Transfer Functions

2

Everything is controlled by probabilities.
I would like to know—who controls probabilities?

Stanisław Jerzy Lec

Since most of stimuli are not electrical, from its input to the output a sensor may perform several signal conversion steps before it produces and outputs an electrical signal. For example, pressure inflicted on a fiber optic pressure sensor, first, results in strain in the fiber, which, in turn, causes deflection in its refractive index, which, in turn, changes the optical transmission and modulates the photon density, and finally, the photon flux is detected by a photodiode and converted into electric current. Yet, in this chapter we will discuss the overall sensor characteristics, regardless of a physical nature or steps that are required to make signal conversions inside the sensor. Here, we will consider a sensor as a "black box" where we are concerned only with the relationship between its output electrical signal and input stimulus, regardless of what is going on inside. Also, we will discuss in detail the key goal of sensing: determination of the unknown input stimulus from the sensor's electric output. To make that computation we shall find out how the input relates to the output and vice versa?

2.1 Mathematical Models

An ideal or theoretical input–output (stimulus–response) relationship exists for every sensor. If a sensor is ideally designed and fabricated with ideal materials by ideal workers working in an ideal environment using ideal tools, the output of such a sensor would always represent the *true value* of the stimulus. This ideal input–output relationship may be expressed in the form of a table of values, graph, mathematical formula, or as a solution of a mathematical equation. If the input–output function is

J. Fraden, *Handbook of Modern Sensors*, DOI 10.1007/978-3-319-19303-8_2

time invariant (does not change with time) it is commonly called a *static transfer function* or simply *transfer function*. This term is used throughout this book.

A static transfer function represents a relation between the input stimulus s and the electrical signal E produced by the sensor at its output. This relation can be written as $E = f(s)$. Normally, stimulus s is unknown while the output signal E is measured and thus becomes known. The value of E that becomes known during measurement is a number (voltage, current, digital count, etc.) that represents stimulus s. A job of the designer is to make that representation as close as possible to the true value of stimulus s.

In reality, any sensor is attached to a measuring system. One of the functions of the system is to "break the code E" and infer the unknown value of s from the measured value of E. Thus, the measurement system shall employ an inverse transfer function $s = f^{-1}(E) = F(E)$, to obtain (compute) value of the stimulus s. It is usually desirable to determine a transfer function not just of a sensor alone, but rather of a system comprising the sensor and its interface circuit.

Figure 2.1a illustrates the transfer function of a thermo-anemometer—the sensor that measures mass flow of fluid. In general, it can be modeled by a square root function $f(s)$ of the input airflow rate. The output of the sensor can be in volts or in digital count received from the analog-to-digital converter (ADC), as shown on the y-axis of Fig. 2.1a for a 10-bit ADC converter. After the output count $n = f(s)$ is measured, it has to be translated back to the flow rate by use of the inverse transfer function. The monotonic square root function $f(s)$ has parabola $F(n)$ as its inverse. This parabola is shown in Fig. 2.1b, illustrating the relation between the output counts (or volts) and the input flow rate. Graphically, the inverse function can be obtained by a *mirror reflection* with respect to the bisector of the right angle formed by x and y-axes.

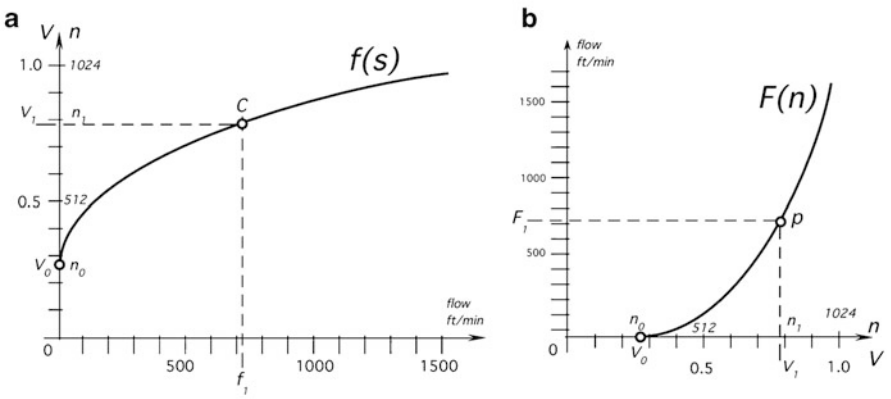

Fig. 2.1 Transfer function (**a**) and inverse transfer function (**b**) of thermo-anemometer

2.1.1 Concept

Preferably, a physical or chemical law that forms a basis for the sensor's operation should be known. If such a law can be expressed in form of a mathematical formula, often it can be used for calculating the sensor's inverse transfer function by inverting the formula and computing the unknown value of s from the measured output E. Consider for example a linear resistive potentiometer that is used for sensing displacement d (stimulus s is this example). The Ohm's law can be applied for computing the transfer function as illustrated in Fig. 8.1. In this case, the electric output E is the measured voltage v while the inverse transfer function is given as

$$d = F(E) = \frac{D}{v_0} v \tag{2.1}$$

where v_0 is the reference voltage and D is the maximum displacement (full scale); both being the constants. By using this function we can compute displacement d from the measured voltage v.

In practice, readily solvable formulas for many transfer functions, especially for complex sensors, does not exist and one has to resort to various approximations of the direct and inverse transfer functions, which are subjects of the following section.

2.1.2 Functional Approximations

Approximation is a selection of a suitable mathematical expression that can fit the experimental data as close as possible. The act of approximation can be seen as a *curve fitting* of the experimentally observed values into the approximating function. The approximating function should be simple enough for ease of computation and inversion and other mathematical treatments, for example, for computing a derivative to find the sensor's sensitivity. The selection of such a function requires some mathematical experience. There is no clean-cut method for selecting the most appropriate function to fit experimental data—eyeballing and past experience perhaps is the only practical way to find the best fit. Initially, one should check if one of the basic functions can fit the data and if not, then resort to a more general curve-fitting technique, such as a polynomial approximation, e.g., as described below. Here are some most popular functions used for approximations of transfer functions.

The simplest model of a transfer function is *linear*. It is described by the following equation:

$$E = A + Bs. \tag{2.2}$$

As shown in Fig. 2.2, it is represented by a straight line with the intercept A, which is the output signal E at zero input signal $s = 0$. The slope of the line is B.

