Minimum-zone form tolerance evaluation using particle swarm optimisation

Yashpal Kovvur, Hemant Ramaswami,
Raj Bardhan Anand and Sam Anand*

Computer-Aided Manufacturing Laboratory,
Industrial and Manufacturing Engineering Program,
University of Cincinnati,
Cincinnati, OH 45221-0072, USA
Fax: +1-513-556-3390
E-mail: kovvuryl@email.uc.edu
E-mail: ramaswht@email.uc.edu
E-mail: anandrb@email.uc.edu
E-mail: sam.anand@uc.edu
*Corresponding author

Abstract: Robust and accurate evaluation of form tolerances is of paramount importance in today's world of precision engineering. Present-day Coordinate Measuring Machines (CMMs) and other optical scanning machines operate at high speed and have a high degree of accuracy and repeatability which are capable of meeting the stringent measurement requirements. However, the evaluation algorithms used in conjunction with them are not robust and accurate enough, because of the highly non-linear nature of the minimum-zone form tolerance formulation. Evolutionary Algorithms (EAs) have proved effective in solving non-linear optimisation problems. In this paper, Particle Swarm Optimisation (PSO) is employed to evaluate various minimum-zone form tolerances. An unconstrained formulation of the minimum-zone form tolerance is used for the optimisation. The methodology is validated by testing on several datasets from published literature and yields equal or better results than other existing minimum-zone algorithms. It is also extremely robust and the quality of the results is not affected by the number of points in the dataset.

Keywords: form tolerance; particle swarm optimisation; PSO; minimum zone; circularity; cylindricity; sphericity; straightness; flatness.

Reference to this paper should be made as follows: Kovvur, Y., Ramaswami, H., Anand, R.B. and Anand, S. (2008) 'Minimum-zone form tolerance evaluation using particle swarm optimisation', *Int. J. Intelligent Systems Technologies and Applications*, Vol. 4, Nos. 1/2, pp.79–96.

Biographical notes: Yashpal Kovvur is a Doctoral student in Industrial and Manufacturing Engineering Program at the University of Cincinnati. He received a Bachelor's degree in Mechanical Engineering from the Indian Institute of Technology, Chennai in 2001 and Master's Degree in Industrial and Manufacturing Engineering from the University of Cincinnati in 2003. His research interests include CAD/CAM, Computational Metrology, Surface Characterisation and Surface Prediction.

Hemant Ramaswami is a Doctoral student in Industrial and Manufacturing Engineering Programme at the University of Cincinnati. He received a Bachelor's degree in Mechanical Engineering from the University of Mumbai, India in 1999 and Master's Degree in Industrial and Manufacturing Engineering from the University of Cincinnati in 2002. His research interests include CAD/CAM, Tolerancing, Computational Metrology and development of algorithms for efficient manufacturing. He is a Member of the ASME and Tau Beta Pi.

Raj Bardhan Anand received a Bachelors of Engineering degree from National Institute of Foundry and Forge Technology, India. Currently, he is a graduate research student in the Department of Mechanical, Industrial and Nuclear Engineering at the University of Cincinnati, OH, USA. His research interests include AI-based approaches to solve complex problems in Manufacturing Systems, Evolutionary Algorithms and Statistical Quality Control.

Sam Anand received an ME (Distinction) in Mechanical Engineering from the Indian Institute of Science, India, and a MS and PhD in Industrial Engineering from the Pennsylvania State University. Currently, he is a Professor of Mechanical Engineering at the University of Cincinnati. He also serves as the Director of the CAM Lab and a Co-Director of the Centre for Global Design and Manufacturing. His research interests are in the areas of Optimisation Methods in Manufacturing, Tolerancing, Computational Metrology, Surface Characterisation, and Product Lifecycle Management. He is a Senior Member of SME and IIE and a Member of ASME.

1 Introduction

The geometric form of any manufactured component deviates from its nominal design to some extent because of systematic and random errors. Current manufacturing standards demand that the components produced adhere strictly to Geometric Dimensioning and Tolerancing specifications (American National Standards Institute, 1994; International Organization for Standardization, 2004) including form tolerance. Computer-Aided Inspection using Coordinate Measuring Machines (CMMs) or other optical scanning machines is the standard procedure followed in most manufacturing applications. The effectiveness of such an inspection method is influenced by the capabilities of the hardware and software used. The high degree of repeatability and accuracy of existing hardware combined with the enhanced data acquisition speeds have made it possible to collect a large sample of points. With the availability of these large datasets in the order of hundreds or thousands of points, more emphasis needs to be placed on researching efficient and robust algorithms to analyse the data collected to provide more accurate results.

As defined by ANSI (1994) and ISO (2004), the minimum zone form tolerance is based on minimising the maximum deviation of the inspected surface from a reference feature. An accurate evaluation of the minimum zone form tolerance is possible either using specific computational geometry techniques or by solving non-linear optimisation problems. This optimisation function has numerous local minima, and most solution methods need a set of very good starting points to obtain the global minimum solution. Also, the formulation for the tolerance zone is typically non-differentiable, which limits the use of conventional gradient-based optimisation techniques.

Particle Swarm Optimisation (PSO), which is one of the most recent and popular Evolutionary Algorithms (EAs), has been shown to perform well under such conditions. In the PSO approach, each particle of the population, called the swarm, progresses towards the optimal solution by adjusting its trajectory toward its own previous best position, and toward the previous best position attained by any member of the population. The advantage of using PSO over other optimisation methods is the relatively insignificant impact of the starting conditions. In this paper, the PSO algorithm has been used to evaluate the circularity, cylindricity, sphericity, flatness and straightness form tolerances.

2 Literature review

Minimum zone form tolerance evaluation has been the subject of many research papers. Some of these papers focus on the presentation of a generalised methodology for multiple form tolerances (Carr and Ferreira, 1995a,b; Dhanish and Shunmugam, 1991). However, most research published concentrate on a single form tolerance. This is probably due to the vastly differing nature of the different form tolerance formulations. Form tolerance evaluation algorithms can be roughly classified into two families:

1 numerical methods

2 geometric methods.

Numerical methods quantify the given dataset as a set of deviations from an ideal geometry and seek to minimise these deviations. These methods are, in general, fast but yield inaccurate values owing to mathematical approximations. Examples of this family are the Chebyshev approximation (Dhanish and Shunmugam, 1991), simplex search (Murthy and Abdin, 1980), linear and non-linear optimisation (Carr and Ferreira, 1995a,b; Zhu et al., 2003), simulated annealing (Hong et al., 2001), Genetic Algorithms (GA) (Lai et al., 2000; Sharma et al., 2000) and Monte Carlo methods (Murthy and Abdin, 1980).

