Optimizing the System Design of a Differential 3rd Order Delta-Sigma (ΔΣ) Modulator

M.Sc(s) Alex Julian Mantilla Rios

Modeling and Simulation 1

Universidad Industrial de Santander - alexmantilla97@gmail.com

***Abstract*—This project report presents the system-level design and optimization of a third-order delta-sigma (ΔΣ) modulator using Simulink. With the increasing demand for high-resolution, energy-efficient analog-to-digital converters (ADCs) in Internet of Things (IoT) devices, this study focuses on using optimization algorithms to improve the performance of ΔΣ modulators. By employing advanced optimization algorithms, the study aims to improve modulator performance while mitigating common challenges such as low-voltage saturation and quantization errors. The report details the theoretical framework, simulation results, and practical considerations, demonstrating a significant improvement in modulator performance up to an SNDR of 102.3 dB.**

***keywords*—Delta-Sigma Modulator, Optimization Algorithms, Energy-Efficient Design, Signal Processing, High-Resolution ADC**

1. Introduction

In recent years, the increasing demand for interconnected systems within the ecosystem of smart cities has led to a proliferation of Internet of Things (IoT) devices. These devices gather data from the environment through sensors and process this information to provide valuable insights for decision-making. However, a critical challenge in the design of these devices is the limited availability of energy, as many of these systems are battery or solar-powered. In this context, energy-efficient solutions are essential to ensure the longevity and sustainability of these systems.

One of the fundamental components of IoT devices is the analog-to-digital converter (ADC). In essence, ADCs are the gateway between the physical and digital worlds for signal processing. From the environment, analog signals are measured through sensors, collecting information for future decision-making. However, this information needs conversion to a digital format to be processed by the core. This data conversion process is crucial for many applications, including telecommunications, instrumentation, and more. As a result, high-performance ADCs are an area of intense research and innovation.

There are many ways to implement data conversion, i.e., different types of ADCs that differ in the operations or processes executed to convert analog information (e.g., a voltage measurement of 0.87 V) to digital data (e.g., the value 010011 for 6 bits resolution). These operations can be complex and different, but the types of ADCs are classified into two groups; Nyquist converters and oversampling converters. As is well known, the Nyquist theorem states that for a signal with a specific bandwidth, it is necessary to sample it at least twice its highest frequency component (f_B) to capture all the information. Nyquist ADCs follow this principle, sampling the input signal at a rate equal to or higher than twice its bandwidth ($f_S >= 2f_B$). Oversampling ADCs, on the other hand, sample the input signal at a much higher rate than its bandwidth, which allows for a more straightforward implementation and a higher resolution.

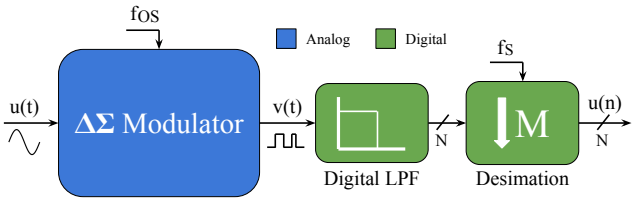


Figure 1. *Delta-Sigma ADC general structure*

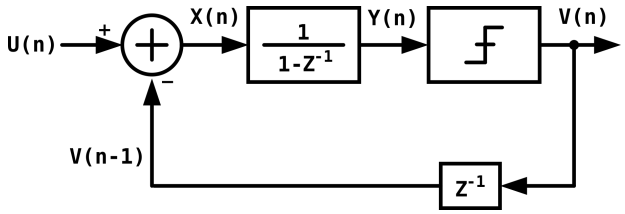
Among oversampling ADCs, one of the most widely used architectures is the Delta-Sigma ADC. Delta-Sigma is a popular choice for its ability to achieve high resolution and robustness against circuit imperfections with minimal power consumption [6], desired characteristics in applications such as biomedical, IoT, and HQ audio. However, they also present challenges such as stability issues, clock jitter sensitivity, and large silicon area. These converters are described in Fig 1, containing the Delta-Sigma Modulator and the Digital Filtering. The first modulates the mean value of a square signal v(t) with the value sampled at the input u(t), while the digital Low Pass Filter (LPF) processes this squared signal as a stream of bits to obtain a fully digital N-bit signal. However, the stream also contains a large amount of noise that is *shaped* by the modulator. Therefore, a decimation filter is required to reduce the sampling rate and remove the noise from the output.

