Using Rotary Encoders to explain Distortion in Data Converters

Simon Bramble

Introduction

"I hear I forget, I see I remember, I do I understand". Yes the Ancient Chinese Philosophers had a lot of wisdom that is still applicable today. I wonder if the person who came up with that phrase had ever tried to understand the distortion characteristics of data converters. Probably not. Many articles are written on the subject and they all look very similar. However, none of them actually look at real data nor look at real world applications. I have often read such articles and, ask me the following week what INL and DNL is, and I cannot tell you. But then I have the disadvantage of not being Ancient, Chinese or a Philosopher. I guess I am doomed from the start. However, after a quick dabble in the lab, I have a deeper understanding of data converter characteristics that will hopefully last beyond the weekend.

Analogue to Digital Converters (ADCs) and Digital to Analogue Converters (DACs) both have limitations that the designer must be aware of. The performance of one application may be compromised by a particular distortion characteristic, whereas another application is insensitive to it and vice versa. It is then up to the designer to decide what device (and its shortcomings) is best for his application.

What Should Happen

Take the fate of a DAC. You write in a digital code of 0000 and the output should be 0V. As you increase the code at the input, the output should increase proportionately. So an 8 bit DAC will have 256 output steps corresponding to 256 input codes. You enter a code of 128 on the input and the output will be half of the full scale voltage. The full scale voltage is normally determined by the voltage at the reference input. So a DAC fed with a 5V reference will give a 2.5V output with an input code of 128. All very simple so far.

Monotonicity

What happens if the output voltage takes a step backwards when the input code increases? Alternatively, in an ADC, if the code decreases with increasing input voltage. This is described as non- monotonic behaviour – the output moves in the opposite direction to the input. Non monotonic behaviour can be catastrophic in a control system where the data converter is part of a feedback system. Negative feedback becomes positive feedback and the system may be prone to oscillation.

Offset Error

Many times the output will not be at 0.000V (to an accuracy of 8 bits) with an input code of 0000. If it is not, this is a measure of the offset voltage and is normally characterised in terms of Least Significant Bits (LSB). 1 LSB is the smallest step the DAC can make and is equal to the reference voltage divided by the total number of steps. In the case of an 8 bit DAC fed with a 5V reference, this will be 5/256 (= 19.5mV).

Gain Error

Likewise at the other end of the scale, the output should be at full scale. If it is not, there is a gain error. Now, it is worth noting that the transfer characteristic of a DAC adheres to the equation

$$y = mx + C$$

where x and y are the axes on the graph (corresponding to input code and output voltage respectively), C is the offset and m is the gradient (or gain).

This means that the y value (the output voltage in the case of a DAC) will contain an offset element and a gain error and it is important to note that the offset needs to be removed before the gain error can be calculated.

In Practice

To test this theory, an AS1505 8 bit quad DAC was wired up to an AS1530 12 bit ADC. Both devices were run from a 5V supply and this also provided the reference for both devices. Thus there would be no differential noise between the references. The circuit was built on copper clad board with one ground plane and star earthing to keep noise to a minimum. A simplified version of the circuit is shown in FIG 1.

