Minimal Exhaustive Search Heuristics (MESH) of point clouds for form tolerances: the minimum zone roundness

Andrea Rossi, Michele Lanzetta

Department of Civil and Industrial Engineering
University of Pisa, Largo Lazzarino, 56122 Pisa, Italy
Email: lanzetta@unipi.it tel.: +39 050 2218122 fax: +39 050 2218065

Abstract

MESH is an ε-approximate algorithm to find the minimum zone center of a given roundness profile, with $\varepsilon=10^{-d}$, where d is the number of required decimal digits.

The proposed MESH algorithm is able to provide *only the accuracy that is necessary* to find the minimum zone error roundness (circularity). The basic principle is to exhaustively assess all MZR center candidates located at the cross points of a mesh, with spacing directly related to the target accuracy. Criteria for the selection of the required manufacturing (designer's) target accuracy (product specifications) are discussed. This result has been made possible by previous work on the limit search space to be searched. The algorithm effectiveness has been shown by computation experiments up to 16,384 cloud datapoints and by comparison with genetic algorithms and an exact method from the literature. The MESH algorithm can also serve for benchmarking purposes to assess the performance of other algorithms in terms of both accuracy and speed. The extension to other form tolerances of the exhaustive mesh based approach is discussed.

Keywords: geometrical tolerancing, circularity, minimum zone tolerance, ε–approximation, centroid neighborhood, computation time

1. Introduction

According to ISO [1] and ANSI [2], the minimum zone tolerance (MZT) method requires that the data sampled by a measuring tool on a machined surface is included within two Euclidean geometric features placed at the minimum distance. The minimum zone roundness (MZR) meets the ISO definition: it determines two concentric circles that contain the roundness profile and such that the difference in radii is the least possible value. The center of the two concentric circles is the minimum zone center C_{MZ} and their difference in radii is the MZR error E_{MZ}.

Coordinate Measuring Machines (CMM) are used to measure roundness errors by collecting an increasing number of datapoints from the profile of rotational parts [3]. CMMs may acquire thousands of datapoints in a circle.

The strategy to equiangular datapoints on the roundness profile is generally adopted in the literature. Conversely, alternative distributions of data are used to assess roundness deviations and number of undulations per revolution [4].

To process a large number of cloud datapoints, the least squares technique is efficient in computation and is widely used on most CMMs, however it does not meet the above mentioned standards, i.e. for roundness the minimum difference in radii of two concentric circles that contain the roundness profile. LSQ is efficient in computation and can be used with a large number of measured points, but in general the roundness error determined is larger than that obtained by MZR algorithms. Therefore, good parts can be rejected resulting in an economic loss. On the other hand, MZR algorithms require the solution of a non-linear problem; they are computationally intensive and sensitive to the number of cloud datapoints.

Two approaches to the MZR problem have been proposed in the literature: computational geometry techniques and nonlinear optimization (Figure 1).

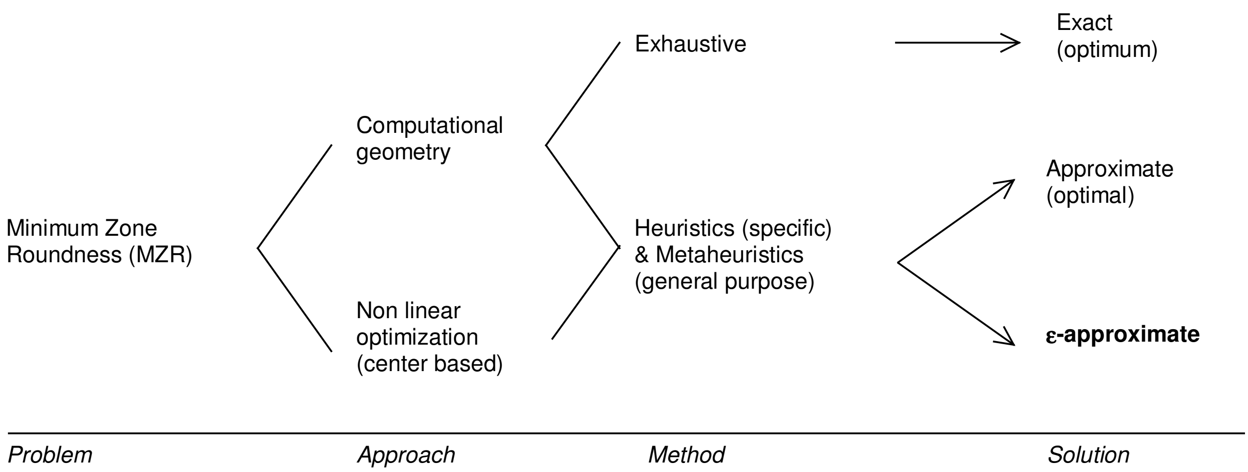

Figure 1: Classification of approaches to the minimum zone roundness (MZR) problem.

The first approach operates directly on the cloud datapoints and is, in general, very computationally intensive, especially when the dataset size is large because, with geometric methods, global optima are found by exhaustively checking every candidate. Samuel and Shunmugam [5] established a minimum zone limaçon based on computational geometry. Other examples of approaches are: the

Voronoi diagram [6] [7], the Chebyshev approximation [8], the simplex search / linear approximation [9] [10], and the steepest descent algorithm [11].

Wang *et al.* [12] and Jywe *et al.* [13] presented a generalized non-linear optimization procedure based on the developed necessary and sufficient conditions to evaluate the roundness error. In order to meet the standards, the minimum zone reference circles should pass through at least four cloud datapoints of the roundness profile. This can occur when two cloud datapoints lie *alternating* on each of the concentric circles. By satisfying this criterion, the problem is solved to the optimality. The computation time required to meet these conditions increases exponentially with the dataset size. The evaluation of the best among all the simple combinations of 4 elements from n is $\frac{n!}{4!(n-4)!}$. For example, for a dataset containing 4,000 cloud datapoints, approximately 1.1×10^{13} pairs of concentric circles should be evaluated.

Despite the computation time (proportionally to the dataset size), this method finds an exact solution (optimum) by an exhaustive approach. Gadelmawla [14] used a heuristic approach to drastically reduce the number of cloud datapoints used by the alternating criterion.

The second approach is based on the minimization of the E_{MZ} as a function of C_{MZ} (Figure 2) and considers primarily the location of the center of the concentric circle pairs (center-based). The inconvenience is that this function has several local minima making the exploration computationally intensive. Examples include metaheuristics like particle swarm optimization (PSO) [15] [16], ant systems [17], simulated annealing (SA) [18], immune evolutionary [19] and genetic algorithms (GAs) [20] [21] [22] [23] [24] [25]. The authors developed a cross-validation method to assess the kind of manufacturing signature [26] on the roundness profile in order to detect critical points such as peaks and valleys [27]. Optimization techniques such as genetic algorithms were developed to minimize the computation time in the roundness error evaluation. A fast genetic algorithm, with convergence speed greater than 0.1 μm per 30 generations as the selected stop condition, was developed and for large signature-based samples in [20] and for non-manufacturing signature specific in [25] with validation by certified software. [25] discussed the statistical distribution of the GA error as a function of the cloud datapoints size and of the search space size because by NPL Chebyshev best-fit circle certified software the minimum zone center is known (for example, it is fixed in (0,0)). Also, the deviations on the roundness profile are equally distributed in the range of the difference of radii between inner and outer minimum zone circles, so results are more general and not manufacturing signature specific. The (known) error of the GA obtained with the best configuration decreases down to $2 \cdot 10^{-5}$ for ∅ 40. In practical application of center based approaches, the minimum zone center is just the unknown of the MZR problem and requires a first estimation, like the centroid position. Non equiangular cloud datapoints do not satisfy the hypothesis in [31] because the centroid position is also affected by the cloud datapoint spacing and a correctly sized search space may not include the minimum zone center. Similarly, the centroid of a partial feature is far from the centroid of a complete feature. If the search region is too large and/or if the geometry of the profile is adverse [28], optimization search techniques, like genetic algorithms, tend to be trapped in local optimal solutions and not reach (converge to) the required accuracy. To avoid these problems the proposed method is compared with the genetic algorithm previously optimized by the authors on the MZR problem [20] with the condition on the search space in [31].

In industrial applications an optimal (not optimum) solution satisfying the design specifications (tolerance), i.e. not exact but acceptable, is usually preferred.

In order to reduce the computation time, through the years, a number of approximate approaches was developed for both computational geometry and nonlinear optimization, including heuristics and metaheuristics (Figure 1). The target accuracy can be met and the computation time can be lower, but one drawback of metaheuristics is that the computation time necessary to meet the target accuracy is undefined *a priori* (approximate solutions) and it may turn out that it is unpredictably unacceptable in practical applications, e.g. for larger dataset size.

With the proposed MESH method, the time required to meet the target accuracy or the target accuracy that can be met with the given time can both be known in advance (ε-approximate algorithm).

