AN ANALYSIS OF VARIOUS DIGITAL FILTER TYPES FOR USE AS MATCHED PRE-SAMPLE FILTERS IN DATA ENCODERS.

William T. Hicks, PE
Newtown, PA

ABSTRACT

The need for precise gain and phase matching in multi-channel data sampling systems can result in very strict design requirements for presample or anti-aliasing filters. The traditional use of active RC-type filters is expensive, especially when performance requirements are tight and when operation over a wide environmental temperature range is required. New Digital Signal Processing (DSP) techniques have provided an opportunity for cost reduction and/or performance improvements in these types of applications. This paper summarizes the results of an evaluation of various digital filter types used as matched presample filters in data sampling systems.[1]

KEYWORDS

Signal Conditioning Filters, Anti-aliasing Filters, Digital Filters, Digital Signal Processing.

INTRODUCTION

During the testing of new aircraft, the vehicle is equipped with numerous transducers or sensors. The sensors measure various parameters such as engine temperature, wing pressure, and the strains and vibrations of various physical supports. The outputs of these sensors are recorded and saved for later analysis. Since this data is analyzed by digital computers, the data must be converted from the analog output signal of the sensor to some digital computer compatible form. The point in the signal chain, at which the analog parameters are converted to digital signals, can be at any location from the transducer output to the computer input. One of the most common places for digitization, is after some signal amplification of the transducer output, and before the various signals are multiplexed into one data stream for recording.

To be digitized, a signal must be sampled fast enough to ensure accurate recording of the information content of the signal, and slow enough to keep the system bandwidth as low as possible. This rate must be higher than any signal that is present in the input

signal stream, including noise. According to the Nyquist sampling theorem, the lowest sample rate to avoid aliasing is twice the maximum signal frequency.

To prevent aliasing of unwanted signals, a filter is installed between the analog signal and the analog to digital converter (A/D). This filter must have the property of passing the signal of interest as undistorted as possible while blocking the noise completely. If a perfect low pass filter existed, the filter could pass everything up to the half sample rate frequency and nothing beyond this point. Because no filter of this type exists, some compromise filter must be chosen. To accommodate the compromise filter, the output of the filter is sampled at five times the cutoff frequency, which prevents aliasing of signals up to two and one half times the maximum signal frequency. This process is called over sampling. While filters will still allow some noise through the filter above the aliasing frequency, the level at this frequency is normally so low that it is below the minimum signal level of the signal to be analyzed.

Instrumentation of two and three dimensional signals is usually done with orthogonal mounted single axis linear sensors. During post processing the output vectors are summed to get the amplitude and direction of the true signal.

Anti-aliasing filtering of the transducer outputs has historically been implemented with analog filters with good matched response. Typically a six pole butterworth filter with a cutoff frequency equal to the maximum signal frequency has been used. However, analog filters require high accuracy, which is difficult to maintain over a range of temperatures and long periods of time. For example, assume two filters with one filter assumed to be off by -5% and the other off by +5% in cutoff frequency (+/- 5% error in center frequency). Now with a half cutoff frequency -5% input, the output is sampled when the -5% filter is at its peak. The other filter is sampled at the same time and will be 22 degrees off. This results in an output of 43% of the correct value, mostly due to the phase error. This gives a vector sum with 23% amplitude error. Next consider the same situation, however with a 2% mismatch in cutoff frequency (+/- 1%). In this case the half cutoff frequency error of the vector sum is only 2 counts. However with this same mismatch and a signal at cutoff frequency, the error increases to 3.3% or 137 counts. Therefore, analog filters must be very accurate or the filter and sample range must be much higher than the signal bandwidth.

