AN ABSTRACT OF THE THESIS OF

Cheng-Hsien Tsai for the degree of Master of Science in Electrical and Computer Engineering presented on August 10, 2021.

Title: 5th-Order Elliptic LPF Using Passive Charge Compensation Technique (PCC) Design

Abstract approved:

____

Gabor C. Temes

The passive charge compensation (PCC) technique was introduced for switched capacitor (SC) circuit to increase the slew rate and enhance the linearity performance, as PCC techniques are used on the Delta-Sigma modulator (DSM) in ADC circuitry. The PCC technique of the project was applied to the design of a SC filter to explore how PCC can relax the constraints in each of the stages. The filter designed in the study consisted of a conventional cascaded low-pass 5th order elliptic filter with bilinear, high-Q, low-Q sections and PCC switched cap branches, which were added at output of the first integrator of each section. When implemented in a TSMC 180nm CMOS process, this architecture showed a significant noise reduction of 12.3dB at the 3rd harmonic distortion and 13.1dB at the 5th harmonic distortion as compared to the conventional scheme. In additional, a rail-to-rail input and output operational amplifier was designed in a 180nm process with the complementary input pair and class-AB output stage.

5th-Order Elliptic LPF Using Passive Charge Compensation Technique (PCC) Design

by

Cheng-Hsien Tsai

A THESIS

submitted to

Oregon State University

in partial fulfillment of
the requirements for the
degree of

Master of Science

Presented August 10, 2021
Commencement June 2022

Master of Science thesis of Cheng-Hsien Tsai presented on August 10, 2021.

APPROVED:

Major Professor, representing Electrical and Computer Engineering

Head of the School of Electrical Engineering and Computer Science

Dean of the Graduate School

I understand that my thesis will become part of the permanent collection of Oregon State University libraries. My signature below authorizes release of my thesis to any reader upon request.

Cheng-Hsien Tsai, Author

ACKNOWLEDGEMENTS

It is an amazing journey to learn new knowledge and face any unknown challenges since I study at OSU. It would be impossible to complete this work without the helps and supports of my professors. I would like to express my heartfelt thanks to my advisor Professor Dr. Gabor Temes for providing me this great research opportunity under his guidance. Especially, Dr. Temes always gives pertinent comments on what I need to improve during our discussion on the progress of the research work. I could not have completed the MS degree without his guidance and support. I would also like to thank Prof. Alan Wang, Prof. Anne Trehu, and Prof. Matthew Johnston for being on committees for my final exam. I would also like to thank my colleagues and friends, Jyotindra, Manju, Jenny, Emanuel, Alex, Ray, Leo, Grace, and Wilson, for all their support during my MS study. I would like to truly thank my wife, Yi-Chun Chen, for fully supporting me in everything including raising two young kids, Ling-Rou and Cheng-Ning, at home and accompanying to the US. Finally, I do appreciate my parents for encouraging me to pursue my dream abroad.

TABLE OF CONTENTS

Page

TABLE OF CONTENTS (Continued)

LIST OF FIGURES

LIST OF FIGURES (Continued)

LIST OF FIGURES (Continued)

LIST OF TABLES

LIST OF TABLES (Continued)

LIST OF APPENDICES

Appendix Page

Chapter 1: Introduction

1.1. Overview

In recent years, the development of the transceiver system in wireless communication and biomedical equipment has motivated many design challenges, specifically in low-voltage and high-performance integrated circuits.

Filter design is one of the most important topics in signal processing. Many techniques are available to design analog as well as digital filters. Many researchers engaged in analog filter design focus on innovative circuit designs with better performance and methods, or on developing more reliable, efficient, and convenient design algorithms. Moreover, in analog and mixed analog-digital circuits, the circuit technique used most often for analog signal processing is based on switched-capacitor (SC) stages for two main reasons. First, chip area is minimized by replacing the resistors with capacitors. Another reason is that in the proper clock operations, the performance of SC circuits is generally comparable to that of conventional RC circuits. In additional, SC circuits fan out at the different bandwidths performing tunable and flexible frequency response when their clock frequency modifies also tunable SC circuits. Thus, SC circuits are used in a wide range of applications, such as analog filters, feedback amplifiers, analog-to-digital converters, and DC-DC regulators. SC circuits mainly consist of MOS switches, OTAs, and capacitors. Hence, they should consume less power, should exhibit small die-area, and must not limit the overall performance of the system.

1.2. Design Objective

The linearity, power, and area efficiencies of a filter design are critical for wireless applications with low hardware complexity and cost. Some approaches for linearity enhancement have been introduced in SC circuits, [1,2]. Both the approaches use an active operational transconductance amplifier (OTA) to implement linearity improvement In this project, we not only used presenting the passive charge compensation (PCC) technique [3] that can improve the linearity performance of a filter with less power consumption as compared to a conventional filter without a CCT circuit, but also extended the application of the PCC technique during this evaluation.

1.3. Thesis Organization

There are five chapters in this thesis. Chapter 1 mentions the background of this thesis and thesis organization. Chapter 2 introduces the fundamental theory and operation of filters. Chapter 3 describes the fundamental theory and operation of the SC circuit, how important the linearity is, and the concepts and PCC technique used in this thesis. In this chapter, the equations for low-pass filters and circuit implementation of SC filters have been derived. The implementation of such filters involved a fully differential, high bandwidth, and conventional common-mode feedback circuit (CMFB). It also discusses the simulation results demonstrating the potential of the PCC techniques for application in analog filters. Chapter 4 concludes this thesis and discusses the perspectives of the future studies on improving the linearity of SC analog filters.

Chapter 2: Filter Introduction

In this chapter, the fundamental concepts in the design of filters, and the frequency response of filters are discussed. Filters are classified according to the functions they perform. As our filters were designed for low frequency (LF) applications, we have focused on low-pass filters (LPFs) in this thesis.

2.1 Background

Filters are used in a wide range of applications such as data conversion, signal processing, and phase-locked loops, owing to their accurate frequency response, linearity, and dynamic range. Figure 2.1 shows the block diagram of a DSP system, filters that locate the front-end of ADC or the back-end of DAC allow the transmission of the desired electric signals within a certain frequency range and cancel out the transmission of the unwanted electric signals outside this range.

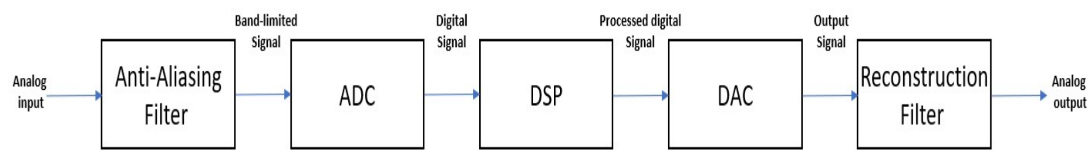


Figure 2.1. Block diagram of a DSP system.