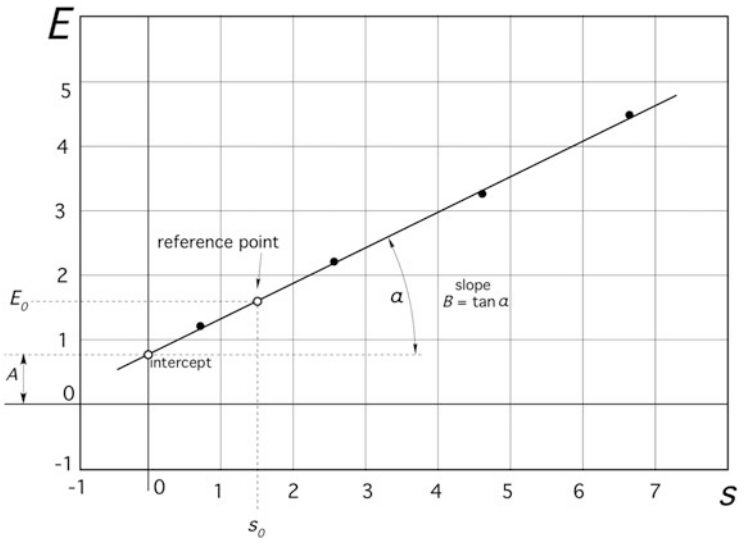

Fig. 2.2 Linear transfer function. *Black dots* indicate experimental data

Sometimes it is called *sensitivity* since the larger this coefficient the greater the stimulus influence. The slope B is a tangent of the angle α. The output E may be the amplitude of voltage or current, phase, frequency, pulse-width modulation (PWM), or a digital code, depending on the sensor properties, signal conditioning, and interface circuit.

Note that Eq. (2.2) assumes that the transfer function passes, at least theoretically, through zero value of the input stimulus s. In many practical cases it is just difficult or impossible to test a sensor at a zero input. For example, a temperature sensor used on a Kelvin scale cannot be tested at the absolute zero (−273.15 °C). Thus, in many linear or quasilinear sensors it may be desirable to reference the sensor not to the zero input but rather to some more practical input reference value s_0. If the sensor response is E_0 for some known input stimulus s_0, Eq. (2.2) can be rewritten in a more practical form:

$$E = E_0 + B(s - s_0) \tag{2.3}$$

The reference point has coordinates s_0 and E_0. For a particular case where $s_0 = 0$, Eq. (2.3) becomes Eq. (2.2) and $E_0 = A$. The inverse linear transfer function for computing the input stimulus from the output E is

$$s = \frac{E - E_0}{B} + s_0 \tag{2.4}$$

Note that three constants shall be known for computing the stimulus s: sensitivity B and coordinates s_0 and E_0 of the reference point.

Very few sensors are truly linear. In the real world, at least a small nonlinearity is almost always present, especially for a broad input range of the stimuli. Thus, Eqs. (2.2) and (2.3) represent just a linear approximation of a nonlinear sensor's response, where a nonlinearity can be ignored for the practical purposes. In many cases, when nonlinearity cannot be ignored, the transfer function still may be approximated by a group of linear functions as we shall discuss below in greater detail (Sect. 2.1.6).

A nonlinear transfer function can be approximated by a nonlinear mathematical function. Here are few useful functions.

The *logarithmic* approximation function (Fig. 2.3) and the corresponding inverse function (which is exponential) are respectively:

$$E = A + B \ln s, \tag{2.5}$$

$$s = \mathrm{e}^{\frac{E-A}{B}}, \tag{2.6}$$

where A and B are the fixed parameters.

The exponential function (Fig. 2.4) and its inverse (which is logarithmic) are given by:

$$E = A\mathrm{e}^{ks}, \tag{2.7}$$

$$s = \frac{1}{k} \ln \frac{E}{A}, \tag{2.8}$$

where A and k are the fixed parameters.

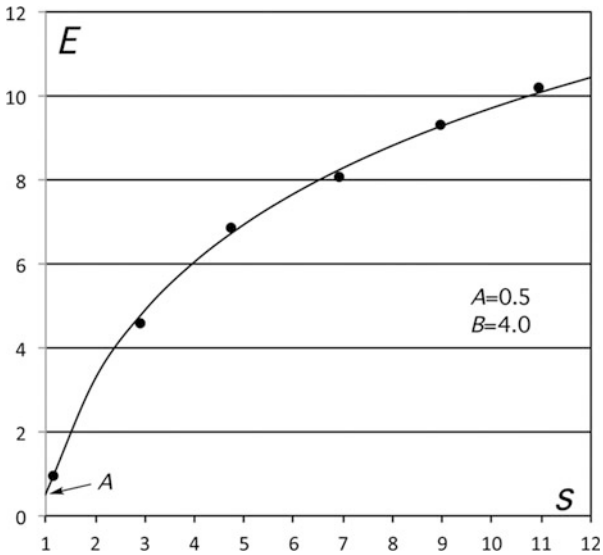

Fig. 2.3 Approximation by logarithmic function. *Dots* indicate experimental data

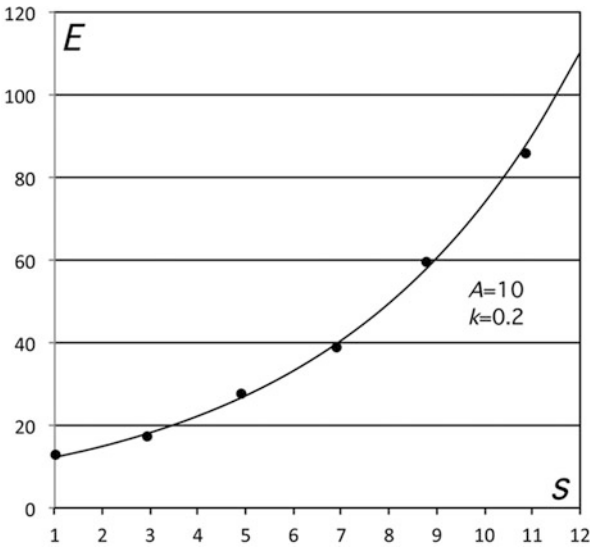

Fig. 2.4 Approximation by an exponential function. *Dots* indicate experimental data