As an alternative to analog filtering, a digital signal processor (DSP) device can be used to do the filtering by mathematical calculations. Digital filters are computers that process the data with formulas that do not change with time or temperature. Removing the analog filter from the system removes its fixed errors and environmental errors. However, a DSP filter requires an increased sample rate from the A/D. Since an A/D is already required in a system with analog filters, the only difference with digital

filters is increased cost or reduced performance for the A/D. This design change usually has no effect on the system budget since the cost of the DSP chip is now less than the cost of a well matched pair or triplet of analog filters. This paper describes various DSP implementations for telemetry, in order to determine some selection criteria for this type of system.

BACKGROUND

As mentioned earlier, the signal conditioning problem has historically been solved with a 6 pole butterworth analog filter, thus I evaluated DSP filters which closely match the gain response properties of the 6 pole butterworth filter. There are two basic types of digital filters: infinite impulse response (IIR) and finite impulse response (FIR).

IIR FILTERS

The digital IIR filter gets its name because the response to an initial impulse can last for an infinite time after excitation. In practice, because of quantified arithmetic, the output usually either settles to a fixed value or oscillates, the later of which is not normally desired for a filter.

An IIR type filter can be designed directly from known analog filter types. This is done by a process called a transformation. An analog to IIR digital transformation only requires selecting a transformation algorithm and knowing the parameters of the digital filter (sampling rate, pass band cutoff frequency and ripple, and stop band frequency and attenuation). For low pass filters, the bilinear transform gives an accurate match of the gain response of the analog filter, provided the sampling rate is not too close to the cutoff frequency.

Referring to Figure 1, the IIR filter works as follows. The input signal (a 16 bit value) enters the first stage of the three stage filter, and is multiplied by b10. The previous two input values (held in memory, or "delayed"), are multiplied by b11 and b12 respectively. The three values are then added with two delayed output values (multiplied by a11 and a12 respectively) and amplified by two (scaled), to get the output of the first stage. The output and the two delayed values of the first stage then become the inputs to the second stage and the outputs of the second stage are the inputs of the third stage. This structure is referred to as the Cascade Form I.[2, p526]

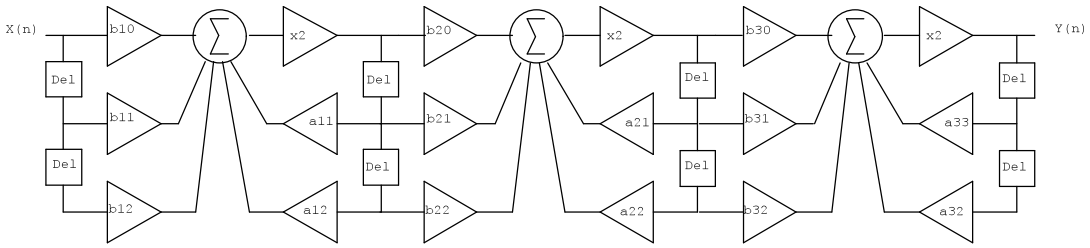

Figure 1
IIR Filter Block Diagram

The IIR filter requires at least 29 clock cycles per data point for an interrupt driven DSP program. This includes 11 overhead steps and 18 loop calculation steps. With a 16 MHZ clock, there are 32 clock cycles available in a 2 channel 250 KSPS/channel system, therefore there is just enough time to perform the calculation.

FIR FILTERS

The second major type of digital filter is the FIR filter. The FIR filter gets its name because it responds to an impulse input for only a finite amount of time. This is because the filter has no feedback of the output signal to be mixed with the input signal. The output is only a linear combination of present and past inputs (see Figure 2). There is no analog equivalent to this type of filter. One problem with this type of filter is that if the sampling frequency is much higher than the cutoff frequency, a very high order filter is required, resulting in a lot of clock cycles to calculate each output point. However, because the output values do not rely on knowledge of previous output values, then decimating of the input sample frequency to get a lower rate output sample frequency can be done by only calculating those desired values. This can result in a much quicker filter. Another option is multistage FIR filters, with the first stages acting as anti-aliasing filters for later stages, making the sum of stages shorter than using one stage.

