MULTICHANNEL TELEMETRY DATA ACQUISITION USING A SYNCHRONOUS DSP FILTERING APPROACH

Paul Dourbal, Ivan Goranov, Chris Dehmelt
L-3 Communications - Telemetry East

ABSTRACT

Today's telemetry data acquisition systems demand an increased number of highly configurable acquisition channels delivering synchronized data samples in a specific format [1]. At the same time, there are additional requirements that these systems be compact and cost effective. In order to design such systems, novel approaches in digital signal processing are required.

In this paper, we compare the typical analog signal sampling approach used in current systems with a flexible system architecture that is based on digital signal processing, allowing for precise synchronization and simultaneous sampling. An appropriate DSP filter structure is discussed, and a Xilinx FPGA based implementation example of this multi-channel filter that utilizes a minimal number of key signal processing components while easing the analog component requirements is presented.

ANALOG AND DSP-BASED FILTER DESIGN APPROACH

For critical vehicle parameters such as vibration, acceleration and pressure, traditional telemetry data acquisition systems sample and digitize analog continuous signals that have passed through analog 6-pole Butterworth low pass filter. One of the key traits of this type of signal conditioning is its analog filtering, which is bulky from an implementation perspective, and whose cutoff frequencies can't be easily configurable to match various channel bandwidths. On the other hand, this approach ensures that a continuous in time and filtered signal is always present at the sampling device which allows taking samples of a signal at any arbitrary point of time.

A DSP based filter is implemented as a chain of steps, consisting of an analog wideband anti-alias filter, sampling device and digital filter. Such filters, compared to analog filters with similar characteristics, require less hardware real estate, can easily by reused for multiple channels, and are much more flexible with easily configurable cutoff frequencies and other parameters.

There is, however, a significant problem. Since a digital filter works at a specific clock rate equal to or proportional to the sampling rate, only discrete signal samples are present at the output of the filter that does not allow taking samples of a signal at any arbitrary point of time. Nonetheless, with the appropriate interpolation techniques, this problem can be solved as will be shown.

For the sample rate $\boldsymbol{F_s}$ [Hz], the digital filter produces a new sample every $\boldsymbol{T = \frac{1}{F_s}}$ [sec.]. If next signal value acquisition is required at the time moment just before the next sample is produced, the maximum relative error becomes

$$\varepsilon = \boldsymbol{\lim_{\tau \to T_-} S(\tau) - S(0) = S(T) - S(0) = S\left(\frac{1}{F_s}\right) - S(0)}\ , \qquad \text{(Eqn. 1)}$$

The error due to discrete sample time a difference between last available filter output sample value and the sample value that would be available at the point in time when the acquisition is required.

This error introduces a noise proportional to the signal amplitude and bandwidth.

For a signal with 3 dB bandwidth equal to the filter cutoff frequency $\boldsymbol{F_c}$ [Hz] and amplitude $\boldsymbol{A}$, and filter sampling rate $\boldsymbol{F_s}$ [Hz], the relative error is

$$\varepsilon = \left|\frac{\boldsymbol{A}\cdot\sin\left(\frac{\pi\cdot\boldsymbol{F_c}}{2\cdot\boldsymbol{F_s}}\right)}{\boldsymbol{A}}\right| = \sin\left(\frac{\pi\cdot\boldsymbol{F_c}}{2\cdot\boldsymbol{F_s}}\right) \approx \frac{\pi\cdot\boldsymbol{F_c}}{2\cdot\boldsymbol{F_s}}\ , \qquad \text{(Eqn. 2)}$$

which corresponds to the signal-to-noise ratio introduced by the time quantization

$$E = -20\cdot log_{10}\frac{\pi\cdot\boldsymbol{F_c}}{2\cdot\boldsymbol{F_s}} = 20\cdot log_{10}\boldsymbol{F_s} - 0.1961 - 20\cdot log_{10}\boldsymbol{F_c}\ \text{[dB]}, \qquad \text{(Eqn. 3)}$$

Let assume that the digital system clock is 100 MHz, and has to provide 32 channel filters with highest cutoff frequency $\boldsymbol{F_c = 30}$ [kHz] $\boldsymbol{= 0.03}$ [MHz], implemented so that one filter section is reused 4 times and the same section is multiplexed by all filters. Thus, the section is used 128 times. These constraints make the sampling rate $\boldsymbol{F_s = \frac{100\,[MHz]}{128} = 0.78125}$ [MHz]. It can be seen that, due to sample time mismatch, those filters have internal signal to noise ratio

$$E = 20\cdot log_{10}\boldsymbol{0.78125} - 0.1961 - 20\cdot log_{10}\boldsymbol{0.3} = \boldsymbol{24.39\ dB}, \qquad \text{(Eqn. 4)}$$

This is surely too noisy of a signal and the situation must be addressed by introducing an interpolation module which takes the filtered signal and interpolates it to a new sampling rate. Suppose the highest system clock frequency, 100 MHz, is desired as an output sampling rate. This decreases the error and signal to noise ratio becomes

$$E' = 20\cdot log_{10}\boldsymbol{100.0} - 0.1961 - 20\cdot log_{10}\boldsymbol{0.03} = \boldsymbol{66.54\ dB} \qquad \text{(Eqn. 5)}$$

The actual signal-to-noise ratio will be lower because of the effect of signal digital quantizing present in any digital filter. The goal of the interpolator is to minimize the sample time mismatch related error thus leaving the quantizing error a dominant component of the filter self-noise.

