Postprint of: Czaja Z., A measurement method for lossy capacitive relative humidity sensors based on a direct sensor-to microcontroller interface circuit, MEASUREMENT, Vol. 170 (2021), 108702, DOI: 10.1016/j.measurement.2020.108702

A measurement method for lossy capacitive relative humidity sensors based on a direct sensor-to-microcontroller interface circuit

Zbigniew Czaja

Gdansk University of Technology,
Faculty of Electronics, Telecommunications and Informatics,
Department of Metrology and Optoelectronics,
ul. G. Narutowicza 11/12, 80-233 Gdansk, Poland,
email: zbczaja@pg.edu.pl.

Abstract

A new time-domain measurement method for determining the capacitance and resistance values of lossy relative humidity capacitive sensors is presented. The method is based on a direct sensor-to-microcontroller interface for microcontrollers with internal analog comparators and timers. The interface circuit consists only of four reference resistors (two reference resistors if a microcontroller includes a voltage reference source), a given sensor and a microcontroller. A systematic error correction algorithm based on a correction dictionary and the *M*-multiple measurement approach are also proposed in the paper. Experimental investigations were carried out using a prototype device based on an 8-bit ATXmega32A4 microcontroller. The experimental research confirmed that the relative errors of measurement of capacitance introduced by the interface circuit are less than 0.71% (for capacitance values 100 – 286 pF), and the relative errors of measurement of resistance are less than 0.74% (for resistance values 1 – 10 MΩ).

Keywords: microcontroller interfacing, capacitive sensors, analog comparators, time-domain measurement

1. Introduction

Capacitive sensors, which are widely used for the measurement of many physical and chemical quantities, can be modelled with lossless capacitance C_x (the perfect capacitive sensor) or modelled with shunt resistance R_x in parallel with C_x (for a lossy capacitive sensor). In the first case, inter alia, these are humidity sensors [1–5], pressure sensors [6,7], position sensors [8], surface-force sensors [9], carbon dioxide sensors [10,11], sensors used to measure the flow of fluids [12] and flour biomass materials [13] through pipelines, and sensors for monitoring the level of liquids [14], as well as the water content in oil [14,15], and soil moisture content [16].

The second group, that is, the lossy capacitive sensors, includes sensors, e.g. to measure the conductivity of NaCl solutions [17], to assess the degradation of the quality of frying oil [18] and permittivity values of edible oil [19], humidity sensors [20,21] and humidity sensors subjected to condensation [22]. For these sensors R_x can play an important role. This value is usually not constant [23], but it can change with the measurand, depends on environmental factors (such as temperature, condensation and pollution) and drifts with time. Consequently, the effects of R_x must be corrected at electronics level or software level.

For these sensors, various interface circuits have been proposed in the literature. For example, in [24] the interface circuit is based on a modified De-Sauty active bridge, in [17] this system is based on a real-dual-frequency method, in [18] it is based on the conversion of impedance to voltage and phase shift, in [25] it is built with a dual-slope capacitance-to-digital converter with a special clock source, in [19] it is based on the measurement of the quadrature and in-phase components of the signal, in [26] the interface consists mainly of the modified Martin relaxation oscillator, a multiplexer and a microcontroller, in [20] it uses a relaxation oscillator and a new capacitance-to-voltage converter, and in [21] a modified dual-slope conversion scheme is proposed. These interface circuits have several analogue active components, e.g. operational amplifiers (they often require their own supply voltages) between the sensor and a processing and control unit (e.g. a microcontroller) and use sinusoidal excitation for the sensor. The advantage of these solutions is high measuring accuracy. Unfortunately, they are characterized by high circuit complexity and costs, which disqualifies them in many sensor applications, where the key factors are low costs, small energy consumption and small dimensions.

Nowadays, capacitive sensors, as well as other types, are increasingly being designed as smart sensors. They consist of an analogue part, which contains a given sensor and conditioning circuits, and a digital part responsible for collecting and processing the measurement data, controlling the system and communication with the environment (all of these tasks can be performed by a microcontroller). Smart sensors can be one of many components of a system, or they can work as independent devices, e.g. as nodes of a wireless sensor network. In this case, they should be as

small and simple as possible to reduce their dimensions and cost, and they should be energy-efficient — they are often battery powered or can acquire energy from the environment based on an energy harvesting technique.

These requirements are met by sensor interface circuits based on the concept of the sensor-to-microcontroller interface [5,22,27,28]. In this case, a microcontroller generates the stimulation signal for the sensor and measures signal parameters of its response without using any active analogue circuits. The interface circuit for a lossy capacitive humidity sensor proposed in [22] consists of only a few discrete elements: the sensor (C_x in parallel with R_x), a calibration capacitor C_c, and two resistors, R_i and R_d. The method takes into account a stray capacitance C_{s1} between a pin of the microcontroller and ground. The determination of the capacitance C_x and parasitic conductance G_x ($1/R_x$) values of the sensor is based on four measurements of discharging times of the $(R_d \parallel R_x)(C_x + C_{s1})$, $R_d(C_c + C_{s1})$, R_dC_{s1}, and $R_x(C_x + C_{s1})$ circuits by the internal timer of the microcontroller.

However, as proposed in [27, 28] the interface circuit can be further simplified by using the internal analogue comparator (AC) of a microcontroller, which is implemented in almost all currently produced 8-bit and especially 32-bit microcontrollers. Based on this approach, a new time-domain measurement method for lossy capacitive sensors was elaborated. In this paper, a version of a grounded (one electrode of the sensor is connected to GND of the interface circuit) lossy capacitive sensor utilising the above method is proposed and analysed. This method was designed and tested for humidity sensors not distant from the system. The concept of the method is derived from the method described in [27]. Based on this, the following new solutions have been proposed:

- A new structure of the interface circuit. The interface circuit consists of only two reference resistors, R_{r1} and R_{r2}, and of course the sensor (Fig. 1). The use of resistors as reference elements is an important advantage of the method, because they are characterised by high performance precision and low temperature coefficient of resistance (e.g. typically $50 \cdot 10^{-6}/^{o}C$).
- The determination of the C_x and R_x values of the lossy capacitive sensor based on a new measurement procedure. It consists of only two measurements of discharging times t_1 and t_2 of the $(R_{r1} \parallel R_x)C_x$ and $(R_{r2} \parallel R_x)C_x$ circuits by the timer controlled by the AC, both built into the microcontroller.
- An implementation of the measurement procedure in the software of an 8-bit microcontroller.
- The dynamic correction algorithm of systematic errors, based on a correction dictionary, which is implemented in the software of the microcontroller.
- The graphic approach to systematic error correction.

It should be emphasized that the interface circuit occupies only three pins of the microcontroller (five pins if we use a resistive divider consisting of R_1 and R_2 (Fig. 1) as the threshold voltage for the AC). This follows from the fact that the same pin, Pin 3, is used to generate the stimulation when charging the interface circuit, and to monitor the voltage level on the sensor during its discharge.

Furthermore, it should be also emphasised that the hardware configuration of the interface circuit is backwards compatible with methods developed for inductive, resistive, and perfect capacitive sensors [27–30]. Therefore, to adapt the system back to such sensors, it is enough to replace the software with the software developed for earlier methods, which is an advantage from a practical point of view.

To analyse and experimentally verify the utility of the proposed measurement method and the interface circuit for grounded lossy capacitive sensors, a shunt resistance R_x was modelled by a set of resistors with resistance values from a range specified in [22], where the P14 Rapid sensor [31] under condensation effects is presented, and a capacitance C_x by the set of capacitors with capacitance values from a typical range specified in [22] and in [5], where two relative humidity sensors [32,33] are accurately characterised.

2. Description of the method

2.1. The operating principle

The proposed direct sensor-to-microcontroller interface circuit for grounded lossy capacitive sensors is shown in Fig. 1. The sensor is modelled by the equivalent circuit $R_x \parallel C_x$, with R_{r1} and R_{r2} are reference resistors. Additionally, if the microcontroller does not have an internal voltage reference source (e.g. a digital to analogue converter (DAC)), we have to use a resistive divider consisting of R_1 and R_2. The operating principle of the circuit in Fig. 1, that is, the measurement procedure, consists of four steps as shown in Fig. 2:

- Two charging steps (step I, and step III), in which C_x is charged to the high output pin voltage V_{in} ($V_{in} \approx V_{cc}$ — the supply voltage) via Pin 3 set as an output pin for a time interval longer than $5R_{pin}C_x$ (R_{pin} is the Pin 3 output resistance, $R_{pin} << R_{r1} < R_{r2}$, and also $R_{pin} << R_x$).
- Two discharging steps, in which C_x is discharged to the voltage level v_1 via $R_{r1} \parallel R_x$ (step II), and next via $R_{r2} \parallel R_x$ (step IV). In this case Pin 3 is set as input pin internally connected to the AC input.

  The states of pins 1, 2 and 3 for all steps are summarised in Table 1.

Table 1. State of pins 1, 2 and 3 and resulting discharging times t_1 and t_2.

Steps	Pin 1	Pin 2	Pin 3	Measured time
I (charging)	'1'	'1'	'1'	-
II (discharging via $R_{r1} \parallel R_x$)	'0'	HZ	HZ	$t_1 = -\ln(v_1/V_{in}) \cdot R_{r1} \parallel R_x \cdot C_x$
III (charging)	'1'	'1'	'1'	-
IV (discharging via $R_{r2} \parallel R_x$)	HZ	'0'	HZ	$t_2 = -\ln(v_1/V_{in}) \cdot R_{r2} \parallel R_x \cdot C_x$

where: '1' — output pin with a digital '1' (with an analogue output voltage V_{in}),

'0' — output pin with a digital '0' (with an analogue output voltage 0 V),

HZ — high-impedance input pin.

During the charging steps, all circuit capacitances (sensor capacitance, capacitances of microcontroller pins, and parasitic capacitances of the circuit) are always charged to V_{in}, which ensures the same initial conditions for all measurements.

The measurement of the time t_k (k = 1, 2) is as follows. Immediately after pin reconfiguration (Pin k is set to '0', other pins to HZ), the embedded timer starts, and C_x is discharged towards ground through resistance $R_{rk} \parallel R_x$. Then, when the discharging exponential signal v_{out} at Pin 3 (Fig. 1 and Fig. 2) reaches the threshold voltage v_1 of the AC, the timer stops.

According to Table 1, the discharging times t_1 and t_2 depend on v_1 and V_{in}, which can drift, for example, with temperature and/or time. However, as proven in [27] thanks to the use the ratio $\alpha = v_1/V_{in}$, the drifts are auto-calibrated by the interface circuit itself, and α is independent of the supply voltage V_{cc} of the microcontroller.

2.2. The algorithm of the measurement procedure

The software implementing the measurement procedure was written in ANSI C for a prototype of a complete solution of a compact smart sensor (Fig. 3) based on an 8-bit ATXmega32A4 microcontroller [33,34] dedicated for grounded lossy capacitive sensors. The microcontroller runs on a crystal oscillator with f_{clk} = 16 MHz (t_{clk} = 62.5 ns) and is powered by V_{cc} = 3.3 V. The 16-bit timer TC0 is controlled via the Event System by the AC (AC0). The threshold voltage v_1 is set by the internal DAC B with the voltage reference equal to V_{cc}.

The flowchart of the algorithm of the measurement procedure is presented in Fig. 4. The algorithm is implemented in two code sections: the main function section and the interrupt services section, and in the configuration of peripheral devices of the microcontroller forming the time measurement block cooperating with the voltage reference block (Fig. 3). The code in the main

section of the measurement function enables the DAC and the AC, sets v_1 at the output of DAC, and next k-times performs the following steps:

- It sets high level at outputs pins PC0, PC1 and PA1 (steps I and III) to charge the interface circuit.
- It reconfigures the PA1 pin as an input of the AC, sets low level at the pin PC(k - 1) and starts timer TC0 in Normal Mode (steps II and IV). Also, flag *end_conv* used to synchronize the software and hardware, is set. Next, we wait for completion of the measurement of time t_k by testing flag *end_conv*. When the AC detects the voltage v_{out} value achieves the value of voltage v_1, it generates an event, which saves the current value of the timer to the capture register of channel A. In turn, this event generates an interrupt. As a result, the measured time t_k is saved and flag *end_conv* is cleared - causing the transition to the main function code. After that, the timer TC0 is stopped.

Finally the code turns off the time measurement and the voltage reference blocks.

The code of the measurement function occupies only 464 Bytes of the program memory of the microcontroller, and the function responsible for configuration of peripheral devices of the microcontroller takes only 138 Bytes (AVR/GNU C Compiler 4.8.1 was used). It should be mentioned that thanks to the use of the Event System the time t_k is extended only by $6 \cdot t_{clk}$ exactly and it can be corrected by software. It results from the fact that $2 \cdot t_{clk}$ are need to run the TC0 timer (sts instruction [34, page 431]), an event lasts for $1 \cdot t_{clk}$ [34, page 75], and the propagation delay of the AC in the LP mode is 175 ns [35, page 66], which is almost $3 \cdot t_{clk}$.

2.3. Circuit analysis

The formula for voltage response $v_{out}(t)$ during the discharging stages (steps II and IV in Fig. 2) of the ideal interface circuit in the time domain has the form:

$$v_{out}(t) = V_{in} \cdot e^{\frac{-t}{R_x \| R_{rk} \cdot C_x}} \quad \text{for } k = 1, 2. \tag{1}$$

When the signal $v_{out}(t)$ reaches the value of v_1, the time t_k is measured. Then:

$$v_1 = V_{in} \cdot e^{\frac{-t_1}{R_x \| R_{r1} \cdot C_x}} \quad \text{and} \quad v_1 = V_{in} \cdot e^{\frac{-t_2}{R_x \| R_{r2} \cdot C_x}} \tag{2}$$

Equating formulae (2), we obtain the equation:

$$\frac{t_1}{R_x \,||\, R_{r1}} = \frac{t_2}{R_x \,||\, R_{r2}} \tag{3}$$

From (3), we can determine the formula for R_x:

$$R_x = \frac{R_{r1} \cdot R_{r2} \cdot (t_2 - t_1)}{R_{r2} \cdot t_1 - R_{r1} \cdot t_2} \tag{4}$$

If we assume that $\xi = R_{r2}/R_{r1}$, we can write the final version of the formula for R_x:

$$R_x = R_{r2} \frac{(t_2 - t_1)}{\xi \cdot t_1 - t_2} \tag{5}$$

From (5), we can see that R_x depends on the constant circuit parameters R_{r2} and ξ and two measured values t_1 and t_2.

Based on the second formula of (2) and $\alpha = v_1/V_{in}$, we can write that:

$$\alpha = \frac{v_1}{V_{in}} = e^{\frac{-t_2}{R_x \| R_{r2} \cdot C_x}} \tag{6}$$

From (6), we can derive the formula for C_x:

$$C_x = \frac{-(R_{r2} - R_{r1})}{R_{r1} \cdot R_{r2} \cdot \ln \alpha} \cdot \frac{t_1 \cdot t_2}{t_2 - t_1} \tag{7}$$

In the similar way as before, we can assume that:

$$\gamma = \frac{-(R_{r2} - R_{r1})}{R_{r1} \cdot R_{r2} \cdot \ln \alpha} \tag{8}$$

Which gives the final formula for C_x:

$$C_x = \gamma \cdot \frac{t_1 \cdot t_2}{t_2 - t_1} \tag{9}$$

In this case, coefficient γ is also the constant circuit parameter.

