

Alvaro José Abackerli

abackerli@ipt.br
Inst. de Pesq. Tecnol. do Est. de São Paulo – IPT
05508-901 São Paulo, SP, Brazil

Paulo Henrique Pereira

pereira_paulo_h@cat.com
Caterpillar Inc
Americas Operations Division
E. Peoria, IL 61630, USA

Nivaldi Calônego Jr.

nivaldi.calonegojr@gmail.com
Univ. do Estado de Mato Grosso – UNEMAT
78200-000 Cáceres, MT, Brazil

A Case Study on Testing CMM Uncertainty Simulation Software (VCMM)

Virtual Coordinate Measuring Machines (VCMMs) are software packages aimed at providing uncertainty estimates for tridimensional measurements. Since they deal with metrology data to derive uncertainty estimates, which are to be used for making decisions related to product quality, they must be evaluated. Although there are international standards to deal with software testing for metrology, the evaluation of VCMMs performance is still in the experimental stage. Here a testing framework was organized focusing on the definition of the test context, the test purpose, the performance criteria and the testing strategy, including the usability of the VCMM interface, which is a common focus in software engineering but not in tests of metrology applications. Experimentation and reference data were also used to check the adequacy of uncertainty estimates against measurement results. The executed tests show that it is possible to provide the necessary evidence of acceptable VCMM performance. It is also demonstrated that no single testing strategy is sufficient to provide the necessary evidence to validate the VCMM from the metrology standpoint.

Keywords: VCMM, coordinate measuring machine, software testing, uncertainty

Introduction

The increasing use of computers in dimensional metrology has highlighted the need to verify the software as an integral part of metrology systems. Early studies were conducted where it was demonstrated that software errors could compromise significant fractions of coordinate measuring machine (CMM) performance, even when used to calculate simple geometries such as circles, spheres, cylinders, etc. (Weckenmann and Heinricholski, 1985; Wäldele et al., 1993).

In recent years, when measuring equipment has become much more powerful and software dependant, questions regarding software performance in metrology have been raised demanding further concern and research to establish its current status (Cox and Harris, 2000; Goulding, 2003; Esward et al., 2003; Greif et al., 2006; Richter, 2006; Carbone et al., 2008; Habra, 2008; Liu et al., 2008). The same applies when simulation models are integrated in the metrology software to be used to preview the CMM behavior, which is the case of the Virtual Coordinate Measuring Machines (VCMM) (Phillips et al., 2002; Takamasu, 2002; Levin, 2008).

In practice, it becomes too costly and time consuming to determine exhaustively that a particular software and its simulation model are valid over the entire domain of applicability. However, to validate software it is sufficient to substantiate *that a computerized model within its domain of applicability possesses a satisfactory range of accuracy consistent with the intended application*, as by Sargent (1999).

Validity (Greif et al., 2006; Richter, 2006; Levin, 2008) is usually verified through a selected subset of tests until sufficient confidence is obtained that the model is valid for an intended application. For this reason, authors state: *a simulation model of a complex system can only be an approximation to the actual system, regardless of how much effort is put into developing the model. There is no such thing as an absolutely valid [simulation] model. The more time (and hence money) is spent on model development, the more valid the model should be in general.* (Page et al. apud Law and Kelton, 1997).

Taking this principle for testing CMM software, international standards (ASME, 2002; ISO, 2001) are designed to address a finite

set of calculations in a defined domain of applicability, and hence they are capable of providing sufficient evidence only about the quality of the performed calculations. Therefore, since only a small number of system functionalities are addressed by these tests, i.e. the calculations, the performance of the entire CMM software system cannot be assured.

Although the formal approach for software validation is scarcely seen in the metrology literature (Greif et al., 2006; Richter, 2006; Levin, 2008), it can also be applied to testing the Virtual Coordinate Measuring Machines – VCMMs (Takamasu et al., 2002; PTB, 2002; Metrosage, 2003). However, many VCMMs use inferential methods such as Monte Carlo Techniques to derive uncertainty estimates (Phillips et al., 2002) and this introduces new perspectives on software testing due to its stochastic nature, making the problem of checking the validity even more demanding.

To provide a practical insight on metrology oriented software testing, involving stochastic procedures thus beyond the scope of ASME and ISO standards (ASME, 2002; ISO, 2001), in this work a case study was used to investigate some relevant software testing techniques and outline a feasible set of steps to test a VCMM. The investigation was oriented to highlight the need for traceability of simulation results to basic SI units (INMETRO, 2003; BIPM, 2006), which is a typical requirement of metrology software (Butler et al., 1999), but not commonly discussed in the context of software testing and engineering (IEEE, 2004).

Nomenclature

- A = Elevation angle for probe orientation, degrees
 B = Azimuth angle for probe orientation, degrees
 r = Radius of a circle, mm
 $u(r)$ = Uncertainty on the radius of a circle, μm
 (X, Y, Z) = Coordinate measuring machine axis
 W = Direction vector of the workpiece
 P = Direction vector of the probe tip

Greek Symbols

- θ = Spacing angle for sampling a circle perimeter, degrees
 σ = Radial standard deviation in a circle measurement, μm

Background on Software Testing for VCMM

There are many available definitions of software testing that rely on non-execution based tests, or static verification (Schach, 1996; Kobrosly and Vassiliadis, 1998), but the insights provided by these approaches are unsuitable to test the VCMM due to the lack of emphasis on dynamic verification and also because of the limited capacity of such tests to guarantee the appropriate results against the expected behavior.