A filter is an electrical network that alters the amplitude and/or phase characteristics of a signal with respect to frequency. The frequency-domain behavior of a filter is described mathematically in terms of its transfer function or network function. This is the ratio of the Laplace transforms of its output and input signals. The voltage transfer function H(s) of the filter shown in Figure 2.2 can be written as follows.

$$H(s) = \frac{V_{out}(s)}{V_{in}(s)} \tag{2.1}$$

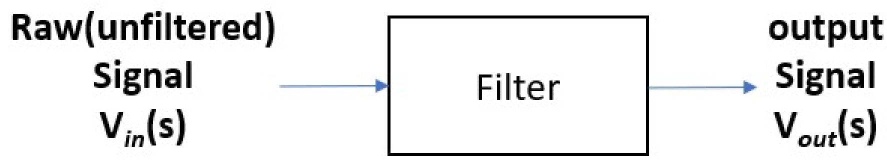


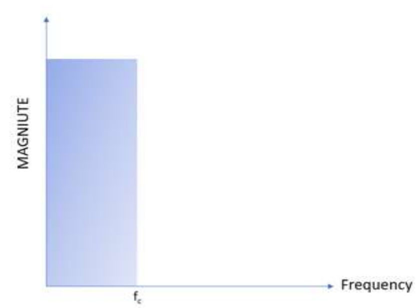
Figure 2.2. Block diagram of a filter.

where $V_{out}(s)$ and $V_{in}(s)$ are the output and input signal voltages, respectively, and s is the complex frequency variable [4].

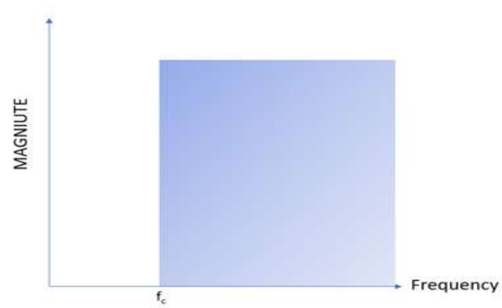
2.2 Type of Filters

Depending on the characteristics of the filter's frequency response, filters can be classified into four types, namely, generally such as low-pass filter, high-pass filter, band-pass filter, band-reject filters, etc. . Those are shown in Figure 2.3.

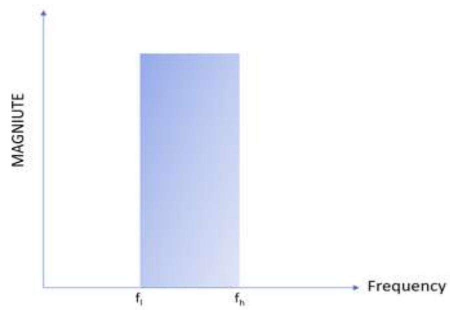
Figure 2.3 (A) shows an ideal low-pass filter. It allows the transmission of the desired electric signals at a frequency lower than the cut-off frequency, f_c. An ideal high-pass filter allows the transmission of the desired electric signals at a frequency higher than the cut-off frequency, f_c (Figure 2.3 (B)). Bandpass filters (Figure 2.3 (C)) pass only the frequencies below f_l and above f_h, and bandstop filters block only the frequencies below f_l and above f_h (Figure 2.3 (D)).



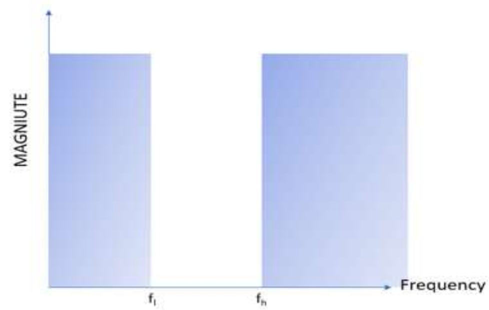
(a) Low-pass filter.



(b) High-pass filter.



(c) Band-pass filter.



(d) Band-stop filter.

Figure 2.3. Four Types of Filters.

2.3 Frequency Response

In the filter design process, first the filter type was selected. Then, the responses of the filters to the individual frequency components that constituted the input signal were defined. In practice, an LPF shows the following responses to different frequencies: pass-band, transition-band, or stop-band, as shown in Figure 2.4.

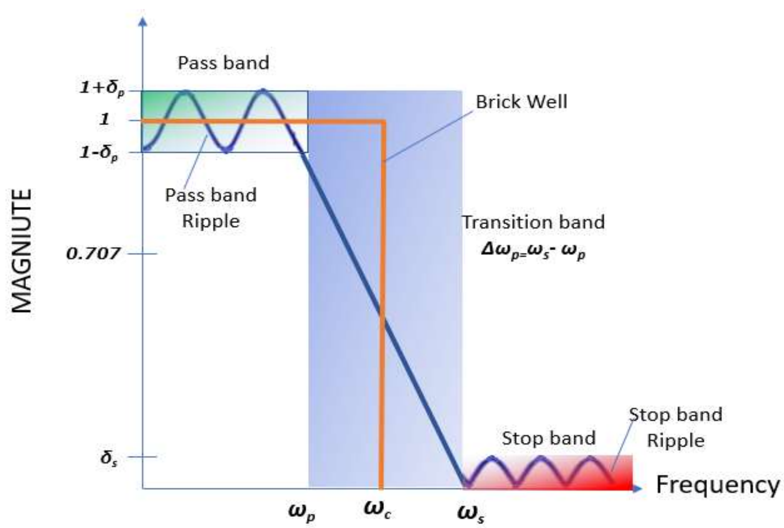


Figure 2.4. Filter parameters.

The pass-band response of a filter is its effect on the frequency components that are passed through unchanged. The frequencies within the stop-band of a filter are sharply attenuated. The transition-band represents the intermediate frequencies, which may receive some attenuation, but are not removed completely from the output signal. In practice, the magnitude may not be a constant in the passband of a filter with a small amount of ripple in the pass band known as the “passband ripple”. Similarly, the filter response does not reduce to zero with a small, non-zero value in the stop-band which is known as the “stop-band ripple”. These ripples are shown in Figure 2.4.

The ripple in the pass-band of a filter is denoted as δ_p, and its magnitude varies from 1- δ_p to 1+ δ_p. δ_s is the ripple in the stop-band [5].

Chapter 3: SC Circuit Introduction and Design Of a 5th-order Elliptic SC Filter

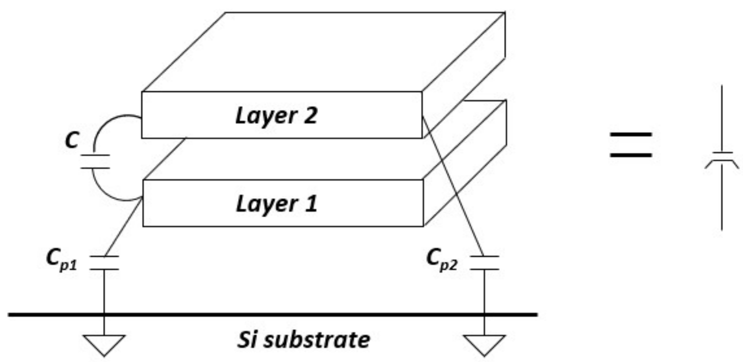
In this chapter, the fundamental concepts of filter design, and the frequency response of filters are discussed. Filters are classified according to the functions they perform. As our filters were designed for LF applications, we will focus on LPFs in this chapter.

