Discretization Method of Continuous-Time Dynamic Linear Model

Sławomir Gryś, and Waldemar Minkina

Abstract—The article presents a new discretization method of a continuous-time linear model of sensor dynamics. It can be useful to reduce measuring errors related to the inertia of the sensor. For example it is important in the measurement of rapid processes as temperature changes in combustion chambers, or for shortening the time needed to establish the sensor readings in a transition state. There is assumed that sensor dynamics can be approximated by linear differential equation or transfer function. The searched coefficients of equivalent difference equation or discrete transfer function are obtained from Taylor expansion of a sensor output signal and then on the solution of the linear set of equations. The method does not require decomposition of sensor transfer function for zeros and poles and can be applied to the case of transfer function with zeros equal to zero. The method was used to compensate the dynamics of sensor measuring fast signals. The Bode characteristics of a compensator were compared with others derived using classical methods of discretization of linear models. Additionally, signals in time were presented to show the dynamic error before and after compensation.

Keywords—sensor dynamics; compensation; discrete models; temperature measurement

I. Introduction

ANY analog sensors, including thermometers, can be characterized by static and dynamic parameters being the source of undesired measurement errors. In many situations, this means lowering the quality of process control and produced goods. A more drastic event is damaging the production line or produced components due to an wrong operation of the automation systems. Examples of industrial applications where a rapid temperature rise can lead to a potential catastrophe include: nuclear power plants, the chemical industry, aviation, and the engineering industry. Compensation of static non-linearity is generally not very problematic if the invariability of its shape over time and the measuring range of the sensor are assumed. The dynamics of a thermometer sensor, on the other hand, is related to its thermal inertia and thermal resistance occurring along the path between the medium under test, the sensor case and its most important sensitive part reacting for a signal change. Basic factors influencing the thermometer dynamics include [1-4]:

- the type of material of the sensor,
- shape and dimensions of the sensor,
- operating point and temperature range to be measured,
- the type of medium whose temperature is being measured, its viscosity, pressure and flow rate,
- the method of sensor placement,
- the pressure of the sensor on the object under test,
- the surface condition of the sensor or the object under test.

The real dynamical properties must be approximated with a mathematical model for the sake of error correction. Typically it is done by using linear and nonlinear differential equations, transfer functions or model with a variable 'time constant'. The model is suited to known and repetitive heat transfer conditions. If they changed, the model, or at least its time constant should be adapted. The typical models and their limitations are widely discussed in the monograph [5] and papers [6, 7].

The signal disturbances due to the limited 'speed' of a sensor can be partially reduced by compensation of dynamical properties of sensor. In the measurement part of the system, the analog element is the sensor and conditioning components converting and amplifying measured quantity to the electrical voltage, matching it to the range of the A/D converter. Further signal processing takes place in the digital form. It can be realized in software or hardware, e.g. by using digital filtering. In digital technology, adaptive compensators sometimes referred to as 'self-tuning compensators' due to the software implementation of mechanisms for updating the compensator coefficients, are much easier to build than in analogue technology [8-15]. The scope of application of analogue compensators is mainly limited to the measurement of small temperature changes and constant heat transfer conditions. More possibilities are offered by digital compensation. Generally, there are used models in the time or frequency domains and we can estimate a digital model of the sensor or look directly for a digital model of compensator. The authors' proposal of both options will be presented in this paper providing the theoretical consideration and example of application.

II. The discretization methods – short overview

Since the A/D converter samples the analogue signal in time, it is convenient to operate with mathematical models in the discrete-time domain, such as differential equations or discrete transfer function. The most common methods for discretizing continuous models are the bilinear method (also called Tustin method), zero-pole matching method, zero/first-order methods, impulse invariant and others [16, 19]. We will use some of them as an example of how to find the model of a compensator of the sensor dynamics. To explain the compensation rule of

First Author is with Częstochowa University of Technology, Faculty of Electrical Engineering, Poland, (e-mail: slawomir.grys@pcz.pl).

Second Author is with Jan Dlugosz University in Czestochowa, Faculty of Science and Technology, Poland (minkinawaldemar@gmail.com).

operation let the sensor's dynamical properties be characterized by simple first-order model (inertia term) in continuous-time domain. The compensator is adding some correcting term 'corr' to the sensor output signal t_T and for this case we have:

$$t_K(\tau) = t_T(\tau) + corr = t_T(\tau) + N_K \frac{dt_T(\tau)}{d\tau} \qquad (1)$$

or in the complex s-operator domain:

$$G_K(s) = 1 + N_K s \qquad (2)$$

where: N_K is the time constant of the compensator, τ is time. The perfect compensation is obtained if the following matching condition will be satisfied $N_K = N_T$, where the last designation is the time constant of the sensor. The general way is used to express the compensator model, i.e. it is just a reciprocal of sensor transfer function given by numerator / denominator form or swaps the symbols of the sensor input and output signals in the differential equation.