In the present context, software testing is defined as *a dynamic verification of the behavior of a program on a finite set of test cases, suitably selected from the usually infinite domain, against the specified expected behavior* (IEEE, 2004). This definition calls for *dynamic verification on a finite set of test cases*, which means running the software with valid inputs in a number of test cases considerably smaller than the very large combination of possible cases of use. It also relies on checking the *selected cases* against the *expected behavior*, demanding definitions of tests and test criteria that fully characterize the real world subjected to simulation.

Notice that, due to the nature of measurement uncertainty (JCGM 100:2008, 2008a), the simulation software does not provide a single correct answer for each set of input parameters, making the evaluation of its expected behavior a complex task that is heavily dependent on expert knowledge.

Considering the above mentioned concepts, in the present discussion the procedures for verification and validation are treated alike as the *set of procedures designed to address the software quality directly, using testing techniques which can locate defects so that can be addressed* (IEEE, 2004). In simpler words, here a broader approach is used to characterize validation and verification at once as the complete set of actions included in the test procedure (Adriole, 1986).

Even under such a broad interpretation of software testing, validation and verification, it is possible to decide whether an observed test output is acceptable. To do so, it becomes necessary a clear definition of test objectives, the applicable testing techniques and, ultimately, the measures of test success (test criteria), which are discussed in the following sections.

Test Objectives and Techniques

Testing objectives and techniques vary depending on the software behavior one is looking to reveal, which may include testing for conformance, configuration, usability, reliability, installation, recovery, performance, back-to-back, etc. (IEEE, 2004).

Here, test for conformance, functional test or correctness tests are used to show any undesirable behavior of the simulation software considering its primary function of generating estimates for task-specific uncertainty (Wilhelm et al., 2001). The applicable strategies include checking against extreme conditions (e.g. zero) and comparing simulation results with known models (e.g. symmetry properties, sample-point distribution). According to Sargent (1992), these strategies are intended to verify whether the software output is valid for a given simulation scenario.

Tests relying on the comparison of calculated simulation results with experimental reference values are also important, given the purpose of the simulator is to generate metrology-related information to be associated to measurement results that are traceable to the international system of units – SI (INMETRO, 2003; BIPM, 2006). Notice that this is not a common approach for commercial off-the-shelf software (not related to metrology) since usually they are not intended to generate results related to a physical quantity. However, since valid metrological results must be traceable to physical quantities, the comparisons with reference values are used to check the compatibility of experimental and simulated results (Butler et al., 1999).

Given the complexity of setups used in coordinate measurements and the wide variety of technical backgrounds among CMM users, configuration and usability tests are also applied to verify how easily the software can be learned and how well it supports user tasks (Abackerli, 1998).

Selection of Test Cases and Failure-Success Criteria

In this investigation the applicable approach for selecting the test cases can be formulated as a combination of the tester's intuition and experience to choose the applicable fault-based test techniques as per the Swebook (IEEE, 2004), specifically aimed at exploring categories of probable system behaviors. This can be done even without formally specifying the ideal system behavior, since expert knowledge can be used to predict expected results for a set of given simulation scenarios. This is the case, for example, when investigating thermal effects on the measured part and on CMM scales, given the appropriate combinations of temperature, thermal expansion coefficients, their uncertainties, etc.

Based on this approach, measures of test success can be devised to evaluate the program under test and its results (i.e., correctness of simulation results), and to check the test's ability to provide information about the overall software quality based on the performed tests (i.e., test coverage). Here, the only performance measure used was the agreement between simulation results and the expected behavior, using either experimental results or known solutions to well defined problems. Details of all performed tests, their results and the achieved improvement from tests are discussed in the following section.

Performed Tests

Based on the above discussion of test context and objectives the attention was focused into two particular groups of tests: configuration and usability testing and conformance, functional or correctness testing. Their details are discussed below.

Configuration and Usability

Configuration and usability are here defined as the ability of the simulation software to faithfully represent the details of a real measurement, which ultimately comes down to the simulation fidelity (Phillips et al., 2003). How accurately the software internally handles these details is a separate issue addressed below.

All real measurements have an almost infinite list of influence quantities that can affect the measurement result while every simulation software has only a finite list of uncertainty sources that can be included in the uncertainty evaluation. In the case under discussion, poor fidelity would not allow the details of the measurement to be properly represented in the simulation software, and hence the simulation would provide an uncertainty statement for a measurement scenario different from the real one. Good fidelity simply means that the details of the actual measurement can be adequately represented in the simulation.

Uncertainty contributors to be represented in the simulation can be considered either as intrinsic to the measurement system (i.e. attached to the CMM) or extrinsic. Extrinsic factors may include operator effects, workpiece fixture variation; workpiece form error and its interaction with the sampling strategy; thermal properties and their effects on the workpiece; workpiece contamination (e.g. dirt, coolant, etc.) and many others. Intrinsic factors may include multiple styli (either a fixed star probe or an articulated stylus), scanning probes, rotary tables, CMM dynamic effects (changes in acceleration and velocity values), etc (Wilhelm et al., 2001).

