Title

Direct Sigma Delta Bitstream Processing for High Performance Feedback Control

Authors

Poverelli, Joseph
Brewer, Forrest

Publication Date

2019

DOI

10.1109/ccta.2019.8920720

Peer reviewed

Direct ΣΔ Bitstream Processing for High Performance Feedback Control

Joseph Poverelli and Forrest Brewer

***Abstract*— The design, noise, and performance analysis of a digital controller architecture directly processing ΣΔ bitstreams is described. The use of ΣΔ bitstream representation allows the controller to achieve very low control latency by removing the bitstream to parallel conversion step. The controller has a small footprint and low power dissipation from multiplierless design. Based on integrated state variables, the controller achieves stable implementation of a wide range of controller designs with near continuous time performance. A generic technique for estimating the signal to noise ratio of the controller using an appropriate model of ΣΔ bitstream signal and noise characteristics is outlined. To demonstrate the effectiveness of the ΣΔ controller in a low latency application, a Q controller for an atomic force microscope cantilever was designed and simulated. The ΣΔ controller was able to achieve similar performance to its ideal continuous time counterpart and surpass its conventional discrete equivalent while maintaining high output signal resolution and having a footprint that fits easily into an inexpensive micro-power FPGA.**

I. Introduction

Modern embedded control systems play an important role in the successful deployment of products in industries serving consumer, industrial, transportation, and military uses. Embedded digital control systems are implemented via microprocessor,digital signal processor (DSP), or a field programmable gate array (FPGA) depending on the required performance. Higher performance controllers requiring minimal latency as well as high bandwidth arecommonly constructed in the analog circuit domain. Advances in signal conversion have lead to the widespread use of ΣΔ based analog-to-digital converter (ADC) and digital-to-analog converter (DAC) components. The single-bit comparitor design allows the highest linearity and resolution available for such converters. Unfortunately, use of ΣΔ converters within a conventional DSP framework requires conversion from the bitstream to parallel sampled data representation at much lower sample rates. This conversion introduces substantial latency (on the scale of the sample bandwidth). Similar delays occur in the controller and output converter. One cannot simply increase the sample rate in a conventional z-transform designed shift based digital controller since this causes the poles and zeros of the design to approach unity, thus greatly increasing coefficient sensitivity and bitwidth.[1]

On the other hand, the oversampled bitstream carries the same information as its average value and can be used directly in an appropriate controller design. Such a design requires integration rather than time-shift as the fundamental

The authors are with the Department of Electrical and Computer Engineering, University of California, Santa Barbara, CA, 93106, USA (email: jpoverelli@ucsb.edu, forrest@ece.ucsb.edu)

Fig. 1. ΣΔ embedded controller signal chain

implementation operator. Time-shift operations on bitstream data are extremely sensitive to correlated bitstream representation noise which is always stronger that the represented signal strength. Such representation noise is inherent in any binary oversampled data stream [2]. The δ-operator is a scaled integration which low-pass filters the noise while maintaining integrated state of the signal. A sequence of such operators can construct an arbitrary-order controller which mimics ideal (continuous-time) controller state dynamics [1]. This is achieved since every component is operating at the over-sampling rate inherent in the bitstream. This feature allows this style of controller to potentially replace analog circuit-based control in many high-performance applications with benefits in cost, lack of component drift, and the ability to implement much higher order controllers with practical components.

There are a surprisingly large family of commercially available components with ΣΔ bitstream interfaces, ranging from high-resolution ADC and DAC designs and well as MEMS-based transducers (audio and higher performance acclerometer/gyro), magnetic field sensors and others. Additionally, since the oversampled bitstream can drive an H-bridge directly or through simple logic, high power actuation is also easily achieved. In systems based on these components, controller implementation and computation are based on conventional DSP techniques after sinc-filter conversion of the bitstream. In contrast, the techniques described in this paper allow for high-order controller implementation in micro-power FPGA components without integrated DSP functionality while retaining analog circuit-level latency and bandwidth. Such a single-input-single-output controller is shown in Figure 1.