The report is organized as follows: Section 2 introduces the concept of delta-sigma modulation, explaining the operating principles and key characteristics of the modulator. Section 3 details the design of the loop filter, including various implementation techniques and their impact on noise shaping and stability. Section 4 covers the application of optimization algorithms, describing the methodology and simulation results obtained using Simulink. Section 5 examines the impact of circuit non-idealities on the modulator's performance and provides strategies for overcoming these challenges. Finally, in Section 6, we provide a summary of our work, highlighting key findings.

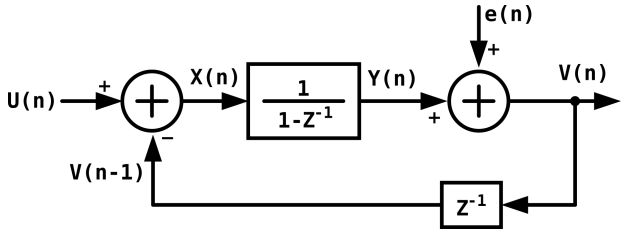
2. Delta-Sigma Modulation

In the previous section, the modulator is mentioned as a black box, a system that converts the amplitude of the input signal u(t) into the average value of a square signal v(t). This section presents how this modulation is implemented and what trade-offs are associated with it. First, it is necessary to show one of the most straightforward systems of this modulation in Figure 2 a). This system comprises four blocks: an adder, an integrator, a quantizer (for this simple case, a 1-bit quantizer), and a delay of 1 sample. Note that this diagram denotes the output signal of the modulator as v(n), while in Figure 1 v(t). This notation is taken based on the time discretization of the system as a function of the sampling frequency f_{OS}.

To understand how the system operates, it is necessary to



(a) *Using the 1-bit quantizer symbol*



(b) *Using a linear model for the quantizer.*

Figure 2. *First Order ΔΣ Modulator block diagram.*

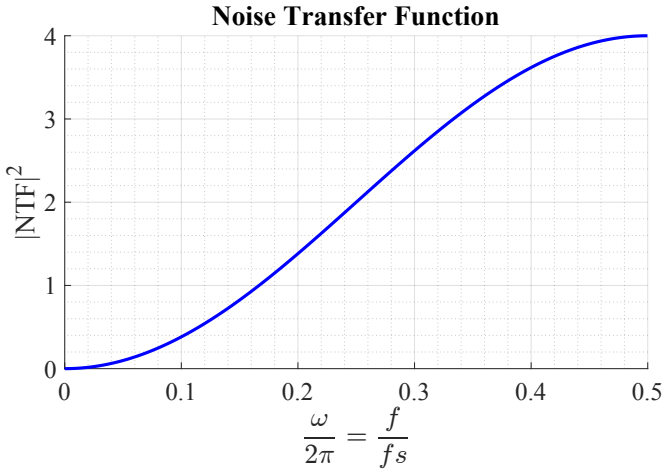


Figure 3. *NTF(z) from Eq. 2*

calculate the transfer function from the input $u(n)$ to the output $v(n)$. However, a linear model of the quantizer is required for this [6]. This explains why this is not an easy task, and, as a standard, this block is modeled as a sum of the quantizer input and a noise signal $e(n)$. This signal is known as quantization noise, as shown in Figure 2b). Note now that the output signal $v(n)$ is now a function of the input signal $u(n)$ and the quantization noise signal $e(n)$. When calculating $v(n)$ in the complex frequency domain z, it is obtained that,

$$V(z) = U(z) + (1 - z^{-1}) \cdot E(z) \tag{1}$$

This equation can be written as,

$$V(z) = STF(z) \cdot U(z) + NTF(z) \cdot E(z) \tag{2}$$

Where, the *Signal Transfer Function* (STF) and the *Noise Transfer Function* (NTF) correspond to,

$$STF(z) = 1$$

$$NTF(z) = (1 - z^{-1})$$

From Eq. 1, the input signal will be seen directly at the output, while the quantization noise signal passes through a filter given by $NTF(z)$. This filter becomes very interesting when replacing $z = e^{jw}$. Figure 3 shows the graph of the squared magnitude of $NTF(z)$ as a function of ω, normalized to an arbitrary sampling frequency f_S. Note that low-frequency spectral components (around 0) are attenuated while high-frequency components (close to 0.5) are amplified; the behavior of a *High-Pass Filter*.

