Uncertainty Calculation of Roundness Assessment by Automatic Differentiation in Coordinate Metrology

Jia-chun LIN(林家春)[1], Michael Krystek[2], Zhao-yao SHI(石照耀)[1]
(1. *College of Mechanical Engineering and Applied Electronics Technology*, *Beijing University of Technology*, *Beijing* 100124, *China*;
2. *Physikalisch-Technische Bundesanstalt*, *Braunschweig D*-38116, *Germany*)

Abstract – Recently, Coordinate Measuring Machines (CMMs) are widely used to measure roundness errors. Roundness is calculated from a large number of points collected from the profiles of the parts. According to the Guide to the Expression of Uncertainty in Measurement (GUM), all measurement results must have a stated uncertainty associated the them. However, no CMMs give the uncertainty value of the roundness, because no suitable measurement uncertainty calculation procedure exists. In the case of roundness measurement in coordinate metrology, this paper suggests the algorithms for the calculation of the measurement uncertainty of the roundness deviation based on the two mainly used association criteria, LSC and MZC. The calculation of the sensitivity coefficients for the uncertainty calculation can be done by automatic differentiation, in order to avoid introducing additional errors by the traditional difference quotient approximations. The proposed methods are exact and need input data only as the measured coordinates of the data points and their associated uncertainties.

Key words – *measurement uncertainty*; *roundness*; *automatic diferentiation*

Manuscript Number: 1674-8042(2010)03-0224-04

dio: 10.3969/j.issn.1674-8042.2010.03.05

1 Introduction

All the international, and national standards existing today leave it open to the designer which method he would like to use for the assessment of roundness in the technical drawings. The four possibilities given by ISO 1101:2004[1] or the Chinese standard GB/T 7234-2004[2] are the Least Squares Circle (LSC), the Minimum Circumscribed Circle (MCC), the Maximum Inscribed Circle (MIC), and the Minimum Zone Circle(MZC).

Although a theorem of Chebyshev makes sure that the MZC always yields the smallest value for the roundness deviation, all the other methods are still in use. This is either due to the fact, that the calculation of the roundness deviation is not strictly based on the MZC by a rule or standard (although strongly recommended by ISO and ANSI), or that no suitable algorithms for the calculation of the MZC are available for the user. Sometimes functional needs, like the mating of the parts, might also force designers to use one of the other methods.

According to the Guide to the Expression of Uncertainty in Measurement (GUM)[3], all the measurement results must have a stated uncertainty associated with them. But in most cases of roundness measurement, either no uncertainty value is given, or the calculation is not based on the model of the respective association criterion for the geometrical feature, because no suitable measurement uncertainty calculation procedure exists. This is especially true of the case of the MZC.

For the case of roundness measurement in coordinate metrology, this paper suggests the algorithms for the calculation of the measurement uncertainty of the roundness deviation based on the two mainly used association criteria, LSC and MZC. In this connection, the calculation of the sensitivity coefficients for the uncertainty calculation shall be done by automatic differentiation, in order to avoid introducing additional errors by the traditional difference quotient approximations. The proposed methods are exact and need as input data only as the measured coordinates of the data points and their associated uncertainties.

2 Definition and calculation of roundness

The following definition of roundness is given in the normative Annex B of the international standard ISO 1101:2004[1]:

The roundness of a single toleranced feature is deemed to be correct when the feature is confined between two concentric circles so that the difference in radii is equal to or less than the value of the specified tolerance.

※ **Received**: 2010-06-22
Project supported: This work is supported by the National Natural Science Foundation of China (No. 50705002,50627501).
Corresponding author: Zhao-yao SHI(shizhaoyao@bjut.edu.cn)

The locations of the centres of these circles and the values of their radii shall be chosen so that the difference in radii between the two concentric circles is the least possible value.

Fig. 1 demonstrates, what the ISO standard requires. The annulus A_2 on the right side with the width Δr_2 and the centre C_2 is the smallest one including all the measured points. Thus its width is identical to the roundness, which is denoted in the main body of the standard by RON_t .