This paper will propose a computation strategy for implementing linear time invariant control algorithms in fixed point hardware that directly process ΣΔ encoded bitstreams. First, the proper utilization of the signal information in an encoded ΣΔ bitstream representation and its power spectral density will be discussed. Next, a description of the filter hardware architecture will be introduced and how it can implement a controller transfer function. Lastly, a high frequency closed loop Q controller for an atomic force microscope cantilever will be implemented with ΣΔ based control to demonstrate the low latency, small footprint, and

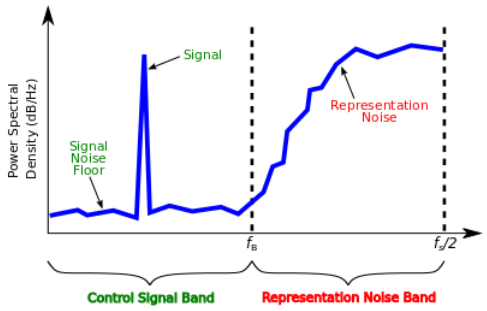

Fig. 2. Anatomy of PDM Signal PSD

low power advantages of the proposed implementation. It will be shown that the closed loop ΣΔ controller can achieve comparable continuous time controller performance while being implemented in relatively inexpensive and low power off the shelf components.

II. Direct ΣΔ Bitstream Processing

There are several difficulties to directly process ΣΔ bitstreams. The first is that the immediate value of the bitstream signal cannot be determined from the immediate state of the stream, but instead requires a history of tens to hundreds of sequential bits. Secondly, the bitstream has a significant amount of noise that is inherent in the binary representation. This noise represents a signal energy that is at least an order of magnitude more that than of the encoded signal. The final difficulty is that of correlation; conventional architectures with feedback can reintroduce representation noise in the signal band and drown out the relevant information.

To characterize the quality and information content of a bitstream signal, the power spectral density (PSD) is used. The PSD reveals characteristics about a bitstream such as the signal it is encoding, the quality of the signal, and the frequency distribution of the noise. The anatomy of a power spectral density of a pulse density modulated signal can be seen in Figure 2.

For control applications, the system dynamics are to be contained in frequencies below the Nyquist bandwidth of f_B. The region below f_B contains a low noise floor bounded by the noise floor of the input converter or the bit-width of the previous controller. In this region, the signal to noise ratio (SNR), the ratio between the signal power σ_x^2 and the noise variance σ_n^2 in the signal band from DC to f_B, is very high. A large SNR can also be translated as a large effective number of bits (ENOB) in a parallel representation. The relationship between the SNR and ENOB for an ideal Nyquist rate data converter is

$$ENOB = \frac{SNR - 1.76}{6.02} \tag{1}$$

derived in [2]. For frequencies from f_B to $f_s/2$, the noise floor increases dramatically and saturates the power spectrum. The concentration of noise power in the higher frequencies is a hallmark of pulse density modulated (PDM) signals and is created by the dithering nature of the bitstream signal. The quality of shaping the representation noise into high frequencies allows PDM bitstreams to achieve very high quality encoding. Implementing successful bitstream processing circuits, requires circuits that take advantage of the low frequency, high SNR baseband without aliasing in the high frequency representation noise.

To construct a frequency dependent model of the PDM bitstream output of a ΣΔ modulator for use in PSD estimation, a linear model of the modulator dynamics can be obtained by making a few assumptions [2]. The assumptions are:

1) The quantization error, e_q, is uncorrelated with itself and the modulator input.
2) The quantization error is a stationary, random process.
3) The quantization error has a uniform probability density function over its entire range.

From these assumptions the following frequency domain model can be derived

$$q_{\Sigma\Delta}(f) = u(f) + NTF_{\Sigma\Delta}(f)\, e_q(f) \tag{2}$$

where f is frequency, $q_{\Sigma\Delta}$ is the bitstream output of the ΣΔ modulator, u is the modulator input, e_q is the additive quantization noise input, and $NTF_{\Sigma\Delta}$ is the noise transfer function of the modulator. It is of note here, that for baseband modulators, the input effectively sees zero phase delay below f_B and is thus abstracted out of the above linear model. The frequency domain model of the bitstream above becomes the basis for the design and analysis of the controller filter structure that is to follow.

.