Another way to reason for this behavior is in terms of power. The filter translates the power distribution of the noise from low-frequency components to high-frequency components. Interestingly, this happens only with the quantization noise but not with the signal. This effect of filtering the noise is called *noise shaping*. Shaping the noise becomes extremely useful when the input bandwidth is close to 0 (i.e., in low-bandwidth applications). Using a sampling frequency far higher from the signal bandwidth reduces the power of the noise contained in this band. The amplified higher frequency noise is filtered by the digital systems of Figure 1. Therefore, it becomes necessary to define the *Over-Sampling Ratio* as

$$OSR = \frac{f_s}{2 \cdot f_B} = \frac{f_s}{f_N} \tag{3}$$

where f_S is the modulator sampling frequency (f_{OS} in Figure 1), f_B is the input maximum frequency component, and f_N is the Nyquist frequency. This way, the *OSR* represents "how many times is the modulator sampled, compared with the Nyquist frequency."

2.1. Higher Order Delta-Sigma Modulator

As mentioned earlier, noise shaping combined with oversampling has shown to be a technique that increases the *Signal to Quantization Noise Ratio* (SQNR) over specific bandwidths when discretizing an analog signal. Intuitively, it can be concluded that the higher the oversampling rate (OSR), the more significant the increase in SQNR. By integrating the power spectral density of quantization noise within the signal bandwidth (q_{rms}), this relationship can be expressed as,

$$q_{rms}^2 = \frac{\pi^2}{3} \frac{e_{rms}^2}{OSR^3} \tag{4}$$

Where "e_{rms}" represents the quantization noise rms value (signal dependent). As expected, the in-band noise decreases with increasing OSR. However, this decrease is relatively slight; doubling the OSR reduces the noise only by 9 dB, enhancing the ENOB by only about 1.5 bits. Of course, another way to increase the modulator SQNR is to increase the loop filter order. Figure 4 shows the second order ΔΣ modulator.

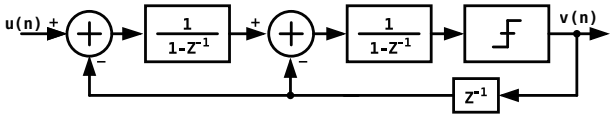


Figure 4. *Second Order Delta-Sigma Modulator*

By solving the equations for this system, the signal ($STF(z)$) and noise ($NTF(z)$) transfer functions can be found given by

$$V(z) = STF(z) \cdot U(z) + NTF(z) \cdot E(z)$$

$$V(z) = U(z) + (1 - z^{-1})^2 \cdot E(z) \tag{5}$$

Note that in this case, $NTF(z)$ now corresponds to the transfer function of a second-order high-pass filter. It is intuitive to think that by adding one more stage of integration (For a 3rd order ΔΣ modulator), the output $V(z)$ is given by

$$V(z) = U(z) + (1 - z^{-1})^3 \cdot E(z) \tag{6}$$

In this way, it can be generalized that for an L-order modulator, the quantization noise transfer function is

$$NTF(z) = (1 - z^{-1})^L \tag{7}$$

And the quantization noise in the signal band of equation 4 results in

$$q_{rms}^2 = \frac{\pi^{2L} \cdot e_{rms}^2}{(2L + 1) \cdot OSR^{2L+1}} \tag{8}$$

As the order of the filter increases, the trade-off of doubling the OSR becomes more favorable, increasing the ENOB by $L + 0.5$ bits.

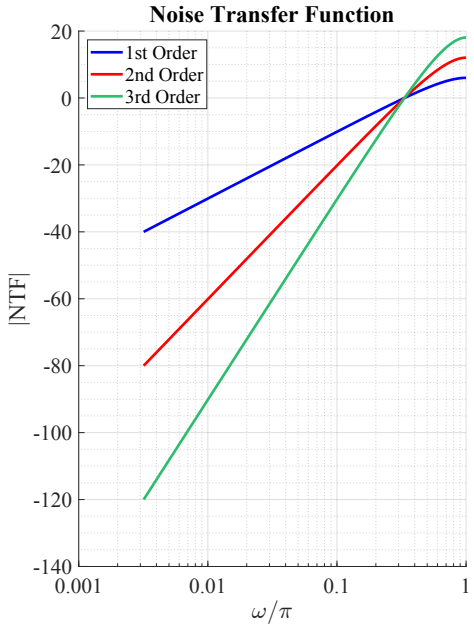


Figure 5. *$NTF(z)$ increasing the loop filter order.*

Figure 5 graphically shows how quantization noise is reduced in a bandwidth close to 0 by increasing the order of the loop filter. As the filter slope increases, the noise of the interest band decreases.

