Choosing Designs of Calibration Transducer Electronic Data Sheets

Lee Barford, Qiang Li*
Integrated Solutions Laboratory
HPL-98-166
September, 1998

E-mail: barford@hpl.hp.com
li@stat.yale.edu

smart sensors,
calibration,
correction,
spline,
uncertainty,
design optimization

The IEEE 1451.2 standard for interfaces to networkable smart sensors and actuators introduced the notion of a calibration Transducer Electronic Data Sheet (TEDS), which contains the information needed by software for mapping between transducer-side values and physical unit values. When a calibration TEDS entry is created for a transducer, a calibration model (the degree of the multinomials and the placement of segment boundaries) must be chosen. Calibration TEDS must fit into small memories, and hardware floating point support is typically unavailable for correction function evaluation, but uncertainty due to error fitting the correction function must be minimized. This paper considers the computing of TEDS memory consumption, the number of floating point computations required for each correction, and the uncertainty when the correction function is a spline. It also describes a prototype tool that makes these computations and graphs them in forms intended to be of use in making calibration model design tradeoffs. The tool also contains a facility for automatic selection of "best" designs.

*Department of Statistics, Yale University, New Haven, Connecticut

Choosing Designs of Calibration Transducer Electronic Data Sheets

Lee Barford* and Qiang Li†

* Hewlett-Packard Laboratories, 1501 Page Mill Rd. MS 4A-D, Palo Alto, CA 94303
Phone: +1-650-857-3606, Fax: +1-650-852-8092, E-mail: barford@hpl.hp.com

† Department of Statistics, Yale University, P.O. Box 208290, New Haven, CT 06520
Phone: +1-203-432-0640, E-mail: li@stat.yale.edu

Abstract *– The IEEE 1451.2 standard for interfaces to networkable smart sensors and actuators introduced the notion of a calibration Transducer Electronic Data Sheet (TEDS), which contains the information needed by software for mapping between transducer-side values and physical unit values. When a calibration TEDS entry is created for a transducer, a calibration model (the degree of the multinomials and the placement of segment boundaries) must be chosen. Calibration TEDS must fit into small memories, and hardware floating point support is typically unavailable for correction function evaluation, but uncertainty due to error fitting the correction function must be minimized. This paper considers the computing of TEDS memory consumption, the number of floating point computations required for each correction, and the uncertainty when the correction function is a spline. It also describes a prototype tool that makes these computations and graphs them in forms intended to be of use in making calibration model design tradeoffs. The tool also contains a facility for automatic selection of "best" designs.*

I. INTRODUCTION

The recently approved IEEE 1451.2 standard for interfaces to networkable smart sensors and actuators [1] [2] [3] introduced the notion of a *Transducer Electronic Data Sheet* (TEDS). A TEDS is a persistent memory that specifies one or more transducers (sensors, actuators, or digital input or output devices) and their triggering, signal conditioning, and signal conversion. The TEDS also describes the physical units measured by each sensor and controlled by each actuator, usually represented as a product of powers of SI base units [4]. (Unit-less and digital data are also permitted.) The TEDS is part of a *Smart Transducer Interface Module* (STIM), which also contains analog-to-digital (A/D) or digital-to-analog (D/A) converters and triggering mechanisms. A TEDS must remain physically associated with the unique set of transducers it describes through its normal operating life.

The TEDS is read, written, and used by a *Network Capable Application Processor* (NCAP). The NCAP's primary purpose is to mediate between the STIM (whose interface is computer network independent) and a particular network. An NCAP can also perform some processing of values going to or coming from the transducers (e.g., computing calibration corrections and converting between values in metric units and values coming from the D/As or going to the A/Ds). Figure 1 shows an accelerometer and its STIM and an NCAP for interfacing a STIM to 10BaseT Ethernet. Figure 2 illustrates the relationships between the TEDS, transducers, NCAP, and network.

It is desirable that the STIM and NCAP add little size or cost to the transducer(s) they describe and interface. Single-chip STIMs have been built [5]. TEDS memories are often small, for example one or two kilobytes. These size and cost considerations also restrict the computing power available in the NCAP. Hardware floating point units, for example, will often be unavailable, making floating point computation slow.

One optional component of a TEDS is the *calibration TEDS*. The calibration TEDS contains all of the information needed by *correction software*, software for mapping between *transducer-side* values (A/D or D/A values received from or to be applied to one transducer) and physical unit values represented in floating point. In other words, correction software simultaneously performs calibration correction and conversion to or from physical units. For example, a calibration TEDS can specify how to convert a transducer-side value from a pressure sensor and an already-corrected reading from a thermocouple into a temperature-compensated pressure in pascals [6].