III. The ΣΔ Controller Architecture

Direct processing of ΣΔ bitstreams requires a specialized filtering architecture that performs the correct signal processing on the compact information representation while simultaneously mitigating its inherent representation noise. The original proposal for a filter architecture that processes ΣΔ bitstreams was introduced by Johns and Lewis in [3]. The concept of using a ΣΔ filter as a controller has been presented by Wu and Goodall in [4] with other examples in [5], [6], [7], [8], [9]. The ΣΔ Controller (SDC) architecture presented below and adapted from previous works is a prime candidate for ΣΔ bitstream filtering and implementing transfer functions.

As can be seen in Figure 3, the SDC architecture is based upon the standard direct-form II transposed transfer function implementation but with some important modifications. First, a digital ΣΔ modulator is placed in the loop and encodes the output of the filter into a bitstream. Having both a 1-bit wide feedforward and feedback path allows one to use multiplexers to implement the filter coefficient gains eliminating digital multipliers. In this case, positive and negative values of the coefficients are selected by the binary value of the bitstream. Bit shifts are used to implement scaling gains at the output of the discrete integrators eliminating the need for additional hardware multipliers. Thus, the entirety of the SDC architecture can be constructed from

just multiplexers, adders, registers, and bitshift operators including the circuitry associated with the output digital ΣΔ modulator. This implementation's logical footprint is very small compared to conventional DSP implementations. In contrast, bit serial constructions using serialized multipliers have an equivalently small footprint, however results are only fed forward at a fraction of the oversampled clock speed. This substantially increases the latency of the controller.

A. Design via Emulation

The SDC architecture provides a computation strategy for implementing transfer functions with bitstream inputs and outputs. Not only does this eliminate the need for any bitstream to bit parallel circuity, but bitstreams from data converters or transducers can be directly connected to the FPGA that implements the filter structure. Any linear control law (e.g lead/lag, PID, LQR, H_∞, etc.) can be transformed into an linear time invariant (LTI) transfer function or a collection/matrix of transfer functions. Once a controller has been synthesized using a continuous time model and technique, the controller state space representation can be transformed via discrete emulation, a popular method for the design of discrete time controllers.

Given a continuous time state space description of a controller

$$\begin{aligned} \dot{x} &= Ax + Bu \\ y &= Cx + Du \end{aligned} \tag{3}$$

where $x \in \mathbb{R}^n$, $y \in \mathbb{R}^m$, $u \in \mathbb{R}^k$, $A \in \mathbb{R}^{n\times n}$,$B \in \mathbb{R}^{n\times k}$,$C \in \mathbb{R}^{m\times n}$,$D \in \mathbb{R}^{m\times k}$, it can then be converted to the δ domain

$$\begin{aligned} \delta x &= A_\delta x + B_\delta u \\ y &= C_\delta x + D_\delta u \end{aligned} \tag{4}$$

using the relations

$$A_\delta = \frac{e^{A\Delta} - I}{\Delta} \tag{5}$$

$$B_\delta = \frac{1}{\Delta}\int_0^\Delta e^{A(t-\tau)} B u(\tau)\, d\tau \tag{6}$$

$$C_\delta = C \tag{7}$$

$$D_\delta = D \tag{8}$$

where Δ is the oversample period. The resulting δ domain model can then realize the transfer function

$$H(\delta) = C_\delta(\delta I - A_\delta)B_\delta + D_\delta \tag{9}$$

$$= \frac{\beta_{\delta 0} + \beta_{\delta 1}\delta^{-1} + \cdots + \beta_{\delta n-1}\delta^{-(n-1)} + \beta_{\delta n}\delta^{-n}}{1 + \alpha_{\delta 1}\delta^{-1} + \cdots + \alpha_{\delta n-1}\delta^{-(n-1)} + \alpha_{\delta n}\delta^{-n}} \tag{10}$$

In the case of MIMO controllers, each transfer function in the controller transfer function matrix can be individually realized with the SDC architecture. Consider for instance the 2x2 multiple-input-multiple-output continuous time controller

$$K(s) = \begin{bmatrix} K_{11}(s) & K_{12}(s) \\ K_{21}(s) & K_{22}(s) \end{bmatrix}$$

synthesized from a controller design technique such as H_∞ or LQG. The controller can be converted to the δ domain and implemented and employed in an embedded control system in a manner such as depicted in figure 4.