2. Search space for the minimum zone center

The proposed heuristic exploits an approximate region for the location of the minimum zone center C_{MZ} by exhaustively searching the center candidates on a mesh, rather than the best quartet of cloud datapoints. It still produces an ε-approximate solution matching the requested tolerance, but with known computation time.

The idea of an exhaustive method based on a mesh located in the search space was presented in [29]. Lei *et al.* proposed a mesh in polar coordinates centered at the least-square center, obtained as intersection of m concentric circles and n radius lines where the center of the mesh is the least square center. They considered a mesh size of $\frac{LSE}{2}$, where LSE is the least-square error. However, they did not provide appropriate experimentation on the fact that this mesh centered at the least-square center includes the minimum zone center.

If, on one hand, the inclusion of C_{MZ} into the mesh is mandatory, on the other hand, the mesh must be dimensioned to prevent exploration of needless areas of the search space. The mesh has two critical parameters to meet the accuracy requirements and be minimal: the mesh position and size; however either the mesh size is about the radius of the roundness profile or there is no guarantee that the minimum zone center is included. Based on authors' previous papers [30] [31], the focus of this work is to give the minimum mesh size that covers the search space size, which offers the guarantee of including the minimum zone center.

Only few contributions are currently available in the literature regarding the setting of the search space of the nonlinear optimization problem. The centroid is usually considered as the center of the search space. In [21] the search space is a square of fixed 0.2 mm side, in [20] it is 5% of the circle diameter [32], the side is determined by the distance of the farthest point and the nearest point from the mean center which is approximated to 2 $E_{C\,n}$, where $E_{C\,n}$ is the roundness error related to the centroid of n equiangular datapoints, defined in equation (6) below. In [33] it is the rectangle circumscribed to the cloud datapoints.
Authors' previous works provide closed form upper bounds of the distance between centroid C_n and minimum zone center C_{MZ}. A first theoretical analysis that provides a closed form expression to minimize the search for a GA with genetic parameters optimized for MZR criterion [20] is available in [31], where the center of the search space S is the centroid and the radius is the upper bound extrapolated asymptotically from the distance between centroid and minimum zone center. This general result is not sampling and form deviation specific. The search space radius of about 0.43 E_C was significantly reduced with respect to those available in the literature.
More recently, the authors provided a new closed form expression of the search space radius for the roundness problem with the MZT criterion, further decreased by 27% [31]. This upper bound was based on a worst case geometrical feature formed by two concentric-opposite arcs. The search space size of π^{-1} E_C is the current lowest upper bound. The computation time with current PCs is negligible for datasets up to 100 equiangular datapoints to achieve estimations of the circularity error better than E_C. By reducing the search space, the computation time decreases but there is still a guarantee that the minimum zone center C_{MZ} is included in the search space. This evaluation can be used outright as a first estimation of the minimum zone center position or as the starting point for a local search, e.g. a search neighborhood of metaheuristics, such as genetic algorithms, particle swarm optimization, taboo search etc. By reducing the search area, the algorithm complexity and the computation time can be reduced. Among possible approaches, current work proposes an *exhaustive* search within the proposed search space, taking advantage of current processors.

The proposed Minimal Exhaustive Search Heuristic (MESH) finds a (practical) solution to the minimum zone roundness problem for any dataset size achieving the accuracy required by the (designer's) specifications. It will be shown that MESH achieves an arbitrary accuracy expressed as a fraction of the roundness error related to the centroid E_C: $10^{-d}\ E_C$, where d is the number of decimal digits required. According to the golden rule of metrology (or ten-to-one gagemaker's rule), d should be such that the uncertainty of the result should not exceed 1/10 of the tolerance. There is an ongoing discussion (e.g. ASME B89.7.3.1) about shifting this ratio to 1/4; in the remainder, for clarity, the 1/10 ratio is conservatively considered.

3. Minimum Zone Roundness (MZR) problem formulation

Figure 2 shows two pairs of concentric circles that include the cloud datapoints centered respectively at C_{MZ} and C and where E_{MZ} and E_C are their differences in radii.

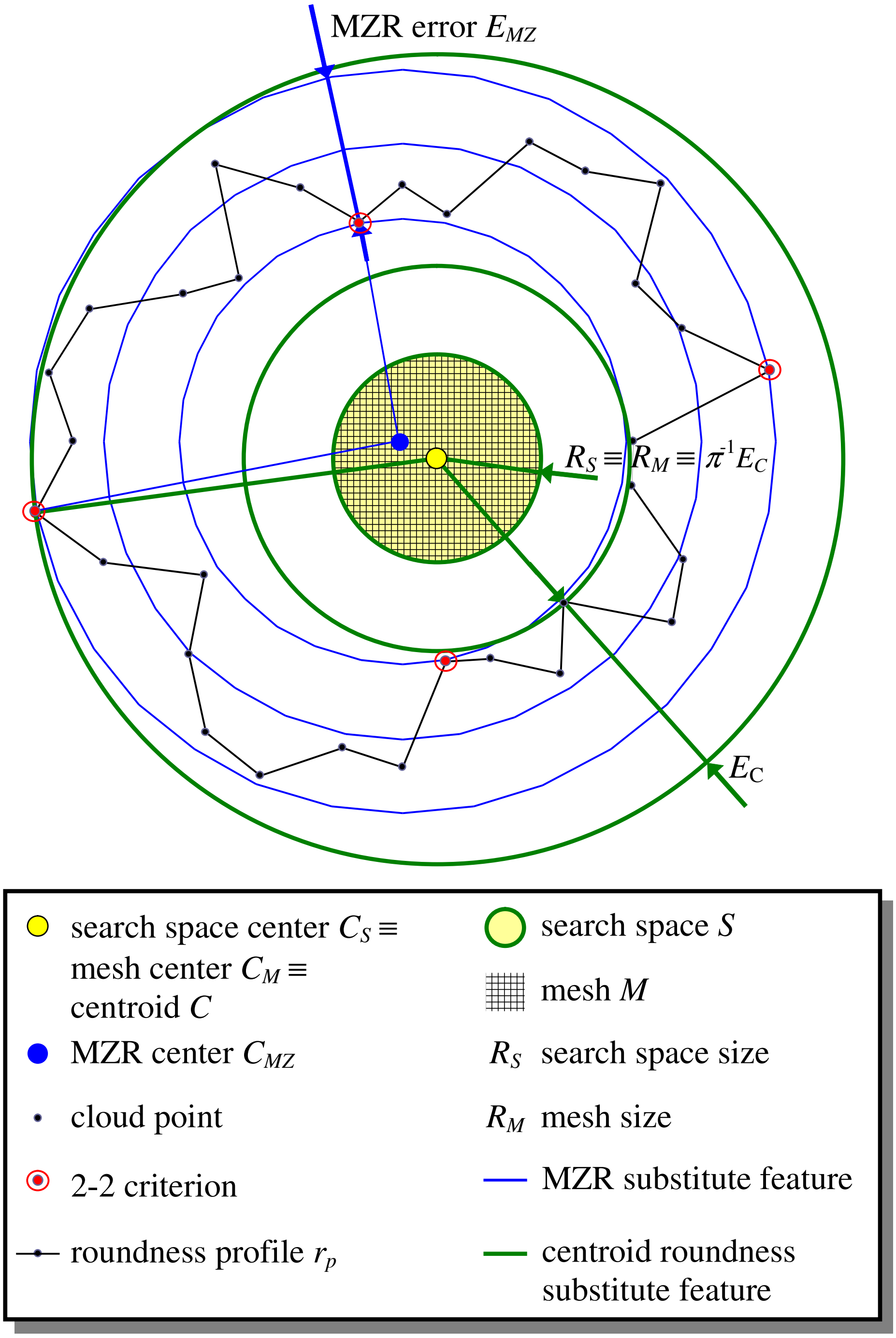

Figure 2: C_{MZ} and C are possible locations of the centers of the two concentric circles (roundness substitute features) containing all the cloud points and their differences in radii are the roundness errors E_{MZ} and E_C, scaled for clarity. With the minimum zone center, C_{MZ} the difference in radii of the MZR substitute feature is minimal and it equals to the minimum zone error E_{MZ}. In the example, the requirement of the alternation theorem [12] [13], represented by the circled red cloud datapoints, is satisfied.

The MZR is the solution of the following optimization problem [20]:

$$\min_{(x,y)\in S}\left[\max_{\theta_i=i\times\frac{2\pi}{n},\, i=1,\ldots,n} r_p(x,y,\theta_i)-\min_{\theta_i=i\times\frac{2\pi}{n},\, i=1,\ldots,n} r_p(x,y,\theta_i)\right] \tag{1}$$

where $S\subset \Re^2$ is the search space, $\theta_i = i\times\dfrac{2\pi}{n}$, i=1,...,n are the angular locations of n equiangular datapoints of the roundness profile $r_p(x,y,\theta_i)$ of the reference circle of center (x,y).