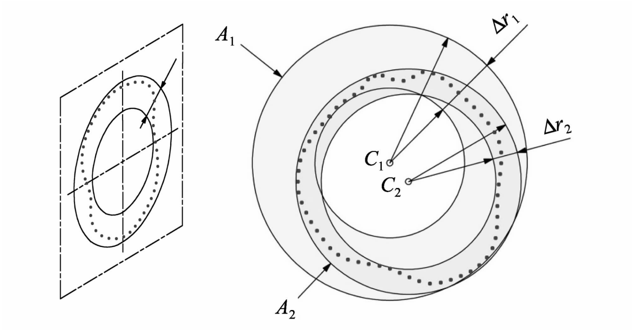

Fig. 1 Definition of roundness according to ISO 1101:2004[1]

The ISO definition clearly supports the minimum zone circle (MZC) association criterion. However, in practice the least squares circle (LSC) association criterion is also used quite often today and thus can not be ignored completely.

For the LSC criterion the roundness is defined as the difference of the maximum and the minimum distances of the measured points from the centre of the least squares circle, i. e. if RON_t denotes the roundness, then

$$RON_t = \sqrt{(x_1 - x_0)^2 - (y_1 - y_0)^2} - \sqrt{(x_2 - x_0)^2 - (y_2 - y_0)^2}. \qquad (1)$$

where (x_1, y_1) and (x_2, y_2) are the coordinates of the two measured points having the largest and the smallest distances from the centre of the LSC with the coordinates (x_0, y_0) (if it should happen, more than one point of one or other kind. We may select just one of them arbitrarily). However, the centre coordinates (x_0, y_0) are dependent of the coordinates of all the measured points.

At first sight it seems not very promising for the uncertainty calculation of the roundness for the LSC criterion, but applying the method proposed in Ref. [4], the uncertainty matrix of the centre coordinates can be easily obtained, which can subsequently be used to deliver the uncertainties and covariances of these two parameters to the uncertainty of the roundness itself, as described in detail in the GUM[3].

In order to detrmine the necessary sensitivity coefficients, the calculation of the partial derivatives of equation (1) with respect to the parameters (x_0, y_0) , as well as the coordinates (x_1, y_1) and (x_2, y_2) respectively, is required. This task can most suitably be performed by a computer program by using automatic differentiation not only for the LSC association[4] , but also for the uncertainty calculation of the roundness. Thus the uncertainty calculation for the roundness in the case of the LSC association criterion can be considered to be solved.

Now turn to the MZC association criterion. For lack of space, how the centre and the two radii for the MZC can be obtained, because the problem is much more complex than the LSC case. However, the optimization theory helps derive conditions, which must be obeyed by any valid solutions.

The MZC association is one of a group of optimization problems, which are summarized under the term Chebyshev approximation. These optimization problems are non-linear and as such do not ensure a unique solution, and even if they do, this solution is not necessarily a global optimum, but can be a local one. However, if the deviation of the measured points is not too big, i. e. in this case if the roundness is small, as it is usually the case in practice, we can be more optimistic a unique solution may be expected, which does not deviate very much from the global optimum, if it deviates at all.

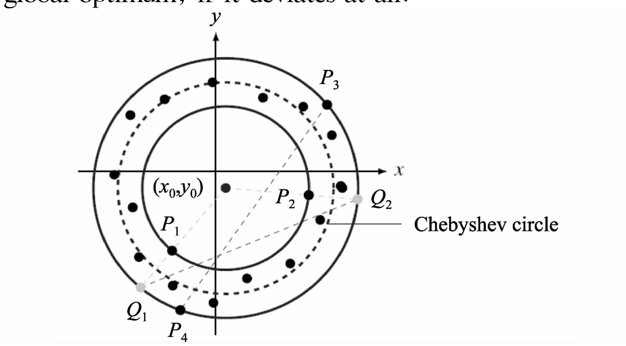

Fig. 2 Typical solution of a MZC approximation

If degeneration can be ignored, which under practical circumstances is mostly the case, the problem has a unique solution, which is controlled by four so-called critical points. There are exactly two critical points on the outer circle and two critical points on the inner circle of the minimum zone annulus, and the cords connecting the projections of the respective points of each of the pairs onto the outer circle do intersect[5] . Fig. 2 shows a typical solution of a MZC approximation.