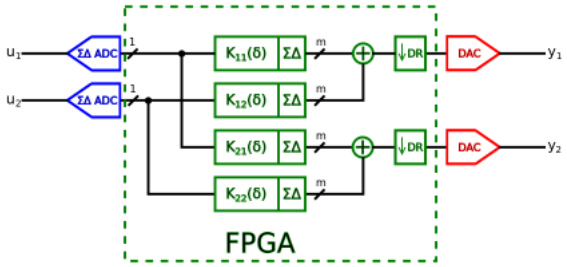

Fig. 4. 2x2 MIMO ΣΔ Controller Implementation

As shown in the diagram, the multibit output of the individual controllers (the output of the filter before the digital ΣΔ modulator) can be combined with a multibit adder to form a composite controller output. The controller output can then be subsampled by a decimation rate of DR and fed to a parallel DAC running at a much lower rate. The multibit output is safe from aliasing the bitstream representation noise for strictly proper transfer functions (the numerator order is less than the denominator order i.e. $\beta_{\delta 0} = 0$) because the input and feedback bitstreams are filtered through the successive δ-operators of the filter.

B. Noise Performance

This section will present an analysis of the noise propagation in the SDC architecture. A major difficulty in processing ΣΔ bitstreams is mitigating the representation noise so that it does not alias down into the signal band and drown out the information present there. The first step to mitigating the noise inside the filter structure is by using a reasonable coefficient scaling policy. Scaling the internal integrators of the filter is important in order to not bury the signal content in the noise floor of the successive stages of the filter structure.

Let

$$f_i(\gamma) = \frac{x_i(\gamma)}{u(\gamma)} \tag{11}$$

be the delta transform transfer function from the filter input u to the ith state variable x_i. Using the following p-norm definition for the complex valued γ-operator

$$\|H(\gamma)\|_p = \left[\frac{1}{2\pi}\int_{-\pi}^{\pi} |H\left(\frac{e^{j\omega}-1}{\Delta}\right)|^p d\omega\right]_{\gamma=\frac{e^{j\omega}-1}{\Delta}}^{1/p} \tag{12}$$

the filter nodes can be unity scaled by the following relation

$$\begin{aligned} ||f(\gamma)||_\infty &= T_s^{-1}||T_0^{-1}(\gamma I - A_\delta)^{-1} B_\delta||_\infty \\ &= \begin{bmatrix} 1 & \cdots & 1 \end{bmatrix}^T \end{aligned} \tag{13}$$

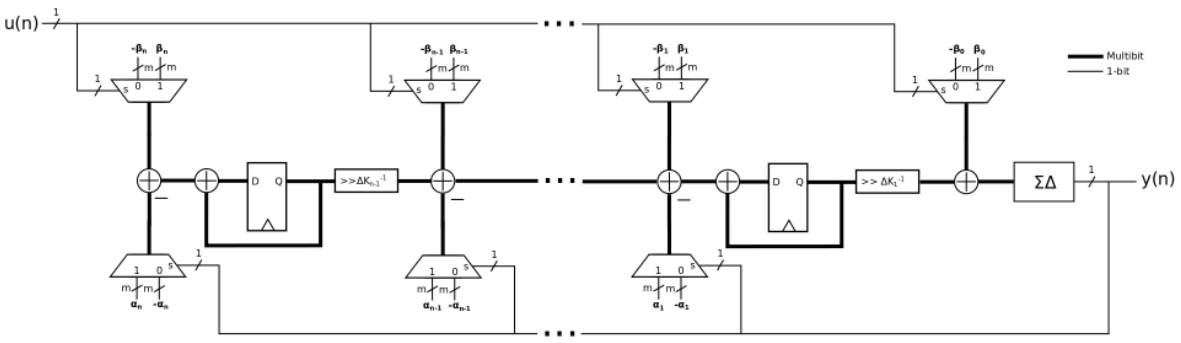

Fig. 3. ΣΔ Controller Register Transfer Level Architecture

The ∞-norm is used for scaling to ensure no overflow in the filter integrator registers. Here, T_s and T_0 are the scaling and transformation matrices described in [10] respectively. Once the scaling is determined, the scaling coefficients can be adjusted as powers of two using the relation

$$\tilde{k}_i = \frac{2^{\lfloor log_2 \Delta \cdot k_i^{-1} \rfloor}}{\Delta} \tag{14}$$

where Δ is the oversample period. The filter coefficients can then be modified via $\tilde{\beta}_{\delta i} = \beta_{\delta i}\tilde{k}_1\tilde{k}_2 \ldots \tilde{k}_n$ and $\tilde{\alpha}_{\delta i} = \alpha_{\delta i}\tilde{k}_1\tilde{k}_2 \ldots \tilde{k}_n$. This will allow the filter scaling gains to be implemented as bitshifts rather than hardware multiplies.

