A Scalar Interpolator/Compressor for the Improvement of Sensor Linearity

Nikos Petrellis
Computer Science and Engineering Dept.
TEI of Thessaly
41110 Larissa, Greece
npetrellis@teithessaly.gr

***Abstract*— A hardware linear interpolator with logarithmic and exponential curve slope correction is described in this paper allowing real time linearity improvement of low resolution sensors like the miniaturized, wearable or implantable ones used for health monitoring. A compression is performed by the interpolator when the input signal is sparse or changes with low frequency. A signal distorted by the digitization process can be approximated even when its available samples are in non-uniform distances. Multiple interpolators with identical or increasing resolution can be connected in series. A three-stage interpolator with 9-bit input and 12-bit output typical resolution is implemented using 14% of the Logic Elements of an Altera Cyclone III EP3C25N Field Programmable Gate Array and is tested using an ADC developed by the author. An improvement close to 100% has been measured at the Signal to Noise and Distortion Ratio (SNDR) and at the Spurious Free Dynamic Range (SFDR) of the ADC when a 2MHz sinusoidal input was used.**

Keywords*— *Interpolation, Seonsors, Analogue/Digital Conversion, Compression, Resolution

I. Introduction

A large number of interpolation methods have been presented in the literature for signal reconstruction including the simpler to implement linear (or first-order hold) interpolation [1] in the time domain or more complicated interpolation schemes in the frequency domain, Lagrange, min-max interpolation, etc. The approaches that are based on the post-processing of an Analogue/Digital Converter (ADC) output include the correction of Differential or Integral Non-Linearity (DNL/INL) static errors. These methods are based on the excitation of an ADC with DC levels or voltage ramps (histograms [2] or best fit curves [3]). The histogram approach includes initially the excitation of an *m*-bit ADC with a ramp signal in order to get the initial 2^m outputs and then the ramp signal is shifted up to retrieve another set of 2^m outputs [4]. The differences between the corresponding values are used for the estimation of the INL/DNL error [5][6]. The DNL linearity errors can be corrected by signed error correcting factors that are stored in large look up tables. The calibration of the DNL error coefficients in real time is also described in [7].

In [8], appropriate Lagrange and min-max interpolation methods are proposed for high precision signal reconstruction, implementable with low complexity hardware. In [9] the interpolation is performed by a weighted summation of a number of preceding and succeeding samples that are available in order to estimate the current signal value. The authors of [9] compare the performance of their own algorithms (Least Mean Lattice - LSL interpolation, QR Decomposition LSL algorithm, QRD-LSL). The dynamic linearity errors that are measured by SNDR, SFDR and Total Harmonic Distortion (THD) are taken into consideration in approaches like [1] and [7].

In the first-order interpolation, two successive samples are assumed to be linearly connected. If the original analogue signal is curved between the successive samples, the first-order hold is not accurate but the error is smaller if the distance between these samples is short. However, higher order interpolation schemes like Cubic spline interpolation, may also fail to find a curve that accurately matches the input signal.

The proposed interpolation method is based on a first-order interpolation in the time domain for lower complexity and is evaluated on an FPGA. Its simplicity stems from the fact that it requires only a small number of adders, counters and comparators. The error of the interpolated values in exponential or logarithmic signals (e.g., a capacitive biosensor output) is further reduced by employing additional error correcting rules.

Low resolution signals like the ones generated by the wearable or implantable smart sensors with extremely small dimensions used in healthcare applications, can be significantly improved by the proposed interpolation method. Experimental results show that the SNDR of a 2MHz sinusoidal signal has been increased from 21dB to approximately 40dB (approximately 100% improvement) using a 3-stage interpolator scheme. The SFDR has been improved in this case from 27dB to 54dB. The interpolator described in this paper is tested on sinusoidal signals digitized by an ADC developed in TSCM90nm CMOS technology [10]. The interpolator can also perform real time compression in the case of sparse or signals that change with a very low frequency. For example, the output of a capacitive pressure sensor was compressed down to the 11% of its original size. A signal is represented at the interpolator output as a sequence of (*V*, *C*) pairs implying that the value *V* of the signal appears *C* times. This type of compression has been adopted by standards like LZ77/78. The proposed interpolator is scalar in the sense that higher linearity can be achieved if multiple interpolators are serially connected.

