptions are set as explained in Sec. 5.4. This example uses the plane-based sampling scenario (Sec. 5.3.2).

We note that in this scenario, a random scan \( k = 1000 \) coordinates from each of the four locations at different \( z\)-height of the part is required \( (R = 4) \). The aim is to classify the dimensional variation of a part \( (\text{called } Y) \) into one of four categories: superior, accept, mediocre, and reject, given the point cloud data matrix \( \Psi_Y \in \mathbb{R}^{(k \times R \times 3)} \).
Design matrix \( \Psi_Y = \begin{bmatrix} \text{Deviation}_X & \text{Deviation}_Y & \text{Deviation}_Z \\ -0.2847 & -0.1541 & 0.0017 \\ -0.2572 & -0.1392 & 0.0015 \\ \vdots & \vdots & \vdots \\ -0.1425 & -0.1201 & 0.0570 \\ 0.1250 & -0.1007 & 0.0082 \end{bmatrix} \begin{bmatrix} 1000 \times 3 \text{ (first location)} \\ \vdots \end{bmatrix} \begin{bmatrix} 4000 \times 3 \end{bmatrix} \)

**Step 0**: Build the dictionary or design matrix \( A \in \mathbb{R}^{m \times (n \cdot C)} \) (Eq. (19)).

Prior to categorizing \( Y \), a dictionary or training data must be created. This knowledge is stored in the design matrix \( A \) (Eq. (19)), which is built from the experimental data (Sec. 3) using the procedure described in Eqs. (1)–(10), and (18). The dimensions of \( A \) are as follows:

- There are \( m \) rows in \( A \): \( m = R \times \theta \), where \( R \) is the number of areas sampled \( (R = 4 \text{ for scenario II sampling}) \), and \( \theta = 20 \) is the number of Laplacian eigenvalues (Eq. (18)). Hence, for scenario II, the number of rows \( m = 80 \). Likewise, for scenario I, \( m = R \times \theta = 10 \times 80 \).
- There are \((n \times C)\) columns in \( A \), where \( n \) is the number of samples taken, and \( C \) is the number of classes. We have \( n = 75 \) and \( C = 4 \). Hence \((n \times C) = 300\).

| Design matrix \( A^{m \times (n \cdot C)} = 80 \times 300 \) | Class 1 (superior) | Class 1 | Class 1 | Class 1 | Class 1 | … | Class 4 (reject) |
|---|---|---|---|---|---|---|
| Sample location \( R \) | Laplacian eigenvalue | Sample 1 | Sample 2 | Sample 3 | Sample 4 | Sample 5 | … | Sample \( n \) |
| Area 1 | \( \lambda_2 \) | 0.8514 | 0.8473 | 0.8809 | 0.8426 | 0.6743 | … |
| Area 1 | \( \lambda_3 \) | 0.8616 | 0.8691 | 0.8899 | 0.8534 | 0.6797 | … |
| Area 1 | \( \lambda_4 \) | 0.9370 | 0.9384 | 0.9505 | 0.9326 | 0.8483 | … |
| Area 1 | \( \lambda_5 \) | 0.9398 | 0.9405 | 0.9513 | 0.9367 | 0.8563 | … |
| Area 1 | \( \lambda_6 \) | 0.9591 | 0.9598 | 0.9546 | 0.9550 | 0.9044 | … |
| … | … | … | … | … | … | … | … |
| Area 4 | \( \lambda_k \) | … | … | … | … | … | … |

**Step 1**: Form the test vector \( Y \) (Eq. (14)) by extracting the Laplacian eigenvalues for the unknown part \( Y \) from its point cloud data \( \Psi_Y \).

From each of the four locations of the obtained point cloud data, \( \theta = 20 \), Laplacian eigenvalues are extracted (Eqs. (1)–(10), and (18)), to form the sample matrix: \( Y \in \mathbb{R}^{[R \times \theta] 	imes 1} \). The number of rows in \( Y \) has to be the same as the number of rows in \( A \)

\[
Y = \begin{bmatrix} 0.8912 \\ 0.8973 \\ \vdots \\ 0.8718 \\ 0.8898 \end{bmatrix} \begin{bmatrix} 20 \times 1 \text{ (first location)} \\ \vdots \end{bmatrix} \begin{bmatrix} 20 \times 1 \text{ (second location)} \\ \vdots \end{bmatrix} \begin{bmatrix} 80 \times 1 \end{bmatrix}
\]

**Step 2**: Obtain the vector of unknowns \( \hat{\omega} \)

The vector of unknowns \( \hat{\omega} \in \mathbb{R}^{1 \times (N)} \) is then calculated in such a way that it minimizes the total amount of estimation error \( \eta \) using LASSO regulation (Eq. (14))

\[
\hat{\omega} = \begin{bmatrix} 0 \\ 0 \\ 0.0723 \\ 0 \\ 0.3422 \\ \vdots \\ 0 \\ 0.0635 \\ \vdots \end{bmatrix} \begin{bmatrix} 75 \times 1 \text{ (class 1)} \\ \vdots \end{bmatrix} \begin{bmatrix} 75 \times 1 \text{ (class 2)} \\ \vdots \end{bmatrix} \begin{bmatrix} 1 \times 300 \end{bmatrix}
\]
Step 3: Classify the sample dimensional variation from $\hat{o}$ using an indicator function $\delta_k$.

Finally, Eq. (17) determines the class index ($c$) of the part $Y$. To obtain the class index, we define an indicator function $\delta_k(\hat{o}, \hat{c}) \in \mathbb{R}^{1/2}(N)$. For instance, for class 2 ($c = 2$), the indicator function in our example and the estimation error are as follows:

$$\delta_{c=2}(\hat{o}) = \begin{cases} 75 \times 1 \text{ (class 1)} \\ 0.0635 \end{cases}$$

Thus, the least amount of error belongs to the first class ($c_1$).

References


In this example, the second norm of the estimation error vectors ($\|\eta_i\|$) is calculated as

$$\|\eta_i\| = \sqrt{(2.1557, 6.3343, 5.7077, 6.1507)}.$$ 

This shows that the least amount of error belongs to the first (superior) class. Therefore, the part $Y$ is categorized in the superior class.


