preprint of a paper published in:
proc. IEEE IMTC 1999, Vol.2, 953-957, Venezia, 24-26 May 1999 – ISBN 0-7803-5276-9

On the Metrological Evaluation of the Software Component of Intelligent Measuring Systems

A. Balsamo[‡], L. Mari[†]
[‡]CNR-IMGC, Istituto di Metrologia “G. Colonnetti”, Torino, Italy
[†]LIUC, Libero Istituto Universitario “C. Cattaneo”, Castellanza, Italy

Abstract

The evaluation of the metrological performance of the software in Intelligent Measuring Systems is discussed and some general strategies are proposed.

1. Introduction

The software component of Intelligent Measuring Systems (IMSs) has been playing a more and more important role, and the problem arises of how to evaluate its metrological performance.

In the case of traditional instruments, the operator is responsible to evaluate the metrological quality of the measuring system and, on this basis, the uncertainty of the measurement results. To accomplish this task, the operator usually exploits its subjective knowledge and professional experience.

The novelty of the IMS concept and architecture and the presence of a substantial software component in the IMSs makes them far too complex for a human being to be able to evaluate them metrologically according to the traditional approach.

Several authors have addressed the general issue of performance evaluation of intelligent instruments [1,2], as well as the more specific one of testing scientific software [3]. This paper discusses the nature and the main features of the problem of the metrological evaluation of the software component of IMSs and suggests some general directions to its solution.

2. Functional structure of IMSs

The functional structure of an IMS can be thought of as made of three components:

A. an *acquisition* subsystem, which interacts with the measurand(s) and transforms the obtained information into digital signals, i.e. numbers (measurement readings) ready to be processed by software; this subsystem performs several operations related to low level sampling strategies, e.g. sampling timing and triggering;
B. a *compensation* subsystem, which transforms the raw measurement readings output by the subsystem A into calibrated measurement values (measurement points) which are required to be traceable to recognised standards (e.g. national standards); this subsystem performs several operations, including the compensations for the calibration of the sensor(s) and for the relevant influence quantities;
C. an *evaluation* subsystem, which transforms the usually numerous measurement points output by subsystem B into a final result (measurement value), often of a measurand that would not be accessible by traditional measuring systems based on a single sensor.

For instance, a voltage signal is sampled by an ADC of a plug-in PC board (subsystem A) and a vector of values is obtained; these values are then compensated for known systematic effects (e.g. ADC non-linearity, temperature, etc.) by the subsystem B and updated accordingly; finally the FFT is calculated (subsystem C) to get the result, i.e. the harmonic spectrum of the signal.

Each of these subsystems consists of both hardware and software/firmware components.

3. Evaluation of the IMS performance

The performance of an IMS can be evaluated according to two different goals and strategies, typically relevant at different stages in the life cycle of an IMS:

1. the evaluation of the *general* performance of an IMS, e.g. for qualifying it against competitors; this is of interest for IMS manufacturers to state performance indicators on data sheets, for third part testers (e.g. testing bodies) to verify conformance to specifications, and for users to select the most suitable IMS on the market for their needs;
2. the evaluation of an IMS performance for a *specific* measurement task, to determine the uncertainty of a measurement result as obtained with a given

measurement strategy and in given environmental conditions; this is of interest for on-field IMS users.

A same measurement task can be carried out with different strategies (as accomplished by the subsystems A-C) chosen freely by the user. So no *a priori* type 2 uncertainty evaluation is possible, unless the IMS is confined to a limited predefined number of measurement tasks, thus loosing its attractive flexibility and versatility. In general, different measurement procedures result in different uncertainty, and each of them requires a non-trivial and time-consuming effort. As a result, the type 2 evaluation is very demanding, and not completely understood in the general case.

For instance a PC-based virtual instrument tasked to compute the correlation (or equivalently the regression line slope) of two supposedly linearly dependent quantities achieves very different uncertainties depending on the sampling strategy implemented by software in the experiment program: the wider the sampled range the lower the uncertainty. While in this simple case the uncertainty is easily computed as an analytical function of the range, in a more complicated case (e.g. the measurement of the phase margin of an automatic control loop), where the final value is possibly the result of a chain of intermediate and complex computations, the impact of different strategies on the measurement uncertainty is hard to predict.

