Propagating uncertainty in instrumentation systems

B. D. Hall
Measurement Standards Laboratory of New Zealand,
Industrial Research Ltd, PO Box 31-310, Lower Hutt, New Zealand.
(email: b.hall@irl.cri.nz)

Abstract

An algorithm to propagate measurement uncertainty information in a modular measurement system is presented. The method is quite general and strictly adheres to current best-practice in the evaluation and reporting of measurement uncertainty. The algorithm can be implemented with little difficulty.

1 Introduction

Instrumentation systems invariably use a single number to represent a measurement result. This is not entirely satisfactory. While any measurement is made with the intention of determining the value of a specific quantity (the 'measurand'), no measurement result should be considered exact; it provides only an *estimate* for the measurand and so a qualifying statement of the uncertainty in this estimate is essential [1, §3.1]. Measurement uncertainty describes the variability in an estimate of a quantity of interest. Together, the measured value and the uncertainty determine a range of values that might reasonably be attributed to a measurand. It follows that measurement uncertainty is important in the design and realization of reliable instrumentation systems.

Modularity is an important engineering concept. A system composed of modular components is flexible, because the configuration can be chosen from among sets of interchangeable parts. In addition, well-designed components generally find a range of applications, resulting in economies of scale through reuse. In instrumentation systems, modularity is underpinned by microprocessor technology and the use of standard communication interfaces between modules. Since these tools became available, some thirty years ago, 'automation' of data acquisition and control has become ubiquitous. Currently, in an effort to further enhance modularity, the Test & Measurement industry is harmonizing industry-wide standards for the communication interfaces of measurement system modules [2–4].

Instrumentation systems rarely report uncertainty explicitly. More often, handbooks and calibration records must be examined and interpreted in the context of a particular measurement. In complex modular systems the relationship between the uncertainty in the measurand and the uncertainty contributed by a particular module may not be obvious. The possibility of changing modules further complicates the situation. In this regard, it is important to distinguish clearly between interchangeability based on measurement function alone, as opposed to function and uncertainty. For instance, a number of different instruments may measure DC voltage over a particular voltage-range, however, the uncertainties in their measurements are unlikely to be comparable. Clearly, when components can be changed in a value-centric system, accuracy is difficult

to predict. This means that laborious testing and re-validation may be necessary to guarantee performance.

While modular systems technology is continually improving, support for handling uncertainty is sorely lacking. Current modular instrumentation systems have no easy way of automatically evaluating measurement uncertainty and this contributes to the burden of testing and validation imposed on performance-critical systems. With modern 'intelligent' instruments, it must be possible to do better. The uncertainty inherent in use of a particular instrument is known to its designers. So, this information could be accessed conveniently in software (essentially the equivalent of what is usually tabulated in instrument manuals), while taking into account the instrument configuration and calibration data. The recent IEEE 1451.1 standard for a 'Smart Transducer Interface' provides for uncertainty reporting [7], encouraging designers to do this.

This article asserts that instrumentation systems can be designed to evaluate and propagate uncertainty automatically. It presents a simple technique that can be used to obtain a dynamic and self-consistent view of system uncertainty. The technique ensures that when modular components are changed uncertainty calculations automatically reflect the new system configuration. The technique strictly adheres to the recommendations of the 'Guide to the Expression of Uncertainty in Measurement' (*Guide*) [1], published by the ISO, which is widely accepted as describing best-practice in this area (a summary is given in the Appendix.).[1]

The article begins by presenting a simple and general algorithm for propagating uncertainty. Then, Section 3 illustrates the approach with a simple example. Section 4 discusses the method and its implementation.

2 An algorithm for propagating uncertainty

This section describes an algorithm for propagating values of uncertainty between modules in an instrumentation system. In what follows a 'module' refers to a component of a measurement system. This article is principally concerned with the set of physically, or logically, distinct modules that constitute a particular measurement system. However, the notion of a module can also be extended to abstract mathematical expressions and this can simplify software development [9].

A module must have a single output but may have several inputs, or none at all. A module without inputs is referred to as a 'leaf' module or 'leaf' input, for example, transducers are generally instances of leaf modules.[2] A module may be used to perform an intermediate measurement (as part of a more complex measurement procedure) or to manipulate the values of an intermediate result.