The architecture of the developed interpolator is described in Section II. Experimental results are presented in Section III.

This work is patent pending (provisional patent with application number 20130100627 - Greek Patent Office)

MOBIHEALTH 2014, November 03-05, Athens, Greece

DOI 10.4108/icst.mobihealth.2014.257197

A potential use of the interpolator modules in a health monitoring application is also described in Section III.

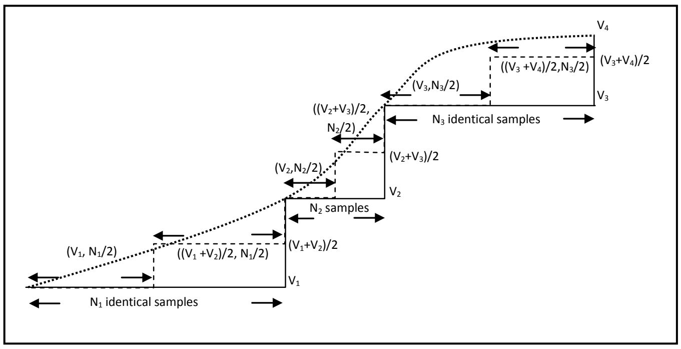


Fig. 1. The concept of the proposed interpolation method

II. Interpolator Architecture

The concept of the proposed first order interpolation method is described in Fig. 1. The dotted line is the original signal while the solid line corresponds to the zero-order hold of the ADC output values (V_1, V_2, V_3, etc) and may not change at regular intervals. The implemented interpolator samples the ADC output with an interpolator clock CLK with higher frequency compared to the sampling rate of the ADC. If for example, V_1 appears N_1, CLK periods and V_2 appears N_2, CLK periods then the last $N_1/2$ appearances of V_1 are replaced by $(V_1+V_2)/2$ and the last $N_2/2$ appearances of V_2 are replaced by $(V_2+V_3)/2$. The dashed line in Fig. 1 shows the resulting interpolated signal. Additional intermediate points can be estimated if this procedure is applied recursively. The new values generated by the interpolator like $(V_1+V_2)/2$, $(V_2+V_3)/2$, belong to the piece of line of a first-order hold.

If the difference between two successive ADC output values is 1, the estimated average value would not be an integer. For this reason, the interpolator outputs the original ADC values and the averages multiplied by 2, in pairs with the corresponding interpolator clock periods, leading to an even simpler implementation, For the example of Fig. 1, the output of the interpolator would be the pairs: $(V_1, N_1/2)$, $((V_1+V_2)/2, N_1/2)$, $(V_2, N_2/2)$, $((V_2+V_3)/2, N_2/2)$, etc.

The error between the input of the interpolator and the original analog signal is higher than the error between the interpolator output values and the analog input provided that the original analog signal is monotonically changing between each pair of successive ADC sample values otherwise, the proposed interpolation scheme does not guarantee a lower approximation error. The last case may occur when too long ADC sampling intervals are used, compared to the period of the input signal.

The architecture of the proposed interpolator appears in Fig. 2. The input digital values have an m-bit resolution and they may have been acquired either by the ADC or by a previous interpolator stage. These values drive a pair of latches (L1 and L2). The latch L1 uses the interpolator clock CLK and when the input changes, a short inequality pulse is generated at the output (NEQ) of the digital m-bit comparator CMP that compares the input and the output of L1. The pulse on the signal NEQ appears delayed by two CLK periods at the signal NED. Moreover, a simple circuitry combines the CLK and the NEQ signal in order to generate a pair of pulses on the signal STB when NEQ is activated and a single pulse on the signal SEL.

