AN EFFECTIVE DIMENSIONAL INSPECTION METHOD BASED ON ZONE FITTING

A Thesis

by

NACHIKET VISHWAS PENDSE

Submitted to the Office of Graduate Studies of
Texas A&M University
in partial fulfillment of the requirements for the degree of

MASTER OF SCIENCE

December 2004

Major Subject: Mechanical Engineering

AN EFFECTIVE DIMENSIONAL INSPECTION METHOD BASED ON ZONE FITTING

A Thesis

by

NACHIKET VISHWAS PENDSE

Submitted to the Office of Graduate Studies of
Texas A&M University
in partial fulfillment of the requirements for the degree of

MASTER OF SCIENCE

Approved as to style and content by:

Richard Alexander (Co-Chair of Committee)	Jyhwen Wang (Co-Chair of Committee)
Yu Ding (Member)	Steve Suh (Member)

Dennis O'Neal
(Head of Department)

December 2004

Major Subject: Mechanical Engineering

ABSTRACT

An Effective Dimensional Inspection Method Based on Zone Fitting.

(December 2004)

Nachiket Vishwas Pendse, B. Eng., Government College of Engineering, Pune, India

Co-Chairs of Advisory Committee: Dr. Richard Alexander
Dr. Jyhwen Wang

Coordinate measuring machines are widely used to generate data points from an actual surface. The generated measurement data must be analyzed to yield critical geometric deviations of the measured part according to the requirements specified by the designer. However, ANSI standards do not specify the methods that should be used to evaluate the tolerances. The coordinate measuring machines employ different verification algorithms which may yield different results. Functional requirements or assembly conditions on a manufactured part are normally translated into geometric constraints to which the part must conform. Minimum zone evaluation technique is used when the measured data is regarded as an exact copy of the actual surface and the tolerance zone is represented as geometric constraints on the data.

In the present study, a new zone-fitting algorithm is proposed. The algorithm evaluates the minimum zone that encompasses the set of measured points from the actual surface. The search for the rigid body transformation that places the set of points in the zone is modeled as a nonlinear optimization problem. The algorithm is employed to find the form tolerance of 2-D (line, circle) as well as 3-D geometries (cylinder). It is also used to propose an inspection methodology for turbine blades. By constraining the

transformation parameters, the proposed methodology determines whether the points measured at the 2-D cross-sections fit in the corresponding tolerance zones simultaneously.

Dedicated to my mother Saroj Pendse and father Vishwas Pendse who have worked hard throughout for my education and gave me the opportunity to come to the United States for higher studies.

ACKNOWLEDGEMENTS

I would like to take this opportunity to express my deep gratitude to Dr. Jyhwen Wang for giving me the opportunity to work on this research project and his guidance throughout, to its completion. Whenever I had problems in my research, he was there to patiently explain things to me. He constantly inspired and motivated me to achieve my academic goals.

It gives me immense pleasure in expressing my heartfelt gratitude to Dr. Richard Alexander for all the cooperation he has rendered in the successful completion of this work. I would also like to thank Dr. Yu Ding and Dr. Steve Suh for serving on my thesis committee. I am highly privileged to have them on my thesis committee.

TABLE OF CONTENTS

	Page
ABSTRACT	iii
DEDICATION	v
ACKNOWLEDGEMENTS	vi
TABLE OF CONTENTS	vii
LIST OF FIGURES	ix
LIST OF TABLES	xi
1. INTRODUCTION	1
1.1 Research objective	3
1.2 Literature survey	4
1.3 Research plan	7
2. DIMENSIONAL INSPECTION USING ZONE FITTING	10
2.1 Control boundaries	12
2.2 Problem description	14
3. AN EFFECTIVE ZONE FITTING METHOD	16
3.1 Mathematical formulation	17
3.2 Zone-fitting algorithm	20
3.3 Evaluation of objective function	22
3.3.1 Modified point location method for 2-D geometries	22
3.3.2 Modified point location method for 3-D geometries	25
3.4 Code-organization for new zone-fitting algorithm	27
3.4.1 Optimization method	31
3.5 Methodology for dimensional inspection with datum	31
3.6 Turbine blade model	33
3.6.1 Modified point location method for airfoil section	37
3.6.2 Program evaluating form tolerance of turbine blade	38

Page

4. RESULTS AND DISCUSSION 42

4.1 Rigid body transformation parameters 42
4.2 The 2-D line model 43
4.3 The 2-D circle model 48
4.4 The 3-D cylinder model 54
4.5 The turbine blade model 61
4.6 Tolerance assessment using actual data 65

5. SUMMARY AND CONCLUSION 69

REFERENCES 71

APPENDIX A 74

APPENDIX B 77

APPENDIX C 80

APPENDIX D 82

VITA 84

LIST OF FIGURES

FIGURE Page

1 Zone fitting 2
2 Proposed approach 7
3 Data localization (Choi and Kurfess, 1999a, b) 11
4 Control boundaries for 2-D curves and 3-D surfaces 12
5 Multiple-offset size tolerance zones (Choi and Kurfess, 1999a, b) 13
6 Framework of the zone-fitting algorithm 21
7 Modified point location method for 2-D geometry (circle) 23
8 Modified point location method for 3-D geometry (plane) 26
9 Flowchart of the zone-fitting algorithm 30
10 Methodology for dimensional inspection with datum 32
11 Eleven independent geometric parameters (Pritchard, 1985) 34
12 Five key points and the five surface functions (Pritchard, 1985) 35
13 Sections of the turbine blade in 3-D space 36
14 Sections of the turbine blade 36
15 Modified point location for airfoil section 37
16 Flowchart for the turbine blade program 39
17 2-D line model 44
18 Residual deviation after zone-fitting for 2-D line (Choi and Kurfess, 1999a, b) 46
19 Residual deviation after zone-fitting for 2-D line (new algorithm) 47
20 Residual deviation after least squares fit for 2-D line 48
21 2-D circle model 50
22 Residual deviation after zone-fitting for 2-D circle (Choi and Kurfess, 1999a, b) 52
23 Residual deviation after zone-fitting for 2-D circle (new algorithm) 53

FIGURE Page

24 Residual deviation after least squares fit for 2-D circle 54
25 Cylinder model (tolerance zone 1 – circular surface) 56
26 Cylinder model (tolerance zone 2 – bottom plane and
tolerance zone 3 – top plane).. 57
27 Residual deviation after zone-fitting for 3-D cylinder
(Choi and Kurfess, 1999a, b) 60
28 Residual deviation after zone-fitting for 3-D cylinder (new algorithm) 61
29 3-D model of the turbine blade 62
30 Tolerance zone for single airfoil section 63
31 Cylindrical rod model.. 65
32 Setup for coordinate measurement 66
33 Residual deviation after zone-fitting for cylindrical rod
(Choi and Kurfess, 1999a, b) 67
34 Residual deviation after zone-fitting for cylindrical rod (new algorithm).... 68
35 Point location method (Preparta and Shamos, 1988) 80

LIST OF TABLES

TABLE		Page
1	Notation used in flowchart	29
2	Summary of files in a project	29
3	Equations representing each piece of airfoil section	37
4	Summary of the files in the turbine blade project	41
5	Input to the primary program (2-D line model)	43
6	Transformation variables for 2-D line model	44
7	Input to the primary program (2-D circle model)	49
8	Transformation variables for 2-D circle model	51
9	Input to the primary program (3-D cylinder model)	55
10	Bilateral tolerance values for the 3-D cylinder	56
11	Transformation variables for 3-D cylinder example	58
12	Minimum zone values for the 3-D cylinder example	59
13	Input to the primary program (3-D turbine blade model)	63
14	Minimum zone values for the 3-D turbine blade example	64
15	Transformation variables for 3-D turbine blade example	64
16	Transformation variables for cylindrical model	66

1. INTRODUCTION

Coordinate measuring machines are widely used to generate data points from an actual surface. The generated measurement data must be analyzed to yield critical geometric deviations of the measured part according to the requirements specified by the designer. However, ANSI standards do not specify the methods that should be used to evaluate the tolerances (Wang, 1992). Since the coordinate measuring machines provide discrete coordinate points, these must be associated with the design geometry to evaluate the actual part deviation. For this purpose, current coordinate measuring machines typically employ data-fitting methods based on the least squares fit and the min-max fit. Generally, the min-max fit returns a smaller maximum deviation than the least squares fit. Therefore, if the least squares fit is used some acceptable parts may be rejected, and if min-max fit is used some unacceptable parts may be accepted. Such different verification results are due to the different interpretations of the measured data and differing criteria of the verification techniques (Choi and Kurfess, 1999a, b).

This thesis follows the style of the *Journal of Manufacturing Science and Engineering*.

Functional requirements or assembly conditions on a manufactured part are normally translated into geometric constraints to which the part must conform. Minimum zone evaluation technique is used when the measured data is regarded as an exact copy of the actual surface and the tolerance zone is represented as geometric constraints on the data.

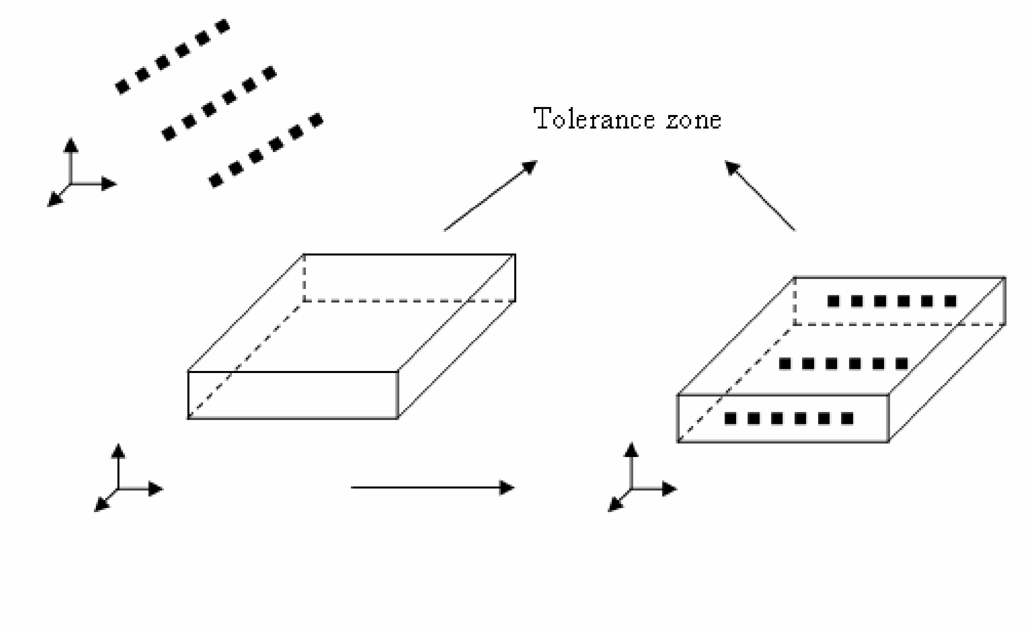

Fig. 1 Zone fitting

The tolerance conformance is achieved when the measured points fit into the tolerance zone i.e. satisfy the geometric constraints (Fig. 1). A tolerance zone is a region of space constructed by offsetting (expanding or shrinking) the object's nominal boundaries (Requicha, 1983). This tolerance zone representation is not completely in conformance with the geometric tolerance standards specified in ANSI standards

(ASME, 1994). However, this representation is more intuitive and can be easily associated with CAD models. Generally, in the evaluation of minimum zone value of form errors, such as straightness, flatness, roundness and cylindricity, nonlinear optimization techniques are applied (Kanada and Suzuki, 1993a).

1.1 Research objective

As described above, there is a discrepancy in the results that are obtained by employing different verification algorithms. The objective of the proposed research is to develop a methodology to address the tolerance assessment problem. With coordinate transformation, zone-fitting algorithms will be developed to determine whether the measured set of points lies in the specified tolerance zone. The methodology will allow inspection to be conducted where a datum is specified. In other words, transformation parameters can be constrained in the zone-fitting algorithm as per the specified datum. Given the nominal surfaces, the developed methodology will evaluate if the measured points lie in the specified tolerance limits, and will further determine the minimum. This will provide important information as to the actual part quality.

The approach employed in the present study is based on the principle put forth by Choi and Kurfess (1999a, 1999b). The proposed research will develop an objective function such that the ambiguity, whether the set of point lies completely in the tolerance zone, is eliminated. The algorithms will also assess the bi-lateral tolerance, providing a better picture of the manufacturing process.

1.2 Literature survey

Many researchers have developed zone-fitting algorithms based on different techniques. The preferred approach adopted by most of them is to model the zone-fitting problem as a nonlinear optimization problem. In the nonlinear optimization problem, an objective function is defined and a solution that optimizes the objective function is sought. Murthy and Abdin (1980) proposed several methods such as Monte Carlo technique, normal least squares fit, simplex search techniques, and spiral search techniques to evaluate the minimum zone deviation. Depending on the requirement and the problem, the individual technique or a combination of the above techniques could be applied to achieve the desired accuracy. The first step for these search techniques to find the minimum zone value is the least squares reference. Shunmugam (1987) proposed a new approach for evaluating form errors of engineering surfaces based on the minimum average deviation (MAD) principle. A stray peak or valley on the actual feature introduces considerable variations in the results obtained by the minimum deviation method. This new approach obtains the minimum zone value by minimizing the average deviation of all the points on the feature.

Carr and Ferreira (1995a, 1995b) developed algorithms which solve a sequence of linear programs that converge to the solution of the nonlinear optimization problem. Different linear programs were used for the flatness and straightness verification models. One of the flatness verification model searches for the reference plane, so that the maximum distance of each data point from this plane is minimized while the other searches for two parallel supporting planes, so that all data points are below one plane, above the other, and the planes are as close together as possible. The straightness verification model follows the second flatness verification model. Tsukada and Kanada (1985) used direct search methods such as the newly improved simplex method and Powell's method for minimum zone evaluation of the cylindricity deviation. Kanada and Suzuki (1993b) use the downhill simplex method and the repetitive bracketing method to evaluate the minimum zone flatness considering the respective convergence criteria. Huang et al. (1993a, 1993b) developed a new minimum zone method for evaluating straightness and flatness errors based on the control line rotation scheme and the control plane rotation scheme. The search for the best fit plane or line starts with the least squares plane or line as the initial condition.

Another general approach to computing the minimum zone solution is based on the computational geometry theory. Lai and Wang (1988) evaluated the straightness and roundness based on the convex polygon method. Hong and Fan (1986) proposed an eigen-polygon method. Etesami and Qiao (1990) use the two-dimensional (2-D) convex hull of the data points to solve the 2-D straightness tolerance problem. Swanson et al. (1994) proposed an optimal algorithm to evaluate the out-of-roundness factor, which determines the extent to which a planar shape deviates from a circle. This algorithm also makes use of the medial axis and farthest neighbor Voronoi diagram, but does not require their intersection, thereby yielding the improvement in complexity. Roy and Zhang (1992) proposed a mathematical formulation determining the roundness error, which exploited the properties of convex hull and Voronoi diagrams to produce a faster algorithm for establishing the concentric circles. Traband et al. (1989) compute the three-dimensional (3-D) convex hull and searches for the minimum zone of the convex hull to compute flatness. These methods are guaranteed to find the minimum zone solution but at a great computational cost (Carr and Ferreira, 1995a, b).

All the algorithms discussed above do not use the equation of the nominal surface in their evaluation of the minimum zone value. The algorithms evaluate the form tolerance such that the location and the nominal size of the feature is not a concern. All these algorithms first use a data fitting method such as the least squares fit or the min-max fit and then apply the tolerance zone to this localized data. Choi and Kurfess's (1999a, 1999b) method determines the deviation of the manufactured part geometry from the nominal geometry. Their zone-fitting algorithm searches for the rigid body transformation that places the set of point in the tolerance zone. The algorithm minimizes the square sum of the distance of every data point from the nominal surface. This new study proposes to develop a new zone-fitting algorithm based on the concepts put forth by Choi and Kurfess (1999a, 1999b). The tolerance zone will be constructed by offsetting (shrinking and expanding) the nominal surface based on the bilateral tolerance specifications and then attempt to fit all data points in this tolerance zone. Then it will repeat the process to fit all data points in the minimum possible zone resulting in the minimum zone value.

1.3 Research plan

The algorithm to be developed will determine whether a measured set of points lies in the specified tolerance zone. Fig. 2 outlines the approach that will be followed to develop the new zone-fitting algorithm. The inputs to the algorithm will be:

1. Measured set of points from the part surface
2. The equation of the nominal surface
3. The inner and the outer (bi-lateral) tolerance limits

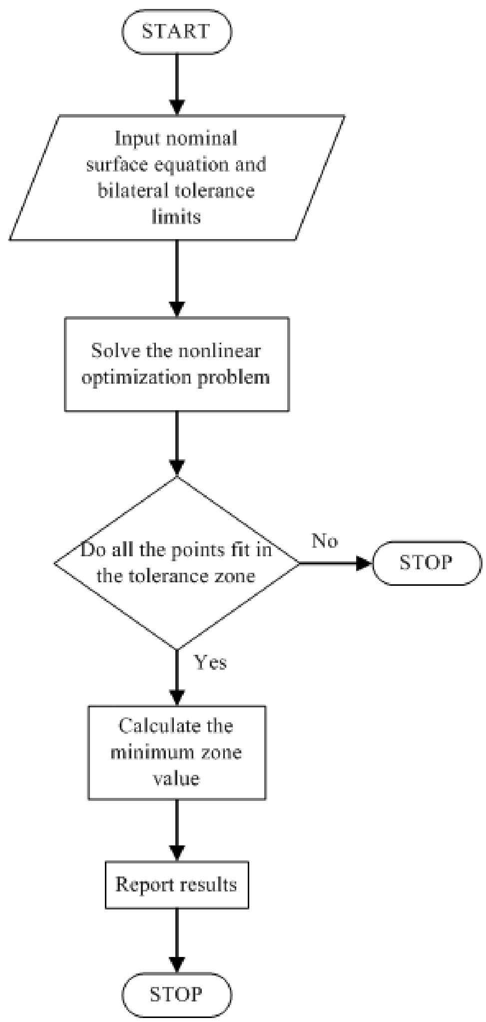

Fig. 2 Proposed approach

Similar to the approach presented by Choi and Kurfess (1999a, 1999b), computing the minimum zone solution will be modeled as a nonlinear optimization problem. The zone-fitting algorithms rely on setting a convergence tolerance. In the proposed work, a different objective function will be used so that it will eliminate the ambiguity, whether the point is in or out of the tolerance zone. The tolerance zone will be constructed parametrically with the equation of the nominal surface and given tolerance limits. The objective function will calculate the number of points that fit in the tolerance zone. The algorithm will minimize the objective function using optimization techniques available in mathematics software. If all the points lie in the tolerance zone, the algorithm will return a set of six transformation (three translational and three rotational) parameters that place the points in the zone.