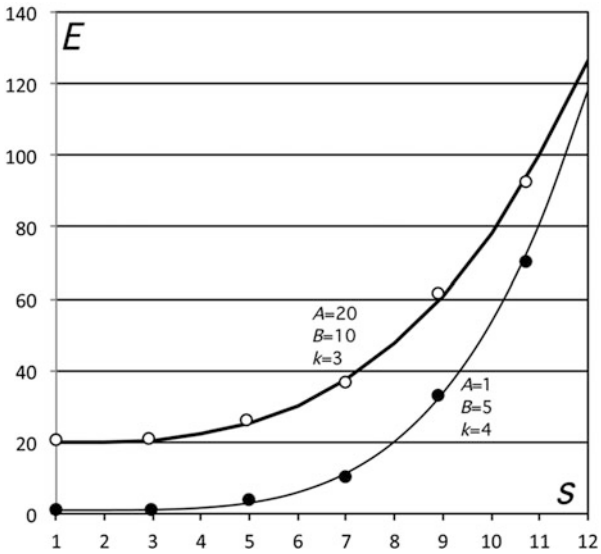

Fig. 2.5 Power functions

The *power* function (Fig. 2.5) and its inverse can be expressed as

$$E = A + Bs^k, \tag{2.9}$$

$$s = \sqrt[k]{\frac{E - A}{B}}, \tag{2.10}$$

where A and B are fixed parameters and k is the power factor.

All the above three nonlinear approximations possess a small number of parameters that shall be determined during calibration. A small number of parameters makes them rather convenient, provided that they can fit response of a particular sensor. It is always useful to have as small a number of parameters as possible, not the least for the sake of lowering cost of the sensor calibration. The fewer parameters, the smaller the number of the measurements to be made during calibration.

2.1.3 Linear Regression

If measurements of the input stimuli during calibration cannot be made consistently with high accuracy and large random errors are expected, the minimal number of measurements will not yield a sufficient accuracy. To cope with random errors in the calibration process, a method of *least squares* could be employed to find the slope and intercept. Since this method is described in many textbooks and manuals, only the final expressions for the unknown parameters of a linear regression are given here for reminder. The reader is referred to any textbook on statistical error analysis. The procedure is as follows:

1. Measure multiple (k) output values E at the input values s over a substantially broad range, preferably over the entire sensor span.
2. Use the following formulas for a linear regression to determine intercept A and slope B of the best-fitting straight line of Eq. (2.2):

$$A = \frac{\Sigma E \Sigma s^2 - \Sigma s \Sigma s E}{k\Sigma s^2 - (\Sigma s)^2}, \quad B = \frac{k\Sigma s E - \Sigma s \Sigma E}{k\Sigma s^2 - (\Sigma s)^2}, \tag{2.11}$$

where Σ is the summation over all k measurements. When the constants A and B are found, Eq. (2.2) can be used as a linear approximation of the experimental transfer function.

2.1.4 Polynomial Approximations

A sensor may have such a transfer function that none of the above basic functional approximations would fit sufficiently well. A sensor designer with a reasonably good mathematical background and physical intuition may utilize some other suitable functional approximations, but if none is found, several old and reliable techniques may come in handy. One is a polynomial approximation, that is, a power series.

Any continuous function, regardless of its shape, can be approximated by a power series. For example, the exponential function of Eq. (2.7) can be

approximately calculated from a third-order polynomial by dropping all the higher terms of its series expansion[1]:

$$E = Ae^{ks} \approx A\left(1 + ks + \frac{k^2}{2!}s^2 + \frac{k^3}{3!}s^3\right) \tag{2.12}$$

In many cases it is sufficient to see if the sensor's response can be approximated by the second or third degree polynomials to fits well enough into the experimental data. These approximation functions can be expressed respectively as

$$E = a_2 s^2 + a_1 s + a_0 \tag{2.13}$$

$$E = b_3 s^3 + b_2 s^2 + b_1 s + b_0 \tag{2.14}$$

The factors a and b are the constants that allow shaping the curves (2.13) and (2.14) into a great variety of the practical transfer functions. It should be appreciated that the quadratic (second order) polynomial of Eq. (2.13) is a special case of the third degree polynomial when $b_3 = 0$ in Eq. (2.14). Similarly, the first-order (linear) polynomial of Eq. (2.2) is a special case of the quadratic polynomial of Eq. (2.13) with $a_2 = 0$.

Obviously, the same technique can be applied to the inverse transfer function as well. Thus, the inverse transfer function can be approximated by a second or third degree polynomial:

$$s = A_2 E^2 + A_1 E + A_0 \tag{2.15}$$

$$s = B_3 E^3 + B_2 E^2 + B_1 E + B_0 \tag{2.16}$$

The coefficients A and B can be converted into coefficients a and b, but the analytical conversion is rather cumbersome and rarely used. Instead, depending in the need, usually either a direct or inversed transfer function is approximated from the experimental data points, but not both.

In some cases, especially when more accuracy is required, the higher order polynomials should be considered because the higher the order of a polynomial the better the fit. Still, even a second-order polynomial often may yield a fit of sufficient accuracy when applied to a relatively narrow range of the input stimuli and the transfer function is monotonic (no ups and downs).

[1] This third-order polynomial approximation yields good approximation only for $ks \ll 1$. In general, the error of a power series approximation is subject of a rather nontrivial mathematical analysis. Luckily, in most practical situations that analysis is rarely needed.

2.1.5 Sensitivity

Recall that the coefficient B in Eqs. (2.2) and (2.3) is called *sensitivity*. For a nonlinear transfer function, sensitivity is not a fixed number, as would be the case in a linear transfer function. A nonlinear transfer function exhibits different sensitivities at different points in intervals of stimuli. In the case of nonlinear transfer functions, sensitivity is defined as a first derivative of the transfer function at the particular stimulus s_i:

$$b_i(s_i) = \frac{\mathrm{d}E(s_i)}{\mathrm{d}s} = \frac{\Delta E_i}{\Delta s_i}, \tag{2.17}$$

where, Δs_i is a small increment of the input stimulus and ΔE_i is the corresponding change in the sensor output E.