Parameters R_{r2}, ξ and γ characterise the configuration of the interface circuit and can be determined only once during the design of the electronic system. Thus, they are constant for a given interface configuration, and they can be constants in the program of the microcontroller. Thanks to this, calculations of R_x and C_x based on (5) and (9) are simple.

2.4. Determination of the α coefficient

Determination of the α parameter and also values of reference resistors R_{r1} and R_{r2} are shown for an example of a prototype of a compact smart sensor presented in Fig. 3. The value ranges for

C_x and R_x are as follows [5,22]: $C_x \in \langle C_{x,\min}; C_{x,\max} \rangle$, $R_x \in \langle R_{x,\min}; R_{x,\max} \rangle$, where $C_{x,min}$ = 100 pF, $C_{x,max}$ = 286 pF, $R_{x,min}$ = 1 MΩ, and $R_{x,max}$ = 10 MΩ.

The coefficient $\alpha = v_1/V_{in}$ ($0 < \alpha < 1$) should be such that the maximum possible relative uncertainty (error) [27] $dC_x(\alpha)$ of the determination of the values of C_x with respect to this coefficient is as small as possible. We can also see from Table 1 that the smaller α, the greater t_1 and t_2, and thus the smaller the effect of the discretisation error on the measurement results.

Based on the theory presented in [27], we can write that:

$$dC_x(\alpha) = \frac{\partial C_x(\alpha)}{\partial \alpha} \cdot \frac{1}{C_x(\alpha)} = \frac{\Delta\alpha}{\alpha \cdot \ln\alpha} \tag{10}$$

The component $\Delta\alpha = \Delta c_{DAC} / 2^N$ represents the integral and differential non-linearity Δc_{DAC} of setting the value c_{DAC} on the DAC, where N = 12 [35, page 66]. It should be mentioned that $\Delta\alpha$ and $\alpha = c_{DAC}/2^N$ are constant for a given microcontroller and they are not sensitive to temperature changes [27]. A graph of $dC_x(\alpha)$ for $\Delta\alpha$ = 6 LSB is shown in Fig. 5. As can be seen, the values of $dC_x(\alpha)$ are minimal for α between 0.2 and 0.5. Hence, α = 0.2002 was chosen, which corresponds to c_{DAC} = 820.

In the case of using a resistive divider as the threshold voltage source of the AC, we suggest the following resistance values: R_1 = 33 kΩ, and R_2 = 130 kΩ, which results in α = 0.2025. Resistors R_1 and R_2 must have the same temperature coefficient of resistance so that α is not sensitive to temperature changes of these resistors. The accuracy of the α setting, i.e. the uncertainty $d\alpha(R_1,R_2)$ of determining this value, depends as follows on the tolerance δR of resistors R_1 and R_2: $d\alpha(R_1,R_2) \approx 0.23 \cdot \delta R$. Thus it is enough for the resistors to have a tolerance of 0.1%, then $d\alpha(R_1,R_2)$ is 0.023%.

2.5. Determination of the values of reference resistors R_{r1} and R_{r2}

If we assume that $R_{r1} < R_{r2}$, then $t_1 < t_2$. If we also assume that $t_2 < t_{2,max}$ (where: $t_{2,max} = t_{clk} \cdot 2^N$, and N = 16), that is, we do not want the timer to overflow, then we can determine R_{r2}:

$$R_{r2} \le R_{r2,max} = \frac{-R_{x,max} \cdot t_{2,max}}{t_{2,max} - R_{x,max} \cdot C_{x,max} \cdot \ln\alpha} \tag{11}$$

Thus, we have $R_{r2,max}$ = 164 MΩ for N = 16, and $R_{r2,max}$ = 8.9 MΩ for N = 15. We should not forget about the additional parasitic capacitances and the measurement errors that can result in exceeding the measuring range of the timer. For this reason, the following resistance value from the

E6 series was selected: R_{r2} = 10 MΩ. For this value: $2^{15} \cdot t_{clk} < t_{2,max} = 34764 \cdot t_{clk} = 2.2$ ms $< 2^{16} \cdot t_{clk}$, and $t_{2,min} = 2340 \cdot t_{clk} = 0.1462$ ms $> 2^{11} \cdot t_{clk}$.

Determination of the R_{r1} value is more complex. It follows from (5) and (8) that to minimise the discretisation error of the time measurements, two opposite criteria must be met: if the value of R_{r1} increases, t_1 also increases, but $(t_2 - t_1)$ decreases. Hence a compromise must be chosen: $t_1 > 2^n \cdot t_{clk}$, and $(t_2 - t_1) > 2^n \cdot t_{clk}$. The smallest values of t_1 and $(t_2 - t_1)$ are determined for $C_{x,min}$ and $R_{x,min}$. For these values, functions $t_1(R_{r1})$ and $(t_2 - t_1(R_{r1}))$ are drawn in Fig. 6. It can be seen in Fig. 6 that this condition is fulfilled for n = 10, that is, for R_{r1} from $0.066 \cdot R_{r2}$ to $0.105 \cdot R_{r2}$. Therefore the value $R_{r1} = 0.1 \cdot R_{r2} = 1$ MΩ from the series E6 was chosen. For this value: $t_{1,min} = 1288 \cdot t_{clk} = 80.5$ µs $> 2^{10} \cdot t_{clk}$, and $t_{1,max} = 6316 \cdot t_{clk} = 394.75$ µs $> 2^{12} \cdot t_{clk}$.

3. Error analysis

The accuracy of determining the values of C_x and R_x is burdened with random and systematic errors. The random errors are related to the measurement errors of the discharging times t_1 and t_2, which will be discussed in the next subsections.

The systematic errors are caused by the uncertainty of the constant circuit parameters R_{r1}, R_{r2}, and α (4) (8), but above all by the influence of the non-ideality of the port pins of the microcontroller and the parasitic capacitances C_{p1}, C_{p2}, and C_{p3} representing, inter alia, capacitances between the printed circuit board traces. It follows from the fact that the values of capacitance C_x are in a range of between 100 pF and 286 pF, and the pin capacitance C_{pin} is at a similar level (about 10 pF). The model of the interface circuit, taking into account the pin parameters of the microcontroller, is shown in Fig. 7.

Whereas output pin resistance R_{pin_on}, at a low voltage level, is around 25 Ω, input pin resistance R_{pin_off} is estimated at around 3 GΩ or more [35, page 67]. That is, they can be omitted in the simplified model of the interface circuit (Fig. 8), because $R_{pin\_on} << R_{rk}$, and $R_{rk} \parallel R_x << R_{pin\_off}$, and R_{pin_off} is connected in parallel with C_{pin}. In this case, the measured value of the sensor capacitance is $C_m \approx C_x + C_{pin3} + C_{p3}$, and the measured value of the sensor shunt resistance should be $R_m \approx R_x$. We can assume that $C_{pink} = C_{pin} + C_{pk}$. Two approaches of systematic error correction to minimise these errors are proposed in Subsections 4.1 and 4.2.

3.1. Assessment of uncertainty for determination of R_x

As presented in [24,27,36], the maximum possible relative uncertainty (error) of an indirectly measurable variable $y = f(x_1, .., x_L)$ is the sum of the maximum relative uncertainties $df(x_l)$ of the directly measurable variables x_l (l = 1, 2, .., L). The relative uncertainties $df(x_l)$ are given by the formula:

$$df(x_l) = \frac{1}{|f(x_{1,2},..,x_L)|} \cdot \left|\frac{\partial f}{\partial x_l}\right| \cdot |\Delta x_l|, \tag{12}$$

where: Δx_l is the maximum absolute uncertainty of a directly measurable variable x_l.

It follows from (4) that indirectly measurable resistance R_x depends on directly fixed/measurable variables R_{r1}, R_{r2}, t_1, and t_2. For these directly variables based on (12), the relative uncertainties $dR_x(R_{r1})$, $dR_x(R_{r2})$, $dR_x(t_1)$, and $dR_x(t_2)$ were derived:

$$dR_x(R_{r1}) = \frac{R_{r2} \cdot t_1}{R_{r2} \cdot t_1 - R_{r1} \cdot t_2} \cdot \frac{\Delta R_{r1}}{R_{r1}} \tag{13}$$

$$dR_x(R_{r2}) = \frac{R_{r1} \cdot t_2}{R_{r2} \cdot t_1 - R_{r1} \cdot t_2} \cdot \frac{\Delta R_{r2}}{R_{r2}} \tag{14}$$

$$dR_x(t_1) = \frac{t_2 \cdot (R_{r1} - R_{r2}) \cdot}{(t_2 - t_1) \cdot (R_{r2} \cdot t_1 - R_{r1} \cdot t_2)} \cdot \Delta t_1 \tag{15}$$

$$dR_x(t_2) = \frac{t_1 \cdot (R_{r2} - R_{r1}) \cdot}{(t_2 - t_1) \cdot (R_{r2} \cdot t_1 - R_{r1} \cdot t_2)} \cdot \Delta t_2 \tag{16}$$

Components $\Delta R_{rk} / R_{rk}$ of formulae (13) and (14) contain the tolerance of R_{rk} and the influence of the thermal drift affecting R_{rk} [27]. They are small (e.g. 0.01% or 0.0023% if we use Agilent 34410A [37] to measure the R_{rk} values or if we use precision resistors with a tolerance equal to or better that 0.05%) in comparison with the Δt_k components of formulae (15) and (16), hence they can be neglected.

The Δt_k components represent the effects of both quantisation and above all trigger noise, that is, the inaccuracy of the functioning of the AC, where $\Delta t_k = f_k(\Delta v_{1,k}, R_{rk}, R_x, C_x)$, and $\Delta v_{1,k}$ is the noise (disturbances) generated, inter alia, by the digital circuits of the microcontroller [27].

Based on the fact that $v_1/V_{in} = c_{DAC}/2^N$, and $\Delta v_{1,k}/V_{in} = \Delta c_{DAC,k}/2^N$, formulae included in Table 1, and the assumption that $t_k' = t_k + \Delta t_k$, we can derive the following formula for Δt_k:

$$\Delta t_k = -R_x \,||\, R_{wk} \cdot C_x \cdot \ln\left(1 + \frac{\Delta c_{DAC,k}}{c_{DAC}}\right), \; k = 1, 2. \tag{17}$$

It was also observed during experimental studies that as the value of $R_{rk} \,||\, R_x$ increases, the noise level also increases. For this reason, it was established that Δt_1 is plotted for $\Delta c_{DAC,1} = 4$, and Δt_2 for $\Delta c_{DAC,2} = 12$ in Fig. 9. Graphs of $dR_x(t_1)$ and $dR_x(t_2)$ were drawn also for these Δt_k values in Fig. 10. It can be seen that $dR_x(t_1)$ is in the range of 0.29% to 0.37%, and $dR_x(t_2)$ is in the range of 0.87% to 1.1%.

3.2. Assessment of uncertainty for determination of C_x

As before, based on (7), the relative inaccuracies $dC_x(\alpha)$, $dC_x(R_{r1})$, $dC_x(R_{r2})$, $dC_x(t_1)$, and $dC_x(t_2)$ for the indirectly measurable capacitance C_x, which depends on directly variables α, R_{r1}, R_{r2}, t_1, and t_2, were derived:

$$dC_x(\alpha) = \frac{\Delta\alpha}{\alpha \cdot \ln\alpha} \tag{18}$$

$$dC_x(R_{r1}) = \frac{R_{r2}}{R_{r2} - R_{r1}} \frac{\Delta R_{r1}}{R_{r1}} \tag{19}$$

$$dC_x(R_{r2}) = \frac{R_{r1}}{R_{r2} - R_{r1}} \frac{\Delta R_{r2}}{R_{r2}} \tag{20}$$

$$dC_x(t_1) = \frac{t_2}{t_2 - t_1} \cdot \frac{\Delta t_1}{t_1} \tag{21}$$

$$dC_x(t_2) = \frac{t_1}{t_2 - t_1} \cdot \frac{\Delta t_2}{t_2} \tag{22}$$

As was previously shown, $dC_x(R_{r1})$, $dC_x(R_{r2})$, and $dC_x(\alpha)$ [27] can also be neglected. The graphs of $dR_x(t_1)$ and $dR_x(t_2)$ were drawn in Fig. 11. In this case, $dC_x(t_1)$ is in the range of 0.18% to 0.37%, and $dC_x(t_2)$ is in the range of 0.1% to 0.55%.

4. Materials and methods

The proposed measurement method for lossy capacitive sensors was experimentally examined based on the compact smart sensor shown in Fig. 3. The ATXmega32A4 microcontroller's timer/counter TC0 is clocked directly by the system clock (t_{clk} = 62.5 ns). The value written to the DAC data register DACB_CH0DATA c_{DAC} = 814, which makes v_{DAC} = 0.66064 V (α = 0.2002). The supply voltage V_{CC} = 3.3008 V, and the measured values of the reference resistors are as follows: R_{r1} = 1.00053 MΩ, and R_{r2} = 10.0092 MΩ. All voltages and resistances were measured with an Agilent 34410A Digital Multimeter. A photograph of the prototype of the smart sensor shown in Fig. 12.

The lossy capacitive sensor was emulated by a set of discrete components (i.e. a capacitor in parallel with a resistor). It used twelve ceramic capacitors $\{C_{x,i}\}_{i = 1, .., 12}$ whose values were measured by an Agilent 4263B LCR METER at 1 kHz using a 16047A Test Fixture by Hewlett-Packard (Table 2). Seven metal film resistors $\{R_{x,j}\}_{j = 1, .., 7}$ were also used (the THT assembly, with

a power rating of 0.6 W, ϕ2.5 × 6.8 mm, TCR = $50 \cdot 10^{-6}$/°C) with the reference values included in Table 3.

Table 2. A set of capacitors used to emulate the capacitance C_x of the sensor.

i	1	2	3	4	5	6	7	8	9	10	11	12
$C_{x,i}$ **[pF]**	100.06	108.94	119.82	131.14	146.83	162.97	179.53	182.21	199.94	219.65	239.37	285.85

Table 3. A set of resistors used to emulate the shunt resistance R_x of the sensor.

j	1	2	3	4	5	6	7
$R_{x,j}$ **[MΩ]**	1.00212	2.18195	3.2994	4.7034	6.8426	8.2014	10.0015

The measurements were carried out 1024 times for all combinations of $C_{x,i}$ and $R_{x,j}$ values, and for a single measurement of times and for multiple measurements of times (M = 64) [27]. Therefore, for each pair of $C_{x,i}$ and $R_{x,j}$, $1024 \cdot (1 + 64) = 66{,}560$ measurements were performed, which gives a total of $12 \cdot 7 \cdot 66{,}560 = 5{,}591{,}040$ measurements. E.g. for the assumed capacitance and resistance ranges of a sensor, t_{clk} = 62.5 ns and M = 64, the duration for multiple measurements procedure is in a range between about 15 ms and 167 ms.

After each h-th single and M-multiple measurement (h = 1, .., 1024) for each i,j-th combination, the measured values of discharging times in raw form, i.e. in the form of a value directly read from the timer/counter TC0, of the first time $\tau_{1,i,j,h}$ and the second time $\tau_{2,i,j,h}$ and also the $C_{m,i,j,h}$ and $R_{m,i,j,h}$ values calculated by the microcontroller based on (23) and (24), were sent to a personal computer via a USB interface.

$$R_{m,i,j,h} = R_{r2} \frac{\left(\tau_{2,i,j,h} - \tau_{1,i,j,h}\right)}{\xi \cdot \tau_{1,i,j,h} - \tau_{2,i,j,h}} \tag{23}$$

$$C_{m,i,j,h} = \gamma \cdot t_{clk} \cdot \frac{\tau_{1,i,j,h} \cdot \tau_{2,i,j,h}}{\tau_{2,i,j,h} - \tau_{1,i,j,h}} \tag{24}$$

where, $\tau_{k,i,j,h} = t_{k,i,j,h}/t_{clk}$, i = 1, 2, .., 12, j = 1, 2, .., 7, and h = 1, 2, .., 1024.