FPGA-BASED MULTI-CHANNEL FILTER IMPLEMENTATION

The digital filter consists of the following components: 16-channel analog-to-digital converter, 16-to-1 and 2-to-1 digital multiplexers, memory for filter coefficients and gains, 64-channel 2nd order recursive filter section [2], 1-to-16 de-multiplexer and 16 single channel interpolators [3].

Analog signals from 16 independent analog channels pass to the 16-channel Analog-to-Digital Converter input through anti-aliasing analog filters with a cutoff frequency of 30 kHz and minimum frequency response slope 18 dB/octave. The multichannel digitized signal goes to the 64-channel filter section through the 16-to-1 and 2-to-1 multiplexers where the latter multiplexer takes intermediate results and redirects them to the same (single) filter section. The output of the final filter stage (the 3-rd section of a 6-pole IIR, or the 4-th one of an 8-pole IIR) goes to the channel interpolators through the 1-to-16 de-multiplexer.

A 64-channel 2nd order IIR filter section implemented in Xilinx FPGA is shown in Figure 1

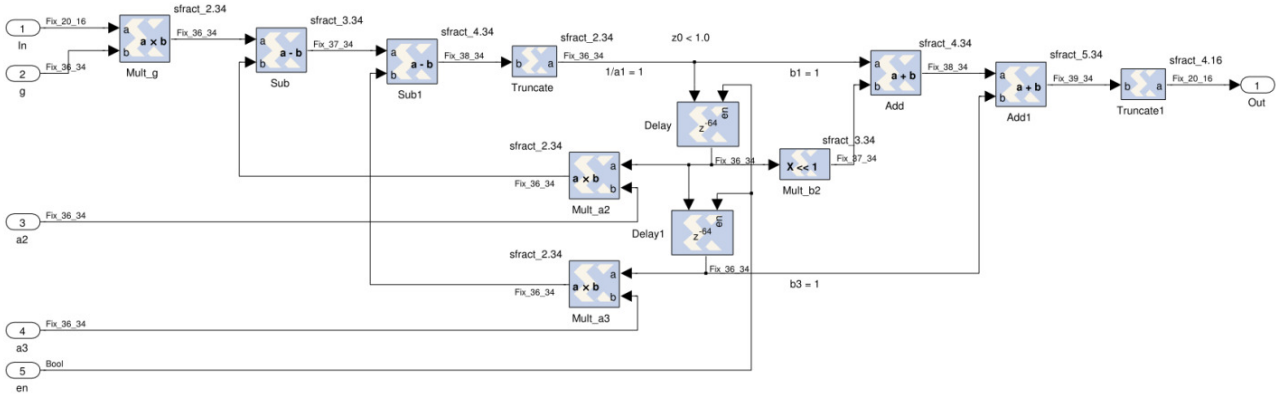

Figure 1- Matlab/Simulink model of a 64-channel 2nd order IIR filter section implemented in Xilinx FPGA

This multi-channel filter section requires three 36 x 36 bit multipliers, four adders/subtracters and two delay lines of 64 positions each.

A Xilinx FPGA based implementation of the interpolator is shown on the Figure 2.