2.2. ΔΣ Modeling

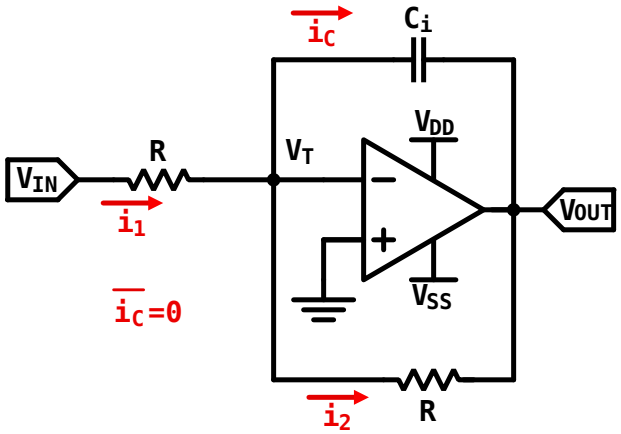
In the previous subsections, an analysis of the modulator at the system level has been carried out. On the other hand, in the following sections, the design of the blocks that make up the modulator is presented one by one. However, to ensure that the modulator works correctly, it is necessary to analyze how it behaves in the face of non-idealities of the circuits that make it up. These non-idealities added to the dependence of the input signal on quantization noise, resulting in a highly complex analysis to define circuit specifications theoretically. For this reason, a simulation is developed in MATLAB: SIMULINK, which evaluates the most relevant non-idealities of each element and establishes a table of minimum specifications for each, which guarantees that the modulator can satisfy table **??**.

The starting point for the design is the choice of loop filter order and oversampling ratio. For this purpose, simulations suggest that at a system level, specifications can be achieved with a 2nd order modulator with an OSR of 128. That is, to achieve the bandwidth of the specification, it is necessary for the system to operate at a sampling frequency (i.e., clock frequency) of

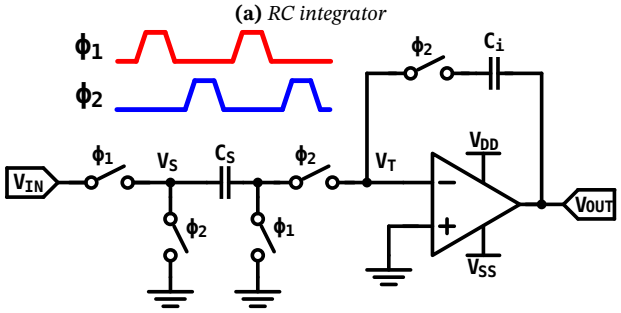
$$f_S(f_{CLK}) = OSR \cdot 2BW = 128 \cdot 40k = 5.12MHz$$

3. Loop Filter Implementation

As mentioned in the previous section, the loop filter is the most relevant system in the design of the modulator because it depends on how "good" the noise shaping will be. That is, the quality of the slopes in Figure 5 depends on the accuracy of this system. Figure 6 shows the core of the two most commonly used topologies for implementing this loop filter. The circuit shown in a) is a continuous-time integrator (RC), while that in b) is a discrete-time integrator (switched capacitors).