Each time, for each $X_{i,j}$ value, where $X_{i,j}$ represented $\tau_{1,i,j}$, $\tau_{2,i,j}$, $C_{m,i,j}$, and $R_{m,i,j}$, the lowest $X_{i,j,min} = \min_h \{X_{i,j,h}\}$, highest $X_{i,j,max} = \max_h \{m_{t,i,h}\}$, and mean $X_{i,j,mean} = \left(X_{i,j,min} + X_{i,j,max}\right)/2$, values from the sets $\{X_{,i,j,h}\}_{h = 1, .., 1024}$ of measurement and calculation results were determined and calculated. The absolute $\Delta X_{i,j}$ and relative $dX_{i,j}$ measurement errors were also defined:

$$\Delta X_{i,j} = X_{i,j,mean} - X_{i,j,reference} \tag{25}$$

$$dX_{i,j} = \frac{\Delta X_{i,j}}{X_{i,j,reference}} \cdot 100\% \tag{25}$$

where $X_{i,,j,reference}$ is the reference value or the value calculated from the theoretical data.

5. Experimental results and discussion

The measurement results of scaling of the interface circuit for the $\tau_{k,mean}$ values are given in Fig. 13. Whereas the absolute measurement errors Δt_1 and Δt_2 of these times for single measurements are plotted in Fig. 14. Comparing the surfaces in this figure with the surfaces in Fig. 9, we can conclude that as the $R_{rk} \parallel R_x$ value increases, the noise level also increases. E.g. for the Δt_1 plot, that is, if we use R_{r1}, it is on the level of about 3 mV (Δc_{DAC} about 4), and for the Δt_2 plot, that is, if we use R_{r2}, it is on the level of about 10 mV (Δc_{DAC} about 12), which is an adverse phenomenon.

Fig. 15 shows the experimental relative errors dC_m and dR_m in the measurements of the C_x and R_x values. The values of dC_m are in the range from 3.2% to 14.7%, and the values of dR_m are in the range from 4.3% to 19.2%. They are at a similar level as for the method proposed in [22].

For large R_x values, the dC_m error is at a constant level. However, the smaller the C_x values, the larger this error. This follows from the fact that the measured value C_m includes, apart from C_x, also the input capacitances of the port pins and parasitic capacitances (Fig. 8). These additional capacitances introduce an absolute error value between 8 pF and 19.2 pF.

Also, the values of error dR_m depend on the values of C_x and R_x simultaneously. The measured value R_m of R_x is 66 kΩ to 1.97 MΩ larger than the set R_x values. This is due to the extension of the discharging times t_1 and especially t_2, which is caused by the need to discharge additional parasitic capacitances by $R_{rk} \parallel R_x$. Therefore, the smaller the C_x values, the greater the impact of the other capacitances on the measurement results.

Hence, errors dC_m and dR_m are caused mainly by parasitic capacitances existing in the interface circuit. They belong to systematic errors, and therefore they can be software corrected based on the results of scaling of the interface circuit, which will be shown in the following subsections.

The effect of noise on the measurement accuracy was also tested. In this case, the relative random errors $dX_{i,j,noise}$ are defined in the following way:

$$dX_{i,j,noise} = \frac{\Delta X_{i,j,max} - \Delta X_{i,j,min}}{2 \cdot X_{i,j,reference}} \cdot 100\% \tag{26}$$

The relative random errors dC_{noise} and dR_{noise} for single measurements are plotted in Fig. 16, and for multiple measurements, in Fig. 17. In the first case, dC_{noise} is in the range from ±0.29% to

±0.71%, and dR_{noise} is in the range from ±0.64% to ±2.3%. Fig. 16 shows that error dC_{noise} decreases as the $C_x \cdot R_x$ time constant increases. Whereas the values of dR_{noise} are mainly determined by the values of C_x.

For the approach in which t_1 and t_2 are measured $M = 64$ times and then averaged [27], these errors have the following values: dC_{noise} is in the range from ±0.04% to ±0.19%, and dR_{noise} is in the range from ±0.09% to ±0.55% (Fig. 17). Thanks to this approach, errors were reduced by about 3.7–7.25 times for dC_{noise} and about 4.2–7.1 times for dR_{noise}.

This allowed a resolution of measuring capacitance from 0.05 pF to 0.34 pF to be obtained, which is a satisfactory result (as presented in [5] for commercial capacitive RH sensors [32,33], the sensitivity varies from 0.2 pF up to 0.5 pF per 1% of RH), and a resolution for measuring resistance from 1.2 kΩ to 55.7 kΩ to be obtained.

5.1. Graphic approach to systematic error correction

The first proposal to reduce systematic errors is a graphic approach. Fig. 18a shows the scaling curves for C_x plotted for the R_x parameter placed in three-dimensional space with coordinates t_1, t_2, and C_x (left graph), and scaling curves for R_x this time plotted for the C_x parameter placed in space with coordinates t_1, t_2, and R_x (right graph), where t_1, t_2 are the measured discharging time values, and C_x, and R_x are the reference values. These spaces together with the curves can be projected onto planes with the coordinates t_1, t_2 as shown in Fig. 18b. In the last step, we can superimpose one plane with the C_x curves onto the other plane with the R_x curves. In this way, we obtain the scaled mesh shown in Fig. 19. This mesh can be scaled: the C_x curves in picofarads, and the R_x curves in megohms.

Using this approach, the determination of the corrected C_{corr} and R_{corr} values of C_x and R_x consists in placing the measurement point E with coordinates t_1 and t_2 on a plane with a scaled mesh (Fig. 19). Then, this point is projected onto the scaled R_x curve along the C_x curve and onto the scaled C_x curve along the R_x curve. We read the C_{corr} and R_{corr} values from these curves.

The advantage of this approach is that we read the corrected values of C_x and R_x directly from the scaled mesh based on only the measured values t_1 and t_2.

Unfortunately, this solution has significant disadvantages that impede its implementation in the microcontroller software. Scales on C_x and R_x curves are nonlinear, and the trajectories of the projections are not straight lines.

5.2. Numerical approach to systematic error correction

The second proposal to reduce systematic errors, based on a correction dictionary, was implemented in the microcontroller software. The correction dictionary is a set of data used for

correction. It was named by analogy with fault dictionaries, which are data sets used for fault detection and location in analogue electronic circuits [38,39].

This set is determined as follows. In the first step, a set of curves $C_{x,j} = f_j(C_{m,i,j},R_{m,i,j})$ was plotted in a space with coordinates C_m, R_m, C_x, and a set of curves $R_{x,i} = f_i(C_{m,i,j},R_{m,i,j})$ was plotted in a space with coordinates C_m, R_m, R_x (Fig 20). It is desirable that $C_{corr} = C_x$, and $R_{corr} = R_x$. Therefore, the C_{corr} curves are an approximation of the C_x curves using a straight line (27), and the R_{corr} curves are an approximation of the R_x curves using a quadratic function (28):

$$C_{corr} = A_1 \cdot C_m + A_2 \tag{27}$$

$$R_{corr} = B_1 \cdot R_m \cdot R_m + B_2 \cdot R_m + B_3 \tag{28}$$

where: A_1, and A_2 are the coefficients of the linear function depended on R_m, and B_1, B_2, and B_3 are the coefficients of the quadratic function depended on C_m (as shown in the projections on the plane C_m, R_m in Fig. 20).

Graphs of A_1, A_2 as a function of R_m are drawn in Fig. 21a, and graphs of B_1, B_2, B_3 as a function of C_m in Fig. 21b. It was tested based on the Matlab function polyfit and found that the curves of these coefficients are best approximated by third-degree polynomials $P(\cdot)$ (dashed lines in the graphs in Fig. 21). Hence, they can be described by:

$$A_1 = P\left(R_m, \left\{d_{1,p}\right\}_{p=1,..,4}\right) \tag{29a}$$

$$A_2 = P\left(R_m, \left\{d_{2,p}\right\}_{p=1,..,4}\right) \tag{29b}$$

$$B_1 = P\left(C_m, \left\{e_{1,p}\right\}_{p=1,..,4}\right) \tag{29c}$$

$$B_2 = P\left(C_m, \left\{e_{2,p}\right\}_{p=1,..,4}\right) \tag{29d}$$

$$B_3 = P\left(C_m, \left\{e_{3,p}\right\}_{p=1,..,4}\right) \tag{29e}$$

where $\{d_{1,p}, d_{2,p}, e_{1,p}, e_{2,p}, e_{3,p}\}_{p = 1, .., 4}$ are the coefficients of individual third-degree polynomials, and they create the correction dictionary. Therefore this dictionary is small because it consists of only 20 elements of the type 'double'. That is, it occupies only 160 bytes of a program memory of the microcontroller.

Hence the systematic error correction algorithm based on this approach is as follows:

1. Values of R_m, and C_m are calculated based on the measured values of times t_1 and t_2, and (23) and (24).
2. Values of A_1, and A_2 are calculated based on R_m and $\{d_{1,p}, d_{2,p},\}_{p = 1, .., 4}$, and (29a) and (29b).

3. Values of B_1, B_2, and B_3 are calculated based on C_m and $\{e_{1,p}, e_{2,p}, e_{3,p}\}_{p = 1, .., 4}$, and (29c), (29d) and (29e).
4. Value of C_{corr} is calculated based on the values of C_m, A_1, and A_2, and (27).
5. Value of R_{corr} is calculated based on the values of R_m, B_1, B_2, and B_3, and (28).

This algorithm was implemented as the function written in ANSI C. It occupies only 1428 Bytes of the program memory and uses 80 Bytes of the SRAM of the microcontroller. Execution time for this function is short. It is about 0.46 ms for an 8-bit microcontroller runs on a crystal oscillator with f_{clk} = 16 MHz.

Thanks to the presented solution, it is possible to reduce the systematic error up to about 23 times for C_x, and up to about 66 times for R_x, as shown in Fig. 22. In this case, the values of dC_m are in the range from -0.55% to 0.64%, and the values of dR_m are in the range from -0.46% to 0.3%. It provides a measuring capacitance resolution from 0.99 pF to 1.83 pF, which, despite everything, is a satisfactory result, especially for using such a simple correction method based on such a small correction dictionary. The improvement of the resistance measurement resolution is also significant. It is in the range from 4.6 kΩ to 30.3 kΩ.

In summary, when we take into account both systematic and random errors, we obtain a measurement relative uncertainty (relative error) of capacitance of less than 0.71% (resolution < 2.03 pF for the range 100–286 pF), and a resistance uncertainty of less than 0.74% (resolution < 58.61 kΩ for the range 1–10 MΩ). However, if we narrow the range of measured capacitances to the range of capacitances corresponding to e.g. the HS1101LF sensor (we take into account $C_{x,6} - C_{x,9}$), then the relative uncertainty of capacitance is less than 0.57% and resolution < 0.08 pF, which in turn corresponds to at least 0.26%RH resolution of this sensor [32].

6. Comparison of the method with the state of the art

The proposed measurement method is compared with a modified 3-point calibration technique (3-PCT) [22], because both methods are based on direct sensor-to-microcontroller interface circuits dedicated to lossy capacitive sensors. The first method was developed for grounded sensors, whereas the second one is for floating sensors. The results of the comparison are included in Table 4. The table also includes a comparison of methods dedicated to perfect capacitive sensors proposed in [27] and [5], because the concepts of the proposed method and the 3-PCT are derived from these methods, respectively.

Table 4. Comparison of the proposed method with the modified 3-point calibration technique.

Reference	**C_x and R_x range**	**Circuit topology**	**Built-in peripherals**	**C_x and R_x accuracy (% error)**	**Measurement time**
This work	100–286 pF and 1–10 MΩ	R_{r1}, R_{r2}	3 I/O pins, timer, AC, and a voltage reference source	< 1.1 (< 0.71*) and < 2.75 (< 0.74*)	$t_1 + t_2$
3-PCT [20]	150–206 pF and 1–10 MΩ	R_d, R_i, C_c	3 I/O pins, timer	< 6 and not specified	$T_x + T_c + T_{off} + T_{ad}$
[27]	100–225 pF and >>10 MΩ	R_r	2 I/O pins, timer, AC, voltage reference source	<0.081 (<0.057*) and not taken into account	t_m
[5]	149–206 pF and >>10 MΩ	R_d, R_i, C_c	3 I/O pins, timer	<0.036 for M = 100 and not taken into account	$T_x + T_c + T_{off}$

* – for an M-multiple measurement approach (M = 64)

where:

T_x, T_c, T_{off}, T_{add} — sensor, reference, offset, and additional measurement discharging times,

t_m — time of charging C_x via R_r to V_m.

We can see in Table 4 that an important advantage of the proposed measurement method is the about six times smaller relative error of capacitance measurement for a single measurement approach (ten times smaller for an M-multiple measurement approach) with respect to the modified 3-PCT method. The authors in [22] presented data for C_x = 150 pF. Whereas, for the proposed method and for a similar value, $C_{x,5}$ = 146.83 pF, a relative error less than 0.75% (0.37%) was obtained, that is, about an 8-fold (16-fold) improvement in the measurement accuracy.

Another advantage is that we used resistors instead of a capacitor as a reference element. This follows from the fact that resistors are characterised by high precision of performance and low temperature coefficient of resistance relative to capacitors.

If we assume that the discharging times for the modified 3-PCT method and the proposed method are similar for maintaining the same measurement resolution of the timer of the microcontroller, we can conclude that the total measurement time of the proposed method is about two times shorter, what is the final advantage of the proposed method as it lowers energy consumption as shown in [27]. It follows from the fact that the core processor consumes the most

energy. E.g. for the ATXmega32A4 microcontroller the power supply current in the active mode at 3.3 V, for a 16 MHz external clock and at 25°C is equal to about 7.5 mA [35, page 70]. Whereas the AC current consumption (low-speed) is 110 μA, and the DAC consumes less than 1.1 mA in the low-power mode. The timer current consumption (Prescaler DIV1) is 19 μA [35, page 62]. That is, the total current consumption for the proposed method is only 8.73 mA. For the modified 3-PCT method this consumption is 7.52 mA. Hence, the energy consumption has decreased about by 1.72 times due to a two-fold reduction in the measurement time.

7. Conclusions

In the paper, a new time-domain measurement method is presented for determining the capacitance and resistance values of lossy relative humidity capacitive sensors, dedicated, among others, to grounded sensors, based on a direct sensor-to-microcontroller interface circuit. The method is designed for microcontrollers equipped with internal analog comparators and timers. The interface circuit consists only of four reference resistors (two reference resistors if a microcontroller includes a voltage reference source) and a sensor. In this method, we proposed: a new structure of the interface circuit, a new measurement procedure and its implementation in a microcontroller software, a new graphic approach to systematic error correction and a new systematic error correction algorithm based on a correction dictionary.

The experimental investigations were performed based on an example of a complete application in the form of a prototype based on an 8-bit ATXmega32A4 microcontroller. The experiments confirmed that relative errors of measurement of capacitance introduced by the interface circuit are less than 0.71%, which corresponds to a capacitance measurement resolution less than 2.05 pF for a range of the measured capacitance values from 100 pF to 286 pF, and relative errors of measurement of resistance are less than 0.74% (a resistance measurement resolution of less than 58.61 kΩ for a range of the measured resistance values from 1 MΩ to 10 MΩ).