3.1 SC Circuits

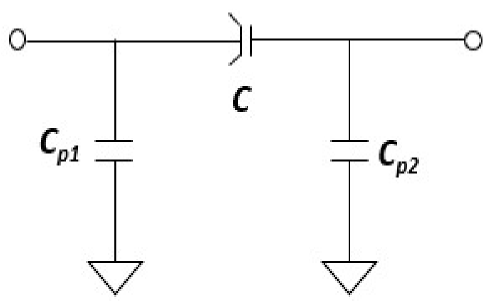
SC circuits can sample data efficiently and accurately and simulate continuous-time functions as a discrete-time signal processor. An SC circuit is realized with some basic function blocks including capacitors, switches, non-overlapping clocks, and OTAs. These blocks are discussed in this section.

3.1.1 Capacitors

A highly linear capacitance in an integrated circuit is constructed by two silicon areas (double poly capacitors), as shown in Figure 3.1(a). The desired capacitor is formed by the intersection of the two silicon layers. By growing a thin oxide between two conductive layers, it usually is accompanied by a 20% bottom plate parasitic capacitor, as shown in Figure 3.1(b)[6].



(a) Physical construction.



(b) Equivalent circuit.

Figure 3.1. Capacitor for SC circuits.

3.1.2 SC circuit

Consider the SC circuit shown in Figure 3.2. Assuming φ_1 and φ_2 are two nonoverlapping clocks. C is charged to V_1 and then V_2 during each clock period. Therefore, the change in charge over one clock period is given by

$$\Delta Q = C(V_1 - V_2) \ . \tag{3.1}$$

Then, we can also determine the equivalent average current over one clock period as follows.

$$I_{eq} = \frac{C(V_1 - V_2)}{T} \ , \tag{3.2}$$

where T is the clock period. The equivalent resistor of the SC shown in Figure 3.14 over one clock period can be expressed as follows.

$$R_{eq} = \frac{V_1 - V_2}{I_{eq}} = \frac{T}{C} = \frac{1}{C * f_s}, \tag{3.3}$$

where fs is the sampling frequency.

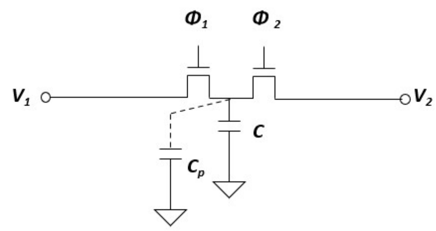


Figure 3.2. An SC resistor.

In Figure 3.2, we ignored the effect of the parasitic capacitors. Here, C_p represents the parasitic capacitor of the top plate of C as well as the non-linear capacitors associated with the two switches. It is in parallel with C, and therefore cause gain error of the circuit transfer function. To overcome this drawback, parasitic-insensitive structures have been developed to realize high accuracy. Figure 3.3 shows a parasitic-insensitive resistor equivalence of a positive SC. Figure 3.4 shows the same for a negative one [6].

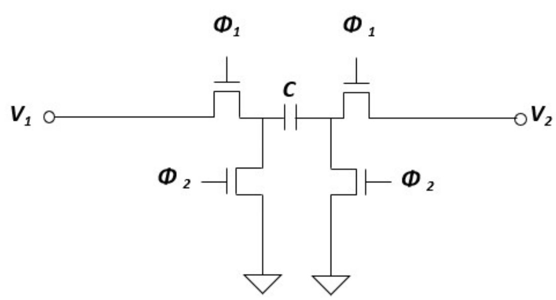


Figure 3.3. Positive parasitic-insensitive SC resistor.

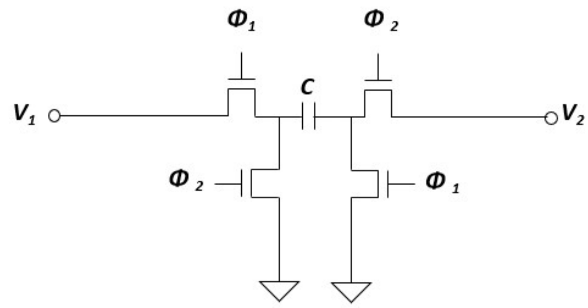


Figure 3.4. Negative parasitic-insensitive SC resistor.

3.1.3 Switches

The switches used in SC filters must have a very high "off" resistance, a relatively low "on" resistance, and no offset voltage when it turns on. In the present-day CMOS technology, MOSFETs are used as switches to meet these requirements.

The non-ideal effects of MOS switches are the major limitation for the resolution of SC circuits. They cause errors by injecting unwanted charges into the circuit when the switches turn "off". There are two types of non-ideal effects, namely charge injection and clock feedthrough.

A. Channel Charge Injection[6]

A simple sampling circuit is shown in Figure 3.5. The channel charge is given by

$$Q_{CH} = WLC_{ox}(V_{DD} - V_{in} - V_{TH}) \tag{3.4}$$

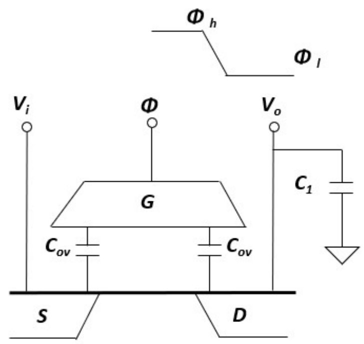


Figure 3.5. Non-ideal effects of MOS switches.

When the transistor turns off, this charge moves to the source and drain, which is called channel charge injection. The charge moving to the input is absorbed by the input source but the output is affected by the remaining channel charge deposited on to the capacitor. The output voltage deviation due to channel charge is

$$\Delta V = \frac{WLC_{ox}(V_{DD}-V_{in}-V_{TH})}{2C_l} \quad (3.5)$$

Assuming that the total channel charge moves on to the sampling capacitor, then the output voltage is given by

$$V_{out} \approx V_{in} - \frac{WLC_{ox}(V_{DD}-V_{in}-V_{TH})}{C_l} \quad (3.6)$$

and ignoring the phase shift between the input and output, the output is given by

$$V_{out} = V_{in}\left(1+\frac{WLC_{ox}}{C_l}\right) - \frac{WLC_{ox}(V_{DD}-V_{TH})}{C_l} \quad (3.7)$$

B. Clock Feedthrough

As can be observed from Figure 3.5, the overlap capacitors between the gate and junctions inject additional charge into the circuit when the switches turn “off”. This effect is called clock feedthrough. The voltage error due to the clock feedthrough is given by

$$\Delta V = -(\emptyset_h - \emptyset_l)\frac{C_{ox}}{C_l+C_{ox}} \quad (3.8)$$

where $\emptyset_h$ is V_{DD} and $\emptyset_l$ is ground.

According to equations (3.6) and (3.7), the error caused by clock feedthrough is small and signal-independent, which can be eliminated by employing a fully-differential structure. On the other hand, the error caused by charge injection is much larger. We can also divide charge injection into two parts, signal-dependent and signal-independent. Switches connected to the analog ground and virtual ground cause the signal-independent error because their turn-on voltage is constant. Just like the error caused by clock feedthrough, these errors can be eliminated by employing a fully-differential structure. Moreover, switches connected to the signal cause the signal-dependent error, which changes with the signal. This error is important because it significantly affects the resolution of the circuit. Therefore, the reduction in this error is a critical issue in SC circuits. These approaches are discussed in the following sections.

1. CMOS Switch

   MOS switches includes NMOS, PMOS, and CMOS switches, as shown as Figure 3.6. NMOS switches are applicable in low-voltage ranges, and PMOS switches are applicable at high voltages. However, CMOS switches combine the advantages of both the NMOS and PMOS switches and work at all voltages.

