On the Monotonicity and Linearity of Ideal Radix-Based A/D Converters

János Márkus and István Kollár, *Fellow, IEEE*

***Abstract*—Both cyclic and pipelined analog-to-digital (A/D) converters are getting more and more popular, as they are relatively easy to design and either have a high throughput (pipelined converters) or small area- and power-consumption (cyclic/algorithmic converters). To avoid saturation and to ensure effective digital calibration, in the analog stage(s) of these converters, instead of the ideal two, often a smaller nominal gain (called radix number) is used. In this paper, it is shown that these radix-based converters have nonmonotonic output and finite linearity. The causes of these phenomena are discussed in detail. Fully digital method is suggested to remove nonmonotonic code transitions and estimation on the maximum differential nonlinearity of the ideal converter as a function of the number of cycles is presented.**

***Index Terms*—Algorithmic, analog-digital (A/D) conversion, cyclic, differential nonlinearity, DNL, linearity, monotonicity, multistage pipelined, nonradix-2, radix less than 2, subradix ADC, subranging A/D converter.**

I. Introduction

SUB-RANGING analog-to-digital (A/D) converters [1] are getting more and more popular in different applications. In these converters the input signal is quantized first by a coarse (e.g., 1-bit) quantizer, then the analog residue is calculated and requantized either by the same circuit (algorithmic/cyclic converter, [1], Fig. 1) or by another similar stage (subranging/pipelined converter, [1]). Cyclic converters are easy to design and have very low area- and power-consumption. Pipelined converters can exhibit high throughput at medium or high resolution, and are commonly used in high-speed digital communication systems.

Fig. 1 shows the block diagram of a 1-bit/stage cyclic converter. Its operation is as follows. In the first cycle, the input signal is sampled by a sample and hold (S/H) circuit, quantized by a 1-bit A/D (i.e., a comparator), the quantized output is converted back to analog again by a D/A and finally, it is subtracted from the input multiplied by the radix number g. The value of g is equal to two in the ideal case. In the next cycle, this residue is used as an input signal to obtain the second Most Significant Bit (MSB), and so on, up to n. The output of the converter is the sequence of the one-bit d_{n-i}s, which is the binary representation of the input signal in the ideal case (d_0 is the Least Significant Bit (LSB) of the converter).

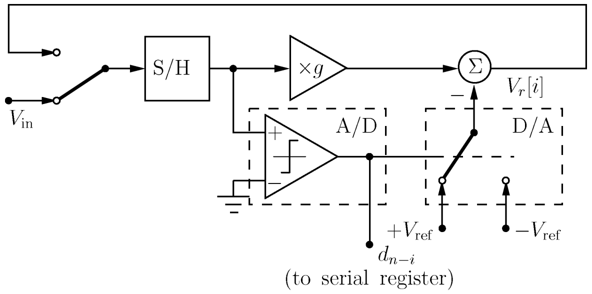


Fig. 1. Block diagram of an algorithmic converter. V_{in} is the input signal, V_{ref} is the reference signal, $V_r[i]$ is the residue signal in the ith cycle, g is the radix number equal to two in the ideal case, and d_{n-i} is the 1-bit digital output.

Due to the finite precision of the analog components, circuit nonideality errors affect the accuracy of the converter. The most important error source is the capacitor mismatch which causes g to become inaccurate. As a result, two types of error can occur [2], [3]. If $g > 2$, at specific inputs at least one stage will be saturated, causing missing-decision-level error (i.e., the output does not change for a wide range of the input signal). If $g < 2$, codes will be missing in the output (i.e., the output jumps larger than one LSB at a code transition). As in the digital domain there is no information about the range of the input signal causing the same output code, missing-decision-level errors must be avoided [2]. This can be ensured generally two ways: the first alternative is to use a nominal $g < 2$ in the circuit to make sure it will not turn over 2 even in a worst-case mismatch error, and use more stages to compensate for the resolution loss (*analog redundancy* [2], [4]). The second alternative is to use more than one bit/stage in the converter, and still use a nominal gain $g = 2$ (*digital redundancy*, [3]). Both of these techniques can be later compensated in the digital domain, and they can also be transformed into each other.

In the following, the 1-bit/stage converters with $g < 2$ [2], [4] are examined in details. The paper is organized as follows. Section II discusses the operation of a radix-based converter in detail, then the nonmonotonic behavior of the converter is examined. Section III discusses the linearity error introduced by the structure. The paper is finished with some concluding remarks.

