IECON 95, ORLANDO, FLORIDA, USA

Digital Filtering in Smart Load Cells

José Higino
Departamento de Electrónica Industrial
Universidade do Minho
Lg Paço, 4719 Braga codex
PORTUGAL
e-mail: jhc@dei.uminho.pt

Carlos Couto
Departamento de Electrónica Industrial
Universidade do Minho
Lg Paço, 4719 Braga codex
PORTUGAL
e-mail: ccouto@dei.uminho.pt

Abstract — **This paper describes an application of a Self-Adaptive Pseudo-Moving Average Filter used in the implementation of a Smart Load Cell, to combine a stable digital output with a fast response to weight changes.**

The Smart Load Cell is a data acquisition solution using a single chip RISC microcontroller with very few other active and passive components around and taking advantage of the ratiometric functioning of load cell. The use of Smart Load Cells with digital outputs needs a cost effective in digital filtering of the final converter results for each Smart Load Cell. The technique is established by theoretical analysis and is justified by means of simulation and experimental results.

The paper also describes an example of software calibration of a Multi-Load-Cells weighbridge, using four smart load cells.

I. INTRODUCTION

Load Cells have been used for many years in a variety of applications and are generally considered to be the force sensor for industrial weighting systems.

The use of load cells with digital outputs [1], i.e., with integrated signal processing was the objective to build one single processing module per sensor (Smart Load Cell) [6].

The amplifier gain and offset adjustment controlled by software, digital filtering and network facilities need processing capabilities, for that reason the use of the high performance low cost microcontrollers available today [4]. Fig.1 shows the conceptual idea of one Smart Load Cell [8]:

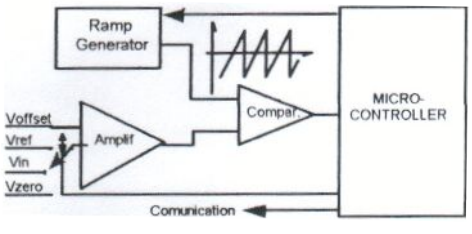

Fig.1 Smart Load Cell

Fig. 2 shows a modern solution for a Multi-Load-Cells industrial weighting system with a cable to travel over all Smart Load Cells.

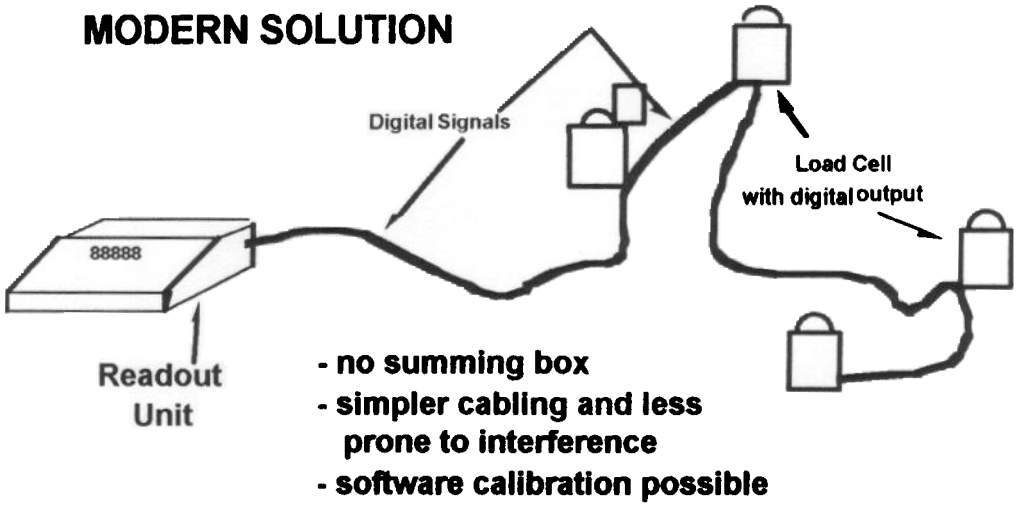

Fig.2 Multi-Load-Cells weighting system

The circuit block diagram in Fig.1 shows: the amplifier stage, single ramp conversion, comparator block and the microcontroller an 8 bit single chip Harvard architecture microcontroller with RISC-like features, the PIC17C42, with interesting characteristics for this type of applications:

- operating speed: DC - 20 MHz clock input (200ns instruction cycle).
- Low cost.
- Small size with EPROM.
- Low power consumption.
- Three 16-bit timer/counters.