If all the points lie in the tolerance zone, the algorithm will further search for the minimum zone in which the points could fit. A bi-directional binary search will be used to find the inner and outer tolerance limits. The tolerance zone will be constructed each iteration and the process described above will be carried out until the minimum zone is found. The algorithm will individually calculate the deviation from the design model in the positive (external of the solid model) and negative (internal of the solid model) directions. This will help in understanding the problems in the manufacturing process. Thus, the proposed algorithm will serve two purposes:

1. To determine whether a measured set of points lies in the specified tolerance zone

2. If they lie in the tolerance zone, to determine the minimum zone in which they fit.

For objects with a complex 3-D geometry (such as a turbine blade), inspection standards are often specified as a collection of 2-D cross-sections. By constraining the transformation parameters, the proposed methodology will determine whether the points measured at the 2-D cross-sections fit in the corresponding tolerance zones simultaneously.

2. DIMENSIONAL INSPECTION USING ZONE FITTING

The coordinate measuring machines measure a set of points from a manufactured surface. To interpret this measured data the coordinate measuring machines use various verification algorithms based on least squares fit and min-max fit. Since the algorithms are based on different criteria, there is discrepancy in the results obtained. However, if the measured points are interpreted as sampled data from the actual surface, the least squares method is a better choice than zone fitting. The zone fitting algorithm models the tolerance assessment problem as a geometric problem, where the tolerance specifications are formulated as geometric constraints. In the present research, the tolerance zone representation adopted is that proposed by Requicha (1983). The tolerance conformance decision is made based on whether the measured set of points fits in the prescribed tolerance zone i.e. if all the points fit in the tolerance zone the part surface satisfies the specified tolerance limits and if any of the points lie outside the tolerance zone the part surface does not satisfy the specified tolerance limits.

In the present research, the zone-fitting algorithm uses the equation of the nominal surface. The nominal surface forms the basis of the tolerance assessment problem. The algorithm evaluates the form tolerance such that the location and the nominal size of the feature are a concern. The coordinates generated by the coordinate measuring machines are in a different reference frame than that of the ideal surface or the design model. To compute the deviation of the actual part surface from the ideal one, the measured set of points must be placed in the same reference frame as that of the design model. To achieve this, the measured set of points must undergo a rigid body transformation (both translational and rotational). This is called as data localization (Fig. 3).

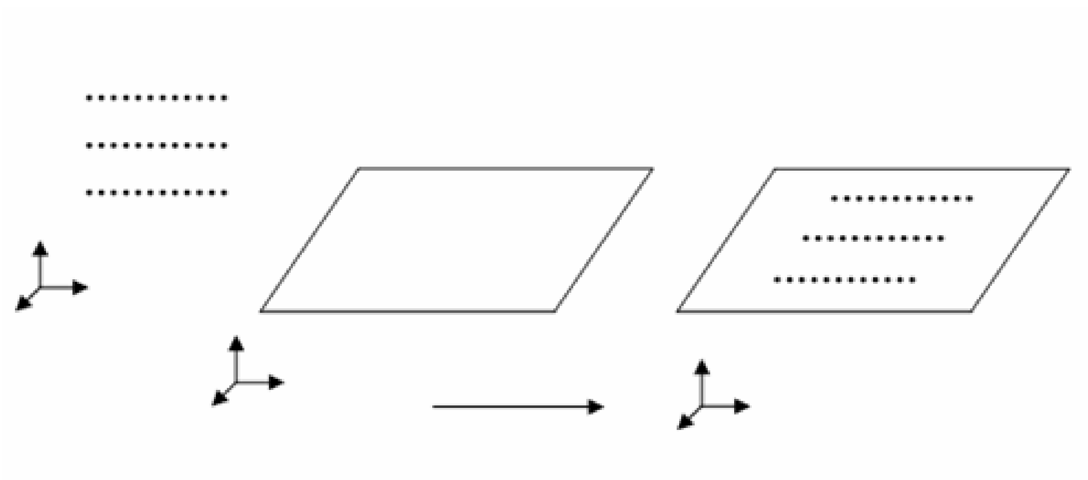

Fig. 3 Data localization (Choi and Kurfess, 1999a, b)

The tolerance zone is constructed from the ideal surface equation and the specified bilateral tolerance limits. In other words, the tolerance zone is constructed by offsetting (expanding and shrinking) the part's nominal boundaries (nominal equations). After the tolerance zone is constructed, the algorithm attempts to fit the set of measured points in it. Based on the result obtained, the algorithm returns a pass/fail decision. Then the algorithm proceeds further to find the minimum zone that encompasses the measured set of points. The minimum zone value gives the deviation of the actual part surface from the ideal part surface.

2.1 Control boundaries

Defining control boundaries around the ideal part surface forms the tolerance zone. The concept of control boundaries remains the same for curves in 2-D or surfaces in 3-D. Fig. 4 shows some examples of control boundaries for the 2-D curve as well as a 3-D surface. The outer and the inner circles form the tolerance zone while in which in the case of the 3-D surface the upper and the lower planes form the envelope in which the measured set of points should lie. The 2-D control boundaries can also be employed in situations where the object has a very complex geometry. The idea is to have multiple sections at different locations from a specified datum reference. Then treat the 2-D curves at each section as a problem involving 2-D form evaluation.

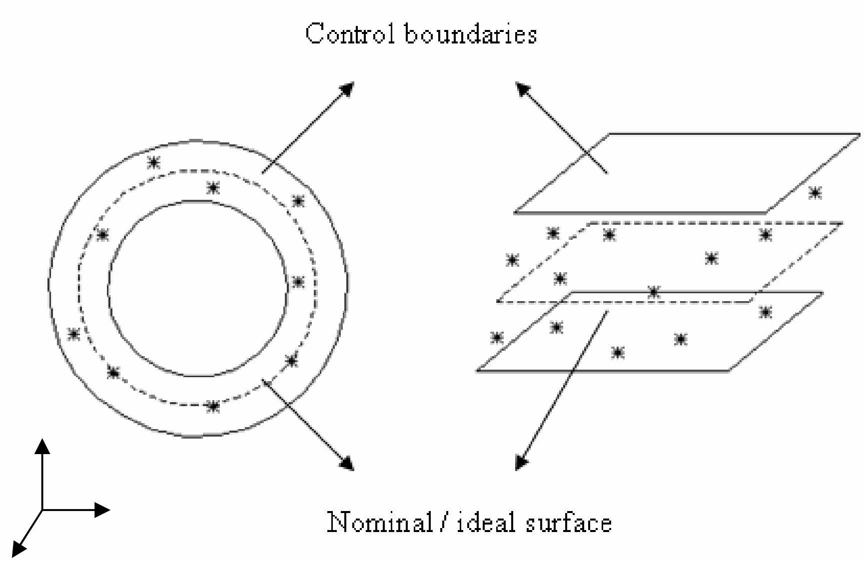

Fig. 4 Control boundaries for 2-D curves and 3-D surfaces

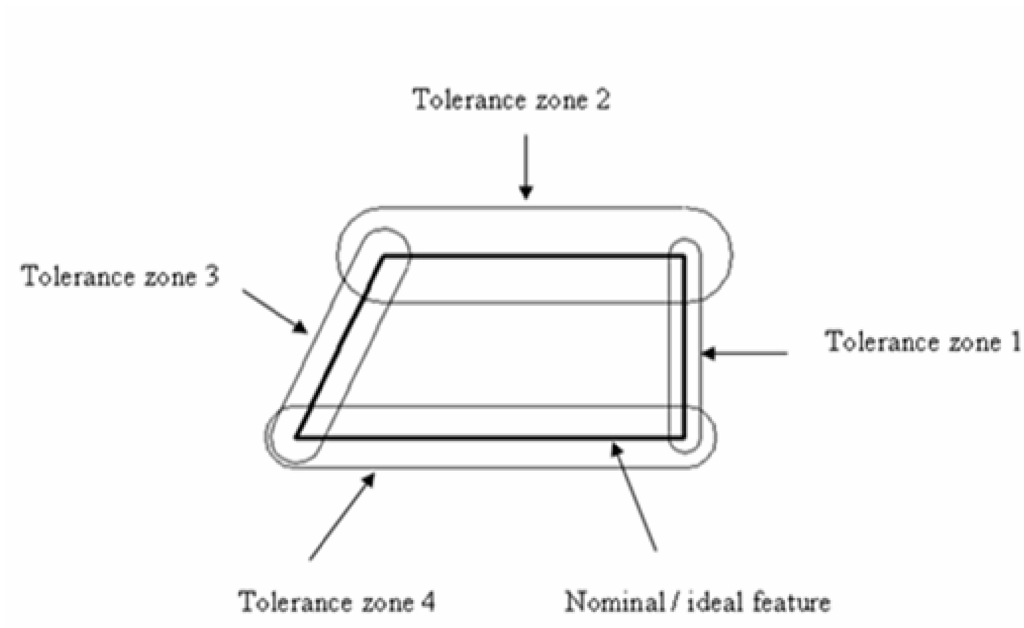

Fig. 5 Multiple-offset size tolerance zones (Choi and Kurfess, 1999a, b)

The control boundaries directly depend on the bilateral tolerances prescribed for a particular geometric feature. The tolerance zone is not necessarily uniform across the entire part (Fig. 5). Different features on a part can have different tolerance zones. Even the inner and the outer offset values (e.g., +2 and -3) for the tolerance zones can be different. The verification algorithms currently employed in the coordinate measuring machines evaluate the deviations from the nominal surface and then compare it to the tolerance specifications to make a decision. However, when tolerance zone is specified as the conformance criterion the deviation need not be evaluated, only determine whether a point lies inside or outside the zone. Assuming that the measured set of points represents the actual surface completely, the actual surface must be verified only against the tolerance zones.

To make zone fitting a valid tool for deciding whether the manufactured part surface satisfies the prescribed tolerance limits, the set of measured points must be considered as an exact copy of the part surface. Since the coordinate measuring machines measure only a finite number of points from the surface, there may be an unmeasured section of the actual surface which has a large deviation. This deviation has no bearing on the conformance decision made by the zone fitting algorithm. Therefore, it must be noted that the results given by the zone fitting algorithm are a function of the set of measured points.

2.2 Problem description

The tolerance zone is a region in the 3-D space where the measured set of points must lie. The rigid body transformation that places the measured points in the tolerance zone is to be found. Consider a surface in the 3-D space and a set of points measured from the actual surface. The reference frames of the design / ideal surface and that of the coordinate measuring machine are different. To find the deviation of the actual surface from the ideal surface, the set of points measured from the actual surface must be localized i.e. the set of points must be transformed (rotation and translation) such that they are in the same reference frame as the design / ideal surface. There exists a set of transformation parameters (3 rotational and 3 translational) that places the points in the tolerance zone. Thus the problem is to search for the rigid body transformation parameters in the six-dimensional parameter space. In order to obtain a solution for the optimization problem, an objective function is defined. The function is so defined that it returns the number of points that lie outside the tolerance zone. If the point is inside the tolerance zone the function value is zero and if the point is outside the tolerance zone the function value is one. Thus the function can be called as a Boolean function. The position of the point depends on the rigid body transformation and hence by minimizing this objective function will give the solution to the optimization problem i.e. the six transformation parameters. This process can be performed repeatedly to find the smallest zone that encompasses the set of measured points. This yields the minimum zone value for the geometric feature that is inspected.

In short, the problem can be defined as the search for the minimum zone that encompasses the set of measured points with the help of rigid body coordinate transformation.

3. AN EFFECTIVE ZONE FITTING METHOD

A new zone-fitting algorithm is developed to obtain the solution to the nonlinear optimization problem i.e. to find the rigid body transformation that places the measured set of points in the zone. This chapter describes in detail the mathematical formulation of the problem and the techniques employed in developing the new algorithm. The algorithm is based on the concepts put forth by Choi and Kurfess (1999a, 1999b). Considerable improvements have been made in this algorithm as compared to the one developed by Choi and Kurfess (1999a, 1999b). The notable improvements are as follows:

1. The zone-fitting algorithms rely on setting a convergence tolerance and thus create an ambiguity of whether all the points lie in the tolerance zone. The new algorithm employs the Boolean function, which by virtue of its definition eliminates this ambiguity.
2. The new algorithm evaluates the bilateral minimum zone values for the features under inspection. Thus the manufacturing-process supervisor has a better grasp of the direction the process heading based on the minimum zone values for that particular feature.
3. The new algorithm is implemented for assessing the tolerance of a complex 3-D object (such as a turbine blade). For these 3-D objects, the inspection standards are often specified as a collection of 2-D cross-sections. By constraining the transformation parameters, the proposed methodology will determine whether the

points measured at the 2-D cross-sections fit in the corresponding tolerance zones simultaneously.

3.1 Mathematical formulation

Consider a surface S(x,y,z) in the 3-dimensional space. Let P be the set of points measured from the part surface. To place P in the same reference plane of the design model, P is transformed to P* by a rotational matrix R and a translational vector t.

$$P = \{p_i, i = 1...n\} \ \text{n : number of measured points}$$

The rotational matrix R and the translational vector t can be combined in a single matrix called the homogeneous transformation matrix H. H is given as follows:

$$H = \begin{bmatrix} R & t \\ 0 & 1 \end{bmatrix}$$

R, the Euler matrix is given by:

$$R = \begin{bmatrix} \cos\Theta\cos\Phi & \cos\Theta\sin\Phi & -\sin\Theta \\ -\cos\Psi\sin\Phi + \sin\Theta\sin\Psi\cos\Phi & \cos\Phi\cos\Psi + \sin\Theta\sin\Psi\sin\Phi & \sin\Psi\cos\Theta \\ \sin\Psi\sin\Phi + \sin\Theta\cos\Psi\cos\Phi & -\sin\Psi\cos\Phi + \sin\Theta\cos\Psi\sin\Phi & \cos\Theta\cos\Psi \end{bmatrix}$$

and,

$$t = \begin{bmatrix} t_x \\ t_y \\ t_z \end{bmatrix}$$

where, Ψ, Θ, Φ are rotations about the X, Y and Z axes respectively, while t_x, t_y, t_z are the translations along X, Y and Z axes respectively. In order to represent the transformation as a matrix multiplication, the vector p needs to be augmented by the addition of the fourth component 1 as follows:

$$p^a = \begin{bmatrix} p \\ 1 \end{bmatrix}$$

where, p^a is the augmented vector p.

$$P^*(\Theta, \Phi, \Psi, t_x, t_y, t_z) = \{u_i \in R^3 | u_i = H(R,t) \times p_i^{\ a}\}, i = 1...n$$

where, $u_i = H(R,t) \times p_i^{\ a}$ and R^3 represents the 3-D space. Individual tolerance zone (f_i) is created around the individual surface (s_i) based on the bilateral tolerances specified for that surface. Let the inner and outer tolerances be d_{in} and d_{out}. Therefore, we have

$$T(S, d_{in}, d_{out}) = \{u \in R^3 | -d_{in} \le dist(u,S) \le d_{out}\}$$

The tolerance zone T is a union of all the individual tolerance zones (f_i) while S is the union of the individual surfaces (s_i). For a given S, T and P, H is to be determined such that P* lies in T i.e. P* belongs to T(S, d_{in}, d_{out}). To find the transformation that places the measured point set P in to the tolerance zone T constitutes a nonlinear optimization problem. The function is defined as,

$$N(u, T(S, d_{in}, d_{out})) = \begin{cases} 0 & if\ u \in T(S, d_{in}, d_{out}) \\ 1 & if\ u \notin T(S, d_{in}, d_{out}) \end{cases}$$

If a point lies in the tolerance zone the value of the function is 1, and if the point is outside the tolerance zone the value of the function is 0. Therefore, the function N (u, T) is a Boolean function. Since we have to find the transformation that places the points in the tolerance zone T, the function N (u, T) can be used as the objective function. Since the minimum state of the function N gives the feasible domain, optimization can find the rigid body transformation that places the points in tolerance zone T. The optimization problem is modeled as follows:

$$Min\,Obj = \sum_{i=1}^{n} N(u_i, T(S, d_{in}, d_{out}))$$

where, $u_i = H(R,t) \times p_i^{\ a}$.

The solution of optimization gives the transformation parameters $(\Theta, \Phi, \Psi, t_x, t_y, t_z)$ that minimize the objective function. The objective value becomes zero when all the points lie in the zone and has a non-zero value if any one of the points is outside the zone.

3.2 Zone-fitting algorithm

To find the minimum zone values for the geometric features under dimensional inspection, the algorithm has to solve the nonlinear optimization problem repeatedly. The inputs to the algorithm are:

1. Measured set of points from the part surface (P).
2. The equation of the nominal geometric feature (S)
3. The inner and the outer (bi-lateral) tolerance limits (d_{in}, d_{out}). The control boundaries i.e. the tolerance zone (T) will be constructed by expanding and shrinking the nominal boundaries of the feature.
4. Initial estimate (x0) of the six transformation parameters that place the points in the zone.

Fig. 6 outlines the framework for the new zone-fitting algorithm in the form of a flow chart. To search for the feasible domain in the six-dimensional parameter space, the objective function (N) in the nonlinear unconstrained optimization problem is minimized. The objective function to be minimized is coded as a separate MATLAB function and is called by the main program to initiate the process of optimization.