II. Monotonicity of Radix-Based Converters

In the case of a cyclic converter with $g < 2$, $|V_{\text{in}}| < V_{\text{ref}}$, and $d_{n-i} \in \{0, 1\}$, the residues during the conversion can be obtained as follows (cf. Fig. 1):

$$\begin{aligned} V_r[1] &= gV_{\text{in}} - (2d_{n-1} - 1)V_{\text{ref}} \\ V_r[2] &= gV_r[1] - (2d_{n-2} - 1)V_{\text{ref}} = \end{aligned}$$

Manuscript received June 15, 2004; revised July 25, 2005. This work was supported in part by the Hungarian Fund for Scientific Research, under Contract OTKA T 033 053 and OTKA TS 049 743.

The authors are with the Department of Measurement and Information Systems, Budapest University of Technology and Economics, H-1521 Budapest, Hungary (e-mail: markus@mit.bme.hu;kollar@mit.bme.hu).

Digital Object Identifier 10.1109/TIM.2005.858104

TABLE I
CODE TRANSITIONS OF A 12-CYCLE CONVERTER WITH $g = 1.95$

n	Code Transition	Step size in LSB_{12}
#0	xxxx xxxx xxx0 → xxxx xxxx xxx1	1
#1	xxxx xxxx xx01 → xxxx xxxx xx10	0.95
#2	xxxx xxxx x011 → xxxx xxxx x100	0.853
#3	xxxx xxxx 0111 → xxxx xxxx 1000	0.662
#4	xxxx xxx01111 → xxxx xxx1 0000	0.292
#5	xxxx xx011111 → xxxx xx10 0000	-0.431
#6	xxxx x0111111 → xxxx x100 0000	-1.841
⋮	⋮	⋮
#11	0111 1111 1111 → 1000 0000 0000	-80.53

$$= g^2 V_{\mathrm{in}} - g(2d_{n-1} - 1)V_{\mathrm{ref}} - (2d_{n-2} - 1)V_{\mathrm{ref}}$$

$$V_r[n] = g^n V_{\mathrm{in}} - g^{n-1}(2d_{n-1} - 1)V_{\mathrm{ref}} - \cdots - (2d_0 - 1)V_{\mathrm{ref}} = = g^n \left(V_{\mathrm{in}} - \sum_{i=1}^{n} g^{-i}(2d_{n-i} - 1)V_{\mathrm{ref}} \right) \quad (1)$$

where V_{in} is the input signal, V_{ref} is the reference voltage of the converter, $V_r[i]$ is the residue of the ith conversion cycle, d_{n-i} is the output of the comparator, and n is the number of cycles the converter operates. From (1), one can get

$$\frac{V_{\mathrm{in}}}{V_{\mathrm{ref}}} = \sum_{i=1}^{n} g^{-i}(2d_{n-i} - 1) + \epsilon, \qquad |\epsilon| \le \frac{1}{g^n} \quad (2)$$

where ϵ is the quantization error. Rearranging (2) leads to

$$V_{\mathrm{in}} = V_{\mathrm{LSB}} \left(\frac{1}{2} \sum_{i=1}^{n} g^{n-i}(2d_{n-i} - 1) + \epsilon' \right) \quad (3)$$

where $V_{\mathrm{LSB}} = 2V_{\mathrm{ref}}/g^n$ and $|\epsilon'| = |\epsilon g^n/2| \le 0.5$. Thus, replacing $j = n - i$ and ignoring the offset caused by the $\{0, 1\}$ representation of the d_j sequence, the digital output can be calculated as

$$D = \sum_{j=1}^{n} g^j d_j. \quad (4)$$

A. Output Calculation

Calculating (4) for all possible input codes would exhibit large jumps at major code transitions, resulting in nonequivalent mapping from the analog input to the digital codes. For example, for an input code of 0111 1111 1111, with $g = 1.95$, $D = \sum_{j=0}^{n-2} g^j = 1630.710$, while for the input code of 1000 0000 0000, $D = g^{n-1} = 1550.175$, resulting in a huge negative jump of 80 LSBs at the MSB transition.

However, as discussed in the introduction, an algorithmic converter with $g < 2$ will exhibit missing codes. Moreover, it was proven in (3) that (due to the negative feedback) the operation of the cyclic converter ensures that the absolute difference between the input signal and its *radix-based* digital representation will always be less than half LSB. Although this behavior ensures that the difference between two consecutive digital output is less than or equal to one LSB, it allows this difference to be negative, thus, it allows nonmonotonic behavior. Table I shows the major code transitions of a cyclic converter with $g = 1.95$. (Note that the major code transition $\#k$ can be calculated as $g^k - \sum_{i=0}^{k-1} g^i = g^k - (g^k - 1)/(g - 1)$.) It can be seen that code transition #5 is negative and its absolute value is less than one LSB. It can be proven that there always exists such a code transition for any $g \in (1, 2)$ [5].

