XIX IMEKO World Congress
Fundamental and Applied Metrology
September 6−11, 2009, Lisbon, Portugal

SOME COMMENTS ON REFERENCE DATA SET GENERATION IN PASSING

Halina Nieciąg [1], Zbigniew Chuchro [2]

[1] The Institute of Advanced Manufacturing Technology, Cracow, Poland, hnieciag@ios.krakow.pl
[2] The Institute of Advanced Manufacturing Technology, Cracow, Poland, zbigniew.chuchro@ios.krakow.pl

Abstract − The paper describes some problems connected to the reference data set generation for the validation of metrological software. The validation concerns the Gaussian procedures for solving the task of the approximation of geometric features.

Keywords: validation, data, software

1. INTRODUCTION

The validation of metrological software raises a number of questions but cardinals are three of them. First one is who shall perform the validation after software has been developed. The next problem relevant to this is what particular requirements shall be formulated if no knowledge about the target software use. And the latter is what the methods should be suitable in software examination.

Common approach to first question is commission to validation from such bodies as PTB (Germany) or NPL (UK) national laboratories developing special testing services. But alternative premise to the first question seems to be a recommendation of ISO standard [1] to undertake own-self software validation by accreditation bodies. Additionally, ISO standard [2] addressed to coordinate metrology provides the strategy and the extent, i.e. number of test cases, to which the key components of CMM's software, the Gaussian procedures, shall be tested. One drawback of this approach is that undertaking validation one needs to solve two latter questions.

2. TO ENFORCE ISO STANDARD

In [2] two methods of testing are recommended: testing applying the reference software or testing by producing the reference data sets. In the first method reference software and software under test are applied to the same data set. The returned results are then compared to each other by computing an absolute value of the difference. The basic of second approach is however designing the reference data sets as software inputs, i. e. direct measured quantities values, and reference values respectively. The outcomes of software subject testing, indirect measured output quantities, after applying the reference data sets are compared with corresponding reference values, like in the method before, which allows establishing if there are no essential discrepancies between them. The methods recommended by standard are easy when we have at our disposal reference software or reference results. In the case of lack of it we had to develop other concepts. Using so-called reference pairs [3] in such a way that the other principles of validation are fulfilled leads to the creation of appropriate data error model being in fact the statistical model of variables for data simulations. Model error building is crucial issue with regard to valid inference about quality of results of evaluated software.

3. ERROR MODEL DUE TO ISO

Although the points produced according to ISO algorithm are located seemingly at uniform pattern nodes of nominal feature segment they are distributed approximately regularly. The randomisation used in order to extract the locations of points conforms to the rule of probing strategy "ad hoc". In accordance to standard [2] measured object influences and experiment influences are realized in data generator through disturbing the nominal data on object having ideal geometry by form deviations and Gauss noise.

The ground assumption of deviations generating is that they should be projected on nominal area perpendicularly that guarantees that the reference values can be considered as known a priori. The assumption is derived from known properties of Gauss-Newton local solution.

In the context of validation however testing the software for checking its fitness for purpose requires forming the data for particular application domain. First of all they should be representative and have properties imitating the typical actual measurement data gathered in typical environment conditions from typical machined surfaces, i. e. they should be determined for "intended use" of software.

4. DESIGNING THE DATA

Respecting above demands we propose designing the data in more natural, corresponding with reality, manner.

Following Forbes: "*Understanding the inexactness or uncertainty associated with the measurement result arising from uncertainty associated with the input data is a key activity in metrology...*" – very important seems identifying the factors which data are subject to.

The universal measurement task realized for geometric features by Gaussian software consists of two stages:

assessing the estimates of measured quantities, following collecting a set of input data, Fig.1.

Applying this consideration one reduces a number of potential sources of inaccuracy into two main factors affecting the results: associated with each program quality, i.e. mainly due to data processing techniques, or relating to coming data.

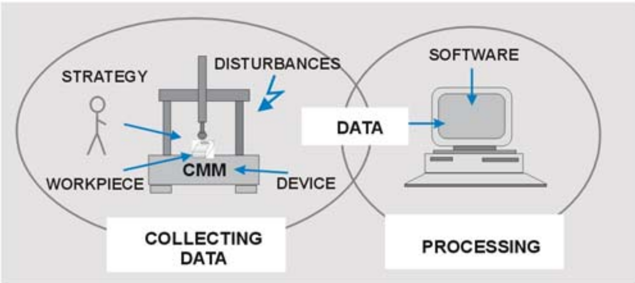

Figure 1. Sources of uncertainty.