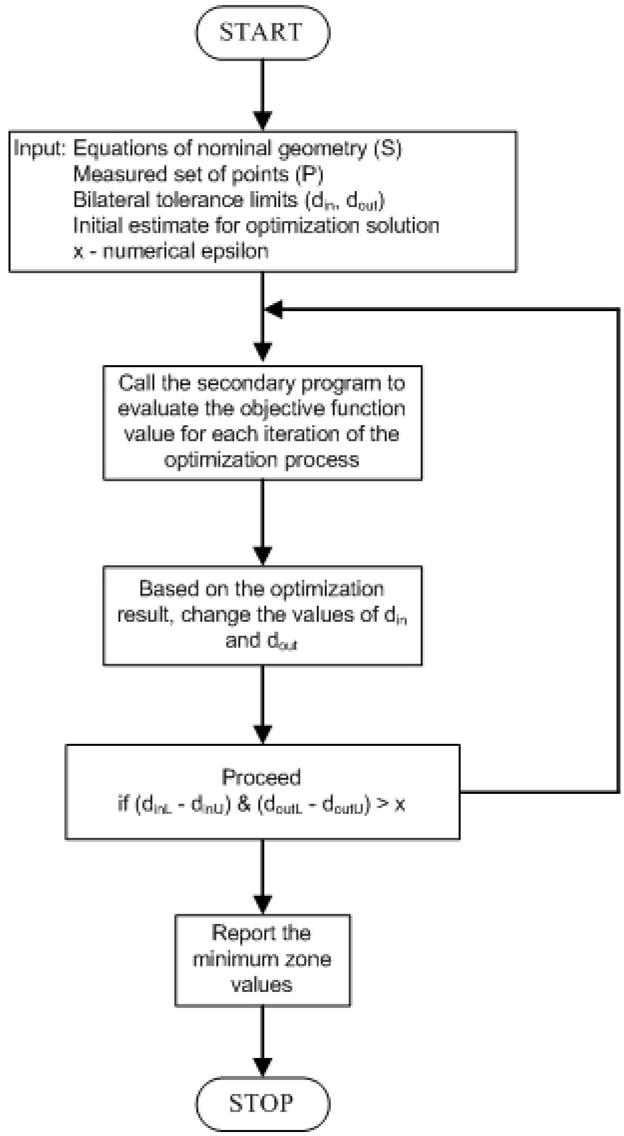

Fig. 6 Framework of the zone-fitting algorithm

The function calculates the number of points that are outside the tolerance zone by employing the modified point location method. The method is a modified form of the point location method and is applied in similar manner to both the 2-D and 3-D geometric features. The method is explained in detail in the next section with 2-D and 3-D examples. After a definite number of iterations the optimization process returns a minimized objective function value i.e. it returns number of points from the point set P* that lie outside the tolerance zone T. Based on this value, the values of d_{in} and d_{out} are either increased or decreased. A binary search technique is utilized to find the minimum bilateral tolerance values.

3.3 Evaluation of objective function

The point location method serves the purpose very well if the problem is to check whether a point lies in the convex polygon. The method is explained in Appendix C. In the new zone-fitting algorithm, geometric features such as circle, composite 2-D geometries which are a collection of higher order curves, and 3-D planes are tackled. Therefore, the point location method is modified to satisfy the requirements of the new algorithm. The implementation of the modified point location method for evaluating the objective function is explained through two examples.

3.3.1 Modified point location method for 2-D geometries

The point location method is modified so that it can be successfully applied to the 2-D geometries. Consider a tolerance zone defined by two circles as shown in the Fig. 7. A circle is defined by its center and the radius. Let r be the radius of the circle and $q = (a,b)$ be the center of the circle. Then the circle is represented as follows:

$$(x-a)^2 + (y-b)^2 = r^2 \tag{3.1}$$

From the above equation, the control boundaries for constructing the tolerance zone can be deduced as follows:

$$(x-a)^2+(y-b)^2=r_i^2 \qquad (3.2)$$
$$(x-a)^2+(y-b)^2=r_o^2$$

where, r_i and r_o are the inner and outer radii respectively.

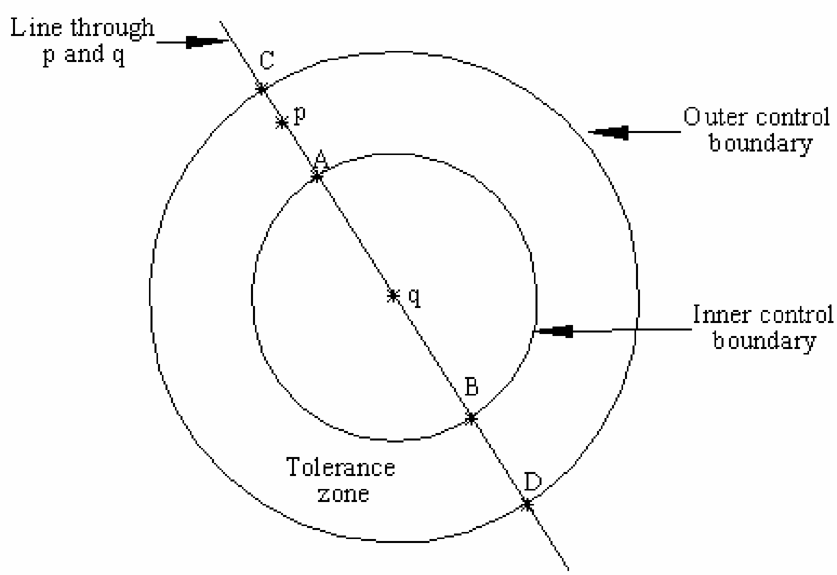

Fig. 7 Modified point location method for 2-D geometry (circle)

We have to find whether the point p lies in the tolerance zone. To determine that the equation of the line passing through points p and q is found out. The equation of the line in 2-D is of the form:

$$a_1 x + b_1 y + c_1 = 0 \tag{3.3}$$

To find the points where the line and the control boundaries intersect, (3.2) and (3.3) are solved simultaneously. Since the equation of circle is of second order, the line intersects the inner and the outer circles in 2 distinct points each as shown in Fig. 3.3. Let the line intersect the inner circle at *A* and *B* and the outer circle at *C* and *D*. By employing the parametric equation for circle, points *A* and *C* are selected for further calculations since they lie in the same quadrant as that of point *p*. Now the line passing through *q, p, A* and *C* is also represented in the parametric form. The parametric form is given by:

$$\begin{aligned} x &= (1-t)x_1 + tx_2 \\ y &= (1-t)y_1 + ty_2 \end{aligned} \tag{3.4}$$

In the current case, the value of parameter *t* is zero at point *q* and *t* is one at point *p* i.e. the starting point of the line is *q* and the end point is *p*. Using (3.4) the values of parameter *t* at points *A* and *C* are calculated. The point *p* lies in the tolerance zone if and only if the following conditions are satisfied.

1. $t_A \leq 1$
2. $t_C \geq 1$

where, t_A and t_C are values of *t* at points *A* and *C* respectively. Similar approach is followed while dealing with composite 2-D geometries which are a collection of 2-D curves. The above procedure is employed for each of the curve in the collection.

3.3.2 Modified point location method for 3-D geometries

Similar to the procedure followed for the 2-D geometry, to find out whether the point lies between two planes. Consider two parallel planes represented as follows:

$$\begin{aligned} a_1x + b_1y + c_1z + d_1 &= 0 \\ a_2x + b_2y + c_2z + d_2 &= 0 \end{aligned} \tag{3.5}$$

where, $a_1 = a_2; b_1 = b_2; c_1 = c_2$.

These two planes form the inner and outer control boundaries forming the tolerance zone as shown in Fig. 8. We have to find whether the point *p* lies in the tolerance zone. To determine that the equation of the line passing through points *p* and *q* is found out. The equation of the line in 3-D is of the form:

$$\frac{x - x_1}{x_2 - x_1} = \frac{y - y_1}{y_2 - y_1} = \frac{z - z_1}{z_2 - z_1} \tag{3.6}$$

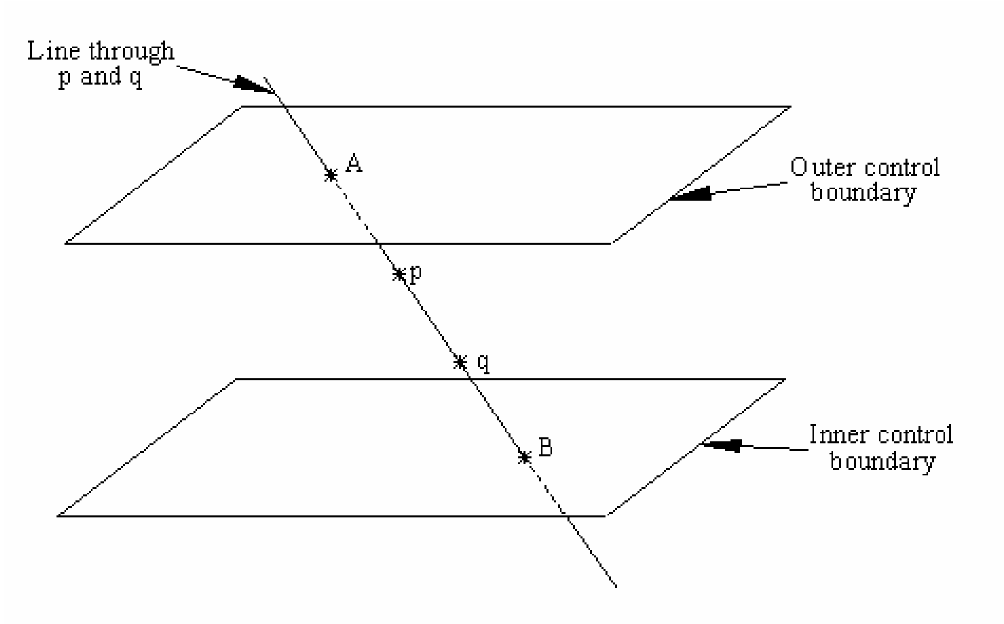

Fig. 8 Modified point location method for 3-D geometry (plane)

To find the points where the line and the control boundaries intersect, (3.5) and (3.6) are solved simultaneously. Let the line intersect the outer plane at *A* and the inner plane at *B*. Now the line passing through *q*, *p*, *A* and *B* is represented in the parametric form. The parametric form is given by:

$$
\begin{aligned}
x &= (1-t)x_1 + tx_2 \\
y &= (1-t)y_1 + ty_2 \\
z &= (1-t)z_1 + tz_2
\end{aligned}
\tag{3.7}
$$

In the current case, the value of parameter *t* is zero at point *q* and *t* is one at point *p* i.e. the starting point of the line is *q* and the end point is *p*. Using (3.7) the values of parameter t at points A and B are calculated. The point p lies in the tolerance zone if and only if the following conditions are satisfied.

1. $t_A \geq 1$
2. $t_B \leq 0$

where, t_A and t_B are values of t at points A and B respectively.

3.4 Code-organization for new zone-fitting algorithm

The new zone-fitting algorithm is written in MATLAB. Separate projects have been developed for assessing the form tolerances of a line, a circle, a 3-D cylinder, and a turbine blade. The files in which, the programs in MATLAB are written, require ".m" as their extension. The projects essentially are made up of three files – two ".m" files and a text file having a ".txt" extension. The functions of each are outlined below:

1. **Input file:** The naming convention followed for the input file is <identifier>.txt. The input file contains the X, Y and Z coordinates of the set of points (**P**). The secondary program accesses the input file and retrieves the coordinates of a point at a time.
2. **Primary program (calling program):** This is one of the ".m" files in the project. This is the main program and the naming convention followed is main_prog_<identifier>.m. The main program initiates the optimization process by calling MATLAB function defining the objective function. Depending on the results from the optimization, the minimum zone values are calculated. The binary search technique is employed.

3. **Secondary program (called program):** This is the other ".m" file in the project and the naming convention followed is opti_func_<identifier>.m. This is the program called by the main program. It contains the definition of the objective function to be minimized and the process required to evaluate the objective function value for every point in the point set. The cumulative objective function value (the number of points of the point set that lie outside the tolerance zone) is returned to the main program.

Fig. 9 gives the flowchart on which the programs for evaluation of the various form tolerances are based. The whole algorithm is divided into two basic parts. In the first part, it is evaluated whether all the points lie in the specified tolerance limits. If the points P* do not lie in the tolerance zone T, the algorithm reports the decision that the geometric feature under inspection does not satisfy the tolerance specifications. If the points P* satisfy the tolerance specifications, the algorithm proceeds to the second part where it calculates the minimum zone in which all the points lie by employing the binary search technique. Even when the points do not fit in the specified zone, the program can be extended to find the zone in which all the points fit. The programs for the different form tolerance evaluations differ from each other in content mainly due to their different mathematical representations. Table 1 gives the notation used in the flowchart and Table 2 gives the summary of the files used in the project.

The method proposed by Choi and Kurfess (1999a, 1999b) employs a numerical epsilon for the binary decision – whether the points fit in the zone. This creates an ambiguity whether all the points fit in the zone. In the new zone-fitting algorithm, no such convergence tolerance is specified thus eliminating the ambiguity. The objective function either returns a zero (when all points fit in the zone) or a non-zero value (any one of the points do not fit in the zone).

Table 1 Notation used in flowchart

Variable name	Initial value	Function
fval	0	Value of the objective function after the optimization process.
d_{inL}	0	Lower limit for $d_{in.}$
d_{inU}	d_{in}	Upper limit for $d_{in.}$
d_{outL}	0	Lower limit for $d_{out.}$
d_{outU}	d_{out}	Upper limit for $d_{out.}$
flag_in	0	To indicate whether a point is outside or inside the inner tolerance band.
flag_out	0	To indicate whether a point is outside or inside the outer tolerance band.
x	1e-7	Numerical epsilon to terminate binary search.

Table 2 Summary of files in a project

	Naming Convention	Function
Primary program (calling program)	main_prog_<identifier>.m	Initiates the optimization process. Calculates the minimum zone values.
Secondary Program (called program)	opti_func_<identifier>.m	Evaluates the cumulative objective function value and returns it to the primary program.
Input file	<identifier>.txt	Stores the X, Y and Z coordinates of the set of points.

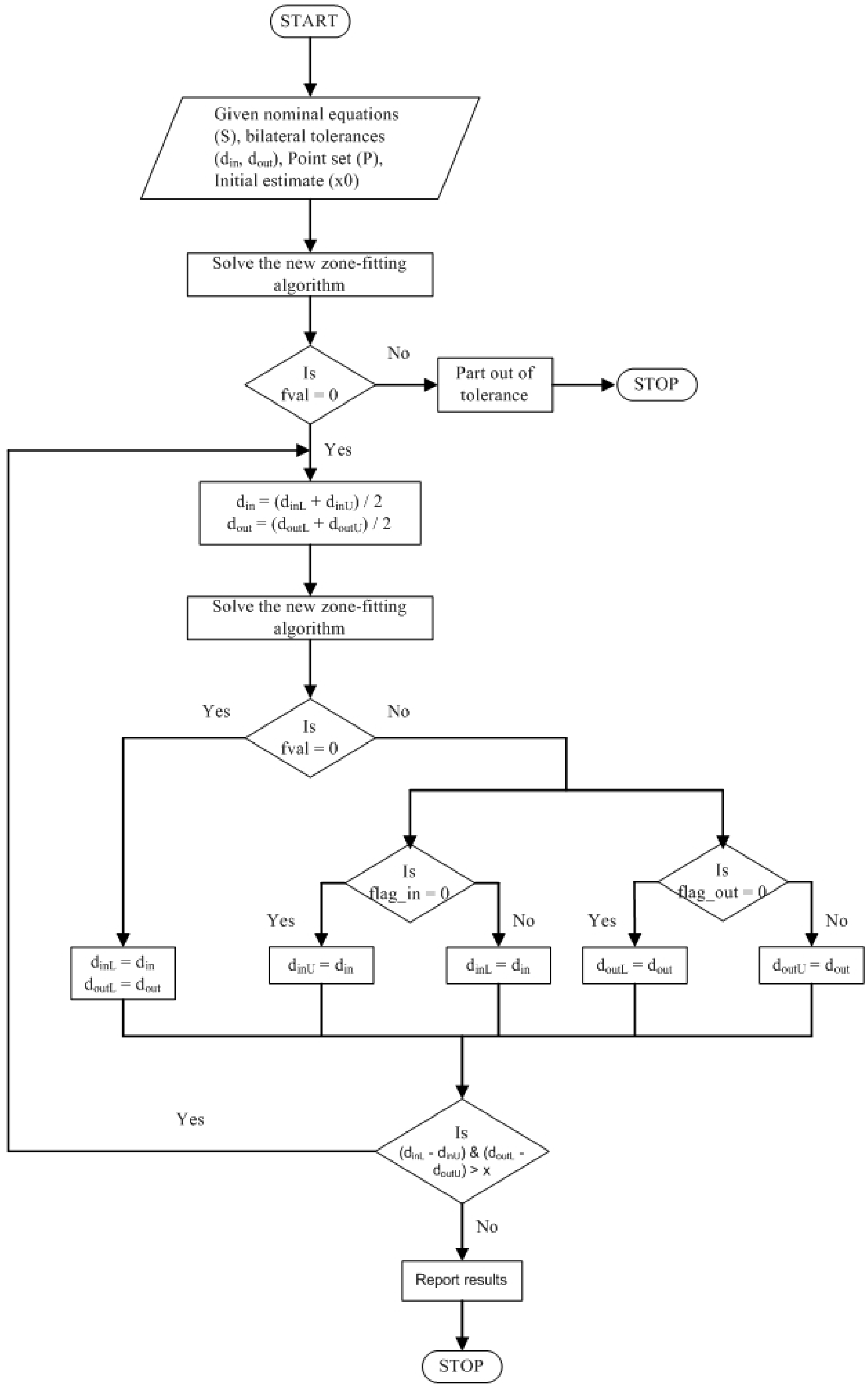

Fig. 9 Flowchart of the zone-fitting algorithm

3.4.1 Optimization method

Optimization techniques are used to find a set of design parameters that can in some way be defined as optimal. An efficient and accurate solution to this problem depends not only on the size of the problem in terms of the number of constraints and design variables but also on characteristics of the objective function and constraints. The search for the transformation parameters (three translational and three rotational) is modeled as an unconstrained nonlinear optimization problem. To carry out the nonlinear optimization, the programs for evaluating the form tolerances employ a function provided by MATLAB – “fminunc”. The function is used to find the minimum of an unconstrained multivariable function. The “fminunc” uses the BFGS Quasi-Newton method with a mixed quadratic and cubic line search procedure. The BFGS Quasi-Newton method is explained in Appendix B.

3.5 Methodology for dimensional inspection with datum

Due to the complex geometries of certain 3-D objects such as turbine blade, carrying out the form tolerance assessment becomes a difficult task. One way to make this task easier is to simplify the complex 3-D geometries by representing them as a stack of 2-D cross-sections. For this representation, it is essential that a reference plane is specified. So now the 2-D cross-sections that constitute the 3-D geometry can be defined by specifying their respective distance from the datum (reference plane). The bilateral tolerance specifications for the 3-D geometry are applied to each 2-D cross-section. Thus the tolerance zones are constructed for each cross-section by offsetting the nominal 2-D geometry of the particular cross-section, with the bilateral tolerance values.