The solution of problem (1) is the minimum zone error defined as:

$$E_{MZ} = \max_{\theta_i=i\times\frac{2\pi}{n},\, i=1,\ldots,n} r_p(C_{MZ},\theta_i)-\min_{\theta_i=i\times\frac{2\pi}{n},\, i=1,\ldots,n} r_p(C_{MZ},\theta_i) \tag{2}$$

where $C_{MZ} = (x_{MZ}, y_{MZ})$ is the minimum zone center.

The search space S for MZR is characterized by its center, $C_S\in\Re\times\Re$, and its size, $R_S\in\Re$, which represents the maximum distance between C_S and the minimum zone center C_{MZ}. The current lowest upper bound of the search space size R_S for MZR according to [31] was based on the maximum distance between centroid C and minimum zone center C_{MZ}

$$C_S \equiv C \tag{3}$$

This result was extended to the discrete case with C_n for C, considering the hypothesis of equiangular datapoints (x_i, y_i) instead of a continuous profile, and

$$C_S \equiv C_n \equiv \left(\frac{1}{n}\sum_{i=1}^{n} x_i\,,\ \frac{1}{n}\sum_{i=1}^{n} y_i\right) \tag{4}$$

The maximum distance R_S was evaluated in [31] by a worst case geometrical feature formed by two concentric opposite semicircles of different radius. The distance R_S between centroid C_n and minimum zone center C_{MZ}, i.e. the current lowest upper bound of the search space S, is evaluated in closed form by the expression:

$$R_S = \pi^{-1} E_{C\,n} \tag{5}$$

where $E_{C\,n}$ with n cloud datapoints is

$$E_{C\,n} = \max_{\theta_i = i\times\frac{2\pi}{n},\, i=1,..,n} r_p(C_n,\ \theta_i) - \min_{\theta_i = i\times\frac{2\pi}{n},\, i=1,..,n} r_p(C_n,\ \theta_i) \tag{6}$$

4. The proposed method: Minimal Exhaustive Search Heuristic (MESH)

MESH is based on an exhaustive search on all the cross points of a two-dimensional mesh M included in the search space S shown in Figure 2.

In MESH, each cross point $C_{MZ}^* \equiv (x,y)\in M$ is considered as the center candidate for C_{MZ} of the roundness profile of the n equiangular datapoints; the related roundness error is given by the objective function evaluated for minimum

$$E_{MZ}^* = \min_{(x,y)\in M}\left[\max_{\theta_i = i\times\frac{2\pi}{n},\, i=1,..,n} r_p(x, y, \theta_i) - \min_{\theta_i = i\times\frac{2\pi}{n},\, i=1,..,n} r_p(x, y, \theta_i)\right] \tag{7}$$

4.1. Mesh parameters

Equation (7) is obtained from (1) where the two-dimensional search space S is discretized on the two-dimensional array M with the following mesh parameters and symbols:

C_M mesh centre

R_M mesh size

ε_M mesh spacing

N^2 number of cross points

4.2. The mesh covers the search space

The mesh M is centered at the search space center C_S

$$C_M \equiv C_S \tag{8}$$

with its same size

$$R_M = R_S$$

(9)

4.3. The mesh parameters after current lowest upper bound

Based on the current lowest upper bound for the search space according to [31], and without loss of generality, the mesh M is centered at the centroid C_n of the n equiangular datapoints in equation (4):

$$C_M \equiv C_n \tag{10}$$

The mesh size, R_M, is evaluated considering that the distance between centroid and minimum zone center is lower than (or equal to) the upper bound R_S

$$R_M = \pi^{-1} E_{C\,n} \tag{11}$$

according to [31], where $E_{C\,n}$ is the roundness error considering the centroid as the center of the roundness profile of the n equiangular datapoints in equation (6).

4.4. The mesh error

Assuming an equally spaced mesh, with a square grid with mesh spacing ε (a first estimation of the final ε_M), determines the following errors for the minimum zone center C_{MZ}* and roundness error E_{MZ}* evaluated by the algorithm with respect to their optima C_{MZ} and E_{MZ} (Figure 3):

$$|C_{MZ}^* - C_{MZ}| \leq \frac{\varepsilon}{\sqrt{2}} \tag{12}$$

$$E_{MZ}^* - E_{MZ} \leq \sqrt{2}\ \varepsilon \tag{13}$$

In equation (12), $\frac{\varepsilon}{\sqrt{2}}$ corresponds to the maximum distance between a cross point (C_{MZ}*) and the farthest point of the mesh square of side ε, i.e. with C_{MZ} located at the midpoint of the mesh square diagonal.

In equation (13), $\sqrt{2}\ \varepsilon$ considers the previous case, with C_{MZ}* coinciding with a cross point and the closest and the farthest cloud points positioned respectively on the two opposite half lines connecting C_{MZ}* and C_{MZ}, along a diagonal of the mesh square.

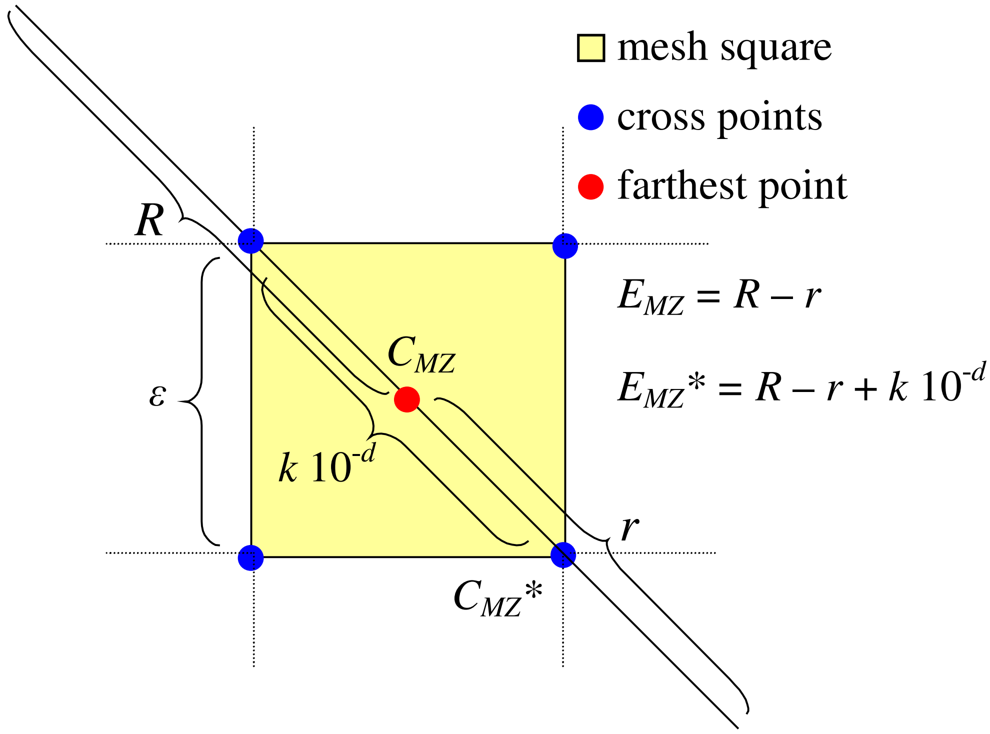

Figure 3: MESH algorithm accuracy $k\ 10^{-d}$ as a function of the relative position between cross points (or minimum zone roundness center candidates C_{MZ}*) and the farthest possible position of the actual minimum zone roundness center C_{MZ}.

Also,

$$k\ 10^{-d} = \sqrt{2}\ \varepsilon, \qquad k \in \Re,\ 1 \le k < 10 \tag{14}$$

where $k\ 10^{-d}$ is the *target* algorithm accuracy.

From equations (12) and (13) and from Figure 3, it can be noticed that, in the worst case represented, the maximum approximation error is double for E_{MZ} with respect to C_{MZ}.

4.5. The MESH algorithm accuracy

The MESH algorithm output are C_{MZ}* and E_{MZ}*; the symbol * expresses that they are an estimation of C_{MZ} and E_{MZ}. The approximation (an overestimate, necessarily) on E_{MZ} is defined by $k\ 10^{-d}$. d represents the order of magnitude or the number of significant digits of the roundness error E_{MZ} and it depends on the design specifications. As an example, if units are in mm, $d = 3$ means a maximum acceptable error of 1 µm and the significant digits from the mesh algorithm are *d.ddd*. The acceptable roundness E_{MZ}^{limit} should be expressed with $d - 1 = 2$ digits (or fewer); e.g. if the maximum acceptable roundness is $E_{MZ}^{limit} = 0.07$ ($d = 2$), parts with roundness error E_{MZ} below 0.070 ($d = 3$) are acceptable (e.g. 0.0696, $d = 4$) and above 0.070 (e.g. 0.0703, $d = 4$) are not.