3.1. Evaluating the general performance

The peculiar versatility of IMSs (i.e. their ability to measure different measurands with different procedures/ strategies of measurement) makes the type 1 evaluation according to a black-box strategy (i.e. the IMS dealt with as a whole) not practical, or even unfeasible. Therefore it is necessary to characterise the single subsystems separately, to evaluate their specific contribution to the global accuracy of the produced results, and hence to evaluate the metrological quality of the IMS.

In this respect, the three subsystems exhibit remarkable differences in the contributions of their software components. While in subsystem A hardware and software are deeply tied and require to be evaluated together, in subsystems B and C the relations between hardware and software loosen up and, more important, the overall metrological behaviour depends more and more on the software component only.

Type 1 evaluation cannot be comprehensive of any possible use of the IMS for obvious practical reasons. Therefore it requires a fair amount of standardisation to define and normalise performance indicators able to capture a high (and meaningful, for the intended typical applications) fraction of the IMS behaviour. For this reason type 1 evaluation is conventional in a sense, as opposed to type 2 evaluation which should give the measurement uncertainty according to the rigorous rules defined in the ISO-GUM [4].

To make the evaluation as general as possible, the IMS can be decomposed into subsystems tested separately. This is applicable in the case of the software components, particularly for those in the subsystem C: in fact they handle numbers only, no physical quantities being involved, so that pure numerical testing can be done in a meaningful way. Usually IMSs are provided by extensive libraries of software modules (e.g. filters and interpolators) offered to users as building blocks for developing application-specific programs; therefore a general software testing of the IMS can be performed by testing these primitives separately.

Again, the testing of these software modules needs conventions and standardisation, for the very same reasons as for the IMS as a whole. Unfortunately the availability of specific standards which normalise scientific software module testing is very limited. The authors are aware of one example [5] -still in the form of ISO/DIS, *Draft International Standard*- which deals with the computation of geometrical features (e.g. spheres, cylinders, cones) as least-squares best-fit associated to sets of point coordinates, a topic of interest in the field of coordinate metrology.

3.2. A metrological-oriented evaluation

In general, the testing of software for IMSs requires at least two different competencies: metrology and software engineering. Our emphasis is here of the former. In other words, we assume that the software implementation of the chosen algorithms is reasonably good and bug-free. Even with this assumption, the software evaluation cannot be exhaustive, in general, for the virtually infinite number of different input data the software can be applied to. The extension of the evaluation to software-specific issues like the presence of bugs or poor engineering would increase enormously the amount of effort required, and deviate from the metrological goal of this analysis: whether the chosen algorithms are correctly implemented.

Further, a corollary of this assumption is that the hardware where the software components of subsystems B and C run is reliable enough to be considered as an ideal “software machine”.

Let us give an example to clarify why it must be accepted that even a thorough test cannot be exhaustive for metrological software. Non-linear problems -including the broad and metrologically interesting class of non-linear optimization- are usually solved numerically by iterative methods, e.g. Gauss method. Often it can be even proved theoretically that a particular method converges to the solution *providing that the starting approximation is sufficiently close to the* *solution itself*: in the solution space a convergence zone exists about the solution. When the starting approximation preliminarily computed by software from the input data lies very close to the border of this convergence zone, even very small variations in the input data may have a dramatic impact on the computed result, like failure to converge at all, or convergence to a different result, e.g. a different local minimum. This show that, given *n* passed tests on a software piece, there is always a risk that the (*n*+1)-th fails. However a proper design of the test keeps this risk to a minimum [6].

Aside the above mentioned need to capture the software behaviour with a necessarily limited number of tests, the testing of software modules of metrological interest brings another problem: the availability of input reference data sets and reference results to compare with. Two ways are possible to solve the problem: either through *reference software* of sufficiently good quality to ensure accurate enough reference results for all practical cases, or through *data generators* which generate reference data sets with theoretically known reference results. The former solution is the most straightforward, but requires reference software which is usually more difficult to develop than the software it will test, because of the required assurance of correctness in the whole input data domain. Again in the field of coordinate metrology, two EU-funded projects explored this solution in detail [7,8]. The latter solution [9] has the advantage to require a less complex implementation, with no iterative algorithms, and to rely on the theoretical proof that the solution is correct.