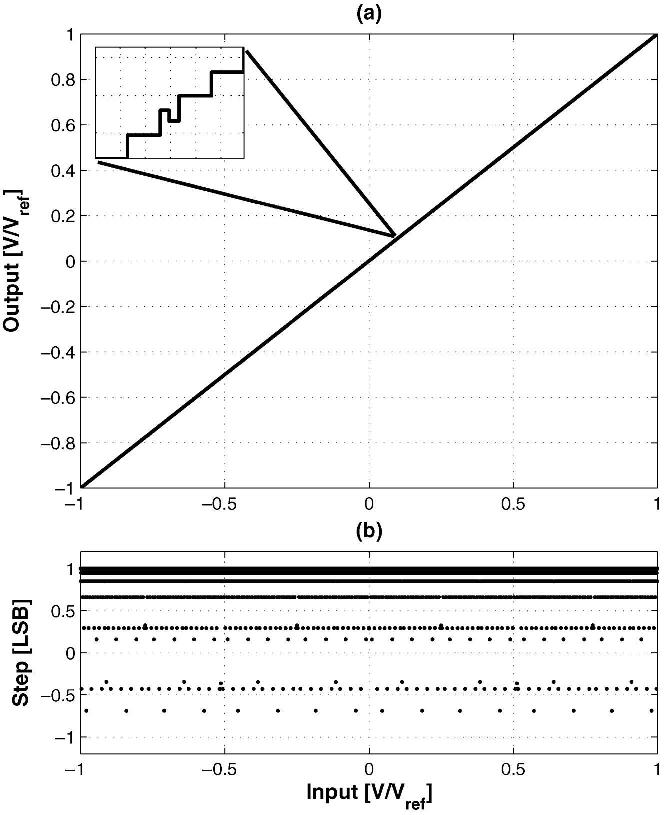


Fig. 2. (a) Output of a 12-cycle ideal radix-based converter. The inset shows a detailed transition, which is nonmonotonic. (b) Steps of the output code transitions. Steps less than 0 correspond to nonmonotonic code transitions.

Based on this derivation, it is expected that an ideal radix-based converter with $g = 1.95$ and $n = 12$ produces major code transitions #0 to #5, but no major code transitions #6 and above. Due to the missing codes, other (positive or negative) step sizes less than one LSB may also be produced. This is illustrated in Fig. 2. In Fig. 2(a) one can see the smooth static characteristic of the converter, however, the zoomed inset shows one of the nonmonotonic jumps in the output (the code transition is $1000\,1101\,1111 \rightarrow 1000\,1110\,0000$, containing the major code transition of #5). Fig. 2(b) illustrates the step-sizes between the codes. According to Table I, most steps belongs to the code transitions #0–#5, however, due to the missing codes, other step sizes exist as well. For example, the most negative step, -0.69 belongs to the code transition $\mathrm{xxxx}\,0111\,1101 \rightarrow \mathrm{xxxx}\,1000\,0010$ which is not a major code transition, but similarly to the negative code transition #5, five consecutive ones turn to five consecutive zeros in it. The difference is that the algorithm uses the available LSBs to get a better representation of the input signal.

B. Requantization

In the discussion earlier, calculation of the output code [(4)] was assumed to be infinitely precise. In a real hardware the output code is calculated, then requantized to $n_{\mathrm{bit}} < n$ bit,