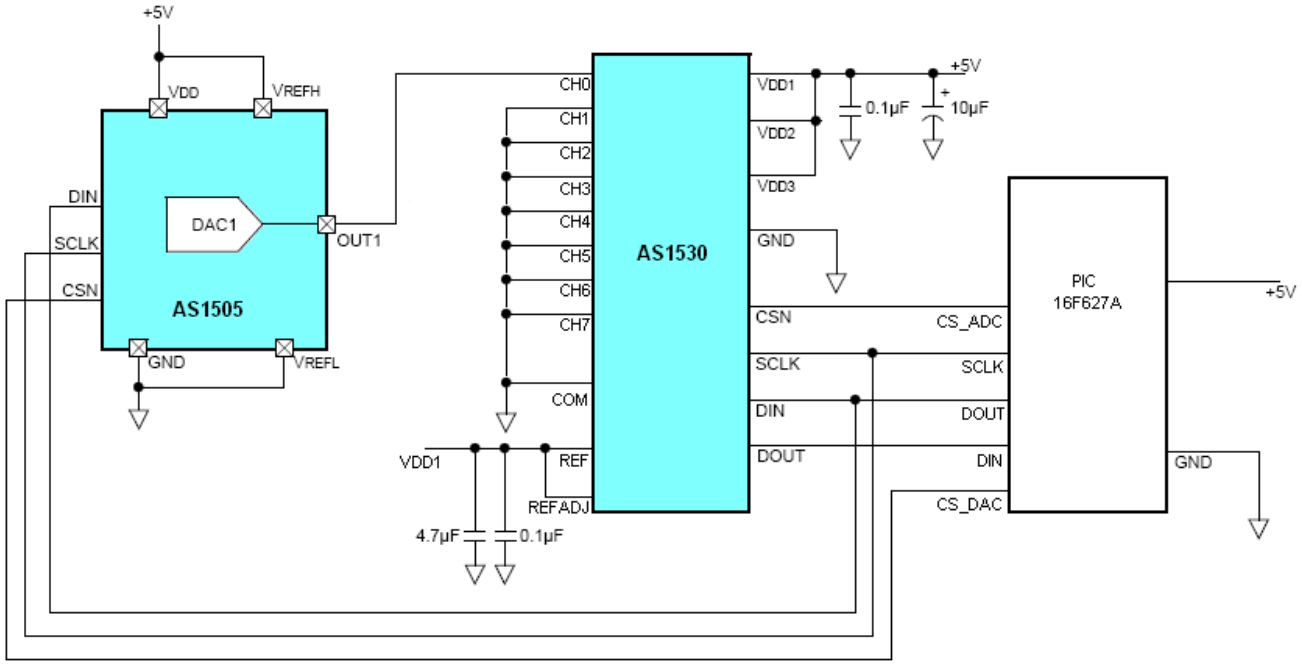

FIG 1

The DAC was incremented and the results taken at each step. The 12 bit results were transmitted over an RS 232 link and loaded into an Excel spreadsheet. Here the results were subjected to the power of Bill Gates and the distortion was closely monitored.

FIG 2 shows the reading (normalised to 8 bits) logged by the PC. The Y axis shows the output of the DAC in LSB. It could just as well be shown as a voltage by dividing by 256 and multiplying by 5, but for measuring offset errors etc, it is simpler to keep it scaled in LSB.

