Chapter 1
Improved PWM A/D conversion technique: working principle and model validation

José Miguel Costa Dias Pereira[1,2]

Abstract

Analog-to-digital conversion plays a central role in any application of digital sensors and sensor systems that require an interface between analog devices, namely analog sensors, and digital devices, namely, microprocessors, digital signal processors or microcontrollers. With the advent of smart sensing, the integration of signal conditioning, analog-to-digital and digital data processing in single hardware devices became a reality. Moreover, the usage of low-cost discrete A/D conversion techniques for applications that are not critic in terms of accuracy, resolution or conversion rate, are considering increasingly mixed hardware and software A/D solutions tailored for specific application demands. In this context, this chapter presents a discrete low-cost A/D conversion solution based on pulse width modulation particularly suited for microcontrollers' integration with smart sensing devices.

1. Introduction

The development that has taken place in the area of microelectronics has allowed a gradual implementation of the functions of acquisition, signal conditioning and signal processing, on a common integrated circuit. Nowadays, smart transducers and actuators include functions that go well beyond those of their classical counterparts. These new transducers, usually denominated as smart transducers, are no longer simple interfaces for the inputs and outputs of the measurement system and implement an increasing number of functions supported by integrated processing units, namely, microcontrollers (μC) and digital signal processors (DSP). The new functions and capabilities associated with smart transducers include self-test, auto-calibration, auto-

[1]ESTSetúbal/IPS, Setúbal, Portugal
[2]Instituto de Telecomunicacões, Lisboa, Portugal

identification, errors' correction, auto-ranging, selection of engineering units, data logging, adaptability, A/D conversion flexibility, among others.
However, the integration of sensors, signal conditioning and signal processing functions in a single integrated circuit (IC), using *micromachining technologies* (MEMS), create new challenges, such as, the ones that results from the interferences between analog and digital signals.
Another limitation that results from the integration of sensing and signal processing units is related with the low accuracy of the analog integrated components, whose absolute variation can be as high as 50% of the nominal value of the components, although, this variation, in relative terms, of components' values with the same nominal value is typically much lower (0.1%).
Another relevant issue that affects measurement accuracy is related with the influence variables effects that affect directly A/D conversion accuracy. These effects must be compensated by measuring the influence variables or, more wisely, by using methods that automatically cancel the negative impact of those variables [1].
Considering the relevance of these issues, this chapter focus a particular type of discrete A/D conversion techniques based on the usage of pulse width modulated (PWM) signals [2-4]. Different PWM modulation schemes are analyzed and an improved PWM based A/D conversion method (PWM), based on the number of pulses counting (NPC), is presented. The originality, simplicity, and flexibility of this method are underlined and the advantages for smart sensor linearization and for the dynamic adjustment of the conversion range will be focused in a second chapter of this book that is dedicated to simulation and experimental results that can be achieved by using the proposed PWM-NPC A/D conversion method. The proposed conversion technique can be implemented easily using any low-cost microcontroller that includes capture, compare and PWM (CCP) modules.
Considering the main characteristics and advantages associated with discrete A/D conversion techniques, it can be referred that these conversion techniques are associated with very simple working principles that use discrete components and low-cost processing devices that are already included in the majority of smart sensing systems. These conversion techniques are adaptive and enable an easy establishment of compromises between different A/D conversion performance parameters, such as, accuracy, resolution and conversion speed, that could be essential in applications that demand very low power consumption levels, such as the ones that use wireless sensors [5-6].
This chapter also includes several simulation and experimental results that validate the working principle of the A/D converter and confirms its design parameters. A second chapter, of this book that must be read after this one presents experimental results related with the application of the proposed A/D conversion method for linearization of transducers characteristics being the achievements, in this area, recognized by several scientific works published by the author [7-9].

2. Basics of PWM A/D conversion

The microcontroller device has a widespread usage in smart sensing systems. Thus it is important to take advantage of its potential to perform A/D conversion tasks. The implementation of discrete A/D conversion can be achieved by using exclusively digital input and output ports of the microcontroller. It becomes possible to implement low cost A/D conversion techniques, using only digital input and output lines of the microcontroller together with passive resistors and capacitors (RC) components and elementary conversion algorithms with low demands on microcontroller's computational load or memory requirements. Of course, this solution has its own limitations, particularly in terms of conversion rate, but cost and flexibility, in terms of accuracy, resolution and conversion speed, are a significant advantage in a large number of smart sensing applications when low cost implementation is crucial to justify their economic viability.

2.1 Traditional PWM based A/D conversion

Traditional PWM and elementary based A/D conversion requires only the generation of a digital PWM signal with adjustable duty-cycle (DC). The main elements of a discrete PWM based A/D conversion includes the PWM signal generated by the microcontroller (V_{PWM}), the low pass filter (LPF) and a voltage comparator (C).

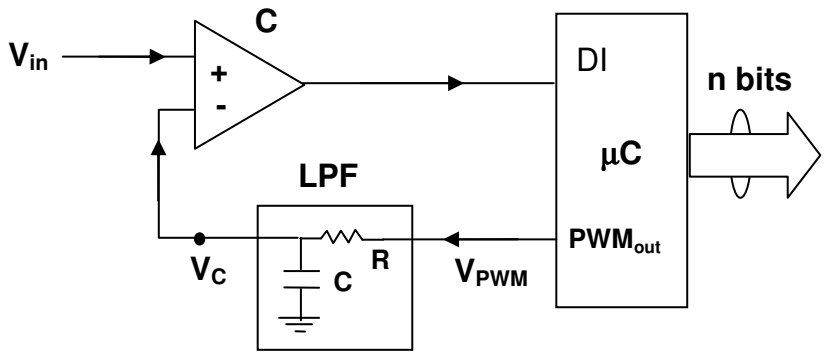

Fig. 1. Main elements and signals of a discrete PWM based A/D converter.

The average value of the V_{PWM} signal represented in figure 2, which is obtained at the output of the LPF, is given by:

$$V_C = \frac{1}{T}\int_T V_{PWM}(t)\cdot dt = \frac{T_2}{T}V_L + \frac{T_1}{T}V_H \tag{1}$$

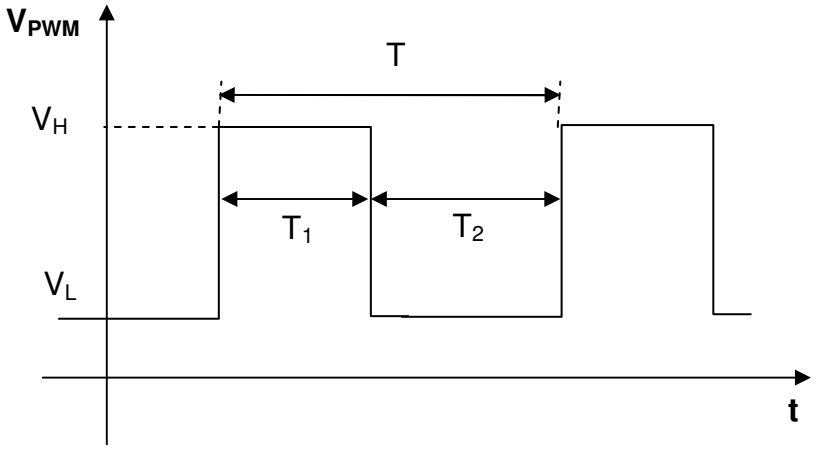

Fig. 2. V_{PWM} signal (T- period; T1/T- DC; VL- low voltage amplitude; VH- high voltage amplitude).