2.1.6 Linear Piecewise Approximation

A linear piecewise approximation is a powerful method to employ in a computerized data acquisition system. The idea behind it is to break up a nonlinear transfer function of any shape into sections and consider each such section being linear as described by Eq. (2.2) or (2.3). Curved segments between the sample points (knots) demarcating the sections are replaced with straight-line segments, thus greatly simplifying behavior of the function between the knots. In other words, the knots are graphically connected by straight lines. This can also be seen as a polygonal approximation of the original nonlinear function. Figure 2.6 illustrates

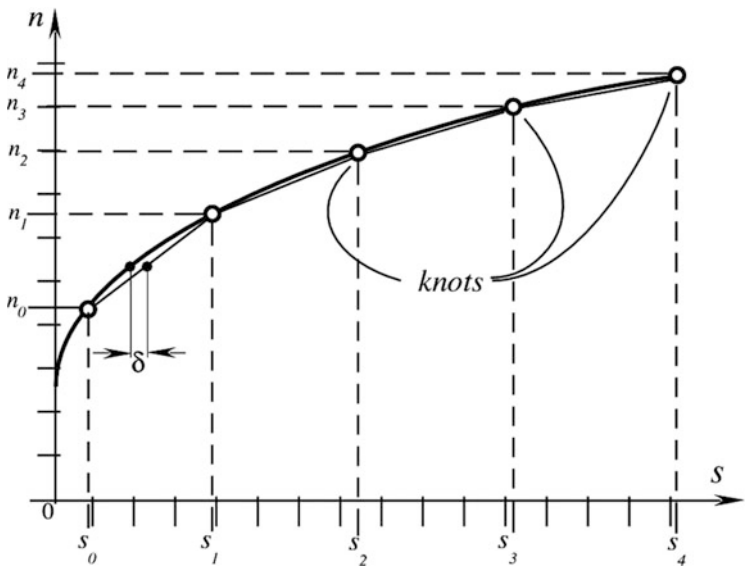

Fig. 2.6 Linear piecewise approximation

the linear piecewise approximation of a nonlinear function with the knots at input values s_0, s_1, s_2, s_3, s_4, and the corresponding output values n_0, n_1, n_2, n_3, n_4 (in this example, the digital counts from an ADC).

It makes sense to select knots only for the input range of interest (a span—see definition in the next chapter); thus in Fig. 2.6 a section of the curve from 0 to s_0 is omitted as being outside of the practically required span limits.

An error of a piecewise approximation can be characterized by a maximum deviation δ of the approximation line from the real curve. Different definitions exist for this maximum deviation (mean square, absolute max, average, etc.); but whatever is the adopted metric, the larger δ calls for a greater number of samples, that is a larger number of sections with the idea of making this maximum deviation acceptably small. In other words, the larger the number of the knots the smaller the error. The knots do not need to be equally spaced. They should be closer to each other where nonlinearity is high and farther apart where nonlinearity is small.

While using this method, the signal processor should store the knot coordinates in a memory. For computing the input stimulus s a linear interpolation should be performed (see Sect. 2.4.2).

2.1.7 Spline Interpolation

Approximations by higher order polynomials (third order and higher) have some disadvantages; the selected points at one side of the curve make strong influence on the remote parts of the curve. This deficiency is resolved by the *spline* method of approximation. In a similar way to a linear piecewise interpolation, the spline method is using different third-order polynomial interpolations between the selected experimental points called knots [1]. It is a curve between two neighboring knots and then all curves are "stitched" or "glued" together to obtain a smooth combined curve fitting. Not necessarily it should be a third-order curve—it can be as simple as the first-order (linear) interpolation. A linear spline interpolation (first order) is the simplest form and is equivalent to a linear piecewise approximation as described above.

The spline interpolation can utilize polynomials of different degrees, yet the most popular being cubic (third order) polynomials. Curvature of a line at each point is defined by the second derivative. This derivative should be computed at each knot. If the second derivatives are zero, the cubic spline is called "relaxed" and it is the choice for many practical approximations. Spline interpolation is the efficient technique when it comes to an interpolation that preserves smoothness of the transfer function. However, simplicity of the implementation and the computational costs of a spline interpolation should be taken into account particularly in a tightly controlled microprocessor environment.

2.1.8 Multidimensional Transfer Functions

A sensor transfer function may depend on more than one input variable. That is, the sensor's output may be a function of several stimuli. One example is a humidity sensor whose output depends on two input variables—relative humidity and temperature. Another example is the transfer function of a thermal radiation (infrared) sensor. This function[2] has two arguments—two temperatures: $T_{b,}$ the absolute temperature of an object of measurement and T_s, the absolute temperature of the sensing element. Thus, the sensor's output voltage V is proportional to a difference of the fourth-order parabolas:

$$V = G\left(T_b^4 - T_s^4\right), \tag{2.18}$$

where G is a constant. Clearly, the relationship between the object's temperature T_B and the output voltage V is not only nonlinear but also in a nonlinear way depends on the sensing element surface temperature T_s, which should be measured by a separate contact temperature sensor. The graphical representation of a two-dimensional transfer function of Eq. (2.18) is shown in Fig. 2.7.

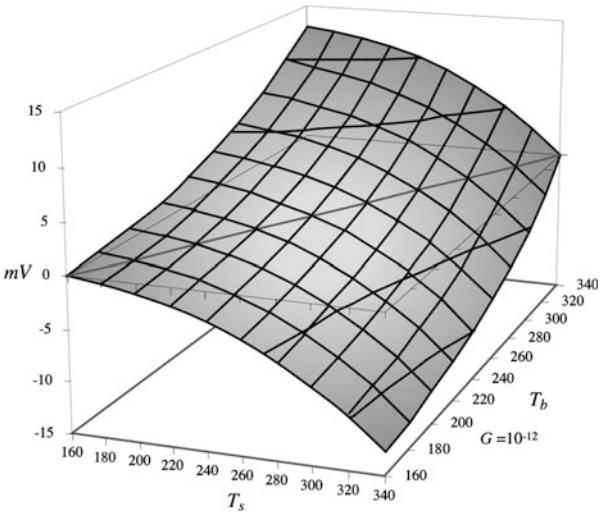

Fig. 2.7 Two-dimensional transfer function of thermal radiation sensor. Temperatures are in K

[2] This function is known as the Stefan-Boltzmann law (Sect. 4.12.3).

2.2 Calibration

If tolerances of a sensor and interface circuit (signal conditioning) are broader than the required overall accuracy, a calibration of the sensor or, preferably, a combination of a sensor and its interface circuit is required for minimizing errors. In other words, a calibration is required whenever a higher accuracy is required from a less accurate sensor. For example, if one needs to measure temperature with accuracy, say 0.1 °C, while the available sensor is rated as having accuracy of 1 °C, it does not mean that the sensor cannot be used. Rather this particular sensor needs calibration. That is, its unique transfer function should be determined. This process is called *calibration*.

A calibration requires application of several precisely known stimuli and reading the corresponding sensor responses. These are called the *calibration points* whose input–output values are the point coordinates. In some lucky instances only one pair is required, while typically 2–5 calibration points are needed to characterize a transfer function with a higher accuracy. After the unique transfer function is established, any point in between the calibration points can be determined.