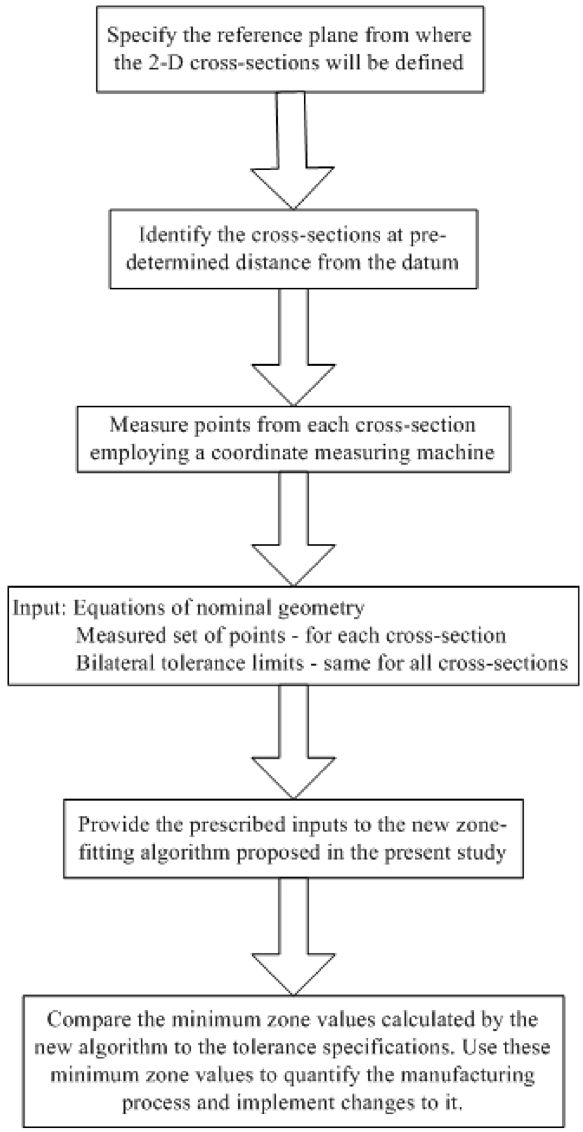

Fig. 10 Methodology for dimensional inspection with datum

In the new zone-fitting algorithm, the rigid body transformation that places the points in the tolerance zone can be constrained. In other words, the degrees of freedom enjoyed by the set of points are reduced. In the proposed methodology, the set of points is allowed to translate along the X and Y axes and rotate about the Z-axis only. Thus by constraining the transformation, the zone-fitting algorithm determines whether the points measured at the 2-D cross-sections fit in the respective tolerance zones simultaneously. However, to address any uncertainty in the height measurement, the set of points may be allowed to translate along the Z-axis. The outline of the methodology for dimensional inspection with datum is shown in Fig. 10. The methodology is explained extensively in the next section using the turbine blade as an example.

3.6 Turbine blade model

The blade suction and pressure surfaces have a very complex geometry; hence, it is not easy to carry out thorough dimensional inspection. The present research proposes a method to carry out a comprehensive turbine blade inspection employing the new zone-fitting algorithm. In the present study, the turbine blade geometry model adopted is that proposed by Pritchard (1985). Pritchard proposes that to uniquely define an airfoil cascade on a cylinder requires only eleven parameters and the immediate result is a nozzle or rotor with analytically defined surfaces. These eleven independent parameters are found to be necessary and sufficient for creating an airfoil (Pritchard, 1985). The eleven parameters are as shown in the Fig. 11. These parameters translate into five points and five slopes on the cylinder of given radius. These five key points on the airfoil surface result from:

1. locating the leading and trailing edge circles in space
2. finding the suction and pressure surface tangency points
3. setting the throat

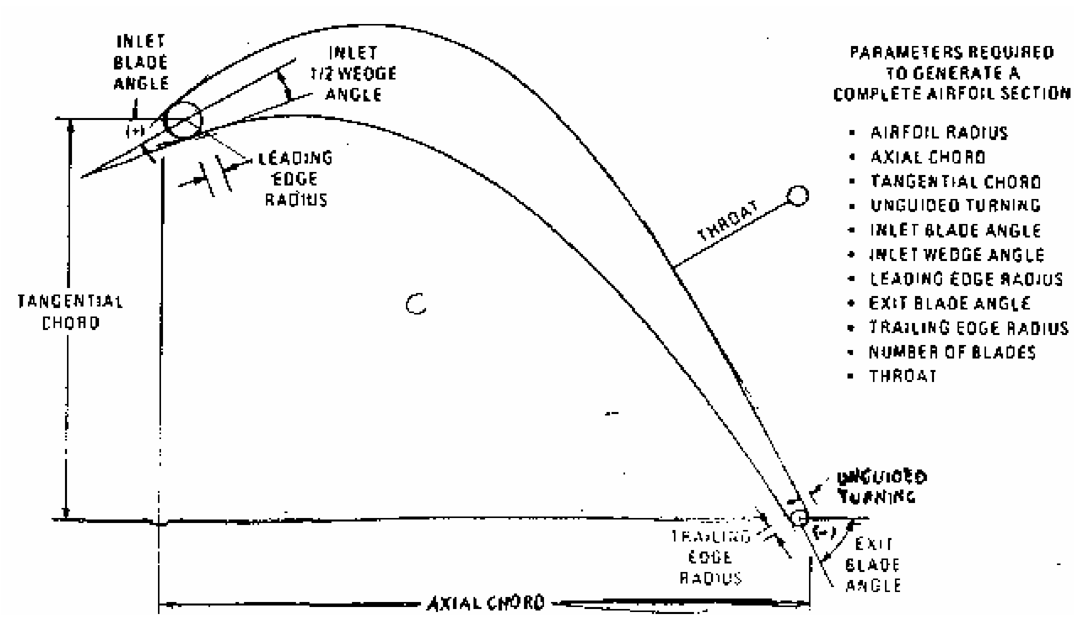

Fig. 11 Eleven independent geometric parameters (Pritchard, 1985)

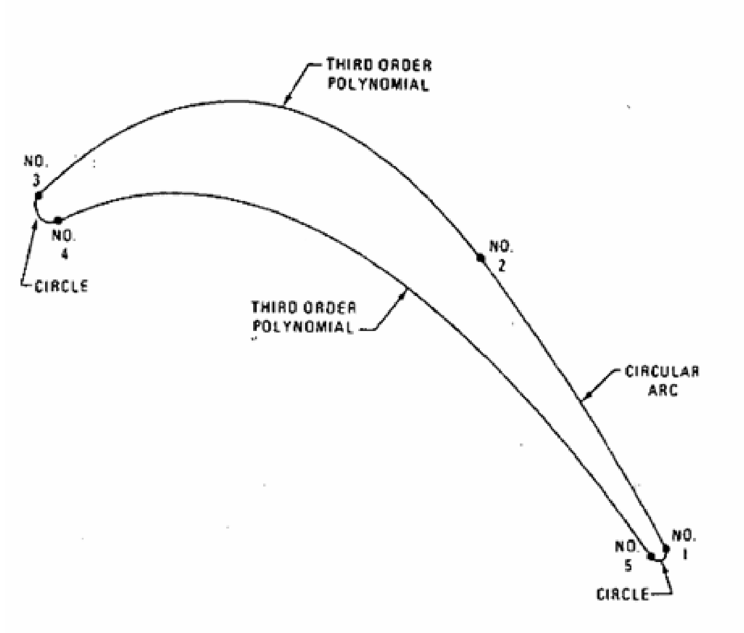

Fig. 12 Five key points and the five surface functions (Pritchard, 1985)

The five key points are computed from equations specified by Pritchard (1985). The five key points are connected by five mathematical functions as shown in Fig. 12. Logical choices for three of these functions are a leading edge circle, trailing edge circle and a circular arc describing the uncovered suction surface past the throat (Pritchard, 1985). The suction surface and the pressure surface are best described by third order polynomials. This model can successfully create airfoils at different heights from the hub of the blade. By stacking these airfoil sections about the stacking axis, the 3-D model of the turbine blade is developed. Figs. 13 and 14 show the turbine blade with six sections developed using the above model.

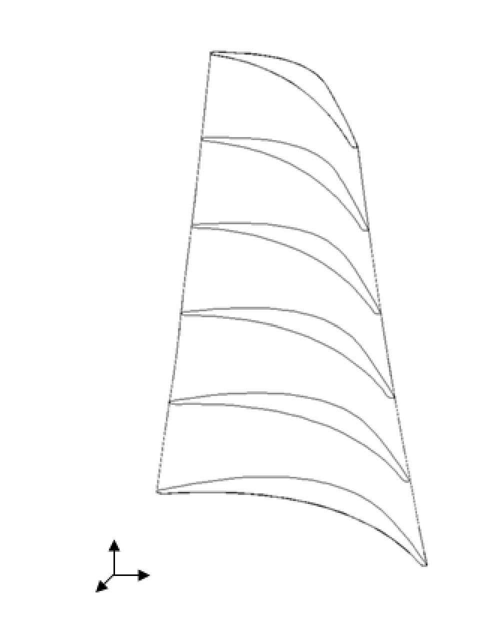

Fig. 13 Sections of the turbine blade in 3-D space

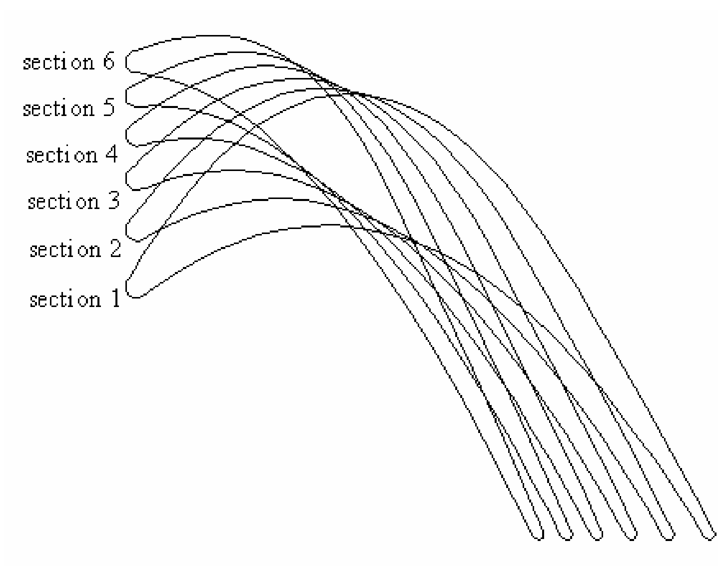

Fig. 14 Sections of the turbine blade

Table 3 Equations representing each piece of airfoil section

	Mathematical function
Leading edge circle	$(x-a_{LE})^2+(y-b_{LE})^2=r_{LE}{}^2$
Trailing edge circle	$(x-a_{TE})^2+(y-b_{TE})^2=r_{TE}{}^2$
Circular arc	$(x-a_C)^2+(y-b_C)^2=r_C{}^2$
Suction surface	$y=a_1x^3+a_2x^2+a_3x+a_4$
Pressure surface	$y=b_1x^3+b_2x^2+b_3x+b_4$

Table 3 gives the mathematical representation for each piece of the airfoil section. Suffices *LE*, *TE* and *C* denote leading edge, trailing edge and circular arc respectively.

3.6.1 Modified point location method for airfoil section

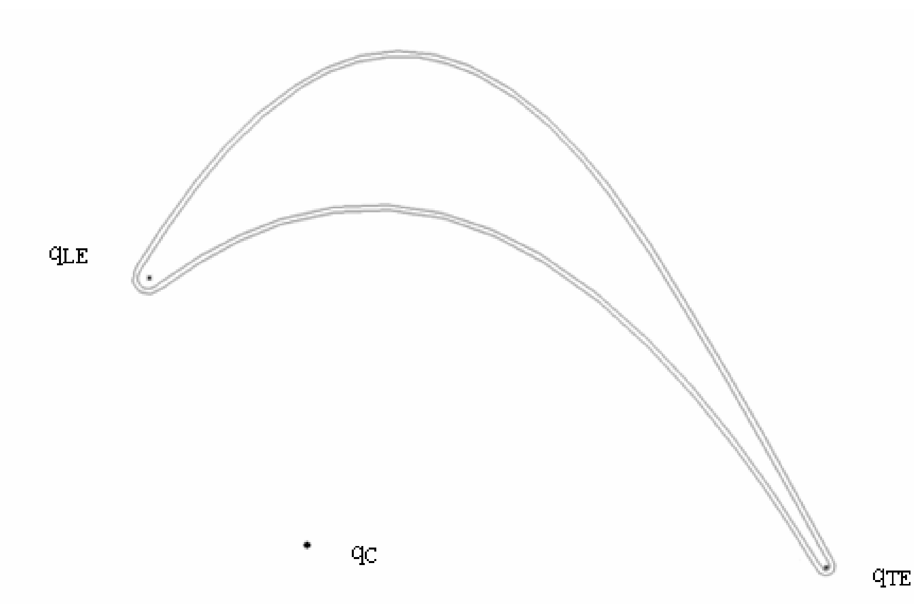

Fig. 15 Modified point location for airfoil section

The point location method for the airfoil section of the turbine blade is similar to that described for the 2-D geometries. The only difference is that there are more than one centroid points defined for the airfoil section. The airfoil section is a composite curve made of 5 mathematical functions. As shown in Fig. 15, q_C, q_{LE} and q_{TE} are the centroid points for the suction curve, pressure curve and the circular arc, leading circle and the trailing circle respectively. The remaining process is exactly the same as described for the 2-D geometry in section 3.3.1.

3.6.2 Program evaluating form tolerance of turbine blade

The project for evaluating the form tolerance of a 3-D turbine blade consists of six files. The files are named as per the convention mentioned earlier.

1. main_prog_blade.m
2. opti_func_blade.m
3. section_calc_blade.m
4. circle_line_blade.m
5. cubic_line_blade.m
6. blade<n>.txt (Number of files is equal to the number of sections)

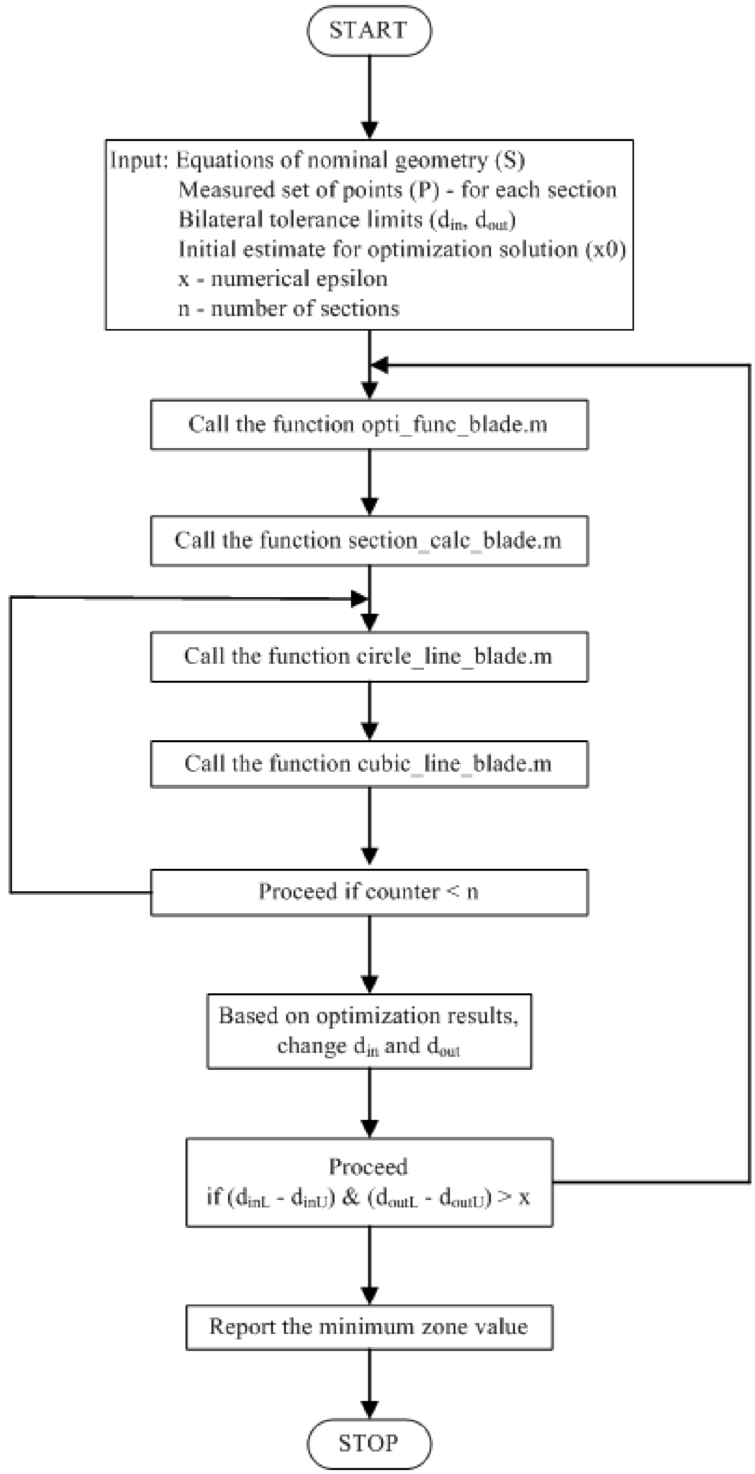

Fig. 16 Flowchart for the turbine blade program

The 3-D turbine blade is constructed from the airfoil sections at specific distance from the base section. For the dimensional inspection, points are measured at each airfoil section and the new zone-fitting algorithm attempts to fit these points in the respective tolerance zones simultaneously. Fig. 16 shows the flowchart of the program for evaluating the form tolerance of the 3-D turbine blade. The extra input to the primary program is the number of sections *n* used to construct the 3-D blade. Since the airfoil section is a composite curve consisting of five surface functions – two third order polynomials and three circular arcs, the processing for evaluating the objective function is divided in two separate functions:

1. To determine the intersection points between the cubic curve and the line joining q_C and *p*.
2. To determine the intersection points between the circular arcs and the line joining respective centroid points (q_{LE} or q_{TE}) and *p*.

The tolerance zone for each airfoil section is constructed by offsetting the composite curve. Hence the control boundaries of the tolerance zone are also composite curves with similar five surface functions. For every point it is determined whether it lies in any part of the tolerance zone. This process is performed for each airfoil section of the blade, and a single minimum zone value is determined for the turbine blade. Table 4 summarizes functions performed by each file in the project.