(a) *RC integrator*



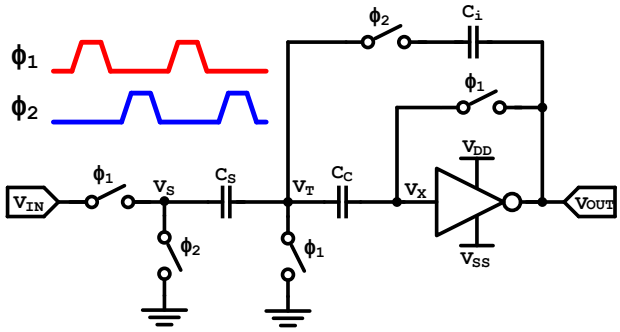
(b) *Switched Capacitors Integrator.*

Figure 6. *Conventional integrator topologies.*

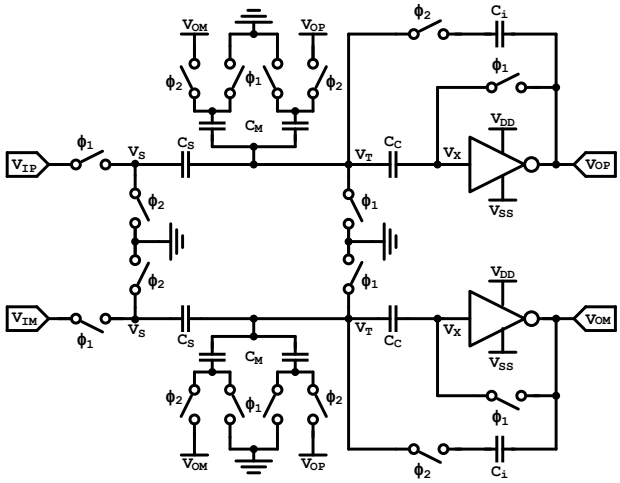
Continuous-time modulators are highly linear and capable of achieving good resolutions with wide bandwidths at the cost of power consumption in the order of mW, or higher [2] [4]. Continuous-time integrators also have an inherent low-pass filter that functions as an anti-aliasing filter. However, as integrated technology reduces its dimensions, CMOS devices are smaller but not resistors. The area used by resistors against

transistors increases. Extra layers to increase the materials' resistivity also increases the chip cost. [8] presents an alternative to this filter, with Gm-C integration topology, eliminating the need for resistors at the cost of limited output excursion ranges, limiting input excursion range. This last factor is critical when reducing the supply voltage.

On the other hand, discrete-time modulators implement switched capacitor integration filters, eliminating the need for resistors and reducing their power consumption [5] [7]. However, these filters require two non-overlapping clock phases to develop integration, so the opamp's ability to respond in just under half of the clock period limits the maximum frequency at which they can operate. In addition, since their operation is based on moving charges from one capacitor to another, the use of CMOS switches can contribute additional charges that generate noise at the filter output.



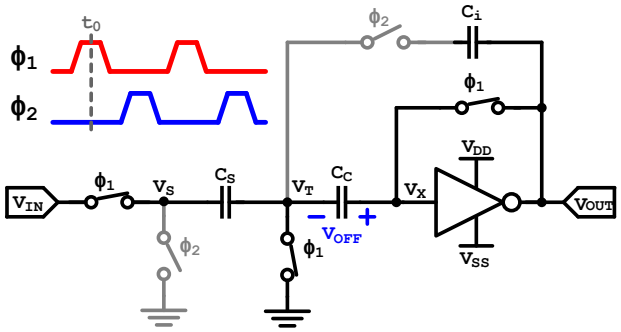
(a) *Single-Ended.*



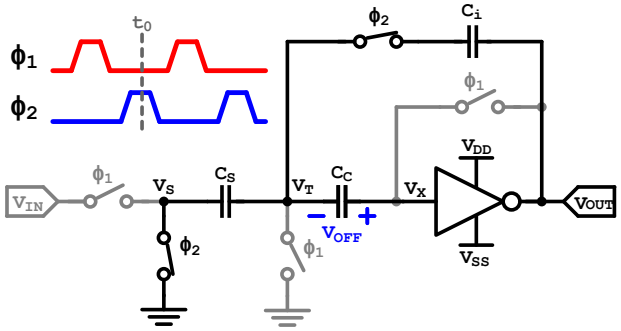
(b) *Pseudo-differential.*

Figure 7. *Inverter-Based switched capacitors integrator.*

One of the power reduction techniques implemented is reducing the supply voltage to the maximum that circuits can tolerate to function correctly. As evidenced in Figure **??**b), the input stage of opamps is often one of the most compromised when doing this. Due to this need, [3] proposes an alternative integrator based on inverters as an amplifier in a switched capacitor circuit, taking advantage of the low consumption of the topology and reducing the supply voltage to values less than $|V_{THP}| + V_{THN}$. Figure 7 shows this SC integrator topology based on the inverter as an amplifier, in a) its single-ended implementation and in b) its pseudo-differential implementation with a group of capacitors (C_M) for common mode feedback ($CMFB$) [1]. The inverter can be designed to force node V_T to signal virtual ground by implementing the auto-zero offset cancellation technique, as do the opamps in Figure 6.



