MEASUREMENT SCIENCE REVIEW, **17**, (2017), No. 4, 187-196

DE GRUYTER OPEN

ISSN 1335-8871

MEASUREMENT SCIENCE REVIEW

Journal homepage: http://www.degruyter.com/view/j/msr

Evaluating Geometric Characteristics of Planar Surfaces using Improved Particle Swarm Optimization

Vimal Kumar Pathak[1], Sagar Kumar[2], Chitresh Nayak[3], NRNV Gowripathi Rao[4]

[1]*School of Automobile, Mechanical and Mechatronics Engineering, Manipal University Jaipur, Rajasthan – 303007, India, vimalpthk@gmail.com*

[2]*Department of Mechanical Engineering, Poornima College of Engineering, Jaipur, Rajasthan – 302022, India*

[3]*Department of Mechanical Engineering, Kautilya Institute of Technology & Engineering, Jaipur, Rajasthan – 302022, India*

[4]*Department of Farm Machinery and Power Eng., College of Technology and Engineering, MPUAT, Udaipur-313001, India*

This paper presents a modified particle swarm optimization (MPSO) algorithm for the evaluation of geometric characteristics defining form and function of planar surfaces. The geometric features of planar surfaces are decomposed into four components; namely straightness, flatness, perpendicularity, and parallelism. A non-linear minimum zone objective function is formulated mathematically for each planar surface geometric characteristic. Finally, the result of the proposed method is compared with previous work on the same problem and with other nature inspired algorithms. The results demonstrate that the proposed MPSO algorithm is more efficient and accurate in comparison to other algorithms and is well suited for effective and accurate evaluation of planar surface characteristics.

Keywords: Planar surfaces, geometric features, inspection, improved particle swarm optimization.

1. INTRODUCTION

Planar features are the most basic primitive elements of mechanical parts. The utmost elementary geometric characteristics that are used to control form and function of planar features are straightness, flatness, perpendicularity, and parallelism [1]. During manufacturing of the part as per the drawing specifications, significant errors are developed in the form of these characteristics. For proper functioning of the parts and assemblies, it is essential to provide tolerances on the features that are functional, regardless of variation in their form. Accurate measurement of the aforesaid errors is crucial to conform to the tolerance specification. In general practice, sometimes it becomes impractical to acquire variation over the whole surface. Consequently, only finite points are taken which represent features of the surface and these points are sufficient for evaluation of form errors. Earlier, coordinate measuring machines (CMM) were widely used for acquiring 3D cloud points and off-line and on-line inspection activities [2]-[4].

Least square method (LSM) is used technique for these geometric characteristics in industries because of its simplicity in computation and uniqueness in solution. However, LSM does not adhere to the standards and will not guarantee the minimum zone solution as specified by standards which may lead to overestimation of tolerances and ultimately leading to rejection of good parts [5]. To replace LSM, several algorithms have been suggested and the majority of them follow the minimum zone principle. Wang et al. [6] presented a generalized non-linear optimization procedure for circularity evaluation based on minimal radial separation criterion. Cheraghi et al. [7] proposed criteria based on the least square cylinder, minimum circumscribed cylinder, and maximum inscribed cylinder for evaluation of cylindricity error. Endrias and Feng [9] formulated the objective function which is a function of the rigid body coordinate transformation parameters. A standard direct search algorithm and downhill simplex search algorithm are employed to minimize the form tolerance objective function. Carr and Ferreira [10] formulate straightness, flatness, and cylindricity as non-linear problems, which were then transformed into a series of linear problems.

Venkaiah and Shunmugam, [11]-[12], introduced distinctive optimization algorithms such as numerical and computational geometry optimization approaches that are used for evaluation of circularity and cylindricity. Seun and Chang [13] developed an interval bias linear neural based approach with least mean squares learning algorithm for straightness and flatness evaluation and analysis. Weber et al. [8] propounded a unified linear approximation technique for use in evaluating the form errors. The non-linear equation for individual form was linearized implementing

DOI: 10.1515/msr-2017-0022

$$d = \frac{|c_1 - c_2|}{\sqrt{1 + m^2}} \quad (1)$$

Taylor expansion and it was solved using a linear program. Although, numerical approaches are ubiquitous methods to solve optimization problems and they are also computationally efficient, they may lead to inaccurate results due to mathematical approximations. On the contrary, some of the nature inspired optimization algorithms have been used for form tolerance evaluation and they include genetic algorithm (GA), ant colony optimization (ACO), particle swarm optimization (PSO), and artificial bee colony (ABC) [14]-[17]. GA was to be more complex than PSO in the principle for the same work [3], [18]. ACO is time consuming and convergence time is also uncertain. ABC has slow convergence rate, easy to fall in local optimum and difficult to find best out of available feasible solutions.

PSO has been widely used to solve continuous problems due to the simplicity of concept and fewer parametric settings than other population based optimization algorithms [19]-[22]. However, classical PSO still has some disadvantages, such as weak local search ability, and may lead to entrapment in local minimum solutions that affects the convergence performance and results in uncertainties in the results obtained. In PSO, updating of new solution is performed only over the existing one without comparing which one is better. This is considered to be caused due to the lack of exploitation capability in classical PSO, which makes it hard to find the best possible solutions [23]. To improve the exploitation capability, a modified particle swarm optimization (MPSO) is proposed for effective form error evaluation, based on the generation of new improved position using the difference in the global and local best positions. The results of proposed algorithm for geometric error evaluation were compared with previous literature and other nature inspired algorithms which confirm the effectiveness of the modified PSO.