In a bilinear method, an approximation is done from the domain of variable s to the domain of variable z by formula:

$$s \approx \frac{2}{h} \cdot \frac{z-1}{z+1} \qquad (3)$$

and h is the discretization step. By insertion the (3) into (2) and making a shift on the time axis, the following rules for coefficients values of the compensator are obtained:

$$a_0 = \frac{2}{h} N_K + 1, \quad a_1 = -\frac{2}{h} N_K + 1, \quad b_1 = 1 \qquad (4)$$

We see now that $m = 1$ and $n = 1$ are in the discrete transfer function of the compensator:

$$G_K(z) = \frac{a_0 + a_1 z^{-1} + a_2 z^{-2} + \cdots + a_m z^{-m}}{1 + b_1 z^{-1} + b_2 z^{-2} + \cdots + b_n z^{-n}} \qquad (5)$$

Another widely used way of discretization is the method with the mapping of zeros and poles. It uses the relationship as follows:

$$z = e^{sh} \qquad (6)$$

Discretization is carried out here in two stages. Knowing the zeros and poles of the continuous-time model, the discrete equivalent is determined according to relation (6). The transmittance coefficients obtained in this way are multiplied by an appropriate constant numerical value, so that the static amplification for continuous and discrete model are equal. The following condition is then satisfied:

$$\lim_{s \to 0} G(s) = \lim_{z \to 1} G(z) \qquad (7)$$

If $m = 1$, $n = 0$ in equation (5), then compensator coefficients can be directly determined from the formulas:

$$a_0 = \frac{1}{1 - exp\left(\frac{-h}{N_K}\right)}, \quad a_1 = -a_0 \cdot exp\left(\frac{-h}{N_K}\right) \qquad (8)$$

This discretization method returns the minimum number of

TABLE I

COEFFICIENTS OF THE DISCRETE COMPENSATOR REALIZING THE TRANSFER FUNCTION (2) AND DETERMINED BY THE INTERPOLATION METHOD

	a_0	a_1	a_2	a_3
$m=1$	$N_K/h+1$	$-N_K/h$	0	0
$m=2$	$3/2 \cdot N_K/h$	$-2 \cdot N_K/h$	$1/2 \cdot N_K/h$	0
$m=3$	$11/6 \cdot N_K/h+1$	$-3 \cdot N_K/h$	$3/2 \cdot N_K/h$	$-1/3 \cdot N_K/h$

coefficients of the equation, i.e. the order of the differential equation (continuous-time) is the same as the order of the difference equation (discrete-time). For low sampling frequencies and thus high discretization steps, a significant error in approximating the continuous model with the discrete model can occur. Reducing this error can be done by increasing the order of the differential equation if the sampling frequency cannot be chosen higher. Such a possibility is provided by another of the discretization methods presented, called the interpolation method, in which the differential is approximated by the backward difference for a given number of interpolation points, using, for example, the Lagrange polynomial. Table I gives the values of the coefficients of the discrete compensator described, in the continuous-time domain, by formula (1) for $m = 1, 2, 3$ a priori assumed interpolation nodes, respectively. In the book (5) there was shown that approximation error can be neglected if $h/N_K < 30$ and selected more than one interpolation parameter, i.e. $m > 1$.

As an alternative to the above-mentioned methods, a method based on Taylor series expansion at successive time instants can be used. This method in matrix notation and using matrix algebra has been proposed by the authors of this paper. Similarly to the interpolation method, the number of coefficients of the differential equation approximating the properties of the differential equation can be a priori assumed. This method will be derived both for sensor and compensator models.

III. PROPOSAL OF DISCRETIZATION METHOD IN MATRIX NOTATION

Let the dynamics of the sensor be described by a differential equation (9) of the order M with constant coefficients, where the inequality $M \geq N$ is assumed to satisfy the rule of causality expected in real measuring situations:

$$A_M \frac{d^M y(\tau)}{d\tau^M} + \cdots + A_1 \frac{dy(\tau)}{d\tau} + A_0 y(\tau) =$$
$$= B_0 x(\tau) + B_1 \frac{dx(\tau)}{d\tau} + \cdots + B_N \frac{d^N x(\tau)}{d\tau^N} \qquad (9)$$

where:

τ - time,
$x(\tau)$ - sensor input signal (measured signal),
$y(\tau)$ - sensor output signal.