As can be shown by the theory of computational geometry, the centre of the two concentric circles is inside of the convex hull of the measured data points at an intersection of a nearest Voronoi edge and a farthest Voronoi edge. This fact is convenient for the application of a fast and reliable exhaustive search algorithm, after the two Voronoi diagrams have been constructed. This does not only guarantee the termination of the algorithm, which, otherwise, is not always easy to prove, but also helps to check for the uniqueness of the solution. A Voronoi edge is defined to be a line segment separating two adjacent regions of a Voronoi diagram. Each point on a Voronoi edge is equidistant from two sites associated with these regions. The nearest point Voronoi region associated with a certain point is the set of all the points in the plane that are closer to that particular point than to any other point, while the farthest Voronoi region associated with a certain point is the set of points in the plane that are farther from that particular point than from any other point. For more details about Voronoi diagrams, refer to any suitable text-books on computational geometry, for example[6].

After the centre coordinates and the critical points have been obtained, equation (1) is again applied to calculate the roundness, where (x_0, y_0) now denotes the centre coordinates of the MZC and (x_1, y_1) and (x_2, y_2), respectively, the coordinates of one of the critical points on the inner and the outer circle at a time. In order to calculate the associated uncertainty, we need the uncertainty matrix of the centre coordinates, but there is no formula which can be used for this purpose, because the exhaustive search is not an algebraic algorithm. To solve this problem, the centre coordinates are recalculated as the coordinates of the intersection point of the two perpendicular bisectors of the line segments, which connect the two points of each pair of the critical points on the inner and the outer circle of the minimum zone annulus, respectively. This obtains a function depending on all the four critical points and thus allows an uncertainty calculation for the centre coordinates by applying the usual rules as given in the GUM. The task to calculate the necessary partial derivatives should be again most suitably performed by a computer program using automatic differentiation.

3 Automatic differentiation

The preceding section has shown that the partial derivatives are essential for the calculation of the measurement uncertainty. Two methods are frequently used today to figure out the partial derivatives of given functions numerically, firstly to derive the necessary formulas analytically by hand or by using a suitable computerized algebriic system, and subsequently to code the expressions in a computer program, and secondly to apply a finite difference approximation. However, there is another technique available namely, automatic differentiation (sometimes also called algorithmic differentiation). Unfortunately, this approach is not widely known within the engineering community, although it is possible to figure out the partial derivatives of the arbitrary order of functions efficiently and accurately. Thus this method is well suited to calculate the necessary partial derivatives.

The idea behind the automatic differentiation is, that differentiation, in principle, as is well known from calculus, is a rule based procedure, which thus can easily be programmed to be done by a computer. The computer program parses a given expression and uses term-rewriting methods to apply successively the rules of differentiation to each sub-term resulting from the parsing process. The details of the underlying ideas to construct suitable algorithms can, for example, be found in Ref.[7] or [8].

The code, representing the formula to be differentiated, is automatically generated from the input expression by a suitable parsing algorithm and usually stored in the computer memory as an Abstract Syntax Tree (AST). The derivative of the code is subsequently obtained by simply applying the rules given in table 1 step by step, using a suitable pattern matching algorithm, while traversing the AST depth first from, left to right. The resulting code is generally much longer than the original one and contains superfluous expressions. Thus, a subsequent coding optimization process is needed, in order to simplify the code. The optimization algorithm uses the well known algebraic rules and is based on the techniques, which have been developed for the optimizing compilers.