2. MATHEMATICAL FORMULATION

The most basic geometric features of planar surfaces contribute significantly to various mechanical products such as rotational parts, assembly part, and injection molds to achieve the desired functionalities. Numerous mechanical components depend on small form error to have adequate performance.

Minimum zone straightness formulation

By measuring a line element of a surface, the measured data points obtained are represented as $D_I(x_i, y_i)$ where $(i = 1,2,3 \ldots n)$. Then, the minimum zone solution of straightness error is calculated by finding two parallel lines minimally distant from each other that enclose all data points, which also defines the smallest feasible region. These lines are represented by $y = mx + c_1$and $y = mx + c_2$, where m, c_1 and c_2 are coefficients. If x and y coordinates are known then c_1 and c_2 become a function of m, where m is the slope of line. Now, the shortest distance, d between these two lines can be calculated by:

The above equation is written in the form of $h(m) = d_{max} - d_{min}$ i.e., straightness error as:

$$d = \frac{\max(y_i - mx_i) - \min(y_i - mx_i)}{\sqrt{1 + m^2}} \quad (2)$$

The distance d, between two parallel lines is a function of m. Now, the minimum zone straightness error objective/fitness function can be expressed as:

$$f(m) = \min(\frac{\max(y_i - mx_i) - \min(y_i - mx_i)}{\sqrt{(1 + m^2)}}) \quad (3)$$

The above objective function can be represented in vectorial form as below:

$$\boldsymbol{X} = f(\boldsymbol{m})$$

where (x_i, y_i, z_i) are 3D point data measured by CMM. The above objective function is a function of m. Accordingly, using PSO and its proposed variant, m is calculated for which the value of the above expression is minimum.

Minimum zone flatness formulation

For calculating the minimum zone flatness error, the two parallel planes are represented by $z = mx + by + c_1$ and $z = mx + by + c_2$, where x, y, z are coordinates and m, b, c_1 and c_2 are coefficients. Similar to straightness, the flatness error can be represented as:

$$\frac{\max(z_i - mx_i - by_i) - \min((z_i - mx_i - by_i)}{\sqrt{(1 + m^2 + b^2)}} \quad (4)$$

where x, y and z are coordinates of point data and m and b are the optimization variables. So, the objective/fitness function for minimum zone flatness error is

$$f(m, b) = \min(\frac{\max(z_i - mx_i - by_i) - \min(z_i - mx_i - by_i)}{\sqrt{(1 + m^2 + b^2)}}) \quad (5)$$

The above objective function can be represented in vectorial form as below:

$$\boldsymbol{X} = f(\boldsymbol{m}, \boldsymbol{b})$$

This is a function of m and b. Consequently, for solving the above objective function by searching the value of m and b for which the objective function $f(m, b)$ is minimum.

Minimum zone perpendicularity formulation

According to ISO [24], perpendicularity can be measured by finding two parallel lines that are perpendicular to datum, minimum distance apart containing the whole data points. Assuming all the measured data points $D_I(x_i, y_i)$ where $(i = 1,2,3 \ldots . n)$ lie between the two parallel lines minimally apart as shown in Fig.1. The two parallel lines signifies the minimum tolerance value within which all data points must fall. The minimum zone method for perpendicularity is defined by the minimum actual datum for planar surfaces. Assuming actual datum line equation can be expressed as:

$$y = mx + c \tag{6}$$

The distance d_i between the measured points $D_I(x_i, y_i)$ of datum line and actual datum line can be expressed as:

$$d_i = \frac{y_i - mx_i - c}{\sqrt{1+m^2}} \tag{7}$$

where n is the number of points measured for defining the datum line. Further, the minimum zone objective function for datum line can be expressed as an unconstrained optimization problem.

$$\text{Min } f(m,c) = \text{max. } (d_i) \tag{8}$$

The above objective function can be represented in vectorial form as below:

$$\boldsymbol{X} = f\,(\boldsymbol{m}, \boldsymbol{c})$$

Now suppose the actual datum line based on optimal solution $Z(m^*, c^*)$ obtained is based on the equation below:

$$y = m^*x + c^* \tag{9}$$

After the establishment of actual datum lines, draw line passing through the earlier measured point $D_I(x_i, y_i)$ which will be perpendicular to the actual datum. This equation of line can be written by taking a_i as intercept of the lines along the y-axes:

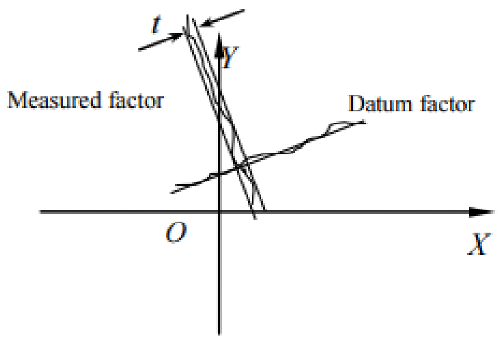


Fig.1. Schematic for determining perpendicularity.