To produce the calibration points, a standard reference source of the input stimuli is required. The reference source should be well maintained and periodically checked against other established references, preferably traceable to a national standard, for example a reference maintained by NIST[3] in the U.S.A. It should be clearly understood that the calibration accuracy is directly linked to accuracy of a reference sensor that is part of the calibration equipment. A value of uncertainty of the reference sensor should be included in the statement of the overall uncertainty, as explained in Sect. 3.21.

Before calibration, either a mathematical model of the transfer function has to be known or a good approximation of the sensor's response over the entire span shall be found. In a great majority of cases, such functions are smooth and monotonic. Very rarely they contain singularities and if they do, such singularities are the useful phenomena that are employed for sensing (an ionizing particle detector is an example).

Calibration of a sensor can be done in several possible ways, some of which are the following:

1. Modifying the transfer function or its approximation to fit the experimental data. This involves computation of the coefficients (parameters) for the selected transfer function equation. After the parameters are found, the transfer function becomes unique for that particular sensor. The function can be used for computing the input stimuli from any sensor response within the range. Every calibrated sensor will have its own set of the unique parameters. The sensor is not modified.
2. Adjustment of the data acquisition system to trim (modify) its output by making the outputs signal to fit into a normalized or "ideal" transfer function.

[3] NIST—National Institute of Standards and Technology: www.nist.gov

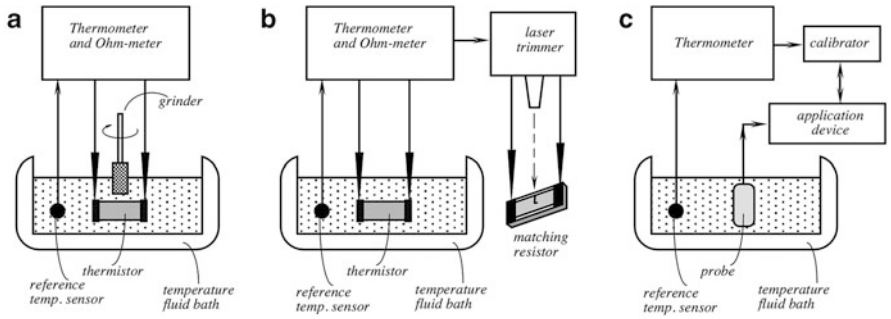

Fig. 2.8 Calibration of thermistor: grinding (**a**), trimming reference resistor (**b**), and determining calibrating points for characterizing transfer function (**c**)

An example is a scaling and shifting the acquired data (modifying the system gain and offset). The sensor is not modified.

3. Modification (trimming) the sensor's properties to fit the predetermined transfer function, thus the sensor itself is modified.
4. Creating the sensor-specific reference device with the matching properties at particular calibrating points. This unique reference is used by the data acquisition system to compensate for the sensor's inaccuracy. The sensor is not modified.

As an example, Fig. 2.8 illustrates three methods of calibrating a thermistor (temperature sensitive resistor). Figure 2.8a shows a thermistor that is immersed into a stirred liquid bath with a precisely controlled and monitored temperature. The liquid temperature is continuously measured by a precision reference thermometer. To prevent shorting the thermistor terminals, the liquid should be electrically nonconductive, such as mineral oil or Fluorinert™. The resistance of the thermistor is measured by a precision Ohmmeter. A miniature grinder mechanically removes some material from the thermistor body to modify its dimensions. Reduction in dimensions leads to increase in the thermistor electrical resistance at the selected bath temperature. When the thermistor's resistance matches a predetermined value of the "ideal" resistance, the grinding stops and the calibration is finished. Now the thermistor response is close to the "ideal" transfer function, at least at that temperature. Naturally, a single-point calibration assumes that the transfer function can be fully characterized by that point.

Another way of calibrating a thermistor is shown in Fig. 2.8b where the thermistor is not modified but just measured at a selected reference temperature. The measurement provides a number that is used for selecting a conventional (temperature stable) matching resistor as a unique reference. That resistor is for use in the interface scaling circuit. The precise value of such a reference resistor is achieved either by a laser trimming or selection from a stock. That individually matched pair thermistor–resistor is used in the measurement circuit, for example, in

a Wheatstone bridge. Since it is a matching pair, the response of the bridge will scale to correspond to an "ideal" transfer function of a thermistor.

In the above examples, methods (a) and (b) are useful for calibration at one temperature point only, assuming that other parameters of the transfer function do not need calibration. If such is not the case, several calibrating points at different temperatures and resistances should be generated as shown in Fig. 2.8c. Here, the liquid bath is sequentially set at two, three, or four different temperatures and the thermistor under calibration produces the corresponding responses, that are used by the calibrating device to generate the appropriate parameters for the inverse transfer function that will be stored in the application device (e.g., a thermometer).

2.3 Computation of Parameters

If a transfer function is linear, as in Eq. (2.2), then calibration should determine constants A and B. If it is exponential as in Eq. (2.7), the constants A and k should be determined, and so on.

To calculate parameters (constants) of a linear transfer function one needs two data points defined by two calibrating input–output pairs. Consider a simple linear transfer function of Eq. (2.3). Since two points are required to define a straight line, a two-point calibration shall be performed. For example, if one uses a forward-biased semiconductor p–n junction (Fig. 2.9a) as a temperature sensor (see Sect. 17.6), its transfer function is linear (Fig. 2.9b) with temperature t being the input stimulus and the ADC count n from the interface circuit is the output:

$$n = n_1 + B(t - t_1). \tag{2.19}$$

Note that t_1 and n_1 are the coordinates of the first reference calibrating point. To fully define the line, the sensor shall be subjected to two calibrating temperatures (t_1 and t_2) for which two corresponding output counts (n_1 and n_2) will be registered. At the first calibrating temperature t_1, the output count is n_1.