Tab.1 Differentiation rules

rule	expression	derivative	comment
1	$a = c$	$\mathrm{d}a = 0$	c is a constant
2	$a = x$	$\mathrm{d}a = 1$	x is the dependent variable
3	$a = u + v$	$\mathrm{d}a = \mathrm{d}u + \mathrm{d}v$	u and v are expressions, depending on tie variable x
4	$a = u - v$	$\mathrm{d}a = \mathrm{d}u - \mathrm{d}v$	
5	$a = u \cdot v$	$\mathrm{d}a = u\mathrm{d}v + v\mathrm{d}u$	
6	$a = u/v$	$\mathrm{d}a = (\mathrm{d}u - a\mathrm{d}v)/v$	
7	$a = f(u)$	$\mathrm{d}a = \mathrm{d}u \cdot f'(u)$	f is an arbitrary function, f' is the derivative of f

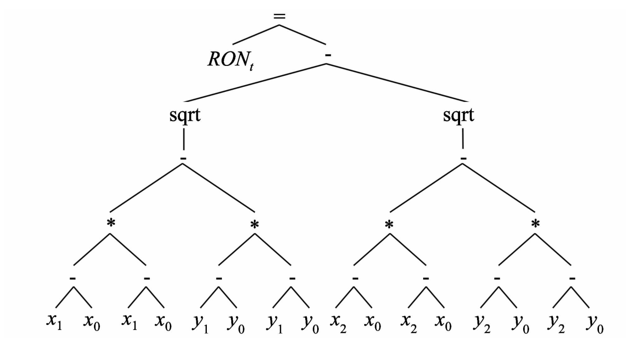

Fig.3 Abstract syntax tree (AST) for the formula

If such an algorithm is applied to the roundness formula as given in equation (1), the results summarized in Fig.3 and Tab.2 are obtained.

The code list resulting from a suitable parsing process is given in the left column. This code block calculates the distance $RON_t = \sqrt{(x_1 - x_0)^2 - (y_1 - y_0)^2} - \sqrt{(x_2 - x_0)^2 - (y_2 - y_0)^2}$, as can easily be verified by successively reinserting the temporary expressions. Internally the code list is stored as an abstract syntax tree (AST), as is depicted in Fig.3, which can automatically be generated by a recursive descending parser from the input expression representing this formula.

The middle column of Tab.2 shows the derivative of the code list, as obtained by a simple application of the rules given in table 1 to the first column of Tab.2 line by line. The computer program is doing just this by pattern matching, while traversing the AST in preorder, i. e.

depth first, from left to right.

Tab. 2 Example for the partial differentiation of the code list with respect to the parameter x_0 for the roundness formula $RON_t = \sqrt{(x_1 - x_0)^2 - (y_1 - y_0)^2} - \sqrt{(x_2 - x_0)^2 - (y_2 - y_0)^2}$

code list	derivative of the code list	after optimization
$t_1 = x_1 - x_0$	$\mathrm{d}t_1 = 0 - 1$	
$t_2 = t_1 * t_1$	$h_1 = t_1 * \mathrm{d}t_1$ $h_2 = t_1 * \mathrm{d}t_1$ $\mathrm{d}t_2 = h_1 + h_2$	
$t_3 = y_1 - y_0$	$\mathrm{d}t_3 = 0 - 0$	
$t_4 = t_3 * t_3$	$h_3 = t_3 * \mathrm{d}t_3$ $h_4 = t_3 * \mathrm{d}t_3$ $\mathrm{d}t_4 = h_3 + h_4$	
$t_5 = t_2 - t_4$	$\mathrm{d}t_5 = \mathrm{d}t_2 - \mathrm{d}t_4$	
$t_6 = \mathrm{sqrt}(t_5)$	$h_5 = \mathrm{d}t_5 / t_6$ $\mathrm{d}t_6 = h_5 / 2$	$\mathrm{d}t_6 = t_1 / t_6$
$t_7 = x_2 - x_0$	$\mathrm{d}t_7 = 0 - 1$	
$t_8 = t_7 * t_7$	$h_6 = t_7 * \mathrm{d}t_7$ $h_7 = t_7 * \mathrm{d}t_7$ $\mathrm{d}t_8 = h_6 + h_7$	
$t_9 = y_2 - y_0$	$\mathrm{d}t_9 = 0 - 0$	
$t_{10} = t_9 * t_9$	$h_8 = t_9 * \mathrm{d}t_9$ $h_9 = t_9 * \mathrm{d}t_9$ $\mathrm{d}t_{10} = h_8 + h_9$	
$t_{11} = t_8 - t_{10}$	$\mathrm{d}t_{11} = \mathrm{d}t_8 - \mathrm{d}t_{10}$	
$t_{12} = \mathrm{sart}(t_{11})$	$h_{10} = \mathrm{d}t_{11} / t_{12}$ $\mathrm{d}t_{12} = h_{10} / 2$	$\mathrm{d}t_{12} = t_7 / t_{12}$
$RON_t = t_6 - t_{12}$	$\mathrm{d}RON_t = \mathrm{d}t_6 - \mathrm{d}t_{12}$	$\mathrm{d}RON_t = \mathrm{d}t_6 - \mathrm{d}t_{12}$