$$a_i = m^*x_i + y_i \tag{10}$$

Now, let the length of line in y axis intercepted by two lines with maximal and minimal intercept of above lines be L,

$$\text{L} = a_{imax} - a_{imin} \tag{11}$$

As the direction cosines can be written in form:

$$\cos\alpha = \frac{m^*}{\sqrt{1+m^{*2}}} \tag{12}$$

So, final perpendicularity error equation can be expressed considering the direction cosine as

$$f = L\frac{m^*}{\sqrt{1+m^{*2}}} \tag{13}$$

The above objective function can be represented in vectorial form as below:

$$\boldsymbol{X} = f\,(\boldsymbol{m}^*, \boldsymbol{L})$$

Minimum zone parallelism formulation

As per the ISO definition, parallelism can be defined by measuring two parallel lines with minimal distance apart and parallel to a defined datum as shown in Fig.2. Assuming all the measured data points $D_I(x_i, y_i)$ where $(i = 1,2,3 \ldots . n)$ lie between the two parallel lines minimally apart. The two parallel lines are referred to as smallest feasible region within which all points must fall. Based on the minimum zone method, the assumed actual datum line equation can be expressed as:

$$y = mx + c \tag{14}$$

The distance d_i between the measured points $D_I(x_i, y_i)$ of datum line and actual datum line can be expressed as:

$$d_i = \frac{y_i - mx_i - c}{\sqrt{1+m^2}} \tag{15}$$

where n is the number of points measured for defining the datum line. Further, the minimum zone objective function for datum line can be expressed as an unconstrained optimization problem.

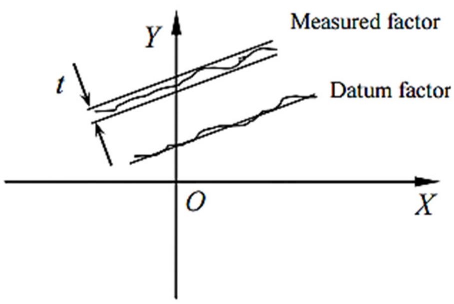


Fig.2. Schematic for determining parallelism error.

$$\text{Min } f(m,c) = \text{max. } (d_i) \tag{16}$$

The above objective function can be represented in vectorial form as below:

$$\boldsymbol{X} = f\,(\boldsymbol{m}, \boldsymbol{c})$$

Similar with case of perpendicularity, the actual datum line based on optimal solution $Z(m^*, c^*)$ obtained is based on the equation below:

$$y = m^*x + c^* \tag{17}$$

The distance d_i^* between the measured points $D_I(x_i, y_i)$ of surface measured and actual datum line can be expressed as:

$$d_i = \frac{y_i - m^*x_i - c^*}{\sqrt{1 + m^{*2}}} \tag{18}$$

The minimum objective function for minimum zone parallelism can be expressed as:

$$f = \min(\max(d_i) - \min(d_i)) \tag{19}$$

The above objective function can be represented in vectorial form as below:

$$\boldsymbol{X} = f\,(\boldsymbol{m^*}, \boldsymbol{c^*})$$

3. MODIFIED PARTICLE SWARM OPTIMIZATION ALGORITHM

This section describes the proposed modified variant of the classical particle swarm optimization algorithm. The exploitation ability directly influences the quality of results, as it is an essential property for any swarm based heuristic optimization technique. The modified variant will help in overcoming the classical PSO drawback of slow convergence due to lack in exploitation abilities.

Standard particle swarm optimization algorithm

The basic particle swarm optimization is a population based method suggested by Kennedy and Eberhart in 1995. PSO is modeled after the simulation of social behavior of birds in a flock [25]-[26]. PSO is initialized by distributing each particle randomly in a D-dimensional search space. The performance of each particle is measured using a fitness or objective function which depends on the optimization problem. Each particle k is represented by the following information:

- x_k, the current position of the particle k
- v_k, current velocity of the particle k
- p_k, personal best position of the particle k
- g_k, global best position of the particle k

The personal best position signifies the best position that particle k has been at so far. The fitness or objective function is defined by eqns. (3), (5), (13), and (19) and lowest for that position of the k^{th} particle. Here, velocity v_k acts like a vector which helps in guiding the particle from one position to another with updated velocity and position at every iteration. The below equation is divided into three parts. First is inertia part described by $w \cdot v_k(t)$, used for providing motion to the algorithm. Second part is cognitive component$rand[0,1] \cdot (p_k(t) - x_k(t))$, which is based on individual knowledge and experience. The third and last part $rand[0,1] \cdot (g_k(t) - x_k(t))$, is known as social component based on individual interaction with their neighbors. New position and velocity for k^{th} particle is updated at every iteration and expressed as:

$$v_k(t+1) = v_k(t) + c_1 rand[0,1](p_k(t) - x_k(t)) + c_2 rand[0,1](g_k(t) - x_k(t)) \tag{20}$$

$$x_k(t+1) = x_k(t) + v_k(t+1) \tag{21}$$

$rand[0,1]$ and $rand[0,1]$ are two statistically independent and uniformly distributed random numbers within the given interval [0,1]. The acceleration coefficients c_1 and c_2 are also important parameters in PSO. c_1 pulls the particle towards the local best position whereas c_2 pulls the particle towards the global best and the sum of these two should be greater than 4 and less than 4.2 ($4 \leq (c_1 + c_2) \leq 4.2$) [27]. So, for balancing exploration and local convergence, the value of c_1 and c_2 is taken 2 each. $p(t)$ is the best position parameter of an individual particle and $g(t)$ is global best position parameter of entire swarms. Shi and Eberhart [28] introduced an inertia weight w into the velocity updating of the PSO that helps in controlling the scope of the search. Often, w decreases linearly from 0.9 to 0.4 over the whole iteration. Here, whole iteration is the maximum iteration needed to get the final result. The velocity updating with inertia weight is shown in (22).