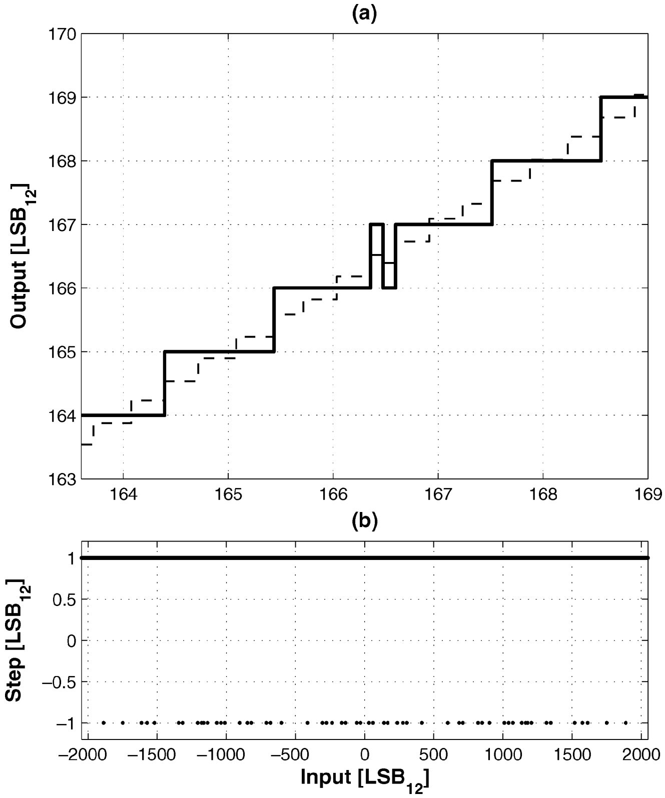


Fig. 3. (a) Enlarged output of a 14-cycle 12-bit converter (dashdot line: 14-cycle output code, solid line: 12-bit requantized binary output code), showing a nonmonotonic code transition. (b) Steps of all output 12-bit binary code. Steps equal to -1 correspond to nonmonotonic output transitions.

TABLE II
NUMBER OF NON-MONOTONIC TRANSITIONS IN A 14-CYCLE, 12-BIT CONVERTER WITH $g = 1.95$

# of g-based codes	13101
# of g-based non-monotonic transition	326
Wrong g-based codes (% of input range)	1.49%
# of 2-based codes	4095
# of 2-based non-monotonic transition	56
Wrong 2-based codes (% of input range)	0.84%

which is the target resolution of the converter [4]. As usually $n_{\mathrm{bit}} \leq n - 2$, the final LSB size will be 2–3 times larger than the step size of the radix-based converter. Thus, most of the nonmonotonic transitions will be smoothed out by the requantization process. However, due to the large number of nonmonotonic transitions and the fact that the step-sizes of the radix-based converter are uneven, there will always be some transitions which crosses one of the quantization thresholds, causing nonmonotonic behavior also in the final output.

This is illustrated in Fig. 3 and Table II. Fig. 3(a) shows an enlarged nonmonotonic step in the final requantized output of a 12-bit, 14-cycle, $g = 1.95$ converter, while Fig. 3(b) shows all the steps of the 12-bit output, where step size -1 corresponds to a nonmonotonic jump. Table II shows the statistics of the same example, showing that in the final output about 1% of the input signal will be mapped wrongly.

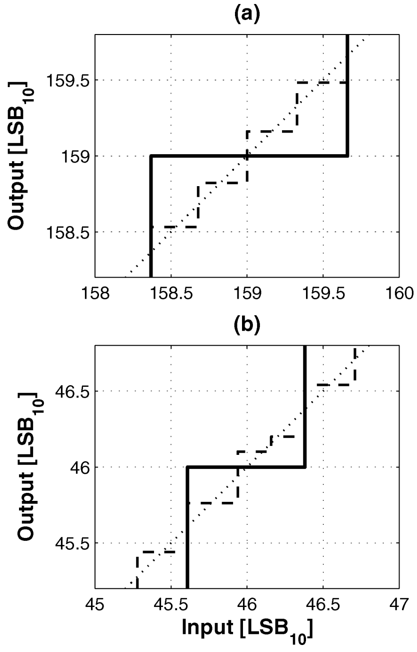


Fig. 4. (a) An example, when the differential nonlinearity $\mathrm{DNL} > 0$ in a 12-cycle 10-bit converter. (b) An example, when $\mathrm{DNL} < 0$ (dotted line: input signal, dashdot line: 12-cycle output code, solid line: 10-bit requantized binary output code).

If this behavior is not acceptable, the following method can be used to avoid nonmonotonicity: It was shown previously that this type of transition occurs only if a given n_0 consecutive ones changes to n_0 consecutive zeros in the output code. This n_0 can be calculated from g [5], e.g., if $g = 1.95$, $n_0 = 5$. Therefore, with a simple digital hardware subtracting one from any code containing n_0 consecutive ones and adding one to any code containing n_0 consecutive zeros before the requantization process removes all nonmonotonic transitions from the digitally calibrated output code. Note that analog noise in the converter may cover up the nonmonotonic code transitions if the noise level is high enough [5].