Algorithms based on geometric methods faithfully formulate the geometrical model of the problem and seek to identify the minimum bounds that define the tolerance. As they adhere more closely to the ANSI definitions of tolerances, more often than not these algorithms provide exact solutions. These methods are typically computational geometry-based and owing to their iterative nature they are computationally expensive.

2.1 Generalised methodologies for form error assessment

Murthy and Abdin (1980) proposed a normal least squares deviation formulation for flatness, straightness, circularity and sphericity and used Monte Carlo, simplex search and spiral search techniques to determine the minimum zone solution. Dhanish and Shunmugam (1991) modelled the ideal form of various features as linear Chebyshev polynomial functions and found an approximation function that minimised the maximum error between the Chebyshev function and the actual surface. They demonstrated this method for straightness, flatness, circularity, sphericity and cylindricity. Carr and Ferreira (1995a,b) presented a methodology based on solving a sequence of linear programs to compute various form tolerances. In this method, the solution of one linear programme is used as the input to the next, thereby reducing the error induced due to the linearisation. Sharma et al. (2000) used GA to evaluate various form tolerances. They used a min-max formulation for the objective function and bounded the ranges of the variables. Weber et al. (2002) presented a unified linear approximation technique for evaluating various form tolerances. They formulated the linear programme using a Taylor expansion of the minimum zone tolerance function.

2.1.1 Minimum zone circularity assessment methods

Roy and Zhang (1992) and Huang (1999b) proposed methods based on the convex hull of the dataset and Voronoi diagrams. Though these algorithms are effective, they require substantial computational resources when used in conjunction with large datasets. Rajagopal and Anand (1999) partitioned the data into quadrants and iteratively selected the basis points from each quadrant to evaluate the circularity. Samuel and Shunmugam (2003) transformed the data with respect to a reference circle in polar coordinates and approximated the transformed data into a feature called Limacon.

2.1.2 Minimum zone sphericity assessment methods

Shunmugam (1986) developed an approach called median technique which considered the extreme points, that is, crest and valley points for evaluating the sphericity, thereby saving considerable time. Kanada (1997) simplified the sphericity definition in terms of 2D measurement of cross-sectional profiles and used statistical techniques to determine the number of cross-section profiles necessary to estimate the sphericity from the 2D roundness values. Huang (1999a) proposed a computational geometry-based technique to evaluate sphericity error using 3D Voronoi diagrams. Samuel and Shunmugam (2002) proposed a computationally robust algorithm to calculate the sphericity error using a 3D assessment feature called limacoid, which is an extension of the limacon concept used for circularity evaluation.

2.1.3 Minimum zone cylindricity assessment methods

Lai and Chen (1996) employed a non-linear transformation method to convert a cylinder into a plane and they applied a flatness evaluation scheme to obtain appropriate control points. The cylindrical parameters were then computed by implementing a series of inverse transformations. Hodgson et al. (1999) proposed a combinatorial solution that efficiently searches for the six points in the dataset that are used to determine the minimum zone cylinders. Barcenas and Griffin (2000) presented a statistics-based technique by developing a parametric representation to mathematically model the part. They used non-linear regression to fit the actual measured points onto the parametric representation. Zhu and Ding (2003) defined a point-to-surface function for a point expressed in the machine reference frame and a surface expressed in its own model frame. This was followed by the derivation of expressions to represent its increment with respect to the differential motion of the surface. The non-linear problem so obtained was solved using approximation algorithms.

2.1.4 Minimum-zone straightness and flatness assessment methods

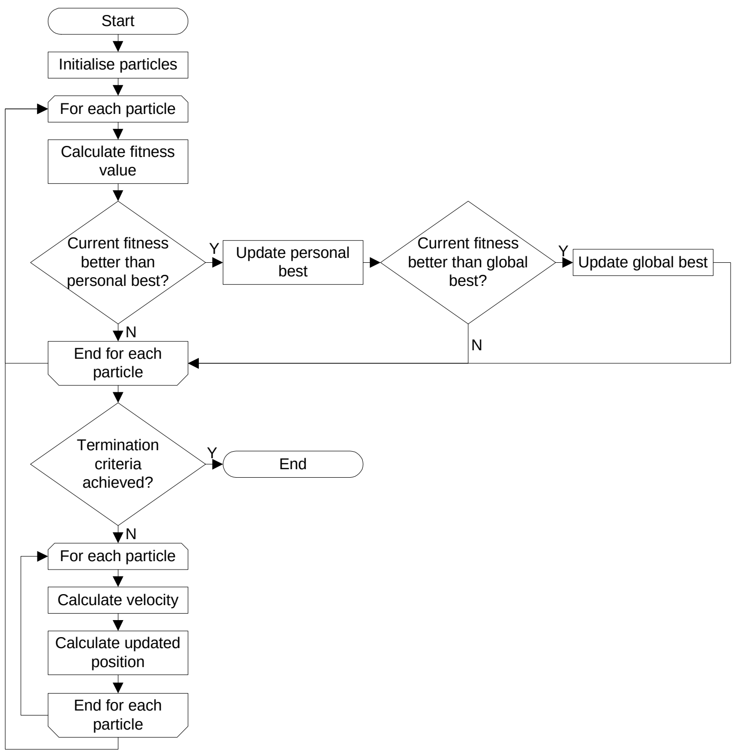
Cheraghi et al. (1996) reduced the non-linear problem to a linear programming problem with two constraints and used a linear search method to find the minimum zone flatness and straightness. Samuel and Shunmugam (1999) used the 2D convex hull and formed pairs of edges and their antipodal points to evaluate the straightness. For evaluating the flatness, the 3D convex hull of the dataset was used with the corresponding antipodal points. Hong et al. (2001) developed a method based on a combination of computational geometry and non-linear optimisation using simulated annealing to solve the non-linear optimisation problem.

2.2 Particle swarm optimisation

Particle swarm optimisation (PSO) is a population-based stochastic technique developed by Kennedy and Eberhart (1995), based on the social behaviour metaphor. In PSO, a population of random solutions is initialized and the optimal solution is searched by updating generations. The potential solutions search through the problem space by tracking the fitness of the other solutions and their personal fitness history.

PSO has similarities to other EA methods such as Genetic Algorithms (GA) with regard to random initialisation of population and searching for the optimal solution by updating generations. However the PSO algorithm does not use any evolution operators such as mutation and crossover and is easier to implement.The flow of the PSO algorithm is depicted in Figure 1.