3.3. Evaluating the performance for specific tasks

We propose a computer-intensive methodology as an approach to the type 2 evaluation. In principle, if one has time enough to repeat a measurement task of a calibrated standard very many times while changing all possible influence quantities in their expected ranges, a type A evaluation (i.e. based on statistical analysis [10]) of the overall uncertainty would be possible by taking statistics of the obtained results and using the calibration value of the standard as a reference. However this procedure is very time-consuming and even unfeasible for the usual impossibility to control all influence parameters at will.

If the subsystem A is substituted by a software simulator which perturbs the measurement points taken on a virtual nominal standard on the basis of reliable models of the effects of all influence quantities, the process is speeded up enormously because is all done in software. The advantages of this approach are:

1. the uncertainty evaluation is tailored exactly to the measurement procedure implemented by the user, and therefore takes account of the chosen strategy;
2. the time required for the evaluation is short -being computers getting faster and faster- and this allows thorough investigations *on-line*;
3. a physical calibrated standard is not required for the evaluation; this removes the problem how to have it calibrated (in the case of very complex tasks it might not be easy at all) and allows *off-line* evaluations valuable for comparing alternative strategies in advance;
4. the variability of the influence quantities is explored by simulation and does not require control over them.

On the contrary, the disadvantages are:

1. a detailed and reliable model of the physical behaviour of the IMS is required, a task that is not trivial in the case of complex IMSs; however it must be realised that the model is required to *simulate* -and not to *predict* and *compensate for*- the influence quantity effects: therefore it can be based also on not measured -and even non observable- parameters;
2. all model parameters must be associated with their uncertainty, including possible correlation, i.e. the full variance-covariance matrix must be determined; this is to include the calibration of the sensor(s) to make the measurement traceable;
3. the value used as reference for the statistic evaluation of the result is assumed to be that obtained by the subsystems B and C of the IMS when input with the nominal (not perturbed) measurement points; therefore possible systematic deviations introduced by subsystems B and C are not captured with this approach; this is the most serious metrological limitation at the present state of the art. However it is the authors' opinion that this possible systematic deviations are negligible in a majority of cases, particularly in well-conditioned measurement tasks; additional testing is required to investigate the significance of this effect in ill-conditioned cases.

4. An application example: the CMMs

IMSs are found not only in the electrical and electronic field, where the increasing availability of plug-in boards promotes the use of open architecture IMSs; there is at least an example of IMS in the mechanical field too: the Co-ordinate Measuring Machines (CMMs). Their diversity from electrical IMSs and still their conformity to the above general description make them a good benchmark for the consistence of the approach.

A CMM is a measuring instrument made of a basement and three carriages movable orthogonally to each other, so that the last carriage in the chain (*ram*) can move in space with respect to the basement that carries the piece under measurement. Each carriages is equipped with a scale which measures the displacement to the previous carriage in the chain; the three readings together give a coordinate triplet of a point localised by a probing system attached to the ram. Very complex geometry's can be measured (e.g. an engine block or a turbine blade) because the individual points are evaluated by software to get overall measurement values, e.g. the concentricity of two opposite sided bores, or the convexity of a surface portion. To formulate a particular measurement procedure, a specialised software language is provided running in the CMM computer, by which the number and disposition of the probed points, as well as the subsequent computation, are specified.

For CMMs, the subsystem A is the mechanical structure (basement and carriages, scales, probing systems), and the electronic servos which move it; the output measurement readings are the coordinate triplets read by the scales. These triplets are then compensated by software for the CMM geometrical errors due to deviations from straightness ad mutual orthogonality of the carriage guideways; a vector is added to each point, and new measurement points are obtained. They are the output of subsystem B. At last, the individual points are input to a software program developed by the user (subsystem C) on the basis of the primitives made available by the CMM language. This way geometrical elements like planes, cylinders and toruses are evaluated first, and then operators like distance, intersection and projection are applied to get the final result.