$$v_k(t+1) = wv_k(t) + c_1 rand[0,1](p_k(t) - x_k(t)) + c_2 rand[0,1](g_k(t) - x_k(t)) \tag{22}$$

The different steps of basic PSO are as follows:

Step 1: Define the PSO parameters and randomly generate a population with initial position ($x_k = x_{k1}, x_{k2}, \ldots x_{kD}$) and velocity ($v_k = v_{k1}, v_{k2}, \ldots v_{kD}$) of all the particles in the entire search space.

Step 2: Evaluate the objective (fitness) function (f_0) of each particle according to eqns. (3), (5), (13), and (19) for each form error. The lower the objective function value is, the better the corresponding particle performs.

Step 3: Update or change the velocity and position of each particle according to relative positions from local best ($pbest$) and global best ($gbest$) using eqns. (21) and (22).

Step 4: Apply boundary constraints on design variables so that the value of design variables lies within the lower bound (LB) and upper bound (UB) and particle does not fly outside the search space.

$$\begin{aligned} &if\ \ x(k,j) < LB(j); \quad x(k,j) = LB(j); \\ &else\ if\ \ x(k,j) > UB(j); \quad x(k,j) = UB(j) \end{aligned}$$

Step 5: Again, fitness function for each particle is calculated using the same eq. (3), (5), (13), and (19). If the current objective function value is less than the previous $pbest$ value then $pbest$ is replaced by the current position.

Step 6: If the current objective function value is less than the previous $gbest$ value then $gbest$ is replaced by the current position.

Step 7: The termination criterion is checked and if it is not met, go back to step 3. The termination criterion could be either max. iteration or good objective or fitness value.

It is observed from the above steps that basic PSO performs exploration in step 3 using equation (21) and (22) by generating new solutions in the search space. However, the exploitation part is seen nowhere in the algorithm, as selection mechanism is missing in PSO. In PSO, only updating of new solution takes place without comparing which one is better. So, basic PSO has only explorative tendency and it lacks the exploitation ability. Therefore, in order to overcome this limitation a modified PSO algorithm is presented here.

Modified particle swarm optimization (MPSO) algorithm

A new variant of PSO is proposed in this paper for the effective form error evaluation. The exploration and exploitation capabilities are two important factors that are considered during the design of an optimization algorithm. Exploitation refers to the use of existing information whereas exploration means generation of new solution in the search space. In PSO, an old solution is replaced by the new one. To overcome all these problems, the modified variant of PSO algorithm generates new swarm position and fitness solution based on the new search equations (23) and (24):

$$v_{new} = p_{best} + rand[0,1](g_{best} - p_{best}) \tag{23}$$

$$x_{new} = p_{best} + v_{new} \tag{24}$$

where p_{best} is the particle best position, g_{best} is the particle global best position. $rand[0,1]$ is the random number generator between 0 and 1 that controls the rate at which the population evolves. The random number generator typically is initialized by this parameter, allowing to yield different values at each trial. The best solutions in the current population are very useful sources that can be used to improve the convergence performance. Also, Eqn. (23) can drive the new candidate solution only around the best solution of the previous iteration. Therefore, the proposed search and updated equations described can increase the exploitation capability of the classical PSO.

Any selection strategy in the algorithm is usually considered as exploitation, as the fitness solution of the individual is used to determine whether or not an individual should be exploited. Therefore, the MPSO particle swarms employ greedy selection procedure among two parallel fitness functions to update the best candidate solution which also helps in improving the exploitation ability of the algorithm. The flowchart of the proposed modified PSO algorithm is shown in Fig.3.

MPSO begins with step 1 of basic PSO algorithm and remains the same till step 5. Afterwards, an additional path for generating new solution by position and velocity updating is introduced in the algorithm using equation (24) and (24). This additional path will provide an extra option for velocity and position updating besides the basic updating used in PSO, providing new objective function (f_1). Both paths run independently for each iteration. The best particle with minimum fitness or objection function will be chosen for the next iteration using greedy selection procedure. A greedy selection scheme is used for selection of the best solution among two possible solutions (the new solution and the old one) and the better one is preferred for inclusion in population based on the fitness or objective function value. In this way, the information of a good particle of the population is distributed among the other particles due to the greedy selection scheme applied, and thus, enhancing the exploitation ability of the algorithm. Further, the final objective function is updated as f_2 with corresponding position of the best particle and is used in the next iteration. At last, the termination criterion is checked and if it is not met, go back to step 3.

4. EXPERIMENTAL IMPLEMENTATION

To test the robustness and efficiency of the proposed MPSO algorithm, various examples from literature are taken for evaluating the geometric characteristics of planar surfaces. A set of data points are taken from literature [29]-[30] for possible solutions (as in this case minimization of form error). However, the data for perpendicularity and parallelism are measured using touch probe CMM. As GA, PSO and MPSO algorithms are stochastic in nature, consequently the results are not repeatable. For the aforesaid reason, all algorithms are run 25 times independently with similar parameters to evaluate these datasets. Further, average of these 25 datasets are taken for providing reliable estimate of the accuracy in results. The algorithm is programmed and implemented in MATLAB R2014a. The parameters used for PSO and MPSO optimization techniques are shown in Table 1.