The usability/configuration analysis was performed through the simulation setup while comparing the simulation process with the actual measurements. It was noticed during the tests that the probe system representation was insufficient to handle a wide variety of probe system configurations, despite the good calculation capability of the tested software. A detailed interface examination led to a review of the software interface using a hierarchical organization of the probe information, as illustrated in Fig. 1.

As per Fig. 1, the new interface is capable of guiding the user through a top-down configuration, allowing selecting probe system components and assigning the applicable performance data. The hierarchical organization of probe system information created a new configuration process in the simulation setup that is very similar to the actual one. Besides the more comprehensible setup, the new probe interface helps to educate the user about probe systems and software setup, thus improving important aspects of configuration and usability. The final result of simulated probe setup process is now a list of probe configurations that are very similar to the actual procedure of probe qualification used in real measurements.

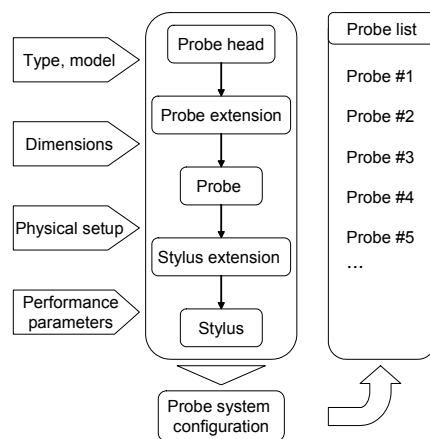


Figure 1. Schematic probe system interface.

Conformance, Functional or Correctness Testing

Conformance, functional or correctness testing was executed using fixed values, the invariance property, reference results, the comparison with well known mathematical models and with experimental data; each of which is discussed in the following paragraphs.

A) Fixed Values

A powerful technique to detect problems in measurement uncertainty simulation is to use fixed or reference values. A fixed or reference value is a special case result that is known under a particular and well-defined simulation condition (Cox and Harris, 1999). It can be obtained by a variety of ways, including using other verified software known to produce a mathematically correct answer under specified conditions. Often the reference value can be obtained using some type of invariant quantity. Two applications of fixed values are discussed below, one using the invariance property of a simulation result and another using a known reference result.

Invariance: The simulation software under consideration can be configured in a variety of ways including deactivating entire classes of uncertainty sources. Thus it is possible to examine a geometrically perfect virtual CMM and introduce as the only uncertainty source of the probing errors. In this case, it is expected that the uncertainty of a simulated measurement, e.g. the diameter of a ring gauge, will not depend on the particular location in the CMM workzone, provided the probing pattern does not change (within

statistical fluctuations) when the machine is moved to another position. The simulation software can place the workpiece at any location in the CMM work zone. It was observed that, after many successive workpiece relocations, the simulated uncertainty was increasing due to the way in which new positions were calculated from previous ones by allowing minor round-off errors to add up. The solution was to transform each location from a single reference position, not the previous position, proving the invariance property to be a valid approach for testing in this case.

Reference result: If a well-tested reference software is available it can be used to create known values. For example, NIST has an algorithm testing service used to check least squares fitting algorithms for numerical correctness (Hopp and Levenson, 1995; Shakarji, 1998; Hogan et al., 2001). This software was used to produce a virtual circle perturbed by a systematic form error (e.g. three lobes) of specified amplitude. Using the reference software, a large set of results was produced for different phase combinations of the sampling points and form error. Two times the standard deviation of these results (approximately 95% confidence limit) was compared to the CMM simulation for an identical form error and sampling strategy. It was noticed that the reference software and the CMM simulation software converged to different values for the uncertainty in the circle diameter. Upon further investigation it was discovered the problem to be an error in the simulation software that created N equally spaced points on a circle, superimposing points at zero and 360 degrees and so creating a double counting of uncertainty on that particular location. A revised version of the simulation software corrected the problem, and agreement to within 0.001 mm or less was achieved between results of the VCMM and the reference software. That made for a strong case to use reference values to test simulation software as far as the test cases can be set up in both software packages, i.e., in the reference software and in the simulator.

Temperature variation can also be easily used to test the simulation software because the model for thermal compensation is well understood. A workpiece temperature can be set to any value, e.g. 21 °C, and a thermal expansion coefficient set to 10 ppm/°C, both with zero uncertainty. These parameters yield a systematic error of exactly +10 μm in a one meter feature of size when measured without thermal compensation, and the simulation software should produce the same average result. Tests were carried out with different settings and different features of size, and the results were properly confirmed in the simulation, showing the effectiveness of relying on well known properties of measured materials.

B) Comparison with Known Models

Sometimes specific uncertainty results can be determined analytically. For example, the measurement of a circle with at least three points, separated by an angle θ between the sampling points, is known to produce an uncertainty in the radius given by $u(r)$ in the equation below, where σ is the standard deviation of a Gaussian distribution of radial perturbations (Phillips et al., 1998). Equation (1) illustrates this uncertainty calculation.

$$u^2(r) = \frac{1 + 2\cos^2\theta}{2(1 - \cos\theta)^2} \sigma^2 \quad (1)$$

The tested software has the ability to simulate the measurement of a reference diameter under perfect measurement conditions, with only the probe having a known standard deviation (σ). A virtual diameter of 5 mm was repeatedly simulated in an ideal measurement scenario, with no errors or uncertainties, except for the probe, to generate uncertainty estimates with different angles (θ) between points. The above equation was then used to

calculate the uncertainty of the measured diameter and the results are shown in Fig. 2.