(a) *Sampling Stage.*



(b) *Integration Stage.*

Figure 8. *Inverter-Based switched capacitors integrator stages.*

To explain this circuit's behavior, the integration process's different stages can be seen in Figure 8. First, in 8a), assuming any time t_0, when $\phi 1$ is high, and $\phi 2$ is low, C_S stores a charge $Q_{CS} = V_{IN} \cdot C_S$ and C_i, as one of its terminals is floating, keep the charge stored in the previous clock cycle, let us call it $Q_{Ci}(n-1)$. Note that the inverter is connected in unity feedback. For this circuit, the inverter is designed in a nominal corner to set $V_X = V_{OUT} = 0$ (signal ground). However, due to variations in the manufacturing process, device sizing, supply voltage, and voltage threshold, V_X may have an offset from signal ground. Then, the voltage drop across the capacitor C_C is $V_X - V_{REF} = V_{OFF}$.

Then, in the next phase 8b), when $\phi 1$ is low and $\phi 2$ is high, node V_S is switched to signal ground, and due to the charge previously stored in C_S and C_C, nodes V_T and V_X will instantly change to $-V_{IN}$ and $-V_{IN} + V_{OFF}$, respectively, but due to the feedback through capacitor C_i and the opamp gain, these nodes tends to return to signal ground and V_{OFF}. From the previous phase, voltage across capacitor C_C was V_{OFF}, and because no current flows to the inverter input, the voltage across this remains constant during this second phase. As V_T recovers, the charges are distributed. At the end of the phase, as $V_{CS} = 0$, the charge initially stored in C_S moves to C_i. Thus, over the clock cycles, the charge stored in C_S due to the input is accumulated in C_i, developing a "charge integration." Given the behavior of the circuit, Figure 8a) is called *sampling stage* and Figure 8b) *integration stage*.

From the previous analysis, V_{OUT} is calculated using the principle of charge conservation,

$$Q_i = Q_f$$

Where Q_i corresponds to the total charge stored by the capacitors at the end of the sampling stage and Q_f the total charge of the capacitors at the end of the integration stage, so that,

$$V_{IN} \cdot C_S + V_{OFF} \cdot C_C + Q_{Ci}(n-1) = V_{OFF} \cdot C_C + V_{OUT} \cdot C_i \tag{9}$$

As $Q_{Ci}(n-1)$ represents the charge stored by C_i in the previous integration stage, it can be seen that.

$$Q_{Ci}(n-1) = V_{OUT}(n-1) \cdot C_i$$

By replacing this value in Eq. 9, the output at the end of the n-th clock cycle is obtained as

$$V_{OUT}(n) = V_{OUT}(n-1) + \frac{C_S}{C_i} \cdot V_{IN}(n) \tag{10}$$

This equation is the differential equation of a system $H(z)$ given by

$$H(z) = \frac{V_{OUT}(z)}{V_{IN}(z)} = \frac{p}{1 - z^{-1}} \tag{11}$$

Where,

$$p = \frac{C_S}{C_i}$$

However, non-idealities in the system, such as the finite gain of the inverter as an amplifier, parasitic elements, and process variations in the sampling and integration capacitors, can eventually produce a load loss of $V_{OUT}(n-1)$. This translates into a factor q in the equation 11 such that

$$H(z) = \frac{V_{OUT}(z)}{V_{IN}(z)} = \frac{p}{1 - qz^{-1}} \tag{12}$$

where q can be an coefficient between 0.95 and 0.99 depending on the type of inverter used. For this work, a value of 0.97 best represents the performance of the inverter used and is implemented for simulink modeling.