The general performance of a CMM is evaluated by a standardised procedure defined by [11]. It is recognised that this test is but nearly sufficient to full characterise the CMM in all operating conditions; however the test is useful at least for the following reasons:

a) different CMM manufacturers have a uniform way of expressing performance figures of their CMMs, thus enabling a potential purchaser to compare among competitors;
b) it gives a means to define by contract when a CMM is to be accepted and the price paid; this is important in view of the high costs of CMMs (20.000 to 500.000 Euros);
c) it is recognised as the state of the art in performance evaluation of CMMs, and is generally accepted by inspectors of Quality Systems according to ISO 9000.

The evaluation of uncertainty of CMMs is not an easy task at all; National Metrological Institutes world-wide are doing research [12,13] on this topic. There seems to be a convergence of approaches toward simulation techniques. An error vector is added to each nominal measurement points according to an error model, and the dispersion of the computed results obtained is taken as the uncertainty. Even if this technique is still being investigated and validated, the preliminary results obtained are encouraging.

It is the authors' opinion that the next generation CMMs will be equipped with sophisticated enough software to yield the measurement uncertainty together with the measurement value, in a quasi-automatic way. This does not mean that the operator will not be asked for competence any longer; just that the competence required will move down to a level in the reach of a majority of (trained) operators.

5. Concluding remarks

It is worthwhile pointing out that the topic covered in this paper arises several significant epistemological issues.

IMSs are contributing to blur the traditional distinction between direct and indirect measurement, physical experiment and computation, and physical experiment and simulation.

From the standpoint of such distinctions , physical and informational domains -or, in procedural terms, state transitions and data processing- are essentially very different, and the measurement is the operation that sets the bridge between these two domains. On the contrary, both physical transformations (by subsystem A) and informational computations (by subsystems B and C) are properly performed within IMSs.

The consequences of this new situation are still to understand in their epistemological implications.

References

[14] A. Balsamo, S. Sartori, *Principle of calibration of intelligent instruments*, IMEKO Acta XII, 1994, 887-892.

[15] L. Finkelstein, *Intelligent and knowledge based instrumentation - An examination of basic concepts*, Measurement **14** (1), 1994, 23-29.

[16] M.G. Cox, *Graded reference data sets and performance profiles for testing software used in metrology*, In P. Ciarlini, M.G. Cox, F. Pavese, editors, *Advanced Mathematical Tools in Metrology III*, 1997, World Scientific, Singapore, 43-45.

[17] ISO - International Organization for Standardization, *Guide to the Expression of Uncertainty in Measurement*, 1993, Geneva.

[18] ISO - International Organization for Standardization, *ISO/DIS 10360-6, Geometrical Product Specification (GPS) - Acceptance test and reverification test for coordinate measuring machines (CMM). Part 6: Estimation of errors in computing Gaussian associated features*, 1998, proposed draft standard.

[19] R. Drieschner, B. Bittner, R. Elligsen, F. Wäldele, *Testing coordinate measuring machine algorithms. Phase II*, Technical Report BCR EUR 13417 EN, Commission of the European Communities, 1991.

[20] G.T. Anthony, H.M. Anthony, B. Bittner, B.P. Butler, M.G. Cox, R. Drieschner, R. Elligsen, A.B. Forbes, H. Gross, S.A. Hannaby, P.M. Harris, J. Kok, *Reference software for finding Chebyshev best-fit geometric elements*, Precision Engineering, **19**, 1996, 28-36.

[21] M.G. Cox, A.B. Forbes, *Strategies for testing form assessment software*, Technical Report DITC 211/92, National Physical Laboratory, Teddington, UK, 1992.

[22] ISO - International Organization for Standardization, *ISO 10360-2, Coordinate metrology - Part 2: Performance assessment of coordinate measuring machines*, 1994, Geneva.

[23] Kunzmann, H., Trapet, E., Wäldele, F., *Concept for the Traceability of Measurements with Coordinate Measuring Machines*, Proc. 7th IPES, Kobe, Japan, 1993, 41-52.

[24] A. Balsamo, M. Di Ciommo, R. Mugno, B.I. Rebaglia, E. Ricci, R. Grella., *Evaluation of CMM uncertainty through Monte Carlo simulations*, submitted for the Annals of the CIRP, 48/1, 1999.