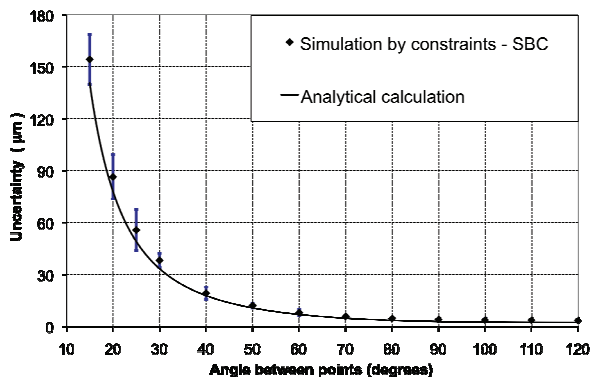


Figure 2. Uncertainty on a reference diameter.

Figure 2 clearly shows the agreement between the simulation results and the analytical solution, which is contained in the 95 % confidence limit for every simulated angle. The plot also shows the

decrease of the uncertainty dispersion with the increase of the angle between points, which agrees with the predicted behavior due to the improvement in the sampling strategy. Again, this points to the good approach of selecting known mathematical models of well established measurement problems and use them as a valid reference data for testing.

C) Experimental Comparison

The most direct method of examining the validity of an uncertainty statement produced by computer simulation is by comparison with physical measurements of calibrated artifacts. The physical measurements for this purpose should be performed so as to allow good simulation fidelity, providing a calculated uncertainty statement that can be directly interpreted, with at least 95 % of the measurement errors.

In the present discussion, two reference artifacts were used to generate results to be compared with the simulation, a 150 mm diameter ring gauge, XX class and a 300 mm diameter ground reference disc (custom made) on two different CMMs, a moving bridge model and a moving table one. Figure 3 illustrates both artifacts under measurement.

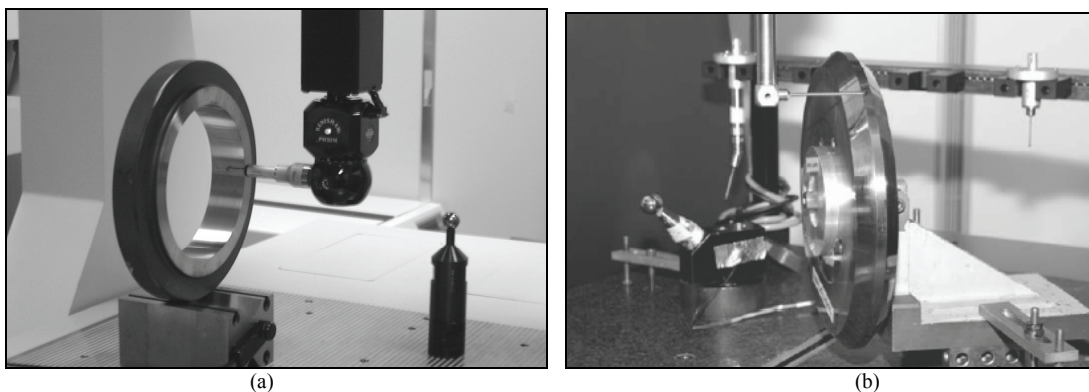


Figure 3. Reference measurements. (a) Above: moving bridge, (b) Below: moving table.

The ring gauge had its diameter and roundness measured from which an approximate form error of 0.25 µm was found. For that, it was sampled three times with 24 equally spaced points in each measurement. The moving bridge machine used for this measurement had a work zone of 460 x 460 x 385 mm. The CMM was equipped with an articulated head and a touch trigger probe. The coefficients of thermal expansion were 9.9 ppm/°C and 11.8 ppm/°C for the machine scales and test artifacts respectively with uncertainty

estimates of 1 ppm/°C. The average room temperature was 20.1 °C with variation limits of ± 0.05 °C.

The measurements were performed with the CMM error compensation turned OFF and ON for the moving bridge model. This compensation corrects for geometrical errors in the CMM structure, giving an effect of two different CMMs. Table 1 gives the performance test results for this CMM in both cases and for the moving table model (only with compensation enabled) as per ISO 10360-5:2000 (2000) and ASME B89.4.1b-2001 (2001) standards.

Table 1. Performance test results.

Performance test	Coordinate Measuring Machines		
	Moving Bridge Machine		Moving Table Machine
	Compensated	Non-compensated	
X linear accuracy ^[3]	5.5 µm	11 µm	1.2 µm
Y linear accuracy ^[3]	6.5 µm	12 µm	1.4 µm
Z linear accuracy ^[3]	3 µm	88 µm	2.2 µm
Volumetric performance ^[3]	13.5 µm	188 µm	3.9 µm
Offset probe volumetric performance ^[3]	50 µm/m	76 µm/m	6.7 µm/m
Repeatability ^[3]	0.8 µm	0.8 µm	0.34 µm
Probe performance (25 points) ^[1]	7 µm	7 µm	0.82, 1.68, 1.82 µm ^[2]
MPE _{AF} ^[1]	10 µm	16.6 µm	---
MPE _{AS} ^[1]	2 µm	3.1 µm	---
MPE _{AL} ^[1]	3.8 µm	5 µm	---

Notes: ^[1] ISO 10360-5 parameters; ^[2] Three styli were used; ^[3] ASME B89.4.1b-2001 parameters.