In mathematical terms, a system measurement can expressed as a function

$$x_{\text{sys}} = f_{\text{sys}}(x_1, \cdots, x_j) \ , \tag{1}$$

where the parameters $x_1, \cdots, x_j$ represent inputs that each contribute a significant component of uncertainty to the result, x_{sys}. When composed of several modules, a

[1] There is an equivalent ANSI document [5] as well as guidelines prepared by NIST [6].

[2] This alludes to the tree-like structure of interconnected modules in a system.

system measurement function can be decomposed into a set of module functions

$$x_i = f_i(\Lambda_i) \ , \tag{2}$$

where 'i' identifies a module, x_i is the output and Λ_i is the set of inputs.[3] Note that the module subscripts are assigned such that $i > j$, where j is the subscript of any member of the set Λ_i.

For a system consisting of m modules, of which l are leaf modules, the system output, x_m, will be obtained recursively: module m calls its input modules, which call their inputs, and so on, down to the leaf inputs $(x_1, \cdots, x_l)$. The process then reverses, as the modules called return values to their callers. The flow of data in this second phase can be regarded as an algorithm for evaluating x_m. It can be expressed iteratively as

For $i = l + 1, \cdots, m$

$$x_i = f_i(\Lambda_i) \ . \tag{3}$$

Note that each step in (3) is associated with a single module and its immediate inputs. This reflects the interconnection topology of the system. The evaluation of a system measurement result is spread throughout the modules of the system.

Equation (3) is implemented implicitly by design (a system of modules are connected to evaluate the desired result). However, a similar algorithm can be used to evaluate the uncertainty. Following the notation of the *Guide*, $u_j(x_i)$ represents the component of uncertainty in x_i due to uncertainty in x_j. Hence the x_j component of uncertainty in the final result, $u_j(x_m)$, can be obtained iteratively, for any given j, as

For $i = l + 1, \cdots, m$

$$u_j(x_i) = \sum_{x_k \in \Lambda_i} \frac{\partial f_i}{\partial x_k} u_j(x_k) \ . \tag{4}$$

This is an application of the chain rule for differentiation combined with the rules governing propagation of uncertainty (see Appendix).

Equation (4) is the main result of this article. Note that the iteration order is the same as in (3) and that the inputs required on the right-hand side of (4) are drawn from Λ_i, the inputs to the module. Therefore the iteration step in (4) represents a calculation that can be performed by the module. It follows that the evaluation of system measurement uncertainty can be delegated to the modules of a system: each module can evaluate the uncertainty components pertaining to its output value. The next section will show this in more detail.

3 Example

Suppose that a system is being designed to estimate the electrical power from a measurement of the potential difference in a calibrated resistor, R. If x_1 is the measured voltage then the power is given by

$$\frac{x_1^2}{R} \ . \tag{5}$$

[3] Λ_i must be a subset of $\{x_j\}$, where $j = 1, 2, \cdots, i-1$.

Further, the resistor is temperature dependent

$$R = R_0\{1 + \alpha(x_2 - T_0)\} \,, \tag{6}$$

where x_2 is the measured temperature, α is the temperature coefficient and R_0 is the resistance at temperature T_0 (α, R_0 and T_0 have negligible uncertainty here).

The measurement system is realized in three modules: a voltage sensor, temperature sensor, and a processing module (e.g., a computer), with which the sensors both communicate (Fig. 1). The sensor modules are regarded as 'black boxes' – their inner workings are hidden. However, both provide an 'uncertainty' function in their communications interface, so that values for $u_j(x_1)$ and $u_j(x_2)$ can be obtained by the processor module. It is assumed that the sensor measurements are independent, i.e. that $u_2(x_1) = u_1(x_2) = 0$.[4] The discussion focuses on the processor module design.

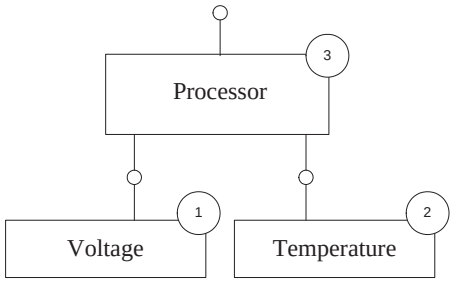

Figure 1: Two sensor modules are connected to a processor module to realize the power measurement system. The small empty circles represent a common communications interface, the larger labeled circles identify the modules. Modules 1 and 2 are 'leaf' modules.