Formula (9) can be represented as (10):

$$A'_M \frac{d^M y(\tau)}{d\tau^M} + \cdots + A'_1 \frac{dy(\tau)}{d\tau} + A'_0 y(\tau) -$$
$$- B'_N \frac{d^N x(\tau)}{d\tau^N} - \cdots - B'_1 \frac{dx(\tau)}{d\tau} = x(\tau) \qquad (10)$$

where:

$B_0 \neq 0$, $A'_i = A_i/B_0$ and $i = 0,1,...,M$, $B'_j = B_j/B_0$, $j = 1,2,...,N$. (11)

A. Discretization of model of dynamics compensator

The perfect compensation demand dictates that the signal $x(\tau)$ must be reproduced by the output signal $y_K(\tau)$ of the sensor-compensator system. Thus, it can be written:

$$y_K(\tau) = A'_M \frac{d^M y(\tau)}{d\tau^M} + \cdots + A'_1 \frac{dy(\tau)}{d\tau} + A'_0 y(\tau) -$$
$$- B'_N \frac{d^N y_K(\tau)}{d\tau^N} - \cdots - B'_1 \frac{dy_K(\tau)}{d\tau} \qquad (12)$$

The above formula describes the dynamics of a compensator in the continuous-time domain. Correspondingly, for discrete-time, a m-order differential equation (13) is defined, in which h in seconds denotes the discretization step as previously:

$$y_K(\tau) = a_0 y(\tau) + a_1 y(\tau - h) + \cdots + a_m y(\tau - m \cdot h) -$$
$$-b_1 y_K(\tau - h) - \cdots - b_n y_K(\tau - n \cdot h) \qquad (13)$$

Hardware based realization of the equation (13) can be carried out by an infinite response integrated digital filter with programmable coefficients. However, often the signal correction is preceded by the identification of the model parameters. An example of a sensor with a model with variable parameters is a thermocouple, whose parameters depend on many factors, including the measured temperature as mentioned in the first chapter. Therefore the model must be tuned to the current measuring conditions. Furthermore, in advanced measurement systems, dynamic correction is only one of many tasks that must be performed in real-time mode. To fulfill the above requirements and stay flexible the signal processor or microprocessor can be used, instead of a digital filter. The signal correction task is then taken by a software routine.

The method described in this chapter, as authors proposal, allows the compensator parameters to be determined for any number of coefficients. It consists of Taylor's series expansion of the input and output signals of the compensator at forthcoming time instants $k = 1, 2,...m$ (n):

$$y(\tau - k \cdot h) = y(\tau) + \sum_{i=1}^{M} (-1)^i \frac{k^i \cdot h^i}{i!} \frac{d^i y(\tau)}{d\tau^i}$$
$$y_K(\tau - k \cdot h) = y_K(\tau) + \sum_{i=1}^{N} (-1)^i \frac{k^i \cdot h^i}{i!} \frac{d^i y_K(\tau)}{d\tau^i}$$
$$(14)$$

The differentiability of these signals is assumed. Making the fusion of equations (13) and (14) we obtain the system of equations (15) resulted from the comparison of equations (12) and (13):

$$a_0 + a_1 + ... + a_m = A'_0 \cdot (1 + b_1 + ... + b_n)$$
$$-h \cdot a_1 - 2h \cdot a_2 - ... - mh \cdot a_m = A'_1 \cdot (1 + b_1 + ... + b_n)$$
$$\frac{h^2}{2} \cdot a_1 + \frac{2^2 h^2}{2} \cdot a_2 + ... + \frac{m^2 h^2}{2} \cdot a_m = A'_2 \cdot (1 + b_1 + ... + b_n)$$
$$\vdots$$
$$(-1)^m \frac{h^m}{m!} \cdot a_1 + (-1)^m \frac{2^m h^m}{m!} \cdot a_2 + ... + (-1)^m \frac{m^m h^m}{m!} \cdot a_m =$$
$$= A'_m \cdot (1 + b_1 + ... + b_n)$$
$$\vdots$$
$$-h \cdot b_1 - 2h \cdot b_2 - ... - nh \cdot b_n = B'_1 \cdot (1 + b_1 + ... + b_n)$$
$$\vdots$$
$$(-1)^n \frac{h^n}{n!} \cdot b_1 + (-1)^n \frac{2^n h^n}{n!} \cdot b_2 + ... + (-1)^n \frac{n^n h^n}{n!} \cdot b_n =$$
$$= B'_n \cdot (1 + b_1 + ... + b_n)$$
$$(15)$$

It is convenient to use a matrix form (16) as more useful for further calculations:

$$\mathbf{X} \cdot \mathbf{W} = \mathbf{Y} \qquad (16)$$

where:

X - generalized matrix of time instants,
Y - vector of coefficients of input differential equation,
W - the vector of estimated coefficients of the difference equation.