Table 4 Summary of the files in the turbine blade project

File	Function
main_prog_blade.m	Initiates the optimization process. Calculates the minimum zone values.
opti_func_blade.m	Evaluates the cumulative objective function value and returns it to the main program.
section_calc_blade.m	Determines the number of points outside the tolerance zone for each section of the turbine blade.
circle_line_blade.m	Determines the intersection points between a circle and a line. Returns an indicator specifying whether a point fits in this particular part of the tolerance zone.
cubic_line_blade.m	Determines the intersection points between a cubic curve and a line. Returns an indicator specifying whether a point fits in this particular part of the tolerance zone.
blade<n>.txt	Stores the X, Y and Z coordinates of the set of points. There are n files one for each airfoil section.

4. RESULTS AND DISCUSSION

This chapter presents the results of the new zone-fitting algorithm. The new algorithm is employed for evaluating the form tolerances of 2-D and 3-D geometric features such as 2-D line, 2-D circle, 3-D cylinder and turbine blade. First, the results of the new algorithm in case of 2-D line and 2-D circle are compared to that of the method proposed by Choi and Kurfess (1999a, 1999b) and to that of the least squares method. Then, the tolerance assessment result for the 3-D cylinder is compared with that of the method proposed by Choi and Kurfess (1999a, 1999b). Lastly, the new algorithm is utilized to evaluate the form tolerance of a turbine blade.

4.1 Rigid body transformation parameters

After the optimization (minimization of the objective function) terminates successfully, the algorithm returns the rigid body transformation parameters $(\Theta, \Phi, \Psi, t_x, t_y, t_z)$ that place all points in the tolerance zone. Each of the parameter is explained below:

1. $\Psi - Rotation\, about\, the\; X - axis.$
2. $\Theta - Rotation\, about\, the\, Y - axis.$
3. $\Phi - Rotation\, about\, the\; Z - axis.$
4. $t_x - Translation\, along\, the\; X - axis.$
5. $t_y - Translation\, along\, the\, Y - axis.$

6. $t_z - Translation\, along\, the\, Z - axis.$

4.2 The 2-D line model

The project for evaluating the 2-D straightness consists of three files. The files are named as per the convention mentioned earlier. Table 5 gives the inputs to the primary program.

1. main_prog_line.m
2. opti_func_line.m
3. line.txt

Table 5 Input to the primary program (2-D line model)

Equation of nominal geometry	$a_1x + b_1y + c_1 = 0$
Input set of measured points	"line.txt"
Bilateral tolerance limits	d_{in} =-1.5 and d_{out} = 1

Fig. 17 shows the nominal 2-D line and the tolerance zone constructed by offsetting (expanding and shrinking) the nominal boundary (line) based on the bilateral tolerance limits. The offset values are: d_{in} = -1.5 and d_{out} = 1. The data is collected by simulating the line model in CAD software and generating 21 points. The rigid body transformations are determined by three methods – the zone-fitting method proposed by Choi and Kurfess (1999a, 1999b), the least squares fit and the new zone-fitting algorithm proposed in the present study. The rotational transformation parameters are in radians while the translational parameters are non-dimensional. Since the line model is defined in 2-D space, not all transformation parameters are evaluated. Only, translation along the X and Y axes and the rotation about the Z-axis is permitted. Table 6 gives the transformation variables, which place the set of points in the tolerance zone, determined by the three different methods.

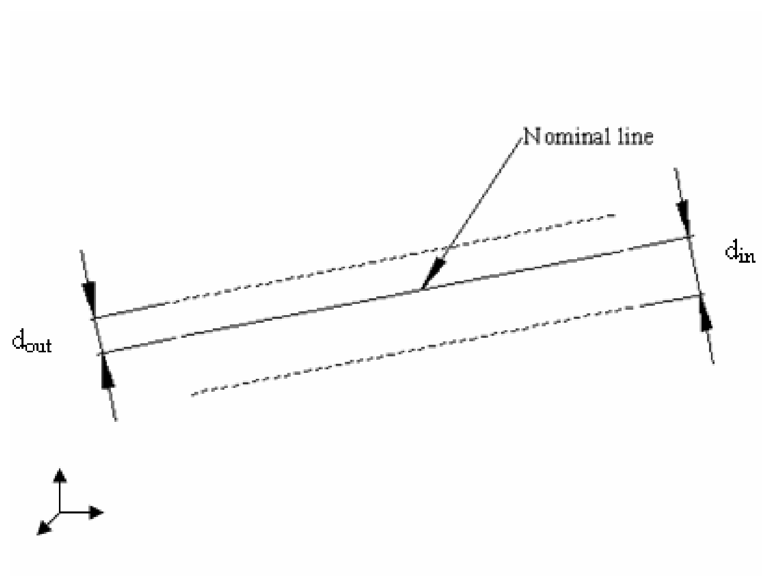

Fig. 17 2-D line model

Table 6 Transformation variables for 2-D line model

Transformation variables	**Zone-fitting (Choi and Kurfess)**	**Least Squares Fit**	**New zone-fitting algorithm**
Φ (radians)	-0.1958	0.034598	0.000
t_x	-0.00037206	0.000680	0.000
t_y	0.28727	0.055640	0.000

The residual deviations of the zone-fitting method proposed by Choi and Kurfess (1999a, 1999b), the least squares fit and the new zone-fitting algorithm are shown in Figs. 18, 19 and 20, respectively. The residual is the distance of a point from the nominal or the fitted geometry. The dotted lines represent the corresponding tolerance zone boundaries while the solid lines represent the minimum zone. The decision whether the measured set of points satisfy the tolerance specifications is made based on the results yielded by the different fitting methods. The zone-fitting method proposed by Choi and Kurfess (1999a, 1999b) and the new zone-fitting algorithm fit the points in the tolerance zone but the least squares fit gives a decision that the 2-D line is out of tolerance.

The conflicting results help us conclude that the tolerance conformance definition must govern the selection of verification algorithm. If it is to be determined whether the points fit in the tolerance zone, zone-fitting algorithm must be employed.

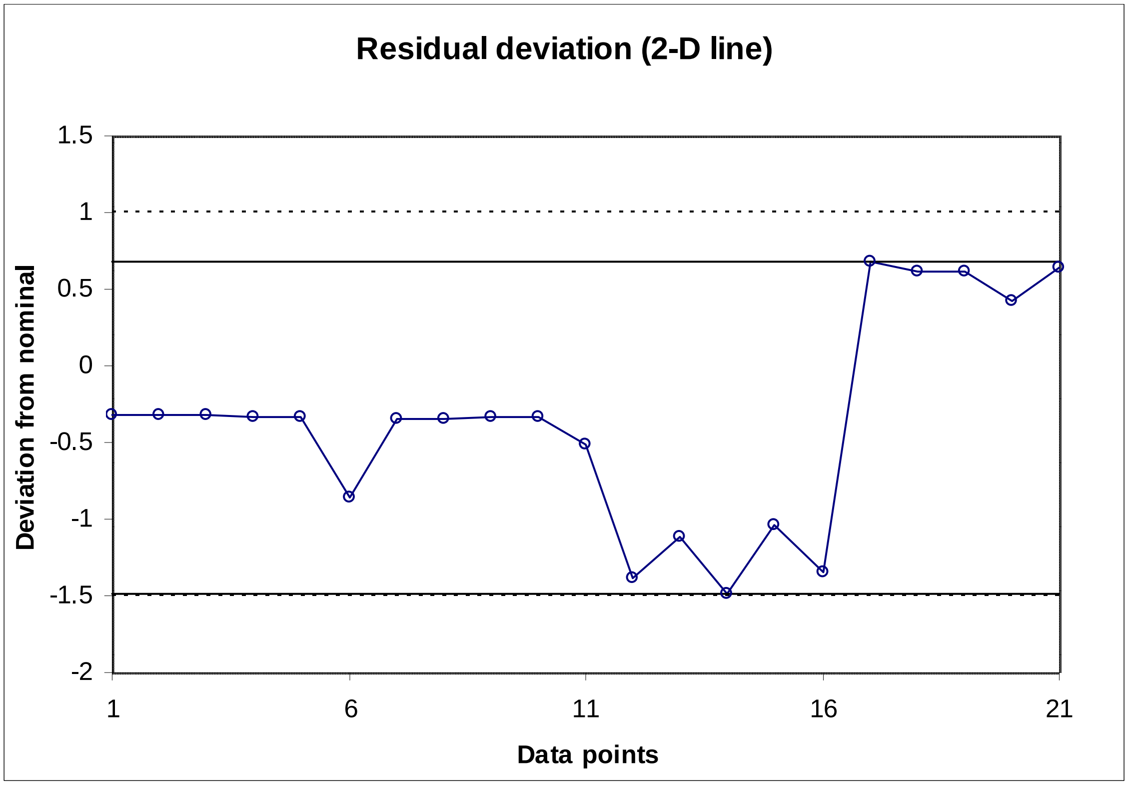

Fig. 18 Residual deviation after zone-fitting for 2-D line (Choi and Kurfess, 1999a, b)

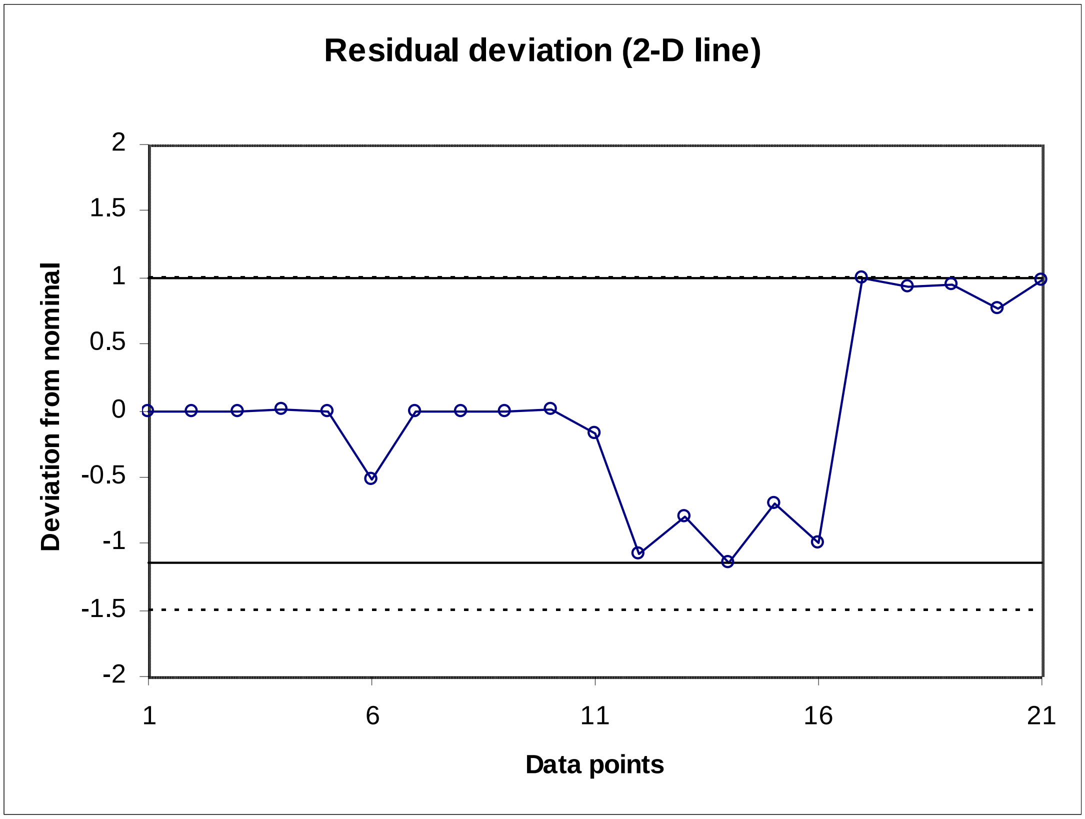

Fig. 19 Residual deviation after zone-fitting for 2-D line (new algorithm)

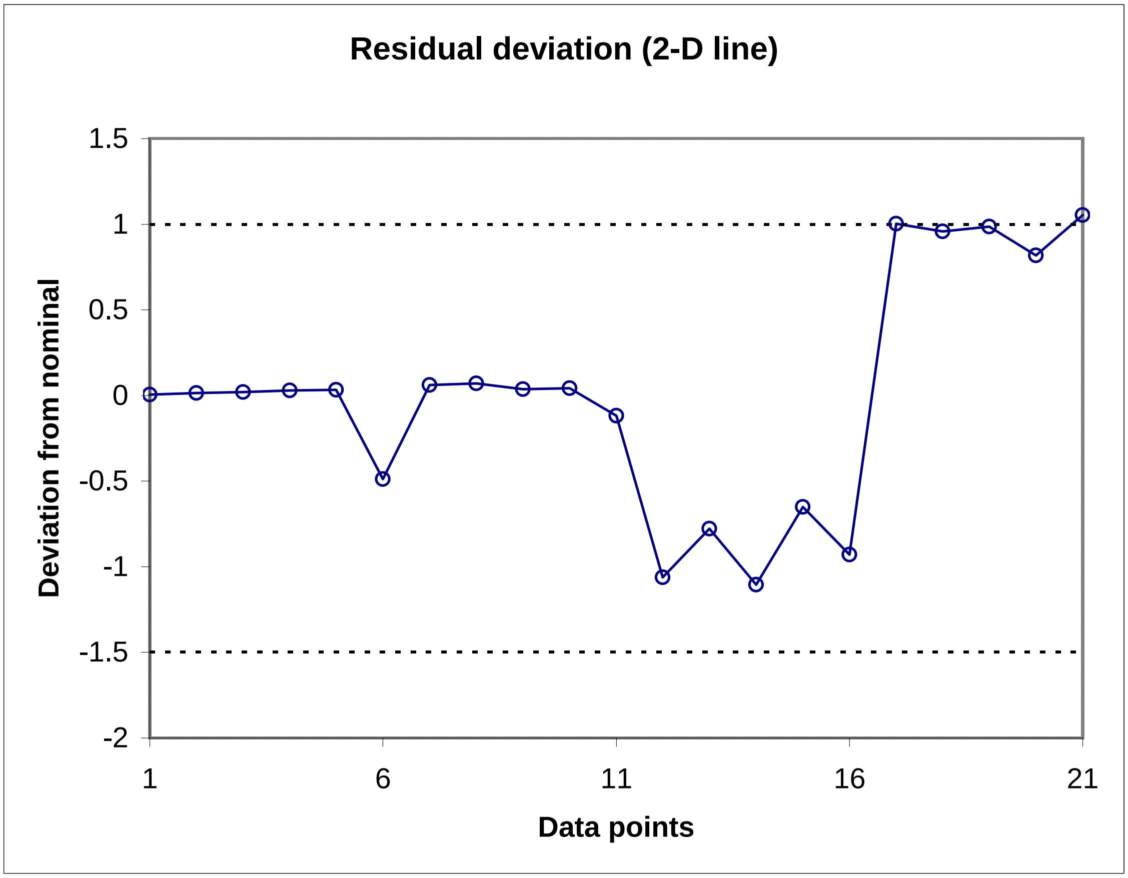

Fig. 20 Residual deviation after least squares fit for 2-D line

4.3 The 2-D circle model

The project for evaluating the 2-D roundness consists of three files. The files are named as per the convention mentioned earlier. Table 7 gives the input to the primary program.

1. main_prog_circle.m
2. opti_func_circle.m
3. circle.txt

Table 7 Input to the primary program (2-D circle model)

Equation of nominal geometry	$(x-a)^2+(y-b)^2=r^2$
Input set of measured points	"circle.txt"
Bilateral tolerance limits	d_{in} =-2 and d_{out} = 2

Fig. 21 shows the nominal 2-D circle and the tolerance zone constructed by offsetting (expanding and shrinking) the nominal boundary (circle) based on the bilateral tolerance limits. The radius of the circle in 25 units and the offset values are: d_{in} = -1.5 and d_{out} = 1.5. The data is collected by simulating the line model in CAD software and generating 21 points. The rigid body transformations are determined by three methods – the zone-fitting method proposed by Choi and Kurfess (1999a, 1999b), the least squares fit and the new zone-fitting algorithm proposed in the present study. The rotational transformation parameters are in radians while the translational parameters are non-dimensional. Since the circle model is defined in 2-D space, not all transformation parameters are evaluated. Only, translation along the X and Y axes and the rotation about the Z-axis is permitted. Table 8 gives the transformation variables, which place the set of points in the tolerance zone, determined by the three different methods.

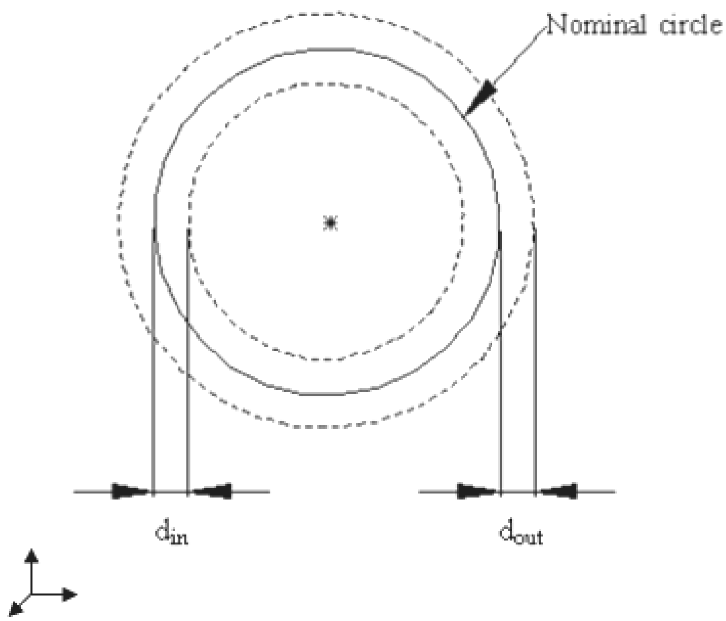

Fig. 21 2-D circle model

The residual deviations of the zone-fitting method proposed by Choi and Kurfess (1999a, 1999b), the least squares fit and the new zone-fitting algorithm are shown in Figs. 22, 23 and 24, respectively. The residual is the distance of a point from the nominal or the fitted geometry. The dotted lines represent the corresponding tolerance zone boundaries while the solid lines represent the minimum zone. The decision whether the measured set of points satisfy the tolerance specifications is made based on the results yielded by the different fitting methods. The zone-fitting method proposed by Choi and Kurfess (1999a, 1999b) and the new zone-fitting algorithm fit the points in the tolerance zone but the least squares fit gives a decision that the 2-D circle is out of tolerance.

The conflicting results help us conclude that the tolerance conformance definition must govern the selection of verification algorithm. If it is to be determined whether the points fit in the tolerance zone, zone-fitting algorithm must be employed.