Such good results were obtained thanks to the implementation of a systematic error correction algorithm based on a correction dictionary and the *M*-multiple measurement approach in the microcontroller software.

The software written for this method was implemented in an 8-bit microcontroller, which confirms that the low computing power and limited program and data memories of 8-bit microcontrollers are enough for this method. Thanks to this, based on the proposed solution, we can extend the functionality of existing microcontroller systems with direct sensor-to-microcontroller interface circuits in a cheap and simple way.

References

[1] T. Islam, A. U. Khan, J. Akhtar, M. Z. U. Rahman, A Digital Hygrometer for Trace Moisture Measurement, IEEE Transactions on Instrumentation and Measurement 61 (10) (2014) 5599-5605.

[2] M. Dokmeci, K. Najafi, A High-Sensitivity Polyimide Capacitive Relative Humidity Sensor for Monitoring Anodically Bonded Hermetic Micropackages, Journal of Microelectromechanical Systems, 10 (2) (2001) 197-204.

[3] Y. Kim, B. Jung, H. Lee, H. Kim, K. Lee, H. Park, Capacitive humidity sensor design based on anodic aluminum oxide, Sensors and Actuators B 141 (2009) 441–446.

[4] A. Rivadeneyra, J. Fernandez-Salmeron, M. Agudo-Acemel, J. A. Lopez-Villanueva, L. F. Capitan-Vallvey, A. J. Palmac, Printed electrodes structures as capacitive humidity sensors: A comparison, Sensors and Actuators A 244 (2016) 56–65.

[5] F. Reverter, O. Casas, Direct interface circuit for capacitive humidity sensors, Sensors and Actuators A 143 (2008) 315–322.

[6] J. P. Sanjurjo, E. Prefasi, C. Buffa, R. Gaggl, A Capacitance-To-Digital Converter for MEMS Sensors for Smart Applications, Sensors 17 (2017) 1312.

[7] S. Bilent, T. H. N. Dinh , E. Martincic, P.-Y. Joubert, Influence of the Porosity of Polymer Foams on the Performances of Capacitive Flexible Pressure Sensors, Sensors 19 (2019) 1968.

[8] Hyeong-Joon Ahn, Soo Jeon, Error analysis of a new cylindrical capacitive sensor (CCS) for measuring five-dimensional motions of a rotor, Mechanical Systems and Signal Processing 29(2012)148–163.

[9] F. Restagno, J. Crassous, E. Charlaix, M. Monchanin, A new capacitive sensor for displacement measurement in a surface-force apparatus, Measurement Science and Technology 12 (2001) 16–22.

[10] J. Boudaden, A. Klumpp, H. E. Endres, I. Eisele, Capacitive CO_2 Sensor, Proceedings 1 (2017) 472.

[11] P. L. Kebabian, A. Freedman, Fluoropolymer-based capacitive carbon dioxide sensor, Measurement Science and Technology 17 (2006) 703–710.

[12] S. C. Bera, H. Mandal, A Flow Measurement Technique Using a Noncontact Capacitance-Type Orifice Transducer for a Conducting Liquid, IEEE Transactions on Instrumentation and Measurement 61 (9) (2012) 2553-2559.

[13] K.. A S Al Khateeb, R. T. Anika, S. Khan, M. Mohamud, A. Arshad, K. Hasan, S. S. Haider, M. M. Shobaki, Experimental evaluation of agricultural biomass flow sensing behaviour using capacitive technique, IOP Conf. Series: Materials Science and Engineering 53 (2013) 012034.

[14] C. Wang, K. Shida, A new method for on-line monitoring of brake fluid condition using an enclosed reference probe, Measurement Science and Technology 18 (2007) 3625–3635.

[15] B. Jakoby, M. J. Vellekoop, Physical sensors for water-in-oil emulsions, Sensors and Actuators A 110 (2004) 28–32.

[16] R. N. Dean, A. K. Rane, M. E. Baginski, J. Richard, Z. Hartzog, D. J. Elton, A Capacitive Fringing Field Sensor Design for Moisture Measurement Based on Printed Circuit Board Technology, IEEE Transactions on Instrumentation and Measurement 61 (4) (2012) 1105-1112.

[17] A. K. R. Segundo, É. S. Pinto, G. A. Santos, P. M. de Barros Monteiro, Capacitive Impedance Measurement: Dual-frequency Approach, Sensors 19 (2019) 2539.

[18] A. Fendri, A. Y. Kallel, H. Nouri, H. Ghariani, O. Kanoun, Measurement System for Lossy Capacitive Sensors: Application to Edible Oils Quality Assessment, Sensors 19 (2019) 4299.

[19] A. U. Khan, T. Islam, B. George, M. Rehman, An Efficient Interface Circuit for Lossy Capacitive Sensors, IEEE Transactions on Instrumentation and Measurement 68 (3) (2019) 829-836.

[20] A. Heidary, G. C. M. Meijer, An Integrated Interface for Leaky Capacitive Sensor with Emphasize on Humidity Sensor, In proc. of I2MTC 2008, Vancouver Island, Canada (2008).

[21] P.Vooka, B. George, A Direct Digital Readout Circuit for Impedance Sensors, IEEE Transactions on Instrumentation and Measurement 64 (4) (2015) 902-912.

[22] F. Reverter, Ò. Casas, A microcontroller-based interface circuit for lossy capacitive sensors, Measurement Science and Technology 21 (2010) 065203.

[23] G. C. M. Meijer, Smart Sensor Systems, 2008, Chichester, U.K. : J. Wiley & Sons.

[24] S. Malik, K. Kishore, T. Islam, Z. H. Zargar, S. A. Akbar, A time domain bridge-based impedance measurement technique for wide-range lossy capacitive sensors, Sensors and Actuators A 234 (2015) 248–262.

[25] P. Vooka, B. George, An Improved Capacitance-to-Digital Converter for Leaky Capacitive Sensors, IEEE Sensors Journal 15 (11) (2015) 6238-6247.

[26] L. Xiujun, G. C. M. Meijer, Elimination of Shunting Conductance Effects in a Low-Cost Capacitive-Sensor Interface, IEEE Transactions on Instrumentation and Measurement 49 (3) (2000) 531-534.

[27] Z. Czaja, A measurement method for capacitive sensors based on a versatile direct sensor-to-microcontroller interface circuit, Measurement 155 (2020) 107547.

[28] Z. Czaja, Time-domain measurement methods for R, L and C sensors based on a versatile direct sensor-to-microcontroller interface circuit, Sensors and Actuators A 274 (2018) 199–210.

[29] Z. Czaja, A microcontroller system for measurement of three independent components in impedance sensors using a single square pulse, Sensors and Actuators A 173 (2012) 284-292.

[30] Z. Czaja, An implementation of a compact smart resistive sensor based on a microcontroller with an internal ADC, Metrology and Measurement Systems 23 (2016) 255-238.

[31] Innovative Sensor Technology IST AG, P14 Rapid Capacitive Humidity Sensor, Available at: https://www.ist-ag.com/sites/default/files/DHP14-Rapid_E.pdf.

[32] TE Connectivity Ltd., HS1101LF Relative Humidity Sensor, SENSOR SOLUTIONS /// HS1101LF HPC052_J (2015).

[33] Philips Components, Humidity sensor 2322 691 90001 Product specification, (1996).

[34] Atmel Corporation, 8-bit XMEGA A Microcontroller, XMEGA AU MANUAL, (2013), Available at: http://ww1.microchip.com/downloads/en/DeviceDoc/Atmel-8331-8-and-16-bit-AVR-Microcontroller-XMEGA-AU_Manual.pdf.

[35] Atmel Corporation, 8/16-bit AVR XMEGA A4 Microcontroller. ATxmega128A4, ATxmega64A4, ATxmega32A4, ATxmega16A4, (2013), Available at: http://ww1.microchip.com/downloads/en/DeviceDoc/Atmel-8069-8-and-16-bit-AVR-AMEGA-A4-Microcontrollers_Datasheet.pdf.

[36] K. Kolikov, G. Krastevy, Y. Epitropov, A. Corlat, Analytically determining of the relative inaccuracy (error) of indirectly measurable variable and dimensionless scale characterizing quality of the experiment, Computer Science Journal of Moldova 20 (58) (2012) 15-32.

[37] Agilent Technologies, Agilent 34410A/11A 6 1/2 digit multimeter user's guide (2012).

[38] Z. Czaja, A method of fault diagnosis of analog parts of electronic embedded systems with tolerances, Measurement 42 (6) (2009) 903-915.

[39] W. Toczek, Z. Czaja, Diagnosis of fully differential circuits based on a fault dictionary implemented in the microcontroller systems, Microelectronics Reliability 51 (8) (2011) 1413-1421.

Figures captions

Fig. 1. Scheme of a proposed direct sensor-to-microcontroller interface circuit.

Fig. 2. Timing of the signal $v_{out}(t)$ at the Pin 3 of the sensor interface circuit for V_{in} = 3.3 V, R_{r1} = 1 MΩ, R_{r2} = 10 MΩ, R_x = 10 MΩ, C_x = 100 pF.

Fig. 3. Block scheme of the compact smart lossy capacitive sensor based on the ATxmega32A4 microcontroller.

Fig. 4. The flowchart of the algorithm of the measurement procedure.

Fig. 5. Graph of the maximum relative uncertainty $dC_x(\alpha)$ as a function of the voltage ratio $\alpha = v_1/V_{in}$.

Fig. 6. Graph of functions $t_1(R_{r1}/R_{r2})$ and $(t_2 - t_1(R_{r1}/R_{r2}))$ for R_{r2} = 10 MΩ.

Fig. 7. Circuit model of a direct capacitive sensor-to-microcontroller interface circuit.

Fig. 8. Simplified circuit model of the interface circuit from Fig. 6 for the step II of the measurement procedure.

Fig. 9. Maximum absolute uncertainties Δt_1 and Δt_2 of discharging times t_1 and t_2 determination, where Δt_1 is plotted for $\Delta c_{DAC,1}$ = 4, and Δt_2 for $\Delta c_{DAC,2}$ = 12.

Fig. 10. Relative uncertainties $dR_x(t_1)$ and $dR_x(t_2)$ of R_x determination.

Fig. 11. Relative uncertainties $dC_x(t_1)$ and $dC_x(t_2)$ of C_x determination.

Fig. 12. Prototype of the proposed circuit for interfacing the lossy capacitive sensor.

Fig. 13. Graphs of the measurement results t_1 and t_2 of scaling of the interface circuit.

Fig. 14. Absolute measurement errors Δt_1 and Δt_2 of times t_1 and t_2 for a single measurement approach.

Fig. 15. Relative systematic measurement errors dC_m and dR_m in the measurements of C_x and R_x values.

Fig. 16. Relative random measurement errors dC_{noise} and dR_{noise} in the measurements of C_x and R_x values for a single measurement approach.

Fig. 17. Relative random measurement errors dC_{noise} and dR_{noise} in the measurements of C_x and R_x values for a multiple measurement approach (M = 64).

Fig. 18. Scaling curves for C_x plotted for the R_x parameter (left graphs) and scaling curves for R_x plotted for the C_x parameter (right graphs): a) in three-dimensional space, b) on the plane with coordinates t_1, t_2.

Fig. 19. Scaled mesh on the plane with coordinates t_1, t_2, where the C_x curves are in picofarads, and the R_x curves in megohms

Fig. 20. Sets of curves: a) $C_{x,j} = f_j(C_{m,i,j},R_{m,i,j})$ in a space with coordinates C_m, R_m, C_x, b) $R_{x,i} = f_i(C_{m,i,j},R_{m,i,j})$ in a space with coordinates C_m, R_m, R_x.

Fig. 21. Graphs of coefficients: a) A_1, A_2 in a function of R_m, b) B_1, B_2, B_3 in a function of C_m.

Fig. 22. Relative systematic measurement errors dC_m and dR_m in the function of C_m and R_m values after software correction.

Table captions

Table 1. State of pins 1, 2 and 3 and resulting discharging times t_1 and t_2.

Table 2. A set of capacitors used to emulate the capacitance C_x of the sensor.

Table 3. A set of resistors used to emulate the shunt resistance R_x of the sensor.

Table 4. Comparison of the proposed method with the modified 3-point calibration technique.