Quantifying of the program impact on results leads unavoidable to an analysis of its numerical uncertainty. Numerical uncertainty of solution is subject to a lot of approximations and limitations in software designing and implementation (e.g. effects of floating point operations, translations, normalizing). The null space technique based on kernel theory [4] has been employed in generating data for numerical uncertainty assessment. An error model used in validation was formed basing on an assumption that data are subject both from measured object and experiment. The procedure for generating takes a following form. The data are randomly chosen from uniformly spaced segments of nominal feature area. The points are disturbed then by form deviations arising from machined surface structure. A real surface is represented by the sum of independent harmonics, with given number, given magnitudes, and random phase angle (1). The current deviation ε_{iR} of any $i-\text{th}$ point is represented by formula:

$$\varepsilon_{iR}(\varphi_i) = \sum_k a_k \sin(k\varphi_i + \psi_k)\,, \tag{1}$$

where k=2,3,…,p is the number of harmonics describing the surface, a_k is the amplitude of $k-\text{th}$ sine wave, the ψ is the phase angle. The quantities a_k and ψ present in formula (1) are assumed to be random variables.

Many potential sources of errors can be observed in each measuring device's axis. Each coordinate of disturbed point is subject to random errors generated from normal, i. e. Gauss distribution: $\delta_x \sim N(0, \sigma_x^2)$ and $\delta_y \sim N(0, \sigma_y^2)$, giving the resultant deviation at point ε_{iN}:

$$\varepsilon_{iN} = \sqrt{\delta_{ix}^2 + \delta_{iy}^2}\,. \tag{2}$$

Finally, the data are apart from nominal positions by total residuals being superposition of both disturbances (1) and (2).

One should be distinguished that both δ_x and δ_y, for simplicity for two-dimensional feature, are independent errors arising from experiment influences (device, environmental factors, strategy), not from dimensional variability. Our strategy differs insignificantly from ISO concept, but the consequences can be meaningful in some cases. The final residuals in many cases do not follow to normal distribution, especially if only few harmonics are representing the surface. The solution returned by software does not fulfill a condition arising from properties of least squares estimate. It can be mathematically correct although not necessarily can conform to the known value specified a priori. The technique of creation of reference pairs is not still valid [3]. We cannot infer about statistical properties of solution without thorough analysis. Another approach was needed. The acceptable bounds on differences between actual solution and specified value had to be estimated developing the modeling technique. This method has been previously discussed in [4]. Although the validated software supplied by data does not distinguish the effects influencing them, the error in evaluated estimates can be decomposed into separate parts and be analysed accordingly if only the certain bounds on errors in data, i.e. confidence intervals of supplied data can be considered as known.

To establish the uncertainty of evaluated quantities basing on the knowledge of data errors one should know if they are valid, that is if we can predict the data uncertainty. The data generated according to any statistical model as in such cases should be validated. This imposes the generated data sets to be carefully investigated.

5. ERROR MODEL IN DATA GENERATION

We consider a model for the deviation $\varepsilon = (\varepsilon_1, ..., \varepsilon_n)^T$, where:

$$\varepsilon_i = \varepsilon_{iR} + \varepsilon_{iN}\,, \tag{3}$$

and n is number of sampled points.

The complete deviation at i-th point is then a sum of elements, which are realizations of two independent random variables, denoted respectively E_R and E_N.

Because the phase angle ψ in (1) is chosen randomly the realisation of E_R at any point on profile (1) follows to Rayleigh or normal distribution depending on the number of harmonics [5]. When this number is small the distribution can differ significantly from the normal distribution.

The second contributor E_N in the form of the superimposed errors being realisation of two independent normally distributed variables (2) can follow to Rayleigh or Maxwell distribution if only they are uncorrelated and have approximately equal variances, i. e. $\sigma_y^2 \cong \sigma_x^2 = \sigma^2$.

Accordingly to the testing rule that known inputs lead to the specified outputs the confidence intervals of input quantities should be known for further analysis and evaluation of the software results, Fig. 2.

Applying simple additive model of errors (3) the bounds encompassing the model location for generated point of the given confidence level have to be estimated.