Having in mind industrial weighting applications where 6000 divisions are needed [7] for the equipment (external divisions) a conversion resolution of at least 60000 divisions (10 internal divisions for each external) with 50 or more readings per second, at least for static weighting applications [3]. For dynamic weighting a faster reading rate is required but with lower resolution.

The observation of the signal from the load cell reveals a high-frequency random oscillatory component other than the oscillation due to the 2nd order transient response normal with load cells [2]. Common sources of noise are: electromagnetic pick-up, thermally unstable circuits and the gain programmable by software (through PWM's output's of the microcontroller) [5]. This signal is amplified before the analog digital conversion, therefore the noise is amplified and an error is introduced in the system.

Adaptive low pass filtering is needed to provide a stable reading of the weight – large time constant – as well as the ability to follow, as quickly as possible, changes in the weight – short time constant.

Analog filtering would be complex and also would not take care of the conversion noise, leaving the software digital filtering as the obvious solution.

II. SELF-ADAPTIVE PSEUDO-MOVING AVERAGE FILTER (SAPMAF)

The simplest method of eliminating the high frequency noise is to use a single first-order low-pass filter. An analog solution would be a standard configuration of a resistor R and a capacitor C (a time constant t=RC). The larger value of t the slower the static value will reached, but the output will contain less high frequency noise. To achieve both, a rapid response and a smooth output, an adaptive filter is once again required. This noise filter must be capable of adapting its time constant in response to the needs of the system.

The adaptive filter is used to eliminate the inherent noise within the system, to obtain a smooth output, to settle the sampling result and to have a fast response when the mass on the platform is changed. When the output is a function of input only, the number of possible responses is limited, because the output will settle within a finite time after the input has ceased to vary.

A recursive filter is simply a linear difference equation with constant coefficients and nothing more; in practice it may be realized by a short program on a general purpose digital computer or by a special purpose integrated circuit chip. A familiar example of a recursive filter is the trapezoid rule for integration [9]:

$$y_n=y_{n-1}+0.5(u_n+u_{n-1})$$

It is immediately obvious that a recursive filter can, as it were, remember all the past data, since the y_{n-1} value on the right side of the equation enters into the computation of the new value y_n, and hence into the computation of y_{n+1}, y_{n+2}, and so on. Other examples of a recursive digital filter are the exponential smoothing forecast:

$$y_{n+1}=au_{n+1}+(1-a)y_n \quad (0<a<1)$$

and the double exponential (this filter is equivalent at two exponential filters from first order in serial, where the second filter deals with the output of the first) with advantages in elimination of the high frequency noise:

$$y''_n = a^2x_n+2(1-a)y''_{n-1}-(1-a)^2y''_{n-2} \quad (0<a<1)$$

other example is the trend indicator:

$$y_n=a(x_n-x_{n-1})+(1-a)y_{n-1} \qquad (0<a<1)$$

Filters that use only past and current values of the data are called causal, for if time is the independent variable, they don't react to future events but only paste ones (causes). However by using recursion (i.e., feedback) to make the output a function of both output and input, the initial condition for the integration is remembered throughout the entire estimation of the integral.