The **W** and **Y** vectors are defined as follows:

$$\mathbf{W} = [a_0, ..., a_m, b_1, ..., b_n]^T \qquad (17)$$
$$\mathbf{Y} = [A'_0, ..., A'_m, B'_1, ..., B'_n]^T \qquad (18)$$

The **X** matrix is defined in two steps. For indices $i = 0, 1,..., m$; $j = 0, 1, ..., m$ we have:

$$x_{0,0} = 1 \quad \text{and} \quad x_{i,j} = (-1)^i \frac{j^i \cdot h^i}{i!} \quad \text{for } i \neq 0 \text{ and } j \neq 0 \qquad (19a)$$

If $n > 0$ then there exist elements of the matrix **X** expressed by formulae (19b) and (19c). Accordingly, for $i = 0, 1,..., m$ and $j = m+1, m+2, ..., m+n$ we have:

$$x_{i,j} = -y_i \qquad x_{j,i} = 0 \qquad (19b)$$

and

$$x_{i,j} = (-1)^{i-m} \frac{(j-m)^{i-m} \cdot h^{i-m}}{(i-m)!} - y_i \qquad (19c)$$

for indices $i = m + 1, m + 2,..., m + n$; $j = m + 1, m + 2,...,m+n$. In formula (19) it are denoted: x - element of matrix **X**, y - element of the vector Y. The algorithm for determining the above coefficients of a difference equation consists of several steps. The input values are the discretization step h and the vector of coefficients of the differential equation **Y**. Based on these inputs, a matrix **X** is generated. The next step is solving the set of equations expressed by (16). It is proposed to use the Gauss elimination method because it reduces the matrix **X** to a triangular matrix. Solving a system of equations with a triangular matrix does not pose serious algorithmic problems and is not too much time consuming.

B. Discrete model of the sensor dynamics

By making slight modifications to the method presented above, a discretization of the sensor model can also be carried out. For this purpose, equation (9) is transformed to the form related to the sensor output signal:

$$t_T(\tau) = B'_N \frac{d^N t(\tau)}{d\tau^N} + \cdots + B'_1 \frac{dt(\tau)}{d\tau} + B'_0 t(\tau)$$
$$-A'_M \frac{d^M t_T(\tau)}{d\tau^M} - \cdots - A'_1 \frac{dt_T(\tau)}{d\tau} \qquad (20)$$

where: $A_0 \neq 0$, $A'_i = A_i/A_0$, $i = 1, 2,...,M$,

$$\text{and} \quad B'_j = B_j/A_0, j = 0, 1,..., N \qquad (21)$$

A difference equation of order m can be defined for discrete-time:

$$y_T(\tau) = b_0 y(\tau) + b_1 y(\tau - h) + \cdots + b_n y(\tau - n \cdot h)$$
$$-a_1 y_T(\tau - h) - \cdots - a_m y_T(\tau - m \cdot h) \qquad (22)$$

Using formula (14), the system of equations (23) is obtained to which the matrix notation (16) also applies.

$$b_0 + b_1 + ... + b_n = B'_0 \cdot (1 + a_1 + ... + a_m)$$
$$-h \cdot b_1 - 2h \cdot b_2 - ... - nh \cdot b_n = B'_1 \cdot (1 + a_1 + ... + a_m)$$
$$\frac{h^2}{2} \cdot b_1 + \frac{2^2 h^2}{2} \cdot b_2 + ... + \frac{n^2 h^2}{2} \cdot b_n = B'_2 \cdot (1 + a_1 + ... + a_m)$$
$$\vdots$$
$$(-1)^n \frac{h^n}{n!} \cdot b_1 + (-1)^n \frac{2^n h^n}{n!} \cdot b_2 + ... + (-1)^n \frac{n^n h^n}{n!} \cdot b_n = B'_n \cdot (1 + a_1 + ... + a_m)$$
$$\vdots$$
$$-h \cdot a_1 - 2h \cdot a_2 - ... - mh \cdot a_m = A'_1 \cdot (1 + a_1 + ... + a_m)$$
$$\vdots$$
$$(-1)^m \frac{h^m}{m!} \cdot a_1 + (-1)^m \frac{2^m h^m}{m!} \cdot a_2 + ... + (-1)^m \frac{m^m h^m}{m!} \cdot a_m = A'_m \cdot (1 + a_1 + ... + a_m)$$