Figure 1

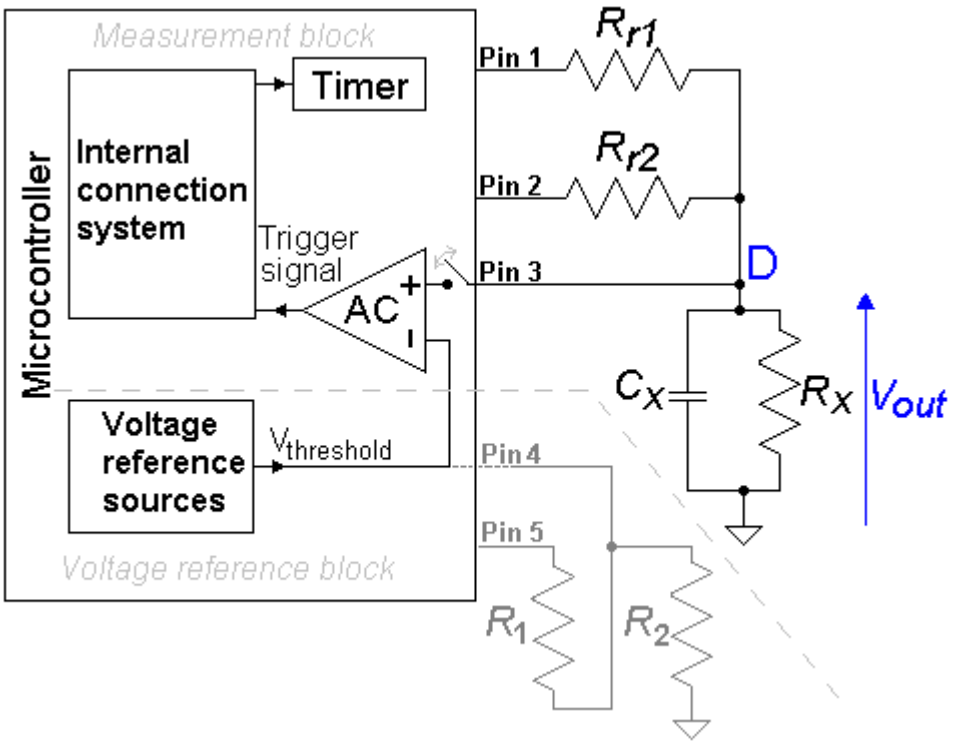


Figure 2

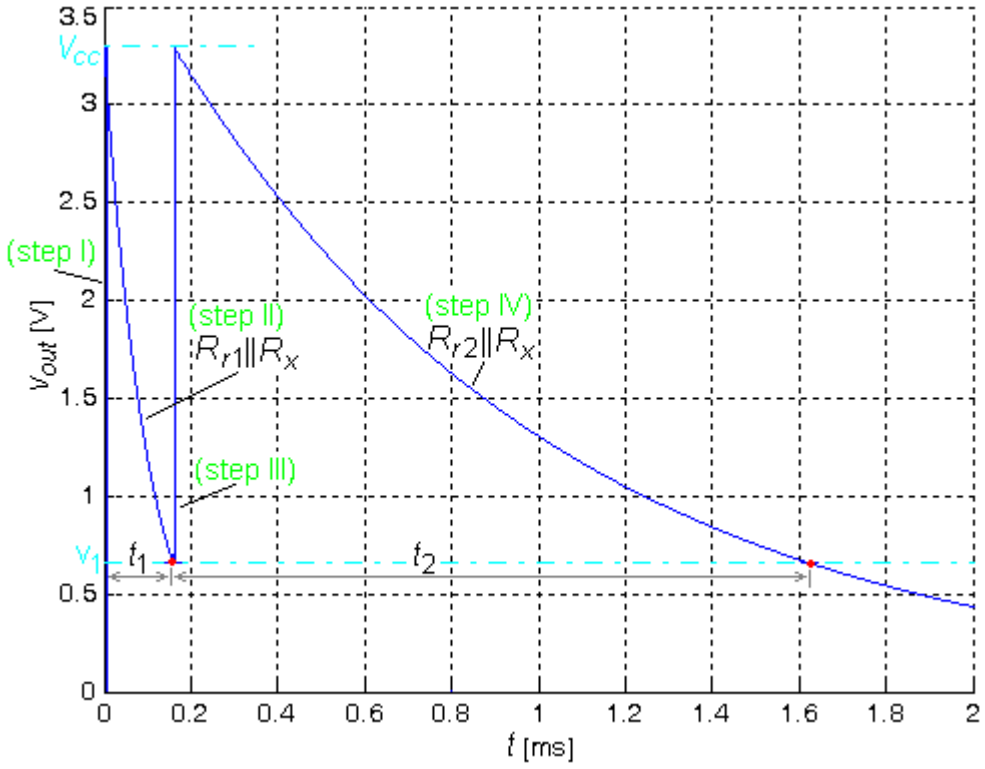


Figure 3

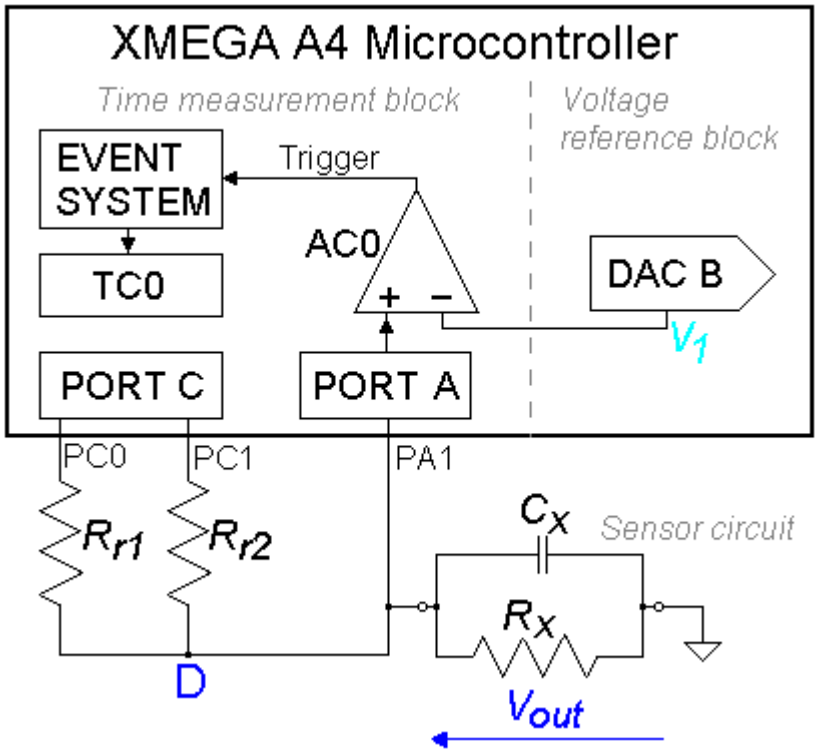


Figure 4

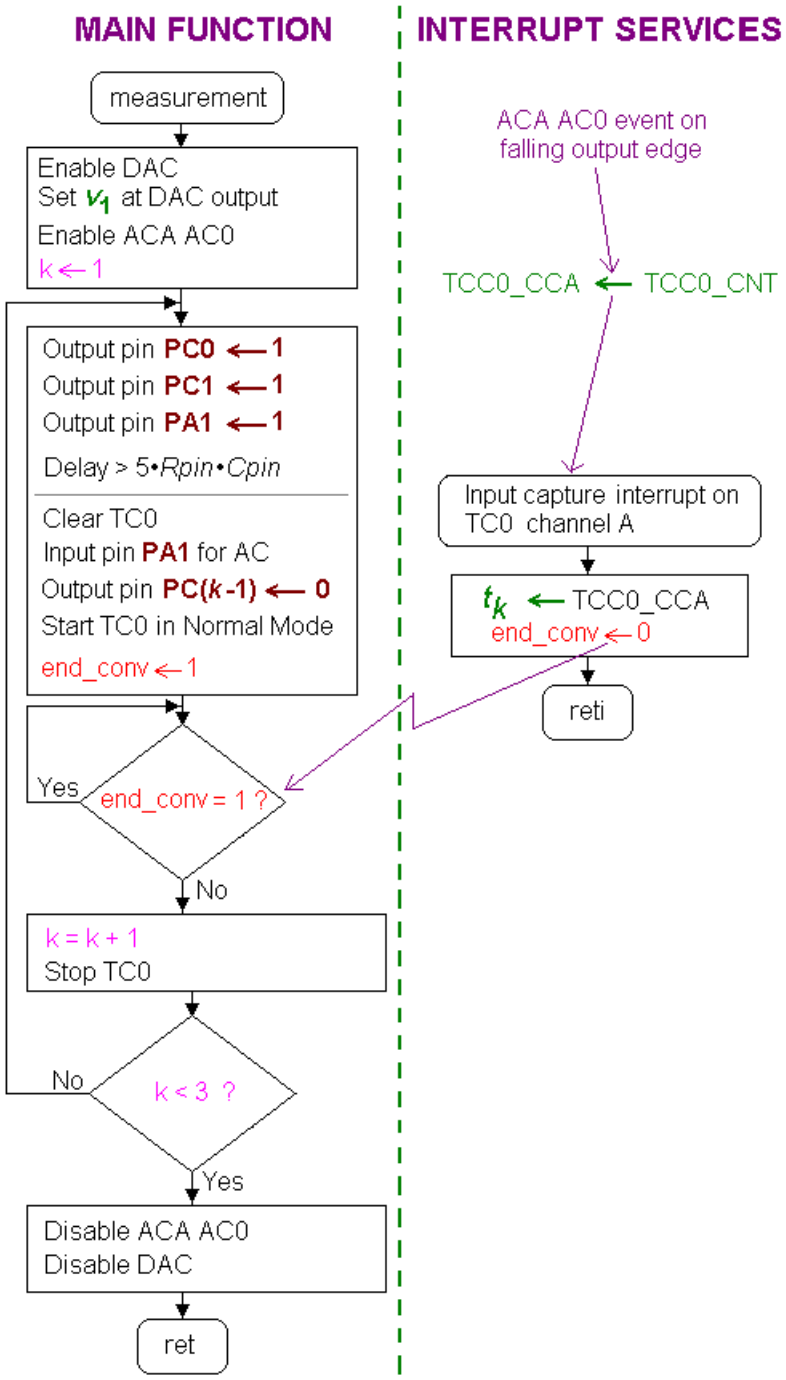


Figure 5

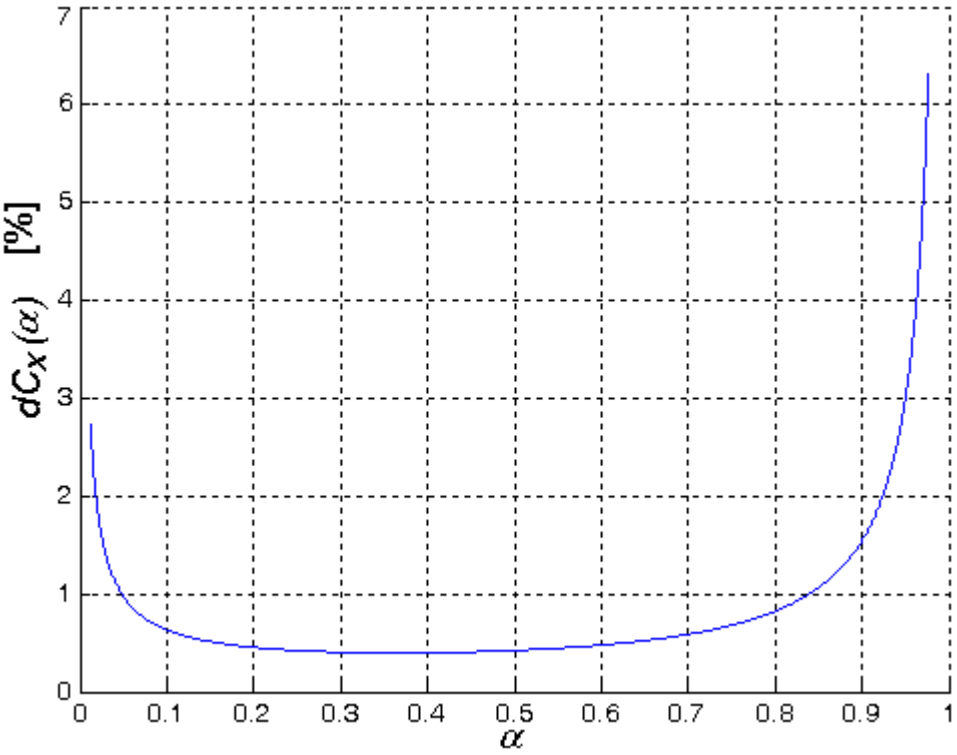