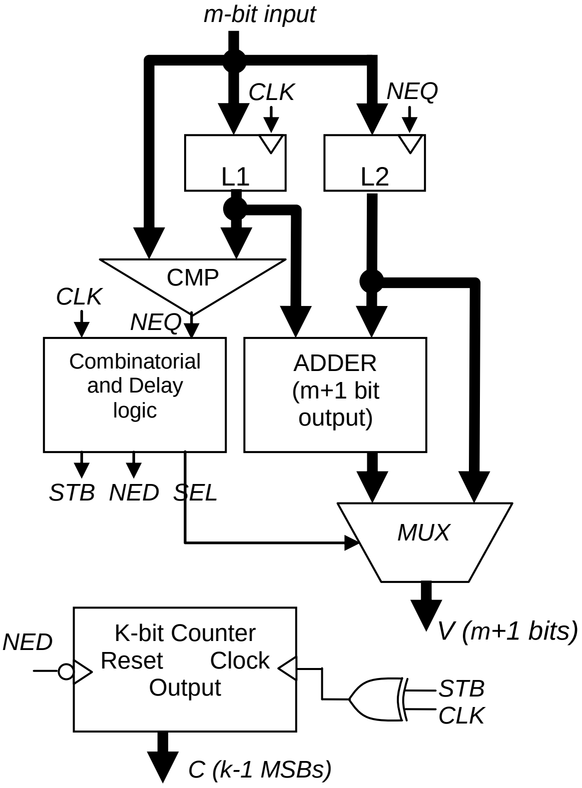


Fig. 2. The architecture of the interpolator (without correction rules).

The output of the L2 latch changes when NEQ is activated thus, the output of L2 stores the previous value of the interpolator input. This previous value is added to the current one (L1 output). A multiplexor (MUX) is controlled by the signal SEL and provides the interpolator output V selecting either the adder or the L2 output shifted left by one bit (i.e., multiplied by 2). In parallel, a K-bit is reset when NED is active and is disabled for two CLK periods when the signal STB is active. The K-1 Most Significant Bits (MSBs) of the Counter form the interpolator C output. An external system can sample the (V, C) pairs when the STB signal is active. The (V, C) pairs represent the signal in compressed form.

The output of the interpolator can be retrieved in uncompressed form by a simple Decompression unit that consists of a pair of FIFO queues and a Down Counter. When a pair of (V, C) values is read from the beginning of the FIFO, the value V is latched at the Decompression unit output until the Down Counter underflows. When an underflow occurs, the Down Counter is loaded with the next C value from the FIFO and the corresponding V value is latched at the output. The interpolator writes two new (V, C) pairs at the Decompression unit FIFOs during the pulses on the STB signal.

Higher linearity improvement can be achieved in most of the cases when multiple interpolators are connected in series. Since each interpolator has a resolution of m-bits at the input and *m+1* bits at the output, the resolution of an interpolator should be one bit higher than that of the preceding stage.

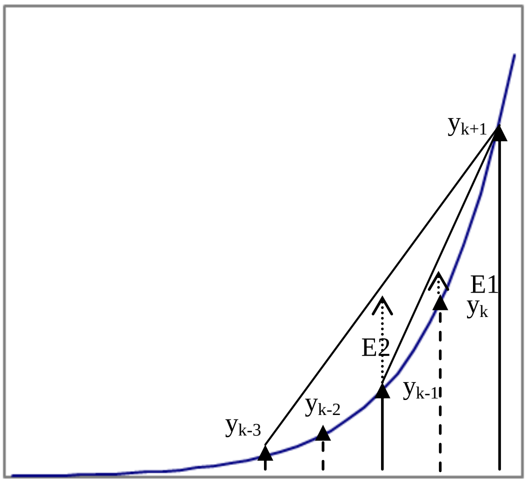


Fig. 3. Correcting exponential curves

However, the resolution of the interpolators may be identical and if it is selected to be high enough (e.g., equal to the resolution *m+i* of the last interpolator stage) then the *m+i-1* LSBs of each Decompression unit output can be used as input to the next stage without any performance loss. Even, when *m*-bit Interpolators are used, the *m* MSBs of the output can be used as input to the next stage and the linearity degradation caused by the omission of the Least Significant Bit (LSB) in each stage is less than 1dB. Thus, it may be worth using this option for lower cost.