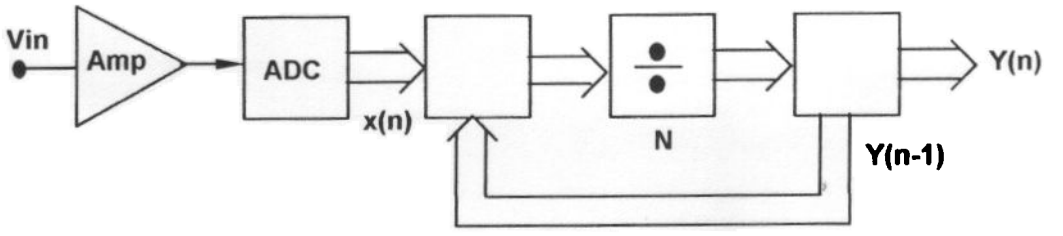

Fig.3 Recursive filter with filtering constant programmable

The next expression shows this scheme:

$$y(n)=y(n-1)+\frac{x(n)-y(n-1)}{N} \quad (1)$$

y(n) - value to sample
x(n) - value from the nth conversion
y(n-1) - last value sampled
N - filtering constant

For N=1, x(n)=y(n), increasing N the influence of the last reading in the value to sample is less. The filter response is fast for low values of N but the output takes much time to settle (small oscillations). For high values of N the filter response is slowly until the output to settle.

The **SAPMAF** has an error integral, D(k):

$$D(k) = \sum_{k=1}^{n} x(k)-y(k-1)$$

Which represents the sum of the differences between the st value sampled and the reading. The error integral gulates: the number of readings to take into consideration d the filtering constant to choose.

The error integral is compared with a constant parameter designated STACK, which represents the maximum variation of the output before to happen a transition. When the conversions are randomly spread around the result shown (reading), D(k), the integral of error fluctuates around zero. When there is a change on the weight, D(k) gross steadily and after few conversions exceeds the threshold level STACK.

The error integral is restarted to zero when its value exceeds the STACK. The filter's algorithm is:

Inputs: y(0), STACK, x(k), N.
Outputs: y(n).
Variables initiation: y(0), STACK, N .

IF $[D(k) = \sum_{k=1}^{n} x(k)-y(k-1)] >$ STACK **THEN**

[$N=N_1$, low filtering constant] with $N_1=1, 2, 4, \ldots$
[short number of readings]
[restarting D(k), with D(k+1)=0]
[to sample $y(n)=y(n-1)+\frac{x(n)-y(n-1)}{N}$]

ELSE
[$N=N_2$, high filtering constant] with $N_2=8, 16, 32, \ldots$
[large number of readings]
[to sample $y(n)=y(n-1)+\frac{x(n)-y(n-1)}{N}$]

By choosing N as 2^n, the filtering algorithm can be made simpler as faster and the division in equation (1) can be implemented by simple shifts.

III. EXPERIMENTAL RESULTS

The microcontroller driven by a clock frequency of 16MHz (250 ns instruction cycle gives a counting period of 250 ns). The conversion time obtained was around 18 ms, time which can be reduced by increasing the microcontroller clock frequency up to 25 MHz. The resolution attained of 60000 divisions can also be increased.

The Fig.4 represents the response of the filter SAPMAF in three different cases, where N, the filtering constant is respectively 2, 8, 32 (the figure was obtained with Simulink, tool of the Matlab version 4.1 for Windows) the normalized frequency was obtained with middle sampling frequency equals a one. For N=2 the response achieved was rapid and the curve in the domain of the frequency is smooth. The response for N=32 was a curve very abrupt and the change of the inputs is very slow in the output.

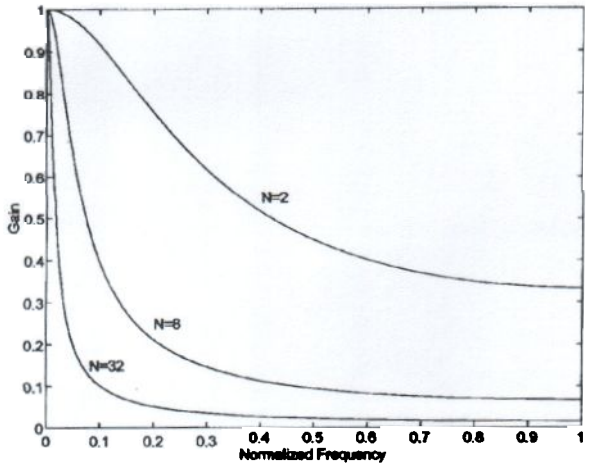

Fig.4 Behavior of the SAPMAF for three values of N

The Fig.5 shows the SAPMAF output versus the input. The output curve tries to follow the input curve (the graphic was built with the TABLE I results).