Figure 1 Outline of PSO algorithm



PSO has been used considerably in the fields of design and manufacturing for problems related to optimal power flow (Abido, 2002), tolerancing (Ramaswami et al., 2005; Zhao and Chen, 2005), structural design (Schutte and Groenwold, 2003) and cutting conditions (Saravanan et al., 2005). The minimum-zone form tolerance formulations in the present paper involve unconstrained non-linear optimisation problems. As mentioned earlier, due to the complexity and unpredictability of such problems, a general deterministic solution is impossible. This provides an opportunity for EAs such as PSO.

2.3 Parameter selection for PSO

The effectiveness of the algorithm is significantly affected by the number of particles in the swarm (*N*) and the initial positions of the particles in the feature space. A very small number of particles will require low execution times. However, the particles may not efficiently search the entire feature space and the algorithm will get stuck in a local optimum. Conversely, if the swarm size is very large, the coverage of the search space will be much better. However, the execution time for each iteration will be very high and this will slow down the algorithm. Van den Bergh and Engelbrecht (2001) studied the effect of swarm size and concluded that for simple swarms, the average final error decreases with increase in the number of particles. They also concluded that the optimum swarm size was highly dependent on the nature of the function.

The initialisation of the swarm also plays a significant role in early convergence of the swarm. Usually, the swarm is initialised using a uniform distribution over the search space. However, this often yields poor results. To overcome this problem, Parsopoulos and Vrahatis (2001) recommended the use of a Sobol sequence generator to ensure that the vectors are uniformly distributed in the search space. Another alternative proposed by Parsopoulos and Vrahatis (2002), which shows an improvement in convergence and success rates, is the use of non-linear simplex method for initialisation.

Other parameters that influence the convergence and success rates are the various weights associated with the particle swarm which govern the velocities of the particles. Shi and Eberhart (1998) described different ranges for the cognitive and social acceleration parameters. These parameters control the motion of any particle in the direction of the location of its personal best performance and the best performance by any particle of the swarm. Clerc and Kennedy (2002) demonstrated that properly defined constriction coefficients can prevent explosion and induce particles to converge at local minima.

Details of the minimum zone form tolerance formulations, their adaptation for PSO, initialisation of the swarm particles and selection of behavioural parameters are discussed in subsequent sections.

3 Methodology

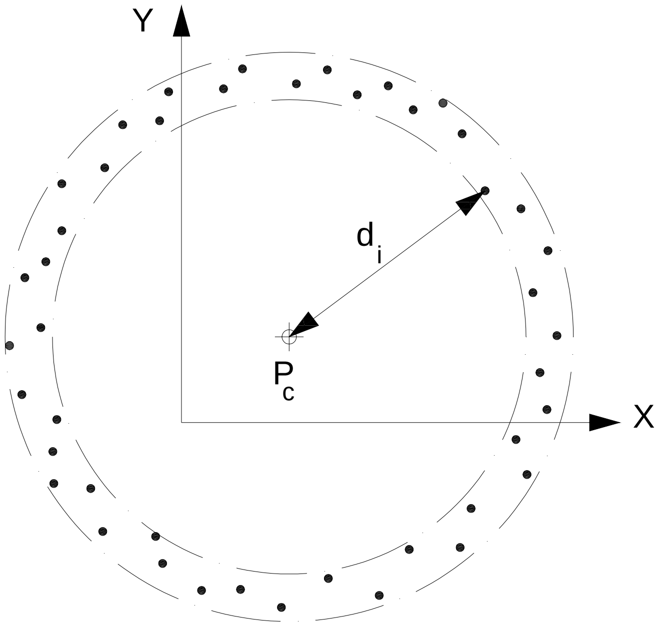
The PSO algorithm is best suited for unconstrained problems. Hence, to utilise the capabilities to the limit, the formulations for the various form tolerances were developed as unconstrained non-linear problems. The formulation is based on finding the parameters of the ideal reference feature which minimises the difference between the distances of the points nearest and farthest with respect to the reference feature.

The definition of the ideal reference feature is different for closed features such as circles and spheres compared to open features such as cylinders, planes and lines. For closed features, the parameters computed automatically indicate the exact location of the reference features. This ensures that the problem has a unique solution. However, for open features, additional restrictions have to be imposed on the reference features to ensure that the solution is unique. Details of the restrictions have been provided later in this section along with the appropriate formulations.

3.1 Minimum-zone circularity formulation

In order to compute the circularity of a set of points, the ideal reference feature is a circle. The only information necessary to compute the circularity are the coordinates of the centre of the minimum-zone circles. As shown in Figure 2, P_c represents the centre of the circles and d_i represents the distance of the *i*th point from the centre.

Figure 2 Modelling of circularity tolerance



The minimum zone circularity problem can then be written as:

$$\text{Minimise}\Big(\max\big(d_i\left(P_c\right)\big)-\min\big(d_i\left(P_c\right)\big)\Big) \tag{1}$$

where

$$d_i\left(P_c\right)=\sqrt{\left(x_i-x_c\right)^2+\left(y_i-y_c\right)^2}, \quad i=1,2,3,\ldots,m$$

$P_i = (x_i, y_i)$, $i = 1, 2, 3,\ldots, m$ are the coordinates of the m points in the dataset and $P_c = (x_c, y_c)$ are the coordinates of the centre of the minimum zone circles.

Since only two parameters, that is, the coordinates of the centre need to be determined, the dimensionality of the problem is two.

3.2 Minimum-zone sphericity formulation

Similar to the circularity problem, the reference feature for sphericity is a sphere. The only parameters required to compute the sphericity value are the coordinates of the centre of the spheres. As illustrated in Figure 3, P_c is the centre of the spheres and d_i is the distance of any point from the centre. The optimisation model is a function of three variables, that is, the three coordinates of the centre of the spheres.