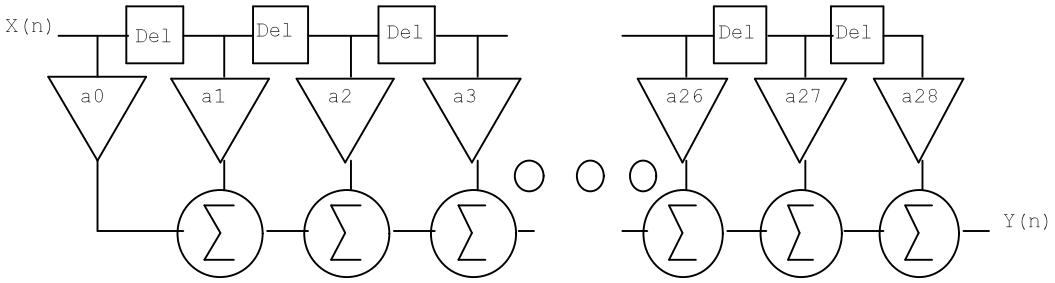

Figure 2
FIR Filter Block Diagram

Referring to Figure 2, the 29th order FIR filter works as follows. The 16 bit input signal is multiplied by a0 and added to the previous input multiplied by a1. This sum is then added to the input from two time periods ago (multiplied by a2) and so on until the sample from 28 periods ago is multiplied by a28 and added into the sum.[3, p68]

The FIR filter takes at least 38 steps to do one calculation. This consists of 10 overhead cycles and 28 cycles of the calculation loop. Calculations could not be done in the time allowed at 16 MHZ without decimation. With the decimation value of five specified for this project, the average number of clock cycles is about 8 per input point.

DECIMATION

Because the DSP filter is after the A/D converter, there can still be frequency fold back or aliasing problems. If the sample rate of the A/D is the same as for the analog filter case, then the same accurate analog anti-aliasing filter would be required. To get around this problem, the input is sampled at a much higher rate than the output sample rate will be. This allows a much simpler or even no input anti-aliasing filter. However, for the output rate of the DSP filter to be at the proper rate, a process known as decimation is used. In this process, only selected output values from the filter output calculations are used. If the input sample rate is some whole number multiple of the output rate, then this process reduces to a very simple procedure of only outputting every Nth output calculation, where N is this ratio.[3, pp87,93,126]

While over sampling can reduce the requirement for the analog anti-aliasing filter, it can only eliminate it in cases of very large over sampling ratios, where noise is always below the low frequency system noise level. Because this filter is so much a function of specific system design, it was not considered in this study, but should be analyzed as part of the total system error in any final system design problem.

CALCULATION ERRORS

When the analog signal is converted to a digital signal, noise is introduced because the analog signal, represented by the digital word, is limited to discrete values. This quantization noise is an initial source of error in the digital portion of the system.

With any computer math calculation, there are also noises introduced that are associated with how the mathematical process is implemented. With the Analog Devices DSP chip, the multiplier-accumulator section has a 40 bit resultant register, and does all accumulations in 40 bit math. There is also a rounding instruction to get a 16 bit result. The rounding need only be done when the result must be moved from the multiplier-accumulator to an output or some other processor component, such as the

shifter. The FIR filters only need rounding, when the calculation must be moved to the output. The IIR filters in contrast, use the shifter for multiplication by two for scaling. This results in rounding between each of the three stages of the filter. The IIR filter is thus more prone to rounding errors than the FIR filter.

The multiplier-accumulator section used in FIR and IIR filters has overflow detection to find out if the results of an operation have exceeded the length of the register. Overflow would result in gross error due to sign inversion. Since the IIR filter also uses the shifter, which does not have an overflow detector, one must limit the input word length to 15 bits (with the 16 bit math) to avoid overflow. This however, decrease the signal to noise ratio.