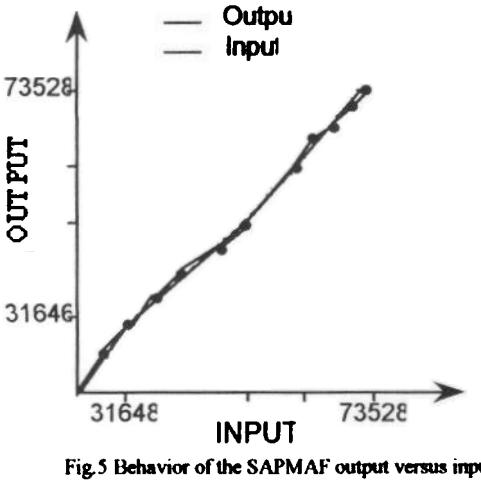

Fig.5 Behavior of the SAPMAF output versus input

The table I shows for a single system of one smart load cell the digital output when a mass is on the mini-platform. The STACK value is 15 (maximum oscillation around average) and filtering constants are for rapid transitions N=1, for smooth transitions N=4. When a sample x_n changes fast, the output's system needs only two conversions to obtain the right one.

According with the integral error, D(k), it's possible choose different filtering constants, allowing high values of N for small oscillation of D(k) (examples: D(k)=2 and N=32, D(k)=6 and N=16, D(k)=10 N=8, D(k)=12 and N=4).

TABLE I

DIGITAL OUTPUT'S FOR ONE LOAD CELL

Mass Kg	Sample x_n	Output y_n	Stack	D(k)	Deviation to average
26	31648	.....	15	0	±7
26	31644	31644	15	- 4	
26	31650	31646	15	2	
51	62616	39389	15	30968	±8
51	62622	62622	15	6	
51	62610	62619	15	- 6	
60	73531	65347	15	10915	±7
60	73525	73525	15	- 6	
60	73538	73528	15	7	
46	56080	69166	15	17451	±8
46	56088	56088	15	8	

IV. SOFTWARE CALIBRATION

The calibration process means the calculation of the multiplying coefficients, which are given by the solution of a set of equations, operation easily performed by any general purpose microcomputer.

To test the software calibration method for the multi-load cell weighbridges, it was decided to use standard readout units instead of the prototypes above referred. A local weighting equipment manufacturer made available two 4 load cells platforms and 8 digital readout units with networking facilities. Load cells taking a maximum nominal weight of 100kg, with 3000div resolution and a sensitivity around 2mV/V, were used. The 4-load cell platforms coupled to a single readout unit is rated to 200kg with a resolution of 100gr. Each of the readout units were calibrated to give around 60kg with a 20gr resolution.

Two sets of tests were done, one for a 4 load cells platform, and another for a 8 load cells system, using two 4 load cells platforms.

The calibration method consists on doing N readings of weight on each load cell obtained by moving a mass with a known weight around the platform. The number of readings is the same as the number of load cells under the platform. The best results are given by the N readings obtained, concentrating the weight as much as possible above each one of the N load cells.