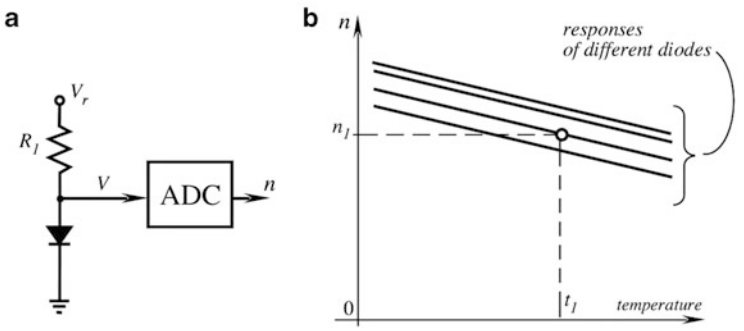

Fig. 2.9 A p–n junction temperature sensor (**a**) and transfer functions for several sensors (**b**). Each diode will produce different n_1 at the same temperature t_1

After subjecting the sensor to the second calibrating temperature t_2, we receive the digital counts for the second calibrating point. The count is

$$n_2 = n_1 + B(t_2 - t_1) \tag{2.20}$$

from which the sensitivity (slope) is computed as

$$B = \frac{n_2 - n_1}{t_2 - t_1} \tag{2.21}$$

and Eq. (2.19) becomes a linear transfer function with now three known parameters: B, n_1, and t_1. The sensitivity (slope) B is in count/degree. In example of Fig. 2.9, the slope B is negative since a p–n junction has a negative temperature coefficient (NTC). Note that the parameters found from calibration are unique for the particular sensor and must be stored in the measurement system to which that particular sensor is connected. For another similar sensor, these parameters will be different (perhaps except t_1, if all sensors are calibrated at exactly the same temperature). After calibration is done, any temperature within the operating range can be computed from the ADC output count n by use of the inverse transfer function

$$t = t_1 + \frac{n - n_1}{B} \tag{2.22}$$

In some fortunate cases, parameter B may be already known with a sufficient accuracy so that no computation of B is needed. In a p–n junction of Fig. 2.9a, the slope B is usually very consistent for a given lot and type of the semiconductor wafer and thus can be considered as a known parameter for all diodes in the production lot. However, all diodes may have different offsets, so a single-point calibration is still needed to find out n_1 for each individual sensor at the calibrating temperature t_1.

For nonlinear transfer functions, calibration at one data point may be sufficient only in some rare cases when other parameters are already known, but often two and more input–output calibrating pairs would be required. When a second or a third degree polynomial transfer functions are employed, respectively three and four calibrating pairs are required. For a third-order polynomial

$$E = b_3 s^3 + b_2 s^2 + b_1 s + b_0 \tag{2.23}$$

to find four parameters b_0 to b_3, four experimental calibrating input–output pairs (calibrating points) are required: s_1 and E_1, s_2 and E_2, s_3 and E_3, and s_4 and E_4.

Plugging these experimental pairs into Eq. (2.23) we get a system of four equations

$$\begin{aligned} E_1 &= b_3 s_1^3 + b_2 s_1^2 + b_1 s_1 + b_0 \\ E_2 &= b_3 s_2^3 + b_2 s_2^2 + b_1 s_2 + b_0 \\ E_3 &= b_3 s_3^3 + b_2 s_3^2 + b_1 s_3 + b_0 \\ E_4 &= b_3 s_4^3 + b_2 s_4^2 + b_1 s_4 + b_0 \end{aligned} \tag{2.24}$$

To solve this system for the parameters, first we compute the determinants of the system:

$$\Delta = \left(\frac{s_1^2 - s_2^2}{s_1 - s_2} - \frac{s_1^2 - s_4^2}{s_1 - s_4}\right)\left(\frac{s_1^3 - s_2^3}{s_1 - s_2} - \frac{s_1^3 - s_3^3}{s_1 - s_3}\right) - \left(\frac{s_1^2 - s_2^2}{s_1 - s_2} - \frac{s_1^2 - s_3^2}{s_1 - s_3}\right)\left(\frac{s_1^3 - s_2^3}{s_1 - s_2} - \frac{s_1^3 - s_4^3}{s_1 - s_4}\right)$$

$$\Delta_a = \left(\frac{s_1^2 - s_2^2}{s_1 - s_2} - \frac{s_1^2 - s_4^2}{s_1 - s_4}\right)\left(\frac{E_1 - E_2}{s_1 - s_2} - \frac{E_1 - E_3}{s_1 - s_3}\right) - \left(\frac{s_1^2 - s_2^2}{s_1 - s_2} - \frac{s_1^2 - s_3^2}{s_1 - s_3}\right)\left(\frac{E_1 - E_2}{s_1 - s_2} - \frac{E_1 - E_4}{s_1 - s_4}\right)$$

$$\Delta_b = \left(\frac{s_1^3 - s_2^3}{s_1 - s_2} - \frac{s_1^3 - s_3^3}{s_1 - s_3}\right)\left(\frac{E_1 - E_2}{s_1 - s_2} - \frac{E_1 - E_4}{s_1 - s_4}\right) - \left(\frac{s_1^3 - s_2^3}{s_1 - s_2} - \frac{s_1^3 - s_4^3}{s_1 - s_4}\right)\left(\frac{E_1 - E_2}{s_1 - s_2} - \frac{E_1 - E_3}{s_1 - s_3}\right), \tag{2.25}$$

from which the polynomial coefficients are calculated in the following fashion:

$$\begin{aligned} b_3 &= \frac{\Delta_a}{\Delta}; \\ b_2 &= \frac{\Delta_b}{\Delta}; \\ b_1 &= \frac{1}{s_1 - s_4}\left[E_1 - E_4 - b_3\left(s_1^3 - s_4^3\right) - b_2\left(s_1^2 - s_4^2\right)\right]; \\ b_0 &= E_1 - b_3 s_1^3 - b_2 s_1^2 - b_1 s_1 \end{aligned} \tag{2.26}$$

If the determinant Δ is small, some considerable inaccuracy will result. Thus, the calibrating points should be spaced within the operating range as far as possible from one another.

When dealing with a large inertia or temperatures, calibration may be a slow process. To reduce the manufacturing cost, it is important to save time and thus to minimize the number of calibration points. Therefore, the most economical transfer function or the approximation should be selected. Economical means having the smallest number of the unknown parameters. For example, if an acceptable accuracy can be achieved by a second-order polynomial, a third order should not be used.

2.4 Computation of a Stimulus

A general goal of sensing is to determine the value of the input stimulus s from the measured output signal E. This can be done by two methods.

1. From an *inverted* transfer function $s = F(E)$, that may be either an analytical or approximation function, or
2. From a *direct* transfer function $E = f(s)$ by use of an iterative computation.

2.4.1 Use of Analytical Equation

This is a straight approach when an analytical equation for the transfer equation is known. Simply measure the output signal E, plug it into the formula, and compute the sought input stimulus s. For example, to compute a displacement from resistance of a potentiometric sensor, use Eq. (2.4). For other functional models, use respective Eqs. (2.4), (2.6), (2.8), and (2.10).