The processor module's (measurement) function is, from (5) and (6),

$$x_3 = f_3(x_1, x_2) = \frac{x_1^2}{[R_0\{1 + \alpha(x_2 - T_0)\}]} \,. \tag{7}$$

The uncertainty component function represents the $i = 3$ iteration of (4), i.e.:

$$u_j(x_3) = \sum_{x_k \in \Lambda_3} \frac{\partial f_3}{\partial x_k} u_j(x_k) \,. \tag{8}$$

Now, $\Lambda_3 = \{x_1, x_2\}$ so (8) can be written explicitly for the cases of $j = 1$ and $j = 2$ as

$$u_1(x_3) = \frac{\partial f_3}{\partial x_1} u_1(x_1) + \frac{\partial f_3}{\partial x_2} u_1(x_2) \tag{9}$$

$$u_2(x_3) = \frac{\partial f_3}{\partial x_1} u_2(x_1) + \frac{\partial f_3}{\partial x_2} u_2(x_2) \,, \tag{10}$$

[4]The notation here for $u_1(x_1)$ and $u_2(x_2)$ is synonymous with $u(x_1)$ and $u(x_2)$ respectively. It is instructive to leave an explicit subscript on u because this corresponds to a parameter that could be passed when calling the uncertainty software function.

and the partial derivatives are, from (7),

$$\frac{\partial f_3}{\partial x_1} = \frac{2x_1}{[R_0\{1+\alpha(x_2-T_0)\}]} \tag{11}$$

$$\frac{\partial f_3}{\partial x_2} = \frac{-R_0\alpha\, x_1^2}{[R_0\{1+\alpha(x_2-T_0)\}]^2}\,. \tag{12}$$

Equations (9) through (12) specify the requirements for an uncertainty interface function for module 3. The function should take a parameter representing the value of j, and return the desired uncertainty component.

In this example, it is tempting to exploit the independence of the two leaf inputs, i.e. that $u_1(x_2) = u_2(x_1) = 0$, so the only contribution to the sum when $j = 1$ is $x_k = x_1$ and when $j = 2$ is $x_k = x_2$. However, this has been avoided in describing the solution above to make the point that it would limit the 'modularity' of the processor module. Suppose that x_1 and x_2 are later exchanged for other devices that *do* have explicit dependencies on other quantities. For instance, the new sensors may be sensitive to ambient temperature, which would introduce another leaf input to the system (ambient temperature) and alter the connection topology of the modules (temperature will be an input shared by the new modules 1 and 2). An implementation of the more general form of (9) through (12) allows, in principle, the system to function correctly in spite of the substantial change to the overall uncertainty calculation.

4 Discussion

The evaluation of uncertainty according to equation (4) is *distributed* among the modules of a system. Each module evaluates the uncertainty components of its output by using information available from its inputs to perform an iteration of (4). This type of calculation is made possible by nature of the equations recommended in the *Guide* and is ideally suited for complex modular systems. The reader should note, however, that there are several important assumptions underlying the *Guide's* approach to propagating and interpreting measurement uncertainties. First, it is assumed that the uncertainty associated with a *measurand* can be approximated by a Gaussian distribution (or Student's t).[5] Second, it is assumed that the measurement function can be approximated by a Taylor series truncated beyond linear terms: in other words, that the measurement function in the vicinity of the measurement point can be considered linear on the scale of variations associated with the uncertainties. If these approximations do not hold then the proposed method may not apply.

No mention has been made about the calculation of 'combined standard uncertainty' (Appendix (16) or (17)), or about evaluating the degrees-of-freedom, which parameterize the uncertainty in an estimate of standard uncertainty (Appendix (18)). These calculations must iterate over the set of leaf-inputs to a system to evaluate (16)-(18), which requires a 'system-wide' view, rather than the single-module one. Nevertheless, implementation is not difficult in practice [10,11] and does not compromise the flexibility and modularity claimed here.

Implementation issues have not been discussed because they depend on the choice of

[5]The *Guide* does allow input distributions to be of different types.

software technology and on the performance and functionality required.[6] In broad terms, each module performs a single iteration of (4), during which calls to that module's inputs obtain values for $u_j(x_k)$. These inputs, identified by the subscripts k, are 'known', they were used implicitly in (3), so only one parameter is required in each uncertainty function call, corresponding to j. In practice, this parameter is a unique software label of some sort assigned to each module.