$$(23)$$

In this case, the **W** and **Y** vectors are defined /as follows:

$$\mathbf{W} = [b_0 \, ... \, b_n, a_1 \, ... \, a_m]^T \qquad (24)$$

$$\mathbf{Y} = [B'_0 \, ... \, B'_n, A'_1 \, ... \, A'_m]^T \qquad (25)$$

The elements of the matrix **X** are defined similarly as in the case of compensator, but the indices m and n must be interchanged. Hence, for indices $i = 0, 1,..., n; \; j = 0, 1,..., n$ ones obtains:

$$x_{0,0} = 1 \; \text{ and } \; x_{i,j} = (-1)^i \frac{j^i \cdot h^i}{i!} \; \text{ for } i \neq 0 \text{ and } j \neq 0 \qquad (26a)$$

If $m > 0$, then there are elements of the matrix **X** expressed by formulae (26b) and (26c). Correspondingly, for indices $i = 0, 1,..., n; \; j = n + 1, n + 2,..., n + m$ are assumed:

$$x_{i,j} = -y_i \qquad x_{j,i} = 0 \qquad (26b)$$

and

$$x_{i,j} = (-1)^{i-n} \frac{(j-n)^{i-n} \cdot h^{i-n}}{(i-n)!} - y_i \qquad (26c)$$

for indices $i = n+1, n+2,..., n+m; \; j = n+1, n+2,..., n+m$.

For quick implementation of this method two MATLAB scripts were prepared based on basic functions and arithmetic operations that makes it easy to adapt to the other languages like C/C++ or Python. It is fully delivered in Appendix for sensor and compensator models as separately codes.

IV. EXAMPLES OF APPLICATION

Example I

Let us assume that the dynamics of any sensor is described in the frequency domain by the following linear model:

$$G_T(s) = \frac{2s+1}{4s+1} \qquad (27)$$

and the sampling frequency is 20 Hz, i.e. the discretization step h=0.05s. For comparison, the discrete form of the compensator will be determined by different methods. The model obtained by zero-pole matching and also bilinear methods are:

$$G_K(z^{-1}) = \frac{1.988 - 1.963z^{-1}}{1 - 0.9753z^{-1}} \qquad (28)$$

The author's method presented in this paper led to the equation with similar but slight different coefficients:

$$G_K(z^{-1}) = \frac{1.976 - 1.951z^{-1}}{1 - 0.9756z^{-1}} \qquad (29)$$

Figure 1 shows the Bode characteristics for discrete transfer functions (28) and (29) compared with continuous-time model as reciprocal of equation (27). All methods returned similar results. The gain of the DC component is the same in all cases, and the differences in discrete zeros and poles are negligibly small. The phase characteristic shows slight differences only for the limit frequency 1/(2·h)=10 Hz or $\pi/h \approx 63$ rad/s.
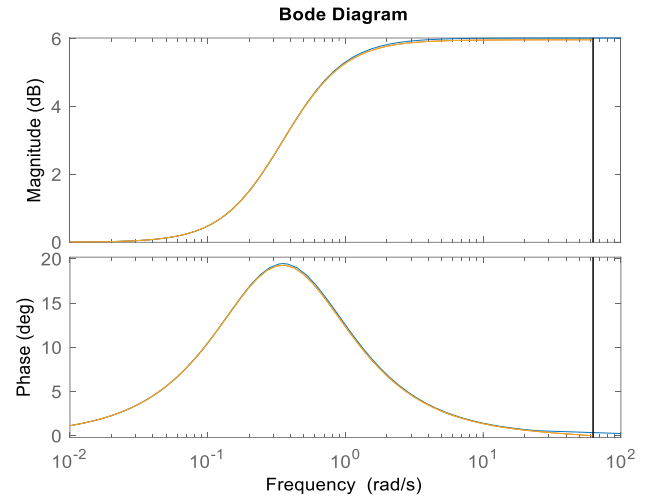

Fig.1. Comparison of the Bode characteristics of the continuous and discrete-time compensators