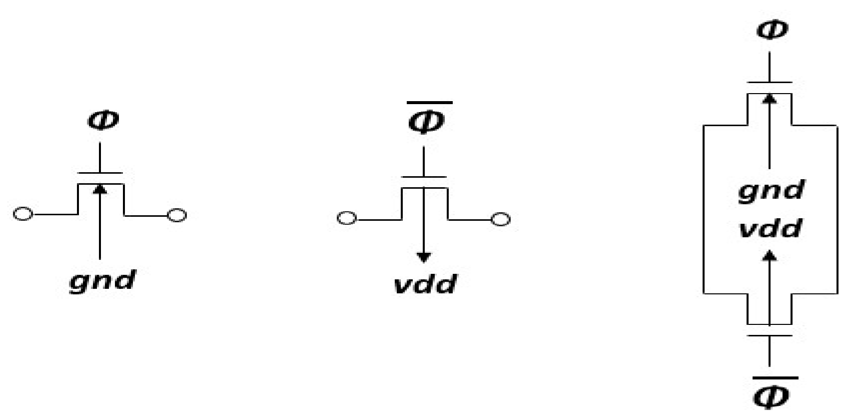


Figure 3.6. MOS switches.

2. Dummy Switch

   Figure 3.7 shows a dummy switch (M2) driven by inverse clock added to the circuit. Therefore, the charge injected by the main switch (M1) can be removed to M2 after M1 turns "off" and M2 turns "on". Note that both the source and drain of M2 are connected to the output node and the size of M2 is half of that of M1 so that $\Delta_{q1} = -\Delta_{q2}$.

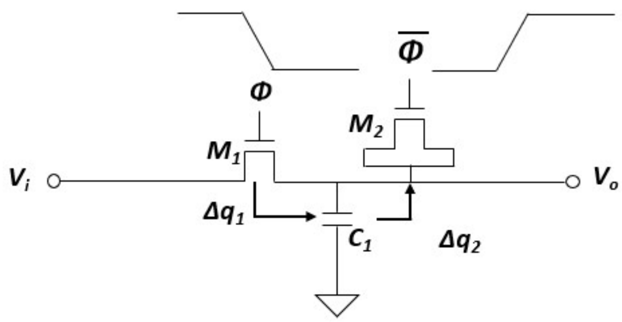


Figure 3.7. A MOS switch with a dummy switch.

3. Bottom-plate sampling method

   As shown in Figure 3.8, we added a pair of clocks (φ_{1a} ,φ_{2a}) that were slightly advance as compared to the original clocks (φ_1 ,φ_2) in an SC integrator. When M1 turns "off", the injected charge 1 Δq does not cause any change in the charge stored in Cl as M2 has already turned "off" and the right side of C1 is connected to an effective open circuit. Therefore, by this approach, the circuit is affected only by M2 and M4, which are connected to the virtual ground or analog ground. The charges injected by these transistors are signal-independent and are cancelled by the fully-differential structure.

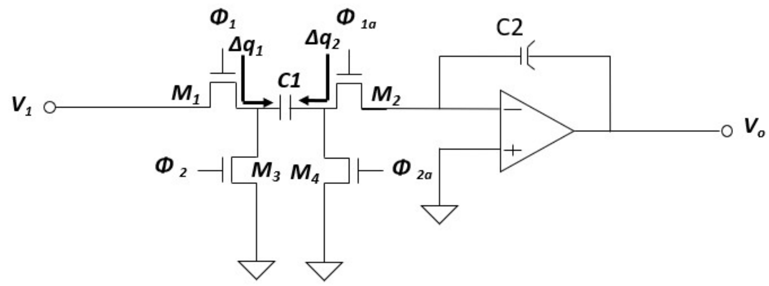


Figure 3.8. Bottom-plate sampling implementation in an SC integrator.

The size of MOS switches is discussed in the following section.

3.2 Design of a 5th Order Elliptic Low Pass SC Filter

To design the low-pass switch capacitor filter, we calculated the transfer function and the capacitance of the filter using Matlab. Then the non-idealities of the operational amplifier (OPA) and sampling switches were simulated in Cadence. To reduce the effect of the chip area on the idealities of the OPA, dynamic range and impedance scaling were conducted. Finally, a folded cascade full differential OPA and transmission gate switches were designed to implement the filter.

3.2.1 Filter Specification

Table I lists the filter specifications, which we aimed to design in this study. Switch capacitor filter is discrete time; and therefore it is was necessary to transfer the design parameters accordingly to discrete time domain specifications.

Table I. 5^{th} order Elliptic Filter Specification.

Parameter	Value
Sampling Frequency	600KHz
DC Gain	0 dB
Passband	0-36 KHz
Ripple in passband	0.229dB
Stopband	72-240 KHz
Gain in Stopband	- 51dB
Minimum Capacitor size	100 fF
Total Capacitance	4560.5fF

The design was implemented considering the addition of some margin on the filter. The pass-band ripple was set at 0.25 dB. The stop-band gain was selected to be -50 dB. Table II lists the orders of different types of filters. The elliptic filter requires the minimum filter order. Therefore, the LPF used in this study was designed using the elliptic filter structure.

Table II. Filter order calculation.

Filter	Filter Order
Butterworth	11
Chebyshev	7
Elliptic	5

3.2.2 Pole and Zero

The Bilinear transform is used to design a sampled-data filter from the analog counterpart. The relationship for the transformation of the s domain to the z domain is as follows.

$$S = \frac{2}{T}\frac{z-1}{z+1} \quad (3.9)$$

where T is the sampling period. As

$$S = j\Omega \quad (3.0)$$

$$z = e^{j\omega} \quad (3.11)$$

Ω and ω can be related as

$$\Omega = \frac{2}{T}\tan\left(\frac{\omega}{2}\right) \quad (3.12)$$

The transfer function, poles, and zeros can be calculated by using the MATLAB elliptic low pass filter design. This function can directly return the value in the z-domain. After getting the pole and zero result in the z-domain, equation (3.9) is used to change the poles and zeros from the z-domain to the s-domain in order to calculate the Q value of the 2ndorder transfer function.

The quality factors (Q) of the s-domain poles are 6.748 and 0.674, respectively, to the two complex poles shown in Table III. The poles and zeros close to each other formed the biquadratic section. These poles and zeros could be obtained through the MATLAB transcript given in appendix I. The corresponding transfer function calculated by Matlab is

$$H(z) = \frac{0.0039z^5 - 0.071z^4 + 0.0044\ \ ^3 + 0.0044z^2 - 0.0071z + 0.0039}{z^5 - 4.2348z^4 + 7.3648z^3 - 6.553\ \ ^2 + 2.9783z - 0.5524} \quad (3.13)$$

The single-ended version of each of the three filter blocks is delivered in [6] and [8], along with derivations for all of the capacitor values.

Table III. Poles and zeros in the s- and Z-domains.