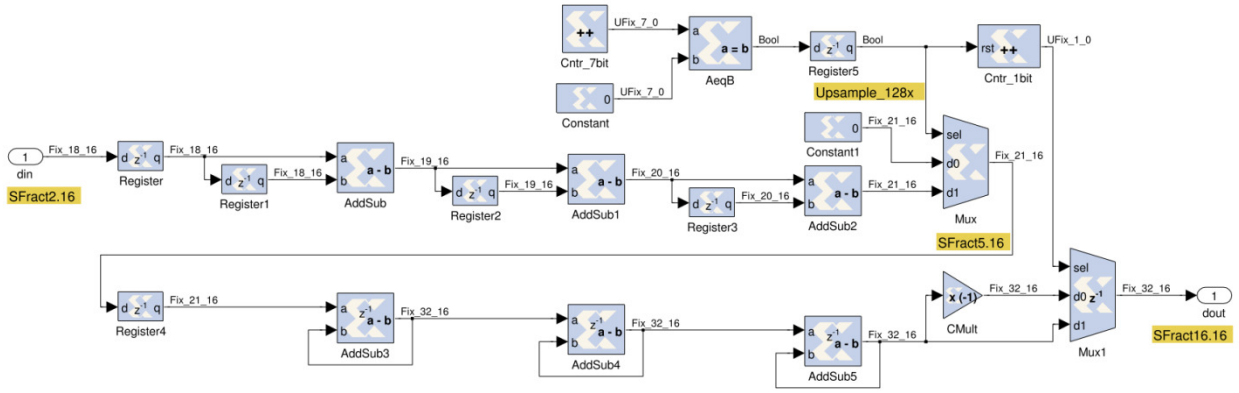

Figure 2- Matlab/Simulink model of a CIC interpolator implemented in Xilinx FPGA

Figure 3, Figure 4, Figure 5 and Figure 6 show the Matlab/Simulink simulation results for the described filter.

The top trace of Figure 3 represents a reference 30 KHz sine signal passed through an analog filter with the cutoff frequency 30 KHz; the 2nd trace from the top is the 30 KHz sine digitized and filtered with the digital filter; the 3rd trace depicts the ratio of the difference between the outputs of the two filters to the maximum output amplitude of the digital filter, and the bottom

trace is the square of the value of the 3rd trace in dB which represents the filter's signal-to-noise ratio caused by sample time mismatch.

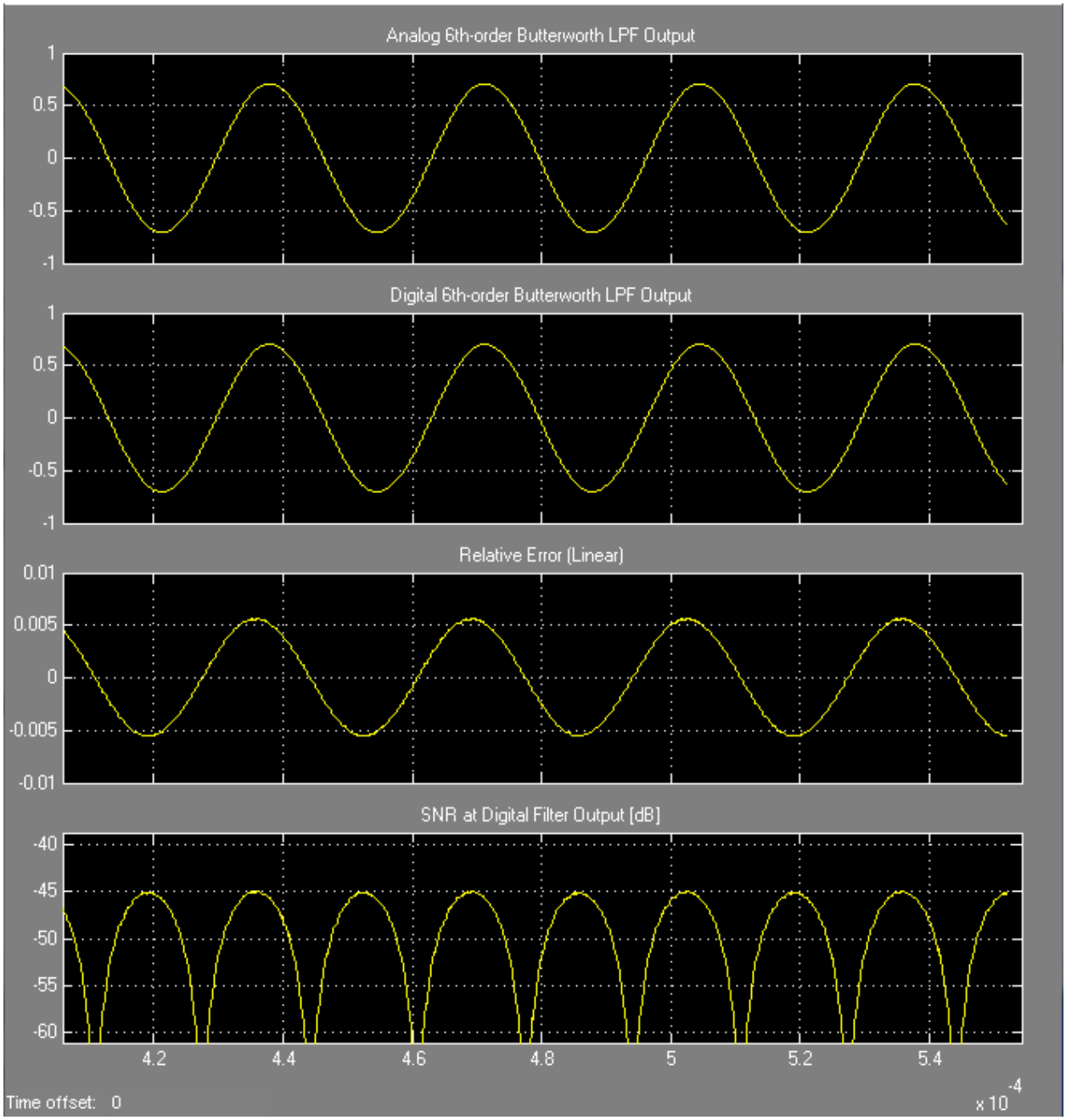

Figure 3- Filter signals and signal-to-noise ratio. Cutoff frequency 30 kHz, input – 30 kHz sine wave.