Example II

Let us assume that the dynamics of any sensor is described in the frequency domain by the following linear model:

$$G_T(s) = \frac{1}{4s+1} \qquad (30)$$

and the discretization step is h=0.01s. The sensor input signal is periodic and constructed as a sum of three cosine functions with different frequencies 10, 20 and 50 rad/s and different max. amplitudes. The white noise is added to simulate the random measuring errors. The undesired property of compensator is strengthen the amplitude of noise, therefore we decided to impose small inertia in model of compensator for noise reducing. The drawback of this solution is additional phase shift of the compensator output signal. The compensator model in continuous-time is as:

$$G_K(s) = \frac{4s+1}{0.1s+1} \qquad (31)$$

Applying the invented method again for this case we have:

$$G_K(z^{-1}) = \frac{36.45 - 36.36z^{-1}}{1 - 0.9091z^{-1}} \qquad (32)$$

The Fig. 2 presents an analyzed signal path in Simulink tool, i.e. sensor input, sensor output and compensator output.
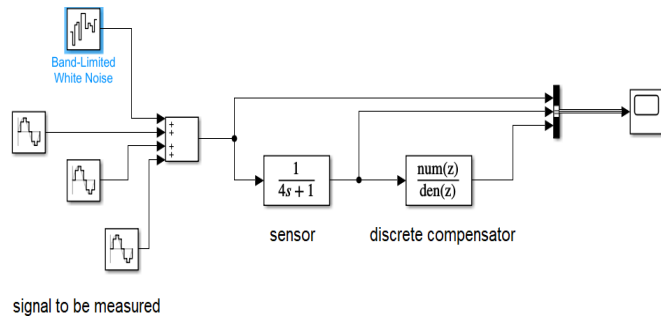
Fig.2. Simulated signal path in the Simulink tool

The results of the simulation are presented in Fig. 3. There can be noticed significant difference between measured signal and that one aimed to be measured. The sensor output signal is strongly attenuated by its dynamics. The compensator output signal is very close to the input. The output noises are at an acceptable level and the above mentioned phase shift is also visible. The axis units and numbers are not essential hence not presented in Fig. 3.
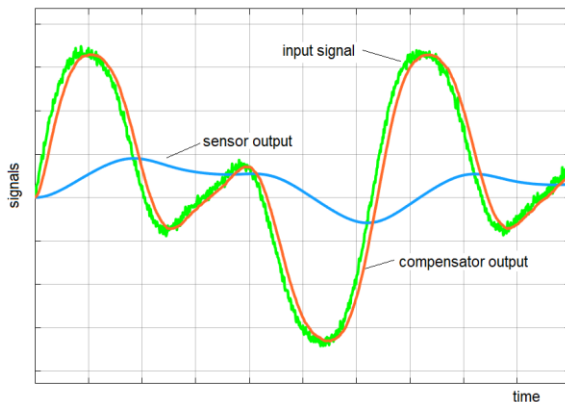

Fig.3. Simulated signals in time using Simulink tool

CONCLUSION

The advantage of the presented method is, above all, the possibility of implementing the algorithm for the discretization of continuous models in popular programming languages, e.g. Python, C/C++, Java, or any dedicated mathematical packages, i.e. Mathcad, Mathematica. The quality of the discrete model obtained by this method for considered cases is similar to that of the method with a representation of zeros and poles but full properties and limitations should be investigated. Obviously, there are no limits to the number of differential equation coefficients. This is especially important when the sampling rate is low, leading to significant errors in the discrete model. The zero-pole matched method requires zero and pole transfer function form as an input. For models of higher orders, often with complex zeros and poles, this may constitute an additional computational difficulty for the user. The proposed method is working with a classical form of transfer function instead.

ACKNOWLEDGMENTS

REFERENCES

[1] M. Boumahdi, "Blind identification using the kurtosis with applications to field data", Elsevier Signal Processing 48 (1996) pp. 205-216, https://www.sciencedirect.com/science/article/pii/0165168495001360.

[2] S. Brohez, C. Delvosalle, G. Marlair, "A two-thermocouples probe for radiation corrections of measured temperature in compartment fires", Elsevier Fire Safety Journal 39 (2004) pp. 399-411, https://www.sciencedirect.com/science/article/pii/S037971120400027X.

[3] P. D. Gillespie, P. C. Hung, R. J. Kee, S. F. McLoone, "Blind characterisation of sensors with second-order dynamic response", IFAC-PapersOnLine 48-28 (2015) pp. 484-489 International Federation of Automatic Control Hosting by Elsevier Ltd., https://www.sciencedirect.com/science/article/pii/S2405896315027998.