A calibration TEDS entry for one transducer represents the required functional relationship as a piecewise-multinomial function. The range of each of the function's input variables is divided into one or more *segments*. Each of the resulting cuboid-shaped *cells* is associated with a multinomial

$$\sum_{i=0}^{D(1)}\sum_{j=0}^{D(2)}\cdots\sum_{p=0}^{D(n)} C_{i,j,\cdots,p}[X_1 - H_1]^i[X_2 - H_2]^j\cdots[X_n - H_n]^p$$

(see Figure 3), where the X_k variables represent the data from a set of inputs (either transducer-side or corrected), and integer dimensions *D(k)* and floating point coefficients $C_{i,j,\ldots,p}$

and offsets H_k are recorded in the calibration TEDS. When a calibration TEDS entry is created, values must be chosen for the following parameters: the number of segments for each input, the degree of the multinomials, and placement of segment boundaries. We call the choice of these parameters *calibration model design.*

The manufacturer of the transducer and STIM is faced with a seemingly simple question: which calibration model is best? Extreme values of the parameters provide simple designs. But each of these simple designs has a difficulty. For example, the segments can be chosen to be equal to the discretization levels of transducer-side values. The result is a lookup table. Both floating point computation time for each correction and approximation error are minimized, but a great deal of STIM memory is consumed. Another naïve answer is to take the most accurate fit. But this approach is likely to be expensive in TEDS memory and in floating point computation, and the multinomials are likely to be fitting noise in the calibration measurements. Another simple choice is to have one large cell and provide one high-degree multinomial correction function. However, evaluation of high-degree polynomials and multinomials leads to notorious numerical accuracy problems, particularly with the single-precision coefficients specified by IEEE 1451.2. High-degree multinomials are not a good choice unless, as for some medical equipment, they are required by regulation.

Transducer manufacturers have another consideration. The same transducer design can be sold to customers with differing needs (e.g., cheap, fast, less accurate vs. expensive, slow, more accurate) through a combination of

- Determining the performance of individual transducers
- Choosing values for various TEDS fields, including choosing an appropriate calibration model design for each market segment

Therefore, in order to make a good calibration model design decision, one must be able take a set of measured calibration data for a transducer and candidate values for the calibration model design parameters and compute

1. Required TEDS memory
2. Number of floating point operations required for each correction (which may be important because hardware support for floating point is typically not available in the NCAP)
3. Uncertainty in the output values due to noise in the calibration measurements and from fitting the correction function

These numbers can be compared across the different values of the calibration model design parameters and an appropriate calibration model design can be chosen. Reinhart [7] describes a calibration TEDS creation tool that provides information about the third of these, through graphing fitting residuals and estimating the residuals' standard deviation.

Formulas for computing the calibration TEDS memory requirements are given explicitly in the standard [1]. A formula for the number of floating point operations per correction can be found by examination of the multinomial evaluation code.

This paper shows how to make the uncertainty computation when the piecewise-multinomial is a regression spline. It also discusses a prototype tool that provides graphical aids for selecting an appropriate calibration model design. The tool also has an automatic design mode, where a "best" design is chosen given lower and upper bounds on the design parameters and functions representing constraints on and relative preferences between the memory, floating point computation, and uncertainty requirements.

II. FITTING AND UNCERTAINTY ANALYSIS

This section deals with the following two problems: given a set of calibration data for a transducer, (a set of transducer side values obtained through appropriate measurements) and a calibration model design (multinomial order, selection of segment boundaries)

1. Fit a correction function, i.e., find an appropriate set of multinomial coefficients $C_{i,j,\dots,p}$ for each cell
2. Estimate the uncertainty induced by stochastic measurement error and by fitting, expressed [8] as standard deviation or confidence intervals

Although not required by the standard [1], the correction function should be continuous and smooth even at cell boundaries. One way to achieve this is to fit using splines [9] (one-dimensional case) or tensor-product splines (multi-dimensional case). For ease of presentation, the rest of this section will deal with the one-dimensional case. The extension to tensor-product splines is straightforward. Previous uses of splines to describe correction functions for particular sensors or actuators include [10] [11] [12]. (The spline literature uses the term "knot" to describe a generalization of the notion of segment boundary. We will continue to use the term "segment boundary" in this paper.)