A number of simple rules can be employed by the interpolator architecture described above for the correction of the interpolated values in the case of exponential and logarithmic curves. The correction rules are based on the fact that the interpolator generates a new value between two successive known samples. The value of three successive known samples at regular intervals can be used to recognize the curvature of the input signal. The exponential or logarithmic functions are used to derive a set of common rules for the correction of the interpolated values.

The general form of an exponential signal is $y_k=\gamma+\beta e^{\alpha k}$. If γ is 0, $\alpha\geq 0$ and $\beta>0$ the corresponding curve is shown in Fig. 3. The interpolator estimates the current value y_k as the sum of y_{k-1} and y_{k+1}. The error *E1* between the real value of y_k and the estimated one: $\hat{y}_k$ is related to the error *E2* between the real value of the sample y_{k-1} and the one that would have been estimated ($\hat{y}_{k-1}$) if the samples y_{k-3} and y_{k+1} had been used as:

$$\frac{E2}{E1}=\frac{y_{k-3}+y_{k+1}-2y_{k-1}}{y_{k-1}+y_{k+1}-2y_k}=(1+e^{-a})^2 \qquad (1)$$

If $\alpha>0$ and $\beta>0$, then y_k is monotonically increasing (confirmed by comparing y_{k-3}, y_{k-1} and y_{k+1}) and $E1\leq E2\leq 2E1$. The same holds if $\beta<0$, but y_k is monotonically decreasing. The conservative error correction rule adopted in either cases checks if *E2* is lower than *E1* or higher than *2E1* and then, *E1* is set to *E2* or *E2/2* respectively. If $\alpha<0$, then $E2>2E1$ and if it does not hold, then the *E1* is set to *E2/2*.

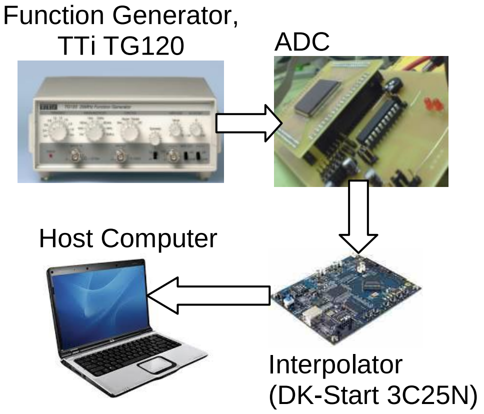
The general form of a logarithmic signal is $y_k=\gamma+\alpha\log(k)$. For this case, in a similar manner with equation (1):

$$\frac{E2}{E1}=\frac{\log((k-3)(k+1)/(k-1)^2)}{\log((k^2-1)/k^2)} \qquad (2)$$

The ratio *E2/E1* in (2) is not defined for $k<4$ and approaches 4 (using the L'Hopital theorem) when $k\rightarrow\infty$. Without loss of generality the natural logarithm has been used. Thus the relationship between *E2* and *E1* can always be expressed as $E2\geq 4E1$. If *E2* is found lower than *4E1* then *E1* is set to *E2/4*. These rules have been incorporated in the VHDL interpolator description that was tested.