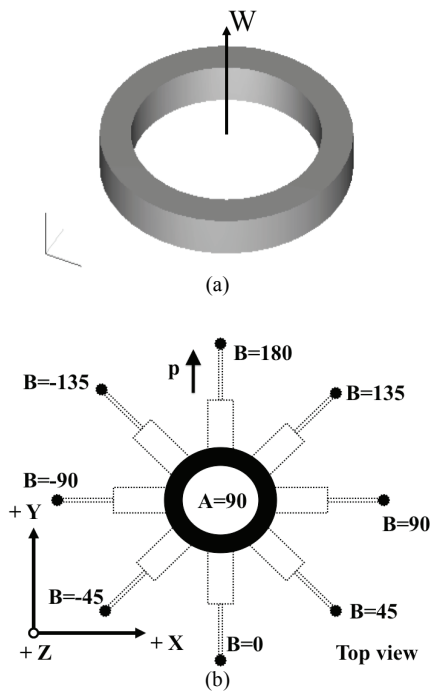


Figure 4. Reference measurements. (a) Above: the workpiece reference, (b) Below: the probe orientations

The adopted strategy for part placement generated two measurements at $B = 0^\circ$, Fig. 4-b, one when the probe was vertical ($A = 0^\circ$) and the ring was on the XY plane, and another when the probe was pointing along the $-Y$ axis of the CMM ($A = 90^\circ$, Fig. 4-b) and the ring was on the XZ plane, Fig. 3-a. The first group of repeated measurements was made with the ring mounted on its edge, so $A = 90^\circ$ in Fig. 4-b, while the probe was rotated about the Z direction in positions evenly spaced every 45° . The first part of Fig. 3-b shows one of these measurements made with $A = 90^\circ$ and $B = -90^\circ$. In the remaining measurement setup the ring gauge was also mounted on its edge but tilted 30° from the vertical plane, creating therefore an angle of 60° between the direction vector W and the direction given by the $+Z$ axis.

In Figure 5, the measured errors were averages obtained from three measurement repetitions in each test. The simulated uncertainties were the average of ten repetitions, each of which made with 250 samples, so a total of 2500 simulations were used per experiment. In the X axis there is the indication of the performed tests, and in the Y axis the measured errors and the simulated uncertainties, both expressed in micrometers. Figure 5 has four different areas to identify the different compensation modes used in the tests. The symbols at the bottom of each figure identify the measured errors and the simulated uncertainties in each test case.

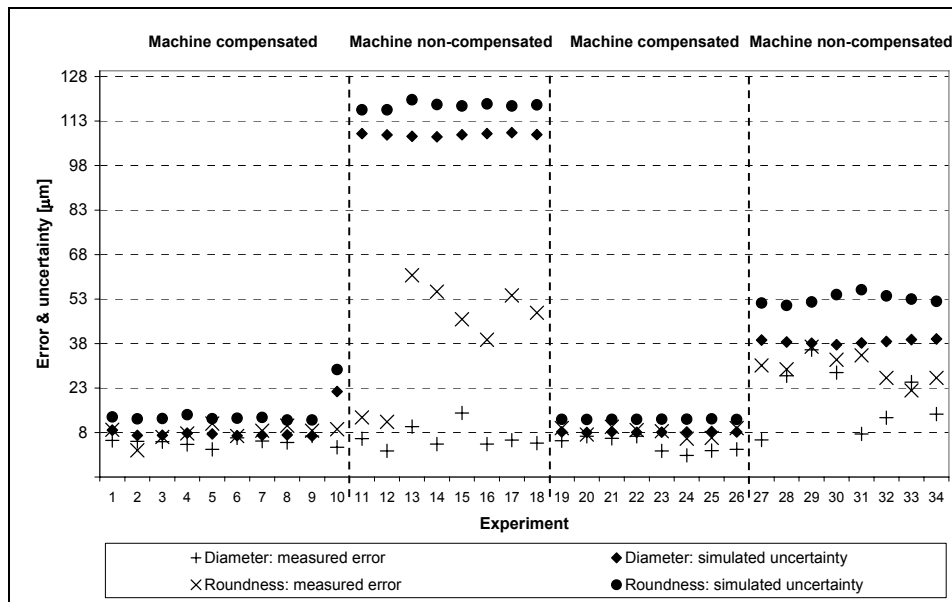


Figure 5. Summary of ring test results (moving bridge machine).

As expected, Fig. 5 shows measured errors smaller than the associated uncertainties, hence in agreement with the expectation of uncertainty as per its definition as a *parameter that characterizes the dispersion of the quantity values that are being attributed to a measurand, based on the information used* (INMETRO, 2003a; JCGM, 2008b).

The moving bridge machine showed a large dispersion of measurement results in the tests 11 to 18 which are associated with equally large uncertainty estimates. Although they appear larger than expected, from these results there was no evidence of poor VCMM performance since the non-compensated measurements in these tests generated diameter errors up to $15 \mu\text{m}$ and roundness up

to $61 \mu\text{m}$. The smaller values of diameter errors were likely affected by the fitting algorithm attenuating the sampling effects on the data.