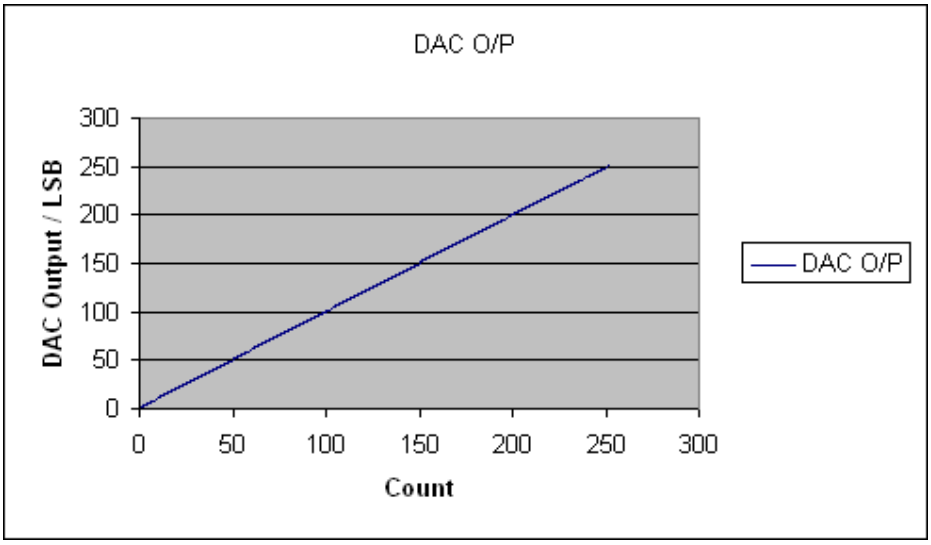

FIG 2

The first 9 results obtained are shown in FIG 3. The least significant byte is shown in column B and shows a result of 13, representing 13 LSBs. This was taken with a 12 bit ADC, so to get the spec in terms of an 8 bit device, this result had to be divided by 16, giving 0.8125 bits, shown in the 'Output' column. The offset error of the ADC was 6 LSB and the DAC was 0.5LSB (equals 8 LSB for a 12 bit system), so the worst case system offset reading is 14. Ours was 13, so we are just about within spec.

	A	B	C	D	E	F	G	H
1	**Count**	**LSB**	**MSB**	**Output**	**Corrected O/P**	**INL**	**DNL**	**Step Size**
2	0	13	0	0.8125	0	0		
3	1	29	0	1.8125	1.005714286	0.005714	0.005714286	1.005714
4	2	44	0	2.75	1.948571429	-0.05143	-0.057142857	0.942857
5	3	60	0	3.75	2.954285714	-0.04571	0.005714286	1.005714
6	4	76	0	4.75	3.96	-0.04	0.005714286	1.005714
7	5	92	0	5.75	4.965714286	-0.03429	0.005714286	1.005714
8	6	108	0	6.75	5.971428571	-0.02857	0.005714286	1.005714
9	7	123	0	7.6875	6.914285714	-0.08571	-0.057142857	0.942857
10	8	140	0	8.75	7.982857143	-0.01714	0.068571429	1.068571

FIG 3

The 'Corrected Output' shown in column E is an adjusted version of Column D with the gain and offset removed. The output (in LSB) with an input count of 253 was 252.375. The offset of 0.8125 was removed from this and then the result was multiplied by 253/(252.375-0.8125), to get the true output with gain and offset removed. Thus, in Excel, this can be represented as:

E2 = (D2-0.8125)*(253/(252.375-0.8125))

This equation was repeated down the entire 'E' column to yield a column of results with offset and gain calibrated out. As expected, Corrected Output matched the input Count when the input count was equal to 253. I need to look closer at the loops inside my C program to see why the downloaded results, although accurate, end at 253.

DNL / INL

The offset and gain errors give us no indication of how straight the curve is – how much the actual DAC characteristic deviates from the theoretical curve. This is the job of the Differential Non Linearity (DNL) and Integral Non Linearity (INL) specs.

The DNL is a measure of the deviation in the step sizes from the ideal of 1 LSB. FIG 4 illustrates this