In order to obtain an output voltage (V_C) almost constant, during the conversion cycle, the value of the RC constant should be much greater than the period of the PWM signal (RC>>T). Under these conditions, the filter output voltage is approximately given by:

$$V_C = \frac{(1-DC)}{100}\cdot V_L + \frac{DC}{100}\cdot V_H \tag{2}$$

where DC is the duty cycle (*"duty cycle")* of the PWM signal defined in percentage terms by:

$$DC = 100\cdot\frac{T_1}{T} \tag{3}$$

Assuming, for simplicity of analysis, that V_L= 0 and V_H= $V_{REF,}$ the input voltages of the comparator are equal if:

$$V_{in} = \frac{DC}{100}\cdot V_{REF} \tag{4}$$

As long as the end of conversion (EoF) condition is given by the equalization of the input voltages of the comparator, the digitalization result of the input voltage is associated with the DC of the PWM signal and the full-scale (FS) range of the A/D converter is equal to V_{REF}. Two features result directly from the operating principle of this type of converter. One concerns the range of conversion that should be between V_L and V_H and the other concerns the resolution of the converter, which is directly

associated with the setting resolution of the DC coefficient. It is also important to refer that if RC>>T the conversion time will be, obviously, large.
Each average value of the PWM voltage, for each value of the DC coefficient, obtained at the low-pass filter (LPF) output, corresponds to the A/D converter transition voltages. For an n bit converter there are 2^n-2 values of transition voltages, being the quantization step (Q) of the converter equal to the difference between two adjacent transition voltages. Using an optimistic assumption that there is a negligible value of the RC time constant, the conversion time (T_C) that is associated with a sequential search of all transition voltages is given by:

$$T_C = \left(2^n - 2\right) \cdot 2^n \cdot T_{CLK} \cong 4^n \cdot T_{CLK} \qquad (2^n >> 2) \tag{5}$$

where T_{CLK} represents the timing resolution of the V_{PMW} DC signal.
Figure 3 represents the conversion time of the A/D converter as a function of the converters' number of bits for two different values of the temporal resolution of the microcontroller.

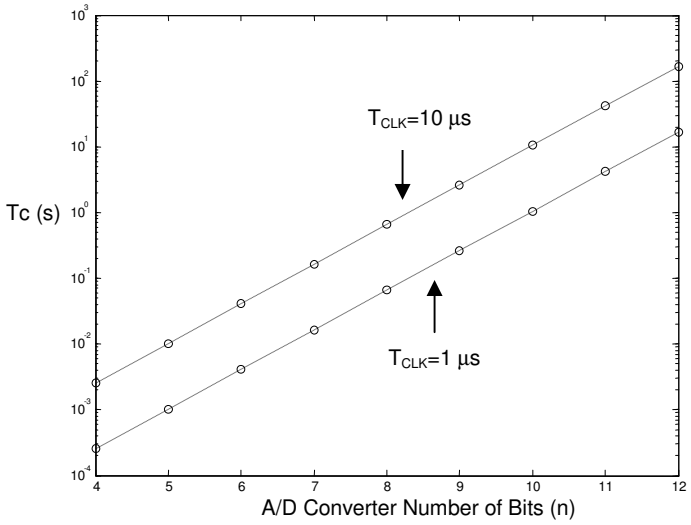

Fig. 3. Conversion time (T_C) of the A/D converter for two different values of the clock signal period (T_{CLK}=1 µs and T_{CLK}=10 µs) when the A/D converter number of bits varies between 4 and 12.

It is clearly visible that the growth of the conversion time is exponential with the converter's number of bits, reaching easily conversion times (T_C) about one second for typical resolution values around 8 bits.
In order to obtain an output LPF voltage (V_C) fluctuation (*"ripple"*) less than 1 LSB, the following condition must be fulfilled:

$$e^{-\frac{1}{RC}\cdot T} < \left(1-\frac{1}{2^n}\right) \tag{6}$$

where T represents the period of the PWM signal and n represents the number of bits of the ADC.

Figure 4 represents the relationship between V_C voltage ripple and the ratio between the time constant (RC) and the PWM signal period (T) for a converter with 8 bits (n = 8). From the figure, it can be verified that to obtain an 8 bit converter, the ratio RC/T must be greater than 255.

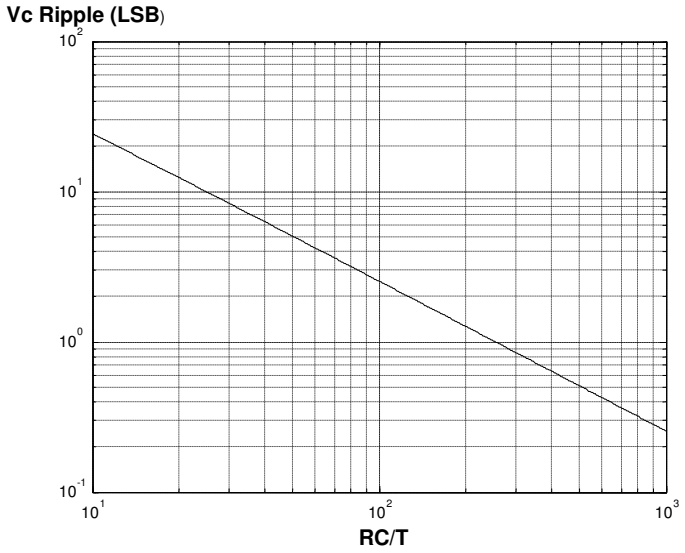

Fig. 4. Relationship between the filter output voltage ripple and the ratio between the time constant (RC) and the PWM signal period (T) for an 8-bit converter.

Based on (6), it is possible to obtain the expression of the minimum value of the time constant (RC_{min}) of the LPF that is given by:

$$RC_{min} = \frac{2^n \cdot T_{CLK}}{ln\left(\frac{2^n}{2^n-1}\right)} \tag{7}$$

From the previous expressions it can be concluded that the usage of this conversion technique is only acceptable for input signals V_{in} with a very low bandwidth. Moreover, the main factors that affect the accuracy of this method are associated with the offset voltage of the comparator and with the impedance of the RC load at the digital output of the microcontroller. The first problem can be minimized with the usage of a precision comparator, or by using self-calibration techniques [10-18] and

the second problem can be minimized by using a voltage follower (*"buffer"*) at the interface between the output of the microcontroller and the LPF input.
By the other hand, as a great advantage of this A / D conversion technique it can be mentioned its simplicity and low cost of implementation that are magnified if the microcontroller has a PWM output port, an integrated comparator and dedicated instructions for programming PWM signals.

3. Improved PWM based A/D conversion

This section describes the operation principle of the PWM NPC A/D converter including simulation and experimental results. Dimensioning of design parameters of the converter is also addressed.