[4] K. Kar, A. K. Swain, R. Raine, "Identification of Time-Varying Thermocouple Systems" Conference TENCON October 2007, https://www.researchgate.net/publication/4310652_Identification_of_time-varying_thermocouple_systems.

[5] W. Minkina, S. Gryś, "Korekcja charakterystyk dynamicznych czujników termometrycznych - metody, układy, algorytmy" ("Compensation of dynamic characteristics of thermometric sensors - methods, systems, algorithms"), Publishing house Wydawnictwo Politechniki Częstochowskiej, Częstochowa, Poland, 2004, 116g pp., ISBN 83-7193-243-X, https://www.researchgate.net/publication/360861645_Korekcja_charakterystyk_dynamicznych_czujnikow_termometrycznych_metody_uklady_algorytmy.

[6] W. Minkina, "Theoretical and experimental identification of the temperature sensor unit step response non-linearity during air temperature measurement" Elsevier, Sensors and Actuators A: Physical A 78 (1999) pp. 81–87, https://www.sciencedirect.com/science/article/pii/S09244247 9900206X.

[7] S. Grys, W. Minkina, "Fast temperature determination using two thermometers with different dynamical properties", Elsevier, Sensors and Actuators A: Physical A 100 (2002) pp. 192-198, https://www.sciencedirect.com/science/article/pii/S0924424702000602.

[8] J. Nabielec, "A "blind" correction of dynamic error of a nonstationary first order transducer for the periodic case-simulation investigation", XVIII IMEKO WORLD CONGRESS, Metrology for a Sustainable Development, September, 17 – 22, 2006, Rio de Janeiro, Brazil, https://www.imeko.org/publications/wc-2006/PWC-2006-TC4-018u.pdf.

[9] J. Nalepa, "The "blind" dynamic error correction method – simulation study for the first – and second – order measurement channel", Measurement Science Review, 2002, Vol. 2, section 1, https://www.measurement.sk/PAPERS/Nalepa.pdf.

[10] A. Olczyk, "Measurements of unsteady flow parameters in pipe-receiver and pipe-turbocharger systems" Metrology and Measurement Systems, https://www.researchgate.net/publication/298829711_MEASUREMENTS_OF_UNSTEADY_FLOW_PARAMETERS_IN_PIPE-RECEIVER_AND_PIPE-TURBOCHARGER_SYSTEMS.

[11] K. Rupnik, J. Kutin, I. Bajsic, "Identification and prediction of the dynamic properties of resistance temperature sensors", Elsevier, Sensors and Actuators A: Physical A 197 (2013) pp. 69-75, https://www.sciencedirect.com/science/article/pii/S0924424713001568.

[12] K. Kar, A. Swain, R. Raine, S. Roberts, R. Stone, "Cycle-by-cycle variations in exhaust temperatures using thermocouple compensation techniques" 2005 SAE International, https://doi.org/10.4271/2006-01-1197.

[13] B. Saggin, S. Debei, M. Zaccariotto, "Dynamic error correction of a thermometer for atmospheric measurements" Elsevier Measurement 30 (2001) pp. 223–230, https://www.sciencedirect.com/science/article/pii/S026322410100015X.

[14] M. Tagawa, Y. Ohta, "Two-thermocouple probe for fluctuating temperature measurement in combustion-rational estimation of mean and fluctuating time constans", Elsevier Science Inc. Combustion and Flame 109 (1997) pp. 549-560, https://www.sciencedirect.com/science/article/pii/S0010218097000448.

[15] W. W. Tan, J. Ch. Tang, A. P. Loh, A. Tay, "In situ measurement of wafer temperature using two sensors with different dynamical properties", Measurement Science and Technology 17 (2006) 2957–2963, http://dx.doi.org/10.1088/0957-0233/17/11/014.

[16] K. J. Åström, B. Wittenmark, "Computer-controlled systems. Theory and design", Prentice Hall, 1990, http://www.ioe.nchu.edu.tw/Pic/CourseItem/4687_Computer_Controlled_Systems_3rd_Part1.pdf.

[17] G. F. Franklin, J. D. Powell, M. L. Workman, "Digital control of dynamic systems", 2nd ed., Addison-Wesley 1990,

https://www.researchgate.net/publication/31849881_Digital_Control_of_Dynamic_Systems-Third_Edition.

[18] J.O. Smith, "Impulse invariant method", Physical Audio Signal Processing, August 2007, https://www.dsprelated.com/dspbooks/pasp/Impulse_Invariant_Method.html.