The top trace of Figure 4 represents a reference step function signal passed through an analog filter with the cutoff frequency 30 KHz, the 2nd trace from the top is the step function digitized and filtered with the digital filter, the 3rd trace depicts the ratio of the difference between the outputs of the two filters to the maximum output amplitude of the digital filter, and the bottom

trace is the square of the value of the 3rd trace in dB which represents filter's signal-to-noise ratio caused by sample time mismatch.

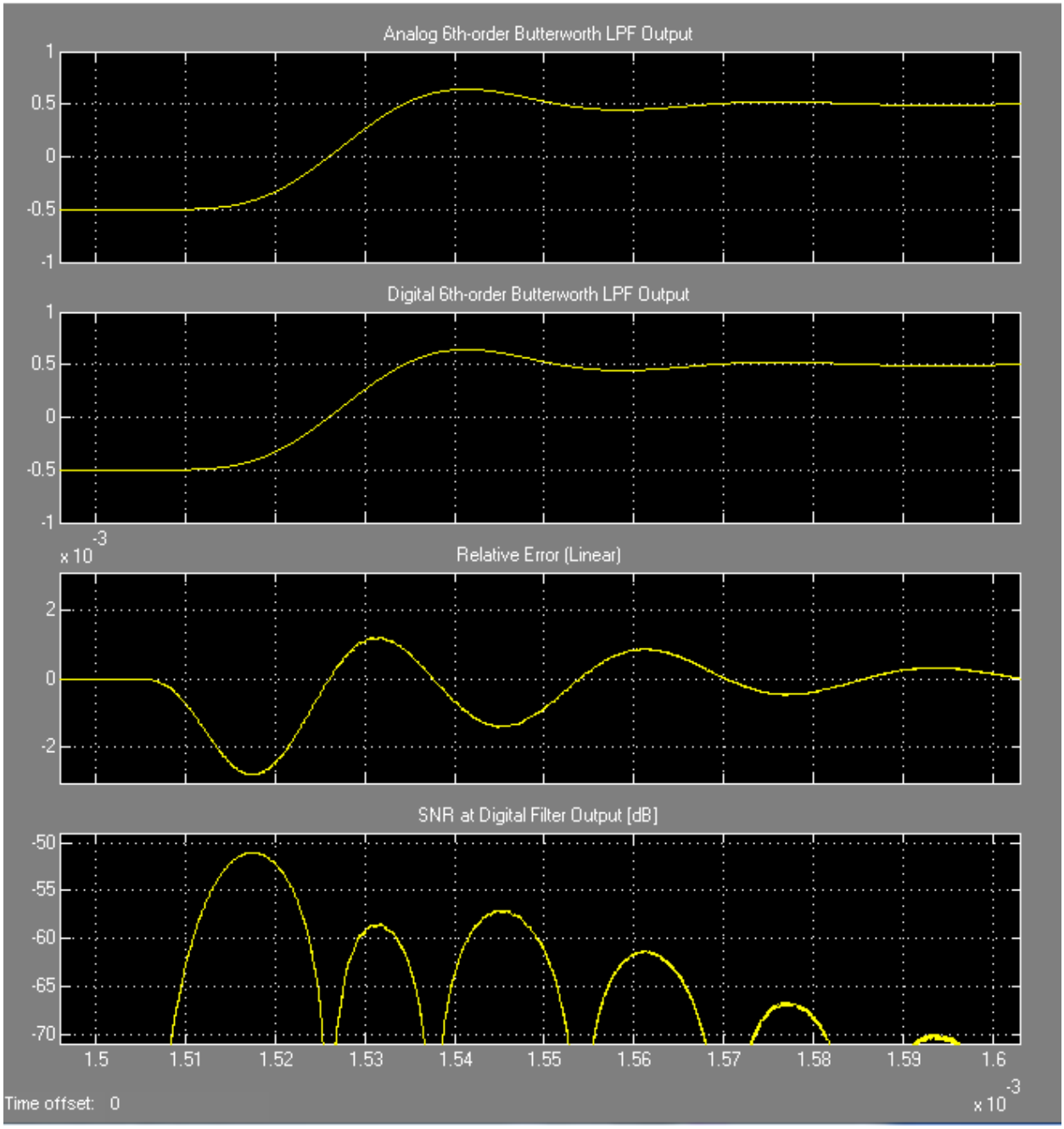

Figure 4- Filter signals and signal-to-noise ratio. Cutoff frequency 30 kHz, input – Step function

The top trace of Figure 5 represents a reference 3.5 KHz sine signal passed through an analog filter with the cutoff frequency 3.5 KHz, 2nd trace from the top is the 3.5 KHz sine digitized and filtered with the digital filter, 3rd trace depicts the ratio of the difference between the outputs of the two filters to the maximum output amplitude of the digital filter, and the bottom trace is the

square of the value of the 3rd trace in dB which represents filter's signal-to-noise ratio caused by sample time mismatch.