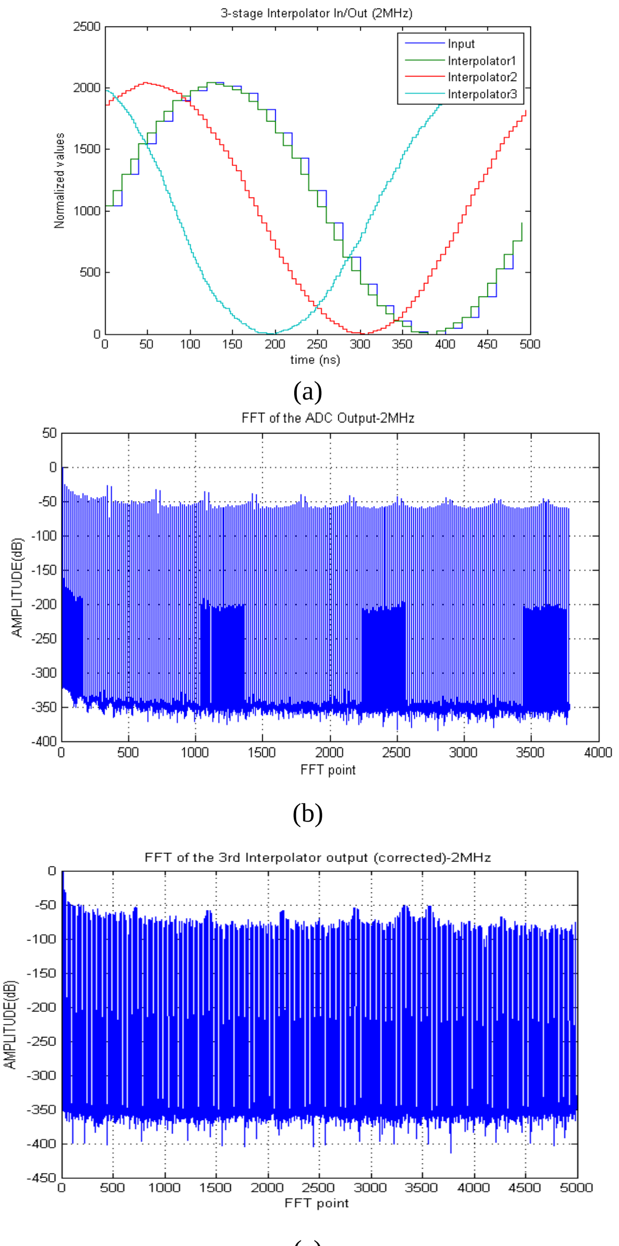
III. Experimental Results / Application

The experimental setup used is shown in Fig. 4. A three-stage interpolator with 9-bit input and 12-bit output resolution along with the necessary decompression units has been described in VHDL and tested using a low cost Altera DK Start 3C25N development kit with a Cyclon III 3C25 FPGA. An interpolator with a 9-bit input resolution required 583 Logic Elements (LEs), the corresponding decompression unit at its output 332 LEs and the total number of LEs that was required for the implementation of the three-stage interpolator was 3,325 (14%). A sinusoidal input signal is generated by a TTi TG120 Function Generator digitized by an ADC developed by the authors [10]. A 2MHz sinusoidal signal was used as input and the interpolator increased its SNDR and SFDR from 21.97dB and 27dB to 40.18dB and 52dB respectively as shown in Fig. 5. The developed three-stage interpolator can increase the SNDR and the SFDR of an ADC output by up to 100% if the initial SNDR is relatively low compared to the static resolution of the ADC. This improvement is lower (approximately 20%) if the initial SNDR of the ADC output is relatively high.



(1)

Fig. 4. Experimental Setup



(a)

(b)

(c)

Fig. 5. A 2MHz sinusoidal signal at the output of the ADC and the interpolator stages (a), the FFT of the ADC output (b) and the FFT of the 3rd interpolator stage (c).

In [1], the SNDR is increased from 51dB to 69dB by an interpolator used at the input of a DAC. Simulations show that the resolution of a 10-bit ADC is increased to 14-bits in [4] by improving the INL with 256 correction codes extracted from the 7 MSBs. The calibration scheme improves the INL of a 10-bit ADC by 2 LSBs in [6] and is tested using an FPGA. Simulations in MATLAB show that the Bayesian calibration of the correction factor look up table in [7] results in an improvement by more than 100% in the SFDR and SNDR. The proposed interpolator scheme here offers many advantages like high SNDR/SFDR improvement and compression capability, with a very low cost and complexity architecture that can be employed in real time environments.