As is seen, the resulting code list is much longer and contains superfluous code like $\mathrm{d}t_1 = 0 - 1$ or $\mathrm{d}t_9 = 0 - 0$. This is generally the case after the differentiation rules have been applied. Depending on the input expression, the derivative of the code list is usually longer than the original code list by a factor of three to ten. Thus a subsequent optimization process is needed to simplify the code resulting from the differentiation algorithm. The optimization algorithm is based on the techniques, which have been developed for the optimizing compilers and uses the well known algebraic rules, as e. g. commutativity and associativity, for algebraic simplification, as well as techniques like constant propagation, constant folding, common code elimination, and dead code elimination. The result of the application of the optimization process for the roundness formula is shown in Column three of Tab. 2. In comparison with Column 2, this code is much more simplified, since it uses sub-expressions of the first column. This strategy is strongly recommended, because the original code needs to be calculated anyway, in order to obtain the value of the objective function.

By applying of the outlined methods, a computer program can be written, which automatically generates another program, which in turn is able to solve a particular uncertainty calculation problem. The input for the program generator is just the roundness function of the problem under consideration and the choice of a particular solving method to be used. The generated program is subsequently compiled as usual and linked with a library providing the necessary supporting functions, which implement the required optimization strategies.

4 Conclusion

It has been shown how automatic differentiation can be applied to the uncertainty calculation of roundness for the two practically important cases of the Least Squares Circle (LSC) and the Minimum Zone Circle (MZC) association criterion. The proposed methods avoid additional errors, which otherwise are caused by the traditionally used difference quotient approximation, are exact and need input data only as the measured coordinates of the data points and their associated uncertainties.

References

[1] ISO 1101, 2004. Geometrical Product Specifications (GPS). Geometrical tolerancing. Tolerances of form, orientation, location and run-out, ISO, Geneva.

[2] GB/T 7234-2004, 2004. Geometrical Product Specifications (GPS). Measurement of roundness. Terms, definitions and parameters of roundness, SAC, Beijing.

[3] ISO 1993, 1993. Guide to the Expression of Uncertainty in Measurement, ISO, Geneva.

[4] Michael Krystek, Zhao-yao Shi, Jia-chun Lin, 2010. Least squares association of geometrical features by automatic differentiation. Key Engineering Materials, 437, p. 222-226.

[5] T. J. Rivlin, 1979. Approximation by circles. Computing, 21, p. 93-104.

[6] M. de Berg, M. van Kreveld, M. Overmars, O. Schwarzkopf, 1998. Computational Geometry. Springer, Berlin.

[7] L. B. Rall, 1981. Automatic Differentiation. Techniques and Applications. Springer, Berlin.

[8] A. Griewank, 2000. Evaluating Derivatives. Principles and Techniques of Algorithmic Differentiation. SIAM, Philadelphia.