III. LINEARITY OF RADIX-BASED CONVERTERS

As it was discussed in the previous section, the step-sizes of a radix-based converter with $g < 2$ are not equal (cf. Table I), meaning that the input range for each 12-bit radix-based code may differ. As the final output is requantized from these data, the converter will map the input signal unevenly even to the final output digital codes. Fig. 4 shows two examples of this effect. Fig. 4(a) shows an example when the differential nonlinearity (DNL) is greater than zero, i.e., the analog input range mapped into one digital code is greater than one V_{LSB}, while Fig. 4(b) shows an example when the DNL is less than zero, i.e., the analog input range mapped into one digital code is less than one V_{LSB}.

To estimate the DNL of the converter, the following method is proposed. As it can be seen in Fig. 4, by adding the step-sizes of consecutive g-based code transitions of the 12-cycle converter, one can get an estimate about the input range of the 10-bit requantized output code. For example, in Fig. 4(b) the final output code 46 consists of the g-based steps $1000\,1100\,1101 \rightarrow \ldots 00\,1110$, $\ldots 00\,1110 \rightarrow \ldots 00\,1111$, and $\ldots 00\,1111 \rightarrow \ldots 01\,0000$. These code transitions in the

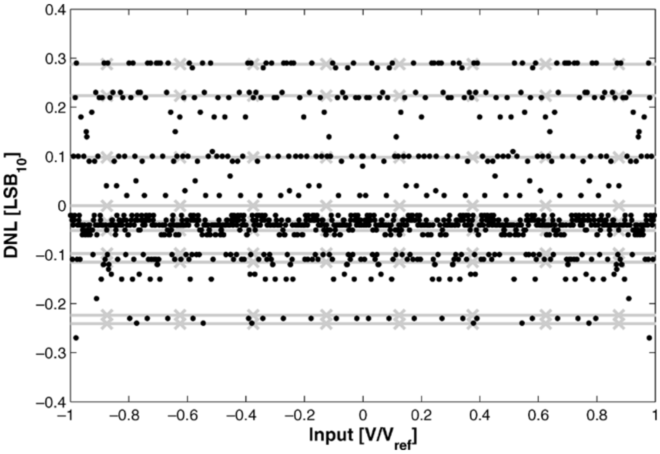


Fig. 5. DNL error of a simulated ideal converter, represented with dots. Solid gray lines with ×-markers show the estimated levels of the DNL error (cf. Table III).

TABLE III
MULTIPLE 12-BIT CODE TRANSITIONS CAUSING ONE 10-BIT TRANSITION IN THE REQUANTIZED FINAL OUTPUT

Code Transition (cf. Tab.I)	Step size in LSB$_{10}$	DNL in LSB$_{10}$
#0 →#1 →#0	1.00	0.00
#0 →#2 →#0	0.97	-0.03
#0 →#3 →#0	0.90	-0.1
#0 →#4 →#0	0.78	-0.22
#0 →#1 →#0 →#2	1.29	+0.29
#0 →#1 →#0 →#3	1.22	+0.22
#0 →#1 →#0 →#4	1.10	+0.10
#1 →#0 →#2	0.95	-0.05
#1 →#0 →#3	0.88	-0.12
#1 →#0 →#4	0.76	-0.24

g-domain are #1, #0, #4 with step sizes of 0.95, 1 and 0.292 in LSB_{12}, respectively (cf. Table I). The sum of these code transitions have to be converted to the 2-based 10-bit step size. Since 1 g-based LSB_{12} is equivalent to $2/g^{12}V_{\mathrm{ref}}$ and 1 2-based LSB_{10} is equivalent to $2/2^{10}V_{\mathrm{ref}}$, the $2.242 \cdot \mathrm{LSB}_{12}$ is equal to $2.242 \cdot 2^{10}/g^{12} = 0.76\mathrm{LSB}_{10}$, resulting in a DNL error of $-0.26\mathrm{LSB}_{10}$. Based on this observation, Table III shows all possible consecutive 12-cycle code transitions, which are mapped into one 10-bit transition and the DNL error associated with them.

To support these derivations, Fig. 5 shows simulation results of the DNL error of an ideal radix-based converter and compares it with the derived DNL-levels of Table III. Solid gray lines with ×-markers show the estimated DNL errors, while black dots represent the DNL-errors of the simulated converter. Most errors are close to the predicted levels. Those codes, where the error is different from the predicted ones, are 10-bit output codes containing 12-bit code transitions with missing codes, thus they cannot be derived from the regular major code transitions of Table III.

