THE IMPROVEMENT OF THE SIMULATION METHOD IN VALIDATION OF CMM'S SOFTWARE

Halina NIECIĄG

AGH – University of Science and Technology, 30 Mickiewicza Av., 30-059 Krakow, hnieciag@agh.edu.pl

Abstract:
The paper presents the part of ongoing validation studies involving the accuracy of software algorithms calculating Gaussian associated features. In studies the numerical simulation method Monte Carlo (MC) was applied due to the number and nature of the variables forming the result of coordinate measurements, as well as due to complex and multidimensional measurands. In the paper an attempt to investigate the possibility of improving the results obtained by MC method is described. The LHS (Latin Hypercube Latin) algorithm was elaborated as an alternative to simple sampling scheme of classical MC algorithm.

Keywords: CMMs, Monte Carlo simulation, validation, software

1. INTRODUCTION

At present, wide range of complex coordinate measuring instruments are dominating in industrial metrology, [1]. A common feature of coordinate measuring systems with a different design, configuration and for different applications is a digital form of measurement information.

The first step in coordinate measurements consists of determining the position of points of the physical object that is measured at a fixed coordinate system. The data are subject to further processing, then the results are interpreted, visualized and reported. The hardware and software layer of the measuring system are involved in first stage of coordinate measurement, while in the following one the software plays a dominant role. The quality of output information depends strongly on the quality of the software.

One form of metrological confirmation [2] ensuring that measuring system meets the requirements of its intended use, is a validation. An important step in validation of the measuring system is software validation, especially an analysis of the quality of the algorithms calculating the values of measurands. Evaluation of information, obtained as a result of a multi-level data processing, relies on the assessment of the uncertainty of the indirect measurement. This paper presents a fragment of studies of the package of programs providing the numerical values of the measured parameters of regular geometric shapes.

2. MEASUREMENT TASK

The basis of the product development process is a geometric model of part having an ideal shape. This ideal shape is formed by ideal objects, created during the design and construction process. After manufacturing the real part features are inspected. The software enables verification of the real objects geometry by algorithms of parametric identification of corresponding them idealized, "*associated*" features, consistently with the concept of design. The basis of the programs of interest are Gaussian algorithms, which are used the most often in standard software. They are reconstructing features like e.g. circles, lines, planes, etc., by minimizing the Euclidean norm of residuals.

The measurement model representing the relation between the input quantities X and output quantities Y is defined for each geometric feature separately. The measurand Y is not measured directly, but is determined from input quantities X according to the mathematical formulation. The mathematical model is expressed in the form of equation (1), which implicitly relates X and Y.

$$F[\,X, W\,] = 0\text{, where} \tag{1}$$

$X = [\,P_1(x_1, y_1, z_1)\ldots P_m(x_m, y_m, z_m)\ldots P_N(x_N, y_N, z_N)]$, in general, represents the 3N input quantities, that is, the three–coordinates of the N–measured points, however Y (a measurand) is an output quantity having frequently the form of vector parameters.

Building the measurement model many factors, which affect the result of the measurement such as a strategy, surface texture, the selected reference surfaces, the position of the measurement space, etc., should take into account.

The uncertainty estimation in coordinate measuring systems requires a separate analysis of each measurement model, which is implemented by the system under certain conditions. This leads to the formulation of the concept of *measurement task*, as an objective of measurement defined by the measurement model, in which the included variables can change in a certain way and within certain limits.

3. UNCERTAINTY ESTIMATION

Several methods of uncertainty estimation are employed in accuracy evaluation. On the whole, there are two basic approaches to this problem: one is relating to the law of propagation of uncertainty, the second one is based on the propagation of distributions. They share a common philosophy to consider a measurement as an experiment of realization of random variables, from which some of them explicitly participate in measurement process, others are derived from outer sources and affect the measuring information. Use of information on variables and their processing should be then performed according to the rules of probability theory.

The character of random variable is overall determined by probability distribution (*PDF*). An assessment the interval expressing the variation of it requires knowledge of this distribution. Propagation of distributions allows to determine the probability distribution for an output quantity from the probability distributions assigned to the input quantities on which the output quantity depends. Simulation Monte Carlo is a method exploited to the propagation of distributions by performing random sampling from probability distributions [3].

Series of standards [4] discusses strategies, which are helpful in assessing an uncertainty evaluation of the measurement task. Software is regarded there as one of the six important sources strongly interacting on task uncertainty. Estimation of the uncertainty contributor from software interaction must be based on separate studies. The division of the whole measurement into elementary measurement tasks corresponds in a natural way to the structural decomposition of software package into programs, implementing these tasks by calculations. The basis for the validation of the programs can be presented in mathematical models (1).