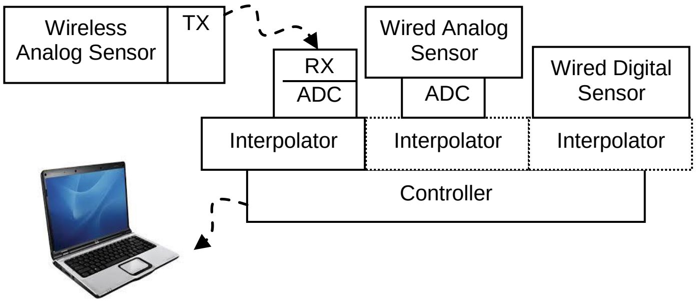


Fig. 6. Example use of the proposed interpolator in a health monitoring application.

The potential use of the proposed interpolator device in a health monitoring application is shown in Fig. 6. A controller is used to gather data from low resolution sensors with no advanced communication interface and this controller transfers the sensor information to a Host computer for data processing and decision making, through a standard wireless interface like WiFi, Bluetooth, etc. The analog wireless (e.g., implantable) and analog wired sensors should have their indications digitized by separate (if no multiplexing is used) ADCs. The sensor data resolution can be enhanced by an Interpolator module that should be connected at the output of an ADC before the data acquisition by the controller. Digital sensors either wired or wireless can feed directly an Interpolator in order to enhance the linearity and resolution of their measurement.

REFERENCES

[1] H. W. Wang, C. F. Chan and C. S. Choy, “High speed CMOS digital-to-analog converter with linear interpolator,” IEEE Transactions on Consumer Electronics, vol. 46(4), pp. 1137-1142, 2000.

[2] F. Stefani, D. Macii, A. Moschitta, P. Carbone and D. Petri, “Simple and Time-Effective Procedure for ADC INL Estimation,” IEEE Trans. on Instrumentation and Measurement, vol. 55(4), pp.1383-1389, Aug. 2006.

[3] I. Kollár, and J. Blair, “Improved Determination of the Best Fitting Sine Wave in ADC Testing,” IEEE Transactions on Instrumentation and Measurement, vol. 54(5), pp. 1978–1983, Oct. 2005.

[4] L. Jin, D. Chen and R. Geiger, “A digital self-calibration algorithm for ADCs based on histogram test using low-linearity input signals,” Proceedings of the IEEE ISCAS’05, pp. 1378 – 1381.

[5] H. C. Hong, F. Y. Su & S. F. Hung, “A Fully Integrated Built-In Self-Test Σ−Δ ADC Based on the Modified Controlled Sine-Wave Fitting Procedure,” IEEE Transactions on Instrumentation and Measurement, vol. 59(9), pp. 2334–2344, Sep. 2010.

[6] B. Provost and E. Sánchez-Sinencio, “A Practical Self-Calibration Scheme Implementation for Pipeline ADC,” IEEE Transactions on Instrumentation and Measurement, Vol. 53(2), pp. 448–456, Apr. 2004.

[7] L. De Vito, H. Lundin and S. Rapuano, “Bayesian Calibration of a Lookup Table for ADC Error Correction,” IEEE Trans. on Instrumentation and Measurement. Vol. 56(3), pp. 873-878, June 2007.

[8] J. J. Fuchs and B. Delyon, “Min-max interpolators and Lagrange interpolation formula,” Proceedings of the IEEE ISCAS ’02, No. 4, pp. 429-432.

[9] Z. Sun and L. Guo, “An Improved LSL Interpolator to Suppress Narrow-Band Interference for Direct-Sequence Spread-Spectrum System

[10] s,” Proceedings of the IEEE ChinaCom’06, pp. 1-5, 2006.

[11] N. Petrellis and M. Birbas, “An Ultra Low Die Area 8-b ADC and its Generic Calibration Logic”, Springer Analog Integrated Circuits and Signal Processing, (2012) 70: pp. 323–335.