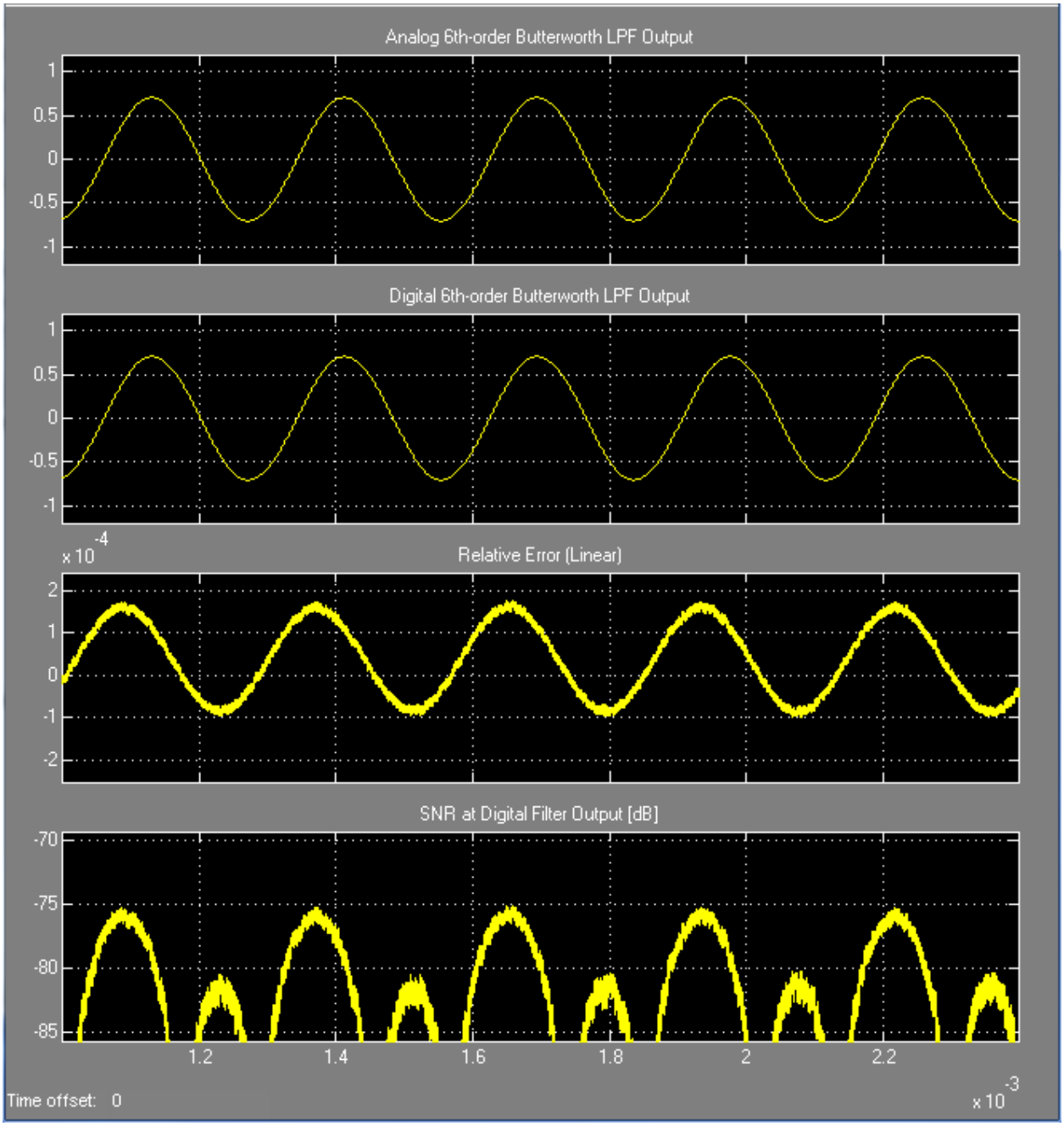

Figure 5- Filter signals and signal-to-noise ratio. Cutoff frequency 3.5 kHz, input – 3.5 kHz sine wave

The top trace of Figure 6 represents a reference step function signal passed through analog filter with the cutoff frequency 3.5 KHz, 2nd trace from the top is the step function digitized and filtered with the digital filter, 3rd trace depicts the ratio of the difference between the outputs of the two filters to the maximum output amplitude of the digital filter, and the bottom trace is the

square of the value of the 3rd trace in dB which represents filter's signal-to-noise ratio caused by sample time mismatch.