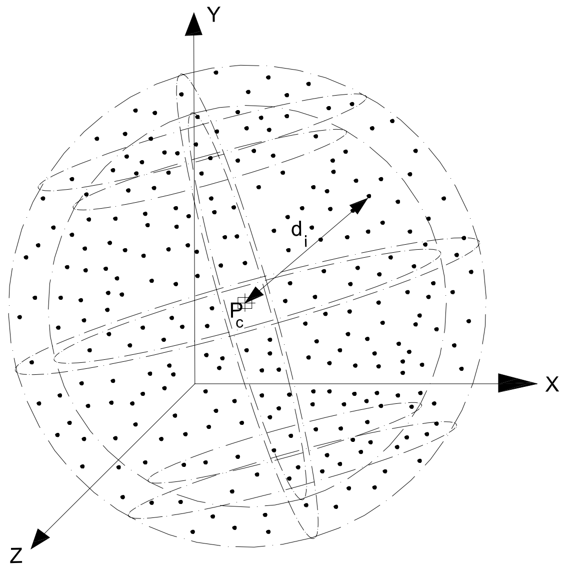
$$\text{Minimise}\left(\max\left(d_i\left(P_c\right)\right)-\min\left(d_i\left(P_c\right)\right)\right) \tag{2}$$

where

$$d_i\left(P_c\right)=\sqrt{\left(x_i-x_c\right)^2+\left(y_i-y_c\right)^2+\left(z_i-z_c\right)^2},\quad i=1,2,3,\ldots,m$$

$P_i = (x_i, y_i, z_i)$, $i = 1, 2, 3,\ldots, m$ are the coordinates of the m points in the dataset and $P_c = (x_c, y_c, z_c)$ are the coordinates of the centre of the minimum zone circles.

Figure 3 Modelling of sphericity tolerance



3.3 Minimum-zone cylindricity formulation

The reference feature for cylindricity is a perfect cylinder, which can be characterised by determining the location and orientation of its axis. As shown in Figure 4, the axis is characterised by a point P_c on it and the unit direction vector T_c, which defines its orientation. The problem in this state has infinite optimal solutions, as any point on the axis will be an optimal solution. To transform the problem to have a unique solution, a restriction is applied on P_c by forcing its z coordinate to be zero. Also, only two of the three components of the unit direction vector, T_c need to be considered as variables because the third component can be determined as $T_z=\sqrt{1-T_x^2-T_y^2}$. The resulting dimensionality of this problem is four.

The minimum zone cylindricity formulation is:

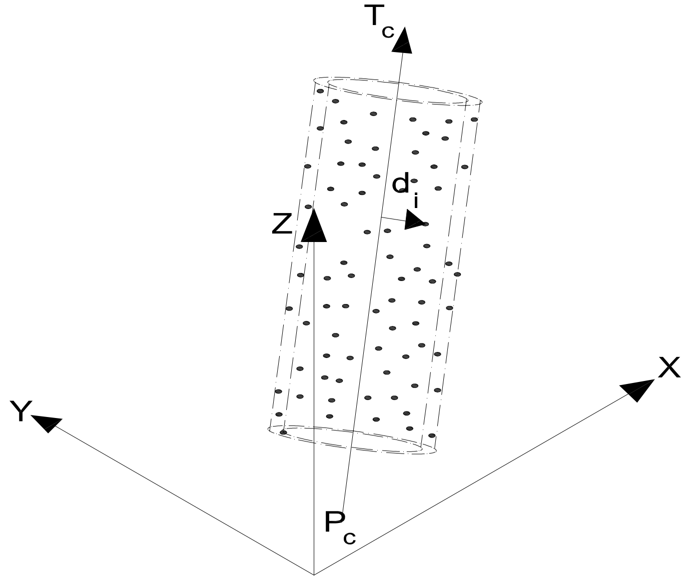
$$\text{Minimise}\left(\max\left(d_i\left(P_c,T_c\right)\right)-\min\left(d_i\left(P_c,T_c\right)\right)\right) \tag{3}$$

where

$$d_i\left(P_c,T_c\right)=\left|T_c\times\left(P_i-P_c\right)\right|,\quad i=1,2,3,\ldots,m$$

P_i = (x_i, y_i, z_i), i = 1, 2, 3,…, m are the coordinates of the m points in the dataset P_c = (x_c, y_c, 0) are the coordinates of a point on the axis of the minimum zone cylinders and $T_c=(T_x,T_y,\sqrt{1-T_x^2-T_y^2})$ are the components of the unit direction vector of the axis of the minimum zone cylinders.

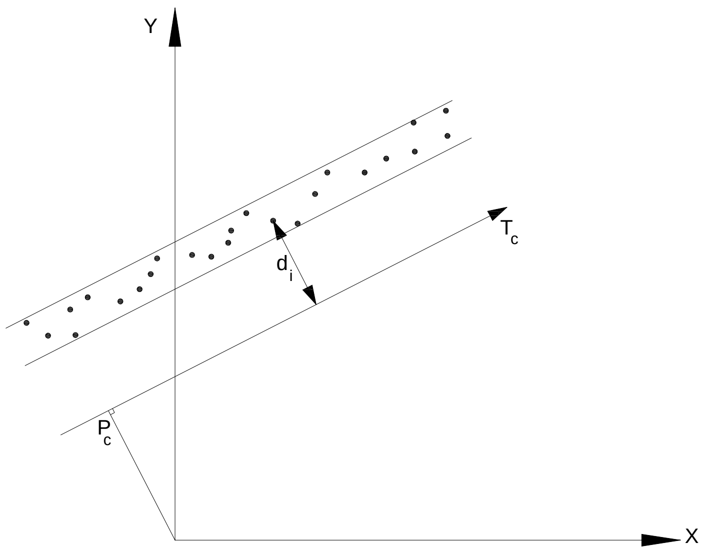
Figure 4 Modelling of cylindricity tolerance



3.4 Minimum-zone straightness formulation

The straightness of a set of data points collected from a line can be evaluated by measuring their distance from any line that is parallel to the minimum-zone reference line. This results in more than one optimal solution to this problem. To eliminate this ambiguity and to ensure that a unique solution can be obtained, an additional restriction is placed on the line by fixing a point (P_c) on the line. This point is selected such a line perpendicular to the reference line and passing through this point will also pass through the origin of the coordinate system, as illustrated in Figure 5. Since, the points in the dataset are in the *x*-*y* plane, the component of the unit direction vector along the *z*-direction is zero. Thus, if the component of the unit direction vector of the line along the *x*-axis is represented by T_x, the component of the unit direction vector of the line along the *y*-axis can be calculated as $T_y=\sqrt{1-T_x^2}$. If the coordinates of P_c are (x_c, y_c) and the value of x_c is known, the value of y_c can be computed as $y_c=-(T_x/T_y)x_c$.

Figure 5 Modelling of straightness tolerance



The number of variables, and consequently, dimensionality of the problem, is two, that is, x_c and T_x.

The minimum zone straightness formulation is:

$$\text{Minimise}\left(\max\left(d_i\left(P_c,T_c\right)\right)-\min\left(d_i\left(P_c,T_c\right)\right)\right) \tag{4}$$

where

$$d_i\left(P_c,T_c\right)=\left|T_c\times\left(P_i-P_c\right)\right|,\quad i=1,2,3,\ldots,m$$

$P_i = (x_i, y_i, z_i)$, $i = 1, 2, 3,\ldots, m$ are the coordinates of the m points in the dataset.