Splines are the subject of a large and beautiful theory. However, we use only a few of their properties, from [9] chap. IX:

- A spline is a piecewise polynomial function. Each piece has the same degree *d*.
- The set of degree *d* splines that are everywhere *d*-1 times continuously differentiable and that have a given set of segment boundaries forms a finite-dimensional vector space. There is a simple, fast algorithm for computing a basis $\{B_i(x) \mid i = 1,\dots,m\}$ of the vector space.
- There exists another simple, fast algorithm to compute the coefficients $C_{i,j,\dots,p}$ given a point in that vector space.

Assume without loss of generality that a calibration correction function is to be fitted for a sensor, rather than an actuator. The physical values Y are to be functionally related to transducer-side values X, but the functional relationship is contaminated with noise:

$$Y = f(X) + \varepsilon = B(X)^T \beta + \varepsilon = \sum_{i=1}^{m} \beta_i B_i(X) + \varepsilon$$

where for any scalar w, $b(w)$ is the vector-valued function $b(w) = [B_1(w), ..., B_m(w)]^T$. ε is a random variable that represents all the information about the noise in the calibration experiment. We are to compute an estimate $\hat{f}(x)$, i.e., estimate $\hat{\beta} = [\hat{\beta}_1, \dots, \hat{\beta}_m]^T$, from measured sensor-side values $\tilde{X} = [x_1, ..., x_n]^T$ and measured transducer-side values $\tilde{Y} = [y_1, ..., y_n]^T$. Let the design matrix $\mathbf{B}$ be given by:

$$\mathbf{B} = \begin{bmatrix} B_1(x_1) & B_2(x_1) & \cdots & B_m(x_1) \\ B_1(x_2) & B_2(x_2) & \cdots & B_m(x_2) \\ \vdots & \vdots & \ddots & \vdots \\ B_1(x_n) & B_2(x_n) & \cdots & B_m(x_n) \end{bmatrix}.$$

The normal equations give the least-squares estimate $\hat{\beta} = (\mathbf{B}^T\mathbf{B})^{-1}\mathbf{B}^T\tilde{Y}$. The variance of the estimate $\hat{\beta}$ is $\mathrm{var}(\hat{\beta}) = (\mathbf{B}^T\mathbf{B})^{-1}\mathbf{B}^T \,\mathrm{var}(Y)((\mathbf{B}^T\mathbf{B})^{-1}\mathbf{B}^T)^T$. If we assume that ε is independently, identically distributed (i.i.d.) with variance σ^2, we have a simplified variance expression $\mathrm{var}(\hat{\beta}) = \sigma^2 (\mathbf{B}^T\mathbf{B})^{-1}$. The fitted correction function is the prediction of Y at a point z

$$Y(z) = \sum_{i=1}^{M} B_i(z)\hat{\beta}_i .$$

$Y(z)$ is a random variable with variance

$$\begin{aligned}\mathrm{var}(Y(z)) &= b^T(z)\,\mathrm{var}(\hat{\beta})b(z) + \sigma^2 \\ &= \sigma^2 b^T(z)(\mathbf{B}^T\mathbf{B})^{-1}b(z) + \sigma^2 .\end{aligned}$$

The uncertainty arising from the fitting process[1] can be stated as the maximum standard deviation

$$\sigma_{\max} = \sup_z \{\sqrt{\mathrm{var}(Y(z))}\} \qquad (1)$$

where z ranges across the usable range of the transducer. The variance of the correction function, $\mathrm{var}(Y(z))$, is a smooth function of z. However, it has many local maxima, making problematic the employment of iterative numeric methods to perform the maximization in (1). In practice, $\sigma_{\max}$ can be approximated by computing $\mathrm{var}(Y(z))$ on a finite grid of values for z, for example those corresponding to the discretization levels of the A/D or D/A connected to the transducer.

Finally, assume that ε is normally distributed in addition to i.i.d. Then the p-percent confidence interval at a point z is

$$Y(z) \pm t^{-1}(p/200, df)\sqrt{\mathrm{var}(Y(z))} , \qquad (2)$$

where t^{-1} is the inverse of the cumulative distribution function of Student's T distribution with df = size(data)-size(parameters) = n-m degrees of freedom.