In order to achieve a better algorithm accuracy, the mesh spacing ε should be reduced proportionally. The mesh spacing ε determines the algorithm accuracy $k\,10^{-d}$ and the approximation (overestimate) on E_{MZ} according to equation (13)

$$E_{MZ}^* - E_{MZ} \leq k\,10^{-d} \tag{15}$$

In the remainder, without loss of generality,

$$k = 1 \tag{16}$$

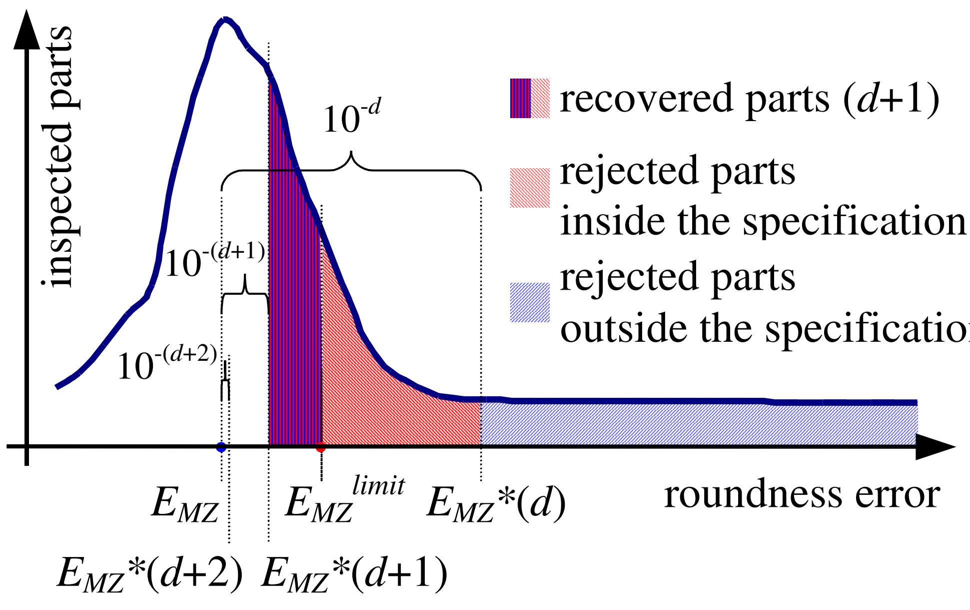

Figure 4: Effect of the mesh spacing and algorithm accuracy on the number of rejected parts

Figure 4 shows qualitatively the effect of the MZR algorithm accuracy 10^{-d} on the number of rejected parts. By increasing the accuracy of the algorithm with d, with

$$E_{MZ} \leq E_{MZ}^{limit} \mid E_{MZ}^* > E_{MZ}^{limit} \tag{17}$$

more false parts outside the specification can be identified and fewer parts inside the specification can be rejected. The case of d+1 is shown in Figure 4 and similarly for d+2 etc. If the distribution of inspected parts is known, from the statistical process control, the number of parts rejected as a function of d and E_{MZ}^{limit} is known quantitatively.

The *actual* algorithm accuracy is defined by A^*, as opposed to the target algorithm accuracy 10^{-d}, as

$$A^* = (E_{MZ}^* - E_{MZ}) / 10^{-d} \qquad A^* \in \Re, 0 \leq A^* \leq 1 \tag{18}$$

A^* can be expressed as a percentage of the algorithm accuracy fixed by (15)

For exhaustive methods [12] [13],

$$d \rightarrow \infty, A^* = 0 \tag{19}$$

Also, with computational approaches, the algorithm accuracy (10^{-d}) improves with the cloud size n

$$d \propto n \tag{20}$$

To prevent the rejection of parts inside the specification because of algorithm error

$$E_{MZ}^* \leq E_{MZ}^{limit} \tag{21}$$

consequently, from equation (15),

$$10^{-d} \leq E_{MZ}^{limit} - E_{MZ} \tag{22}$$

The first attempt of mesh spacing ε is $10^{-d} / \sqrt{2}$ from (14).

A smaller mesh spacing, ε, would lead to an unjustified increase in computation time, by increasing the number of cross points N^2. For a *square* mesh M circumscribed to the circular search space S, of size R_M, the number of cross points $N_{circ} \in \aleph$ is given by rounding up the terms in

$$N_{circ}^2 = (\lceil 2\, R_M / \varepsilon \rceil + 1)^2 = (\lceil 2\ \sqrt{2}\ \ 10^d\, R_M \rceil + 1)^2 \tag{23}$$

N_{circ}^2 reduces by $\pi/4$ excluding the cross points outside the circular search space S. From equation (11)

$$N^2 = \pi/4\ N_{circ}^2 = \pi/4\ (\lceil 2\sqrt{2}\ \ 10^d\, R_M \rceil + 1)^2 = \pi/4\ (\lceil 2\sqrt{2}\ \ 10^d\, E_{Cn} / \pi \rceil + 1)^2 \tag{24}$$

An approximate numerical expression is not provided because it is prone to approximation errors with low N.

In the numerical example above, assuming $E_{C\,n} = E_{MZ}^{limit} = 0.07$, the number of cross points for a circular mesh at different target accuracy N^2 (d=2) ≅ 31, N^2 (d=3) ≅ 3,119, N^2 (d=4) ≅ 311,944.

The first attempt mesh spacing ε will be conservatively reduced in order to accommodate a constant spacing ε_M between cross points

$$\varepsilon_M = 2\, R_M / (N_{circ} - 1) = 2\, E_{Cn} / (\pi\, (N_{circ} - 1)) \tag{25}$$

In the numerical example, ε_M (d=2) ≅ 0.006, ε_M (d=3) ≅ 0.0007, ε_M (d=4) ≅ 0.00007.

The number of operations (and computation time) of the MESH algorithm #MESH is proportional to

$$\#\mathrm{MESH} \approx \mathrm{O}(n\, N^2) \tag{26}$$

If the algorithm stops as soon as C_{MZ}^* is found, on average, half of #MESH operations will be necessary.

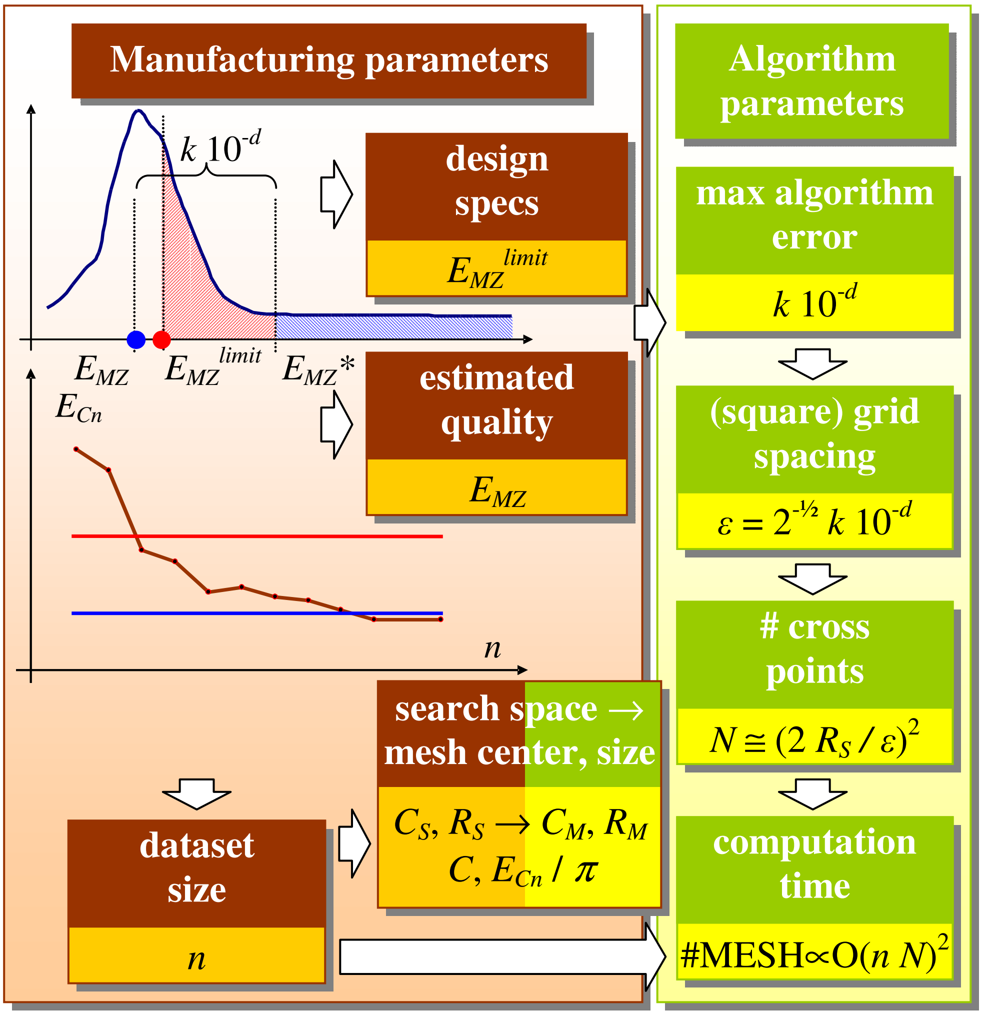

Figure 5: Setting the MESH algorithm parameters.