4. INPUT QUANTITIES MODEL

According to the above concept, building a model of *X*, *Y*, *Z* input quantities, brings in effect to the construction of a probabilistic model of their errors. Realization of each random variable can be considered as a sum of the "true" value and the deviation (an error). True value is known from model assumption and it corresponds to the coordinates of a point (P_n), lying on the ideal surface ($x_{n,ref}$, $y_{n,ref}$, $z_{n,ref}$). The deviation Δ_n at the same point is a realization of certain number of error random variables. In designing a probabilistic model of errors, the principle analogous to the deterministic concept of composing a vector error may be applied [5], where total error is treated as a superposition of partial errors generated by specific factors or coming from the distinct sources.

4.1 Impact factors

Unlike the recommendations found in [4], in software validation only these factors accompanying the actual measurement are taken into account, which may affect the software evaluation. Lot of others, such as a surface contamination, a way of part clamping, etc, should be subordinated to the proper rules of the measurement technique. Applying the right measurement practice we can significantly reduce their impact on the measurement results, and the same the measurement uncertainty.

In our examination of Gaussian software two factors were included, which could not be eliminated in the actual measurements even though applying appropriate measurement practice.

The first is related to the impact on result of the actual state of a measured object. Considering the periodic nature of machined surfaces (obtained e.g. by milling or grinding) the deviations of measured surface from the ideal can be modeled by a sum of elementary harmonics functions, [6].

Second factor represents the cumulative effect of influence quantities being an aggregate of all unidentified uncertainty sources.

4.2. Error model

Analyzing the final effect being a superposition of realizations of different random variables, one can note that it can be determined by a position and width of its distribution. The value of the deviation at any point is formed so, for whatever reason, by two components (2): the value of random error ε (ε_x, ε_y, ε_z) and one having the systematic nature Δ_s ($\Delta_{x,s}$, $\Delta_{y,s}$, Δ_{zs}), that is corresponding to the traditional classification of errors.

$$\Delta_n = \Delta_s + \varepsilon \, . \tag{2}$$

In the following studies an universal concept of modeling error by the superposition of these two components has been applied. Depending on this which from these two components are important and significant, the interpretation of them can be different. A systematic component is most affected by device geometry errors. Random component is derived from the interaction of many unregulated factors, but on can assume that dominating is a probing error from measuring head, [5].

5. MEASUREMENT STARTEGY

Validation tests suitability of the software for its intended use, firstly by the uncertainty estimation of the measurement task, secondly, by verifying that it meets an acceptance criterion. Measurement strategy significantly affects the uncertainty of measurement task. It is a powerful means of controlling the coordinate measurement uncertainty, not very aptly attributed to the area of measurement technique. The strategy involves all possible decisions, which operator could undertake while performing measurement (e.g. the number of sampled points and their location on measured surface, approach direction, length of probe path, the kind of the selected probe, certain settings like measurement velocity, etc.). The evaluation of the measurement task should refer to the specified level of all factors, in which this task is executed.

An acceptance criterion checking the success of the test, is in the form of interval, that should not exceed the uncertainty, [7]. Again, the limits of such intervals must be determined and valid under stipulated conditions.

6. NUMERIC SIMULATIONS

The choice of the appropriate method for testing the software is dictated in many ways, since they differ not only from effort and time consuming, but above all else the degree of reliability of the results. Although the correct result will never be known for experimenter, the simplifying assumptions (e.g. linear approximation) can lead to the invalid uncertainty statements. The Monte Carlo (MC) methods are devoid of these disadvantages. They can be used for non–linear models or for solution of any complicated tasks without necessity of the additional assumptions.

The essence of the MC method is based on repetitively done simulating experiment, which rely on drawing the probable values of input quantities from assigned distribution, then on calculating the numerical values of the output quantity, in accordance with measurement model [3]. It enables to obtain an approximate numerical representation of the distribution function of examined quantities.

The simulation model is determined by the model of quantities (2), defined relevantly to the algorithm under test. In one cycle of replications ("*trial*") the set of coordinates, as realizations of random *X, Y, Z,* for *N*–data points is generated by simulations, $\{ x_n , y_n , z_n \}_{n=1}^{N}$. For each random sample the resulting model value of *Y* one obtains, using the particular algorithm. After sorting *Y* values from many replications into strictly increasing order, the discrete representation of G_y distribution function one obtains, [3]. It enables forming the coverage interval, the length of which is a measure of the uncertainty of output *Y*. As the endpoints of this interval the quantiles 0,025 and 0,975 are usually accepted, giving the coverage probability p = 0,954.