Table 8 Transformation variables for 2-D circle model

Transformation variables	Zone-fitting (Choi and Kurfess)	New zone-fitting algorithm
Φ (radians)	0.00020792	-2.305687e-020
t_x	1.5086e-009	0
t_y	-0.0042413	0

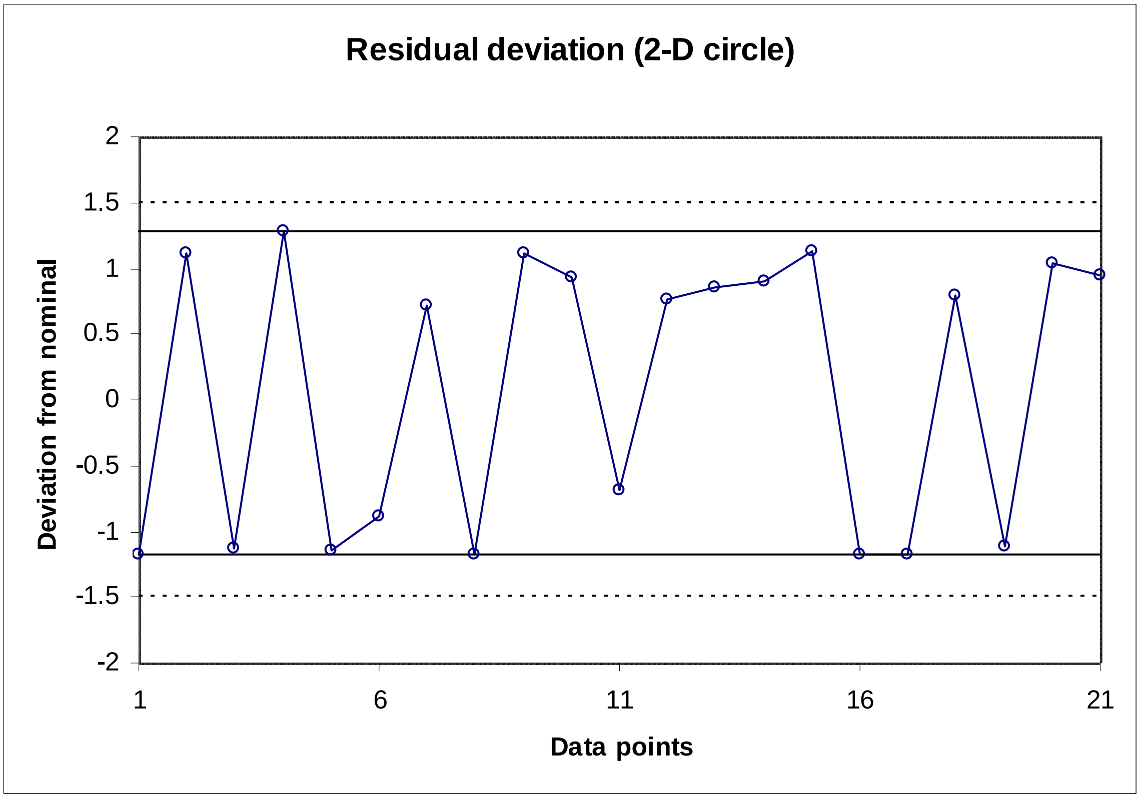

Fig. 22 Residual deviation after zone-fitting for 2-D circle (Choi and Kurfess, 1999a, b)

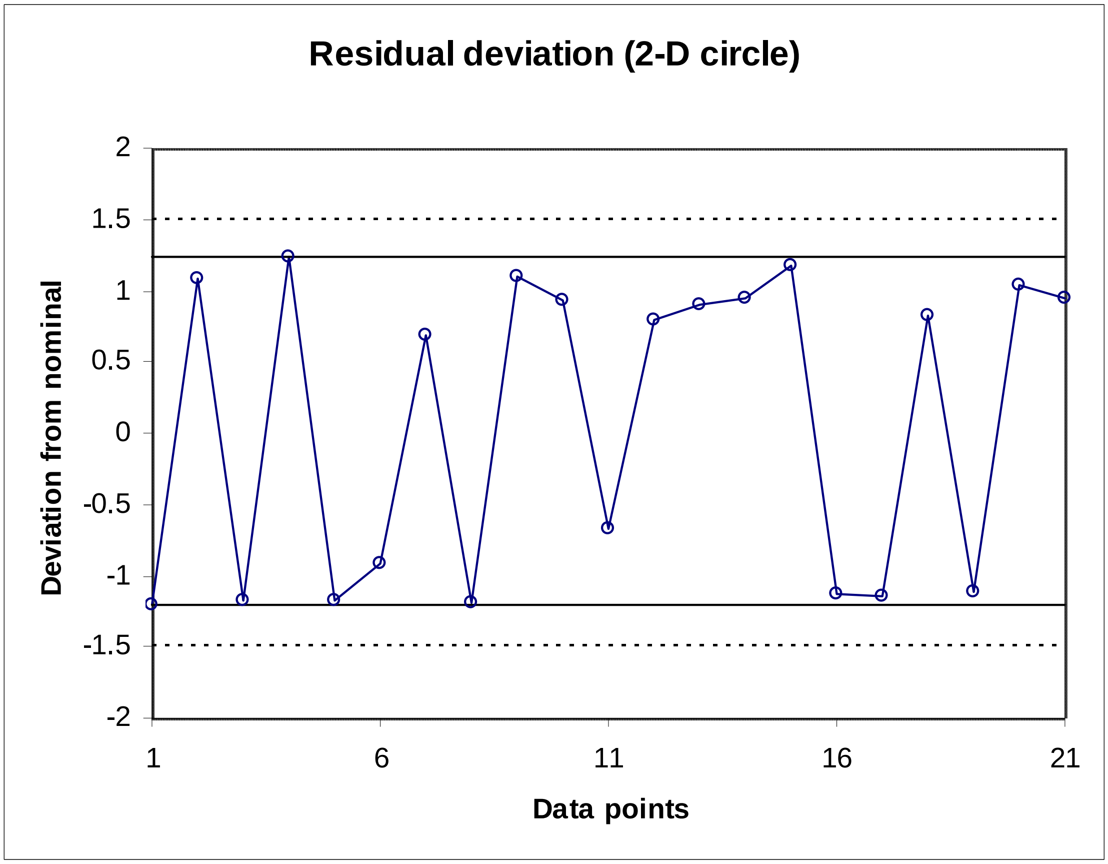

Fig. 23 Residual deviation after zone-fitting for 2-D circle (new algorithm)

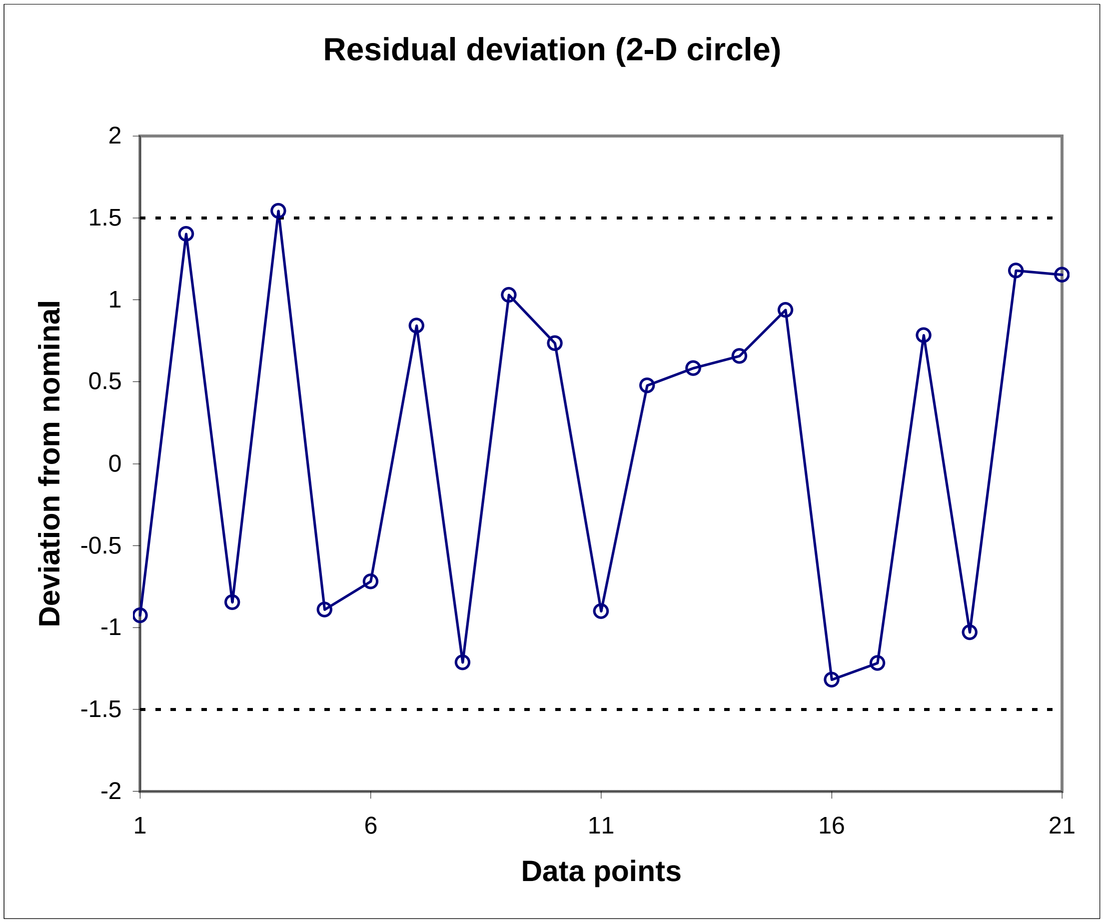

Fig. 24 Residual deviation after least squares fit for 2-D circle

4.4 The 3-D cylinder model

The project for evaluating the form tolerance of a 3-D cylinder consists of three files. The files are named as per the convention mentioned earlier. Table 9 gives the input to the primary program.

1. main_prog_cylinder.m
2. opti_func_cylinder.m

3. cylinder.txt

In the case of the 3-D cylinder, three nominal geometries are specified – the top plane, the bottom plane and the circle. The tolerance zones are individually constructed based on the respective bilateral tolerance specifications. For every point it is determined whether it lies in any of the tolerance zones. In this way the minimum zone values for each tolerance zone are determined.

Table 9 Input to the primary program (3-D cylinder model)

Equation of nominal geometry – Top plane, bottom plane and circle	$(x-a)^2+(y-b)^2=r^2$ $a_1x+b_1y+c_1z+d_1=0$ $a_2x+b_2y+c_2z+d_2=0$
Input set of measured points	"cylinder.txt"
Bilateral tolerance limits – different for each tolerance zone	d_{intp} = -2 and d_{outtp} = 2 d_{inbp} = -2 and d_{outbp} = 2 d_{inc} = -1.5 and d_{outc} = 1.5

Figs. 25 and 26 show the nominal 3-D cylinder and the tolerance zones constructed by offsetting (expanding and shrinking) the nominal boundaries (circular surface, top plane and bottom plane) based on the bilateral tolerance limits. Separate tolerance zones are constructed for each geometric feature. A point is said to satisfy the tolerance specifications if it fits in any one of the three tolerance zones. The radius of the cylinder is 10 units and the offset values for each of the three geometric features are given in Table 10.

Table 10 Bilateral tolerance values for the 3-D cylinder

	d_{in}	d_{out}
Circular surface	-1.5	1.5
Top plane	-2	2
Bottom plane	-2	2

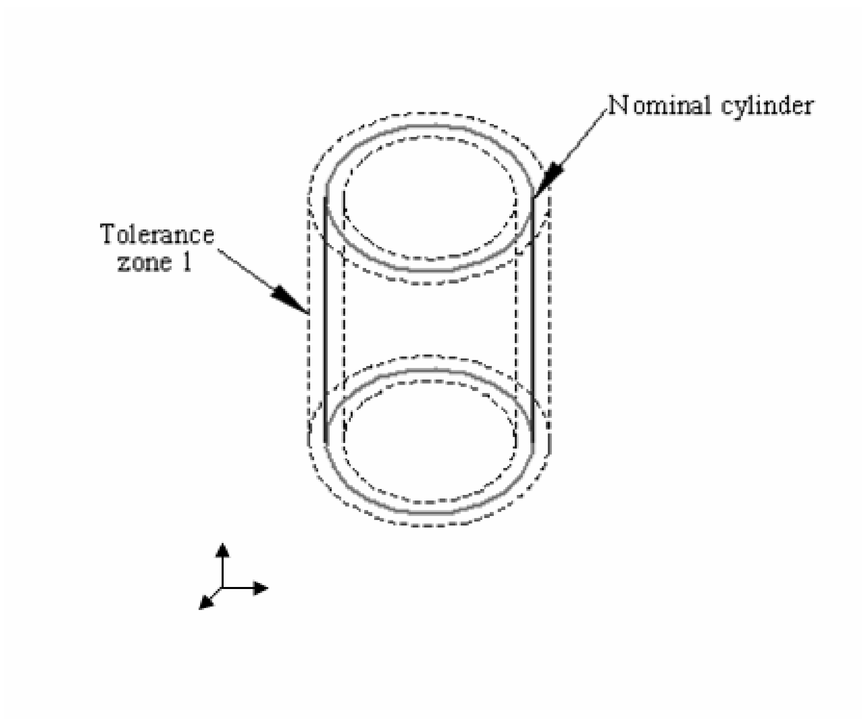

Fig. 25 Cylinder model (tolerance zone 1 – circular surface)

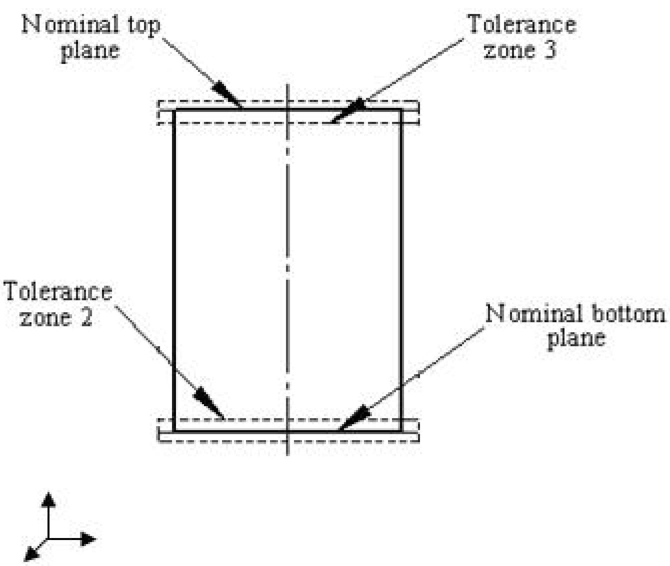

Fig. 26 Cylinder model (tolerance zone 2 – bottom plane and tolerance zone 3 – top plane)

The data is collected by simulating the 3-D cylinder model in CAD software and generating 44 points for the 3-D cylinder (circular surface, the top and bottom planes). The rigid body transformations are determined by two methods – the zone-fitting method proposed by Choi and Kurfess (1999a, 1999b) and the new zone-fitting algorithm proposed in the present study. The rotational transformation parameters are in radians while the translational parameters are non-dimensional. Table 11 gives the transformation variables, which place the set of points in the tolerance zone, determined by the two methods.

Table 11 Transformation variables for 3-D cylinder example

Transformation variables	Zone-fitting (Choi and Kurfess)	New zone-fitting algorithm
Ψ (radians)	0.0002374	0.0277
Θ (radians)	0.00075411	0
Φ (radians)	-0.003137	-0.0020
t_x	0.077277	-0.0671
t_y	0.017196	0.0323
t_z	-0.079719	-0.0342

The residual deviations of the zone-fitting method proposed by Choi and Kurfess (1999a, 1999b) and the new zone-fitting algorithm are shown in Figs. 27 and 28, respectively. The residual is the distance of a point from the nominal geometry. The dotted lines represent the corresponding tolerance zone boundaries while the solid lines represent the minimum zone. The decision whether the measured set of points satisfy the tolerance specifications is made based on the results yielded by the two zone-fitting methods. The zone-fitting method proposed by Choi and Kurfess (1999a, 1999b) and the new zone-fitting algorithm fit the points in the tolerance zone, but the minimum zone values evaluated are different. Table 12 gives the minimum zone values calculated by the two methods. The values given by the new zone-fitting algorithm are higher than that of the zone-fitting method proposed by Choi and Kurfess (1999a, 1999b). The difference in the results can be attributed to the different objective functions. The method proposed by Choi and Kurfess (1999a, 1999b) relies on specifying a convergence tolerance, while the new zone-fitting algorithm doesn't set a convergence tolerance. With the Boolean objective function employed by the new zone-fitting algorithm, the ambiguity whether the point lies inside or outside the tolerance zone is completely eliminated.