Table 1. Parameters used for PSO and MPSO.

S. No.	PSO and MPSO parameters
1	Swarm Size: 50
2	Maximum Number of iterations: 100
3	c1, c2 = 2.05, 205
4	w_{start}, w_{end} = 0.9, 0.4

Practical examples (straightness)

For the purpose of comparison, four examples available in literature [29] are selected. The real data points measured using CMM for straightness evaluation are shown in Appendix A with allowable tolerance of 0.00165 inch. Table 2 shows the results presented in literature [30] along with the solution provided by the proposed MPSO algorithm. For example 1, it is observed that minimum zone straightness error obtained by LSM is 0.0017, Optimization Technique Zone (OTZ) [8] is 0.0017, Linear Approximation Technique (LAT) [8] is 0.0017, GA [3] is 0.001672, and PSO [29] is 0.001711, while the minimum zone straightness error obtained by the proposed MPSO is 0.00160. If the allowable straightness tolerance is 0.00165 inch, all the algorithms

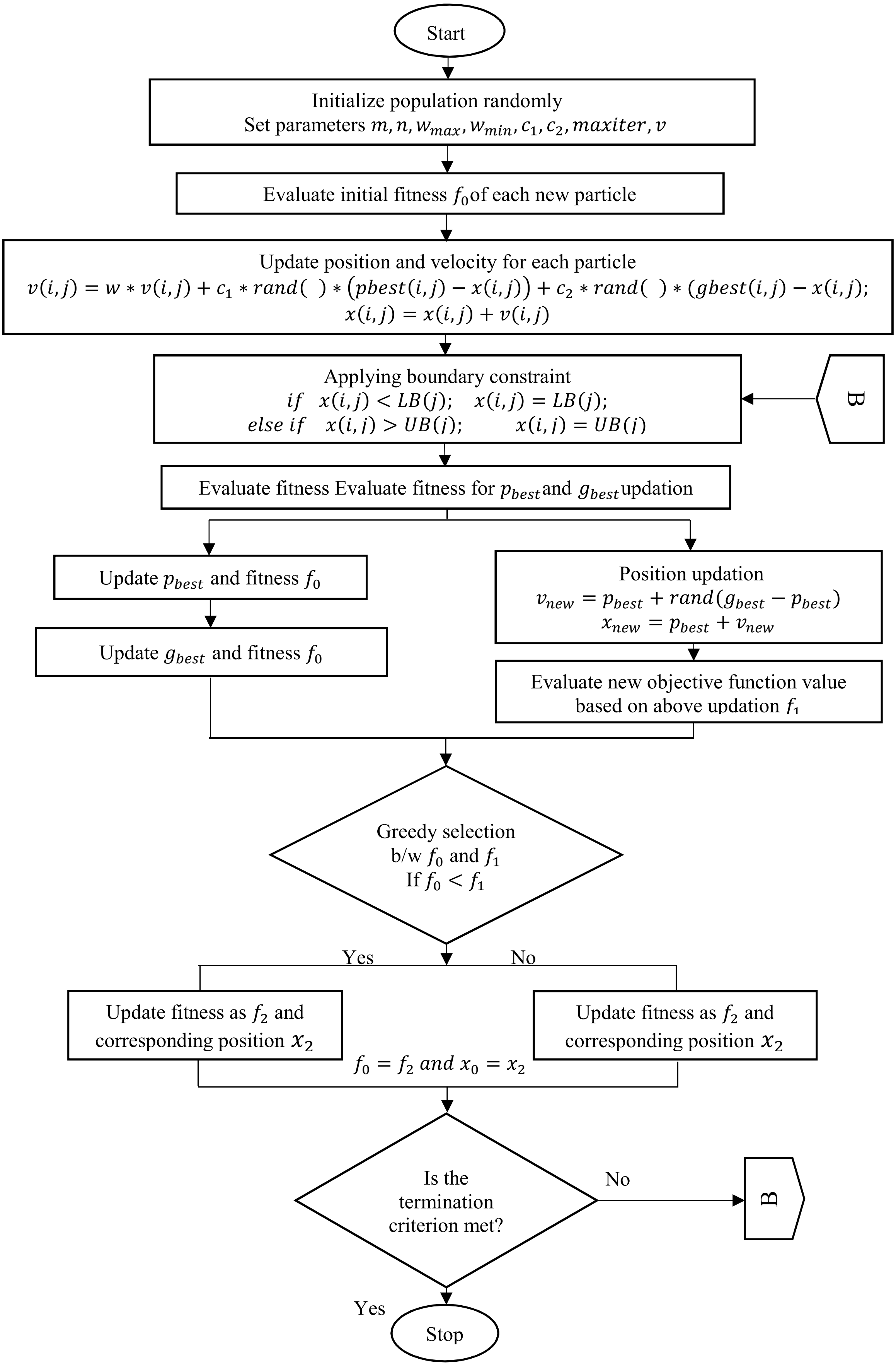


Fig.3. Flowchart of modified particle swarm optimization (MPSO) algorithm.

except the MPSO algorithm overestimate the tolerances and hence result in rejection of good parts. This signifies the importance of the proposed algorithm in accurate evaluation of minimum zone tolerance and also helps in preventing the rejection of good part based on product specifications. This will further help in minimizing the economic loss occurring in manufacturing of the part. The result shows that MPSO algorithm has higher computational accuracy and its optimization result surpassed those from the other methods [3], [8], [29] and from LSM. The iterative curve for PSO and MPSO is shown in Fig.4.a), Fig.4.b) confirming better performance and efficiency of the proposed MPSO algorithm.