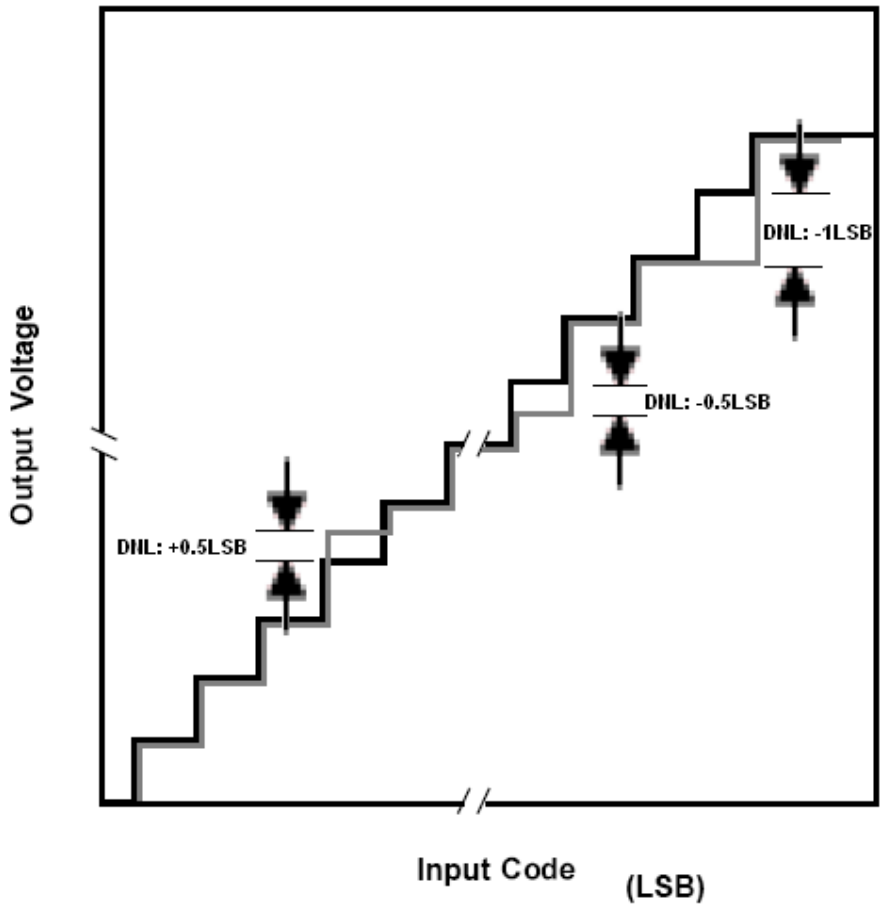

FIG 4

As the input code increases, the DAC output increases. The output should increase in steps of 1LSB. For an 8 bit system with a 5V reference, this is 19.5mV. If the DAC output increases by more than this, it has a DNL error, equal to the step size minus 19.5mV and measured in LSB.

Referring back to our Chinese proverb, I have looked at these graphs many times and can never remember what axis the distortion should be measured against. Applying some logic to the graphs will help us remember which axis to look at. Looking at FIG 4, it is obvious that the DAC code will *only increment in fixed step sizes* (it is digital), so any distortion measured has to be measured *on the analogue axis*. The digital axis will only ever increment one bit at a time. Likewise with an ADC, the analogue input is plotted on the x axis and the digital output code on the y axis which will only ever increment 1 LSB at a time. Therefore the DNL for an ADC is measured on the x axis (again, the analogue axis).

It is also worth noting that if the DNL reaches -1LSB a missing code results. This is illustrated in FIG 4. Missing codes are easier to picture with an ADC, whereby an increasing analogue input voltage causes an increasing

digital output code. If the output code does not change for a given incremental input voltage, the code is quite literally 'missed'

If the step size of the DAC continues to deviate from the ideal 1LSB, the curve of the DAC will deviate from the ideal straight line characteristic.