Poles & Zeros	Z-Domain	S-Domain	Q-factor
P1, P2	0.884 ± 0.37i	0.543 ± 4.922i	0.674
P3, P4	0.831 ± 0.238i	-2.968 ±5.59i	6.748
P5	0.805	9.25641	-
Z1, Z2	0.804 ± 0.595i	-0.000952 ± 3.03i	-
Z3, Z4	0.595 ± 0.803i	-0.00218 ±1.984i	-
Z5	-1	α	-

3.2.3 Filter Design

The 5th order transfer function was designed as a cascade of a linear section and two second order sections. The poles and zeros were used to form biquadratic sections, a bilinear section, a high-quality (high-Q) factor section, and a low-quality (low-Q) factor section as follows.

a. Bilinear section

$$H_{Lear} = \frac{0.003943(z+1)}{z-0.805} \tag{3.14}$$

b. High-Q section, Q = 6.748

$$H_{High-} = \frac{z^2-1.0607z}{z^2-1.768z+0.9184} \tag{3.15}$$

c. Low-Q section, Q = 0.674

$$H_{Low-} = \frac{z^2-1.193z+}{z^2-1.662z+\ .7472} \tag{3.16}$$

The pole location, zero location, and frequency response plots are shown in Figures 3.9, 3.10, and 3.11, respectively.

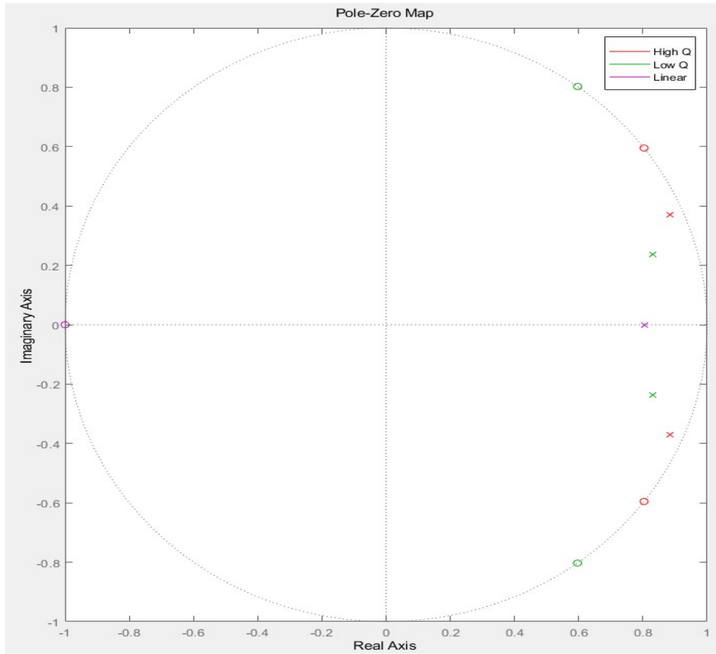


Figure 3.9. Z-domain pole and zero map.

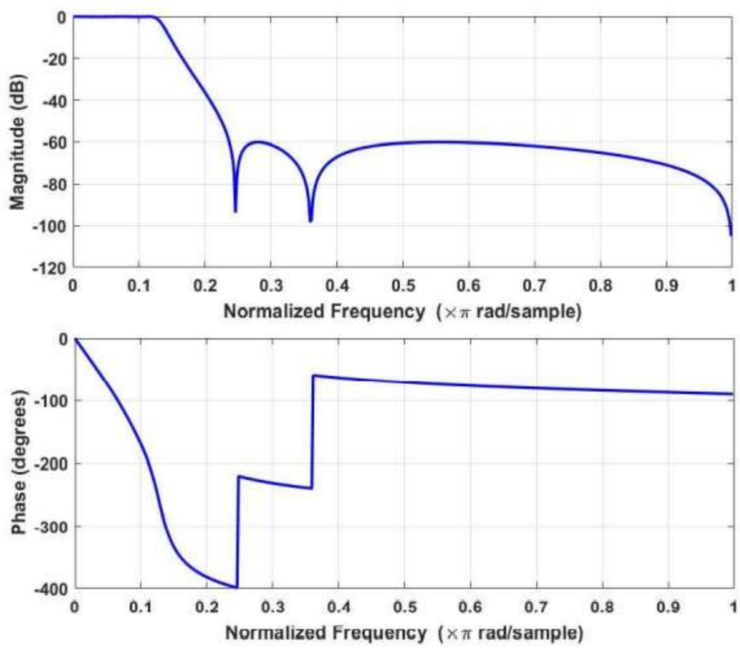


Figure 3.10. Z-domain frequency response.

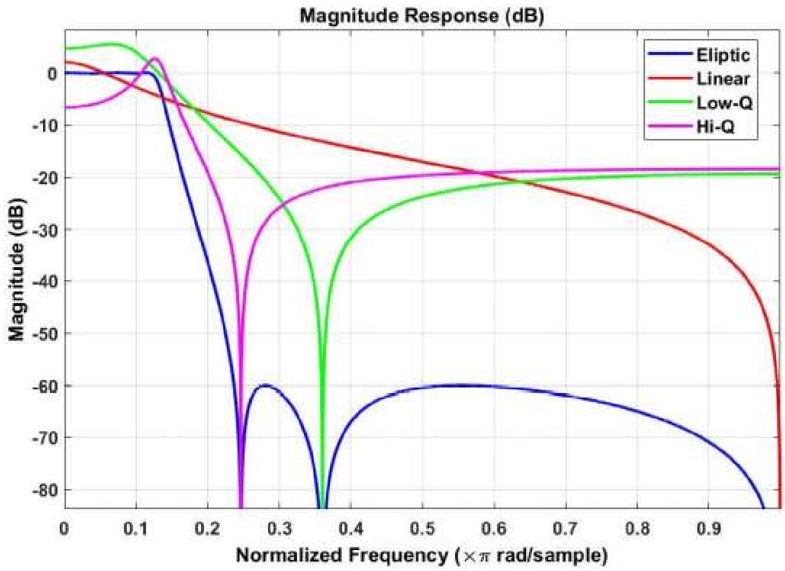


Figure 3.11. Frequency response of the elliptic filter, linear section, low-Q section, and High-Q section.

The frequency response of the elliptic filter with ripple plot, as obtained using MATLAB, is shown in Figure 3.12.

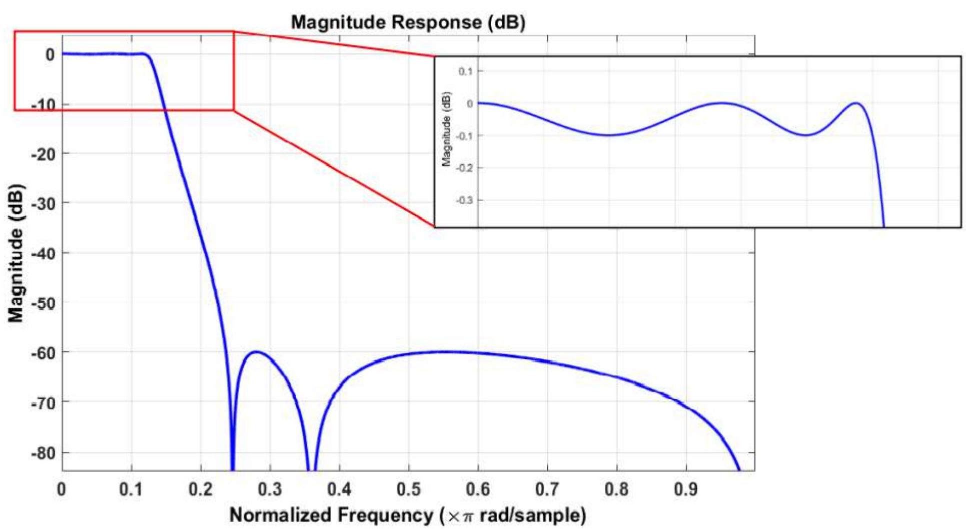


Figure 3.12. Frequency response of the elliptic filter with pass-band ripple.