Practical examples (flatness)

The sampling data available in literature [30] are selected as shown in Appendix B with 25 data points for each measurement. A plane part with length and width of 140 mm and 120 mm, respectively, is considered with allowable tolerance of 0.018 mm. For part inspection, it is important to follow an appropriate sampling strategy. The sampling strategy suggests selection of exact location for each measurement point. Two sampling data sets are taken which means location of points is the same for both measurements. The results for flatness error evaluation are tabulated in Table 3. It is observed that the minimum zone flatness error obtained by the proposed MPSO for 2 times sampling are 0.0174 and 0.0178, respectively, with a mean of 0.0176. The result is of practical significance as the allowable maximum tolerance is 0.018 mm, with GA and PSO providing 0.0187 mm tolerance. On the contrary, the result of MPSO is 0.0176 mm, which is under the allowable tolerance limit. This result shows that the good part may get rejected if LSM, GA and PSO algorithm is used, due to overestimation of flatness. Also, it is well in agreement with the results reported in literature [30] and far better than those obtained by LSM. The iterative curves when making assessment of flatness for PSO and MPSO are shown in Fig.5.a), Fig.5.b).

Table 2. Results of straightness evaluation.

Ex	OTZ [8]	LAT [8]	GA [3]	PSO [29]	MPSO
1	0.0017	0.0017	0.001672	0.001711	0.001602
2	0.0014	0.0014	0.001428	0.001401	0.001395

Table 3. Results of flatness evaluation (mm).

Examples	LSM	Improved GA	PSO	MPSO
1st time sampling	0.0219	0.0184	0.0184	0.0174
2nd time sampling	0.0229	0.0189	0.0189	0.0178
Mean		0.0187	0.0187	0.0176

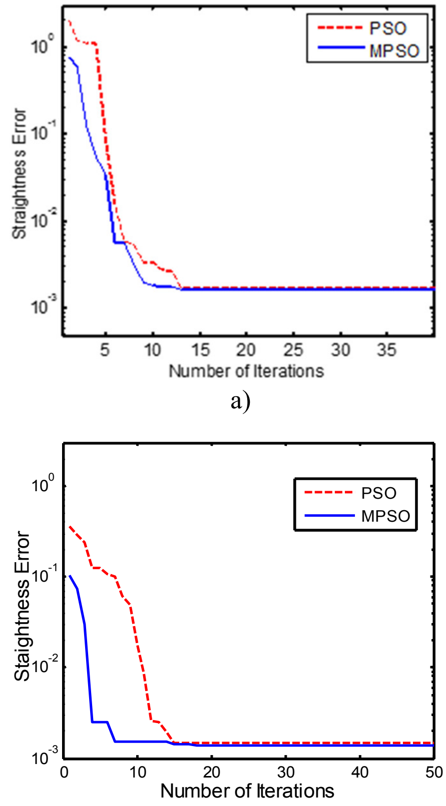


Fig.4. PSO and MPSO Convergence for straightness error.

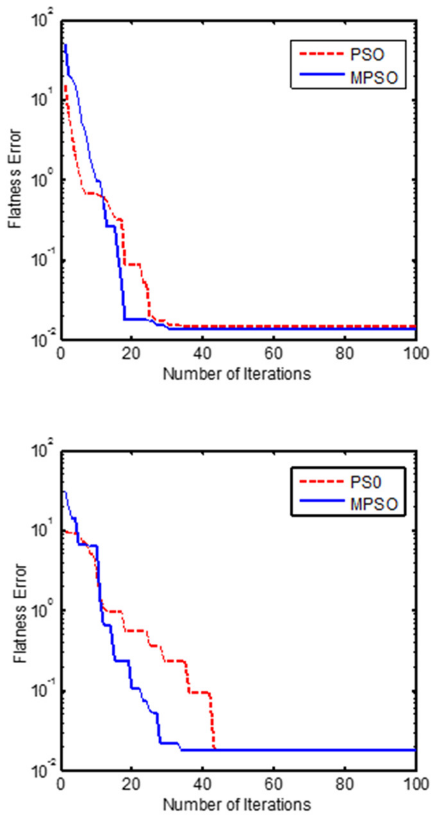
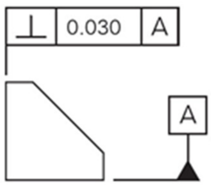


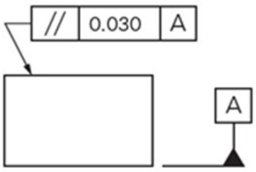
Fig.5. PSO and MPSO Convergence for flatness error.