For the 4 load cells platform 4 sets of 4 readings were made, and the weights found were used to workout the multiplying coefficients. These factors affecting each one of the readings, enables the correct evaluation of the weight above the platform. A system of 4 equations and 4 unknowns was built:

$$K_1W_{11} + K_2W_{12} + K_3W_{13} + K_4W_{14} = W$$
$$K_1W_{21} + K_2W_{22} + K_3W_{23} + K_4W_{24} = W$$
$$K_1W_{31} + K_2W_{32} + K_3W_{33} + K_4W_{34} = W$$

$K_1W_{41} + K_2W_{42} + K_3W_{43} + K_4W_{44} = W$

The solution of this system gives the K factors required to evaluate the weight of an unknown mass (with the W_{rc} readings calculated with a calibrated mass positioned in four different places). With the following W_{rc} readings calculated with a calibrated mass of 20kg positioned in four different places:

- W_{1c} readings 3.86, 9.96, 6.82, 0.72;
- W_{2c} readings 1.74, 2.94, 10.88, 5.58;
- W_{3c} readings 4.50, 0.74, 3.54, 13.26;
- W_{4c} readings 13.30, 2.92, 1.48, 4.38;

the K_c factors evaluated:

$K_1 = 0.90025$, $K_2 = 0.91580$, $K_3 = 0.99196$, $K_4 = 0.88685$.

After calibration the resolution of the next equation determines the weight of the mass on the platform (with Wi's the output's of each one Smart load Cell).

$$\mathbf{Mass = K_1*W_1 + K_2*W_2 + K3*W_3 + K4*W_4}$$

Using the K's factors several (25) weighting operations were done, with different masses, located in different points of the platform, having been recorded very encouraging results, with errors below 50gr (4000 divisions in 200kg).

For the composite platform with 8 load cells the test was repeated and the 8 multiplying coefficients were calculated. The weighting tests done confirmed the approach followed giving errors below 100gr, i.e. again 4000 divisions in 400kg.

V. CONCLUSIONS AND FUTURE WORK

The use of digital signal processing techniques can greatly improve the frequency response of transducers. The **SAPMAF** had shown to be able to give the required performance (the compensated output has a fast response time although the steady-state condition). The results achieved so far were very encouraging regarding static or quasi-static weighting applications. Now we are already working on the new filtering algorithms having in mind dynamic weighting. In dynamic weighting systems conventional filtering methods employed have limitation in improving accuracy and throughput rate. The Kalman filter may provide effective alternative to the conventional method especially when the system is non-linear and low frequency noise is incorporated in the bandwidth of the useful signal.

We are also trying a new calibration system based on a very simple backpropagation neural network with as many inputs neurons as the number of load cells (four in our case), five hidden neurons and the output neurons defined by the number of bits required by the range for the weighting (0 to 1000kg requires at least 10 neurons) [10].

ACKNOWLEDGMENTS

We wish to thank CEL-Cachapuz Electrónica Lda, a electronic weighting equipment manufacturer, for promptly making available to us the platforms and the readout units used in this work.

VI. REFERENCES

[1] C. Couto, J. Higino, "Intelligent Signal Processing For Ratiometric Data Acquisition: A Low Cost Solution For Load Cells", Proceedings of IMEKO TC-4, Brussels, Belgium - May 1994.

[2] J.E. Brignell, "Adaptive filters in load cell response correction", Proceedings of Eurosensors VI-1992. University Southampton.

[3] "A high performance industrial weighting system", Application note 295, National Semiconductor.

[4] C. Couto, J. Higino, "Smart Load Cells based on Switched-Capacitors ", Proceedings of ICSPAT'94, Dallas, USA - October-1994.

[5] W. J. Shi, "Dynamic frequency compensation for transducers", Ph.D. Thesis, University of Southampton, 1992.

[6] J.H. Huijsing, "Integrated Smart Sensors", Sensors and Actuators, Vol A30, pp167-174, 1992.

[7] "A 20 - bit (1ppm) Linear Slope Integrating A/D Converter", Application Note 260, National Semiconductor.

[8] J.M.Riviére, J.M. Robert, F. Hermann, C. Aubum, "Modelling of smart sensors: Application to a smart temperature transmitter", France.

[9] R.W. Hamming, "Digital filters", Prentice-Hall International Editions, USA - 1989.

[10] A. Tavares, J. Higino, C. Couto "Calibration of a Multi-Load Cells Weighting System Based on Neural Networks", accepted to ICSPAT'95, Boston, USA. October-1995.