III. CALIBRATION MODEL DESIGN TOOL

We have developed a prototype tool to assist in choosing a calibration TEDS model design. The tool is able to read a file containing calibration measurements $\tilde{X}$ and $\tilde{Y}$. Currently, only one- and two-dimensional inputs are allowed. That is, the tool can be used to generate piecewise polynomials and binomials. The current implementation uses segments of uniform length. It provides both graphical aids and an automatic optimization aid for choosing a calibration model design. Once a calibration model design is chosen, the tool is capable of writing out a text file containing the entries to insert into the calibration TEDS.

The graphical aids available are illustrated by example graphs obtained from a single, typical set of calibration measurements, from a gas flow rate sensor.

One set of graphical aids shows the TEDS memory requirement, floating point operations requirement, and $\sigma_{\max}$ (or any two of them) for a set of possible choices of the model design parameters. The user chooses lower and upper bounds on the polynomial degree. Points representing the possible design parameter choices are plotted on axes of memory and floating point requirement and $\sigma_{\max}$. Each point has a numeric label. Accompanying textual output maps each numeric label to a particular selection of design parameter values. This plot is useful if one design parameter is constrained and another is to be optimized.

For example, it might be desired to find the design that yields the lowest $\sigma_{\max}$ when only a certain number of bytes is available in which the store the calibration TEDS. An example of a plot that can be used to solve this problem is shown in Figure 4. Suppose that 256 bytes are available for the calibration TEDS. Then design 75 has the lowest $\sigma_{\max}$ among the designs requiring less than 256 bytes. The $\sigma_{\max}$ axis is logarithmic rather than linear for two reasons:

- to provide easier visual comparison of two designs that both have their respective $\sigma_{\max}$ close to the minimum, and
- to provide easier visual detection of over-fitting (when $\sigma_{\max}$ ceases to decrease monotonically with increasing

[1] $\sigma_{\max}$ is not the total uncertainty in measurements taken with the sensor, because it does not account for systematic errors in the calibration measurements.

degree or number of segments, noise in the measurements is probably being fitted).

The tool also provides plots of the number of segments vs. memory, floating point operation requirement, or $\sigma_{\max}$. This may be useful when the polynomial degree is known from some *a priori* consideration.

In addition to the graphical aids, the prototype tool provides a simple yet powerful design optimization facility that chooses a “best” design based on cost functions (functions on memory, on floating point operations, and on $\sigma_{\max}$). The design optimization facility is similar to that described in [13], used in the design of printed circuit boards. Let *cm(x)* be the cost of *x* bytes calibration TEDS memory. Let *cf(x)* be the cost of requiring *x* floating point operations per conversion. Let *ce(x)* be the cost of having $\sigma_{\max} = x$. The design optimization facility minimizes the total cost function *cm(x)+cf(x)+ce(x)*.

In addition to indicating preference, the cost functions can also express hard constraints through floating point infinities, now available on virtually all computers and virtually all software floating point packages. For example, if calibration TEDS memory is limited to *b* bytes, then set

$$cm(x) = \left\{ \begin{matrix} 0 \text{ if } x \le b \\ \infty \text{ if } x > b \end{matrix} \right\}. \tag{3}$$

If any of memory, floating point operations, or $\sigma_{\max}$ does not matter in a particular situation, then the corresponding cost function can be set to equal zero everywhere.

In the current implementation, the cost functions are arbitrary bodies of code whose names are passed to the optimization routine. The optimization routine makes no assumptions about the cost functions. For example, (3) shows that discontinuous and infinite-valued cost functions are useful. Therefore, all possible designs are evaluated to find the one with minimum cost. At first glance, it might seem that evaluating all possible designs would require a prohibitive amount of computer time. However, this is not the case, because the number of polynomial orders that must be considered is small. The polynomial order should be limited to at most three or perhaps four because of inaccuracy in evaluating high-degree polynomials. This limitation is especially important with the single-precision coefficients specified by 1451.2. Furthermore, the number of segments must be less than the number of measurements. Therefore, the number of possible designs grows only linearly with the number of measurements.

Note that if the cost functions were restricted to being monotone non-decreasing then a branch and bound algorithm could be used to lessen the number of designs that need to be evaluated in order to find the optimal one.

Once a calibration model design is chosen, whether graphically or using the design optimization facility, it should be inspected for goodness of fit. The tool provides a group of plots, shown simultaneously, for this purpose (Figure 5). One plot shows the measured data and fitted correction function. The second plot shows the residual (difference between measured values and predictions from the correction function) and the 95% confidence interval computed using equation (2). Since the computation of the confidence interval assumes that the residual is normally distributed, the third plot is provided. It is a diagnostic plot for determining whether or not residual is approximately normally distributed. In this third plot, if the residual is approximately normally distributed then the points representing the residual will be approximately arranged straight-line [14].