In examination of Gaussian software the classical MC algorithm has been developed using variant of the simple sampling, [8]. In this sampling method the values of variables reflect the points randomly distributed over the established ranges of their variations. Assuming a sufficiently large number of replications, the resulting estimate will be sufficiently accurate. But also "technical aspect" may affect the accuracy of the characteristics of random *Y*, i.e. how the input random variables are generated, what is called a sampling scheme. An attempt was made to modify the application of the classical algorithm and examine whether improvement of "uniformity" of sampling may affect the magnitude or reliability of estimate of the coverage interval for analyzed output quantities. The LHS (Latin Hypercube Sampling) method was chosen for this purpose.

7. STRATIFIED SAMPLING ALGORITHM

The LHS is a kind of stratified MC method, which is improving the uniformity of the sampling and can lead to the decreasing of variance, [9]. In this method the range of PDF for each variable is divided into separable intervals of equal probability. From each interval one value of each variable is selected randomly, due to the probability density of the chosen interval. The name of this sampling method is derived from a hyper–cube, which is composed of so formed sets of values of several variables. The algorithm implemented for examination of software was as follows:

1. The PDF for k=3 variables (i. e. *X*, *Y*, *Z*) was divided into a lot of compartments. The same number of independent random numbers were drown from interval [0÷1], and were placed in different layers.
2. In one cycle of simulation, comprising j th number of *M*–replications with regard to the LHS scheme, one random value for each of *N*–data points and for each of k – variables was generated. These values were then used to calculate the values of the coordinates of simulated points.

After obtaining the set of data points the calculations by relevant algorithm were performed, like in simple MC.

8. EXPERIMENTS AND RESULTS

Comparative simulation experiments were carried out using both of sampling schemes (MC and LHS) in order to generate the coordinates of points on the spherical surface. A sphere is a feature determined in three dimensions by nonlinear equations, so the observations from such investigations can have an universal meaning. Moreover, this feature occurring first of all in bearing industry is characterized by high accuracy demands, [10].

In our experiments the same model and data like in the work [8] were assumed. Two numbers of replications of simulated measurement cycles were established: M=10^2 and M=10^4. All simulation programs have been developed in the form of scripts for the MATLAB environment.

8.1 Sampling by a classical MC and LHS algorithm

In MC classical algorithm for each *j*–replication ($j=1,2,\ldots,M$), the three coordinates ($x_{n,j}$, $y_{n,j}$, $z_{n,j}$) of each *n*–point (P_n) from *N*–data points were calculated respectively as a sum of reference coordinate values ($x_{n,ref}$, $y_{n,ref}$, $z_{n,ref}$), the systematic errors Δ_s=($\Delta_{x,s}$, $\Delta_{y,s}$, Δ_{zs}), and random errors $\varepsilon_{,nj}$ =($\varepsilon_{x,n,j}$, $\varepsilon_{y,n,j}$, $\varepsilon_{z,n,j}$). For each (P_n) a simulated numerical value of each coordinate was calculated using such a sum e.g. for (x_n) having the form (3).

$$x_n = x_{n,ref} + \Delta_{x,s} + \varepsilon_{x,n,j} \;. \qquad (3)$$

Analogously, the numerical values of (y_n) i (z_n) were calculated. The values of errors ($\Delta_{x,s}$, $\Delta_{y,s}$, Δ_{zs}), for each variant of *j*-simulation cycle were established arbitrarily. The random errors were assigned to the Gaussian distribution $\boldsymbol{N}(0, \sigma_{\varepsilon,x})$, $\boldsymbol{N}(0, \sigma_{\varepsilon,y})$, $\boldsymbol{N}(0, \sigma_{\varepsilon,z})$.

The output quantities of the sphere are the radius of the sphere (r_o), and the coordinates of its center (x_o, y_o, z_o).

Due to simulating model, the assumed model values of (r_o, x_o, y_o, z_o) as the reference values may be accepted.

In the LHS sampling the same model and scenario experiment was used. As above, each coordinate of point was a sum of a reference coordinate, systematic error value, and one from drawn errors values. Random errors values were generated after making a division of the range of Gaussian PDF into many layers having equal probability.

8.2. Results

The exemplary results of simulations performed with M=10^2 are visualized on Fig. 1 and Fig. 2. The coverage intervals, GUM confidence intervals and the biases are reported. For M=10^4 replications the results are presented on Fig. 3 and Fig. 4.