TEST PLAN

SYSTEM LIMITS

The input is a 12 bit 2's complement signed value, which is sign shifted and zero filled by the DSP device input circuitry to be a 16 bit number. With 12 bits, there is 72 dB of dynamic signal range. The DSP is an Analog Devices ADSP2101 16 bit fixed point processor and is using 1.15 number representation for numbers (15 bit fraction plus sign). The DSP instruction clock rate is 16 MCPS. The filter cutoff frequency is 10 KHZ, the sample rate is 250 KSPS, and the output rate after decimation is 50 KSPS. No decimation was done to the data in order to make it easier to analyze.

TEST METHOD

The input to the system comprises two scalars that represent the X and Y orthogonal information of a two dimensional vector. The input signal consists of level shifts from zero to some portion of full scale. By varying the amplitudes of the two input signals, the magnitude and angle of the output vector is varied. The effects of various filter configurations versus various input amplitude and angle combinations can thus be studied.

FIGURES OF MERIT

The output amplitude of the vector sum of the two channels is calculated using the Euclidean distance. This is compared to the output amplitude of the total vector going through the same DSP processor, on a point by point basis. The maximum error point is then used for comparison.

SELECTION OF TEST VECTORS

I used 3-4-5 and 5-12-13 right triangles for the test and reference vectors, all three calculations can add to the error, as would be expected in most real signals. Selecting these values allows exact binary number inputs to be generated, which allows all error to be from the DSP mathematics and none from the input data.

FILTER TYPES USED

Three digital filter types were selected. First a direct substitution 6 pole butterworth IIR filter was analyzed, because of its close correlation to existing analog filter designs. The Cascade Form I IIR filter was chosen for its easy implementation in the Analog Devices DSP chip, requiring only 18 clock cycles for the filter output calculation. Next a 6 pole bessel IIR filter was studied, because of the better linear phase response of this filter type and its occasional use in analog filter anti-aliasing designs. Finally an FIR Kaiser windowed filter was analyzed. The Kaiser window was selected because of its ability to adjust both main lobe and side lobe width with two independent parameters.[4, pp452-457] The program used to select the coefficients for the Kaiser windowed filter has selectable side lobe peak and stop band attenuation and has 6dB/octave rolloff of stop band peaks.[5, p2.18]

TEST RESULTS

FIR

From the data generated, the FIR filter has one Least Significant Bit (LSB) or less error in both angle tests. This is as good as is theoretically possible in fixed point math. The error as a percent of full scale is 1/19881 x 100% = 0.005% or 0.2 LSB in the 12 bit A/D signal.

The calculation done by an FIR filter program, consists only of a long stream of multiply-accumulate instructions, followed by a rounding of the results. Error is only introduced in the final rounding to 16 bits, because the accumulator is 40 bits wide and the sum of 16 bit wide multiplies is always perfectly represented. This means that the three numbers in the error calculation are at most one half LSB off from being perfect.

IIR BUTTERWORTH

The IIR Butterworth step response error was almost 21 counts on the 3-4-5 test, with 6 counts of final (stabilized) error. For the 5-12-13 test, the error was almost 31 counts, with almost 16 counts of steady state error. As a percent of full scale the worst error is 30.8/26478 x 100% = 0.12%, or about 5 counts of the 12 bit A/D input word.

Three sources of computational errors are present in IIR filters. First, each stage saves two old values with their errors, to be combined with new data in each new calculation. Second, this filter has three stages, each of which feeds the next, and errors of one stage can be multiplied by the succeeding stages. Finally, each stage has a scaling factor that is necessary, because at least one coefficient in each stage is greater than one, while the DSP is using fractional math. The scaling is accomplished by the coefficients being divided by two, then the result multiplied by two. This results in the loss of signal level of one LSB in each stage, with the error being fed back into each stage and as the input of the next. The accumulative effect is the loss of at least a factor of 8, just from these shifts.