Noise from the ΣΔ filter comes from three sources. All ΣΔ modulator noise sources take on the model described in the previous section. The three noise sources are as follows:

1) The representation noise power from the input modulator to the output of the filter can be determined by

$$\sigma^2_{\Sigma\Delta_1} = \eta_{\Sigma\Delta_1} \int_{-f_B}^{f_B} |H_{\Sigma\Delta}(f) NTF_{\Sigma\Delta_1}(f)|^2 \, df \tag{15}$$

where $\eta_{\Sigma\Delta_1}$is the power spectral density of the additive modulator quantization noise, $NTF_{\Sigma\Delta_1}(f)$ is the noise transfer function of the input modulator, and $H_{\Sigma\Delta}(f)$ is the transfer function of the total filter from input to output.

2) The representation noise power from the output modulator to the output of the filter can be determined by

$$\sigma^2_{\Sigma\Delta_2} = \eta_{\Sigma\Delta_2} \int_{-f_B}^{f_B} |F_{\Sigma\Delta}(f) NTF_{\Sigma\Delta_2}(f)|^2 \, df \tag{16}$$

where $\eta_{\Sigma\Delta_2}$is the power spectral density of the additive modulator quantization noise, $NTF_{\Sigma\Delta_2}(f)$ is the noise transfer function of the digital output modulator, and $F_{\Sigma\Delta}(f)$ is the transfer function from the additive noise input to the filter output.

3) The rounding noise after each of the internal scaling gains k_i in the filter. Let

$$F_{k_i}(\gamma) = \frac{y(\gamma)}{e_{k_i}(\gamma)} \tag{17}$$

be the transfer function from the additive noise input of the scaling gain rounding to the output of the filter. The output noise power contribution from each scaling coefficient rounding is determined by

$$\sigma^2_{k_i} = \eta_{k_i} \int_{-f_B}^{f_B} |F_{k_i}(f)|^2 \, df \tag{18}$$

where $\eta_{k_i}$is the power spectral density of the ith additive rounding noise. Here, rounding is assumed and that the quantization error e_{k_i} at each scaling gain is an uncorrelated random Gaussian stationary processes.

The total estimated noise power that can be expected at the ΣΔ filter output can be determined by

$$\sigma^2_{total} = \sigma^2_{\Sigma\Delta_1} + \sigma^2_{\Sigma\Delta_2} + \sum_{i=1}^{N} \sigma^2_{k_i} \tag{19}$$

With the use of 2nd or higher order ΣΔ modulator bitstream encodings, the additional noise exhibited by the filter is minimal when compared to the product quantization error inherent in conventional DSP solutions. The total noise power σ^2_{total} can be used as a metric for the controller bitwidth resolution.

Using the design by emulation technique, a continuous time controller can be converted to the δ domain where the corresponding coefficients are found by mapping into the SDC direct-form II transposed structure. The noise analysis above can then be used to gain an accurate estimation of the noise propagation to the output of the filter within the controller bandwidth from DC to f_B based on the number of bits ascribed to the coefficient and integrator bitwidths.

IV. Motivating Example

To demonstrate the effectiveness of ΣΔ based control in a high bandwidth, low-latency application, an AFM cantilever controller was simulated in the Matlab/Simulink environment. The Q-Controller presented in [11] describes an AFM cantilever with a differential sensing interface. The goal of the described Q controller design is to reduce the Q-factor of the first resonant mode of the cantilever tip. This allows higher performance and greater accuracy for the cantilever operation in tapping mode, especially for actuation signals close to resonance, of this high performance design.