Similar tests were run using a high accuracy moving table machine (Fig. 3) with a workzone of $800 \times 600 \times 600 \text{ mm}$, equipped with an analog probe whose performance values are also given in Table 1. The temperature was maintained at $20 \pm 1^\circ\text{C}$ with an estimated uncertainty of 0.1°C . The expansion coefficients were $9 \text{ ppm}/^\circ\text{C}$ and $11.8 \text{ ppm}/^\circ\text{C}$ for the machine scales and the workpiece. The 300 mm reference disc had an approximate form error of $0.2 \mu\text{m}$ and was sampled on 360 points. The styli used in these measurements were “L” shaped with each leg 80 mm . Due to some fidelity issues with the software version under test the

dimensions of the styli used were approximated in the simulations. However, there was no evidence of significant changes in the calculated measurement uncertainties due to the approximation of the probe styli representation in the simulation software. Figure 6 shows similar test results to those already discussed, but here all tests were performed with the machine compensated for errors.

Similar analysis and conclusions can be extracted from Fig. 6 for the moving table CMM data. Again, no evidence of poor VCMM performance can be pointed out from these results. Moreover, the simulated uncertainties were quite compatible with the performed measurement tasks under the described measurement conditions, which indicate a strong evidence of the adequacy of using the discussed experiments for testing the simulated uncertainties.

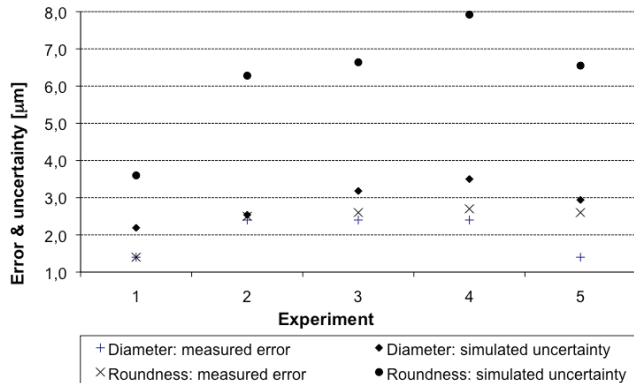


Figure 6. Summary of disc test results (moving table machine).

Additional Aspects of the VCMM Testing

Finally, from the performed experiments a few other aspects can be pointed out as important concerns, when investigating the performance of software designed for generating uncertainty statements.

Blunders: it has been said that the largest source of measurement error is misinterpretation of Geometric Dimensioning and Tolerancing (GD&T) (ASME, 2009; ISO 2004) callouts on drawings. While such problems clearly exist, mistakes of this type are considered by the Guide to the Statement of Uncertainty in Measurement (GUM) (JCGM, 2008a) as “blunders” and are not to be included in the measurement uncertainty statement. To the list of blunders one can include improperly setting up an uncertainty simulation scenario and hence getting inappropriate uncertainty statements.

Uncertainty of the Measurand: the GUM identifies that a poorly specified measurand is a legitimate source of measurement uncertainty as it gives rise to multiple possible “true values” all of which can be assigned as the “value of the measurand”. It is the metrologist’s obligation to consider this issue and quantify it. By carefully defining the measurands, this source of uncertainty was not considered in this investigation (which is also frequently overlooked in traditional uncertainty analysis).

Simplified, Incomplete, or Incorrect Mathematical Modeling: assuming that the simulation software allows the inclusion of a particular influence quantity, e.g. the form error of a workpiece or the use of multiple styli, some sort of mathematical model must be invoked to calculate the errors due to this influence. These models can range from complex ones, derived from first principles, to simple approximations, to just plain wrong models that do not describe the behavior of the influence quantity.

Input Parameters: all uncertainty models require some input parameters to estimate the effects of influence quantities. These input values may be extracted from actual measurement data collected on the CMM under simulation, or may be user supplied. These input values describe such details as the type of probe, the magnitude of the probing error, the type of CMM, the magnitude of the CMM structural errors, the type of workpiece form error and its magnitude, etc. Even a highly detailed and accurate model will yield nonsense output if the input values are incorrect (garbage in-garbage out principle).

Coding Errors: in all complex software systems coding errors are present. These can range from simple incorrect logic statements to more subtle effects such as failing to clear registers or unanticipated interactions, when function or object calls occur in varying sequential order.

Conclusions

This case of testing CMM measurement uncertainty statement produced by computer simulation showed a variety of results. All errors found in the physical measurements were bounded by the corresponding calculated uncertainty intervals. The physical measurements are only able to test error sources actually present during the measurement. For example, if the tested CMM had no XZ or YZ axis squareness error, then it would be impossible to determine whether the simulation software could correctly handle this class of error. Since, in general, the details of the CMM errors are unknown, physical testing on several machines is desirable in order to get a better representation of all potential error sources to more fully exercise the simulation software. To the extent that the uncertainty appeared to be overestimated in some cases of the physical measurements, Figs. 5 and 6, this may indicate that further refinements in the simulation software may be useful.

The physical measurement results were unable to detect problems in the simulation software involving the part placement transformation matrices, the switching probe error, or the redundant point sampling strategy error, as these effects were typically less than 1 µm and not easily discernable in the results. Reference values used for testing can catch these errors relatively quickly in cases where one knows what reference value to employ and what corresponding measurement to simulate. This suggests that a well-documented list of reference value tests might be a useful tool before starting a more extensive (and expensive) program of physical measurements of calibrated parts.