IIR BESSEL

The error of the 5-12-13 test for the Bessel IIR filter was about 6 counts maximum, and about 3 counts after stabilization. For the 3-4-5 filter the results were a maximum error of 8 counts, which was in the steady state region. This error as a percent of full scale is 8/20454 x 100% = 0.04% or less than 2 counts of error in the 12 bit A/D signal.

Comparing the two IIR filters, the butterworth filter seems to have a much worse error. In comparing the coefficients of each filter, it was observed that the butterworth filter uses much less input signal (less than half for each stage), and a corresponding greater amount of feedback. This results in the reduced output signal of each stage being used as a more predominant value in the calculation, therefore, the calculation is more prone to error.

OVERDRIVE TESTS

The input step function was increased until the output started oscillating. This was to test the stability of the DSP performing the various mathematical operations. As expected, the FIR filter did not oscillate at all, and worked with a full scale input. The bessel filter failed with a 99% of full scale input. The butterworth filter failed with a 90% of full scale input. The IIR filters oscillated at almost full scale peak to peak,

because of the inability of the Analog Devices DSP to detect overflow in shift operations.

CONCLUSIONS

This study has been comparing digital filters used in telemetry signal conditioning. Based on the tests done in this study, it is obvious that the FIR filter gives much less errors than either IIR filter tested, and would be the filter of first choice. The butterworth error of 5 counts would reduce system accuracy to less than 10 bits. The bessel error of two counts would reduce the system to 11 bits. In both cases, the worst case error might be higher. For the FIR filter with its 0.2 LSB error, the filter can be considered to be transparent to the system from an error standpoint. A comparison of the three filters errors is shown in figure 3. Finally, as discussed earlier, any of these filters would outperform analog filters in consistency of accuracy and stability.

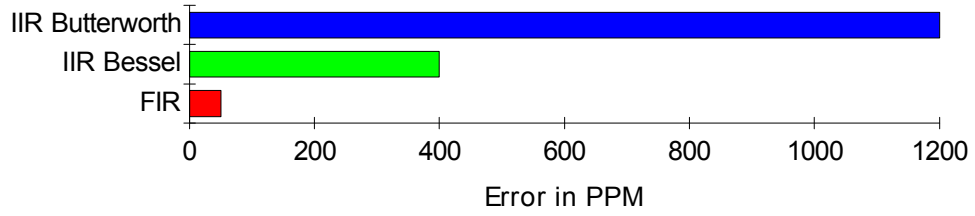

Figure 3
Comparison of filter errors

Other factors must be considered in a real system. This would be especially true for a system with selectable filter cutoff frequencies and one sample rate. This could result in FIR coefficient files that are so long that the DSP could not handle them. Then some type of multi-stage FIR filter might be required, with its resultant increased error, due to each stage having to be rounded off before the next stage.

ACKNOWLEDGMENTS

I wish to thank my paper advisor, Dr. Dominique M. Richard, for his guidance in all phases of the work on this paper. I would also like to thank Mr. Jerry Purcell of Momentum Data Systems, for his assistance in using and understanding his company's software programs.

REFERENCES

1 This paper is a summary of a paper I did for my Master of Engineering degree at The Pennsylvania State University. Further details and data are available from the author.

2 Babst, Jere, ed. Digital Signal Processing Applications Using the ADSP-2100 Family, Volume 2, Englewood Cliffs, NJ: Prentice-Hall, Inc., 1995

3 Mar, Amy, ed. Digital Signal Processing Applications Using the ADSP-2100 Family, Volume 1, Englewood Cliffs, NJ: Prentice-Hall, Inc., 1992

4 Oppenheim, Alan V., Ronald W. Schafer, Discrete-Time Signal Processing, Englewood Cliffs, NJ: Prentice-Hall, Inc., 1989

5 QEDesign 1000 for Windows, Version 2.0, (program and instructions) Costa Mesa, CA: Momentum Data Systems, 1993