3.2.4 Circuit implementation

1. Dynamic and chip area scaling

Dynamic range scaling is performed to optimize the output swing of each node of a filter. This can be done by measuring the output voltage of the amplifier in each stage.

Then by scaling the area of the capacitors, which occupy larger area than the other components, we built out the core die on a chip. Reducing the chip area is one of the most important considerations in circuit design. Therefore, chip area scaling is implemented in the design. This technique not only reduces the on-chip area, impedance level scaling, and chip area scaling, but also minimizes the noise. Chip area scaling is carried out by using the smallest capacitance connected to the input node of the OTA and setting it to the minimum allowable capacitance. In addition, all the other capacitances connected to the input node (for that stage) are scaled according to the ratio of the scaled factor for the smallest capacitance. Then, this procedure is repeated for all the stages, including the bilinear, high-Q, and low-Q. The output response does not change after implementing the chip area scaling. As we noticed, it multiplies all the capacitances by a scaling factor to the entire stage instead of only multiplying to the input node-connected capacitances, which do not affect the overall transfer function. Thus, the overall chip area is minimized.

2. Filter Cascade

The order of a filter is implemented to achieve high performance. The linear section is placed at the input in order to reject high-frequency noise. The high-Q section is placed in the middle to reduce the sensitivity and power supply rejection ratio. The low-Q section is placed at the end. The filter cascading structure is shown in Figure 3.13.

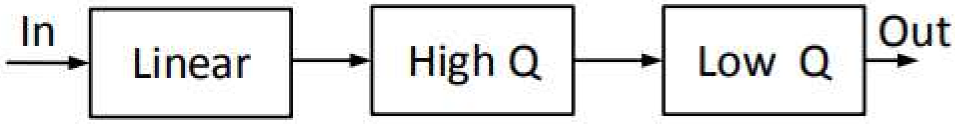


Figure 3.13. The cascading structure of an elliptic filter.

3. Fully differential structure

In most of the analog applications, it is desirable to keep signals in the differential mode. Fully-differential signals imply that the difference between two lines represents the signal component. Thus, any noise that appears as a common-mode signal on those two lines does not affect the signal. Fully differential circuits should also be balanced, implying that the differential signals operate symmetrically around a DC common-mode voltage, which is called the analog ground. Fully differential circuits have another advantage that if each single-ended signal is distorted symmetrically around the common-mode voltage, the differential signal will have only odd-order distortion terms. These terms are often much smaller than the single-ended structure. Consider the block diagram shown in Figure 3.15, if two non-linear elements are identical then each of the outputs can be determined as a Taylor series expansion given by

$$V_i = k_1V_i + k_2V_i^2 + k_3V_i^3 + \cdots \quad (3.17)$$

$$-V_i = -k_1V_i + k_2V_i^2 - k_3V_i^3 + \cdots \quad (3.18)$$

where k_i are the constant terms. In this case, the differential output signal, V_{diff}, consists of only the odd-order terms,

$$V_{diff} = 2k_1V_i + 2k_3V_i^3 + 2k_5V_i^5 + \cdots \quad (3.19)$$

With these two important advantages, most of the modern switched-capacitor circuits are realized using fully differential structures.

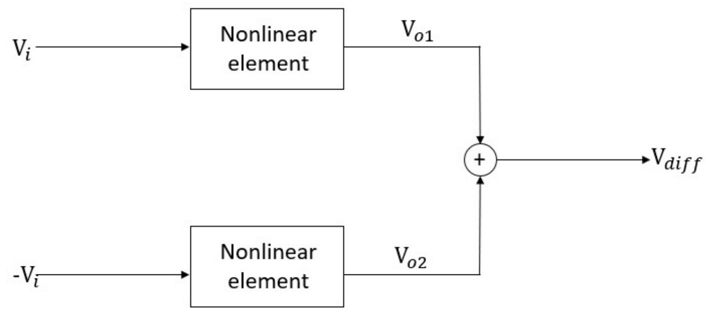


Figure 3.14. A fully differential structure with even-order term cancellation.

4. Linear section

The schematic of a linear section is shown in Figure. 3.15. The signal-flow-graph is shown in Figure. 3.16, and the corresponding values of the capacitors are listed in Table IV. The transfer function of the linear section is given by equation (3.20), which can be derived from its signal-flow-graph. The capacitor calculation is performed using equation (3.14) to determine C_{1_S1}, C_{2_S2}, and C_{3_S1}, when simplifying the calculation, C_A is set to 100fF.

$$H(z)_{Lear} = \frac{V_o(z)}{V_i(z)} = -\frac{\frac{C_{1\_S1}}{C_{F1}}(1-z^{-1})+\frac{C_{2\_S1}}{C_{F1}}}{1-z^{-1}+\left(\frac{C_{3\_S1}}{C_{F1}}\right)} = -\frac{\left(\frac{C_{1\_S1}+C_{2\_S1}}{C_{F1}}\right)z-\frac{C_{1\_S1}}{C_{F1}}}{\left(1+\frac{C_{3\_S1}}{C_{F1}}\right)z-1} \tag{3.20}$$

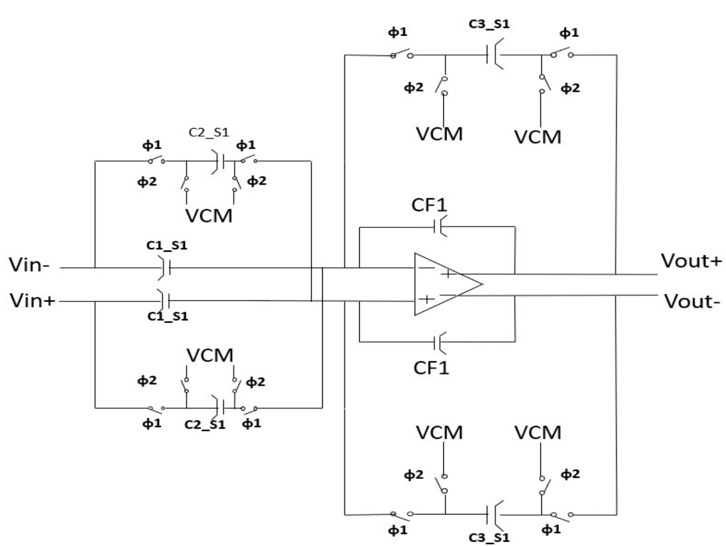


Figure 3.15. First linear stage.

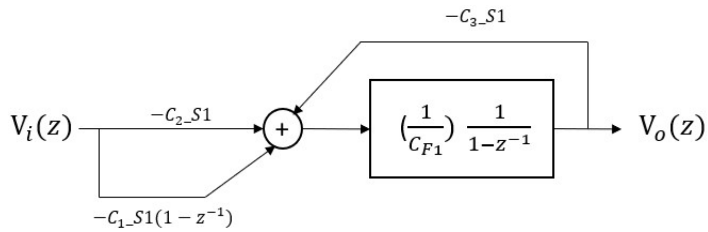


Figure. 3.16 Signal-flow-graph of the linear section.