Table 12 Minimum zone values for the 3-D cylinder example

	Zone-fitting (Choi and Kurfess)			**New zone-fitting algorithm**		
	Circular surface	Top plane	Bottom plane	Circular surface	Top plane	Bottom plane
Min d_{in}	-1.0664	-1.5664	-1.707	-1.7078	-1.7182	-1.6537
Min d_{out}	1.1914	1.582	1.6289	1.1517	1.6514	1.7182
Min tol. Zone	2.2578	3.1484	3.3359	2.8595	3.3696	3.3719

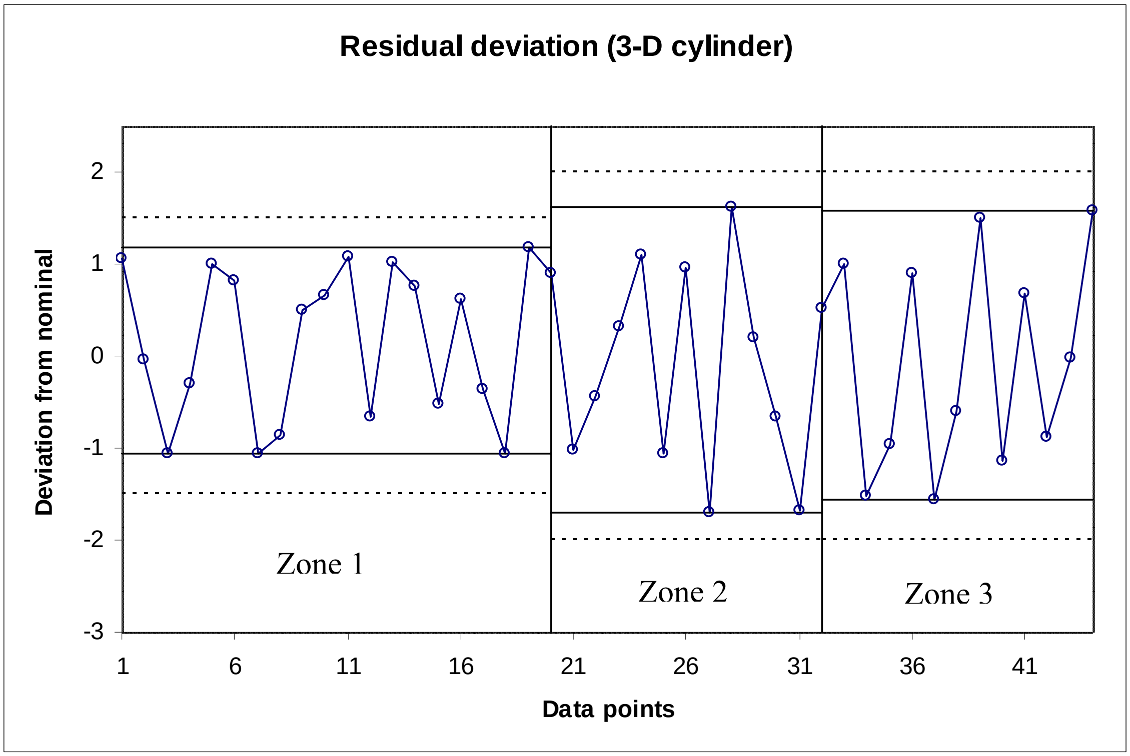

Fig. 27 Residual deviation after zone-fitting for 3-D cylinder (Choi and Kurfess, 1999a, b)

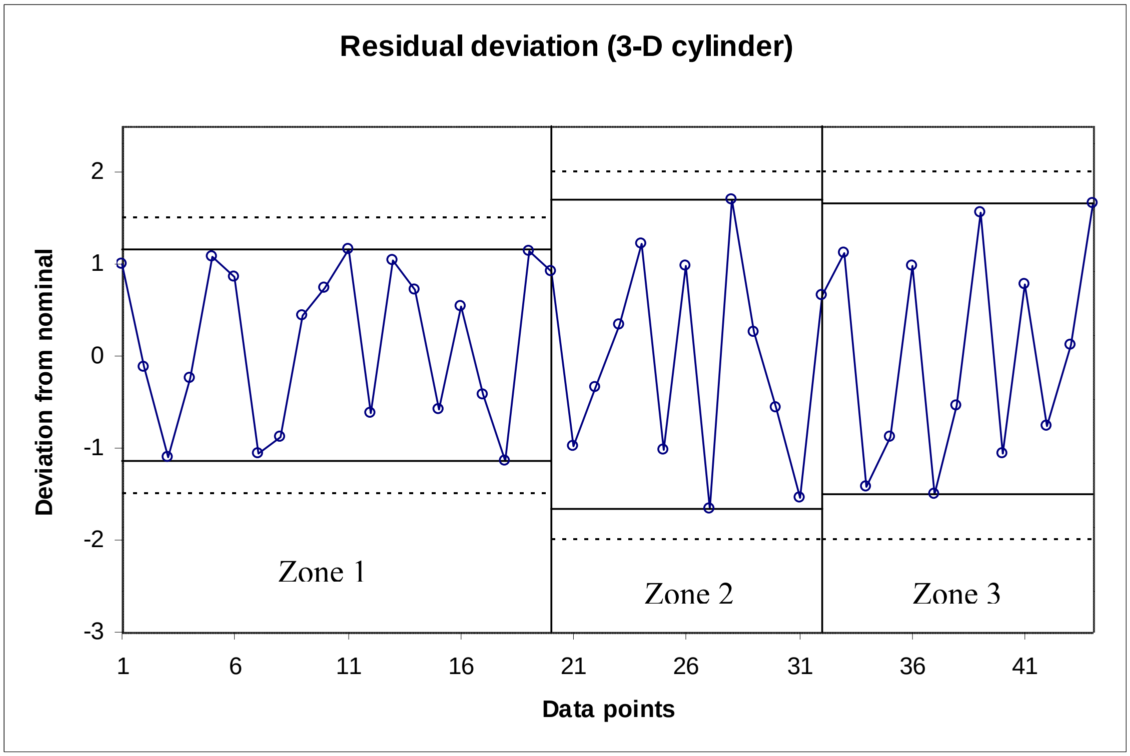

Fig. 28 Residual deviation after zone-fitting for 3-D cylinder (new algorithm)

4.5 The turbine blade model

The new zone-fitting algorithm is utilized to evaluate the form tolerance on the complex 3-D surface geometry. Table 13 gives the input to the primary program. Fig. 29 shows the 3-D model of the turbine blade. One of the current practices followed in the industry for its dimensional inspection is by comparing it with the master blade. The master blade is secured in a fixture, and then dial gauges are used – one each for the leading and trailing edges, two gauges on the suction surface and two on the pressure surface. The readings on the dial gauges are noted and the master blade is removed. Now the blade under inspection is placed in the fixture and with similar arrangement of the dial gauges the readings on each are noted. If the readings are within the tolerance specifications then the blade is determined as good. This method is very inadequate since it essentially assesses the tolerance at a very few number of points, thus failing to give the overall assessment of the blade surface. Fig. 30 shows the tolerance zone for a single airfoil section.

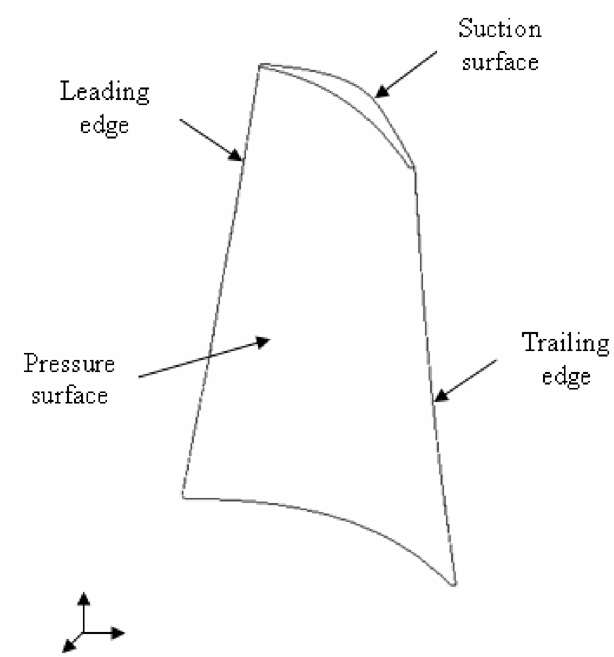

Fig 29 3-D model of the turbine blade

Table 13 Input to the primary program (3-D turbine blade model)

Equation of nominal geometry – set of five surface functions for each section	$(x-a_{LE})^2+(y-b_{LE})^2=r_{LE}{}^2$ $(x-a_{TE})^2+(y-b_{TE})^2=r_{TE}{}^2$ $(x-a_C)^2+(y-b_C)^2=r_C{}^2$ $y=a_1x^3+a_2x^2+a_3x+a_4$ $y=b_1x^3+b_2x^2+b_3x+b_4$
Input set of measured points	"blade<n>.txt"
Bilateral tolerance limits – same for all the sections	d_{in} = -0.0002 and d_{out} = 0.0002

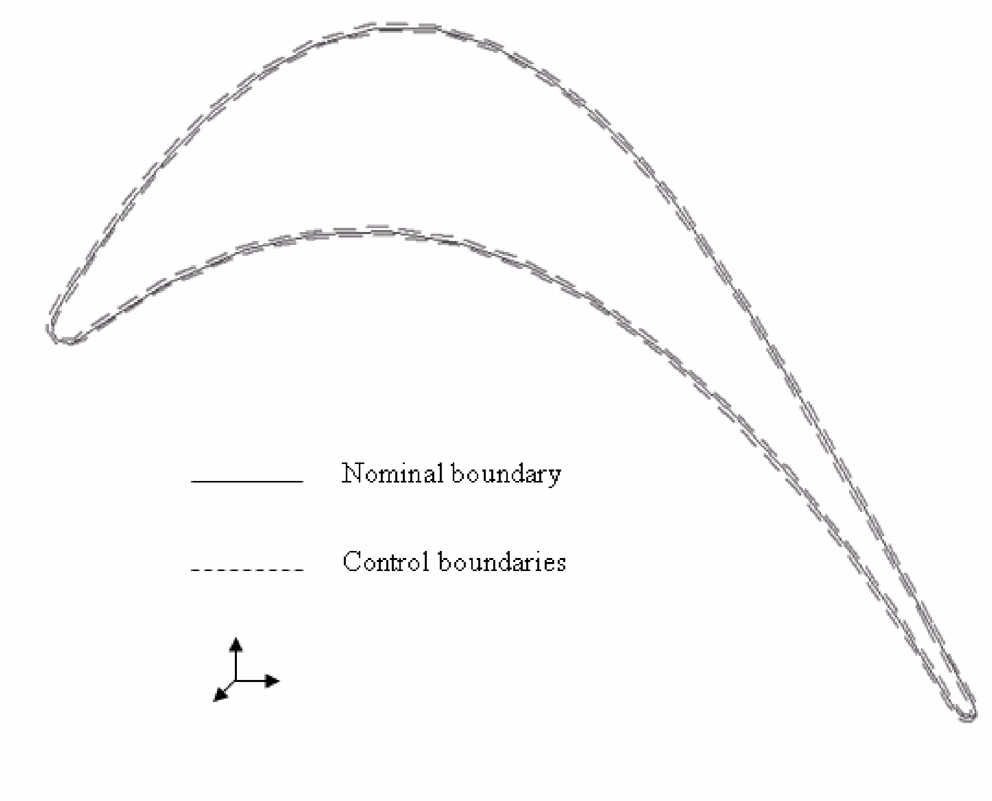

Fig. 30 Tolerance zone for single airfoil section

The data is collected by simulating the 3-D turbine blade model in CAD software and generating 20 points for each airfoil section. A total of six sections determine the complete 3-D turbine blade. The bilateral tolerance values specified are d_{in} = -0.0002 and d_{out} = 0.0002. The rigid body transformations parameters are determined by the new zone-fitting algorithm proposed in the present study. The rotational transformation parameters are in radians while the translational parameters are non-dimensional. Since the circle model is defined in 2-D space, not all transformation parameters are evaluated. Only, translation along the X and Y axes and the rotation about the Z-axis is permitted. The minimum zone value calculated and the transformation that places the points in the respective zones simultaneously are given in Tables 14 and 15 respectively.

Table 14 Minimum zone values for the 3-D turbine blade example

Min d_{in}	-0.00015547
Min d_{out}	0.00015547
Min tolerance zone	0.00031094

Table 15 Transformation variables for 3-D turbine blade example

Transformation variables	New zone-fitting algorithm
Φ (radians)	4e-006
t_x	0
t_y	-2.34e-007

4.6 Tolerance assessment using actual data

In this example, the zone-fitting method proposed by Choi and Kurfess (1999a, 1999b) and the new zone-fitting algorithm is employed to evaluate the form tolerance of the circular surface of a cylinder. The data used for this is measured from the circular rod of nominal diameter 19.101 mm (Fig. 31). The offset values are: d_{in} = 0 and d_{out} = 2.54 mm. Such unilateral tolerances are specified for assembly situations where a minimum hole diameter with a maximum shaft diameter is specified.

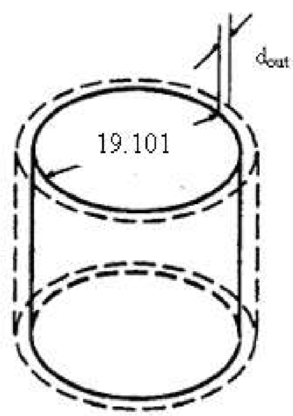

Fig. 31 Cylindrical rod model

The transformations determined by the different zone-fitting methods are given in Table 16 and the residual deviations are shown in Figs. 33 and 34. The dotted lines represent the corresponding tolerance zone boundaries while the solid lines represent the minimum zone. Since the cylinder does not have measured points on the top and bottom surface, t_z has not been determined. Also Φ is not determined due to the symmetry about the *Z* axis.

Table 16 Transformation variables for cylindrical model

Transformation variables	Zone-fitting (Choi and Kurfess)	New zone-fitting algorithm
Ψ (radians)	-0.003115	-0.00000
Θ (radians)	-0.010452	0.00000
t_x	-0.031556	-0.0267
t_y	0.032872	0.0407

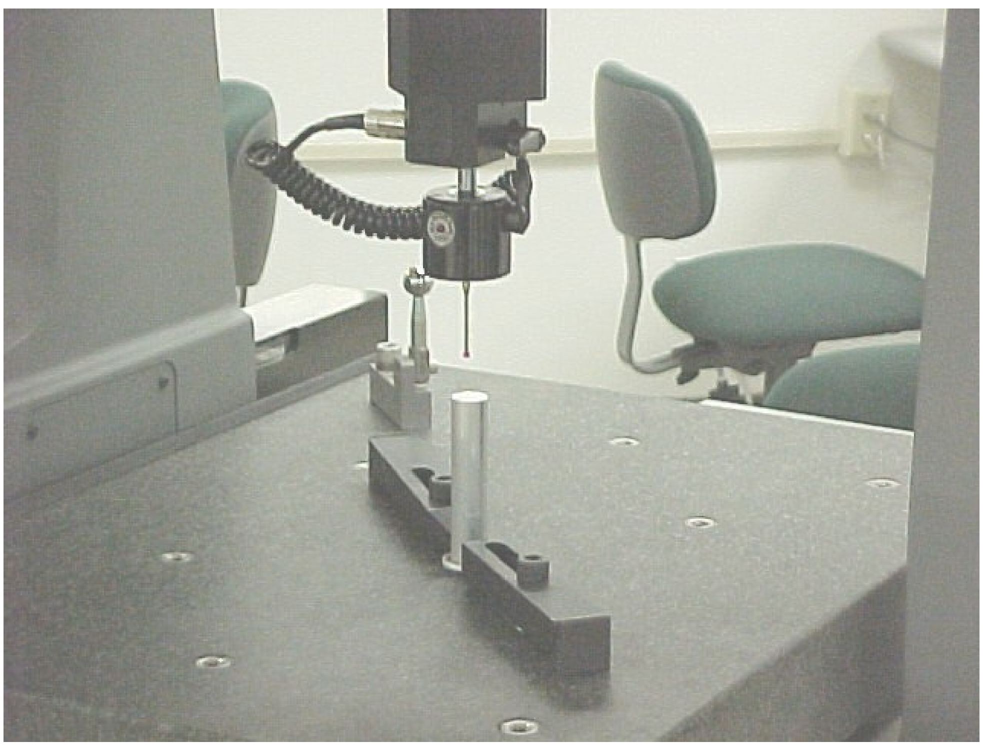

Fig. 32 Setup for coordinate measurement

Fig. 32 shows the setup used to collect data from the circular surface of the cylindrical rod. The minimum zone values calculated by the zone-fitting method proposed by Choi and Kurfess (1999a, 1999b) and the new zone-fitting algorithm are 2.123287 mm and 2.22603 mm respectively. The value calculated by the zone-fitting method proposed by Choi and Kurfess (1999a, 1999b) is higher than that calculated by the new zone-fitting algorithm.

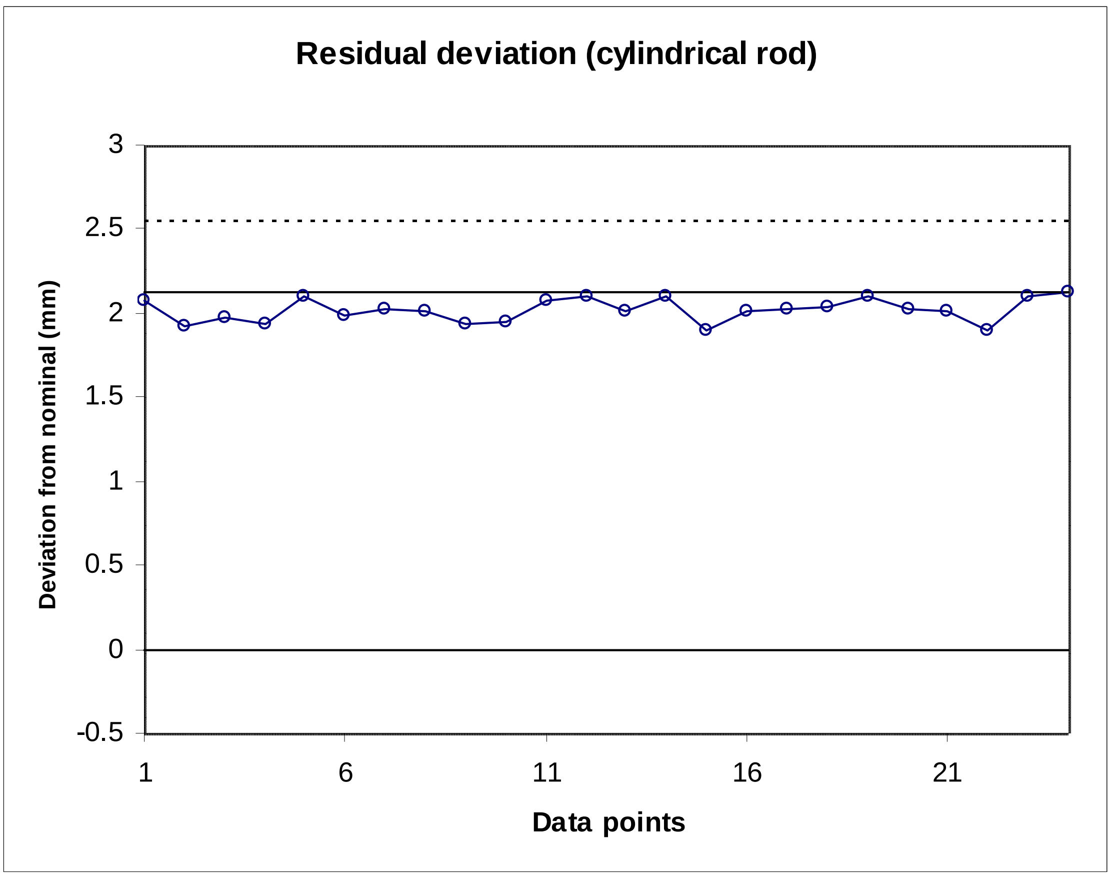

Fig. 33 Residual deviation after zone-fitting for cylindrical rod (Choi and Kurfess, 1999a, b)