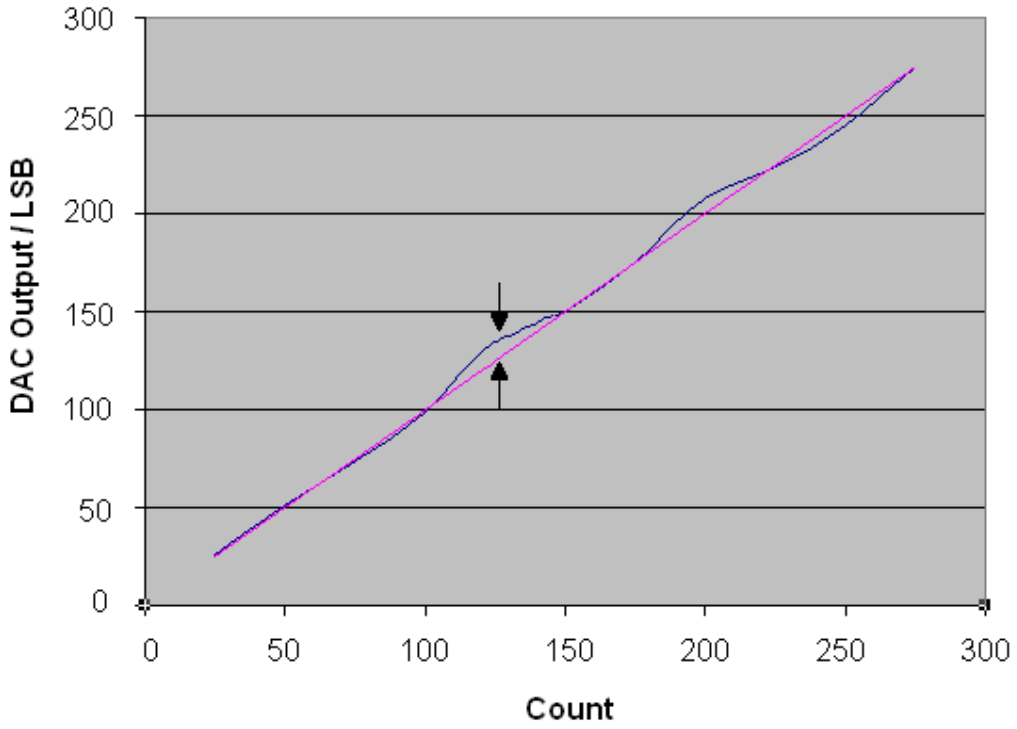

FIG 5

FIG 5 shows such a situation where the build up of positive DNL errors (+0.5LSB etc) has resulted in the blue curve deviating significantly from the ideal straight line. The worst case deviation is a measure of the DAC's INL. Again, INL is measured against the analogue axis and (in the case of a DAC) is measured as a vertical displacement from the ideal curve, measured in LSB. The INL in FIG 5 is approximately 10LSB.

If we were looking at an ADC, the analogue input would be on the x axis and the digital code would be on the y axis and the INL distortion would be measured against the x axis.

Column G in FIG 2 shows the DNL error of the DAC under test and is derived by taking the Corrected Output and subtracting the previous result then subtracting 1. Naturally, an accumulation of DNL error results in an INL error as we move away from the ideal curve.

In FIG 2, Column F shows INL and is derived by subtracting the theoretical output (Column A) from the Corrected Output (Column E).

So what do INL and DNL mean in practise? Imagine a lighting rig, mounted on a motorised platform, tracking a performer on stage. An angular rotation of the rig translates (approximately) to a linear distance on the stage. If the performer moves with a constant speed across the stage, the motor has to rotate with a constant angular velocity. If the rotary encoder regulating the motor's motion has a DNL error, the step size will not be constant, translating into a non linear motion of the light on the stage. Suddenly the performer disappears into the shadows.

As mentioned before, the INL error is an accumulation of DNL errors, so if the light has to move from one side of the stage to the other, if the motor's rotary encoder has an INL error, this means that the light will not light the correct part of the stage, either lighting to the left or the right of the desired spot.

To put this to the test, a practical experiment was rigged up in my garage. For the experiment, you will need some quick setting cement, a wide open space (at least 3m), an AS5040 evaluation kit, a cushion if the floor is cold and an understanding wife who doesn't mind the garage being out of action for a few days.

The rotary encoder was cemented into the garage floor and connected to a 1 metre long batten and the angle of rotation measured as the encoder incremented from code to code. FIG 6 shows how the angle was derived, by taking linear measurements (x) from the zero angle point.

I felt like a real engineering pioneer hoping this rudimentary experiment would prove my theory. Sitting in the garage, listening to internet radio through my wireless link, yes it was all very rudimentary.