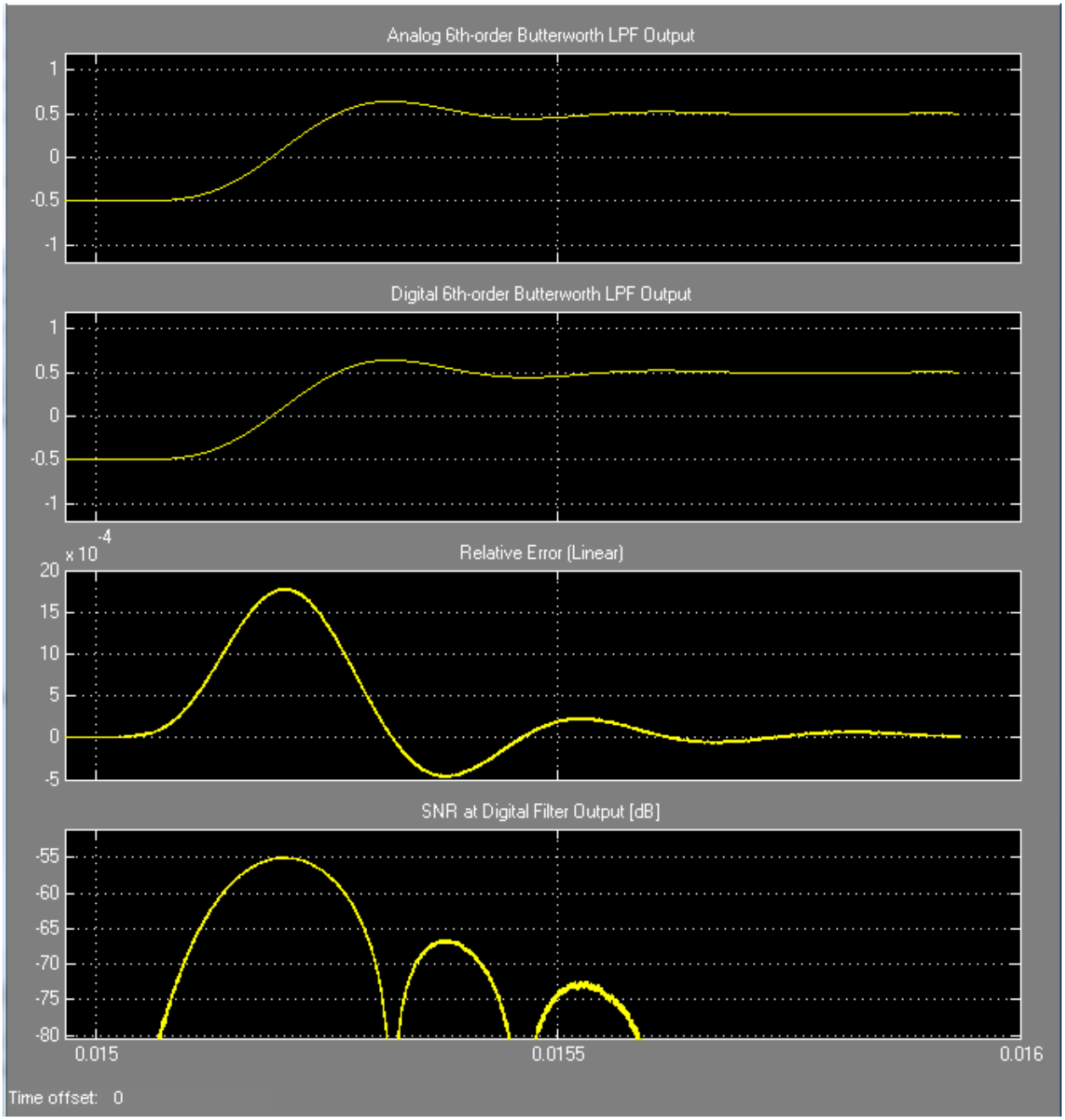

Figure 6- Filter signals and signal-to-noise ratio. Cutoff frequency 3.5 kHz, input – Step function

EXPERIMENTAL RESULTS

The simulation results are in agreement with the values predicted according to [2.4]. The filter was implemented and tested with a 3 kHz cutoff frequency and test results are shown in Figure 7

and Figure 8. On both figures, the top trace shows the input sine waveforms, 1.5 kHz or 3 kHz, and bottom traces show the output filtered signals. The 3 kHz output waveform is attenuated by 3 dB in accordance with the filter's cutoff frequency.