3.1 Block diagram and principle of operation

The working principle of this conversion technique is substantially different from the one described previously. As a matter of fact, the principle is quite similar to the sigma-delta conversion technique [19-22] but the converter feedback loop includes the processor device, usually a microcontroller, and charging/discharging RC circuit that receives high and low voltage pulses from the processor device. Recent advances in technology enable the usage of Arduino [23-24] and ESP [25-26] based solutions to take advantage of their increased gains in the manipulation of the resolution and frequency of PWM signals.
Figure 5 represents the basic circuit diagram of this converter and it will be demonstrated that its conversion time is much lower than the one obtainable with the traditional PWM A/D converter, previously described. From now on, the improved PWM A/D conversion technique will be denominated as PWM A/D conversion based on the number of pulses counting (PWM NPC A/D).

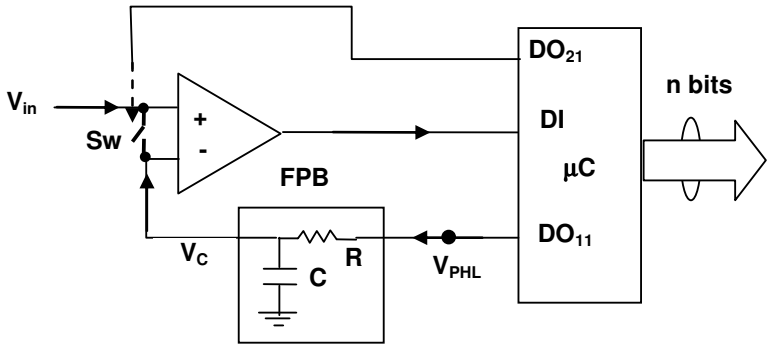

Fig. 5. Circuit of a PWM NPC A/D converter (initialization SW ON: V_{C0}=Vin).

In this variant of the PWM based A/D conversion method the RC time constant is scaled for an incremental voltage variation (ΔV_C) during the time of each pulse V_{PHL} generated at the output of the microcontroller to have an absolute value about 1 LSB. Regarding the working principle of the PWM NPC A/D, it can be described as follows. After the start of conversion (SoC) trigger, the capacitor is initialized with the value voltage to be converted (V_{in}) and during the conversion cycle V_{PHL} is setup by the microcontroller according to the output signal delivered by the comparator. If V_{in} is higher than V_C, V_{PHL} is set to its high voltage value (V_H), otherwise, if V_{in} is lower than V_C, V_{PHL} is set to its low voltage value (V_L). At the end of the conversion (EoC) cycle the total number of high pulses (m) and low pulses (p) is directly related with the result of the input signal digitalization.

To analyze in a greater detail the working principle of this converter, several simulation routines were developed in MATLAB. Three simulation cases will be presented and their results analyzed. In the simulation a constant voltage (V_{in}) will be considered with a normalized amplitude, relatively to the full-scale range (FS) of the converter, varying in the [-1; 1] interval with increments equal to 0.5 LSB units. Different values of the ratio between the charging and discharging pulses (T_P) and the RC circuit time constant will be considered. In all the simulations, and for convenience of graphical interpretation of the results, it is considered an hypothetical ideal converter with 5 bits being the conversion results given by the number of charging pulses (p), with amplitude equal to V_H, and the number of discharging pulses (m), with amplitude equal to V_L, that occur during the conversion cycle containing a total of 32 (2^n) charging or discharging pulses.

Assuming an ideal behavior of the PWM NPC A/D converter ($V_{ADC} \cong V_{in}$), it is possible to obtain the following relationship:

$$V_{ADC} = \frac{p \cdot V_H + m \cdot V_L}{p + m} \tag{8}$$

where V_{ADC} represents the voltage amplitude of the digitized voltage, p and m represent the number of charging and discharging pulses, with amplitude equal to V_H and V_L, respectively.

Since the FS range limits are associated with V_L and V_H, the quantization step amplitude is given by:

$$Q = \frac{V_H - V_L}{2^n} \tag{9}$$

where n represents the number of bits of the converter.

By its turn, the normalized quantization error, in LSB units, is given by:

$$E_Q = \frac{V_{ADC} - V_{in}}{Q} \tag{10}$$

3.2 Expression of the converter characteristic function

Based on the block diagram represented in Figure 5, it can be concluded that the feedback loop, comprising the comparator, the microcontroller and the low-pass filter, acts so maintain the voltage on the capacitor (V_C) around the input voltage to be converted (V_{in}) with amplitude variations in the order of 1 LSB. Once the voltage across the capacitor is initialized to the value V_{in}, that voltage V_c remains close to V_{in} the average value of the voltage V_{PHL} during the conversion time (T_C) must be in the case ideal, equal to the voltage to be converted being the voltage variation at the output of the LPF filter given by:

$$\Delta V_C = (V_F - V_{C0}) \cdot (1 - e^{-\frac{\Delta t}{\tau}}) \tag{11}$$

where V_{C0} represents the initial capacitor voltage, the time constant τ is the time constant of the LPF and V_F represents the charging or discharging voltage level delivered by the microcontroller (V_{PHL}). Being T_H and T_L the time duration associated with the high and low voltage pulses, with amplitudes equal to V_H and V_L, respectively, the capacitor voltage at the end of the conversion cycle, is given by:

$$V_C(t) = V_{in} + m(V_H - V_{in}) \cdot (1 - e^{-\frac{T_H}{\tau}}) - p(V_{in} - V_L) \cdot (1 - e^{-\frac{T_L}{\tau}}) \tag{12}$$

where m represents the number of high voltage pulses and p represents the number of low voltage pulses.

Since, it is assumed that at the end of the conversion period the voltage $V_C(t)$ remains equal to $V_{in} \pm Q$, where Q is much smaller than V_{in}, the following relationship is a valid approximation for $V_C(T_C)$:

$$V_{in} = \frac{mk_H}{mk_H + pk_L} V_H + \frac{pk_L}{mk_H + pk_L} V_L \tag{13}$$

Being the coefficients k_H and k_L defined by:

$$\begin{aligned} k_H &= (1 - e^{-\frac{T_H}{\tau}}) \\ k_L &= (1 - e^{-\frac{T_L}{\tau}}) \end{aligned} \tag{14}$$

If the durations of the high and low pulses are equal ($T_H = T_L$), the relationship between the voltage to be converted by the PWM NPC converter and the number pulses, is given by:

$$V_{in} = \frac{m}{m + p} \cdot V_{REF} \tag{15}$$

being considered, for simplicity, that $V_H = V_{REF}$ and $V_L = 0$.

Thus, since the total number of pulses (m + p) is equal to 2^n, we can conclude that there is a linear relationship between V_{in} and the positive pulses number (m) generated during the conversion cycle, being the quantization step of the converter equal to

FS/2^n. Note that for the voltage values considered in (15), the FS of the converter is equal to V_{REF}.

3.3 RC time constant dimensioning

In order to obtain a voltage variation on the capacitor (*"ripple")* of 1 LSB (Q=FS / 2^n), the following relationship must be verified:

$$\Delta V_C = (V_F - V_{C0}) \cdot \left(1 - e^{-\frac{T_P}{T}} \right) \cong \frac{Q}{2} \tag{16}$$

where V_F represents the high voltage pulse amplitude and V_{CO} represents the initial voltage of the capacitor.
Considering an input voltage in the middle of the conversion range (V_{CO}=FS/2) and 2^n>>1, and performing a first order development of the exponential function (T_P<<τ), it is possible to obtain:

$$\frac{T_P}{\tau} = \frac{1}{2^n} \tag{17}$$

Considering that the microcontroller pulse has a time resolution equal to T_{CLK}, the conversion time is given by:

$$T_C = 2^n \cdot T_{CLK} \quad (2^n >> 1) \tag{18}$$

Comparing expressions (5) and (18) with it is possible to conclude that the conversion time of the PMW NPC A/D converter is, at least, 2^n times lower than the conversion time of the traditional A/D PWM converter.
Figure 6 depicts the conversion time of the PWM NPC A/D converter as a function of the number of bits of the converter for two different values of the temporal resolution of the microcontroller PWM signal (T_{CLK}).