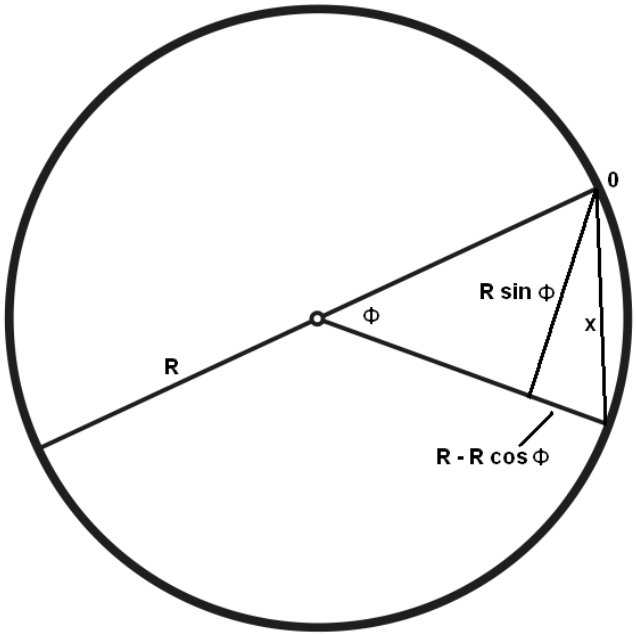

FIG 6

Then distance x (see FIG 6) was measured for each angle and the angle calculated from the formula

$$x^2 = (R\sin\phi)^2 + (R - R\cos\phi)^2$$

You then get a high school student to tell you that

$$\phi = \cos^{-1}\left(\frac{2 - x^2}{2}\right)$$

(the radius of the batten = 1)

Once again, this can be calculated swiftly in a spreadsheet with the resulting characteristic of the rotary encoder looking like FIG 7

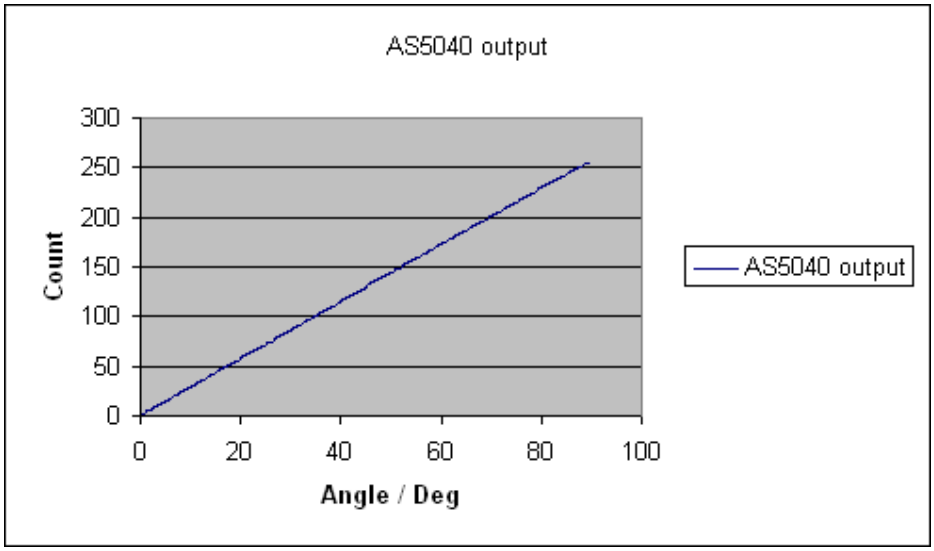

FIG 7

As expected the count increased with angular displacement. The DNL results were then calculated by subtracting one result from the previous result then subtracting 1LSB. The least significant byte in this case is 360/1024 (=0.3516), measured in degrees.