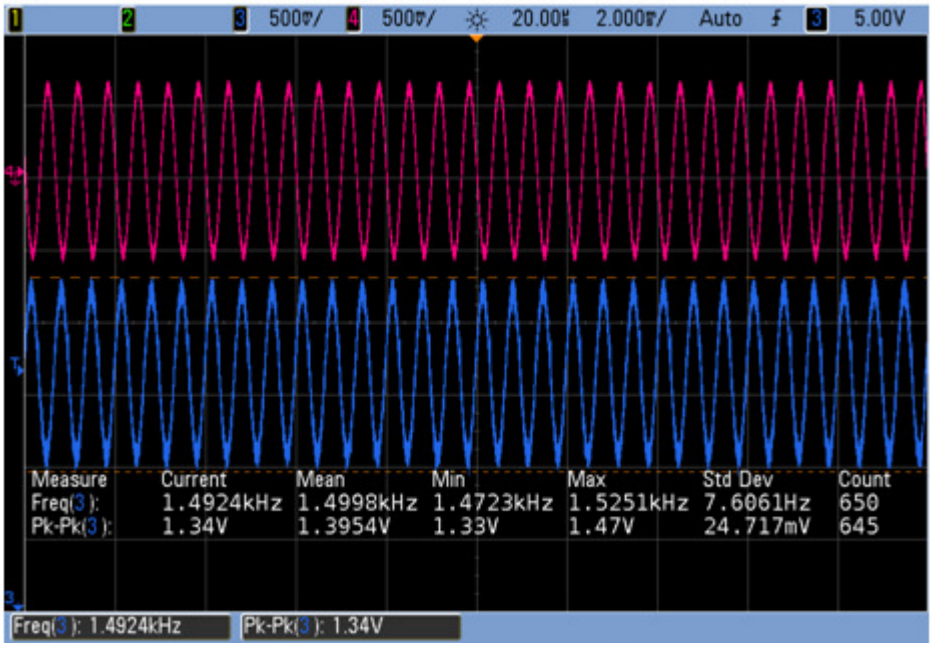

Figure 7- Input and output of the filter implemented on Xilinx Spartan-3 FPGA. Cutoff frequency 3 kHz. Sine wave 1.5 kHz

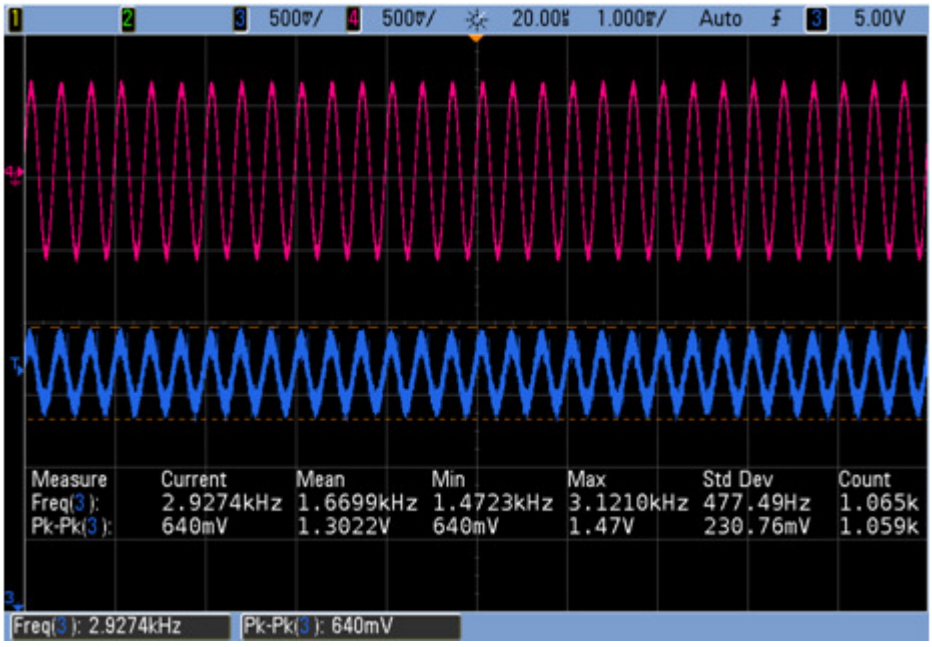

Figure 8- Input and output of the filter implemented on Xilinx Spartan-3 FPGA. Cutoff frequency 3 kHz. Sine wave 3 kHz.

SYSTEM SYNCHRONIZATION

The ability to synchronize the sampling of a unique set of sensor inputs is a key feature of a distributed data acquisition system, providing the flight test engineer with the ability to correlate measurements from different locations on the vehicle. With analog filtering, some systems implemented the sample and hold function using analog components, with data from each being converted by using a multiplexer and single Analog-to-Digital Converter; others dedicated a chain consisting of filtering and an Analog-to-Digital Converter for each input channel and would implement simultaneous sampling by triggering the Analog-to-Digital Converter upon the occurrence of the synchronization event (such as a frame mark).