[19] T. Laakso, V. Valimaki, "Splitting the unit delay", IEEE Signal Processing Magazine, Vol. 13, No. 1 (1996), pp. 30-60.

APPENDIX – METHOD IMPLEMENTED AS A MATLAB SCRIPTS

```
% discrete model of the sensor;
% G(s)=Y(s)/X(s)=(...+B1ps+Bp0)/(...+Ap1s+Ap0) – transfer function of sensor
% inputs: [Bp], [Ap], h,
% where h – discretization step
% G(z)=Y(z)/X(z)=(b0+b1z^-1+...)/(1+a1z^-1+...);
% outputs: [b],[a];
A=[]; B=[]; A1=[]; B1=[]; x=[]; M=0; K=0; a=[]; b=[]; a_b=[]; C=[]; C1=[];
        %h=0.1;      <- only as an example
        %Bp=[1]      <- please uncomment
        %Ap=[10 1]  <-
A=Ap/Ap(length(Ap));
B=Bp/Ap(length(Ap));
K=length(A)-1;
M=length(B)-1;
B1=fliplr(B);
for i=1:K
 A1(i)=A(i);
 end;
 A1=fliplr(A1);
C1=[B1 A1];
C=C1';
% the first quadrant of the coordinate system
for i=0:M
 if i==0 sil=1; else sil=sil*i; end;
 for j=0:M
 x(i+1,j+1)=((-1)^i)*(h^i)*(j^i)/sil;
 end;
end;
if K==0 a_b=inv(x)*C; a_b
 elseif K>0 ;
 else
% bad inputs!
end
% the second and third quadrants of the coordinate system
for i=0:M
 for j=M+1:M+K
   x(i+1,j+1)=-C(i+1);
   x(j+1,i+1)=0;
 end;
end;
% the fourth quadrant of the coordinate system
 sil=1;
for i=M+1:M+K
 sil=sil*(i-M);
 for j=M+1:M+K
 x(i+1,j+1)=((-1)^(i-M))*(h^(i-M))*((j-M)^(i-M))/sil-C(i+1);
 end;
end;
a_b=inv(x)*C
for i=0:M
  b(i+1)=a_b(i+1);
end;
for i=M+1:M+K
 a(i-M+1)=a_b(i+1);
end
```

```
a(1)=1;
sysz=tf(b,a,h)
bode(sysz);
%-------------- end of script ----------------

% discrete model of the compensator;
% G(s)=Y(s)/X(s)=(...+B1ps+Bp0)/(...+Ap1s+Ap0) – transfer function of sensor
% inputs: [Bp], [Ap], h
% where h – discretization step
% G(z)=X(x)/Y(z)=(a0+a1z^-1+...)/(1+b1z^-1+...) – transfer function
% of the compensator
% output: [a],[b];
A=[]; B=[]; A1=[]; B1=[]; x=[]; M=0; K=0; a=[]; b=[]; a_b=[]; C=[]; C1=[];
        %h=0.1;      <- only as an example
        %Bp=[1]      <- please uncomment
        %Ap=[10 1]  <-
A=Ap/Bp(length(Bp));
B=Bp/Bp(length(Bp));
K=length(B)-1;
M=length(A)-1;
A1=fliplr(A);
for i=1:K
B1(i)=B(i);
end;
B1=fliplr(B1);
C1=[A1 B1];
C=C1';
% the first quadrant of the coordinate system
for i=0:M
 if i==0 sil=1; else sil=sil*i; end;
 for j=0:M
 x(i+1,j+1)=((-1)^i)*(h^i)*(j^i)/sil;
 end;
end;
if K==0 a_b=inv(x)*C; a_b
 elseif K>0 ;
 else
%   bad inputs!
end
   % the second and third quadrants of the coordinate system
for i=0:M
 for j=M+1:M+K
   x(i+1,j+1)=-C(i+1);
   x(j+1,i+1)=0;
 end;
end;
 % the fourth quadrant of the coordinate system
 sil=1;
for i=M+1:M+K
 sil=sil*(i-M);
 for j=M+1:M+K
 x(i+1,j+1)=((-1)^(i-M))*(h^(i-M))*((j-M)^(i-M))/sil-C(i+1);
 end;
end;
a_b=inv(x)*C
for i=0:M
  a(i+1)=a_b(i+1);
end;
for i=M+1:M+K
 b(i-M+1)=a_b(i+1);
end
b(1)=1;
sysz=tf(a,b,h)
bode(sysz);
%-------------- end of script ----------------
```