Calibration of a Multi-Load-Cells Weighting System Based on Neural Networks

Tavares, A.
Higino, J.
Couto, C.
Dep. Electrónica Industrial
Universidade do Minho - 4709 Braga Codex, Portugal
(FAX +351 53 615046, phone +351 53 615010)

ABSTRACT

Multi-load cells weighting systems are based on a platform supported by four or more load cells, normally in parallel inputting the same signal conditioning unit. Because of mechanical and electrical paralleling tuning the gain of a load cell affects the behavior of the others, making the calibration difficult and tedious, specially with weightbridges for cars and trucks, requiring the motion of heavy weights around large platforms. A software calibration can be achieved by calculation of the multiplying coefficients, which are given by the solution of a set of linear equations, operation easily performed by any general purpose microcomputer. This paper shows that the weight output can also be calculated using neural networks techniques. This new calibration system is based on a very simple backpropagation neural network with as many input neurons as the number of load cells (four in our case), six hidden neurons and one linear output neurons. We trained the system by examples (i.e. with examples data pairs) to determine the synapse connections.

Key Words: Neural Networks, Backpropagation Network, Learning Process, Nguyen-Widrow method.

INTRODUCTION

The use of load cells with digital outputs (smart load cells), i.e., with integrated signal processing, allows the gain adjustment to be a simple multiplication of the load cell numerical output by a coefficient, operation which does not affect the other load cells outputs. Each smart load cell uses a single chip RISC type microcontroller with very few other active and passive components around, taking advantage of the ratiometric functioning of the load cells. Once the microcontroller is needed, all architecture should be rethought in order to maximize its use, reducing the hardware and its specifications. The need for thermally stable circuits and components is minimized through the use of the same amplification chain for both signal and reference, together with software calibration and digital filtering.

However, this solution needs a cost effective signal processing circuit including amplification, A-to-D conversion and networking capabilities.

Having in mind industrial weighting applications where 6000 divisions are needed 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. For dynamic weighting a faster reading rate may be required but normally with lower resolution.

1290
PROC. OF INTERNATIONAL CONFERENCE ON SIGNAL PROCESSING APPLICATIONS AND TECHNOLOGY (ICSPAT'95). BOSTON, MA, USA

For the specifications referred there are already suitable components in the market, namely amplifiers and A-D converters [1,2], some of these even with networking facilities [2]. However some other facilities toward intelligent sensing [3,4] are also desirable: amplifier gain and offset adjustment controlled by software, scaling and eventually digital filtering of the converter results.

This contribution describes a feasibility study towards a solution for the problem taking advantage of the high performance low cost microcontrollers available today and of the ratiometric functioning of the load cells [2].

SIGNAL PROCESSING CIRCUIT

Smart Load Cell

Fig. 1 shows the conceptual idea of one Smart Load Cell:

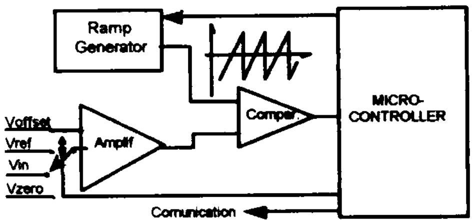

Fig.1- Circuit block diagram

The same amplifier chain is properly switched to deal with the zero, the load cell signal and the conversion reference. In this way the thermal stability requirements for the amplifier can be relaxed as a change in gain will affect all the three entities defining the A-D conversion. A simpler architecture can be used just enough to ensure that the gain remains constant during the conversion period.

A single ramp conversion was used because it requires the minimum hardware and it allows a higher rate of conversions. However this simple conversion technique is not intrinsically compensated as dual ramp conversion for example, requiring some pos-conversion processing specially in this case where the zero, the signal and the reference are allowed to change.

The counting associated with the single ramp A-D conversion, the control of the switches in the amplifier stage and the control of the ramping capacitor discharge are tasks to be performed by a microprocessor architecture. Some number crunching to work out the conversion result, which may include scaling, eventually some digital filtering and the communication with the outside world, are the other tasks to the microprocessor

<u>The Microcontroller:</u> The circuit was developed around 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.