IV. CONCLUSIONS

The IEEE 1451.2 calibration TEDS offers transducer manufacturers new opportunities and new challenges. Among the challenges is selecting appropriate calibration models for each sensor or sensor design. The selection is complicated because the resulting correction functions must be described in a limited amount of memory and are typically evaluated on embedded processors without hardware floating point. We have described a tool that provides graphical and design optimization aids for selecting an appropriate calibration model design for a particular transducer and its intended use.

The current implementation of the tool uses cardinal splines, splines with equally-spaced segment boundaries. The tool could be improved through the use of a heuristic (there are many in the spline literature) for placing segment boundaries so that a better fit than that of a cardinal spline is obtained for the same number of segments.

ACKNOWLEDGEMENTS

Figure 1(a) is used with the permission of Electronics Development Corporation. Figures 2 and 3 are from the IEEE 1451.2 standard [1]. The authors wish to thank Stan Woods and John Eidson of the 1451.2 committee for helping to ensure the correctness of our discussion of that standard.

REFERENCES

[1] ---, *A Smart Transducer Interface for Sensors and Actuators -- Transducer to Microprocessor Communication Protocols and Transducer Electronic Data Sheet (TEDS) formats*, IEEE Standard 1451.2, 1998.

[2] Woods, S. P., Kang Lee, Bryzek, J., “An Overview of the IEEE-P1451.2 Smart Transducer Interface Module,” J. Analog Integrated Circuits and Signal Processing, vol. 14, no. 3, 1997.

[3] Lee, K., “The Proposed Smart Transducer Interface Standard,” Proc. IEEE Instrumentation and Measurement Conference, pp. 129-133A, 1998.

[4] Hamilton, B., “A Compact Representation of Physical Units,” Hewlett-Packard Laboratories Technical Report HPL-96-61, Hewlett-Packard Company, Palo Alto, California, 1996.

[5] Cummins, T., Brannick, D., *et al.*, “An IEEE 1451 Standard Transducer Interface Chip,” IEEE International Solid-State Circuits Conference, pp. 276-7, 1998.

[6] Moore, J., “Use of the IEEE 1451.2 Correction Engine to Compensate a Multivariable Smart Pressure Transmitter,” Proc. Sensors Expo, Helmers Publishing, Peterborough, NH, pp. 225-227, 1998.

[7] Reinhart, R. A., “Calibration Visualization and Automation Software for IEEE 1451.2 Compliant Smart Sensors,” Proc. Sensors Expo, Helmers Publishing, Peterborough, NH, pp. 229-234, 1998.

[8] ---, *Guide to the Expression of Uncertainty in Measurement*, ISO, 1993.

[9] de Boor, C., *A Practical Guide to Splines*, Springer Verlag, 1978.

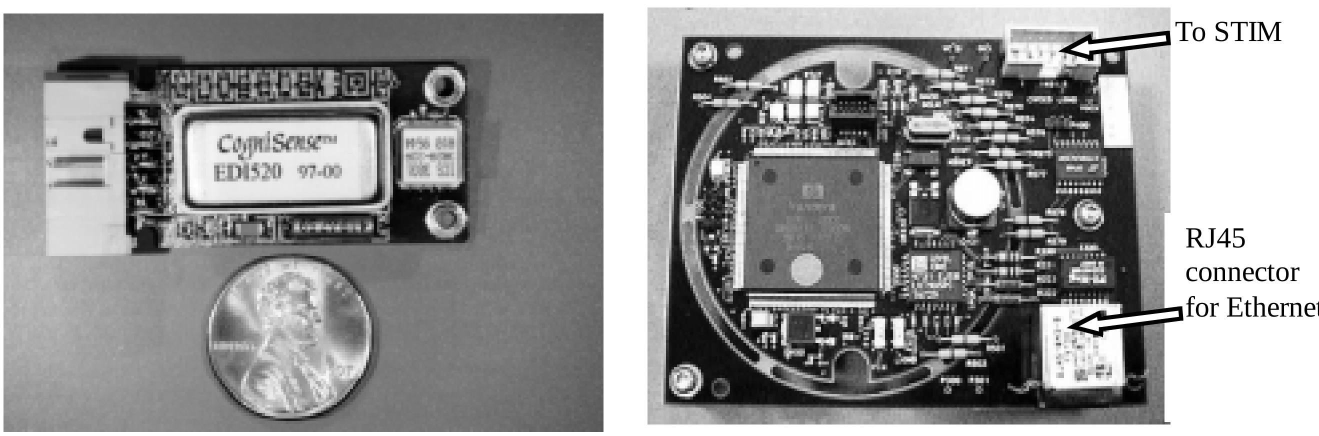
[10] Stahl, K., Thomasson, R., “Using CPS120 (Curved Position-sensitive Detector Covering 120 Degrees) Powder Diffraction Data in Rietveld Analysis,” J. Appl. Crystallography, vol.25, pt. 2, pp. 251-8, 1992.