Figure 6

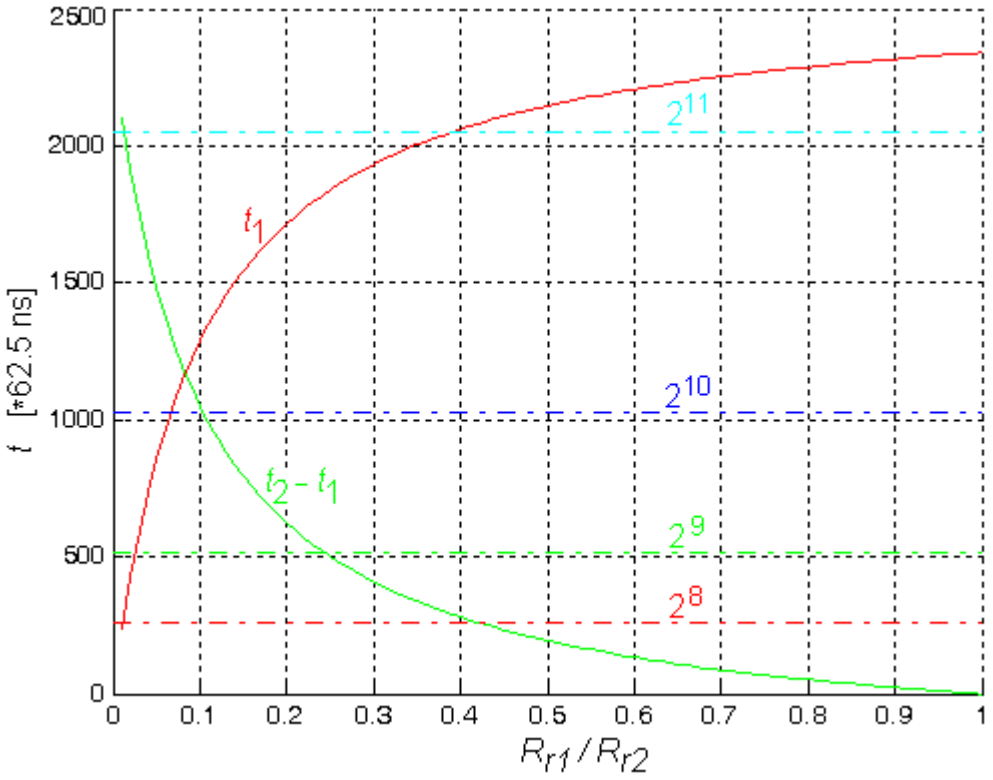


Figure 7

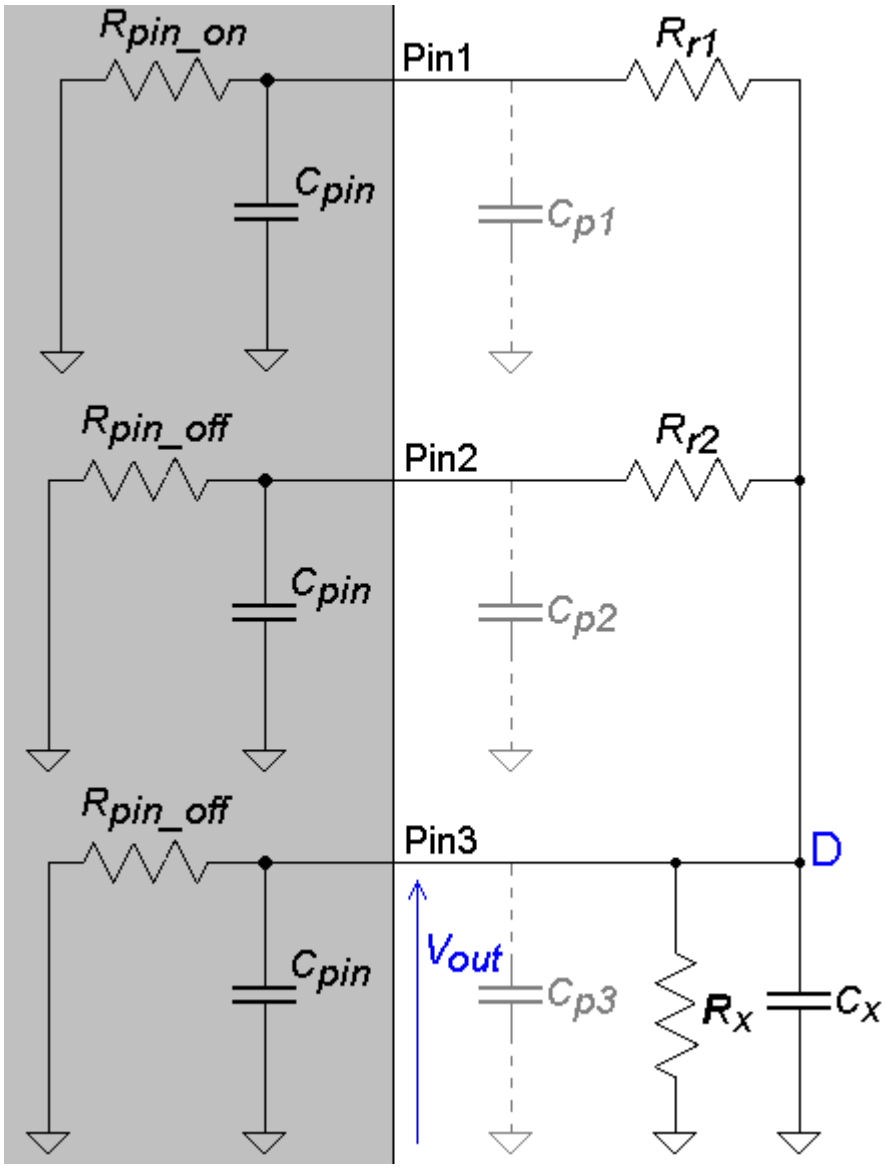


Figure 8

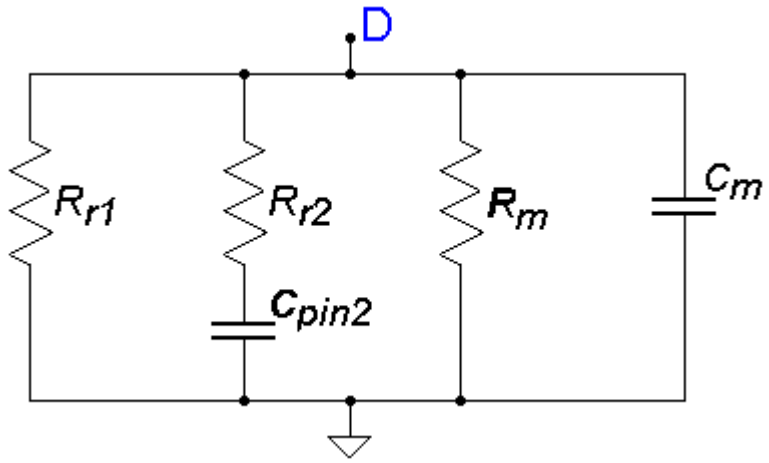


Figure 9

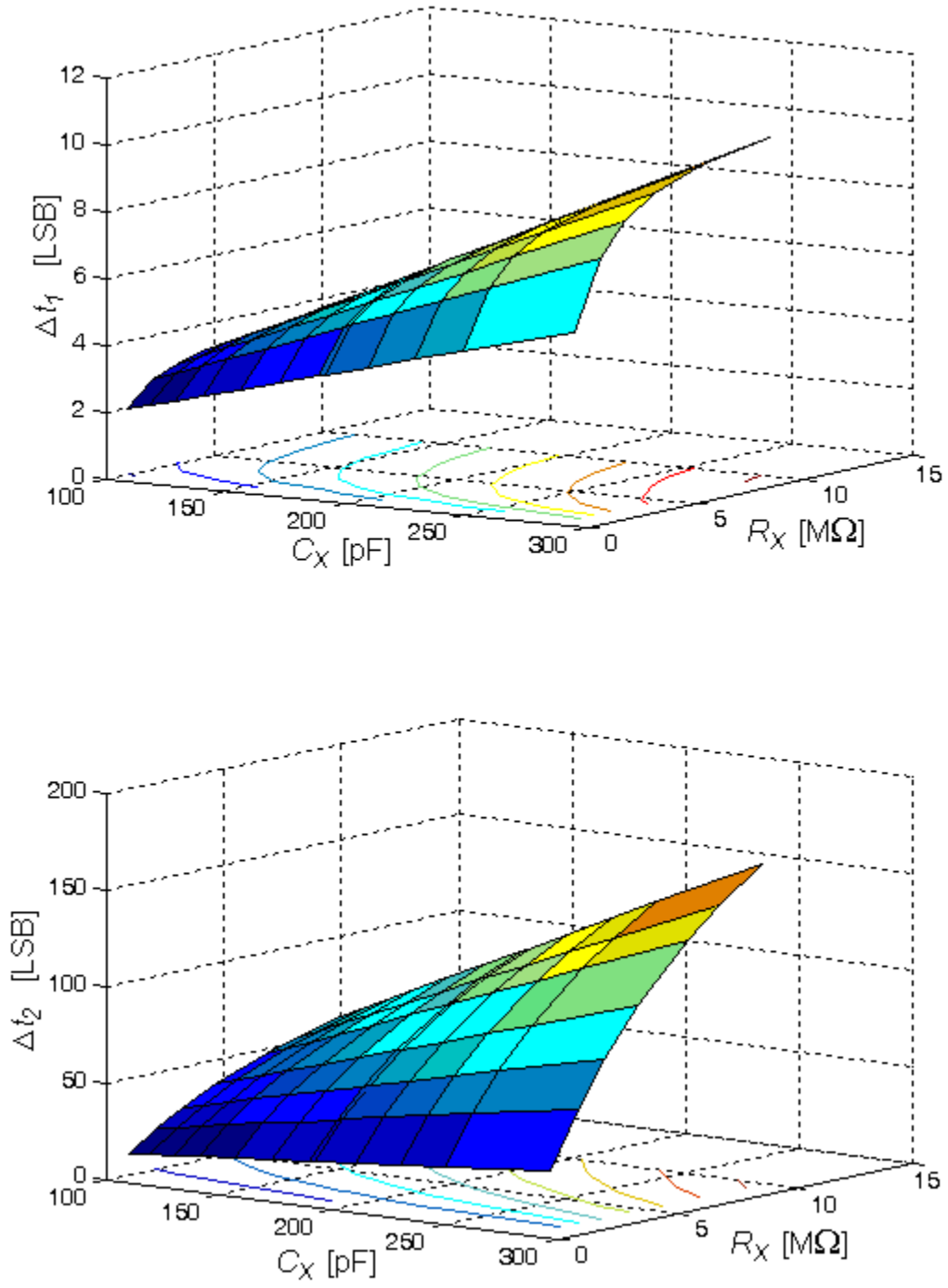
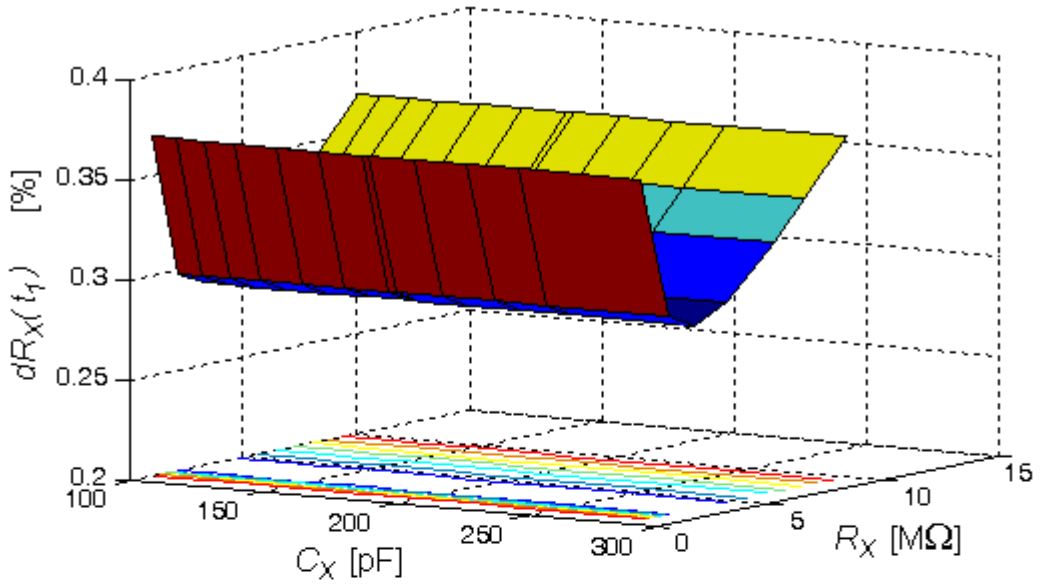