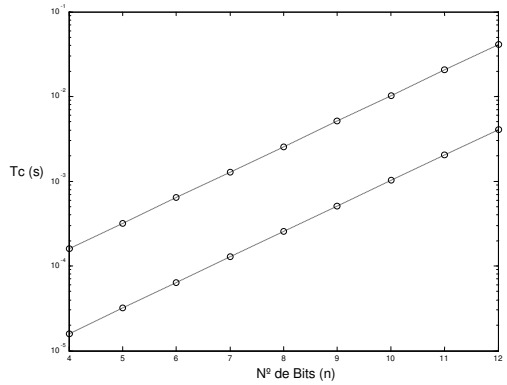

Fig. 6. Conversion time (T_C) of the PWM NPC A/D converter for two different values of the clock signal period (T_{CLK}=1 µs and T_{CLK}=10 µs) when the A/D converter number of bits varies between 4 and 12.

Comparing the results represented in the previous figure with the results previously obtained (figure 3), the conversion time that is obtained with this conversion technique is much lower than the one obtained with the traditional PWM conversion technique, which means that it possible to increase substantially the conversion rate of this converter and the bandwidth of the input signals that can be digitized.

3.4 Conversion range

The conversion range of the PWM NPC A/D can be dynamically and easily settled by changing the voltage values associated with the high and low voltage pulses. If V'_H and V'_L are the new voltage values associated with the high and low voltage pulses, respectively, the resolution of the converter is given by:

$$Q' = \frac{V'_H - V'_L}{2^n} \tag{19}$$

which corresponds to an A/D converter resolution gain, given by:

$$n_G = log_2\left(\frac{V_H - V_L}{V'_H - V'_L}\right) \tag{20}$$

As an example, figure 7 represents a circuit that that subdivides the FS range of conversion into 4 segments, which correspond to a theoretical 2-bit resolution gain. If the 2 lines conversion codes, C_L and C_H, are, for example, equal to '01', the conversion range will vary between FS/4 and FS/2 and the number of bits of the converter will be equal to n+2, instead of n. Obviously, the number of bits gain can be increased using additional resistors in the voltage divider that could be an integrated R/2R resistor ladder network. However, calibration requirements are more critical since there are new error sources associated with the accuracy of the resistor ladder network and with internal resistances of the analog multiplexers (S).

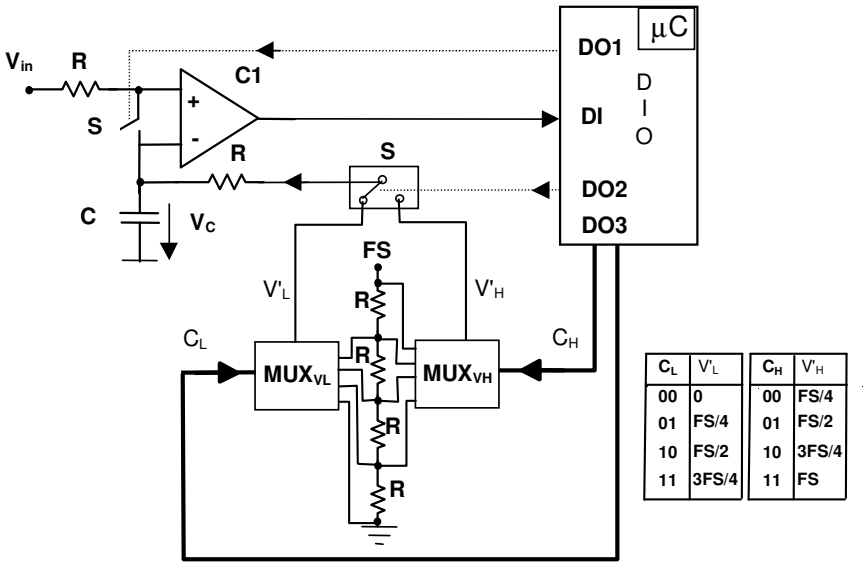

Fig. 7. Circuit that subdivides the FS range of the PWN NPC A/D converter into 4 segments (S- analog switch, DI- digital input, DO- digital output, MUX- analog multiplexer, FS- full-scale range).

4. Simulation results

In order to validate the theoretical analysis and conclusions in the previous sections, several programs were developed in MATLAB to evaluate the magnitude of the quantization error as a function of the ratio between the pulse duration (T_P) and the time constant (RC) and also the ratio between the converter's number of bits (n) and the oversampling factor (N). This ratio was defined as the ratio between the sampling frequency and the frequency of the signal input (V_{in}) that is assumed to be sinusoidal. Simulation routines contain an option parameter to considerer that during the conversion cycle the input voltage is constant (S&H=ON) or variable, according to the dynamic variation of the input signal (S&H=OFF).

4.1 Voltage variation in the capacitor during the conversion cycle

Figure 8 represents the amplitude variation of the average voltage module in the capacitor as a function of the normalized input signal amplitude for three different values of the ratio T_P/RC.

The simulation parameters were defined as follows: V_H=1/2; V_L=-1/2; n=5 bits; N=128; S&H=ON).

The average value of the amplitude of the voltage module of the variations in the capacitor is normalized relatively to the quantization step (Q) and the amplitude of the input voltage (V_{in}) is normalized relatively to the conversion range ([-1/2, 1/2]).

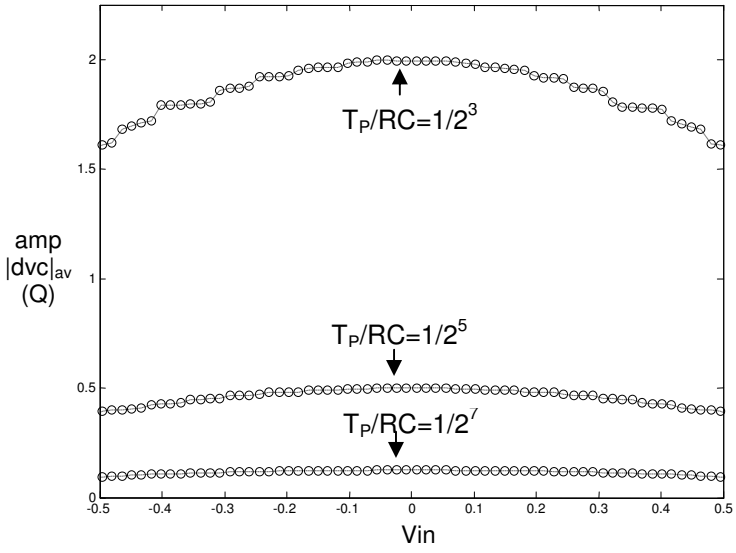

Fig. 8. Average value of the amplitude of the voltage variations in the capacitor as a function of the normalized input voltage for three different values of the ratio T_P/RC.