[11] Camp, W. O., *et al.*, “Voltage Controlled Oscillator with Correction of Tuning Curve Irregularities,” U.S. Patent No. 5179725, 1993.

[12] Kildal-Brandt, P. A., *et al.*, “Normalizing Analyzer Systems to a Standard Analyzer,” U.S. Patent no. 5257212, 1993.

[13] Kuokka, D., Jefferson, S., Barford, L., Frayman, F., “A parametric design assistant for concurrent engineering,” Artificial Intelligence for Engineering Design, Analysis and Manufacturing, vol.9, no.2, pp. 135-44, 1995.

[14] Mandel, J., The Statistical Analysis of Experimental Data, chap. 5.3, Dover, 1984.



(a) **(b)**

Figure 1: A STIM and an NCAP. (a) An accelerometer and its STIM. (b) An NCAP that interfaces a STIM to 10BaseT Ethernet. Horizontal dimension is about 10 cm. The circular part of the board can be cut out, creating an even smaller NCAP.

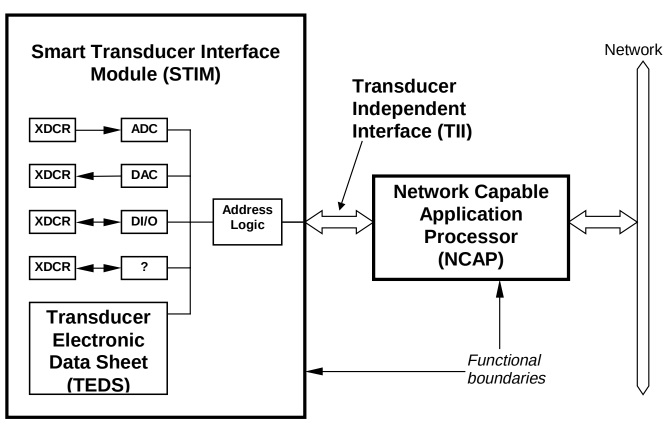


Figure 2: Relationship of transducers, TEDS, NCAP, and network.

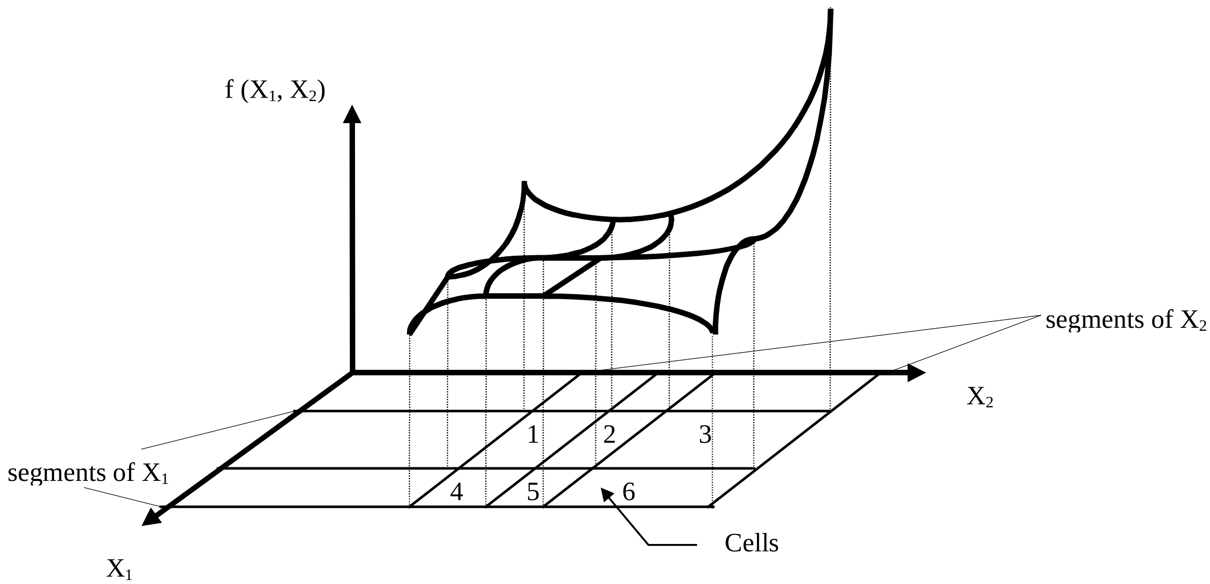


Figure 3: A correction function represented by a piecewise multinomial.