$P_c = (x_c, y_c, 0)$ are the coordinates of a point on a line parallel to the minimum zone lines lying on the perpendicular to the line from the origin and $T_c = (T_x, \sqrt{1-T_x^2}, 0)$ are the components of the unit direction vector of the minimum zone lines.

3.5 *Minimum-zone flatness formulation*

Similar to the argument put forth in the straightness formulation, the flatness can be evaluated using a plane that is parallel to the minimum-zone reference plane and located anywhere in space. In this case, to uniquely define the solution, the reference plane is restrained by forcing it to pass through the origin, as shown in Figure 6. The only parameters to be determined in order to calculate the flatness tolerance are the direction cosines of the normal to the reference plane. A reasoning similar to the one used in the straightness formulation can be used to determine the parameters for the minimum-zone plane. If T_x and T_y are the components of the unit normal to the plane along the x and y directions, then the z-component can be calculated as $T_z = \sqrt{1-T_x^2-T_y^2}$. The resulting dimensionality of the flatness evaluation formulation is two.

The minimum-zone flatness formulation is:

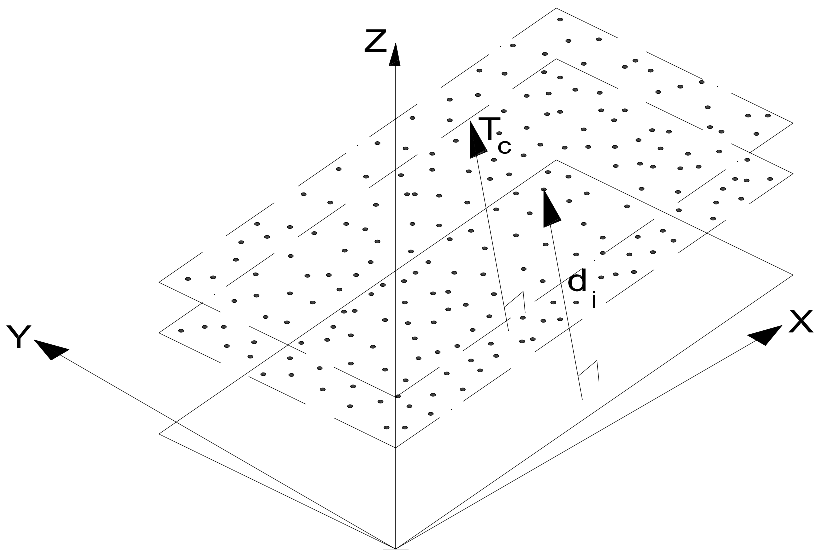
$$\text{Minimise}\left(\max\left(d_i\left(T_c\right)\right)-\min\left(d_i\left(T_c\right)\right)\right) \quad (5)$$

$$d_i\left(T_c\right)=\left|T_c \bullet P_i\right|, \quad i=1,2,3,\ldots,m$$

$P_i = (x_i, y_i, z_i)$, $i = 1, 2, 3,\ldots, m$ are the coordinates of the m points in the dataset.

$T_c = (T_x, T_y, \sqrt{1-T_x^2-T_y^2})$ are the direction cosines of the normal to the minimum zone planes.

Figure 6 Modelling of flatness tolerance



4 Adaptation of the form tolerance formulation for PSO

The PSO algorithm operates by evaluating the value of the objective function at each particle's location and using the information about the location of the particle with the minimum function value.

As discussed by Parsopoulos and Vrahatis (2002), the movement of the particles across iterations is achieved by calculating the new velocity using Equation (6) and determining the new position by adding the velocity to the current position as shown in Equation (7).

$$V_i^{k+1}=\chi\left(wV_i^k+c_1r_{i1}^k\left(P_i^k-X_i^k\right)+c_2r_{i2}^k\left(P_g^k-X_i^k\right)\right) \quad (6)$$

$$X_i^{k+1}=X_i^k+V_i^{k+1} \quad (7)$$

The role of the inertia weight (w) is extremely important for convergence of the PSO. It controls the impact of previous history of velocities on the current velocity. In essence, it controls the balance between global and local exploration capabilities of the swarm. If the inertia weight is large, the swarm shows an exploring behaviour (particles travelling to unexplored areas). If the inertia weight is small, the swarm shows an exploiting behaviour (fine tuning in the current search area). The selection of a proper value of w provides a balance and yields better solutions.

In this paper, the inertia weight is initially set to 1.2 to favour global exploration. It is exponentially reduced to 0.3 over 150 iterations as suggested by Parsopoulos and Vrahatis (2002). This is done to favour the local exploration capabilities once the particles are in the neighbourhood of the global optimum. Once it reaches a value of 0.3, it is held at that value to provide sufficient inertia for the algorithm to converge. Thus the inertia weight for any iteration (w) can be calculated as

$$w = \max\left(w_{\text{orig}} \times e^{-\lambda \times \text{iter\_no}}, w_{\min}\right) \tag{8}$$

where w_{orig} is the inertia weight at the start; iter_no is the current iteration number; $w_{\min}$ is the inertia weight at the end of the decay period; λ is the exponential decay coefficient calculated as

$$\lambda = \frac{\ln\left(w_{\text{orig}} / w_{\min}\right)}{\text{decay\_iter}} \tag{9}$$

where decay_iter is the number of iterations over which the decay occurs.

The constriction factor χ is used to control and constrict velocities. The cognitive acceleration parameter c_1 controls the influence of the personal best value of the particle on the velocity of the particle. This parameter forces the particle to move towards the location where it had its personal minimum function value. The social acceleration parameter c_2 determines the influence of the best particle in the swarm on the particle. This parameter influences the particle to move towards the best location found by any particle until that iteration. By adjusting these parameters, the particle can be made to learn more either from its own experience or from the experiences of other particles.

In this paper, the following values were used, as suggested by Parsopoulos and Vrahatis (2002): $c_1 = c_2 = 2$, which provides for equal learning from self-experience and experience of other particles. The constriction factor χ is set to 0.73 to provide for some damping of the velocities.

The initial positions of the 50 particles are generated using a combination of the simplex search and random generation using a Sobol sequence generator, as suggested by Parsopoulos and Vrahatis (2001, 2002). This seems to be helpful when dealing with highly non-linear functions since the simplex search generates some particles that are advancing in the direction of a minimum, whose global optimality cannot be guaranteed. The Sobol sequence is a quasirandom sequence which is less random than a pseudorandom number sequence. However, it is much more effective for tasks such as global optimisation because it tends to sample the space more uniformly than random numbers. The particles generated using the Sobol sequence generator provides for greater exploration of the function space.