As expected, there is a maximum amplitude of the average module voltage in the capacitor for the central value of the conversion range and the amplitude of the voltage also increases with T_P/RC ratio. The simulation results also confirm that in order to obtain an average voltage variation of the voltage across the capacitor equal to the quantization step (Q) the ratio between the pulse duration and the time constant (T_P/RC) must be equal to $1/2^n$ where n represents the number of bits of the PWM NPC A/D converter.

4.2 Quantization errors for a sinusoidal signal

Figure 9 represents the quantization error, over a period of a sinusoidal input signal with unitary normalized amplitude and the following simulation conditions: N=1024, n=5, $T_P/RC=1/2^5$ and S & H=ON (with input signal retention).
The measured errors, in LSB units, are evaluated as the difference between the voltage associated with each output code converter and the input voltage to be converted (V_{in}), for each one of the 1024 (N) samples contained in the sinusoidal signal cycle. Using a ratio of T_P/RC given by (17), the amplitude variation of the voltage across the

capacitor, during each sample conversion cycle, is approximately equal to 1 LSB, as expected.

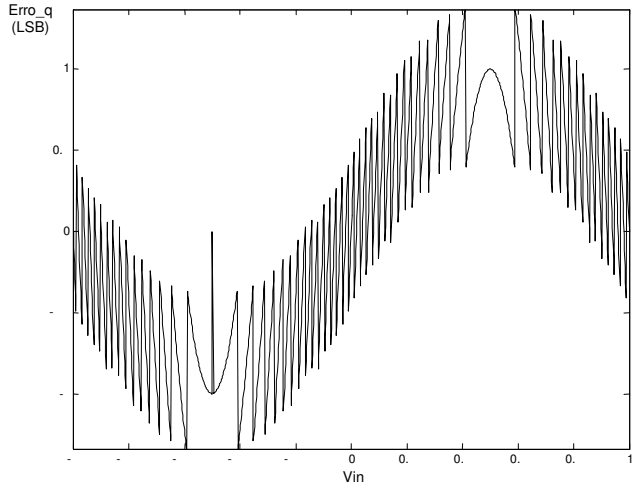

Fig. 9. Simulation results of the quantization errors during a period of a sinusoidal signal with a unitary normalized amplitude (simulation parameters: N = 1024; n = 5; $T_P/RC = 1/2^5$; S&H=ON).

The maximum amplitude of the quantization error occur for input voltage values that approach the limits of ADC conversion range. Figure 10 represents the amplitude of the quantization errors when the amplitude of the input sinusoidal signal is reduced to 1/4 of the maximum normalized amplitude value (amp = 1/4).

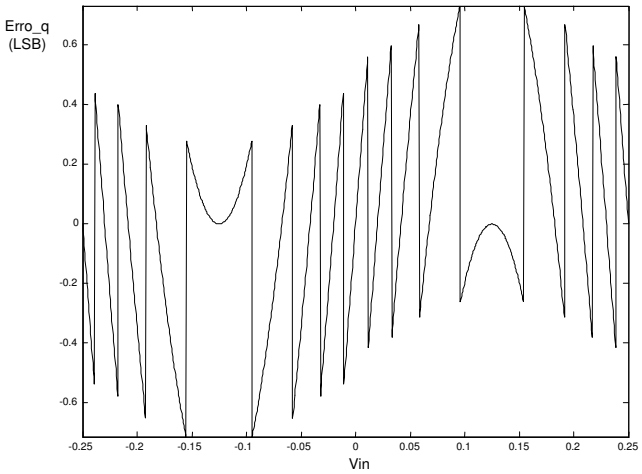

Fig. 10. Simulation results of the quantization errors during a period of a sinusoidal signal with a 1/4 normalized amplitude (simulation parameters: N = 1024; n = 5; $T_P/RC = 1/2^5$; S&H=ON).

In this case the amplitude of the quantization error is reduced significantly compared to the case discussed previously, since the input signal voltage values are away from the boundaries values of the ADC conversion range.
By its turn, Figure 11 represents the conversion errors obtained with the following simulation conditions: N=64; n=8; $T_P/RC=1/2^8$ and S&H=OFF (without input signal retention). In this example, being de S&H parameter OFF, the conversion error function is much higher and its profile is modulated by a cosine function variation. This result was also expected since the ratio between the number of codes of the A/D converter (2^n) and the oversampling factor (N) is equal to 1/4 being the delay of the conversion error function, relatively the sinusoidal variation of the input signal (V_{in}), equal to a quarter of period.

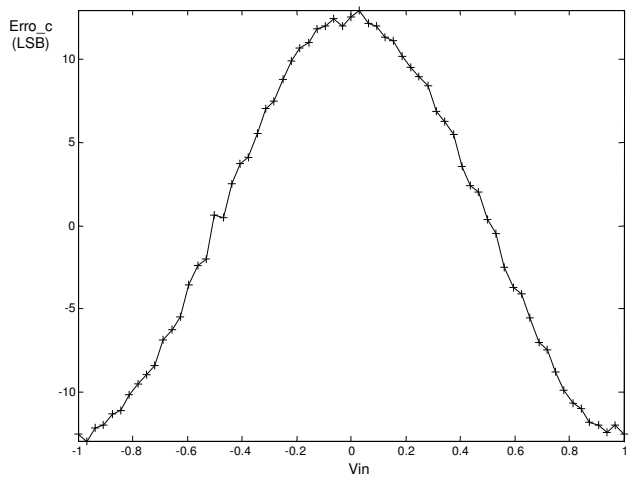

Fig. 11. Simulation results of the conversion errors during a period of a sinusoidal signal with a unitary normalized amplitude (simulation parameters: N = 64; n = 8; $T_P/RC = 1/2^8$; S&H=OFF).

In order to analyze the importance of the S&H capability in the PWM NPC A/D converter, a new simulation was performed for a different value of the oversampling factor (N) when the others simulation parameters remain equal. Figure 12 represents the conversion errors that are obtained using the following simulation conditions: N=512; n=8; $T_P/RC=1/2^8$ and S&H=OFF (without input signal retention). Comparing these results with the ones previously obtained with N=64, the main conclusions remains but the amplitude of the conversion error is substantially reduced, from almost 12 to 2 LSB, since the variation of the input signal during the conversion period decreases as N increases.

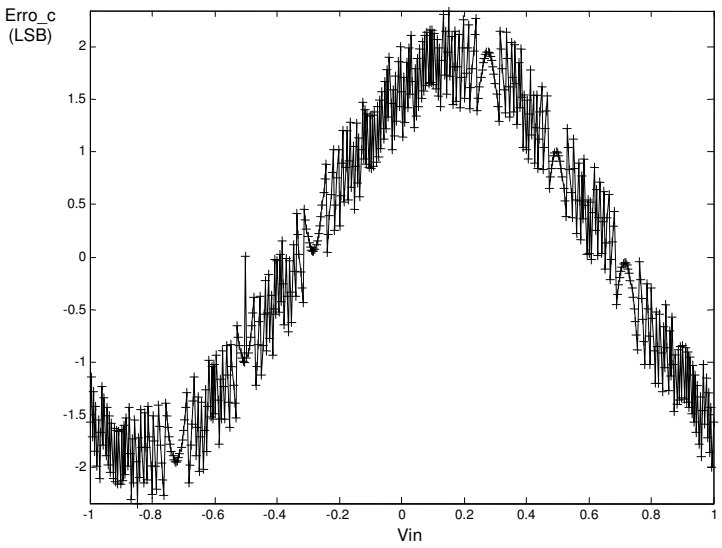

Fig. 12. Simulation results of the conversion errors during a period of a sinusoidal signal with a unitary normalized amplitude (simulation parameters: N = 512; n = 8; $T_P/RC = 1/2^8$; S&H=OFF).