Table IV First Stage capacitances.

	Bilinear Section Caps value table	
	DR Unit (fF)	Area (fF)
CF1	825	825
C1_S1	100	100 (Unit Cap)
C2_S1	200	200
C3_S1	200	200

5. High-Q section

The high-Q section, which has a pole quality factor of 6.748, is placed at the middle. The schematic of the high-Q section and the capacitances in the second stage (Table V are shown in Figure 3.17. The signal-flow-graph is shown in Figure. 3.18. The quality factor of each filter is determined from the pole frequency as follows.

$$Q = \frac{|\omega_p|}{2Re(\omega_p)} \tag{3.21}$$

The transfer function of the linear section is expressed as (3.21), which can be derived from its signal-flow-graph and equation (3.15).

$$H(z)_{High-} = \frac{V_o(z)}{V_i(z)}$$

$$= -\frac{(C_{3\_S2})z^2+(C_{1\_S2}C_{5\_S2}+C_{2\_S2}C_{5\_S2}-2C_{3\_S2})z+(C_{3\_S2}-C_{2\_S2}C_{5\_S2})}{z^2+(C_{4\_S2}C_{5\_S2}+C_{6\_S2}C_{5\_S2}-2)z+(1-C_{6\_S2}C_{5\_S2})} \tag{3.22}$$

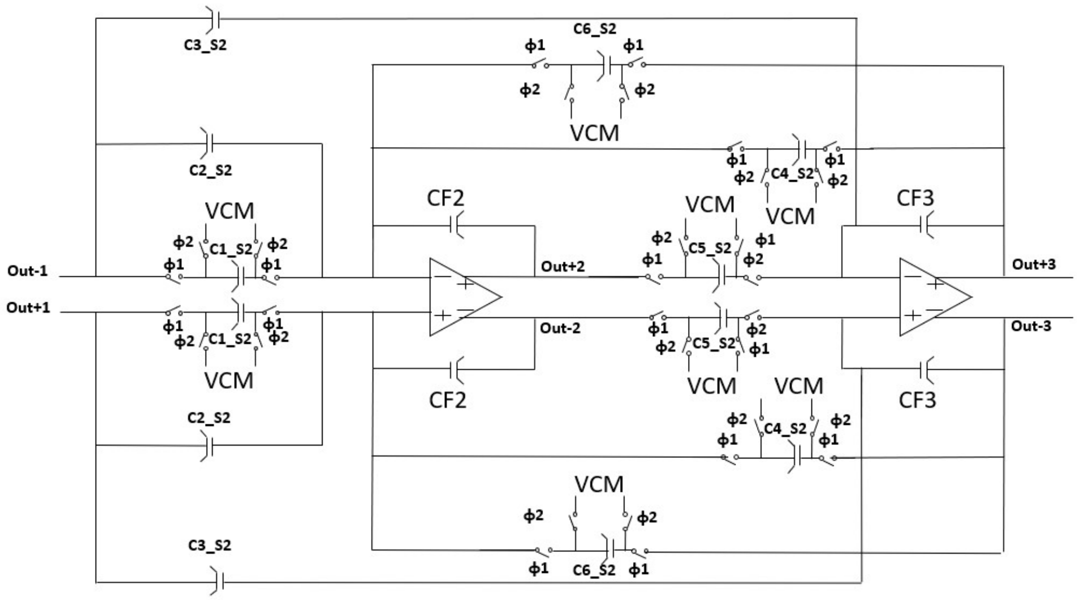


Figure. 3.17 Second high-Q stage.

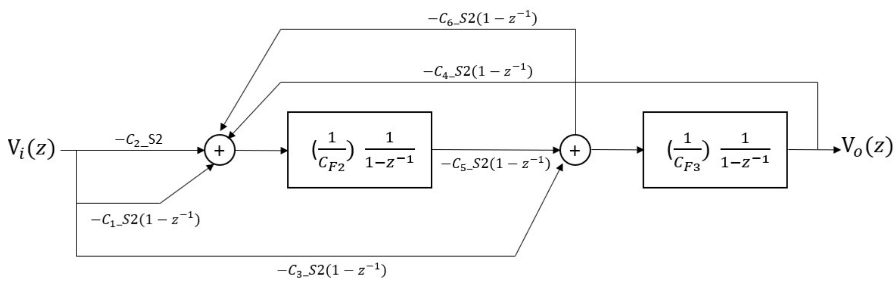


Figure 3.18. Signal-flow graph of the high-Q section.

Table V. Capacitance of the second stage.

	High-Q Section Caps value table	
	DR Unit (fF)	Area (fF)
C6_S2	103	100 (Unit Cap)
CF2	463	348
CF3	374	363
C1_S2	133	100 (Unit Cap)
C2_S2	0	0
C3_S2	100	97
C4_S2	184.5	179
C5_S2	141	106
C6_S2	103	100 (Unit Cap)

6. Low-Q section

The low-Q section, which has a pole quality factor of 0.67, is placed at the end. The schematic of the low-Q section and the capacitances in the third stage (Table VI) are shown in Figure 3.19. The signal-flow-graph is shown in Figure. 3.20.

The transfer function of the linear section is expressed as (3.23), which can be derived from its signal-flow graph and equation (3.16).

$$H(z)_{Low-Q} = \frac{V_o(z)}{V_i(z)}$$

$$= -\frac{(C_{2\_S3}+C_{3\_S3})z^2+(C_{1\_S3}C_{5\_S3}-C_{2\_S3}-2C_{3\_S3})z+(C_{3\_S3})}{(1+C_{6\_S3})z^2+(C_{4\_S3}C_{5\_S3}-C_{6\_S3}-2)z+1} \tag{3.23}$$

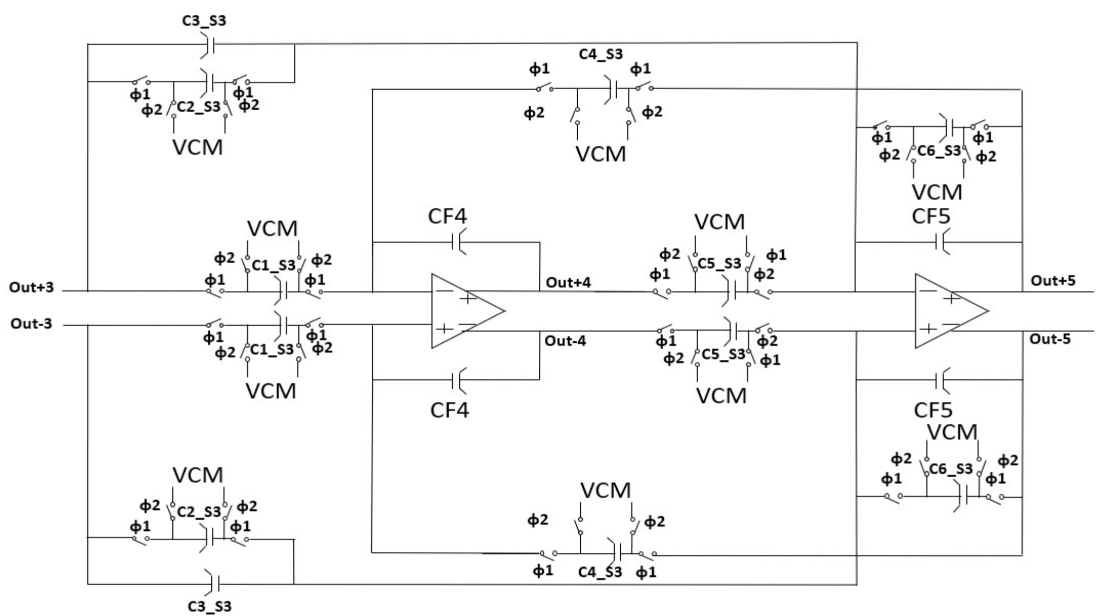


Figure 3.19. Third low-Q stage.