A ΣΔ version of the closed loop Q controller was implemented as shown in Figure 5. In this design, a ΣΔ modulator

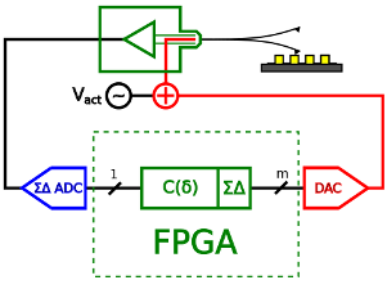

Fig. 5. AFM ΣΔ Q-Controller Implementation

is assumed for the ADC in order to provide a bitstream input into the FPGA. A bit parallel output of the filter was routed from the final state integrator and directed into an ideal 16 bit parallel DAC running at a clock rate of 1MHz which is about ten times the controller Nyquist rate. The output interface is chosen in order to not drive the higher order resonance modes of the cantilever with the representation noise of a bitstream.

The estimated transfer function estimate from the actuation input Voltage to the output Voltage of the sense amplifier circuit is given as

$$G(s) = \frac{-0.73s^3 + 1.28 \cdot 10^5 s^2 - 6.57 \cdot 10^{10} s + 1.45 \cdot 10^{16}}{s^3 + 1.12 \cdot 10^6 s^2 + 9.52 \cdot 10^{10} s + 1.05 \cdot 10^{17}}$$

and the PPF controller is given as

$$C(s) = \frac{k_c \omega_c^2}{s^2 + 2\zeta_c \omega_c s + \omega_c^2}$$

where $k_c = 0.959$, $\zeta_c = 0.178$, and $\omega_c = 48.547$ kHz. Simulations were performed in the Matlab/Simulink environment with the continuous time controller, a conventional discrete controller, and an SDC. It is important to note that the conventional discrete controller is implemented by first decimating an input ΣΔ modulator running at 30.72 MHz with a 3rd order Sinc filter at a rate of 25 and 50. The discrete controller as well as the output DAC then runs at the decimated clock rate. The discrete controller was derived from the continuous controller using the bilinear transform at the appropriate sampling period. A plot of the closed loop magnitude response performance can be seen in Figure 6 with the SDC running at a clock frequency of 30.72 MHz. The closed loop magnitude response of the continuous, discrete, and ΣΔ Q controllers were determined using the ratio of Welch's power spectral density estimates from the actuation Voltage input and sense Voltage output of the Simulink model.

Using the controller parameters seen in Table I, ΣΔ controllers running at three different clock frequencies were simulated with their corresponding closed loop magnitude response plotted in Figure 7. Table II lists the relevant design information associated with each controller. While there is no converter or sensor noise modeled in the simulation, the SDC control band SNR can still be estimated and measured from

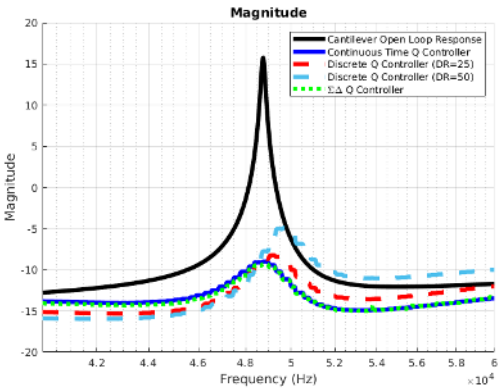

Fig. 6. AFM Continuous, Discrete, and ΣΔ Q-Controller Magnitude Response

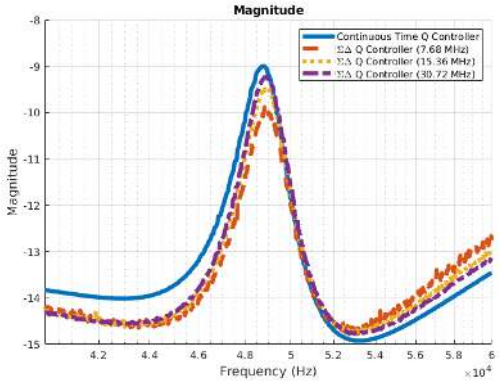

Fig. 7. AFM ΣΔ Q-Controller Magnitude Response

the fixed point implementation. The SNR numbers assume a full scale sinusoidal input while integrating the noise PSD over the entire control band from DC to f_B. It is also of note to specify that the ideal Q controller coefficients were chosen to obtain a specific closed loop Q value. The SDC control loop magnitude response varies slightly from the ideal model due to the discretization process at different sampling frequencies.