Figure 10



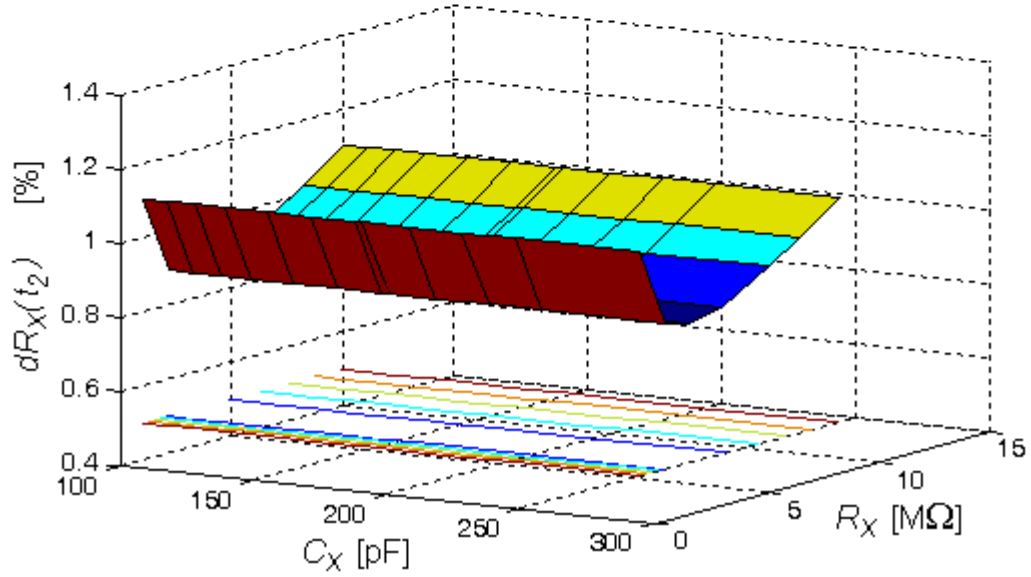


Figure 11

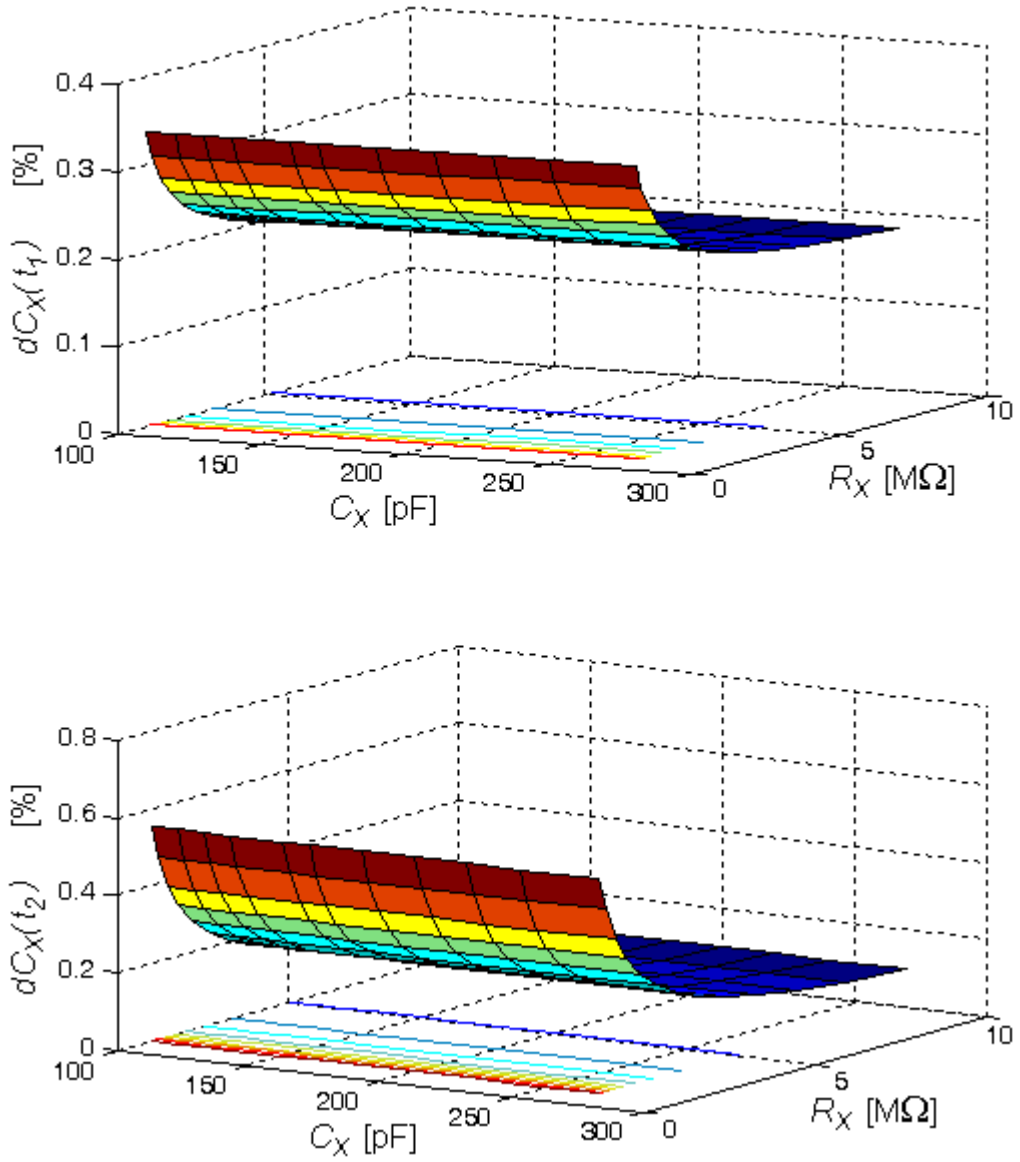


Figure 12

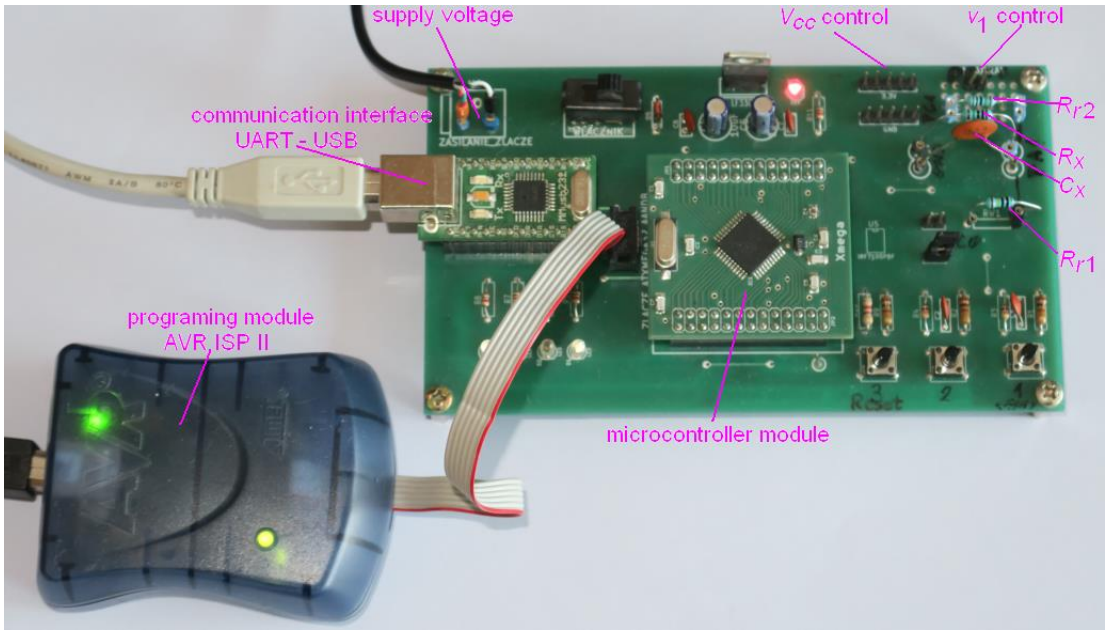


Figure 13

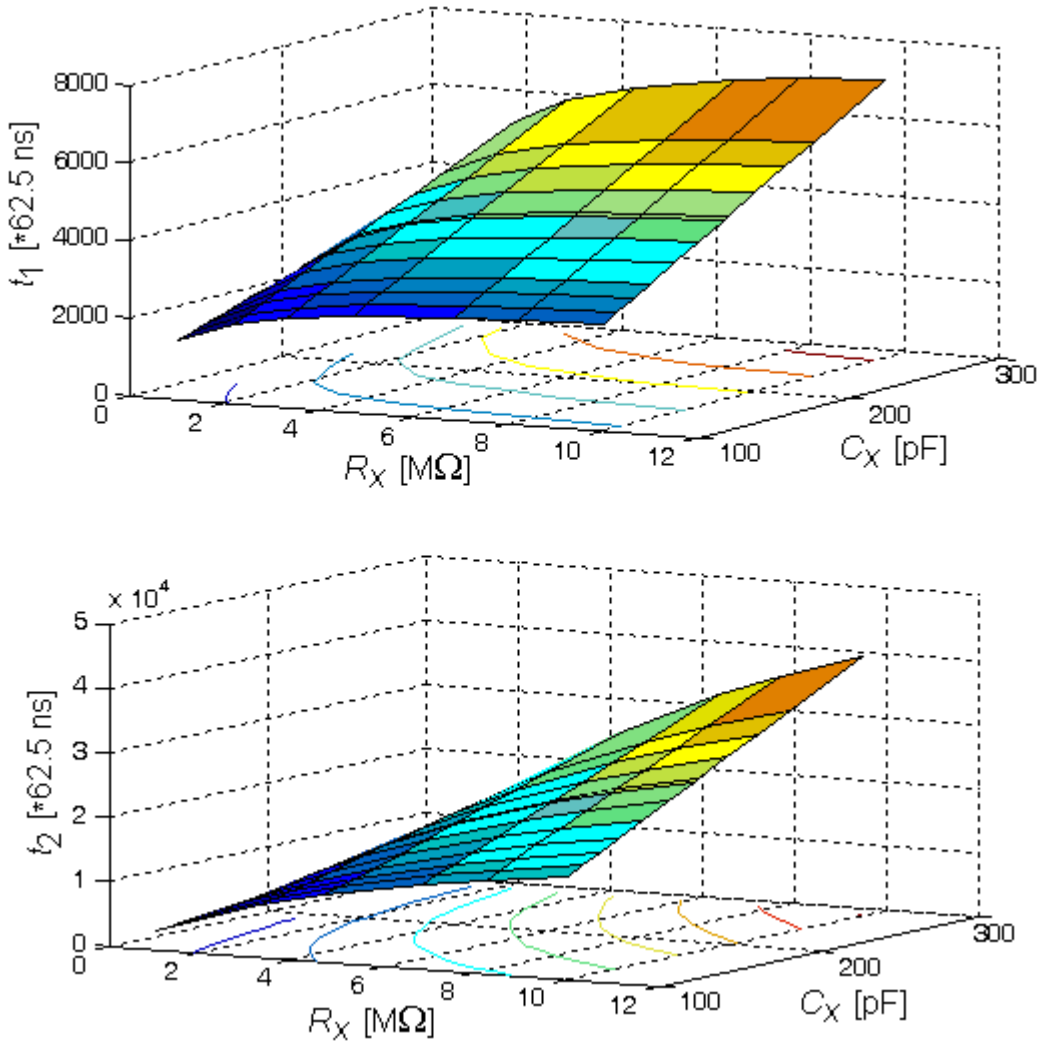


Figure 14

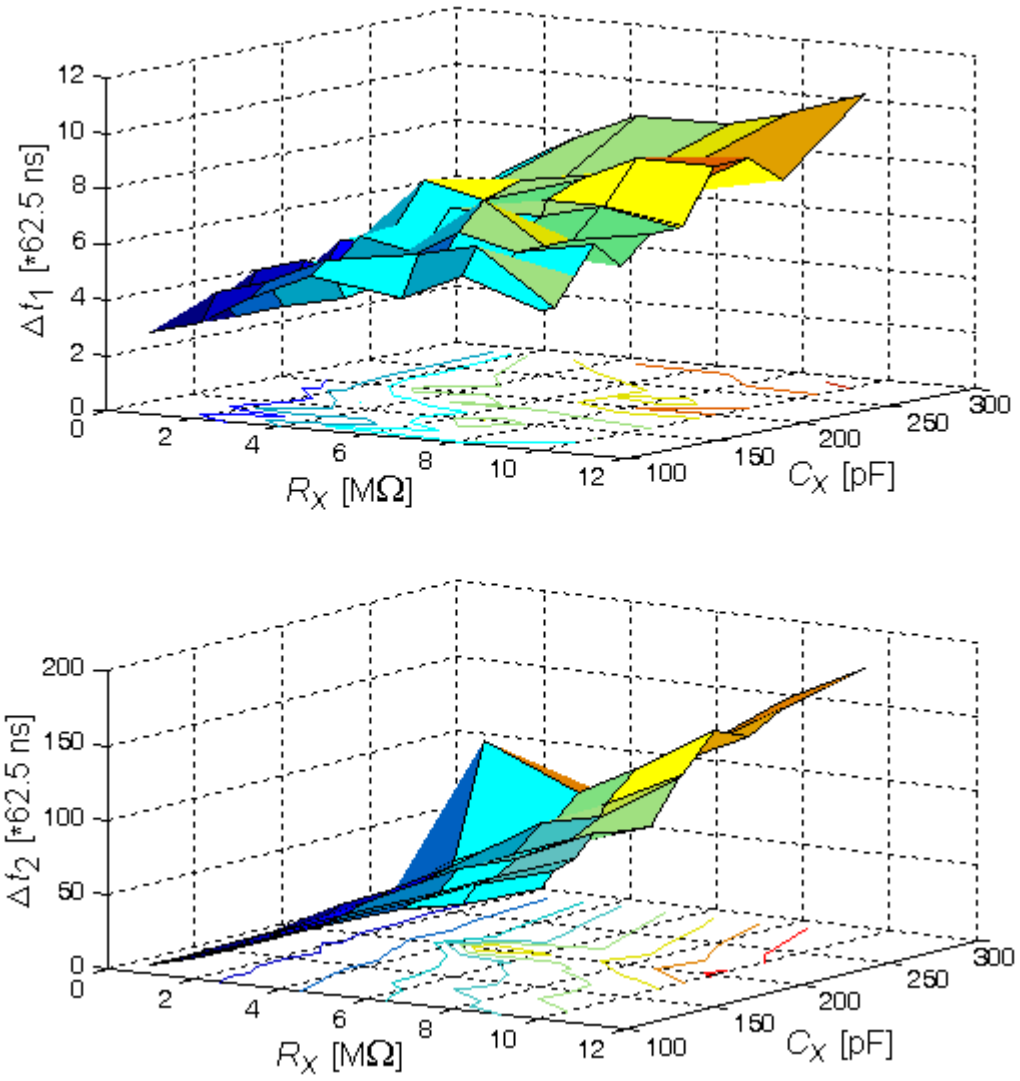
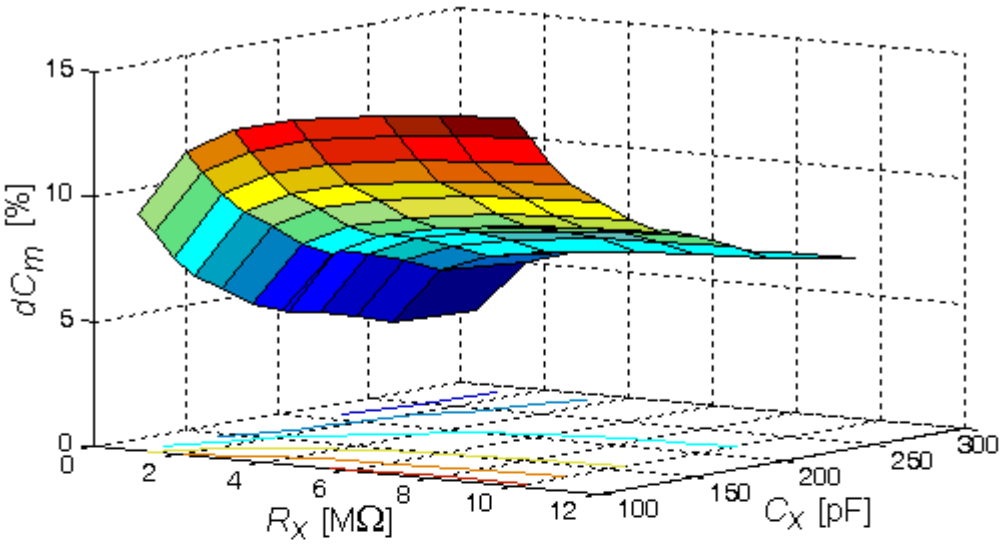


Figure 15



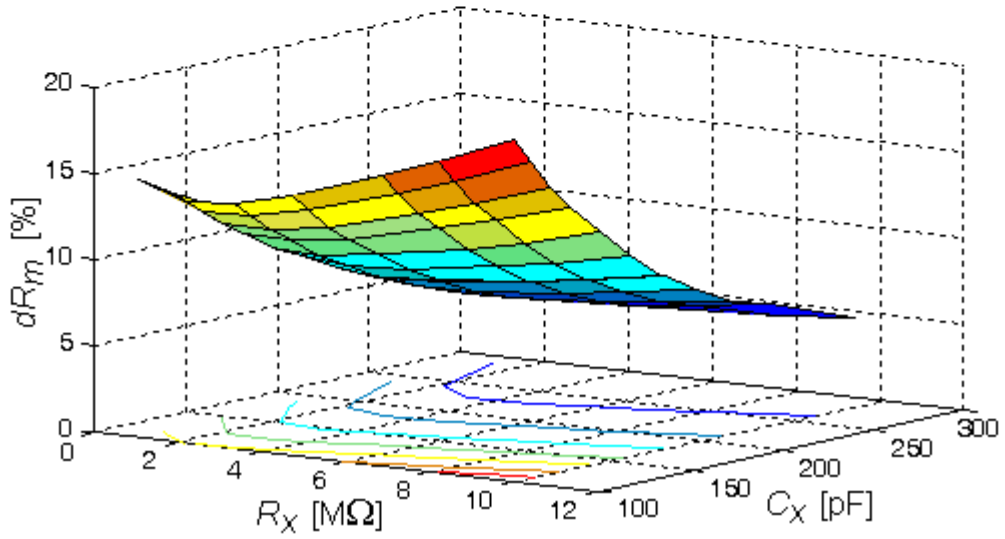


Figure 16

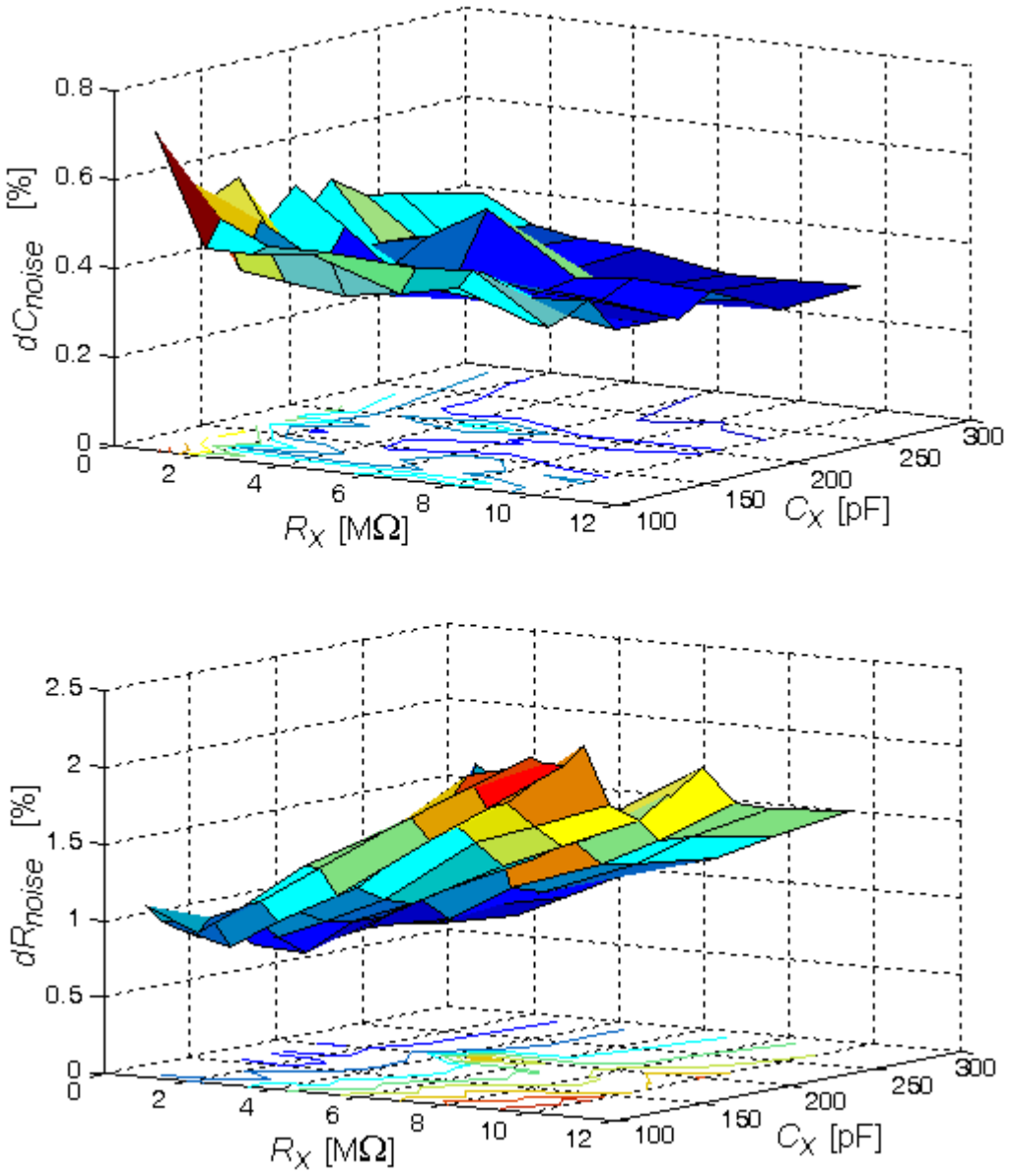
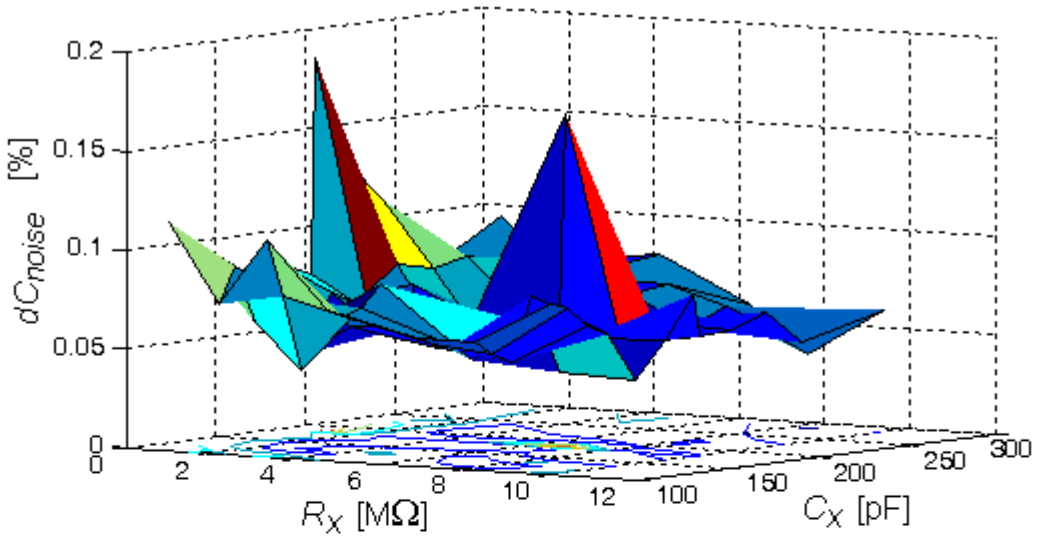