The simplex search is initialised from the least squares solution and run for 150 function evaluations and all the simplex vertices are collected. If the number of simplex vertices generated is greater than the number of particles in the swarm, half the swarm is populated by randomly selecting from the simplex vertices. The remaining particles are generated using the Sobol sequence generator which is seeded using a randomly generated six-digit number. If the number of simplex vertices is less than the number of particles required, half of the vertices are randomly selected as initial locations of the particles. The remaining particles are generated using the Sobol sequence generator with a random six-digit number seed. The implementation of the Sobol sequence generator used in this research was adapted from the work of Burkardt (2005). The performance of each particle is measured according to the fitness function, which is the difference in distance between the nearest and farthest points from the reference feature.

Multiple termination criteria are available in the algorithm. If the particles are in the vicinity of the optimal solution, the algorithm terminates if all the particles converge to within 1×10^{-8} of the best solution. Successful termination is also achieved if the difference between the objective function values of all the particles is less than 1×10^{-8}. Further, if the number of iterations exceeds 500, it is assumed that the algorithm has failed and it is terminated.

5 Results

The formulations for the different form tolerances were implemented in Matlab and tested on a Windows platform machine with a 3.2 GHz Pentium® 4 processor and 2 GB RAM. The datasets used for testing the algorithm were selected from published literature. This permitted the comparison of the present method with the published results. The form tolerance of each dataset was evaluated 100 times and the average behaviour of the algorithm was studied. The results are presented in Tables 1–5. The form tolerance values provided by the proposed method are compared with the values in published literature to study the performance of the algorithm.

For verifying the circularity error formulation, datasets 1 and 2, with 25 and 24 points, respectively, were obtained from Zhu et al. (2003). Dataset 3, having 39 points, was obtained from Jywe et al. (1999), while dataset 4, consisting of 20 points, was obtained from example 5 of Rajagopal and Anand (1999). The comparison of the circularity values obtained using PSO and other published literature is shown in Table 1.

Table 1 Comparison of minimum-zone circularity results

No.	*Source*	*No. of points*	*Least squares*	*Published*	*Particle swarm optimisation*	*Average No. of iterations*
1	Zhu (2003)	25	0.029807	0.029280	0.029280	84.47
2	Zhu (2003)	34	0.039101	0.038231	0.038231	87.06
3	Jywe (1999)	39	0.0092	0.0085	0.0085	59.28
4	Rajagopal (1999)	20	1.7486	1.6711	1.6711	75.25

To verify the sphericity error formulation, datasets 1 and 2, with 40 and 36 points, respectively, were selected from the work of Chen and Liu (2000). Dataset 3, with 100 points, was was obtained from Huang (1999a). Dataset 4, with 50 points, was obtained from the work of Samuel and Shunmugam (2003). Table 2 shows the comparison of the sphericity values obtained using PSO and other published literature.

Table 2 Comparison of minimum-zone sphericity results

No.	*Source*	*No. of points*	*Least squares*	*Published*	*Particle swarm optimisation*	*Average No. of iterations*
1	Chen (2000)	40	0.009081	0.008327	0.008327	53.71
2	Chen (2000)	36	0.01007	0.00967	0.00967	50.51
3	Huang (1999a)	100	1.1419	1.0	1.0114	78.54
4	Samuel (2003)	50	0.008487	0.00766011	0.0076602	63.06

Dataset 1 for cylindricity evaluation, having 40 points, was obtained from dataset 3 of the paper by Carr and Ferreira (1995b). Dataset 2, with 24 points, was obtained from the work of Lai et al. (2000). Datasets 3 and 4, with 16 and 24 points, were obtained from datasets 1 to 4 of Weber et al. (2002). The comparison of the cylindricity values obtained using PSO and other published literature is shown in Table 3.

Table 3 Comparison of minimum-zone cylindricity results

No.	*Source*	*No. of points*	*Least squares*	*Published*	*Particle swarm optimisation*	*Average No. of iterations*
1	Carr (1995b)	40	0.01037	0.00941	0.00941	94.23
2	Lai (2000)	24	0.002879	0.002788	0.002788	88.67
3	Weber (2002)	16	0.00672	0.0060	0.005894	85.21
4	Weber (2002)	24	0.000384	0.0004	0.0003756	83.24

To verify the straightness formulation, datasets 1 to 3 with 5, 10 and 25 points, respectively, were obtained from the correspondingly numbered dataset in the work of Carr and Ferreira (1995a). Dataset 4, with 35 points, was selected from dataset 2 in the work of Weber et al. (2002). Table 4 shows the comparison of the results obtained using PSO with other published results. The datasets for verification of the flatness formulation, with 15, 25, 25 and 70 points, respectively, were obtained from the correspondingly numbered datasets in the work of Carr and Ferreira (1995a). The comparison of the results obtained using PSO with other published results is shown in Table 5.

Table 4 Comparison of minimum-zone straightness results

No.	*Source*	*No. of points*	*Least squares*	*Published*	*Particle swarm optimisation*	*Average No. of iterations*
1	Carr (1995a)	5	0.0028	0.002666	0.002666	88.63
2	Carr (1995a)	10	0.89565	0.8578577	0.8578577	83.95
3	Carr (1995a)	25	0.001463	0.001311	0.001311	79.72
4	Weber (2002)	35	0.001656	0.0014	0.001412	79.56

Table 5 Comparison of minimum-zone flatness results

No.	*Source*	*No. of points*	*Least squares*	*Published*	*Particle swarm optimisation*	*Average No. of iterations*
1	Carr (1995a)	15	0.0028	0.0025000	0.0024999	91.19
2	Carr (1995a)	25	0.0059	0.0048636	0.0048636	88.76
3	Carr (1995a)	25	0.002709	0.0026273	0.0026273	94.5
4	Carr (1995a)	70	0.009218	0.0087600	0.0087600	90.4

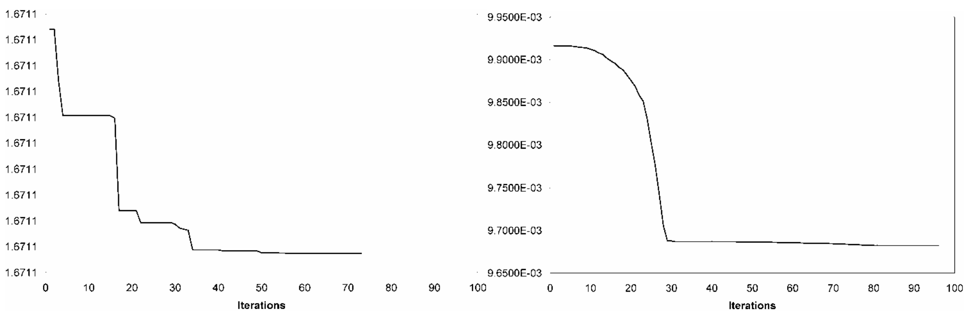
Since the objective of all minimum-zone formulations is to find the minimal separation between the two ideal features while encompassing all the points, a lower separation value indicates a more accurate assessment of the error. It can be seen from Tables 1–5 that the results obtained using PSO are lower than or equal to other published results. Thus, it can be inferred that using the PSO ensures that the solution does not get stuck in local minima.