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

The conversion time obtained was around 18ms, 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.

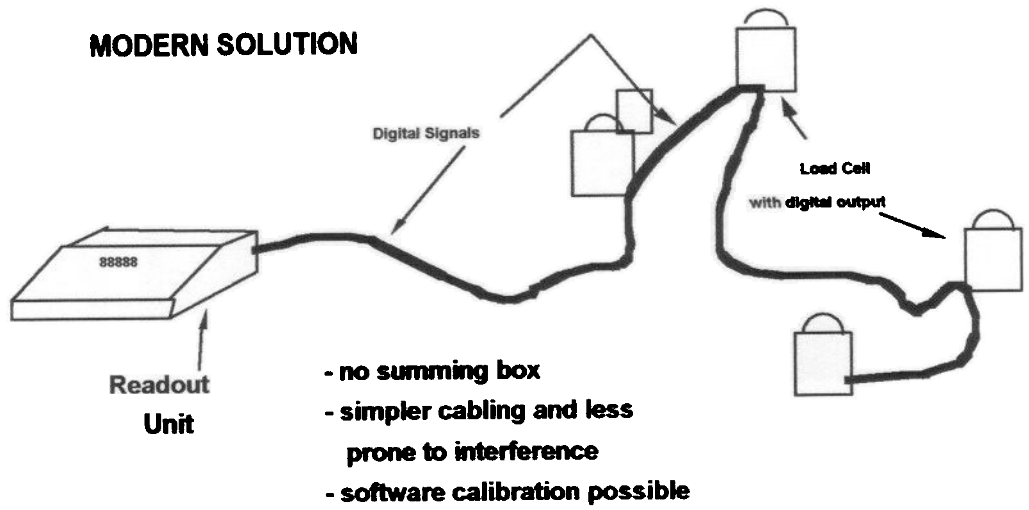

Fig. 2- Multi-Load-Cells weighting system

A SIMPLE CALIBRATION PROCESS

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).

$$\textbf{Mass} = K_1 * W_1 + K_2 * W_2 + K_3 * W_3 + K_4 * W_4$$

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.

NEURAL NETWORKS CALIBRATION

The reasons why another approach using a BPN (Back Propagation Network) network was tried, comes from the following facts:

-Three-layer feedforward networks are universal aproximators, that unlike other systems of functions such polynomials have associated a straightforward methods for adapting the net to approximate a given function [7].

-The generalization capability: The network tends to give reasonable answers when presented with inputs that they have never seen. That's to say, a new input will lead to an output similar to the correct output for input vectors used in training that are similar to the new input being presented [8].

- If an algorithm can be called a neural network, its chances of attracting attention are vastly increased (neural networks crazy) [6].

The training of a BPN has three stages: the feedforward of the input training pattern, the calculation and backpropagation of the associated error, and adjustment of the weights. After training, application of the net involves only the computation of the feedforward phase. Even if the training is slow, a trained net can produce its output very rapidly.

The action of the BPN is determined by two things: the architecture - number of input neurons, hidden neurons and outputs - and the value of the weights.

The number of the inputs and outputs neurons are determined by the application. The system built dictates 4 input neurons, one for each load cell, and one output neuron since the desired output is a numerical value in weighting range.

Early it was used binary outputs each one representing a bit of the desired numerical output. With this approach the network size increases a lot and slowing down the training and feedforward phases.

Having the number of input (n_input) and output neurons (n_output), the kernel of the problem becomes the choice of the hidden neurons:

- too few it will starve the network of the resources it needs to solve the problem [6].

- too many will increase the training time, perhaps making it so long that it becomes impossible to train adequately in a reasonable period of time (and sometimes may lead to overfitting problem) [6].