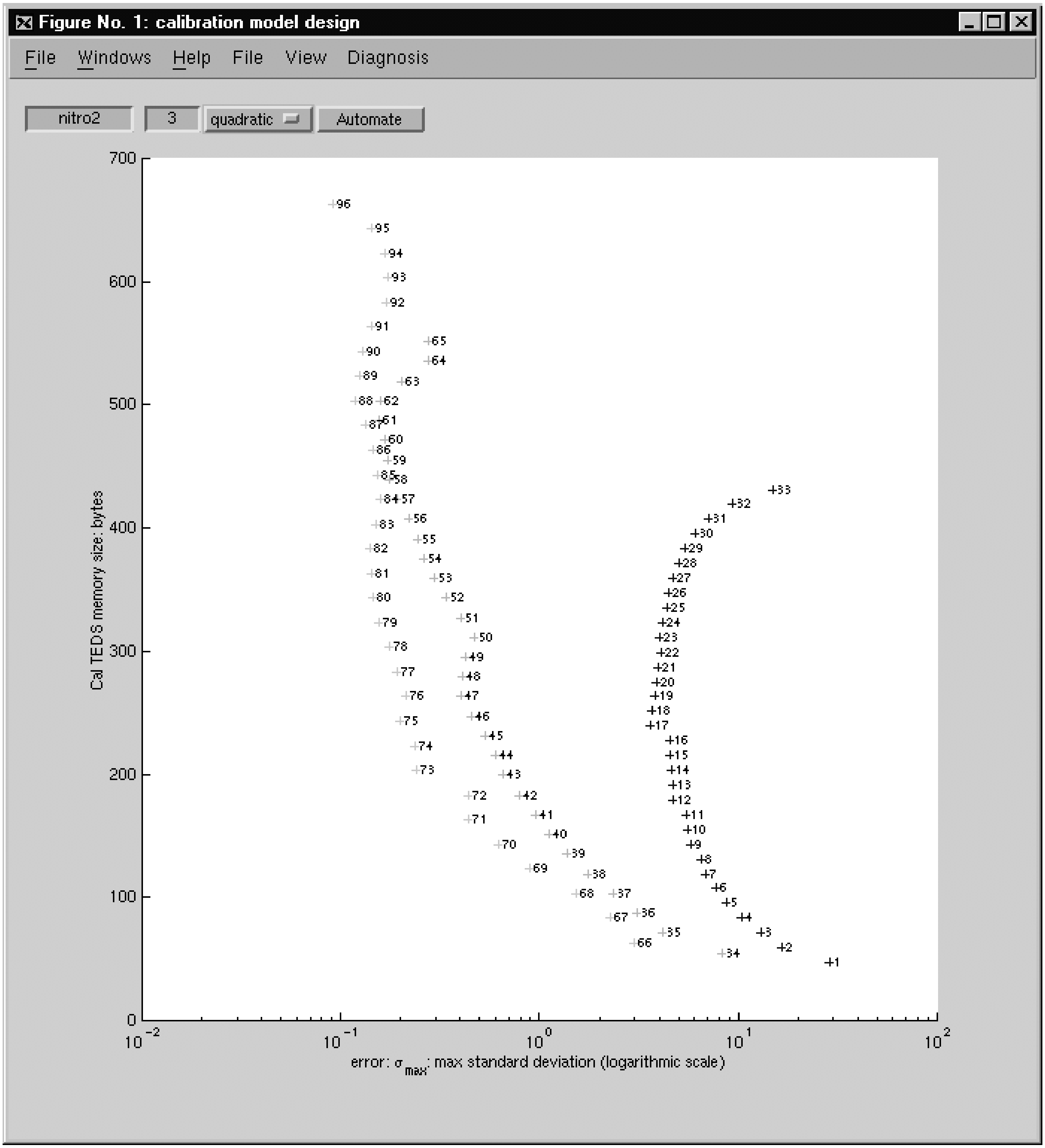


Figure 4: Parametric plot of maximum standard deviation vs. memory requirement for a number of calibration model designs. Numbers labeling points refer to design parameter values given in a separate textual listing. Shown are all possible designs using evenly-spaced segment boundaries and quadratic, linear, or constant splines.