The implemented MESH algorithm parameters are set as summarized in Figure 5.

5. Extension to other form tolerances

The proposed MESH approach can be easily extended to other form tolerances both in 2D and in 3D, by the following steps

- definition of the search space (location and size), which should include the mesh;
- definition of the mesh parameters, depending on the approximation required for the form tolerance.

Given a cloud of points, the centroid used in the roundness case is substituted by likeness with the principal axis for other form tolerances like straightness, cylindricity, conicity, flatness etc.

For example, as shown in Figure 6, to find the minimum zone straight line, two meshes perpendicular to the principal axis should be defined. Each straight line passing through each pair of cross points belonging to both meshes should be exhaustively checked in order to find the minimum zone straight line. For straightness in 2D, the two meshes degenerate on a single mesh and straight line candidates pass through cross point pairs taken from that mesh.

Roundness and sphericity are characterized by point symmetry, consequently the principal axis degenerates to a single point. The feature centers are searched for respectively in a 2D and a 3D mesh (Figure 7).

In all cases, the mesh should be dimensioned in order to coincide with the search space (9). The search space shape should be defined for each form tolerance and it should have the minimum size because it affects directly the MESH algorithm performance, because every single candidate should be checked.

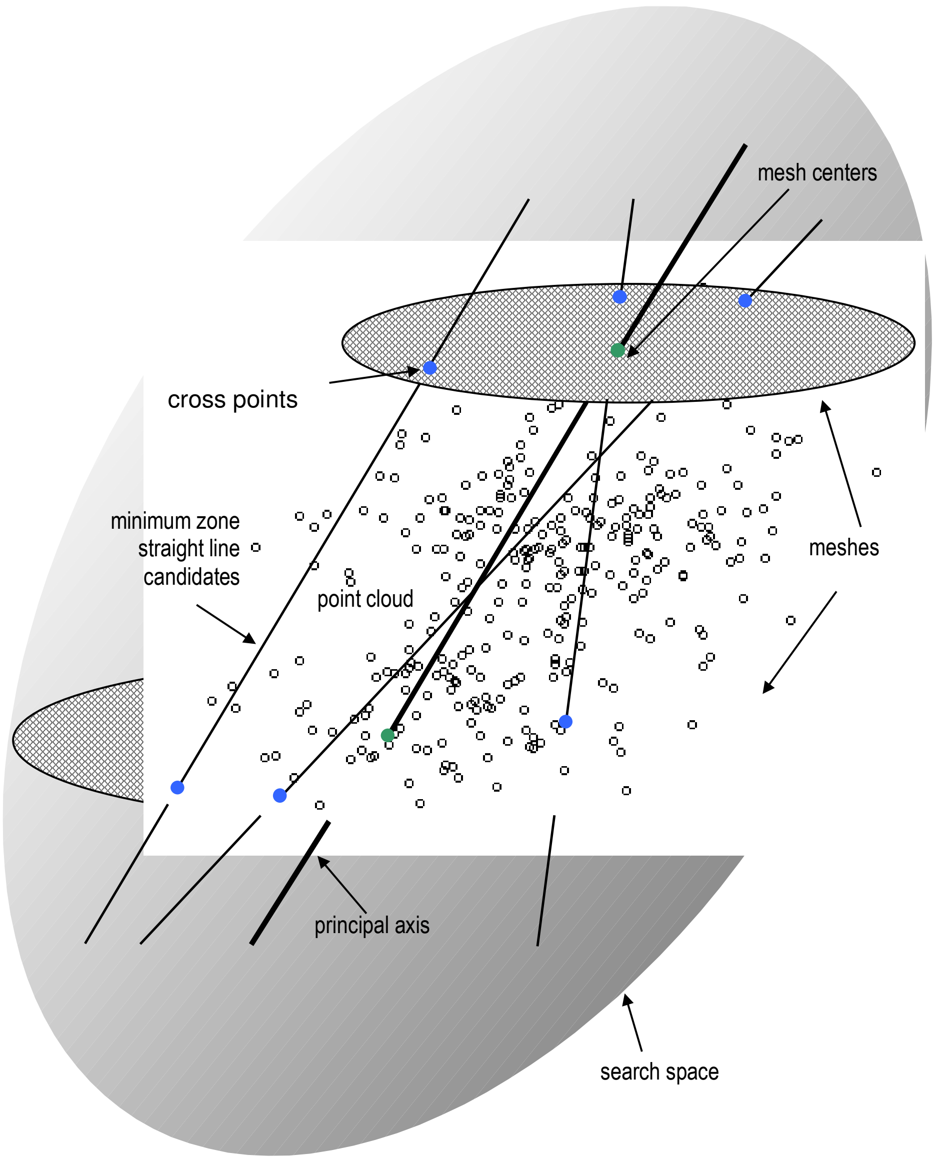

Figure 6: Example extension of the MESH algorithm to other form tolerances with axial symmetry.

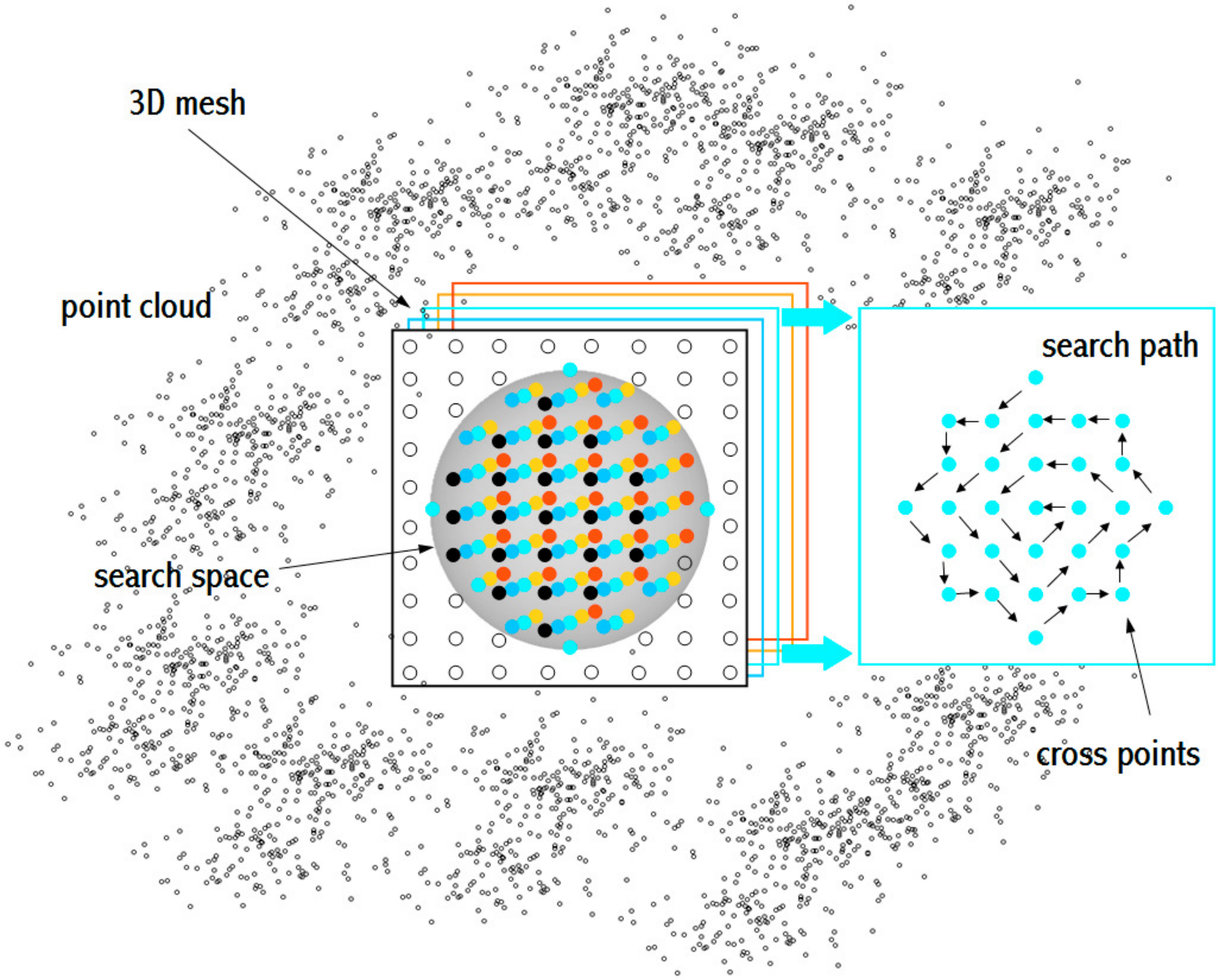

Figure 7: Example extension of the MESH algorithm for sphericity evaluation.