DSP based filter implementations require continuous sampling to generate the desired output products. When used within a data acquisition system, failure to synchronize the sampling or the filtering operating can result in either a redundant filter output being inserted into the PCM output stream as highlighted in Figure 9, or a completely missed filter sample. While this is not typically noticed when observing real-time data, power spectral density plots will show artifacts that have been created by the mismatch between the DSP filter operation and the PCM output.

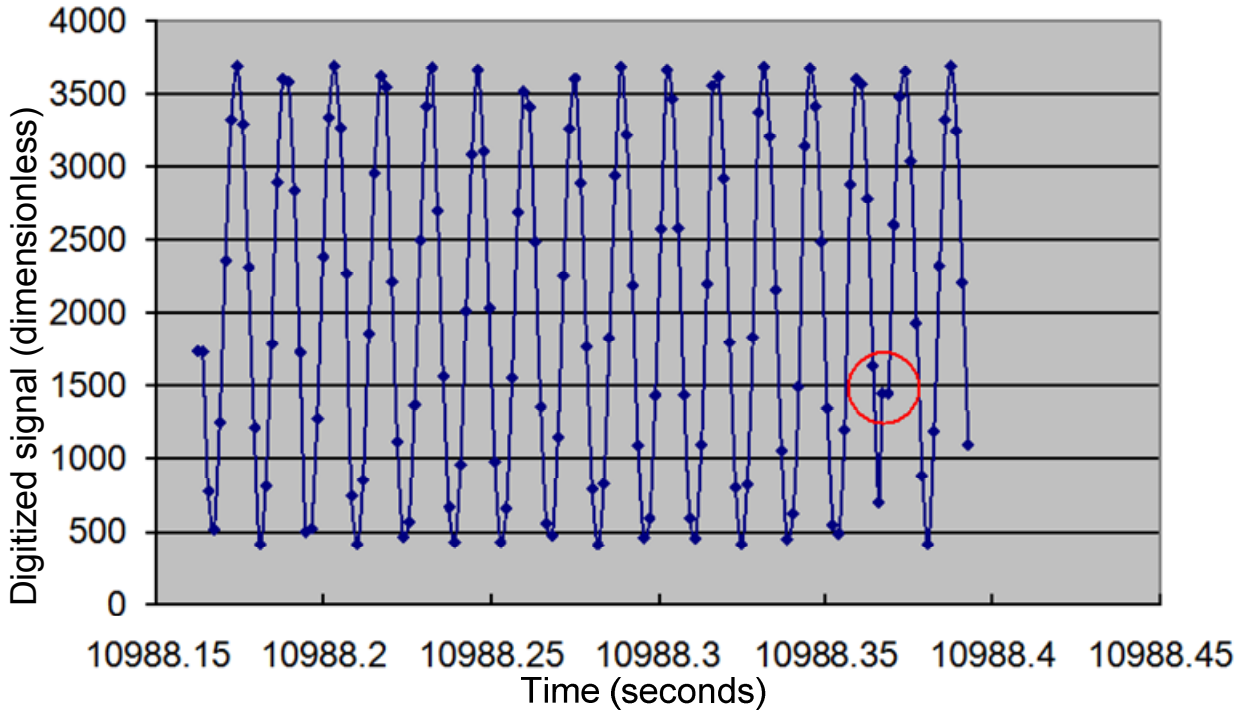

Figure 9- Redundant PCM/DSP Filtering Anomaly Example

The use of a high ratio oversampling interpolator after digital filter allows for the development and implementation of new methods of filtering that can be tightly coupled to the operation of the PCM format. Not only do these methods eliminate the anomalies that occur with the heritage DSP filters, but also provide lower latency and more accurate results.

CONCLUSION

The traditional analog design approach provides simultaneous sampling of the telemetry signals, but requires implementing individual filters with a fixed bandwidth for each channel. Contrary to the analog design, a digital signal processing based approach along with the use of a novel and efficient component base enables a multichannel filter implemented on the same set of DSP basic components which significantly decreases the acquisition system physical dimensions and increases the number of channels. Additionally, any DSP filter is highly configurable since its response is defined by the set of coefficients stored in memory. Finally, sample time mismatch, caused by the limited sampling rate of a digital filter which also prevents the implementation of simultaneous sampling, can be successfully mitigated by augmenting the DSP filter with a digital interpolator described in this paper.

REFERENCES

[1] Telemetry Standard RCC Document 106-07, September 2007.

[2] Sophocles J. Orfanidis, “Introduction to signal processing”, Prentice-Hall, 1996.

[3] Eugene B. Hogenauer, “An economical class of digital filters for decimation and interpolation,” IEEE Transactions on Acoustics, Speech and Signal Processing, pp. 155-162, April 1981.