Figure 17



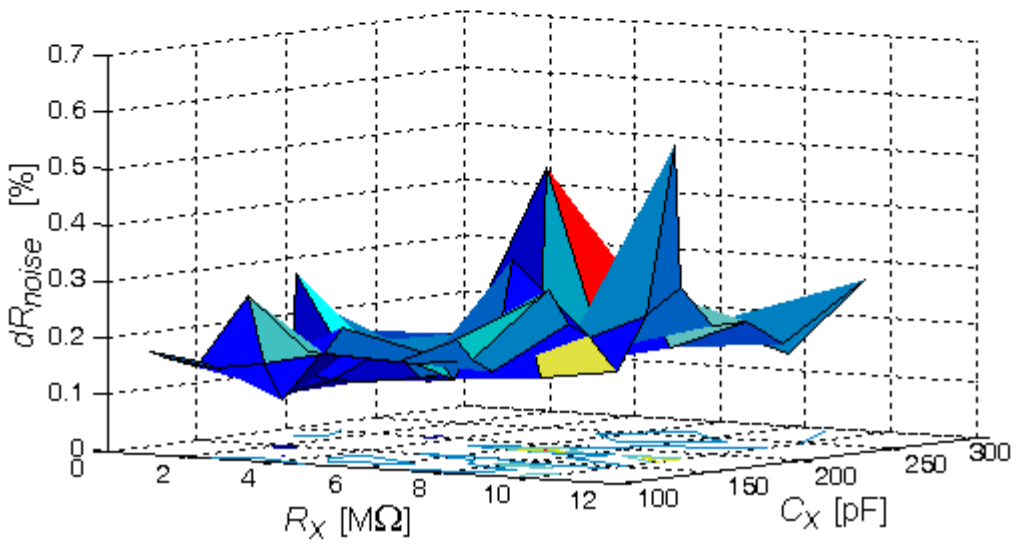


Figure 18

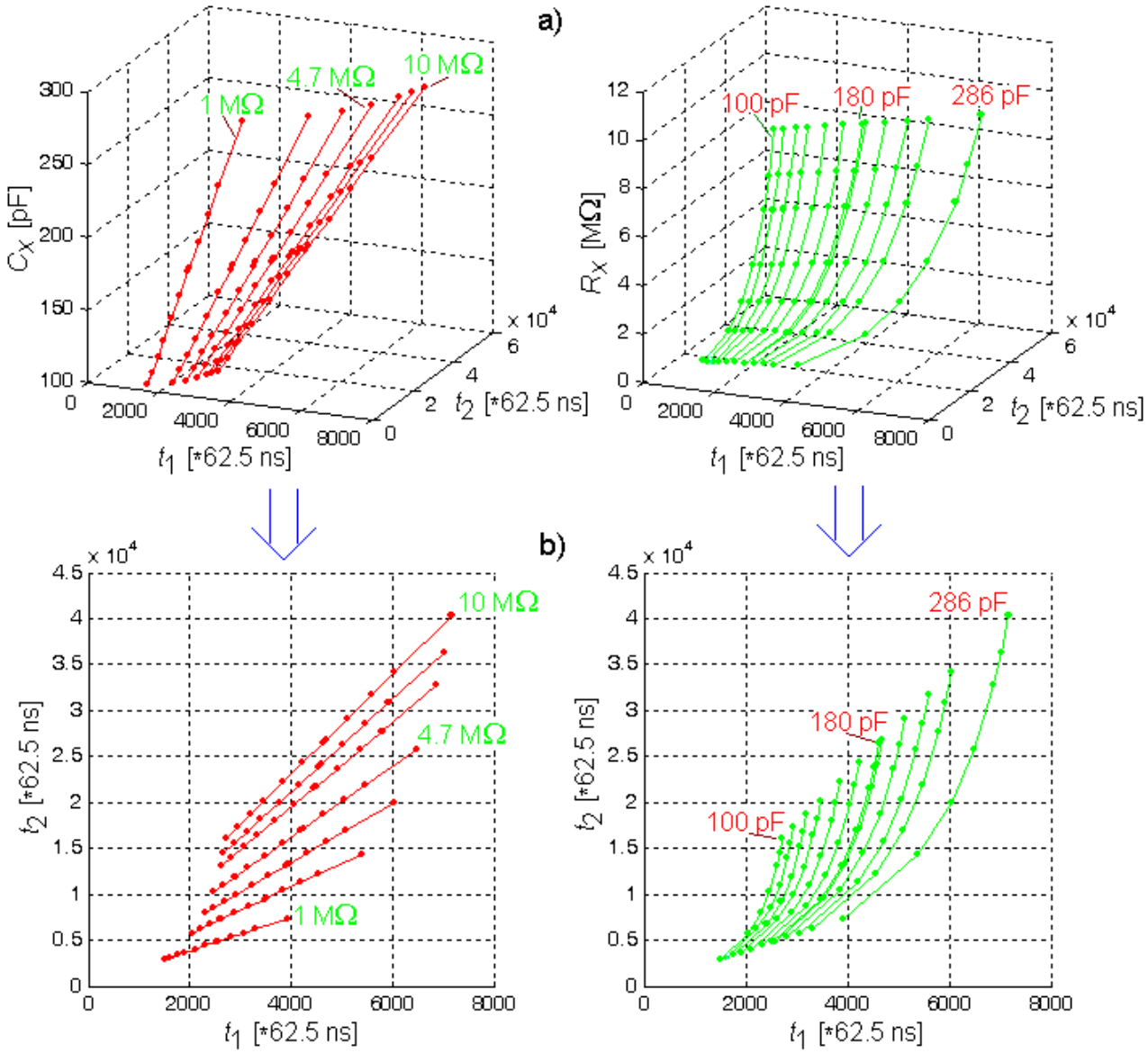


Figure 19

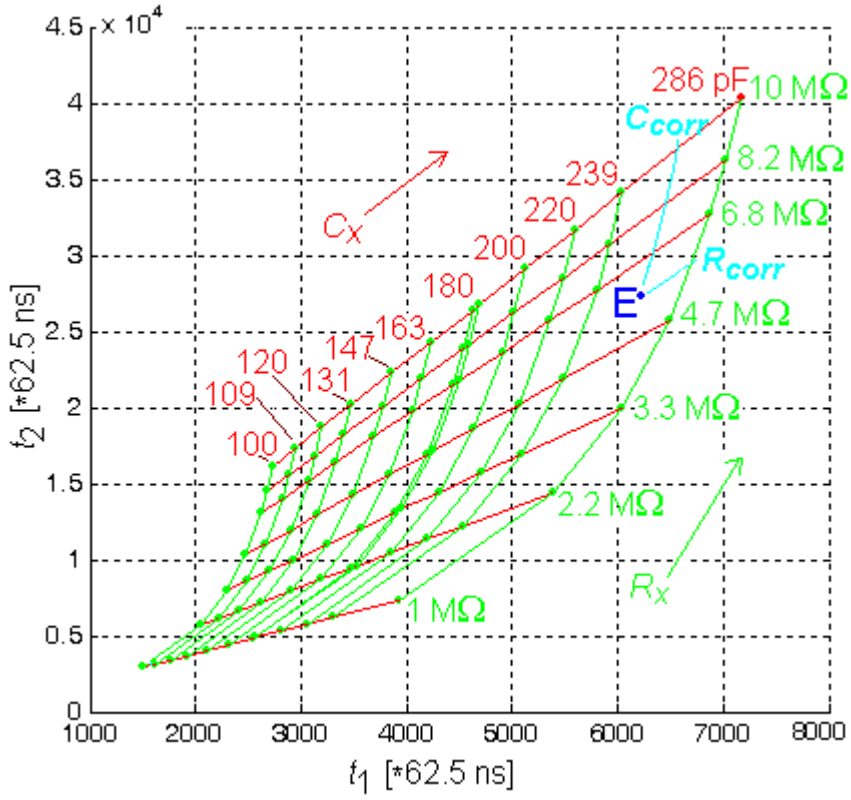


Figure 20

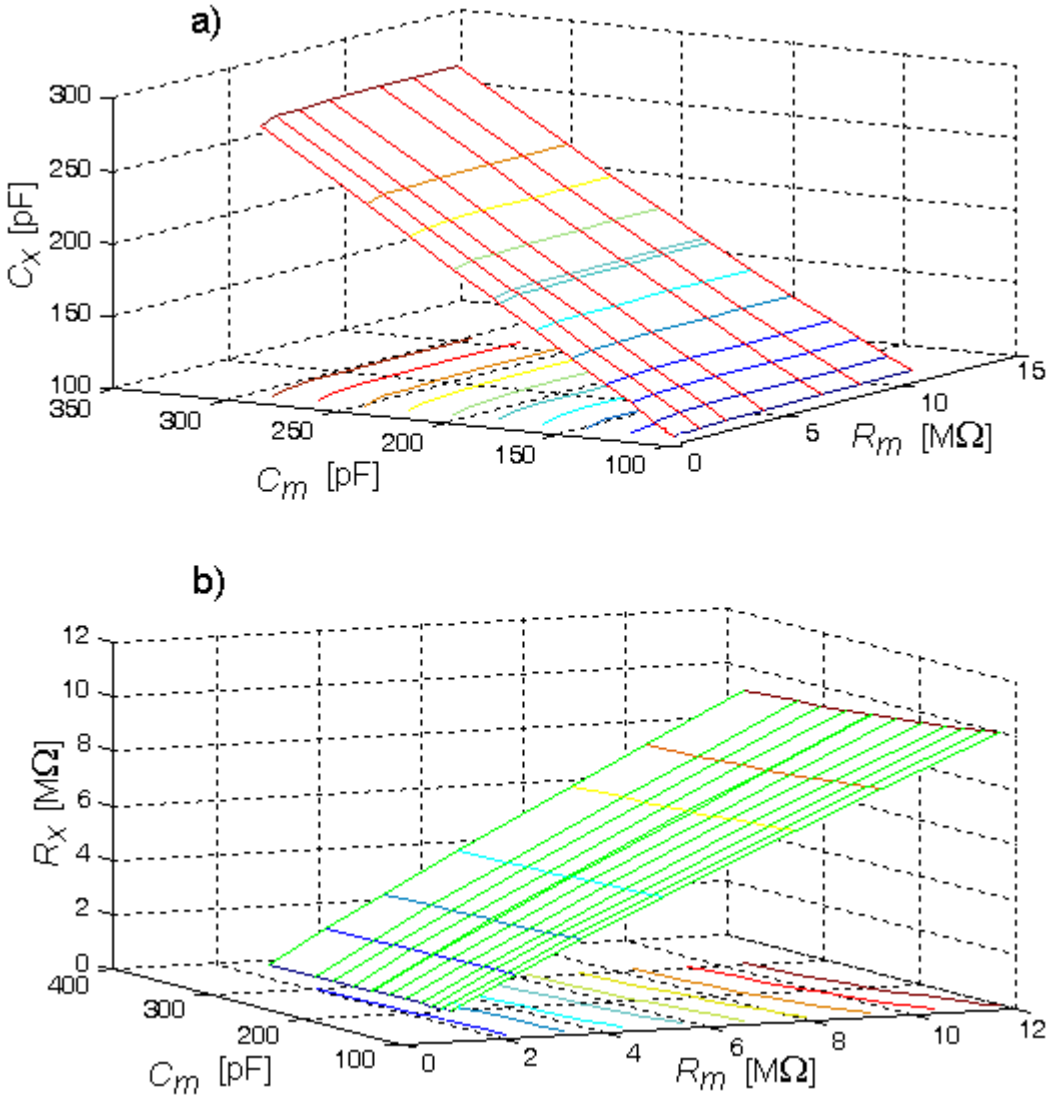


Figure 21

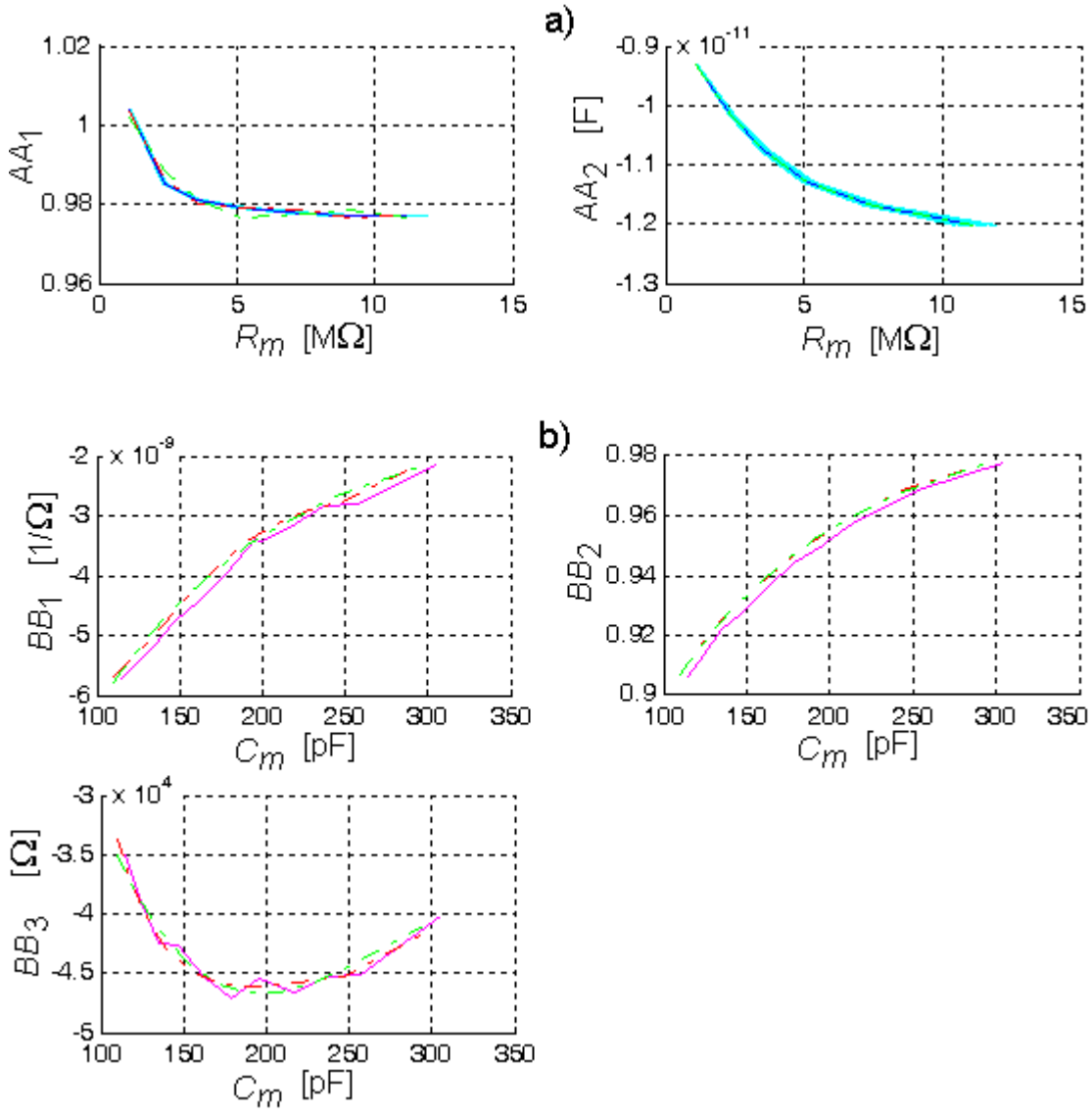


Figure 22