Practical examples (perpendicularity and parallelism)

The test parts for the perpendicularity and parallelism error evaluation are shown in Fig.8.a), Fig.8.b). The coordinates of the datum are measured first and then the target surface is sampled using CMM with PC-DMIS software. The coordinates of measured data of datum A and the target surface for perpendicularity and parallelism are shown in Appendix C and D, respectively. The results for perpendicularity and parallelism error are tabulated in Table 4. It is observed that the minimum zone perpendicularity error obtained by the least square method (LSM), particle swarm optimization (PSO) and the proposed MPSO algorithm are 16.581 µm, 9.820 µm and 8.631 µm, respectively. The straightness error for datum line for LSM, PSO and MPSO is reported as 12.865 µm, 9.37 µm and 8.52 µm, respectively. It can be seen from the results that the perpendicularity error for the proposed MPSO algorithm shows better results than LSM and standard PSO. Similarly, for parallelism error, the MPSO algorithm outperforms the other mentioned methods.



a)



b)

Fig.6. Test parts for a) perpendicularity and b) parallelism evaluation.

Fig.7. shows the searching process of PSO and MPSO with iteration for the two geometric errors (i.e. perpendicularity and parallelism). Obviously, the convergence and optimization accuracy of MPSO is higher than standard PSO, which indicates that MPSO reaches to the optimum value earlier than standard PSO. The result is of practical significance as the allowable maximum tolerance is 0.010 mm for perpendicularity and 0.015 mm for parallelism, with LSM and PSO providing 0.016 mm (16.58 µm) and 0.011 mm (10.82 µm) tolerance, respectively. On the contrary, the result of MPSO is 0.0086 mm (8.63 µm), which is under the allowable tolerance limit as shown in Fig.7.a). This result shows that the good part may get rejected if LSM and PSO algorithm is used, due to overestimation of perpendicularity. Similarly, the MPSO algorithm obtained the parallelism error within the allowable tolerance of 0.015 mm as reported in Table 4. The proposed algorithm can significantly affect the inspection procedure as good parts get rejected if LSM and simple PSO are used.

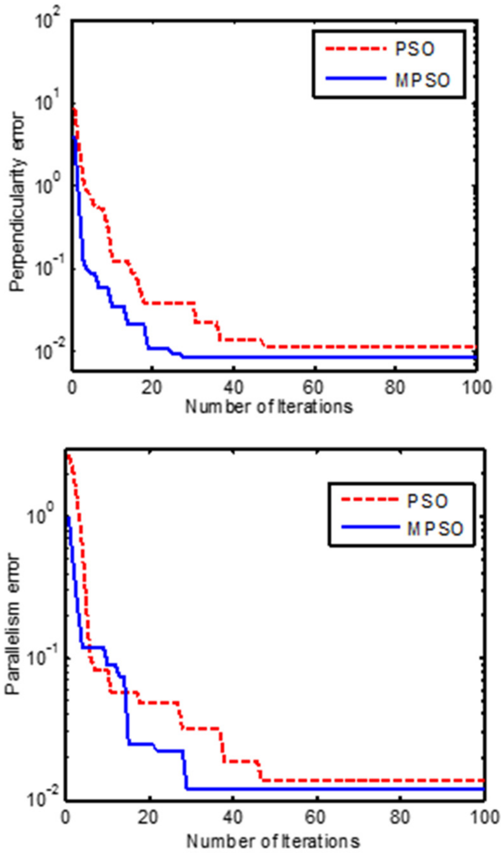


Fig.7. PSO and MPSO algorithm for perpendicularity and parallelism error.

Table 4. Perpendicularity and parallelism results (µm).

Method	Perpendicularity		Parallelism	
	Measured surface	Datum straightness	Measured surface	Datum straightness
LSM	16.581	12.865	15.983	11.760
PSO	10.820	10.37	13.212	10.232
MPSO	8.631	8.52	12.145	9.875

5. CONCLUSION

This paper presents a novel improved particle swarm optimization (MPSO) algorithm for geometric characteristics evaluation of the planar surfaces, which are in accordance with ISO 1101. The proposed algorithm overcomes the insufficiency of the classical PSO in terms of a weak exploitation behavior by introducing an improved solution search equation based on the best solution of the previous iteration. Additionally, a greedy selection procedure is added to further improve the exploitation ability of the classical PSO. A simple objective function for all geometric characteristics in planar surfaces was formulated as an unconstrained optimization problem. Numerical examples have been illustrated to verify geometric errors from coordinate data effectively. Compared to conventional or existing heuristics optimization methods, the proposed MPSO algorithm not only has the advantage of a simple realization in computers and good flexibility, but it was shown to have improved the geometric error evaluation accuracy. The implementation of the proposed MPSO algorithm can ensure that direct form error can be evaluated without any conversion. Consequently, this algorithm could be implemented for inspection and form error evaluation on CMMs.

REFERENCES

[1] American Society of Mechanical Engineers. (2009). *Y14.5 - 2009. Dimensioning and tolerancing*. Standard ASME Y14.5-2009.

[2] Wang, M., Xi, L., Du, S. (2014). 3D surface form error evaluation using high definition metrology. *Precision Engineering*, 38, 230-236.

[3] Cui, C., Li, B., Huang, F., Zhang, R. (2007). Genetic algorithm based form error evaluation. *Measurement Science and Technology*, 18, 1818-1824.

[4] Pathak, V.K., Singh, A.K., Sivadasan, M., Singh, N.K. (2016). Framework for automated GD&T inspection using 3D scanner. *Journal of the Institution of Engineers (India): Series C*, DOI 10.1007/s40032-016-0337-7.

[5] Samuel, G.L., Shunmugam, M.S. (2000). Evaluation of circularity from coordinate and form data using computational geometric techniques. *Journal of the International Societies for Precision Engineering and Nanotechnology*, 25, 251-263.

