(SAN) Successive Approximation using a base 2 Normalization feedback loop

This application provides a micro controller friendly way of bringing two different data-sets together for quantitative comparison in the digital realm without the need to scale by difficult Division or Multiplication routines. The algorithm uses simple "Addition" and "Shift Right" to achieve a common mode BASE 2 representation of each data-set. The bit resolution is dynamic in the sense that for each desired bit of resolution, you only need to iterate the search loop N-bit number of times. Once both data-sets are 'normalized' to BASE 2 any further data comparisons can be easily handled.

Take for example you have a data-set that has a minimum value of 152 and a maximum value of 1774 coming from some outside sensor value. The goal in this example is to fit or scale the data-set so it can be compared to a second data-set that has a minimum value of 1000 and a maximum value of 2000.

Initially you might think to find the delta of both data-sets, with the first data-set having a delta of 1622 and the second data-set having a delta of 1000. Logic suggests that you scale the first data-set by 0.6165 to obtain a value that can be compared to the second data-set. While this would work, multiplying by 0.6165 is difficult in a micro controller and it doesn't allow much room for dynamic changes that might occur due to temperature variations or component tolerances if you needed to change the scaling.

If you can measure or know the upper and lower limits of your data-sets, then the SAN approach provides a dynamic solution that solves component mismatches and temperature variations.

The SAN algorithm has a few simple rules:

1. Calculate mid-reference value by adding the low-reference value and the high-reference value and then dividing by two (shift right)
2. If the data-set number is greater than the mid-reference, the output-bit is a "1"
3. If the data-set number is less than or equal to the mid-reference, the output-bit is a "0"
4. If the output-bit is "1", replace the low-reference with the mid-reference value
5. If the output-bit is "0", replace the high-reference with the mid-reference value
6. Repeat for each bit of resolution that you desire. The result of 1's and 0's will be the base 2 normalized value for that data-set

Example: (for our data-set value we'll pick a random value of 1200 and normalize it to 8-Bits

High-Reference	Low-Reference	Mid-Reference	Output-Bit	
1774	152	963	1	128
1774	963	1368	0	64
1368	963	1165	1	32
1368	1165	1266	0	16
1266	1165	1215	0	8
1215	1165	1190	1	4
1215	1190	1202	0	2
1202	1190	1196	1	1

The decimal equivalent of the binary value b'10100101' is 165 which represents 64.7% of the range (165/255). If we take the original value of 1200, subtract the offset of 152 to get 1048 and divide by the delta of the data-set 1622, we get 64.6% which is very close, but also remember this is only an 8-bit representation and by using the SAN algorithm the offset adjustment using the original values was automatically taken care of.

Ok, so what now?

Suppose we did the same thing to the second data-set? ... in this case we'll pick a random value of 1800 and apply the same algorithm we did to the first data-set.

High-Reference	Low-Reference	Mid-Reference	Output-Bit	
2000	1000	1500	1	128
2000	1500	1750	1	64
2000	1750	1875	0	32
1875	1750	1812	0	16
1812	1750	1781	1	8
1812	1781	1796	1	4
1812	1796	1804	0	2
1804	1796	1800	0	1

The decimal equivalent of the binary value b'11001100' is 204 which represents 80% of the range (204/255). If we take the original value of 1800, subtract the offset of 1000 to get 800 and divide by 1000, the delta of the data-set we get 80%

Now since the two data-sets have been normalized to BASE 2 it's now easy to determine the amount of error between both data-sets. In the case of this example, the first data-set is from an external potentiometer that represents a physical position of a servo. The second data-set is from an input pulse that represents the position we want to move the servo to. We can now see that by looking at the two BASE 2 numbers the position that we want to be is 204, while the position we are currently at is 165. This tells us which direction to turn the servo motor so that both BASE 2 numbers will eventually agree with one another.