The average number of iterations is a good indicator of the computation time involved in the procedure. The average computation time for all the tested datasets ranged between 0.1 and 2 sec. It can also be observed that the number of sample points does not significantly affect the number of iterations required to reach the global optimum. It was observed that the algorithm never terminated due to the maximum number of iteration criterion. This clearly indicates the robustness of the algorithm and its capability to converge to the optimal solution without being affected by the random initialisation.

Comparison of the results over multiple iterations also showed that the algorithm consistently found the same result, despite having random starting points, which indicates that the method is not easily trapped in local minima.

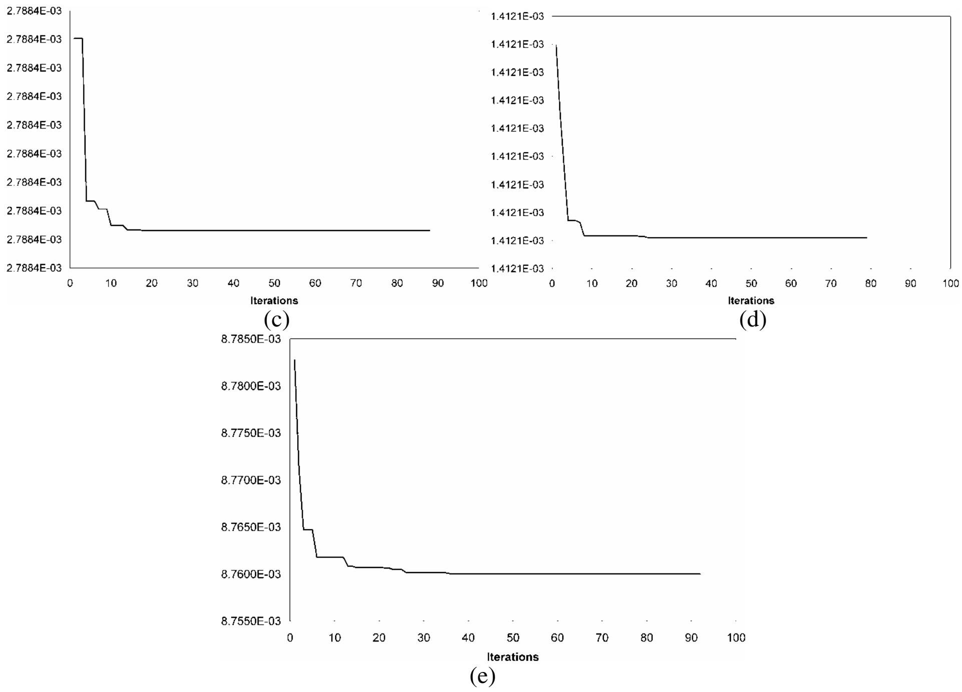
Figure 7 shows the convergence trend of the PSO algorithm for all five form tolerances. It can be seen that, in each case, the algorithm converges in less than 100 iterations. It can also be observed that the algorithm reaches the optimum value after approximately 40 iterations or less. The additional iterations only ensure that a better solution does not exist in the path of any of the particles.

Figure 7 Convergence trend of algorithm for all form tolerances (a) Circularity (b) Sphericity (c) Cylindricity (d) Straightness (e) Flatness



(a) (b)

Figure 7 Convergence trend of algorithm for all form tolerances (a) Circularity (b) Sphericity (c) Cylindricity (d) Straightness (e) Flatness (continued)



6 Conclusions and future direction

A unified EA-based approach is presented to solve the minimum zone form tolerance problem. The algorithm is accurate, and provides equal or better results as compared to other published results. It is also extremely robust, and performs well for small and large datasets. The computational efficiency of the method is also good because it only involves calculation of distances.

The present implementation of PSO is also insensitive to the starting points for the particles. The performance of the algorithm can potentially be improved by developing better heuristics to initialise the particles. Also, since the parameters are not controlled adaptively, convergence to the global optimum takes a considerable time, though the swarm gets to the neighbourhood of the global optimum position very quickly.

Future work in this area could concentrate on using methods to map the function space to a different domain in order to reduce the non-linearity and obtain a smoother function surface. Other areas of research could involve investigating the potential of adaptively changing the swarm parameters to enhance the performance of the swarm.

Acknowledgements

The authors acknowledge the support provided by the University of Cincinnati through the PACE Centre for Global Design and Manufacturing to Hemant Ramaswami, Yashpal Kovvur and Raj Bardhan Anand.

References

Abido, M.A. (2002) 'Optimal power flow using particle swarm optimization', *Electrical Power and Energy Systems*, Vol. 24, pp.563–571.

American National Standards Institute (1994) *ANSI/ASME Y14.5.1M-1994 (1994) Standard on Dimensioning and Tolerancing*, New York, American National Standards Institute.

Barcenas, C.C. and Griffin, P.M. (2000) 'Parametric based determination of cylindrical variations in geometric tolerancing', *Transactions of ASME, Journal of Manufacturing Science and Engineering*, Vol. 122, pp.549–555.

Burkardt, J. (2005) 'Matlab code for Sobol Quasirandom Sequence', Available at: http://www.scs.fsu.edu/~burkardt/m_src/sobol/sobol.html.

Carr, K. and Ferreira, P. (1995a) 'Verification of form tolerances Part I: basic issues, flatness, and straightness', *Precision Engineering*, Vol. 17, pp.131–143.

Carr, K. and Ferreira, P. (1995b) 'Verification of form tolerances Part II: cylindricity and straightness of a median line', *Precision Engineering*, Vol. 17, pp.144–156.

Chen, C-K. and Liu, C-H. (2000) 'A study on analyzing the problem of the spherical form error', *Precision Engineering*, Vol. 24, pp.119–126.

Cheraghi, S.H., Lim, H.S. and Motavalli, S. (1996) 'Straightness and flatness tolerance evaluation: an optimization approach', *Precision Engineering*, Vol. 18, pp.30–37.