2.4.2 Use of Linear Piecewise Approximation

Refer to Sect. 2.1.6 for description of the approximation. For computing stimulus s, the very first step is to find out where it is located, in other words, between which knots lays the output signal E? The next step is to use the method of *linear interpolation* for computing the input stimulus s.

Here how it works.

First, determine where the output is located, that is, in between which knots? For example, we found that the output is somewhere in between the knots p_1 and p_4 as illustrated in Fig. 2.10. The sensor's output $E = n$ is in counts from the ADC. A large triangle is formed with the corners at the points p_1, p_3, and p_4. The unknown stimulus s_x corresponds to the measured ADC output counts n_x. This is indicated by the point p_5 on the approximation straight line, thus forming a smaller triangle between the points p_1, p_2, and p_5. Both triangles are similar, which allows us to derive a linear equation for computing the unknown stimulus s_x from the value of n_x:

$$s_x = s_i + \frac{n_x - n_i}{n_{i+1} - n_i}(s_{i+1} - s_i) \tag{2.27}$$

This equation is easy to program and compute by an inexpensive microprocessor, which keeps in its memory a look-up table containing the knot coordinates (Table 2.1).

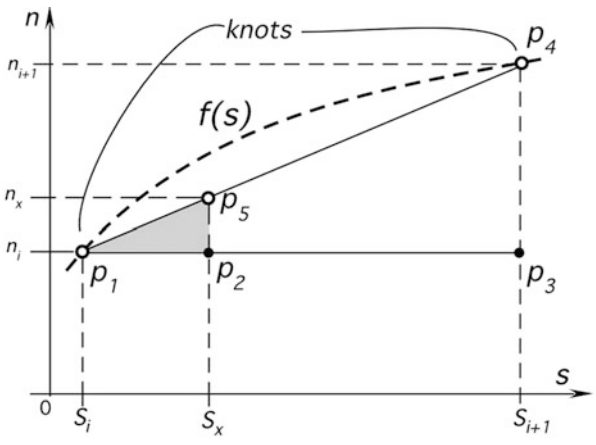

Fig. 2.10 Computation of stimulus from linear piecewise approximation

Table 2.1 Look-up table of knots for computing the input from the measured output

Knot	0	1	2	...	i	...	k
Output	n_0	n_1	n_2	...	n_i	...	n_k
Input	s_0	s_1	s_2	...	s_i	...	s_k

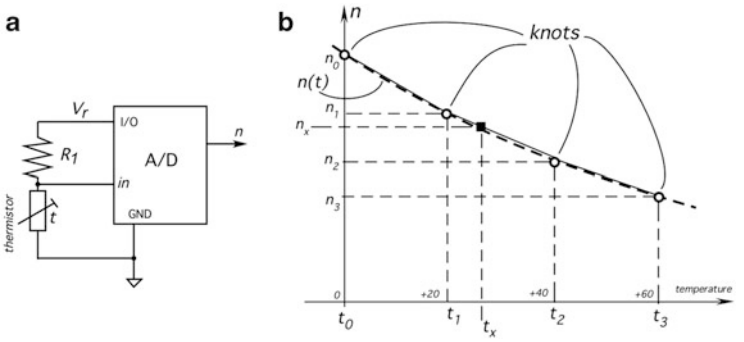

Fig. 2.11 Thermistor circuit (**a**) and its linear piecewise approximation (**b**) with four knots

For illustration, let us compare uses of a full functional model of the transfer function and a linear piecewise approximation. Obviously, the full functional model gives the most accurate computation. Figure 2.11a shows a thermistor temperature sensor with a pull-up resistor R_1 connected to a 12-bit analog-to-digital (ADC) converter (a full scale $N_0 = 4095$ counts corresponding to the reference voltage V_r). The thermistor is used to measure temperature in the total input span from 0 to +60 °C.

The output count of the thermistor measurements circuit can be modeled by a nonlinear function of temperature:

$$n_x = N_0 \frac{R_r e^{\beta\left(T_x^{-1} - T_r^{-1}\right)}}{R_1 + R_r e^{\beta\left(T_x^{-1} - T_r^{-1}\right)}}, \tag{2.28}$$

where T_x is the measured temperature, T_r is the reference temperature, R_r is resistance of the thermistor at reference temperature T_r, and β is the characteristic temperature. All temperatures and β are in degrees kelvin.

After manipulating Eq. (2.28), we arrive at the inverse transfer function that enables us to compute the input temperature in kelvin:

$$T_x = \left[\frac{1}{T_r} + \frac{1}{\beta}\ln\left(\frac{n_x}{N_0 - n_x}\frac{R_1}{R_r}\right)\right]^{-1} \tag{2.29}$$

The above Eqs. (2.28) and (2.29) contain two unknown parameters: R_r and β. Thus, before we proceed further, the entire circuit, including the ADC, shall be

calibrated at temperature T_r and also at some other temperature T_c. In the circuit, we use a pull-up resistor $R_1 = 10.0$ kΩ. For calibration, we select two calibrating temperatures in the operating range as $T_r = 293.15$ K and $T_c = 313.15$ K, which correspond to 20 °C and 40 °C, respectively.

During calibration, the thermistor sequentially is immersed into a fluid bath at these two temperatures and the ADC output counts are registered respectively as

$n_r = 1863$ at $T_r = 293.15$ K
$n_c = 1078$ at $T_c = 313.15$ K

By substituting these pairs into Eq. (2.28) and solving the system of two equations, we arrive at parameter values $R_r = 8.350$ kΩ and $\beta = 3895$ K. This completes the calibration.

Now, since all parameters in Eqs. (2.28) and (2.29) are fully characterized, Eq. (2.29) can be used for computing temperature from any ADC count in the operating range. We assume this is the most accurate way of computing true temperature. Now, let us see what involves using the linear piecewise approximation.

Let us break up the transfer function of Eq. (2.28) just into three sections (Fig. 2.11b) with two end knots at 0 and 60 °C (the span limits) and two equally spaced central knots at 20 and 40 °C. We will use linear approximations between the neighboring knot temperatures[4] $t_0 = 0$ °C and $t_1 = t_r = 20$ °C, $t_2 = 40$ °C, and $t_3 = 60$ °C.

From calibration, we find the ADC outputs at these knot temperatures:

$n_0 = 2819$ for $t_0 = 0$ °C
$n_1 = n_r = 1863$ for $t_1 = t_r = 20$ °C
$n_2 = 1078$ for $t_2 = 40$ °C
$n_3 = 593$ for $t_3 = 60$ °C

The count–temperature coordinate pairs are plugged into a look-up Table 2.2.