The technique can be applied to complex systems using existing instrument technology (i.e. no built-in uncertainty function). In this case, the systems designer will write software to encapsulate instruments and present a suitable module interface to the other system components. In other words, the system designer will write the uncertainty function, based on information provided by the manufacturer. This kind of designer-provided software is similar to the idea of a 'Role Control Module' (RCM) for instruments as discussed in [4]. An RCM defines generic instrument functions required of an instrument by a particular system. In this way the role of an instrument is quite narrowly defined within the context of the system. The task of writing an uncertainty function therefore applies only to the particular modes of operation required by the RCM.

As mentioned in Section 2, the development of software supporting this technique can be greatly simplified by extending the concept of a module to abstract elements of an instrument measurement function. By applying techniques of object-oriented software engineering, the need for an explicit expression of the module function derivatives can be eliminated entirely; in the example there would be no need to derive equations (11) and (12). Instead, a set of software objects representing the module function can be assembled that evaluate *both* function values and uncertainties. This technique is described in more detail in [8] and applied to the same example of Section 3 in [9].

For a measurement result to be considered reliable, the measurement procedure should be demonstrably 'traceable' to primary standards and the uncertainties quantified [12]. Traceable measurements are increasingly demanded, because they carry an assurance of quality and reliability. The analysis of uncertainty contributions in a particular measurement procedure is a task requiring specialist skills. There are various considerations that apply when assessing uncertainties in measurement, which are covered in the *Guide* [1] (and [5,6]). This paper simply assumes that systems and instrument designers will have this competency. It should be clear, however, that by adhering to common guidelines and dealing with uncertainty calculations at module-level, rather than a system-level, the system uncertainty calculations will be simplified. It will also be easier to test and validate systems, because modules can be treated independently. So, in general terms, the techniques described here will help to design and maintain complex measurement systems.

5 Conclusions

By applying a simple algorithm, modular measurement systems can propagate uncertainty information when it is supplied by their component parts. Such a system system must comprise modules that report uncertainty using a communications interface software function. Modules calculate the uncertainty in their own outputs according to

[6] Some examples are available, in C++ and Basic [9–11].

the algorithm, which gives rise to propagation of the uncertainty. In this way, calculation of uncertainty is distributed among the modules of a system. Such a design is capable of reporting self-consistent uncertainty values, which recognize the individual contributions of each system module. The design allows modules to be exchanged (plug-and-play) without compromising system integrity. It also allows modules to vary uncertainty values according to the operating point of the measurement.

The method strictly follows current best-practice in evaluation and reporting of measurement uncertainty. It appears that the proposed algorithm is novel and that its application to measurement systems is new. The technique is straightforward to implement, requiring little more programming than is currently required in measurement systems.

Acknowledgments

The author is very grateful to R. Willink for many fruitful discussions regarding this work and for suggestions that have improved this manuscript.

References

[1] IS0 1993 *Guide to the Expression of Uncertainty in Measurement* (Geneva: International Organization for Standardization)

[2] Lee K B and Schneeman R D 1999 Internet-based distributed measurement and control applications *IEEE Instrum. Meas. Mag.* **June** 23

[3] Licht T R 2001 The IEEE 1451.4 proposed standard *IEEE Instrum. Meas. Mag.* **March** 12

[4] Mueller J and Oblad R 2000 Architecture drives test system standards *IEEE Spectrum* **September** 68

[5] ANSI/NCSL Z540-2-1997, *American National Standard for Expressing Uncertainty – U.S. Guide to the Expression of Uncertainty in Measurement*

[6] Taylor B N and Kuyatt C E *Guidelines for Evaluating and Expressing the Uncertainty of NIST Measurement Results* (NIST Technical Note 1297)

[7] IEEE Std 1451.1-1999 'IEEE Standard for a Smart Transducer Interface for Sensors and Actuators – Network Capable Application Processor Information model' (IEEE, New York, 2000)

[8] Hall B D 2002 Calculating measurement uncertainty using Automatic Differentiation *Meas. Sci. Technol.* submitted

[9] Hall B D 2001 Automatic uncertainty calculation for smart measurement systems *Proc. Sensors for Industry Conf. (Rosemount, Ill.)* (Nov. 2001) 290