However, it is important to underline that the increasing of the oversampling factor is limited by the specifications of the microcontroller that is used. Moreover, an increase of the oversampling error is associated, for a given conversion bandwidth, by other non-idealities of the hardware, namely by the increase of the distortion of $V_{PWM,}$ provided by the microcontroller, when the pulse width value (T_P) decreases.

In order to analyze the variation of the conversion errors as a function of the input signal variation, several simulations where performed for different values of input signal amplitudes. Figure 13 represents the error that where obtained for different values of normalized input signal amplitudes using the following simulation conditions: N=64; n=8; $T_P/RC=1/2^8$ and S&H=OFF (without input signal retention). As expected, the amplitude of the conversion error increases with the amplitude of the input signal and, for a given amplitude, the profile variation of the error function is proportional to the input signal slew-rate, being the error maximum when the input signal has a maximum or minimum derivative value (zero crossings for sinusoidal signals). By its turn, the quantization error is null when the input sinusoidal function has a maximum or minimum amplitude value (null value of the derivative of a sinusoidal signal).

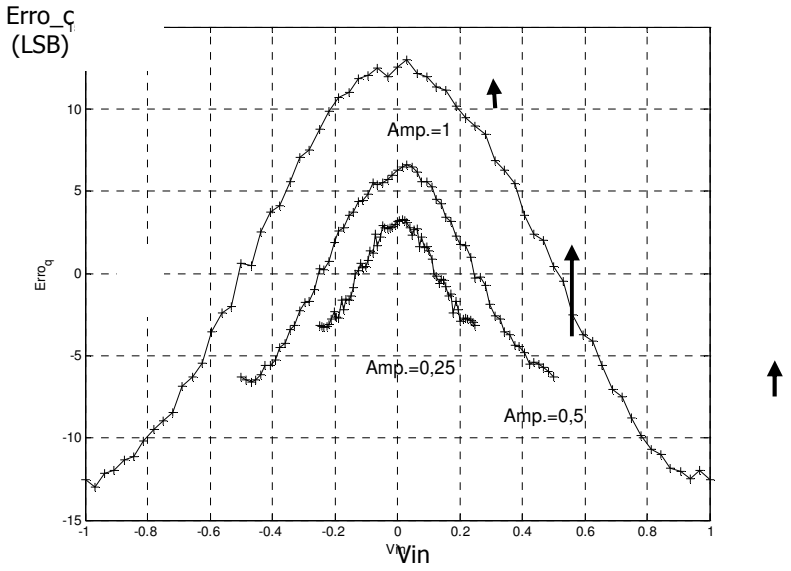

Fig. 13. Simulation results of the conversion errors during a period of three sinusoidal signal with different normalized amplitudes (simulation parameters: N = 64; n = 8; $T_P/RC = 1/2^8$; S&H=OFF).

To conclude the simulation tests the effect of noise was briefly analyzed. Figure 14 represents the conversion errors for the following simulation conditions: N=128; n=8; $T_P/RC=1/2^8$; S&H=ON (with input signal retention) and a random Gaussian input noise amplitude equal to 1 LSB. As expected, the effect of the noise increases the amplitude of the PWM NPC A/D conversion errors. Comparing the present results with those obtained without noise, it's possible to conclude that the maximum amplitude of the conversion error almost doubled compared when the maximum amplitude of this error in the absence of noise (1 LSB).

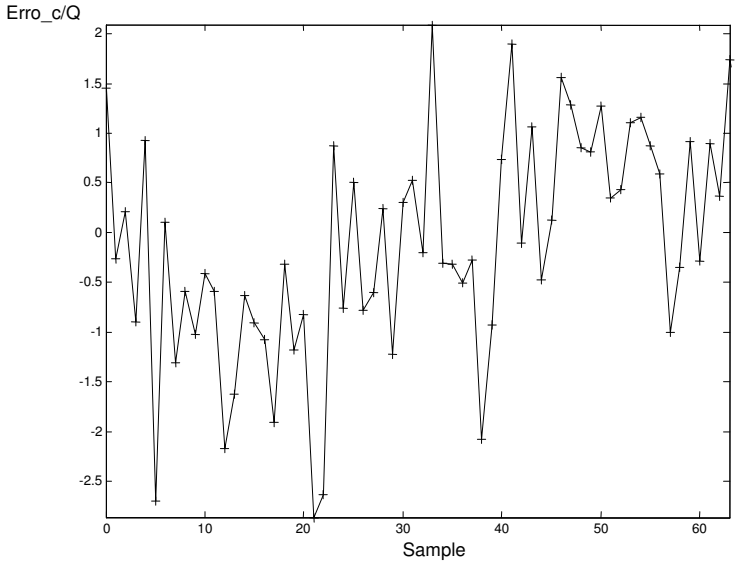

Fig. 14. Simulation results of the conversion errors during a period of a sinusoidal signal with a unitary normalized amplitude (simulation parameters: N = 64; n = 8; $T_P/RC = 1/2^8$; S&H=ON; noise_amp=1 LSB).

5. Experimental results

A basic prototype was developed to test the PWM NPC A/D converter. The main elements used to implement the experimental prototype include a microcontroller [27-28], a comparator [29] and a low-pass filter (RC). Some bi-directional I/O ports of the microcontroller were used to control analog relays to initialize A/D conversion, to define A/D conversion range and to implement auto-calibration based on three reference voltage values: ground; FS and FS/2. The experimental apparatus also include a data acquisition board with a maximum sampling rate of 200 kS/s [30] and a function generator with a spectral purity better than -70 dBc [31]. The tests were performed with a sinusoidal signal of 1 kHz and amplitude equal to 1 V.

In order to evaluate offset and gain errors of the A/D converter some results that are presented in this section did not consider a previous cancelation of the offset and gain errors.

Figure 15 represents the simulation errors expected for the following simulation conditions: N=128; n=8; $T_P/RC=1/256$ and S&H=ON (with input signal retention). The mean value of the absolute conversion error and its standard deviation error are equal to 0.2767 and 0.3355 LSB units, respectively.

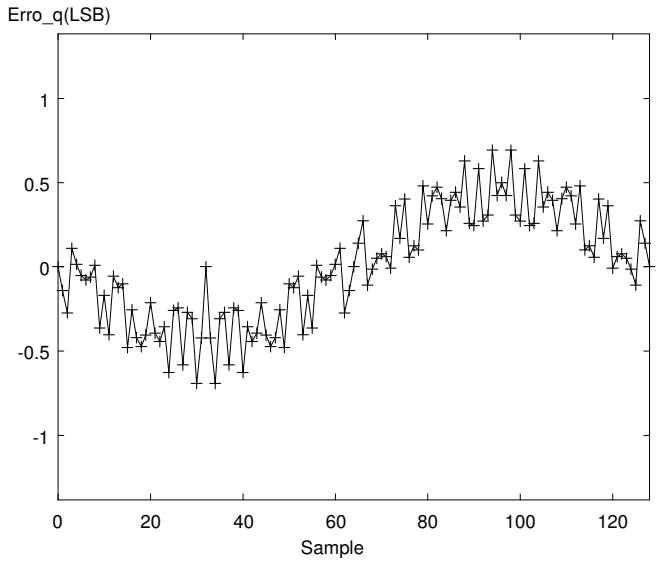

Fig. 15. Simulation results of the quantization errors during a period of a sinusoidal signal with a unitary normalized amplitude (simulation parameters: N = 128; n = 8; T_P/RC = 1/256; S&H=ON).