The DNL graph is shown in FIG 8

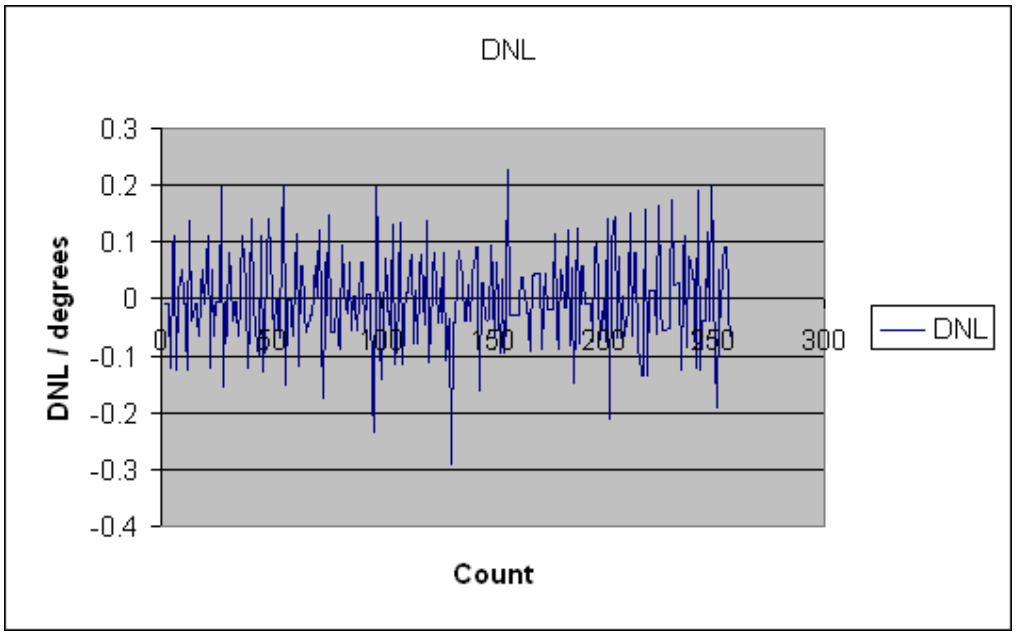

FIG 8

It can be seen that the DNL is at its worst around a count of 155 with a peak above 0.2 degrees. The photo in FIG 9 shows that this is indeed the case with steps from 154 to 155 and from 156 to 157 being about 1LSB. However, the step from 155 to 156 is much larger. There it is, as I live and breath, a DNL error. This stuff actually ties in with the theory.

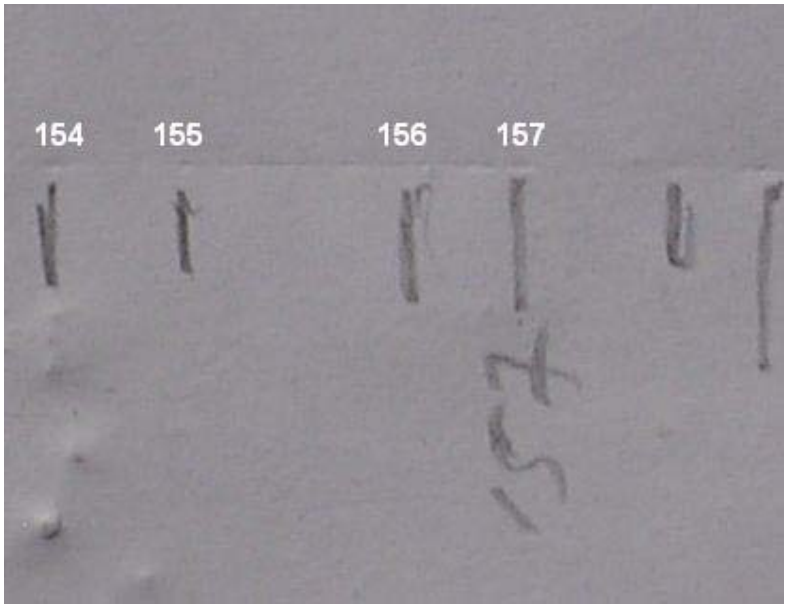

FIG 9

Conclusion

Data converters interface real world analogue signals to the digital world. An increasing input from zero to full scale, whether analogue or digital, is meant to produce a full span output, but this is not always the case. Offset and gain errors can be calibrated out and most ADCs and DACs are guaranteed monotonic and specify what resolution they can give with no missing codes. The harder characteristics to calibrate out are the INL and DNL errors, so before deciding on a data converter to use, these need to be closely examined as well as the test specifications they are measured under to see if they will inflict unwanted errors on the system.

Simon Bramble is available to expound the merits of data converters and rotary encoders at after dinner speeches, weddings and Bar Mitzvahs.