6. Results and discussion

Table 1 considers 12 datasets containing 8 to 16,384 equiangular datapoints generated with the software described in [34] with known $C_{MZ} \equiv (0,0)$, mean radius = 20 and $E_{MZ} = 0.06$.

For each dataset size n, E_{C_n} is calculated by equation (4) and the search space for the minimum zone center is searched within a circular area of radius $R_M = \pi^{-1}\ E_{C_n}$ by the proposed MESH algorithm according to equation (11). Three different orders of magnitude d for the accuracy in the estimation of the E_{MZ} corresponding to the mesh spacing ε_M are considered. To help the discussion, three maximum acceptable roundness values E_{MZ}^{limit} at different accuracy levels d are also specified. The number of necessary mesh cross points N^2 for the three accuracy levels is determined from equation (24).

Table 1: Datasets containing equiangular datapoints as in [34] with known $C_{MZ} \equiv (0,0)$, mean radius = 20 and $E_{MZ} = 0.06$. The results of the proposed MESH algorithm are identified by *.

Dataset size *n*	**8**	**16**	**32**	**64**	**128**	**256**	**512**	**1,024**	**2,048**	**4,096**	**8,192**	**16,384**
E_{C_n} (4)	0.0676	0.0667	0.0632	0.0627	0.0614	0.0616	0.0612	0.061	0.0606	0.0602	0.0603	0.0602
$R_M = \pi^{-1} E_{C_n}$ (11)	0.0215	0.0212	0.0201	0.0200	0.0195	0.0196	0.0195	0.0194	0.0193	0.0192	0.0192	0.0192
$E_{MZ}^{limit} = 0.07$ $N^2 = 31$ $d = 2$												
E_{MZ}*	0.062	0.060	0.063	0.063	0.061	0.062	0.061	0.061	0.061	0.060	0.060	0.060
time [s]	0.00	0.00	0.00	0.00	0.00	0.00	0.01	0.01	0.02	0.06	0.14	0.22
x_{MZ}*	0.0000	0.0001	0.0007	0.0020	0.0011	0.0008	0.0004	0.0007	0.0002	0.0001	0.0000	-0.0001
y_{MZ}*	0.0012	0.0002	0.0030	0.0006	0.0004	0.0004	0.0007	0.0000	0.0003	-0.0001	-0.0002	0.0000
mean radius*	20.0000	20.0001	20.0015	19.9994	19.9996	19.9999	20.0001	19.9998	20.0000	20.0000	20.0000	20.0000
ε_M (25)	0.0054	0.0053	0.0050	0.0050	0.0049	0.0049	0.0049	0.0049	0.0048	0.0048	0.0048	0.0048
A* (18)	17%	3%	32%	27%	14%	16%	12%	10%	6%	2%	3%	2%
$E_{MZ}^{limit} = 0.061$ $N^2 = 3{,}119$ $d = 3$												
E_{MZ}*	0.0602	0.0603	0.0602	0.0602	0.0604	0.0604	0.0603	0.0601	0.0606	0.0602	0.0603	0.0602
time [s]	0.00	0.01	0.02	0.05	0.10	0.18	0.38	0.74	1.52	3.06	7.51	11.76
x_{MZ}*	0.00000	-0.00010	-0.00010	-0.00010	0.00010	0.00020	0.00020	0.00010	0.00020	0.00010	0.00001	-0.00010
y_{MZ}*	0.00010	0.00020	-0.00020	-0.00010	0.00020	-0.00020	-0.00010	0.00000	0.00030	-0.00010	-0.00017	0.00000
mean radius*	20.00000	20.00010	20.00000	20.00000	20.00000	20.00000	20.00000	20.00000	20.00000	20.00000	20.00000	20.00000
ε_M (25)	0.00067	0.00066	0.00063	0.00062	0.00061	0.00061	0.00061	0.00061	0.00060	0.00060	0.00060	0.00060
A* (18)	20%	30%	20%	20%	40%	40%	30%	10%	60%	20%	31%	20%
$E_{MZ}^{limit} = 0.0601$ $N^2 = 311{,}943$ $d = 4$												

E_{MZ}*	0.06004	0.06001	0.06004	0.06002	0.06003	0.06002	0.06002	0.06002	0.06000	0.06004	0.06003	0.06003
time [s]	0.06	1.12	2.23	4.48	8.96	17.68	35.31	70.46	140.91	281.25	732.94	1131.46
x_{MZ}*	0.000028	-0.000002	-0.000018	0.000009	0.000022	0.000004	-0.000016	0.000002	0.000000	-0.000020	0.0000008	0.000018
y_{MZ}*	0.000012	-0.000006	0.000026	-0.000008	-0.000004	-0.000011	-0.000008	-0.000021	-0.000003	-0.000014	0.000014	0.000003
mean radius*	19.999987	20.000000	20.000008	20.000001	19.999992	20.000004	19.999996	20.000010	20.000001	20.000000	20.000000	19.999996
ε_M (25)	0.000067	0.000067	0.000063	0.000063	0.000061	0.000061	0.000061	0.000061	0.000061	0.000060	0.000060	0.000060
A* (18)	39%	9%	42%	17%	27%	15%	24%	23%	4%	42%	30%	28%

For completeness, also the center coordinates (x_{MZ}^*,y_{MZ}^*) and mean radiuses* are reported.

The processing time to evaluate E_{MZ}* by the proposed MESH algorithm with 3.1 GHz i-3 Pentium 3.5GB RAM is also listed.

As predicted by (26), the processing time is linear with the cloud size and is in the order of 10 s with current processor up to $d = 3$ and for $d = 4$ up to $n = 256$.

The processing time to evaluate E_{C_n} is negligible for all dataset sizes and has not been included.

The algorithm accuracy A* by equation (18) is as low as 2% and the maximum is 60%. A* conservatively does not exploit all the available range up to 100% because the mesh spacing reduces from ε to ε_M by equation (25) to design a square mesh with N_{circ} cross points on each side.

It can be noticed that the target accuracy in the estimation of E_{MZ} (on the d^{th} digit) is achieved for all dataset sizes as planned by the mesh spacing ε selected with equation (14), starting from 8 cloud data.

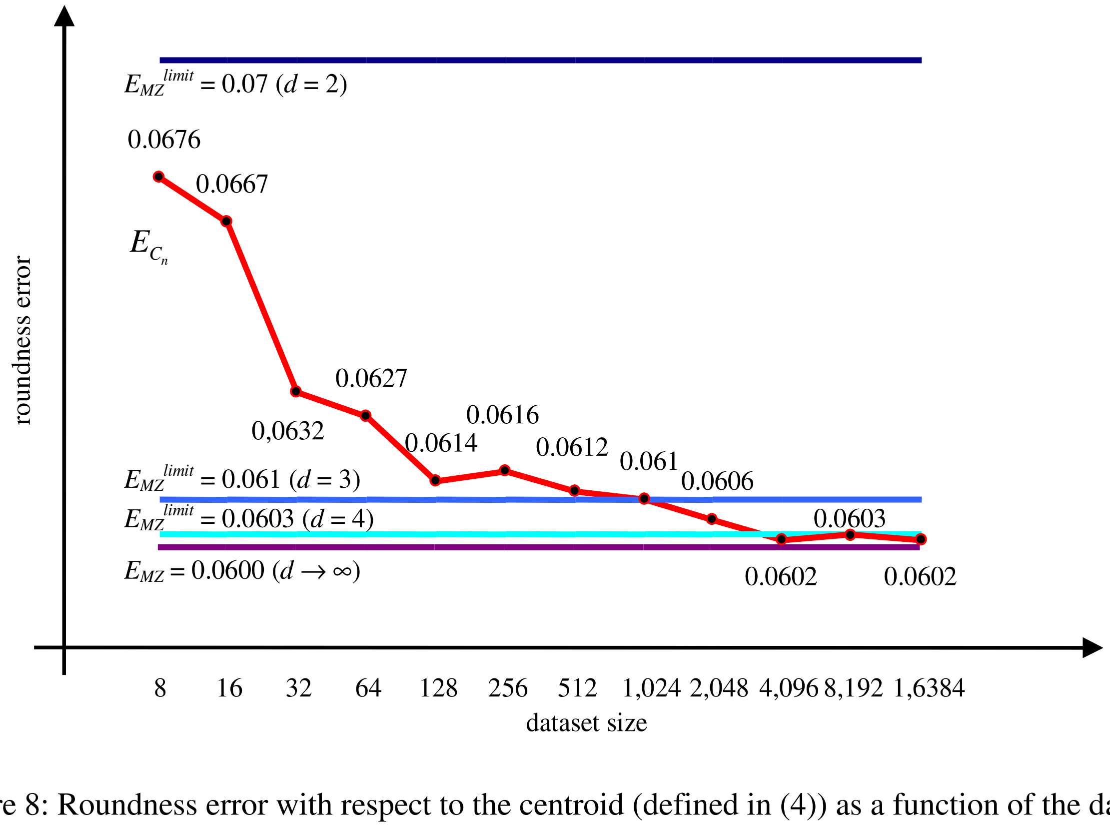

Figure 8: Roundness error with respect to the centroid (defined in (4)) as a function of the dataset size (generated as in [34]) for three maximum acceptable roundness values E_{MZ}^{limit} at different accuracy levels d. By construction of the MESH algorithm $E_{MZ}^* \leq E_{MZ}^{limit}$. Actual E_{MZ}^* values are listed in Table 1.