Conversion errors that were obtained, without compensation of offset and gain errors, are represented in figure 16.

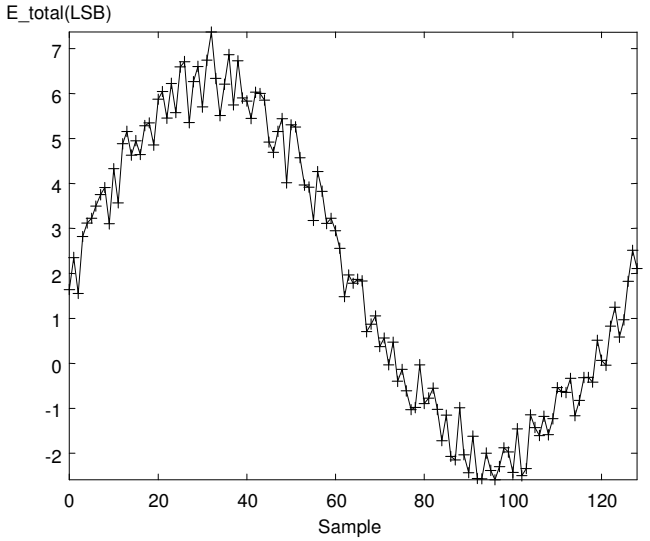

Fig. 16. Experimental results of the conversion errors without compensation of offset and gain errors (S&H=ON).

The high magnitude of the errors is caused by the offset and gain errors of the converter that were not compensated. The mean value of the absolute conversion error and its standard deviation error are now equal to 2.9814 and 3.0344 LSB units, respectively. Using the sine wave curve fitting technique [32] it was possible to evaluate an offset error equal to approximately 2 LSB and a relative gain error, referenced to FS, approximately equal to 2%.
Figure 17 represents the difference between the A/D converter error, after removing the offset and gain errors, and the quantization error obtained by simulation. The mean absolute value of this difference error and its standard deviation error are now equal to 0.6375 and 0.7457 LSB units, respectively. The amplitude of these errors is acceptable and the difference corresponds approximately to a random noise with amplitude approximately equal to 0.5 LSB.

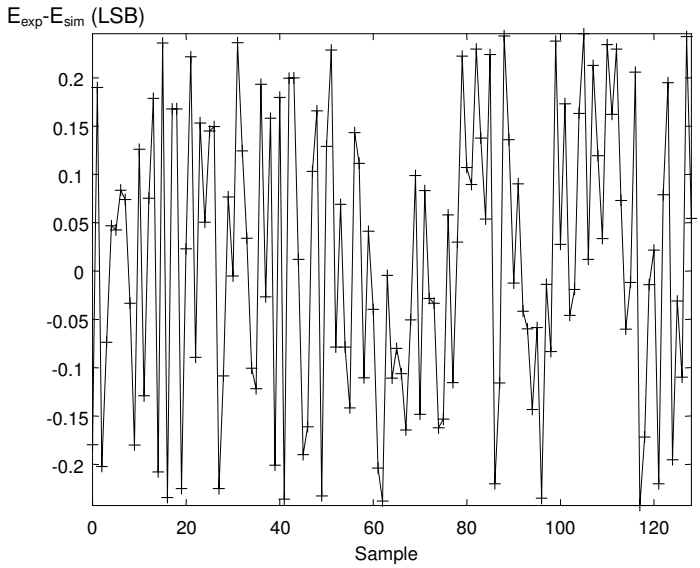

Fig. 17. Experimental results of the difference between the conversion errors, after compensation of offset and gain errors, and the quantization errors (S&H=ON).

Experimental tests were also performed without retention of input signal during the conversion cycle (S&H=OFF). As an example, figure 18 represents the difference between the A/D converter error, after removing the offset and gain errors, and the quantization error obtained by simulation without S&H usage. In this case, the mean absolute value of this difference error and its standard deviation error are now equal to 2.9974 and 3.0207 LSB units, respectively. As expected, in this case the amplitude of these errors are not explained by a random noise with an amplitude approximately equal to 1 LSB since the main cause for the errors is now related with the inexistence of the retention function (S&H=OFF).

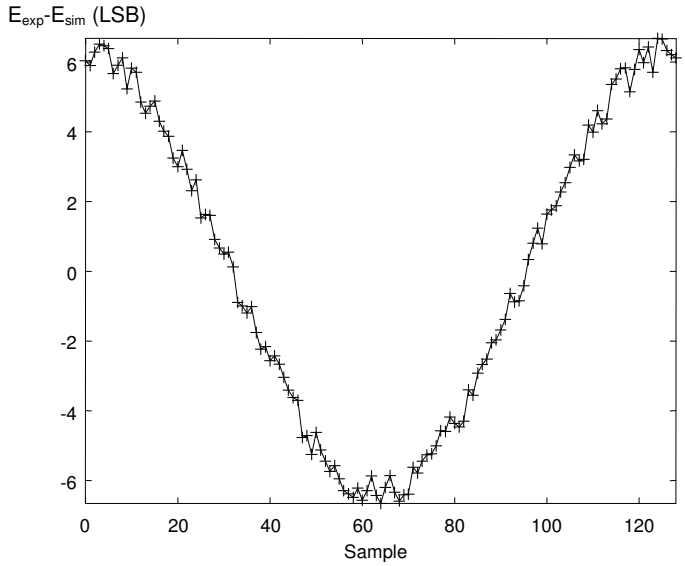

Fig. 18. Experimental results of the difference between the conversion errors, after compensation of offset and gain errors, and the quantization errors (S&H=OFF).

6. Conclusions

This chapter underlined the main advantages associated with an improved variant of a discrete A/D method based on PWM. The proposed A/D conversion method can be easily implemented in low-cost microcontrollers that include CCP capabilities even if they haven't internal ADC. The main advantages of the presented A/D method include a flexible and adaptive adjustment of A/D conversion range, accuracy and non-linear conversion transfer characteristic profile that can be used to perform the linearization of, previously calibrated, analog measurement channels. All the adjustments that were referred can be performed in real time by software to optimize the performance of any smart sensing system. The chapter also includes a detailed explanation of all the elements of the discrete A/D converter and of the operation principle of the PWM A/D converter based on pulse counting. Several theoretical relationships were deduced and validated by simulation and experimental results.

References

[1]. Miguel Dias Pereira, Octavian Postolache, Pedro Silva Girão, "Using neural Network techniques in Environmental Sensing and Measurement Systems to Compensate for the Effects of Influence Quantities", *Instrumentation & Measurement Magazine*, Vol. 17, No. 6, December 2014, pp. 26-33.

[2]. Mauro Santos, Nuno Horta, Jorge Guilherme, "A survey on nonlinear analog-to-digital converters", *VLSI Journal in Integration*, 47(1), January 2014, pp. 12-22.