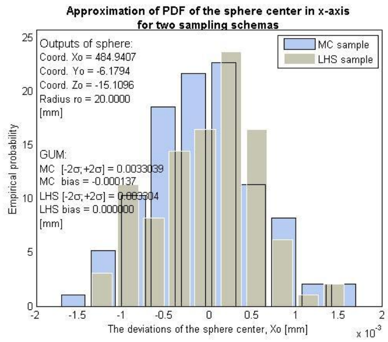

Figure 1. The empirical PDF for x_o, M=10^2

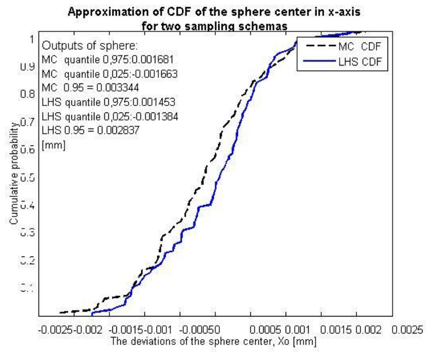

Figure 2. The empirical CDF for x_o, M=10^2

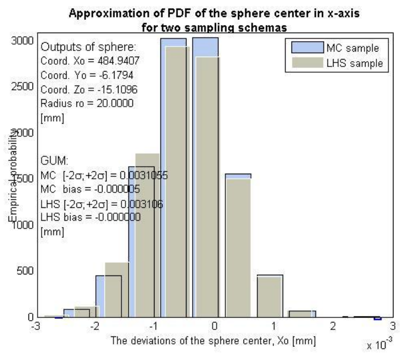

Figure 3. The empirical PDF for x_o, M=10^4

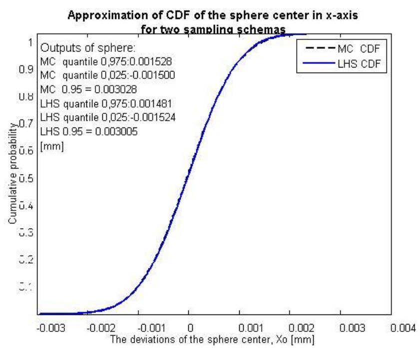

Figure 4. The empirical CDF for x_o, M=10^4

All experiments show that LHS provides slightly less variations in drown values in relation to classical MC. Respectively, LHS coverage intervals of output quantity (coordinate x_o) were slightly shorter than ones obtained by simple sampling, in some cases were almost identical, depending on M–number of trials. When M–number was large there a high consistency in probability densities was observed, otherwise the discrepancy could be noticed.

9. SUMMARY

The decisive advantage of LHS seems to be increased reliability of estimates of the Y from this, which is obtained by simple sampling MC. The LHS biases determined with relation to the model values were significant smaller. The histograms and CDF plots show, that a full stratification of the sampled distribution provides a better representation of “shape”, without clustering effects, especially when a number of replications is not to large.

Summarizing, in the paper the LHS method is suggested as the alternative to the classical MC simulation, believing that for more realistic model including form deviations, it may more accurately reflect the distribution of Y values and be therefore more suitable for testing software in validation.

REFERENCES

[1] E. Ratajczyk, „Współrzędnościowa technika pomiarowa”, Oficyna Wydawnicza Politechniki Warszawskiej, 2005.

[2] ISO-10012: 2003, “Measurement management systems” — Requirements for measurement processes and measuring equipment.

[3] JCGM 101:2008, Evaluation of measurement data Supplement–1 to the “Guide to the expression of uncertainty in measurement”– Propagation of distributions using a Monte Carlo method.

[4] ISO/TS 15530-4: 2008. Geometrical Product Specifications (GPS) – Coordinate measuring machines (CMM): Technique for determining the uncertainty of measurement – Part 4: Evaluating task-specific measurement uncertainty using simulation.

[5] J. Sładek, „Dokładność pomiarów współrzędnościowych”, Wydawnictwo Politechniki Krakowskiej. Kraków, 2011.

[6] S. Adamczk, D. Janecki, K. Stępień, „Badanie metod pomiaru i oceny błędów kształtu kulistych części maszyn”, Mechanik, Nr 12/2011, ss. 958–961.

[7] H. Nieciąg, “The assessment of the criterion in tests of acceptance type of the metrological software in coordinate measuring systems”, *Proc. Xth International Scientific Conference* „Coordinate Measuring Technique” ATH, TC14 IMEKO, Bielsko–Biała 2012.

[8] R.A. Tabisz, H. Nieciąg, „Symulacja wpływu czynników systematycznych i losowych na wyniki pomiarów współrzędnościowych maszyn pomiarowych”, *PAK*, Vol. 58 nr 12/2012, ss. 1065-1067.

[9] M. D. McKey, W. J. Conover, “Comparison of three methods for selecting values of input variables in the analysis of output from a computer code”, *Technometrics*, Vol. 21, No 2, May 1979, ss. 239-245.

[10] S. Adamczk, D. Janecki, K. Stępień, “Specific problems of sphericity measurements by the radial method”, *Implementation of Coordinate Metrology*, WNATH Bielsko Biała, Nr 12/2011, ss. 79–88.