[10] Hall B D 2001 Calculating measurement uncertainty with reverse automatic differentiation *Industrial Research Report* **1041**

[11] Perera S, Hall B D 2002 Measurement uncertainty software tools for PCs *Industrial Research Report* (in preparation)

[12] J. V. Nicholas and D. R. White "Traceable temperatures" 2nd ed. (John Wiley & Sons, Chichester, 1995)

Appendix

A Standard practice

Guidelines for the evaluation of measurement uncertainties have been published in the *Guide to the Expression of Uncertainty in Measurement* [1]. The recommendations of *Guide* are accepted as current best-practice and have been adopted by national measurement institutes and accredited measurement laboratories world-wide.

The measurement of a quantity X_m, the measurand[7], is described by a function $X_m = f(X_1, X_2, \ldots, X_l)$, interpreted as "*...that function which contains every quantity, including all corrections and correction factors, that can contribute a significant component of uncertainty to the measurement result*" [1, 4.1.2]. The quantities $X_1, X_2, \ldots, X_l$ and X_m are random variables, however, the parameters describing their distribution (mean and standard deviation) are not known exactly. Estimates of the means, $x_1, x_2, \ldots, x_l$, must be used to estimate the measurand as

$$x_m = f(x_1, x_2, \ldots, x_l) \; . \tag{13}$$

A 'standard uncertainty', $u(x_i)$, is associated with each x_i and is understood to be an estimate of the standard deviation of X_i. The quality of this estimate can be characterized by a number of 'degrees-of-freedom', ν_i. If ν_i is infinite the estimate is considered exact, otherwise ν_i is related to the relative uncertainty of $u(x_i)$ [1, G.4.2]. The degrees-of-freedom has its conventional meaning when associated with a normally distributed random variable.

The uncertainty in x_m depends on the uncertainties in $x_1, x_2, \ldots, x_l$ and on the form of f. The *Guide* denotes the components of the uncertainty in x_m, due to individual input quantities, as

$$u_i(x_m) \equiv \frac{\partial f}{\partial x_i}\, u(x_i) \; , \tag{14}$$

where the partial derivative is more formally expressed as

$$\frac{\partial f}{\partial x_i} \equiv \left.\frac{\partial f}{\partial X_i}\right|_{x_1, x_2, \ldots, x_l} \; . \tag{15}$$

The combined standard uncertainty in x_m is the root-sum-square of these components[8]

$$\begin{aligned} u_{\rm c}(x_m) &= \left[\sum_{i=1}^{l} u_i^2(x_m)\right]^{1/2} \\ &= \left[\sum_{i=1}^{l} \left(\frac{\partial f}{\partial x_i}\right)^2 u^2(x_i)\right]^{1/2} \; , \end{aligned} \tag{16}$$

[7] In the *Guide*, 'Y' denotes the measurand (and y its estimate).

[8] The notation for combined uncertainty used by the *Guide* is $u_{\rm c}(x_i)$. In the body of the paper the subscript 'c' is omitted when the meaning is clear from the context (e.g. $u(x_i)$).

provided input quantities are uncorrelated. With correlation the combined standard uncertainty should be evaluated as

$$u_{\rm c}(x_m) \quad = \quad \left[\sum_{i=1}^{n} \sum_{j=1}^{l} \frac{\partial f}{\partial x_i} \frac{\partial f}{\partial x_j} u(x_i) u(x_j) r(x_i, x_j) \right]^{1/2} \tag{17}$$

where $r(x_i, x_j) \equiv u(x_i, x_j)/u(x_i)\, u(x_j)$ is the correlation coefficient and $u(x_i, x_j)$ is the estimated covariance of X_i and X_j.

If one or more of the input-quantity uncertainties has finite degrees of freedom, a value for the 'effective degrees-of-freedom', $\nu_{\rm eff}$, of $u(y)$ should be calculated using the Welch-Satterthwaite formula [1, G.4]

$$\frac{u_c^4(x_m)}{\nu_{\rm eff}} = \sum_{i=1}^{l} \frac{u_i^4(y)}{\nu_i} \; . \tag{18}$$

The Welch-Satterthwaite formula does not apply when correlations are present – the *Guide* makes no specific recommendations in this case.