[3]. Dariusz Kościelnik, Marek Miśkowicz, "Time-to-digital converters based on event-driven successive charge redistribution: A theoretical approach", In Special Volume, *Measurement*, 45(10), December 2012, pp. 2511-2528.

[4]. H. Danneels, F.Piette, V. De Smedt, W. Dehaene, G. Gielen, "A novel PLL-based frequency-to-digital conversion mechanism for sensor interfaces", *Sensors & Actuators: A. Physical*, Linz, Austria, 172(1), December 2011, pp.220-227.

[5]. M. Kuorilehto et al., "Ultra-Low Energy Wireless Sensor Networks in Practice: theory, realization and deployment", *John Wiley and Sons, Ltd*., Finland, 2007.

[6]. M. Alves, J.M. Dias Pereira, J.M. Fonseca, "A Piezoelectric Device for Measurement and Power Harvesting Applications", *DoCEIS - 6th Doctoral Conference on Computing*, Electrical and Industrial Systems, Portugal, April 2015.

[7]. J.M. Dias Pereira, O. Postolache, P. Silva Girão, "A Digitally Programmable A/D Converter for Smart Sensors Applications", *IEEE Transactions on Instrumentation and Measurement*, Vol.56, No.1, February 2007, pp. 158-163.

[8]. José Dias Pereira, Carlos Banha, Octavian Postolache, Pedro Silva Girão, "Improving Accelerometers Performance Using Smart Sensing Techniques", *IEEE Sensors 2006*, Daegu, Coreia do Sul, October 2006, pp. 654-657.

[9]. J.M. Dias Pereira, O. Postolache, P. Silva Girão, "A Self-Adaptable Method to Optimize the Performance of Frequency-to-Code Conversion Based Measurement Systems", *International Workshop on Intelligent Data Acquisition and Advanced Computing Systems: Technology and Applications (IDAACS'2005)*, Sófia, Bulgária, September 2005.

[10]. Randy Frank, "Understanding Smart Sensors", *Artech House*, 1996.

[11]. Sleiman Bou-Sleiman, Mohammed Ismail, "Built-in-Self-Test and Digital Self-Calibration for RF SoCs", *Springer (SpringerBriefs in Electrical and Computer Engineering)*, 2012.

[12]. Gerard Meijer, Kofi Makinwa, Michiel Pertijs, "Smart Sensor Systems: Emerging Technologies and Applications", *Wiley*, 2014.

[13]. Mike Cable, "Calibration: A Technician's Guide", *ISA*, 2005.

[14]. Frank Van der Goes, Gerard Meijer, "A Simple Accurate Bridge-Transducer Interface with Continuous Autocalibration", *IEEE Transactions on Instrumentation and Measurement*, Vol. 46, No.3, June 1997, pp. 704-710.

[15].Frank Van der Goes and G.C. Meijer, "A Universal Transducer Interface for Capacitive and Resistive Sensor Elements", *Analog Integrated Circuits and Signal Processing*, Vol. 14, 1997, pp. 249-260.

[16]. E.E. Hemayed, *Proceedings of the IEEE Conference on Advanced Video and Signal Based Surveillance*, Miami, USA, 2003.

[17]. Oscar S. van de Ven, Johan G. Vogel, Sha Xia; Jo W. Spronck, Stoyan Nihtianov, "Self-aligning and self-calibrating capacitive sensor system for displacement measurement in inaccessible industrial environments", *IEEE Transactions on Instrumentation and Measurement*, Volume 67, Issue 2, Feb 2018, pp. 350-358.

[18]. J.G. Vogel, Stoyan Nihtianov , "Auto-alignment of a High-Precision Eddy-current Displacement Sensor Using a Thermal Slider Actuator", *In Proceedings of the 18th international conference of the EUSPEN*, 2018, pp. 171-2.
[19]. Rudy van de Plassche, "Integrated A/D and D/A Converters", *Kluwer Academic Publishers*, 1994.
[20]. J.M Dias Pereira, Vítor Viegas, Octavian Postolache, P.S. Girão, "Software based Sigma-Delta Converter with Auto-Calibration Capabilities", *1st IFSA Frquency & Time Conference (IFTC' 2019)*, Barcelona, Spain, 2019 (submitted).
[21]. James C. Candy, Gabor C. Temes, "Oversampling Delta-Sigma Data Converters – Theory, Design and Simulation", *The Institute of Electrical and Electronics Engineers, Inc*., New York – IEEE Press, 1992.
[22]. C.Dunn and M.Sandler, "Efficient Linearisation of Sigma-Delta Modulators using Single-Bit Dither", *Electronic Letters*, Vol.31, No.12, 1995, pp.941-942.
[23]. B. N. Karthik, L. Durga Parameswari, R. Harshini, A. Akshaya, "Survey on IOT & Arduino Based Patient Health Monitoring System", *International Journal of Scientific Research in Computer Science, Engineering and Information Technology*, Volume 3, Issue 1, 2018, pp. 1414-1417.
[24]. Antonio Carlos Bento , "IoT: NodeMCU 12e X Arduino Uno, Results of an experimental and comparative survey", *International Journal of Advance Research in Computer Science and Management Studie*, Volume 6, Issue 1, 2018, pp. 47-56.
[25]. Espressif Systems, ESP8266EX Datasheet (https://espressif.com/sites/default/files/documentation/0a-esp8266ex_datasheet_en.pdf).
[26]. Alessandro Pasotti, "Controlling Arduino from OpenERP", 2012 (https://www.itopen.it/controlling-arduino-from-openerp/).
[27]. Tim Wilmshurt, "Designing Embedded Systems with PIC Microcontrollers: Principles and Applications", *Newnes*, 2006.
[28]. Microchip Technology, "PIC16F84A Data Sheet" (https://www.google.pt/search?ei=REQuXduiKvnYgwfcgbjYDg&q=pic16f84+datasheet&oq).
[29]. Texas Instruments, "LM311 Datasheet PDF" (https://www.google.pt/search?ei=3kQuXbuKN5aCjLsPgO-MmAI&q).
[30]. National Instruments, "NI myDAQ with NI LabVIEW, Multisim, & Ultiboard" (https://www.studica.com/National-Instruments-students-ni-labview-mydaq/ni-mydaq.html).
[31]. Agilent, "33120A Function / Arbitrary Waveform Generator, 15 MHz" (https://www.keysight.com/upload/cmc_upload/).
[32]. Institute of Electrical and Electronics Engineers, Inc., "IEEE Standard for Terminology and Test Methods for Analog-to-Digital Converters, IEEE Std 1241", October 2000.

Abbreviations and symbols

A/D - Analog-to-Digital
ADC - Analog-to-Digital Conversion
C - Comparator
CCP - Compare, Capture and PWM
DC - Duty-Cycle
DSP - Digital Signal Processor
EoC - End of Conversion
FS - Full-Scale
I/O - Input-Output
LPF - Low-Pass Filter
LSB - Least Significant Bit
MEMS - Microelectromechanical Systems
n - Number of bits of the ADC
NPC - Number of Pulses Counting
PWM - Pulse Width Modulation
Q - Quantization step amplitude
S&H – Sample-and-Hold
T - Period
T_C - Conversion Cycle time
(μC) - Microcontroller