Several methods we had at our disposal to realize this task. They can be roughly divided into analytical methods (theoretical functional characteristics identifying the distribution), numerical (approximation by means of Monte Carlo method) or approximate ones basing on the random variable moments assessment. We analysed a number of them trying to chose among them the most useful and convenient for our purpose. The most common used appears to be the Monte Carlo simulation implementing the propagation of the distribution [6].

Figure 2. Dependences in validation.

Indeed, if E_N and E_R are independent random variables with known density $f_N(\zeta)$ and $f_R(\eta)$ respectively, defined for all ζ, then their sum E_Σ is a random variable with a density function $f_E(\upsilon)$ being the convolution of $f_N(\zeta)$ and $f_R(\eta)$. But in some special circumstances, when the actual distribution of variables can deviate from model assumptions, more appropriate seemed us to develop different concept.

6. THE CONCEPT

The values of parameters and factors needed in designing the reference data sets are assigned strictly with respect to ISO standard recommendations: the number of harmonics as well as magnitude of random effects, the number of sampled points, and others. The set of constants and randomised quantities constitutes the particular test case and determine unique reference data set, thus each modelled case is represented by separate data set.

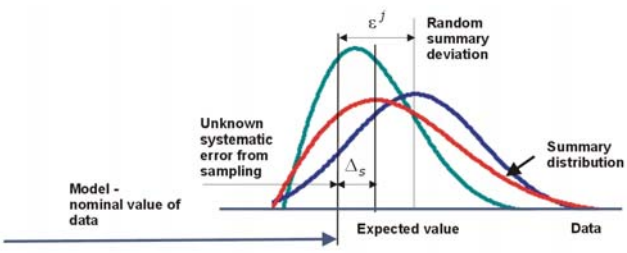

Figure 3. Error model.

Each of realization of data is a sum of its "true" value, known from model assumption and equal to the point nominal location and deviation being realization of certain random variable, Fig. 3. From statement on statistic mean results that analyzing the confidence intervals of random coordinates leads to analyzing the confidence interval of random errors.

Their distribution is considered as unique, due to single test case distribution of random variable. It can be analysed at fixed but arbitrary data point. Thus after evaluation of summary deviations for each constructed data point they are all transformed to one model point.

Finally, the random sample ε, like in (3), is derived considered to be adequate to represent a modelled test case (e.g. depending on number of harm.), certainly under usual conditions.

But this fact appears the main assumption underlying the bootstrap methods, which can be useful for finding the distribution that results from convoluting the component distributions [7]. One advantage of this concept is that an exact analytical expression for resulting distribution is not indispensable to be a priori specified. From sample the necessary information is possible to extract using bootstrap intervals estimation.

7. IMPLEMENTATION

We have implemented the following algorithm [8].

1. Sampling. Let original sample of data errors was drawn at random from some unknown distribution. The assumption has been made that the sample ε constituted the underlying distribution ("iid" sample).
2. Resampling. We generated random samples $\varepsilon^* = (\varepsilon_1^*, ..., \varepsilon_m^*)^T$ of size m, being the random vector of ε a large number of times. The obtained values were used to form an empirical distribution. The distribution of E_Σ was approximated by means of empirical distribution basing on sample ε^*.
3. Construction of the 100α percentiles. The α-th percentil of N ordered values, where $\alpha \in (0,1)$, was determined after sorting all values, first corresponding to the nearest integer number of $N(\frac{\alpha}{2})$, and respectively the second as $N - N(\frac{\alpha}{2}) + 1$. Next the $(1-\alpha)100\%$ confidence interval was easily found.
4. Execution. The results were based on N=1000 replications. The algorithm has been implemented in Matlab ver. 7.7.0.471, in order to test this concept. The pseudo-random number generator built in Matlab has been used.
   We made many trials plotting the final distribution, as well as calculating the bootstrap mean and confidence intervals.

8. AN EXAMPLE

A practical example illustrates this application. Let's consider a circle feature. For randomly distributed points the random signals in each coordinate were drown from identical distribution N(0, σ=0.002), Table A.6 [2], giving the resultant residual. The sinusoidal components have magnitudes drown from interval ± 0.005 mm, Table A.5 [2]. Table 1 includes further assumptions of test parameters for designing three samples of data.

Table 1. Set of test parameters values.