Clerc, M. and Kennedy, J. (2002) 'The particle swarm – explosion, stability and convergence in a multidimensional complex space', *IEEE Transactions on Evolutionary Computation*, Vol. 6, No. 1, pp.58–73.

Dhanish, P.B. and Shunmugam, M.S. (1991) 'An algorithm for form error evaluation – using the theory of discrete and linear chebyshev approximation', *Computer Methods in Applied Mechanics and Engineering*, Vol. 92, pp.309–324.

Hodgson, T.J., Kay, M.G. and Mittal, R.O. (1999) 'Evaluation of cylindricity using combinatorics', *IIE Transactions*, Vol. 31, pp.39–47.

Hong, L.J., Shultes, B.C. and Anand, S. (2001) 'Robust evaluation of straightness and flatness tolerance using simulated annealing', *Transactions of the North American Manufacturing Research Institution of SME*, pp.533–540.

Huang, J. (1999a) 'An exact minimum zone solution for sphericity evaluation', *Computer-Aided Design*, Vol. 31, pp.845–853.

Huang, J. (1999b) 'An exact solution for the roundness evaluation problems', *Precision Engineering*, Vol. 23, pp.2–8.

International Organization for Standardization (2004) *ISO 1101-2004: Geometrical Product Specifications (GPS) – Geometrical Tolerancing – Tolerances of form, Orientation, Location and Run-out*, Geneva, Switzerland, International Organization of Standardization.

Jywe, W-Y., Liu, C-H. and Chen, C-K. (1999) 'The min-max problem for evaluating the form error of a circle', *Measurement*, Vol. 26, pp.273–282.

Kanada, T. (1997) 'Estimation of sphericity by means of statistical processing for roundness of spherical parts', *Precision Engineering*, Vol. 20, No. 2, pp.117–122.

Kennedy, J. and Eberhart, R. (1995) 'Particle swarm optimization', *Proceedings of IEEE International Conference on Neural Networks*, pp.1942–1948.

Lai, J. and Chen, J. (1996) 'Minimum zone evaluation of circles and cylinders', *International Journal of Machine Tools and Manufacture*, Vol. 36, pp.435–451.

Lai, H-Y., Jywe, W-Y., Chen, C-K. and Liu, C-H. (2000) 'Precision modeling of form errors for cylindricity evaluation using genetic algorithms', *Precision Engineering*, Vol. 24, pp.310–319.

Murthy, T.S.R. and Abdin, S.Z. (1980) 'Minimum zone evaluation of surfaces', *International Journal of Machine Tool Design Research*, Vol. 20, pp.123–136.

Parsopoulos, K.E. and Vrahatis, M.N. (2001) 'Modification of the particle swarm optimizer for locating all the global minima', *Artificial Neural Networks and Genetic Algorithms*, Springer (Wien), Computer Science series, pp.324–327.

Parsopoulos, K.E. and Vrahatis, M.N. (2002) 'Initializing the particle swarm optimizer using the nonlinear simplex method', in A. Grmela and N.E. Mastorakis (Eds). *Advances in Intelligent Systems, Fuzzy Systems, Evolutionary Computation WSEAS Press*, pp.216–221.

Rajagopal, K. and Anand, S. (1999) 'Assessment of circularity error using a selective data partition approach', *International Journal of Production Research*, Vol. 37, No. 17, pp.3959–3979.

Ramaswami, H., Kovvur, Y. and Anand, S. (2005) 'Minimum-zone circularity evaluation using particle swarm optimization', *Proceedings of 2005 ASME International Mechanical Engineering Congress and Exposition*.

Roy, U. and Zhang, X. (1992) 'Establishment of a pair of concentric circles with the minimum radial separation for assessing roundness error', *Computer-Aided Design*, Vol. 24, No. 3, pp.161–168.

Samuel, G.L. and Shunmugam, M.S. (1999) 'Evaluation of straightness and flatness error using computational geometric techniques', *Computer-Aided Design*, Vol. 31, pp.829–843.

Samuel, G.L. and Shunmugam, M.S. (2002) 'Evaluation of sphericity error from form data using computational geometric techniques', *International Journal of Machine Tools and Manufacture*, Vol. 42, pp.405–416.

Samuel, G.L. and Shunmugam, M.S. (2003) 'Evaluation of circularity and sphericity from coordinate measurement data', *Journal of Materials Processing Technology*, Vol. 139, pp.90–95.

Saravanan, R., Siva Sankar, R., Asokan, P., Vijayakumar, K. and Prabhaharan, G. (2005) 'Optimization of cutting conditions during continuous finished profile machining using non-traditional techniques', *International Journal of Advanced Manufacturing Technology*, Vol. 26, pp.30–40.

Schutte, J.F. and Groenwold, A.A. (2003) 'Sizing design of truss structures using particle swarms', *Structural Multidisciplinary Optimization*, Vol. 25, pp.261–269.

Sharma, R., Rajagopal, K. and Anand, S. (2000) 'A genetic algorithm based approach for robust evaluation of form tolerances', *Journal of Manufacturing Systems*, Vol. 19, pp.46–57.

Shi, Y. and Eberhart, R.C. (1998) 'Parameter selection in particle swarm optimization', *Evolutionary Programming VII: Proceedings of the 7th Annual Conference on Evolutionary Programming*, New York, pp.591–600.

Shunmugam, M.S. (1986) 'On assessment of geometric errors', *International Journal of Production Research*, Vol. 24, No. 2, pp.413–425.

Van den Bergh, F. and Engelbrecht, A.P. (2001) 'Effects of swarm size on cooperative particle swarm optimisers', *Proceedings of the Genetic and Evolutionary Computation Conference 2001*, San Francisco, USA.

Weber, T., Motavalli, S., Fallahi, B. and Cheraghi, S.H. (2002) 'A unified approach to form error evaluation', *Precision Engineering*, Vol. 26, pp.269–278.

Zhao, J.W. and Chen, G.Q. (2005) 'Roundness error measurement based on particle swarm optimization', *Journal of Physics: Conference Series*, Vol. 13, pp.261–263.

Zhu, L.M., Ding, H. and Xiong, Y-L. (2003) 'A steepest descent algorithm for circularity evaluation', *Computer-Aided Design*, Vol. 35, pp.255–265.

Zhu, L.M. and Ding, H. (2003) 'Application of kinematic geometry to computational metrology: distance function based hierarchical algorithms for cylindricity evaluation', *International Journal of Machine Tools and Manufacture*, Vol. 43, pp.203–215.