TABLE I
ΣΔ Q CONTROLLER PARAMETERS

Parameter	Controller 1	Controller 2	Controller 3
f_B	60 kHz	60 kHz	60 kHz
f_s	7.68 MHz	15.36 MHz	30.72 MHz
Controller Latency	1.3 μs (5 clock cycles)	326 ns (5 clock cycles)	163 ns (5 clock cycles)
Estimated SDC SNR	61dB (9.84 ENOB)	76.5dB (12.4 ENOB)	91.5dB (14.9 ENOB)

To estimate resource utilization and power dissipation in an FPGA, the three ΣΔ controllers were mapped into the low power ICE40LP8K-CM81 FPGA from Lattice Semiconductor. The Lattice iCEcube2 software was used to synthesize, place, and route the hardware description of the controllers to determine the number of flip flops and look up tables

TABLE II
ΣΔ Q CONTROLLER FIXED POINT PARAMETERS

Oversample Frequency	Coefficient Values	Coefficient Bitwidths (integer, fractional)	Integrator Bitwidths	Scaling Bitshifts	Measured SDC Control Band SNR	Power Dissipation	LUT/FF
$7.68MHz$	$\tilde{\alpha}_1 = 32.728317e3$ $\tilde{\alpha}_2 = 52.551870e3$ $\beta_0 = 0$ $\tilde{\beta}_1 = 1.578625e3$ $\tilde{\beta}_2 = 50.397243e3$	$BW_{\beta_0} = (1, 12)$ $BW_{\alpha_1,\beta_1} = (17, 0)$ $BW_{\alpha_2,\beta_2} = (18, 0)$	$BW_{x_1} = 24$ $BW_{x_2} = 24$	$BS_{\Delta k_1^{-1}} = 21$ $BS_{\Delta k_2^{-1}} = 4$	$61.47dB$	$1.94mW$	178/76
$15.36MHz$	$\tilde{\alpha}_1 = 31.196118e3$ $\tilde{\alpha}_2 = 52.742832e3$ $\beta_0 = 0$ $\tilde{\beta}_1 = 791.249610$ $\tilde{\beta}_2 = 50.580376e3$	$BW_{\beta_0} = (1, 14)$ $BW_{\alpha_1,\beta_1} = (17, 0)$ $BW_{\alpha_2,\beta_2} = (18, 0)$	$BW_{x_1} = 25$ $BW_{x_2} = 25$	$BS_{\Delta k_1^{-1}} = 22$ $BS_{\Delta k_2^{-1}} = 5$	$71.16dB$	$3.12mW$	195/82
$30.72MHz$	$\tilde{\alpha}_1 = 30.425712e3$ $\tilde{\alpha}_2 = 52.837353e3$ $\beta_0 = 0$ $\tilde{\beta}_1 = 396.100583$ $\tilde{\beta}_2 = 50.671022e3$	$BW_{\beta_0} = (1, 16)$ $BW_{\alpha_1,\beta_1} = (17, 0)$ $BW_{\alpha_2,\beta_2} = (18, 0)$	$BW_{x_1} = 26$ $BW_{x_2} = 26$	$BS_{\Delta k_1^{-1}} = 23$ $BS_{\Delta k_2^{-1}} = 6$	$86.57dB$	$4.36mW$	214/88

(LUT) used as well as estimate the power dissipation in each scenario. An ICE40LP FPGA from Lattice was chosen not only due to its low operating power, but also because the FPGA does not contain any built in DSP hardware accelerators. While the ICE40LP family of FPGAs might typically be used for glue logic or interfacing, the low complexity and multiplierless design of the ΣΔ controller allows normally resource hungry controller implementations to be mapped into low cost, low power, low resource FPGAs. In fact, the ICE40LP8K FPGA can fit ten to twelve second order controllers based on the resource utilization numbers given above. One could also implement a higher order controller to dampen higher frequency resonance peaks in the Q controller which would further increase an AFM's scanning speed.