Code of sample	Radius, center x_o, y_o [mm]	Phase angle φ	Number of points	Number of harmonics
FM_001	Random from [50÷450] [35÷315]	Random from $[\pi \div 2\pi]$	5	2
FM_002	Ditto	Ditto	5	5
FM_003	Ditto	Ditto	10	5

In Table 2 we present the values of mean, upper and lower bounds of 99.73% confidence interval. They were obtained by means of bootstrap method and alternatively by conventional analysis under the normal distributed data assumption.

Table 2. Results of the experiment.

Estimate of statistics		FM_001	FM_002	FM_003
$\mu(X)$	Original sample	-0.0012	0.0022	-0.0013
	Bootstrap sample	-0.0010	0.0003	-0.0008
$U_{1-\alpha}(X)$	Original sample	-0.0060÷0.0060	-0.0043÷0.0043	-0.0036÷0.0036
	Bootstrap sample	-0.0049÷0.0025	-0.0006÷0.0051	-0.0035÷0.0012
$\mu(Y)$	Original sample	0.0042	0.0008	0.0030
	Bootstrap sample	0.0065	0.0010	0.0023
$U_{1-\alpha}(Y)$	Original sample	-0.0040÷0.0040	-0.0041÷0.0041	-0.0033÷0.0033
	Bootstrap sample	-0.0010÷0.0072	-0.0022÷0.0033	-0.0006÷0.0050

From results one can evaluate the differences between these two methods. The interval length computed basing on the quantils of standard normal distribution can be sometimes almost 67% greater than that one basing on bootstrap method. The bootstrap distributions for two first analysed samples are showed as well on Fig.4.

One can notice that using bootstrap method can be valuable in the case of small number of harmonics, where data can differ significantly from Gaussian data. Deciding on it we should however answer on some questions. The reliability of bootstrap results depends not only on number of replications but also on how the original sample is representative. To invoke the asymptotic properties of data we should determine among others the representative sizes of data for each test case.

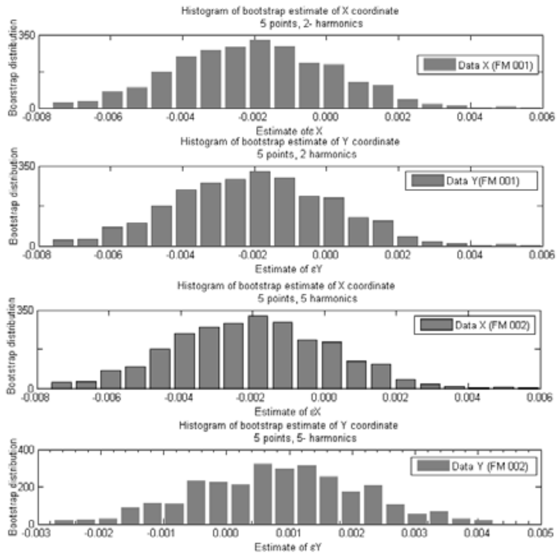

Figure 4. Histograms for bootstrap samples.

9. CONCLUSION

To ensure compliance with standard regulation submitting software to validation is important necessity for software developers. The activities have been planned and occurred in various stages for validation of metrological software in our Institute. The paper describes some problems involved the validation by generating the reference data sets. To solve any of them the bootstrap method can be applicable especially in cases when little is known about statistical distribution of the generated data and the size of data set is small. The method was shown to be very simple in implementation and can stand a useful tool in validation.

REFERENCES

[1] PN-EN ISO/IEC 17025-1: 2005, "General requirements for the competence of testing and calibrations laboratories", PKN Warszawa.

[2] PN-EN ISO 10360-6: 2003, "Estimation of errors in computing Gaussian associated features", PKN Warszawa

[3] http://www.npl.co.uk/.

[4] H. Nieciag, Z. Chuchro, „Validation of metrological software", *12-th IMEKO Symposium*, pp 421-426, Annecy, France, Sept. 2008.

[5] A. N. Gawriłowa, "Accuracy of manufacturing. The probabilistic and statistical methods of the manufacturing accuracy analysis", WNT Warszawa 1978.

[6] ISO/TC 213 N 659: 2004, "Guide to the expression of uncertainty in measurement (GUM) – Suplement 1: Numerical methods for the propagation of distributions".

[7] C. Domański, K. Pruska, "Unclassical statistic methods", PWE Warszawa 2000.

[8] A. M. Zoubir, B. Boashash, "The bootstrap and its application in signal processing", *IEEE Signal Processing*, pp. 56-76, January 1998.