A good starting point could be the use of 2 hidden neurons (n_hidden) found by applying the geometric pyramid rule [6]:

$$\text{n\_hidden} = \sqrt{n\_input * n\_output}$$

With this architecture the network did not converge to any acceptable error probably because the problem is too So, a criterion for evaluating the performance (based on MSE-Mean Square) was fixed, the number of hidden neurons was then slightly increased, and net trained and tested for an acceptable performance. This process was repeated until the error was found acceptable, or no significant improvement was noted. After fixing the number of hidden neurons, the following strategy revealed good to improve the network performance:

- Initially the BNP is trained with a goal error slightly above the best one and final weights and biases recorded.

-Afterwards the training is repeated using the recorded values as the initial conditions, and the goal error is decreased. This process can be repeated till the final solution.

Applying the above ideas and the Nguyen-Widrow method to find the initial weights, the final setup for the training phase was:

Learning rate:	0.02
Error_goal:	10g
Momentum:	0.92
n_hidden:	6
Max_epoch:	10000

EXPERIMENTAL RESULTS

Below, the training and feedforward results over the following reduced training set are presented:

Normalized Training Set:

31	381	535	336
14	121	6	115
164	26	11	80
6	136	136	176
5kg	**10kg**	**10kg**	**20kg**

The optimal weights and biases for input/hidden layers were then found:

w_{ih}

0.4724	-0.2980	-0.4614	-0.6926
0.4508	0.0265	-0.1695	0.1433
0.9989	0.1822	0.0746	0.6048
0.7771	0.6920	-0.0642	-0.9339
-0.5336	-0.1758	-0.4256	0.0689
-0.3784	0.6830	-0.6433	-0.0030

b_{ih}=[0.9107, 0.4966, 0.1092, 0.7815, 0..2497, 0.6841]

The weights and biases for hidden/output layers (w_{ho},b_{ho}):

w_{ho}=[-0.7327, 1.4564, -0.0933, -0.3123, 0.3769, 0.7365]

b_{ho}=0.9150

In the feedforward phase, samples that do not participate in the training were tested giving a final error ±20gr.

The software was implemented in C++ on Windows environment (using OWL). To test the software and validate the final results, Neural Network Toolbox (MATLAB) was used to compare the performance.

CONCLUSIONS

The training was realized using few input samples (due the time consuming with too large training set).

The results obtained are very encouraging, since the final error is smaller than the one obtained with the first calibration process.

Others improvements were also registed:

- The weighting measurement was found less sensitive to the position of the load in the platform.

- Better linearity all over the weighting range.

The modification of the training method is now being tried in order to incorporate an adaptive learning - the learning rate is changed according to the convergence trend.

ACKNOWLEDGMENTS

We wish to thank CEL-Cachapuz Electrónica Lda, a electronic weighting equipment manufacturer, for promptly making available to us the a Multi-Load Cells weighting system used in this work.

REFERENCES

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

[2] "Intelligent Signal Processing for Ratiometric Data Acquisition: A Low Cost Solution for Load Cells", C. Couto, J. Higino, Proceedings of IMEKO TC-4, Brussels, Belgium, 1993.

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

[4] "Modelling of Smart Sensors: Application to a Smart Temperature Transmitter", J. M. Riviére, M. Robert, C. Aubrun, Proceedings of IMEKO TC-4, Brussels, Belgium, 1993.

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

[6] "Practical Neural Network Recipes in C++", Timothy Masters, Academic Press, 1993.

[7] "Neural Network and Fuzzy Logic Application in C/C++", S. Welstead, Wiley & Sons, 1994.

[8] "Fundamentals of Neural Networks", Lauren Fausett, Prentice-Hall, 1994.

[9] "Backpropagation in Perceptrons with Feedback", L. B. Almeida, Neural Computers, Berlin, Spring-Verlag, pp. 199-208, 1988.

[10] "A Method for Self-Determination of Adaptive Learning Rates in Back Propagation", M. Weir, Neural Networks, 4:371-379, 1991.

1295