It is worth mentioning that in a simulation there will always be real measurement factors that are not fully represented, creating the need to account for uncertainty sources that are not included in the simulation. In testing simulation software against actual measurements great care must be taken to minimize the effect of actual influence quantities that are not accounted for in the simulation setup. Hence, measurement tests must be formulated to include all relevant influence factors with good fidelity, so the calculated uncertainty can be properly compared with the observed measurement errors.

Finally, back-to-back tests can also be implemented under a well controlled environment using similar software packages in the market (PTB, 2002; Takamasu et al., 2002). However, since test outputs will naturally differ to some extent due to the differences in the simulation scenarios, the compared packages can handle; back-to-back tests should not be considered until further analysis of their validity and cost effectiveness. Further testing involving reliability test, installation test, recovery and performance tests were not covered in this study. Future investigation may include combinations of them to test the software behavior under different simulation scenarios.

Acknowledgements

The authors would like to thank their own institutions and their partner companies for supporting this research. Special thank is due to CAPES – *Coordenação de Aperfeiçoamento de Pessoal de Nível Superior* for sponsoring the project under grant number BEX 2300/02-8.

References

- Abackerli, A.J., 1998, "On the virtual measuring instruments: practical design considerations", In: *Annals of the CSME'98 Symposium on Manufacturing, Automation and Robotics*, Toronto, Canada, pp. 116-122.
- Adriole, S.J. (Ed.), 1986, "Software Validation, Verification, Testing and Documentation", Princeton – NJ: Petrocelli Books, ISBN 089-433-269-4.
- ASME B89.4.10-2002, 2002, "Methods for Performance Evaluation of Coordinate Measuring Machine Software", New York, NY: The American Society of Mechanical Engineers – ASME, 25 p.
- ASME B89.4.1b-2001, 2001, "Methods for Performance Evaluation of Coordinate Measuring Machines", New York, NY: The American Society of Mechanical Engineers – ASME, 78 p.
- ASME Y14.5M-2009, 2009, "Dimensioning and Tolerancing", New York, NY: The American Society of Mechanical Engineers – ASME, 232 p.
- BIPM. 2006. *The International System of Units (SI)*. 8th Edition. Bureau International des Poids et Mesures. Paris, France.
- Butler, B.P. et al., 1999, "Model validation in the context of metrology: a survey", Version 1.0 Teddington, Mddlx, UK: Centre for Information Systems Engineering – NPL, *NPL Report*, CISE 19/99, 69 p., ISSN 1361-407X.
- Carbone, P. et al., 2008, "A comparison between foundations of metrology and software measurement", *IEEE Transactions on Instrumentation and Measurement*, Vol. 57, No. 2, pp. 235-241.
- Cox, M.G., Harris, P.M., 1999, "Design and use of reference data sets for testing scientific software", *Analytica Chimica Acta*, Vol. 380, No. 2-3, pp. 339-351, doi:10.1016/S0003-2670(98)00481-4.
- Cox, M.G., Harris, P.M., 2000, "Guidelines to help users select and use software for their metrology applications", Teddington, Mddlx – UK: Centre for Mathematics and Scientific Computing – NPL, *NPL Report*, CMSC 04/00, 28 p., ISSN 1471-0005.
- Esward, T.J. et al., 2003, "Mathematics and Scientific Computing: recommendations for the software support for metrology programme 2004-2007", Teddington, Mddlx, UK: Centre for Mathematics and Scientific Computing – NPL, *NPL Report*, CMSC 20/03, 25 p. ISSN 1471-0005.
- Goulding, J., 2003, "New Directions – The Status of Mathematics and Software in Legal Metrology: recommendations for the software support for metrology programme 2004-2007", Teddington, Mddlx, UK: Centre for Mathematics and Scientific Computing – NPL, *NPL Report*, CMSC 19/03, 18 p. ISSN 1471-0005.
- Greif, N. et al., 2006, "Software validation in metrology: A case study for a GUM-supporting software", *Measurement*, Vol. 39, pp. 849-855, doi:10.1016/j.measurement.2006.04.005.
- Habra, N. et al., 2008, "A framework for the design and verification of software measurement methods", *The Journal of Systems and Software*, Vol. 81, pp. 633-648, doi:10.1016/j.jss.2007.07.038.
- Hogan, M.D. et al., 2001, "Information technology measurement and testing activities at NIST", *Journal of Research of The National Institute of Standards and Technology*, Vol. 106, No. 1, pp. 341-370.
- Hopp, T.H., Levenson, M.S., 1995, "Performance measures for geometric fitting in the NIST algorithm testing and evaluation program for coordinate measurement systems", *Journal of Research of The National Institute of Standards and Technology*, Vol. 100, No. 5, pp. 563-574.
- IEEE (Ed.), 2004, "SWEBOOK – Guide to the Software Engineering Body of Knowledge", Version 4.0 Los Alamitos, CA, USA: IEEE Computer Society, ISBN 0-7695-2330-7, <http://www2.computer.org/portal/web/swebok> [12 June 2009].