Using Table III, the maximum DNL of this ideal radix-based converter can be estimated as about 0.3 LSB. This DNL is proportional to the ratio of the code width of the n-cycle converter and the target code width of the converter. Thus, increasing the number of cycles in the converter while keeping the target bit number the same will decrease this DNL error to an acceptable level for a desired application.

IV. CONCLUSION

In this paper, properties of radix-based converters were discussed. It was shown that even ideal converters produce nonmonotonic output. A detailed example with nonmonotonic code transitions was discussed and a fully digital method was suggested to eliminate these transitions completely from the output of the converter. Moreover, it was shown that even ideal radix-based converters have limited linearity, which is inherited from the topology, thus it cannot be removed from the system. The expressions derived for the maximum differential nonlinearity for an ideal system can be used to estimate the required number of stages n in a given design, to push the linearity error below the required value of the specification.

REFERENCES

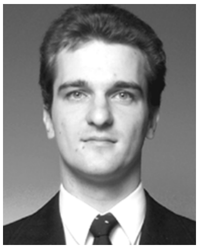
[1] D. Johns and K. Martin, “Nyquist-rate A/D converters,” in *Analog Integrated Circuit Design*, 1st ed. Hoboken, NJ: Wiley, 1996, ch. 13, pp. 487–530.

[2] A. N. Karanicolas, H.-S. Lee, and K. L. Bacrania, “A 15-b 1-Msample/s digitally self-calibrated pipeline ADC,” *IEEE J. Solid-State Circuits*, vol. 28, pp. 1207–15, Dec. 1993.

[3] H.-S. Lee, “A 12-b 600 ks/s digitally self-calibrated pipelined algorithmic ADC,” *IEEE J. Solid-State Circuits*, vol. 29, pp. 509–15, Apr. 1994.

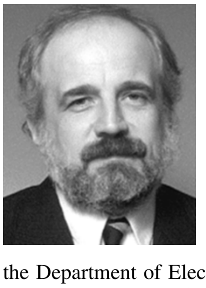
[4] O. E. Erdogan, P. J. Hurst, and S. H. Lewis, “A 12-b digital-background-calibrated algorithmic ADC with -90-dB THD,” *IEEE J. Solid-State Circuits*, vol. 34, pp. 1812–20, Dec. 1999.

[5] J. Márkus and I. Kollár, “On the monotonicity and linearity of ideal radix-based A/D converters,” in *Proc. IEEE Instrumentation and Measurement Technology Conf. (IMTC’04)*, vol. 1, Como, Italy, May 18–20, 2004, pp. 696–701.



János Márkus was born in Tata, Hungary, in 1976. He received the M.S. degree in electrical engineering from Budapest University of Technology and Economics (BUTE), in 1999.

Currently, he is a Senior Lecturer with the Department of Measurement and Information Systems (DMIS), BUTE, finalizing his Ph.D. dissertation. In 2000, he was a design engineer with Panda Audio Ltd. (a former subsidiary of Akai Professional). In 2001 and 2002, he was an exchange visitor with the School of Electrical Engineering and Computer Sciences, Oregon State University, Corvallis. His current research interests include digital signal processing, quantization errors in DSP systems, A/D and D/A converter testing, and delta-sigma data converter structures.



István Kollár (M’87–SM’93–F’97) was born in Budapest, Hungary, in 1954. He received the M.S. degree in electrical engineering from the Technical University of Budapest, Budapest, Hungary, in 1977, and the “Candidate of Sciences” and “Doctor of the Academy,” degrees from the Hungarian Academy of Sciences in 1985 and 1998, respectively.

From 1989 to 1990, and during spring 2005, he was a Visiting Scientist with the Vrije Universiteit Brussel, Belgium. From 1993 to 1995, he was a Fulbright Scholar and Visiting Associate Professor with the Department of Electrical Engineering, Stanford University, Stanford, CA. Currently, he is a Professor of electrical engineering with the Budapest University of Technology and Economics. His research interests are in the areas of digital and analog signal processing, measurement theory, and system identification. He has published approximately 110 scientific papers, and is coauthor of *Technology of Electrical Measurements* (New York: Wiley, 1993). He authored *Frequency Domain System Identification Toolbox for Matlab*.

Dr. Kollár is a member of the Administrative Committee of the IM Society. He is also an active member of EUPAS in IMEKO. He was Editor of *Periodica Polytechnica Ser. Electrical Engineering* from 1988 to 1991, then Chief Editor of *Periodica Polytechnica* from 1991 to 1997. He was technical co-chairman of IMTC/2001.