Figure 8 elucidates the relationship between E_{C_n} and E_{MZ}^*; this latter is always lower than E_{MZ}^{limit} by the MESH algorithm.

A sharp reduction of E_{C_n} (and consequently of the search space (5)) as a function of n is also shown in Figure 8 in the range 8 – 128.

A minimum dataset size is required for E_{C_n} to achieve the target roundness E_{MZ}^{limit}, corresponding respectively to 8, 1,024 and 4,096 cloud datapoints for the different orders of required accuracy d = 2, 3 and 4, as opposed to the MESH algorithm that matches the target roundness E_{MZ}^{limit} for all cloud sizes n.

Increasing the dataset size n reduces E_{C_n} and the search space is reduced proportionally according to (5). Of course, by increasing the dataset size n, the sampling time increases. A tradeoff should be met in order to minimize the total time spent for cloud sampling and computation time by the MZR algorithm.

In summary, it has been shown that, although E_C is always greater than E_{MZ} (or equal), with larger dataset size, E_C can be an adequate replacement.

Using E_{MZ} versus E_C has the benefit of reducing the number of rejected parts with roundness outside of the specification tolerance.

From a practical viewpoint, directly calculating E_C (closed form) with a larger cloud or using a more accurate MZR algorithm to calculate E_{MZ} (higher computation time) with a smaller cloud depends on a case by case basis on the ratio between sampling time by the CMM and computation time $O(n\ N^2)$ (26); this latter is a function of the cloud size n and of the mesh spacing (23). As technology advances, on one hand users can benefit of faster data sampling by continuous probe scanning and high resolution optical scanning, while on the other they can benefit of faster processors.

6.1. Benchmarking with literature

In this section, it will be shown that the MESH algorithm is incumbent with respect to the GAs in [25] and in [20].The search space of the minimum zone center by the GA in [25] is restricted to an arbitrary (large) search space centered on the centroid. The centroid in turn is farther from the minimum zone center (about 50 times with respect to MESH), because the cloud datapoints generated by the NPL certified software are not equiangular. Table 2 shows the better performance of the MESH algorithm in term of computation time and E_{MZ}^*, as a result of the lower search space radius, which produces a density of center candidates hundreds of times higher despite the lower MESH cross points number.Table 3 compares the minimum zone roundness error E_{MZ}* of the proposed MESH algorithm with results obtained by a new implementation of the genetic algorithm previously optimized by the authors on the MZR problem [20] with the condition on the search space in [31] on the same datasets of Table 1 containing equiangular datapoints generated as in [34], with known $C_{MZ} \equiv (0,0)$, mean radius = 20 and $E_{MZ} = 0.06$.

Table 2: Comparison between the proposed exhaustive MESH approach and the best result of a metaheuristics from the literature, with cloud datapoints parameters ∅ 40, $E_{MZ} = 0.06$ and target accuracy $d = 2$. The C_{MZ} candidates density is given by C_{MZ} candidates/search space.

Algorithm	Cloud size n	Search space size R_M	C_{MZ} candidates	C_{MZ} candidates density	E_{MZ}^*	Pentium® CPU	Time [s]
MESH	4,096	0.0192	31 (cross points)	26,767	0.06	3.1 GHz i3	0.06
	256	0.0196		25,686	0.062		0.00
GA [25]	10,000	1	70 (chromosome population)	70	0.0617	1.2 GHz M	4.48 (n 500)

Table 3: Comparison between the exhaustive MESH approach and a metaheuristics on datasets of Table 1. The results of the two algorithms are identified by *. For E_{MZ}^* the significant digit is **enhanced**; red if the tolerance E_{MZ}^{limit} is not met.

	Cloud size	**MESH parameters**			E_{MZ}^{limit}	E_{MZ}^*		**mean radius***		**Algorithm complexity**		
#	***n***	***d***	ε_M	N_{circ}		**MESH**	**GA [20] [31]**	**MESH**	**GA [20] [31]**	**GA time [s]**	**#MESH (26) ≈ #GA (28)**	**#EXACT (30)**
1.	32	2	0.0050	8	0.0**7**	0.0**6**3	0.0**6**2	20.0006	19.9977	0.00	1,608	35,960
2.	256					0.0**6**2	0.0**6**3	20.0000	19.9991	0.01	12,868	174,792,640
3.	2,048					0.0**6**1	0.0**6**1	20.0000	19.9997	0.07	102,944	730,862,190,080
4.	16,384					0.0**6**0	0.0**6**0	20.0000	20.0001	0.28	823,550	3,001,300,362,981,380
5.	32	3	0.00063	64	0.06**1**	0.06**0**2	0.06**0**5	19.99999	19.99768	0.00	102,944	35,960
6.	256					0.06**0**4	0.06**0**3	19.99999	19.99909	0.01	823,550	174,792,640
7.	2,048					0.06**0**6	0.06**1**0	20.00000	19.99972	1.72	6,588,397	730,862,190,080
8.	16,384					0.06**0**2	0.06**0**2	19.99999	20.00014	1.78	52,707,179	3,001,300,362,981,380
9.	32	4	0.000063	638	0.060**1**	0.060**0**4	0.060**2**5	19.999992	19.997684	2.88	10,230,132	35,960
10.	256					0.060**0**2	0.060**2**8	19999998	19.999092	20.97	81,841,052	174,792,640
11.	2,048					0.060**0**0	0.06**10**4	20.000008	19.999715	200.44	654,728,417	730,862,190,080
12.	16,384					0.060**0**3	0.060**1**9	20.000002	20.000140	1871.23	5,237,827,339	3,001,300,362,981,380

In order to compare the performance of the MESH and the genetic algorithm, the number of operations of the genetic algorithm, #GA, should be of the same order of that of the MESH algorithm, #MESH, in (26)

$$\#\text{MESH} \approx \#\text{GA} \tag{27}$$

The number of operations of the GA, #GA, is the sum of $ps \times pc$ crossover operations (where ps is the population size and pc is the probability of crossover) and $ps \times pm$ mutation operations (where pm is the probability of mutation) at each generation (for a maximum number of $N_{generations}$). In addition, for each MZR center candidate C_{MZ}*, the maximum and minimum distance from all the (n) cloud points are evaluated

$$\#\text{GA} \approx \text{O}(n\ ps\ (pc + pm)\ N_{generations}) \tag{28}$$

As opposed to the MESH algorithm, although genetic algorithms iteratively improve solutions, they may not converge [20], consequently a stop condition should be given on the number of generations $N_{generations}$. From (24), (26), (27) and (28)

$$N_{generations} = \lceil \pi (N_{circ} / 2)^2 / (ps\ (pc + pm)) \rceil \tag{29}$$

According to the optimal GA in [20], $ps = 70$ is the population size, $pc = 0.7$ is the probability of crossover and $pm = 0.007$ is the mutation probability.

As anticipated, the number of operations for the exact algorithms from [12] [13] is

$$\#\text{EXACT} = \frac{n!}{4!(n-4)!} \tag{30}$$

The cloud size considered are $n = 2^{5+3j}$, $j = 0 \div 3$.

The last two columns clearly show the benefit of the MESH (and of the GA) algorithm where the target accuracy of the MZR error is met at a fraction of the required computation time of an exact solution.

By comparing E_{MZ}^{*} with E_{MZ}^{limit} in Table 3 it can be observed that the MESH algorithm generally exceeds the GA, although they both take advantage of the restricted search space defined in the worst case analysis in [31].

Regarding the algorithm complexity O(n N^2), by definition, the number of operations is the same for the two algorithms (27). The cloud size n is the same for both algorithms and it results that also the number of center candidates is the same: for the MESH algorithm they are all evaluated sequentially, for the GA the are evaluated in subsequent generations (29). The main difference is that the MESH algorithm considers a uniform distribution within the search space, which will satisfy the accuracy requirement by the definition of the mesh spacing ε_M (25). On the opposite, generation after generation, the GA restricts the search area in local minima by progressively increasing the density of MZR center candidates in smaller search spaces. This confirms a known drawback of metaheuristics, which are prone to being trapped in local minima; the benefit of the MESH algorithm of considering all regions of the search space S is also highlighted.