[6] Wang, M., Cheraghi, H., Masud, A.S. (1999). Circularity error evaluation theory and algorithm. *Precision Engineering*, 23, 164-176.

[7] Cheraghi, S.H., Jiang, G.H., Ahmad, J.S. (2003). Evaluating the geometric characteristics of cylindrical features. *Precision Engineering*, 27, 195-204.

[8] Weber, T. Motavalli, S., Fallahi, B., Cheraghi, S.H. (2002). A unified approach to form error evaluation. *Precision Engineering*, 26, 269-278.

[9] Endrias, D.H., Feng, H.Y. (2002). Minimum-zone form tolerance evaluation using rigid-body coordinate transformation. *Journal of Computing and Information Science in Engineering*, 3, 31-38.

[10] Carr, K., Ferreira, P. (1995). Verification of form tolerances part I: Basic issues, flatness, and straightness. *Precision Engineering*, 17, 131-143.

[11] Venkaiah, N., Shunmugam, M.S. (2007). Evaluation of form data using computational geometric techniques-Part I: Circularity error. *International Journal of Machine Tools and Manufacture*, 47, 1229-1236.

[12] Venkaiah, N., Shunmugam, M.S. (2007). Evaluation of form data using computational geometric techniques-Part II: Cylindricity error. *International Journal of Machine Tools and Manufacture*, 47, 1237-1245.

[13] Suen, D.S., Chang, C.N. (1997). Application of neural network interval regression method for minimum zone straightness and flatness. *Precision Engineering*, 20, 196-207.

[14] Liu, J., Wang, G., Pan, X. (2011). Minimum zone form tolerance evaluation for cylindrical surfaces using adaptive ant colony optimization. *Journal of Computational Information Systems*, 7, 4480-4490.

[15] Andrea, R., Michele, A., Matteo, B., Michele, L. (2011). Fast genetic algorithm for roundness evaluation by the minimum zone tolerance (MZT) method. *Measurement*, 44, 1243-1252.

[16] Wen, X.L., Huang, J.C., Sheng, D.H., Wang, F.L. (2010). Conicity and cylindricity error evaluation using particle swarm optimization. *Precision Engineering*, 34, 338-344.

[17] Luo, J., Wang, Q., Fu, L. (2012). Application of modified artificial bee colony algorithm to flatness error evaluation. *Optics and Precision Engineering*, 20, 422-430.

[18] Lai, H.Y., Jywe, W.Y., Chen, C.K., Liu, C.H. (2000). Precision modeling of form errors for cylindricity evaluation using genetic algorithm. *Precision Engineering*, 24, 310-319.

[19] Lee, C.T., Lee, C.C. (2014). On a hybrid particle swarm optimization method and its application in mechanism design. *Proceedings of the Institution of Mechanical Engineers, Part C: Journal of Mechanical Engineering Science*, 228, 2844-2857.

[20] Li, Z., Tian, G., Cheng, G., Liu, H., Cheng, Z. (2014). An integrated cultural particle swarm algorithm for multi-objective reliability-based design optimization. *Proceedings of the Institution of Mechanical Engineers, Part C: Journal of Mechanical Engineering Science*, 228, 1185-1196.

[21] Lin, C.H. (2016). An intelligent dynamic control of continuously variable transmission system using modified particle swarm optimization. *Proceedings of the Institution of Mechanical Engineers, Part C: Journal of Mechanical Engineering Science*, 230, 2181-2207.

[22] Tseng, K.Y., Zhang, C.B., Wu, C.Y. (2010). An enhanced binary particle swarm optimization for structural topology optimization. *Proceedings of the Institution of Mechanical Engineers, Part C: Journal of Mechanical Engineering Science*, 224, 2271-2287.

[23] Rao, R.V., Savsani, V.J. (2012). *Mechanical Design Optimization Using Advanced Optimization Techniques*. Springer.

[24] International Organization for Standardization. (1996). *Technical drawings—Geometrical tolerancing*. ISO/DIS 1101-1996.

[25] Kennedy, J., Eberhart, R.C. (1995). Particle swarm optimization. In *IEEE International Conference on Neural Networks*. IEEE, 1942-1948.

[26] Eberhart, R.C., Kennedy, J. (1995). A new optimizer using particle swarm theory. In *Proceedings of the Sixth International Symposium on Micro Machine and Human Science (MHS '95)*. IEEE, 39-43.

[27] Eberhart, R., Shi, Y. (2001). Particle swarm optimization: Developments, applications and resources. In *Proceedings of the 2001 Congress on Evolutionary Computation.* IEEE, 81-86.

[28] Shi, Y.H., Eberhart, R.C. (1999). Empirical study of particle swarm optimization. In *IEEE Congress on Evolutionary Computation*. IEEE, 1945-1950.

[29] Cui, C., Li, T., Blunt, L.A., Jiang, X., Huang, H., Ye, R., Fan, W. (2013). The assessment of straight and flatness errors using particle swarm optimization. *Procedia CIRP*, 10, 271-275.

[30] Wen, X.L., Zhu, X.C., Zhao, Y.B., Wang, D.X., Wang, F.L. (2012). Flatness error evaluation and verification based on new generation geometrical product specification (GPS). *Precision Engineering*, 36, 70-77.

Received March 18, 2017.
Accepted July 31, 2017.