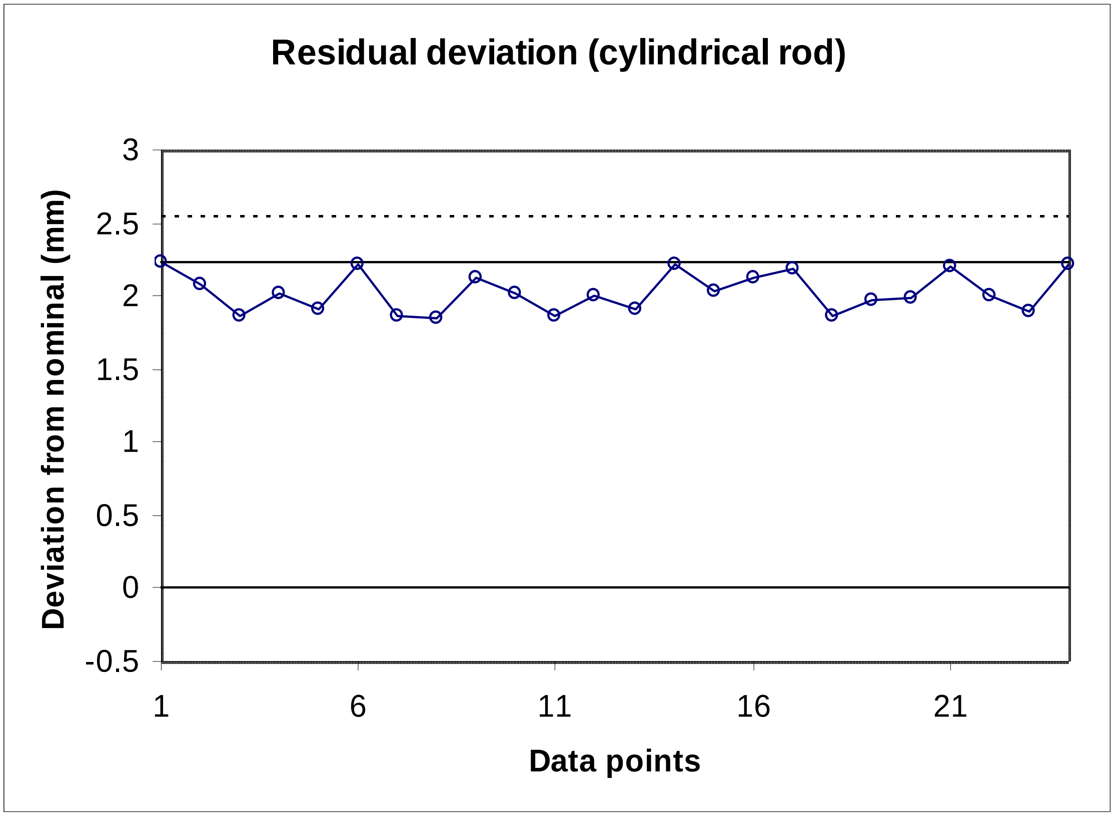

Fig. 34 Residual deviation after zone-fitting for cylindrical rod (new algorithm)

5. SUMMARY AND CONCLUSION

A new zone-fitting algorithm was developed to evaluate the form tolerances of geometric features. It determines the rigid body transformation that places the set of points measured from the actual surface in the specified tolerance zone. The search for the transformation parameters (three translational and three rotational) is modeled as an unconstrained nonlinear optimization problem. The objective function is defined in such a manner that no convergence tolerance needs to be set. Thus it eliminates the ambiguity whether a point is in or out of the tolerance zone. Given the nominal surfaces, the developed algorithm evaluates if the measured points lie in the specified tolerance limits, and further determines the minimum zone in which the measured set of points lies. This provides vital information as to the actual part quality based on which the manufacturing process is adjusted.

The developed algorithm is employed to evaluate the form tolerance of a 2-D line, 2-D circle, 3-D cylinder and the 3-D turbine blade. The results are compared to that of least squares fit and the method proposed by Kurfess and Choi (1999). From the results it can be inferred that zone-fitting algorithms are better suited when validating a tolerance zone specification. However, the results of the fitting algorithm must be treated carefully as only a finite number of points can be measured from the unknown actual surface. The measured points are considered exact copy of the actual surface. The new zone-fitting algorithm has the following advantages:

1. The new algorithm employs the Boolean function, which by virtue of its definition eliminates the ambiguity whether a point is in or out of the tolerance zone.
2. The new algorithm evaluates the bilateral minimum zone values for the features under inspection. Thus the manufacturing-process supervisor has a better grasp of the direction the process is heading based on the minimum zone values for that particular feature.
3. The new algorithm is implemented for assessing the tolerance of a complex 3-D objects (such as a turbine blade). For these 3-D objects, the inspection standards are often specified as a collection of 2-D cross-sections. By constraining the transformation parameters, the proposed methodology will determine whether the points measured at the 2-D cross-sections fit in the respective tolerance zones simultaneously.

REFERENCES

ASME Y14.5M-1994, 1994 National Standard on Dimensioning and Tolerancing, American Society of Mechanical Engineers, New York, NY.

Carr, K., and Ferreira, P., 1995a, "Verification of Form Tolerances Part I: Basic Issues, Flatness and Straightness," *Precision Engineering,* Vol. 17, No. 2, 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, No. 2, pp. 144-156.

Choi, W., and Kurfess, T. R., 1999a, "Dimensional Measurement Data Analysis, Part 1: A Zone Fitting Algorithm," *Journal of Manufacturing Science and Engineering,* Vol. 121, pp. 238-245.

Choi, W., and Kurfess, T. R., 1999b, "Dimensional Measurement Data Analysis, Part 2: Minimum Zone Evaluation," *Journal of Manufacturing Science and Engineering,* Vol. 121, pp. 246-250.

Etesami, F., and Qiao, H., 1990, "Analysis of Two-dimensional Measurement Data for Automated Inspection," *Journal of Manufacturing Systems,* Vol. 9, pp. 21-34.

Hong, J. T., and Fan, K. C., 1986, "An Algorithm for Straightness Calculation from Geometrical Viewpoint," *Proc. of 1st ROC-ROK Metrology Standard Symposium,* Taipei, Taiwan, pp. 103-110.

Huang, S. T., Fan, K. C., and Wu, J. H., 1993a, "A New Minimum Zone Method for Evaluating Straightness Errors," *Precision Engineering,* Vol.15, No. 3, pp. 158-165.

Huang, S. T., Fan, K. C., and Wu, J. H., 1993b, "A New Minimum Zone Method for Evaluating Flatness Errors," *Precision Engineering,* Vol. 15, No.1, pp. 25-32.

Kanada, T., and Suzuki, S., 1993a, "Evaluation of Minimum Zone Flatness by Means of Nonlinear Optimization Techniques and Its Verification," *Precision Engineering,* Vol. 15, No. 2, pp. 93-99.

Kanada, T., and Suzuki, S., 1993b, "Application of Several Computing Techniques for Minimum Zone Straightness," *Precision Engineering,* Vol. 15, No. 4, pp. 274-280.

Lai, K., and Wang, J., 1988, "A Computational Geometry Approach to Geometric Tolerancing," *16th North American Manufacturing Research Conference,* pp. 376-379.

Murthy, T. S. R., and Abdin, S. Z., 1980, "Minimum Zone Evaluation of Surfaces," *International Journal of Machine Tool Design and Research,* Vol. 20, pp. 123-136.

Preparata, F. P., and Shamos, M.I., 1988, *Computational Geometry* (1st Ed.), Springer, Berlin, pp. 41-44.

Pritchard, L. J., 1985, "An Eleven Parameter Axial Turbine Airfoil Geometry Model," ASME Paper No. 85-GT-219.

Requicha, A. A. G., 1983, “Toward a Theory of Geometric Tolerancing,” *International Journal of Robotics Research,* Vol. 2, No. 4, pp. 45-60.

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.

Shunmugam, M. S., 1987, “New Approach for Evaluating Form Errors of Engineering Surfaces,” *Computer-Aided Design,* Vol. 19, No. 7, pp. 368-374.

Swanson, K., Lee, D. T., and Wu, V. L., 1995, “An Optimal Algorithm for Roundness Determination on Convex Polygons,” *Computational Geometry,* Vol. 5, No. 4, pp. 225-235

Traband, M. T., Joshi, S., Wysk, R. A., and Cavalier, T. M., 1989, “Evaluation of Straightness and Flatness Tolerances Using Minimum Zone,” *Manufacturing Review,* Vol. 2, pp. 189-195.

Tsukada, T., and Kanada, T., 1985, “Minimum Zone Evaluation of Cylindricity Deviation by Some Optimization Techniques,” *Bull. Japan Society of Precision Engineering,* Vol. 19, No. 1, pp. 18-23.

Wang, Y., 1992, “Minimum Zone Evaluation of Form Tolerances,” *Manufacturing Review,* Vol. 5, No. 3, pp. 213-220.

APPENDIX A

A.1 Least squares fit method

The least squares fit is extensively used as the verification algorithm in the coordinate measuring machines. In the least squares fit, a mathematical procedure for finding the best-fitting curve to a given set of points by minimizing the sum of the squares of the offsets ("the residuals") of the points from the curve is employed. The sum of the squares of the offsets is used instead of the offset absolute values because this allows the residuals to be treated as a continuous differentiable quantity. Least squares fitting proceeds by finding the sum of the squares of the deviations R^2 of a set of n data points from a function f.

$$R^2 \equiv \sum [y_i - f(x_i, a_1, a_2, \ldots, a_n)]^2 \qquad \text{(A.1)}$$

The function f can be a polynomial of any order, as per the desired fitting accuracy. Although the unsquared sum of distances might seem a more appropriate quantity to minimize, use of the absolute value results in discontinuous derivatives, which cannot be treated analytically. The square deviations from each point are therefore summed, and the resulting residual is then minimized to find the best-fit curve. This procedure results in outlying points being given disproportionately large weighting. The condition for R^2 to be a minimum is that

$$\frac{\partial(R^2)}{\partial a_i} = 0 \tag{A.2}$$

for i = 1…n. If we consider f as a linear function i.e. a linear fit, we have

$$f(a, b) = a + bx, \tag{A.3}$$

so

$$R^2(a, b) \equiv \sum_{i=1}^{n} [y_i - (a + bx_i)]^2 \tag{A.4}$$

$$\frac{\partial(R^2)}{\partial a} = -2 \sum_{i=1}^{n} [y_i - (a + bx_i)] = 0 \tag{A.5}$$

$$\frac{\partial(R^2)}{\partial b} = -2 \sum_{i=1}^{n} [y_i - (a + bx_i)] x_i = 0. \tag{A.6}$$

These lead to the equations

$$na + b\sum_{i=1}^{n} x_i \quad = \quad \sum_{i=1}^{n} y_i \tag{A.7}$$

$$a\sum_{i=1}^{n} x_i + b\sum_{i=1}^{n} x_i^2 \quad = \quad \sum_{i=1}^{n} x_i y_i. \tag{A.8}$$

In matrix form,

$$\begin{bmatrix} n & \sum_{i=1}^{n} x_i \\ \sum_{i=1}^{n} x_i & \sum_{i=1}^{n} x_i^2 \end{bmatrix} \begin{bmatrix} a \\ b \end{bmatrix} = \begin{bmatrix} \sum_{i=1}^{n} y_i \\ \sum_{i=1}^{n} x_i y_i \end{bmatrix}, \tag{A.9}$$

Thus,

$$\begin{bmatrix} a \\ b \end{bmatrix} = \begin{bmatrix} n & \sum_{i=1}^{n} x_i \\ \sum_{i=1}^{n} x_i & \sum_{i=1}^{n} x_i^2 \end{bmatrix}^{-1} \begin{bmatrix} \sum_{i=1}^{n} y_i \\ \sum_{i=1}^{n} x_i y_i \end{bmatrix}. \tag{A.10}$$

The values of *a* and *b* in equation A.10 are substituted to equation A.3 to yield the best fit curve for the given data. The deviations from this best fit curve are calculated to obtain the form tolerance.

APPENDIX B

B.1 BFGS Quasi-Newton method

Central idea underlying quasi-Newton methods is to use an approximation of the inverse Hessian. However, form of approximation differs with different methods. The quasi-Newton methods that build up an approximation of the inverse Hessian are often regarded as the most sophisticated for solving unconstrained problems. In quasi-Newton methods, instead of the true Hessian, an initial matrix H_0 is chosen (usually H_0 = I) which is subsequently updated by an update formula:

$$H_{k+1} = H_k + H_{ku}$$

where, H_{ku} is the update matrix. This updating can also be done with the inverse of the Hessian H^{-1}. Let $B = H^{-1}$; then the updating formula for the inverse is also of the form is given by:

$$B_{k+1} = B_k + B_{ku}$$

Given two points x_k and x_{k+1}, we define

$$g_k = \nabla y(x_k) \text{ and } g_{k+1} = \nabla\, y(x_{k+1})$$

Further, let $p_k = x_{k+1} - x_k$, then

$$g_{k+1} - g_k \approx H(x_k)\, p_k$$

If the Hessian is constant, then

$$g_{k+1} - g_k = H\, p_k \quad \text{which can be rewritten as} \quad q_k = H\, p_k$$

If the Hessian is constant, then the following condition would hold as well

$$H^{-1}_{k+1}\, q_i = p_i \qquad\qquad 0 \le i \le k$$

This is called the quasi-Newton condition. Remember that

$$q_i = H_{k+1}\, p_i \;\text{ and } H^{-1}_{k+1}\, q_i = p_i \;\text{ (or, } B_{k+1}\, q_i = p_i) \quad 0 \le i \le k$$

Both equations have exactly the same form, except that q_i and p_i are interchanged and H is replaced by B (or vice versa). This leads to the observation that any update formula for B can be transformed into a corresponding complimentary formula for H by interchanging the roles of B and H and of q and p. The reverse is also true. Broyden–Fletcher–Goldfarb–Shanno (BFGS) formula update of H_k is obtained by taking the complimentary formula of the DFP formula, thus:

$$H_{k+1} = H_k + \frac{q_k q_k^T}{q_k^T p_k} - \frac{H_k p_k p_k^T H_k}{p_k^T H_k p_k}$$

By taking the inverse, the BFGS update formula for B_{k+1} (i.e., H^{-1}_{k+1}) is obtained:

$$B_{k+1} = B_k + \left(\frac{1 + \mathrm{qkTBkqk}}{\mathrm{qkTpk}}\right)\frac{p_k p_k^T}{p_k^T q_k} - \frac{p_k q_k^T B_k + B_k q_k p_k^T}{q_k^T p_k}$$

APPENDIX C

C.1 Point location method (Preparata and Shamos, 1988)

In the point location method, the convex polygon **C** is partitioned into *n* wedges by the rays as shown in the Fig. 3.2. Each wedge is divided into two pieces by a single edge of **C**. One of these pieces is wholly internal to **C**, while the other is wholly external. Let *q* be the point internal to **C** and since the rays occur in angular order, the wedge in which the point *z* lies can be found by a single binary search. The procedure is outlined in 2 simple steps:

1. The point *z* lies between the rays defined by p_i and p_{i+1}, if $\angle zqp_{i+1}$ is a right turn and $\angle zqp_i$ is a left turn.
2. Once p_i and p_{i+1} are found, then z is internal to **C** only if $\angle \mathrm{p_i}\, p_{i+1} z$ is a left turn.

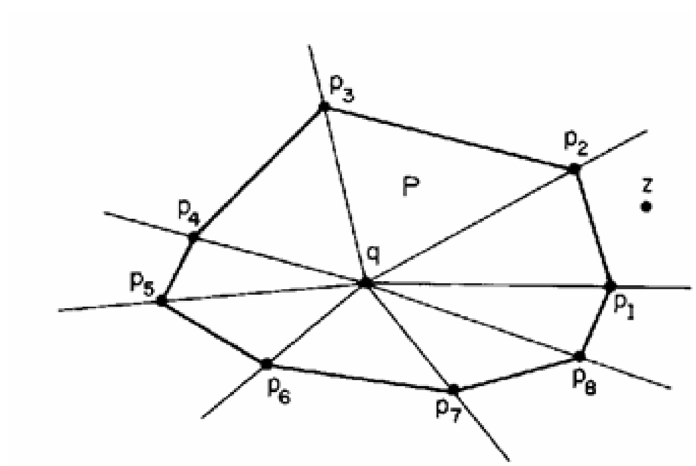

Fig. 35 Point location method (Preparta and Shamos, 1988)

To decide whether an angle makes a left or a right turn, a 3 x 3 determinant in the points' coordinates is evaluated. Let $p_i = (x_i, y_i)$, then the determinant for $\angle p_1 p_2 p_3$ is given as follows:

$$\begin{vmatrix} x_1 & y_1 & 1 \\ x_2 & y_2 & 1 \\ x_3 & y_3 & 1 \end{vmatrix}$$

The determinant gives twice the signed area of the triangle ($p_1p_2p_3$). The area is positive if and only if the angle $p_1p_2p_3$ makes a left turn and negative if the angle $p_1p_2p_3$ makes a right turn.

APPENDIX D

The nomenclature used in the thesis is as follows:

1. S – any geometric surface/curve in 3-D space
2. P – set of measured points from the actual surface
3. H – homogeneous transformation matrix
4. R – Euler matrix
5. t – translation vector
6. $\Psi - Rotation\ about\ the\ X - axis.$
7. $\Theta - Rotation\ about\ the\ Y - axis.$
8. $\Phi - Rotation\ about\ the\ Z - axis.$
9. $t_x - Translation\ along\ the\ X - axis.$
10. $t_y - Translation\ along\ the\ Y - axis.$
11. $t_z - Translation\ along\ the\ Z - axis.$
12. P* - transformed set of points
13. N – Boolean function
14. p – point vector
15. p^a – augmented point vector
16. T – tolerance zone
17. d_{in} – inner tolerance limit
18. d_{out} – outer tolerance limit

19. d_{inL} – lower limit for d_{in}

20. d_{inU} – upper limit for d_{in}

21. d_{outL} – lower limit for d_{out}

22. d_{outU} – upper limit for d_{out}

23. fval – minimum state of the objective function returned after optimization

24. x0 – initial estimate for the optimization solution

25. x – numerical epsilon used to terminate the binary search

VITA

Nachiket Vishwas Pendse obtained his Bachelor of Engineering degree in Mechanical Engineering from Government College of Engineering, Pune – India. In the Fall of 2002 he began studying for his Master of Science degree in Mechanical Engineering at Texas A&M University and received it in December 2004.

Nachiket Vishwas Pendse

Parambi, 378/28,

Pradhikaran, Pune – 411044

Maharashtra, India