The previous simulations demonstrate that the ΣΔ Q controller is able to emulate a continuous time controller transfer function while directly processing the bitstream from the front end modulator. The entire ΣΔ Q controller has a signal chain latency of only five clock cycles which is a minute fraction of the multiple decimated clock cycles required for the conventional discrete case. Consequently, the conventional discrete controller closed loop magnitude response deviates substantially from that of the ideal continuous controller. When increasing the oversample clock frequency, the ΣΔ Q controller response holds closer to the continuous case but at a cost. In terms of power and footprint, doubling the clock frequency increases the power dissipation by a factor of about 50% in the controller while the LUT/FF utilization increases only modestly. Faster clock speeds also allow for higher controller SNRs which is demonstrated in the estimated and actual output noise powers. The estimated output noise power from Table I matches quite well to the actually noise power listed in Table II which validates the accuracy of the noise analysis.

V. CONCLUSIONS

In this paper, the design and noise analysis of a digital controller architecture that processes ΣΔ bitstreams is presented. The multiplierless and scaled integrator state based design allows the ΣΔ controller to implement near continuous time performance in a relatively compact hardware footprint. The inherent representation noise of ΣΔ bitstreams was discussed and a noise analysis was proposed to estimate the total output noise propagation in the filter structure. Using an AFM cantilever Q controller as a motivating example, simulations of closed loop performance demonstrated that the ΣΔ controller could emulate its continuous time counterpart better than that of the conventional discrete controller running at a decimated clock rate. By increasing the oversample clock rate, it was shown that the controller SNR and ability to emulate the continuous time controller improved but at a cost of increased power dissipation and a moderate increase in FPGA resource utilization. Despite the trade off, the ΣΔ controller is able to fit within low cost FPGAs running on micro-watt power.

REFERENCES

[1] G. C. Goodwin, R. H. Middleton, and H. V. Poor, "High-speed digital signal processing and control," *Proceedings of the IEEE*, vol. 80, no. 2, pp. 240–259, Feb 1992.

[2] R. Schreier and G. Temes, *Understanding Delta-Sigma Data Converters*. Wiley, 2004.

[3] D. Johns and D. Lewis, "Sigma-delta based iir filters," in *Circuits and Systems, 1991., Proceedings of the 34th Midwest Symposium on*, May 1991, pp. 210–213 vol.1.

[4] X. Wu and R. M. Goodall, "One-bit processing for digital control," *IEE Proceedings - Control Theory and Applications*, vol. 152, no. 4, pp. 403–410, July 2005.

[5] X. Bai, X. Wu, and S. Xiao, "A design of one bit processing based model predict control," in *Proceedings of the 31st Chinese Control Conference*, July 2012, pp. 5910–5914.

[6] D. Al-Makhles, A. Swain, and N. Patel, "Delta-sigma based bitstream controller for a d.c. motor," in *TENCON 2012 IEEE Region 10 Conference*, Nov 2012, pp. 1–5.

[7] D. J. Almakhles, A. K. Swain, and N. D. Patel, "Stability and performance analysis of bit-stream-based feedback control systems," *IEEE Transactions on Industrial Electronics*, vol. 62, no. 7, pp. 4319–4327, July 2015.

[8] A. Swain, D. Almakhles, Y. Hou, N. Patel, and U. Madawala, "A sigma-delta modulator based pi controller for bidirectional inductive power transfer systems," in *2016 IEEE 2nd Annual Southern Power Electronics Conference (SPEC)*, Dec 2016, pp. 1–6.

[9] D. Almakhles, N. Pyle, H. Mehrabi, A. Swain, and A. P. Hu, "Single-bit modulator based controller for capacitive power transfer system," in *2016 IEEE 2nd Annual Southern Power Electronics Conference (SPEC)*, Dec 2016, pp. 1–6.

[10] N. Wong and T.-S. Ng, "A generalized direct-form delta operator-based iir filter with minimum noise gain and sensitivity," *IEEE Transactions on Circuits and Systems II: Analog and Digital Signal Processing*, vol. 48, no. 4, pp. 425–431, April 2001.

[11] M. B. Coskun, H. Alemansour, A. G. Fowler, M. Maroufi, and S. O. R. Moheimani, "q control of an active afm cantilever with differential sensing configuration," *IEEE Transactions on Control Systems Technology*, pp. 1–8, 2018.