The GA parameters that were optimized for fast convergence in the order of $d = 3$, maybe require tuning at higher accuracy ($d > 3$), particularly seeking for a relationship between ps and $N_{generations}$.

For completeness also the mean radius and computation time of the GA are reported. The computation time obtained with the cited processor provides an order of magnitude, and is similar to that listed in Table 1 for MESH by construction (29).

7. Conclusions

The paper presented MESH, an ε-approximate algorithm based on the principle *only the accuracy that is really necessary*.

The benefit of the ε-approximate approach is that by an exhaustive search it is able to provide the target accuracy by construction and the computation time is given and known in advance, as opposed to most metaheuristics, which may not converge to the target accuracy.

The basic principle is to find an estimate E_{MZ}^* of the theoretical minimum zone roundness E_{MZ} with maximum error $k\ 10^{-d}$, by exhaustively assessing all MZR center candidates located at the cross points of a mesh with spacing $\varepsilon = k\ 10^{-d} / \sqrt{2}$.

The proposed method is able to provide E_{MZ}* with the desired accuracy within minutes for thousands of cloud datapoints and in negligible time for few cloud datapoints. As shown in experiments, an accuracy to the fourth decimal digit (*d*) can be met using only 300k cross points.

This result has been made possible by previous work on the limit search space to be searched, currently $\pi^{-1} E_C$.

Future research can be directed to use a similar approach to determine the minimum dataset size (e.g. depending on the profile signature and sampling error) as a function of the desired accuracy (and mesh spacing).

The proposed MESH method can also be used to separate the sampling error component from the algorithm error, because it is able to make this latter as negligible as desired.

The proposed MESH method is being extended to spheres [34], cylinders, straight lines and planes following a similar approach, anticipated in the paper.

The MESH algorithm can also serve for benchmarking purposes to assess the performance of other algorithms, such as genetic algorithm and exact methods, as experimented in this work.

8. Acknowledgements

The authors wish to thank prof. A. Weckenmann (Erlangen University) for the insightful discussions and for suggesting the potential method extension to other form tolerances.

9. References

[1] ISO 1101, Geometrical Product Specifications (GPS)–tolerances of form, orientation, location and run out, 2nd ed. International Organization for Standardization, Geneva, Switzerland, 2004, December.

[2] ASME B89.3.1 Measurement of Out-Of-Roundness, 2003.

[3] Novaski O, Chautard Barczak AL. Utilization of Voronoi diagrams for circularity algorithms. Prec Eng 1997;20;188–95.

[4] Whitehouse DJ. Handbook of surface metrology. CRC Press,1994.

[5] Samuel GL, Shunmugam MS. Evaluation of circularity from coordinate and form data using computational geometric techniques, Prec Eng 2000;24;251–63.

[6] Roy U, Zhang X. Establishment of a pair of concentric circles with the minimal radial separation for assessing roundness error. Comput Aided Design 1992;24(2);161–68.

[7] Li X, Liu H, Li W. Development and application of α-hull and Voronoi diagrams in the assessment of roundness error. Meas Sci Technol 2011;22;045105 doi:10.1088/0957-0233/22/4/045105.

[8] Dhanish PB, Shunmugam MS. An algorithm for form error evaluation – using the theory of discrete and linear chebyshev approximation. Comput Method Appl M 1991;92;309–24.

[9] Weber T, Motavalli S, Fallahi B, Cheraghi SH. A unified approach to form error evaluation. Prec Eng 2002;26;269–78.
[10] Murthy TSR, Abdin SZ. Minimum zone evaluation of surfaces. International Journal of Machine Tool Design Research 1980;20;123–36.
[11] Zhu LM, Ding H, Xiong YL. A steepest descent algorithm for circularity evaluation. Comput Aided Design 2003;35(3);255–65.
[12] Wang M, Cheraghi SH, Masud ASM. Circularity error evaluation: theory and algorithm, Prec Eng 1999;23(3);164–76.
[13] Jywe W-Y, Liu G-H, Chen C-K. The min-max problem for evaluating the form error of a circle. Measurement 1999;26;273-82.
[14] Gadelmawla ES. Simple and efficient algorithm for roundness evaluation from the coordinate measurement data. Measurement 2010;43;223-35.
[15] Kovvur Y, Ramaswami H, Anand BR, Anand S. Minimum-zone form tolerance evaluation using particle swarm optimization. International Journal of Intelligent Systems Technologies and Applications 2008;4(1);79-96.
[16] Mao J, Cao Y, Yang J, Implementation uncertainty evaluation of cylindricity errors based on geometrical product specification (GPS). Measurement 2009;42(5);742-47.
[17] Ke K. Minimum Zone Evaluation of Sphericity Error Based on Ant Colony Algorithm. In: Electronic Measurement and Instruments, 2007. ICEMI'07. 8th International Conference on. IEEE 2007. 2 535-38.
[18] Shakarji CM, Clement A. Reference Algorithms for Chebyshev and One-Sided Data Fitting for Coordinate Metrology. CIRP Ann Manuf Technol 2004;53(1);439-42.
[19] Wen XL, Song A. An immune evolutionary algorithm for sphericity error evaluation. Int J Mach Tools Manuf 2004;44;1077–84.
[20] Rossi, A, Antonetti M, Barloscio M, Lanzetta M. Fast genetic algorithm for roundness evaluation by the minimum zone tolerance (MZT) method. Measurement 2011;44(7);1243-52 doi:10.1016/j.measurement.2011.03.031
[21] Wen X, Xia Q, Zhao Y. An effective genetic algorithm for circularity error unified evaluation. Int J Mach Tools Manuf 2006;46;1770–77.
[22] Cui C, Li B, Huang F, Zhang R. Genetic algorithm-based form error evaluation. Meas Sci Technol 2007;18(7);1818 doi:10.1088/0957-0233/18/7/004.
[23] Jianxi P, Xiao Y, Jianping L, ZhiYuan L. Research of sphericity error evaluation method of calculating the uncertainty in the specification. In: Computational Intelligence and Design (ISCID), 2011 Fourth International Symposium on. IEEE, 2011. 2 330-33.
[24] Chen M-C. Analysis of Spherical Form Errors to Coordinate Measuring Machine Data. JSME International Journal Series C 2002;45(2):647-56.
[25] Meo, A, Profumo, L, Rossi, A, Lanzetta, M. Optimum dataset size and search space for minimum zone roundness evaluation by genetic algorithm. Measurement Science Review 2013;13(3);100-107.
[26] Colosimo BM, Moroni G, Petrò S. A tolerance interval based criterion for optimizing discrete point sampling strategies. Prec Eng, 2010;34(4);745-54.
[27] Rossi A. A form of deviation-based method for coordinate measuring machine sampling optimization in an assessment of roundness. P I Mech Eng B-J Eng 2001;215;1505–18.
[28] Weckenmann A, Heinrichowski M. Problems with software for running coordinate measuring machines: the use of virtual volumetric standards. Prec Eng 1985;7(2);87-91.
[29] Lei X, Zhang C, Xue Y, Li J. Roundness Error Evaluation Algorithm Based on Polar Coordinate Transform. Measurement 2011;44 (2);345-350.

[30] Rossi A, Lanzetta M, Optimal blind sampling strategy for minimum zone roundness evaluation by metaheuristics. Prec Eng, 2013;37(2);241-247 doi:10.1016/j.precisioneng.2012.09.001.

[31] Rossi A, Lanzetta M. Roundness: a closed form upper bound for the centroid to minimum zone center distance by worst-case analysis. Measurement, 2013;46(7);2251-58 doi: 10.1016/j.measurement.2013.03.025.

[32] Sharma R, Rajagopal K, Anand S. A Genetic Algorithm Based Approach for Robust Evaluation of Form Tolerances. J Manuf Syst, 2000;19(1);46-57.

[33] Yan L, Yan B, Cai L, Hu G, Wang M. Research on roundness error evaluation of shaft parts based on genetic algorithms with transfer-operator. In: Electronic Measurement & Instruments, 2009. ICEMI '09. 9th International Conference on. IEEE 2009. 2 362-66. doi:10.1109/ICEMI.2009.5274566.

[34] Rossi A, Chiodi S, Lanzetta M. Minimum centroid neighborhood for minimum zone sphericity, Prec Eng 2014;38(2);337-47 doi:10.1016/j.precisioneng.2013.11.004.

[34] Rossi A, Chiodi S, Lanzetta M. Minimum centroid neighborhood for minimum zone sphericity, Prec Eng 2014;38(2);337-47 doi:10.1016/j.precisioneng.2013.11.004.