4. Simulink System Modeling

To analyze the modulator at the system level, the complete modulator is modeled in Simulink. Figure 9 shows the differential version of the 3rd order modulator to be designed. The modulator is highlighted in purple in the center, while a sinusoidal input with an amplitude of $500mV$ is defined on the left, and the filtering required for this modulator is defined on the right. Within the modulator block, 3 stages of differential integration can be observed, each stage attempting to more accurately reproduce the system described in Figure 8b. Each of the integrators described is individually modeled in Simulink in the manner shown in Figure 10. Where V_{in} is the signal path and V_f is the converted signal from the modulator output to develop the noise shaping. The output signal is given by the equation 13.

$$Vout(n) = \frac{ps}{1 - qz^{-1}} V_{in}(z) + \frac{pf}{1 - qz^{-1}} V_f(z) \tag{13}$$

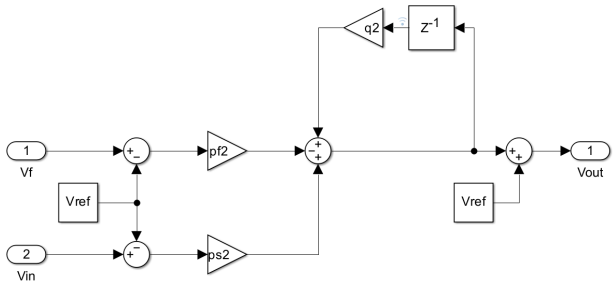


Figure 10. *Simulink Integrator model.*

These ps and pf values are very important for the resolution the modulator can achieve. Therefore, an optimization algorithm will calculate the best values for these gains. Finally, comparator power and delays do not cause any error in resolution, so they are not detailed in this analysis.

5. System Optimization

The system is optimized by 2 algorithms that have shown good results in previous experiences: "Levenberg-Marquadt" and "Adam". The objective function or loss function is given by the value of SNDR (dB) measured in the simulation multiplied by -1. In this way, the "-SNDR" term is minimized, which is the same as maximizing the pure value of SNDR.

```
%% Clean workspace
clc; clear all; close all;

%& Calculate the parameter to model de modulator
config_model

%% Search for optimal solution.
% Define an arbitrary initial point for training
initial_model = [0.3; % ps1 = pf1
                 0.3; % ps2 = pf2
                 0.8; % ps3 = pf3
                 ];

% Define the objective function to optimize
obj_function = @run_sim_and_get_SNDR;

% Calculate the model by Adam Method.
open_system('../DS3or.slx')
solution = adam(0.01, initial_model,
    obj_function);
disp(solution.value);
save('adam.mat',"solution")

% SNDR from simulation as Objective Function
function SNDR = run_sim_and_get_SNDR(model)
    % Calculate the parameter to model de
    modulator
    config_model

    ps1 = model(1);
    pf1 = ps1;
    ps2 = model(2);
    pf2 = ps2;
    ps3 = model(3);
    pf3 = ps3;
    assignin('base','ps1',ps1)
    assignin('base','pf1',pf1)
    assignin('base','ps2',ps2)
    assignin('base','pf2',pf2)
    assignin('base','ps3',ps3)
    assignin('base','pf3',pf3)

    % Run the sim
    fprintf(".")
```

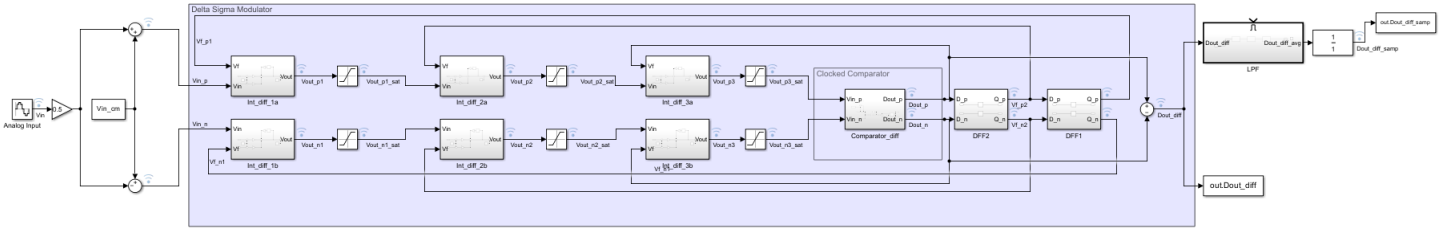


Figure 9. *Differential model of the 3rd order delta sigma modulator.*

Method	Iters	Initial Point	Optimal Solution	Objective Function
Adam	150	0.35 0.35 0.8	0.3242 0.3331 0.7955	-102.3 dB
Levenberg Marquadt	150	0.35 0.35 0.8	0.2933 0.3372 0.8586	-101.4 dB