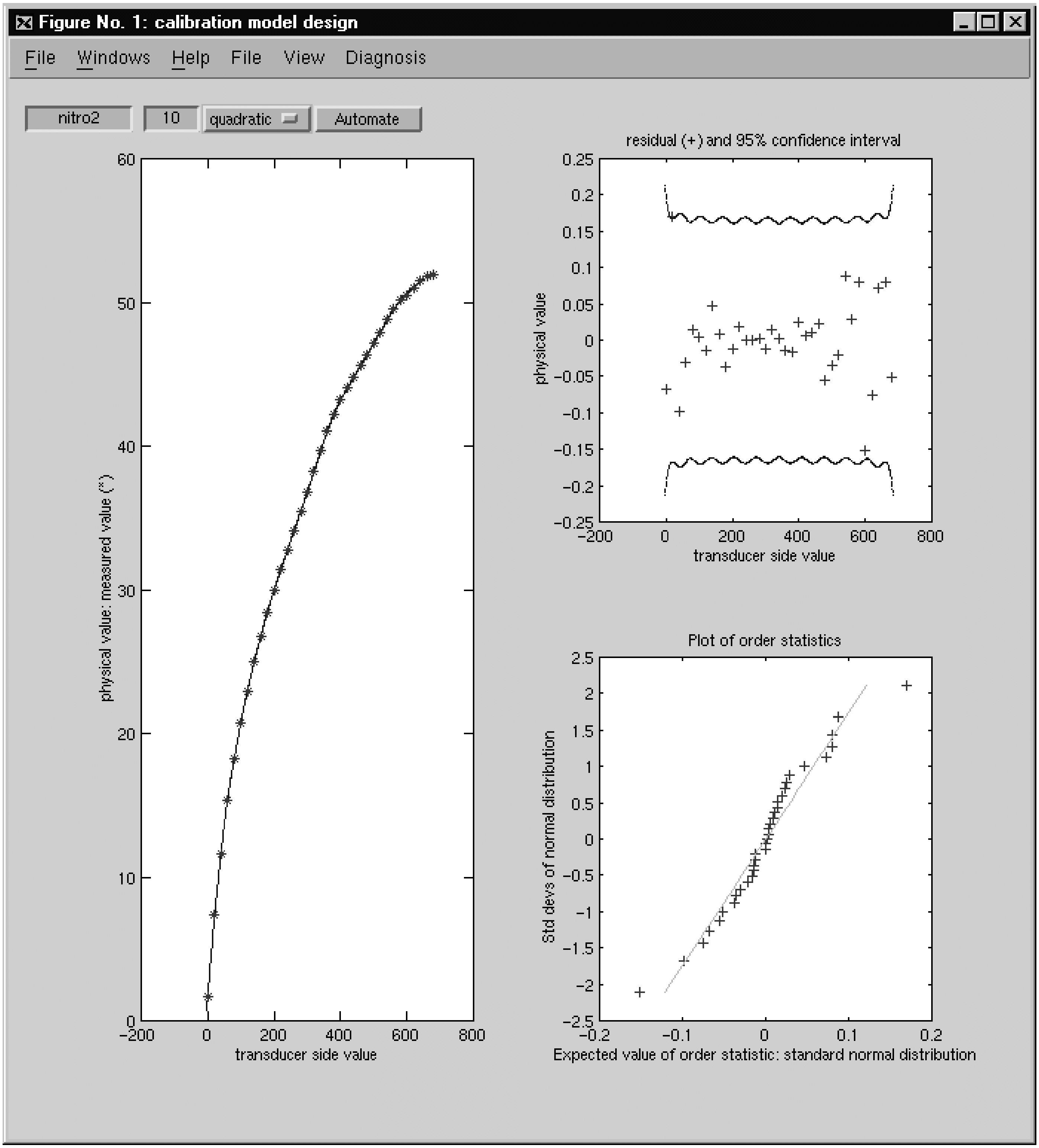


Figure 5: Plot of correction function corresponding to calibration design 75 in Figure 4. Left side: correction function and measurements. Upper right: fitting residual and 95% confidence interval. Lower right: plot for verifying whether residual is normally distributed.