- INMETRO (Ed.), 2003, "Sistema Internacional de Unidades – SI", 8 ed., Rio de Janeiro, RJ: INMETRO – Instituto Nacional de Metrologia, Normalização e Qualidade Industrial, 116 p., ISBN 85-87-87090-85-2. (In Portuguese).
- INMETRO (Ed.), 2003a, "VIM- Vocabulário internacional de termos fundamentais e gerais de metrologia", 3 ed., Rio de Janeiro, RJ: INMETRO – Instituto Nacional de Metrologia, Normalização e Qualidade Industrial, 75 p., ISBN 85-87090-90-9. (In Portuguese).
- ISO 1101:2004. *Geometrical Product Specifications (GPS) – Geometrical tolerancing – Tolerances of form, orientation, location and run-out*.
- ISO 10360-5:2000, 2000, "Geometrical product specification (GPS): Acceptance, test and reverification test for coordinate measuring machines (CMM): Part 5: CMMs using multiple-stylus probing systems", Genève, Switzerland: International Organization for Standardization – ISO, 12 p.
- ISO 10360-6:2001, 2001, "Geometrical product specification (GPS): Acceptance, test and reverification test for coordinate measuring machines (CMM): Part 6: estimation of errors in computing Gaussian associated features", Genève, Switzerland: International Organization for Standardization – ISO, 19 p.
- JCGM 100:2008, 2008a, "Evaluation of measurement data – Guide to the expression of uncertainty in measurement", (GUM) 1st ed., Sèvres Cedex, France: Joint Committee for Guides in Metrology (JCGM). Bureau International des Poids et Mesures, 120 p.
- JCGM 200:2008, 2008b. *International vocabulary of metrology – Basic and general concepts and associated terms (VIM)*. Joint Committee for Guides in Metrology.
- Kobrosly, W., Vassiliadis, S., 1998, "Survey of software functional testing techniques", In: *Proceedings of the IEEE Southern Technical Conference*, Binghamton, NY, USA, pp. 127-134, doi: 10.1109/STIER.1988.95474.
- Levin, S.F., 2008, "Statistical methods and metrological validation of measurement system software", *Measurement Techniques*, Vol. 51, No.11, pp. 1162-1170.
- Liu G. et al., 2008, "Analysis of key technologies for virtual instruments metrology", *Proc. of SPIE 4th International Symposium on Precision Mechanical Measurements*, Vol. 7130, pp. 71305B-1–71305B-6, doi: 10.1117/12.819751
- METROSAGE, 2003, "PUNDIT/CMM: User Manual", Version 1.10 Volcano, CA, USA: Metrosage LCC, 140 p.
- Page, E.H. et al., 1997, "A case study of verification, validation, and accreditation for advanced distributed simulation", *ACM Transactions on Modeling and Computer Simulation*, Vol. 7, No. 3, pp. 393-424.
- Phillips, S.D. et al., 1998, "The estimation of measurement uncertainty of small circular features measured by coordinate measuring machines", *Precision Engineering*, Vol. 22, No. 22, pp. 87-97.
- Phillips, S.D. et al., 2002, "The calculation of CMM measurement uncertainty via the method of simulation by constraints", In: *Annals of the 12th Annual meeting of the ASPE*, Norfolk, VI - USA, American Society for Precision Engineering – ASPE, pp. 443-452.
- Phillips, S.D. et al., 2003, "The validation of CMM task specific measurement uncertainty software". In: *Proceeding of the ASPE summer tropical meeting – coordinate measuring machines*, Charlotte, NC, USA, American Society for Precision Engineering – ASPE, pp. 1-6. http://www.mel.nist.gov/publications/get_pdf.cgi?pub_id=822066 [12 June 2009]
- PTB, 2002, "VCMM User Manual: Models, Parameters & Operation", Braunschweig, Germany: Physikalisch-Technische Bundesanstalt – PTB, 93 p.
- Richter, D., 2006, "Validation of software in metrology", *Computer Standards & Interfaces*, Vol. 28, pp. 253-255, doi:10.1016/j.csi.2005.07.004
- Sargent, R.G., 1999, "Validation and verification of simulation models", In: *Proceedings of the 31st Conference on Winter Simulation: Simulation – a Bridge to the Future*, Vol. 1, pp. 39-48, <http://doi.acm.org/10.1145/324138.324148> [12 June 2009].
- Schach, S.R., 1996, "Testing: principles and practice". *ACM Computing Surveys*, Vol. 28, No. 1, pp. 277-279, <http://doi.acm.org/10.1145/234313.234422> [12 June 2009].
- Shakarji, C.M., 1998, "Least-squares fitting algorithms of the NIST algorithm testing system", *J. Res. Natl. Inst. Stand. Technol.*, Vol. 103, No. 6, pp. 633-641, <http://www.nist.gov/jres> [10 Jul 2009].
- Takamasu, K. et al., 2002, "International Standard Development of Virtual CMM (Coordinate Measuring Machine)", Tokyo, Japan: NEDO International Joint Research Project (FY 1999 – FY 2001) – Final Research Report, 159 p.
- Wäldele, F. et al., 1993, "Testing of coordinate measuring machine software", *Precision Engineering*, Vol. 15, No. 2, pp. 121-123.
- Weckenmann, A., Heinricholski, M., 1985, "Problem with software for running coordinate measuring machines: the use of virtual volumetric standards", *Precision Engineering*, Vol. 7, No. 2, pp. 87-91.
- Wilhelm, R.G., et al., 2001, "Task specific uncertainty in coordinate measurement", *CRP Annals - Manufacturing Technology*, Vol. 50, No. 2, pp. 553-563, doi:10.1016/S0007-8506(07)62995-3.