As an example, to compare temperatures computed form the functional model of Eq. (2.29) and Table 2.2, consider that at some unknown temperature the ADC outputs count $n_x = 1505$. We need to find that temperature. From Table 2.2 we determine that this measured count n_x is situated somewhere between the knots 1 and 2. To find temperature t_s, the measured counts and the knot values are plugged into Eq. (2.27) to arrive at

Table 2.2 Look-up table for computation of temperature

Knot	0	1	2	3
Counts	2819	1863	1078	593
Temp (°C)	0	20	40	60

[4] Note that the reference temperature in Celsius $t_r = t_1 = T_r - 273.15$, where T_r is in kelvin.

$$t_x = t_1 + \frac{n_x - n_1}{n_2 - n_i}(t_2 - t_1) = 20 + \frac{1505 - 1863}{1078 - 1863}(40 - 20) = 29.12\,^{\circ}\mathrm{C} \quad (2.30)$$

Now, to compare two methods of calculation, use a real transfer function Eq. (2.29) by plugging into it the same $n_x = 1505$. After calculation, we get the stimulus temperature $t_x = 28.22$ °C. This number is lower than the one computed from Eq. (2.30). Hence, the linear piecewise approximation with only two central knots overestimates temperature by 0.90 °C which may be a too much of an error. For a more demanding application, to reduce errors use more than two central knots.

2.4.3 Iterative Computation of Stimulus (Newton Method)

If the *inverse* transfer function is not known, the iterative method allows using a *direct* transfer function to compute the input stimulus. A very powerful method of iterations is the Newton or secant method[5] [1–3]. It is based on first *guessing* the initial reasonable value of stimulus $s = s_0$ and then applying the Newton algorithm to compute a series of new values of s converging to the sought stimulus value. Thus, the algorithm involves several steps of computation, where each new step brings us closer and closer to the sought stimulus value. When a difference between two consecutively computed values of s becomes sufficiently small (less than an acceptable error), the algorithm stops and the last computed value of s is considered a solution of the original equation and thus the value of the unknown stimulus is found. Newton's method converges remarkably quickly, especially if the initial guess is reasonably close to the actual value of s.

The output signal is represented through the sensor's transfer function is $f(s)$ as $E = f(s)$. It can be rewritten as $E - f(s) = 0$. The Newton method prescribes computing the following *sequence* of the stimuli values for the measured output value E:

$$s_{i+1} = s_i - \frac{f(s_i) - E}{f'(s_i)} \quad (2.31)$$

This sequence after just several steps converges to the sought input s. Here, s_{i+1} is the computed stimulus value at the iteration $i+1$, wherein s_i is the computed value at a prior iteration i and $f'(s_i)$ is the first derivative of the transfer function at input s_i. The iteration number is $i = 0, 1, 2, 3, \ldots$. Note that the same measured value E is used in all iterations.

Start by guessing stimulus s_0, then use Eq. (2.31) to calculate the next approximation to the true stimulus s. Then, do it again by using the result from the prior approximation of s. In other words, computation of the subsequent s_i is performed

[5] This method is also known as the Newton–Raphson method, named after Isaac Newton and Joseph Raphson.

several times (iterations) until the incremental change in s_i becomes sufficiently small, preferably in the range of the sensor resolution.

To illustrate use of the Newton method let us assume that our direct transfer function is a third degree polynomial:

$$f(s) = as^3 + bs^2 + cs + d, \tag{2.32}$$

having coefficients $a = 1.5$, $b = 5$, $c = 25$, $d = 1$. The next step is plugging Eqs. (2.32) into (2.31) to arrive at the iteration of s_{i+1}:

$$s_{i+1} = s_i - \frac{as_i^3 + bs_i^2 + cs_i + d - E}{3as_i^2 + 2bs_i + c} = \frac{2as_i^3 + bs_i^2 - d + E}{3as_i^2 + 2bs_i + c} \tag{2.33}$$

This formula is used for all subsequent iterations. Let us assume, for example, that we measured the sensor's response $E = 22.000$ and our guess for the true stimulus is $s_0 = 2$. Then Eq. (2.33) will result in the following iterative sequence of the computed stimuli s_{i+1}:

$$\begin{aligned} s_1 &= \frac{2 \cdot 1.5 \cdot 2^3 + 5 \cdot 2^2 - 1 + 22}{3 \cdot 1.5 \cdot 2^2 + 2 \cdot 5 \cdot 2 + 25} = 1.032 \\ s_2 &= \frac{2 \cdot 1.5 \cdot 1.032^3 + 5 \cdot 1.032^2 - 1 + 22}{3 \cdot 1.5 \cdot 1.032^2 + 2 \cdot 5 \cdot 1.032 + 25} = 0.738 \\ s_3 &= \frac{2 \cdot 1.5 \cdot 0.738^3 + 5 \cdot 0.738^2 - 1 + 22}{3 \cdot 1.5 \cdot 0.738^2 + 2 \cdot 5 \cdot 0.738 + 25} = 0.716 \\ s_4 &= \frac{2 \cdot 1.5 \cdot 0.716^3 + 5 \cdot 0.716^2 - 1 + 22}{3 \cdot 1.5 \cdot 0.716^2 + 2 \cdot 5 \cdot 0.716 + 25} = 0.716 \end{aligned} \tag{2.34}$$

We see that after just the third iteration, the sequence of s_i converges to 0.716.

Hence, at step 4, the Newton algorithm stops and the stimulus value is deemed to be $s = 0.716$. To check accuracy of this solution, plug the s number into Eq. (2.32) and obtain $f(s) = E = 22.014$, which is within 0.06 % of the actually measured response $E = 22.000$.

It should be noted that the Newton method results in large errors when the sensor's sensitivity becomes low. In other words, the method will fail where the transfer function flattens (1st derivative approaches zero). In such cases, the so-called Modified Newton Method may be employed. In some cases when the first derivative cannot be easily computed analytically, one uses instead a sensitivity value devised from Δs and ΔE as in (2.17).

References

1. Stoer, J., & Bulirsch, R. (1991). *Introduction to numerical analysis* (2nd ed., pp. 93–106). New York, NY: Springer.
2. Kelley, C. T. (2003). *Solving nonlinear equations with Newton's method. Number 1 in Fundamental algorithms for numerical calculations*. Philadelphia, PA: SIAM.
3. Süli, E., & Mayers, D. (2003). *An introduction to numerical analysis*. Cambridge, UK: Cambridge University Press.