Table 1. *Optimal Solution found.*

```
    out = sim('DS3or');

    % Get the SNDR
    [~,Dout_diff] = getData(out.Dout_diff);
    [X,f,~] = calculate_fft(Dout_diff,fclk);
    X = 20*log10(X);

    BW_index = find(f == (fs/2));
    % Get only interest BW
    X = X(1:BW_index);
    % Output signal amplitude
    [amp_dB,index] = max(X);
    % Remove main signal from spectrum to see
    noise floor
    X(index) = X(1);
    % Output noise floor
    [noise_floor,~] = max(X);
    % Calculate Measured SNDR
    SNDR = amp_dB - noise_floor;
    SNDR = -SNDR;
end
```

Code 1. *Matlab Code Used to test the algorithms.*

The algorithms are tested using the Matlab script described in the code 1, where the structure "solution" stores the output information of the method used. After 150 iterations of both methods, the results are presented in the table 1.

6. Conclusions

The project successfully highlights the critical role of optimization algorithms in improving the design of third-order delta-sigma (ΔΣ) modulators. Through extensive simulations in Simulink, we have validated the theoretical concepts and demonstrated practical improvements in modulator performance. The focus on energy efficiency and high resolution is particularly relevant for IoT applications, where power consumption and signal accuracy are critical. The results suggest that with further refinement and consideration of real-world non-idealities, the proposed design techniques can significantly contribute to the advancement of ADC technology.

References

[1] L. Wu, M. Keskin, U. Moon, and G. Temes, "Efficient common-mode feedback circuits for pseudo-differential switched-capacitor stages", in *2000 IEEE International Symposium on Circuits and Systems (ISCAS)*, vol. 5, 2000, 445–448 vol.5. DOI: 10.1109/ISCAS.2000.857467.

[2] T. Wang and L. Liang, "Analysis and design of a continuous-time sigma-delta modulator with 20 mhz signal bandwidth, 53.6 db dynamic range and 51.4 db sndr", in *4th IEEE International Symposium on Electronic Design, Test and Applications (delta 2008)*, 2008, pp. 79–84. DOI: 10.1109/DELTA.2008.17.

[3] Y. Chae and G. Han, "Low voltage, low power, inverter-based switched-capacitor delta-sigma modulator", *IEEE Journal of Solid-State Circuits*, vol. 44, no. 2, pp. 458–472, 2009. DOI: 10.1109/JSSC.2008.2010973.

[4] J. Zhang, L. Yao, and Y. Lian, "A 1.2-v 2.7-mw 160mhz continuous-time delta-sigma modulator with input-feedforward structure", in *2009 IEEE Custom Integrated Circuits Conference*, 2009, pp. 475–478. DOI: 10.1109/CICC.2009.5280787.

[5] L. H. C. Ferreira and S. R. Sonkusale, "A 0.25-v 28-nw 58-db dynamic range asynchronous delta sigma modulator in 130-nm digital cmos process", *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, vol. 23, no. 5, pp. 926–934, 2015. DOI: 10.1109/TVLSI.2014.2330698.

[6] S. Pavan, R. Schreier, and G. Temes, *Understanding Delta-Sigma Data Converters*, 2nd. Hoboken, New Jersey: John Wiley & Sons, Inc, 2017.

[7] S.-H. Liao and J.-T. Wu, "A 1 v 175 μw 94.6 db sndr 25 khz bandwidth delta-sigma modulator using segmented integration techniques", in *2018 IEEE Custom Integrated Circuits Conference (CICC)*, 2018, pp. 1–4. DOI: 10.1109/CICC.2018.8357086.

[8] C. Lee, T. Jeon, M. Jang, S. Park, Y. Huh, and Y. Chae, "26.6 a 6.5μw 10khz-bw 80.4db-sndr continuous-time Δ-Σ modulator with gm-input and 300mvpp linear input range for closed-loop neural recording", in *2020 IEEE International Solid- State Circuits Conference - (ISSCC)*, 2020, pp. 410–412. DOI: 10.1109/ISSCC19947.2020.9063074.