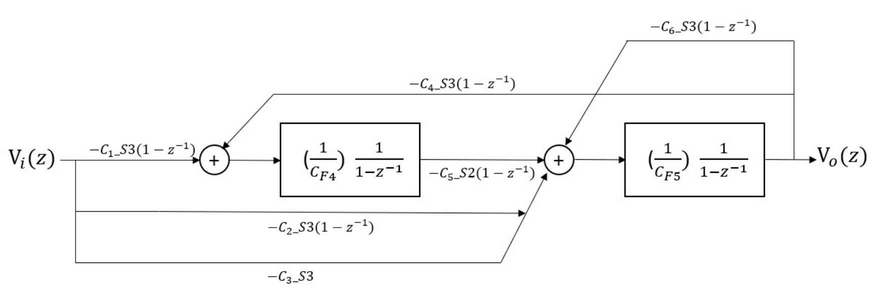


Figure 3.20. Signal-flow graph of the low-Q section.

Table VI. Third stage capacitances.

	Low-Q Section Caps value table	
	DR Unit (fF)	Area (fF)
CF4	530.5	530
C1_S3	141.7	141
C2_S3	0	0
C3_S3	100	100 (Unit Cap)
C4_S3	100	100 (Unit Cap)
CF5	500	500
C5_S3	302.5	302
C6_S3	169	169

7. Switches Sizing [8]

Clock feedthrough and charge injection cause setup error voltage in sampling signals. And the nonlinearity of switches introduces harmonics to the sampling signals. At the beginning, we took biquad filters as two first-order systems with a single delay around the loop because of the negative SC resistor. In addition, the output error of an ideal first-order system is given by the following equation.

$$V_e(t) = e^{-\frac{t_s}{\tau}} \tag{3.24}$$

where τ is the RC time constant of the first-order system and t_s is the settling time of the first-order system. Assuming the circuit has a resolution of N bits, then (3.16) can be rewritten as

$$e^{-\frac{t_s}{\tau}} \leq \frac{1}{2^{N+1}} \tag{3.25}$$

If we set the capacitance loading to 2 pF, the settling time as the half of the clock period, and the resolution as 8 bits, then the turn-on resistance limitation is given by

$$R_{on} \leq \frac{t_s}{(N+1)\ln(2)C_l} \approx 0.13\ k\Omega \tag{3.26}$$

From equation (3.25), we can determine the size of the NMOS switches. As for the CMOS switches, we still have to consider the ratio of the sizes of the NMOS and PMOS switches.

First of all, we recall that the equation of transistor current vs. voltage in the triode region is given by

$$I_d = \mu C_{ox}\left(\frac{W}{L}\right)[\left(V_{gs} - V_{th}\right)V_{ds} - \frac{1}{2}{V_{ds}}^2] \qquad (3.27)$$

Then turn-on resistance of MOS switches is given by

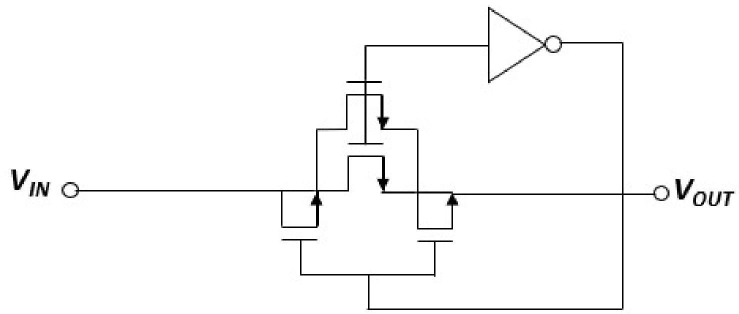
$$R_{on} = (\frac{\partial I_d}{\partial V_{ds}})^{-1} = \frac{1}{\mu C_{ox}\left(\frac{W}{L}\right)(V_{DD}-V_{in}-V_{TH})}) \qquad (3.28)$$

According to (3.19) the size of the switch transistor is

$$\frac{W}{L} = \frac{1}{\mu C_{ox}\left(\frac{W}{L}\right)(V_{DD}-V_{in}-V_{TH})R_{on}} \approx 3 \qquad (3.29)$$

Assuming the length of the channel to be minimum i.e., L = 0.18 µm, then W = 0.54 µm. Generally, the mobility of a PMOS switch is the one-third of that of an NMOS switch. Thus, the size of a PMOS switch should be three times larger that of an NMOS switch to provide equivalent resistance.

Figure 3.21 shows the switch used in our LPF. It consisted of four NMOS transistors with the W/L ratio of 0.54/0.18, and a CMOS inverter was used for providing an opposite phase clock to the dummy switches.



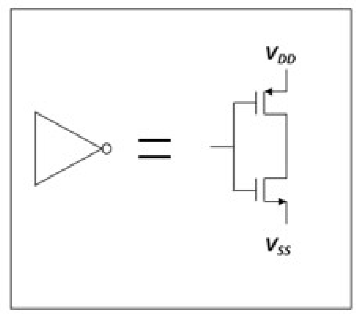


Figure 3.21. Proposed charge injection canceling switch.

$$\Delta V = -\frac{Q_{ch}}{2(C_l+C_{ov})} \approx -\frac{WLC_{ox}(V_{gs}-V_{th})}{2C_l} \tag{3.30}$$

7. Design of OTA

The TSMC 180nm CMOS technology was used to design the OTAs. The OTAs designed in this study are shown in Figures 3.22–3.26. The folded cascade structure was used, the minimum length of the transistors in the output path was set to 1 µm to achieve a gain of 60 dB. The CMFB was utilized to provide an output DC operation point for the OPA, and the trans-conductance of the CMFB circuit was about the half of the input trans-conductance that could provide enough common-mode bandwidth. The size of the transistors, the bias circuit, and specification are listed in Tables VII – Table XV. The frequency response of the first stage OTA is summarized in Table VII, the DC gain was approximately 54.15 dB, the bandwidth was 6.2 MHz, and the phase margin of the OTA was 92.35°. The frequency response of the filter at each stage and the overall frequency response of the filter with the real OTA are summarized in Tables VII–XV. The pass-band ripple was approximately 0.229 dB and the stop-band attenuation was -51dB. When the amplitude of the input sine signal was 2 Vpp and the frequency was 30 kHz, the output waveform of the filter and the FFT results are shown in Figures. 3.27– 3.30. Sampling caused glitches in the output signal. The 3rd harmonic was -110.25 dB and the 5th harmonic was -124.42 dB.

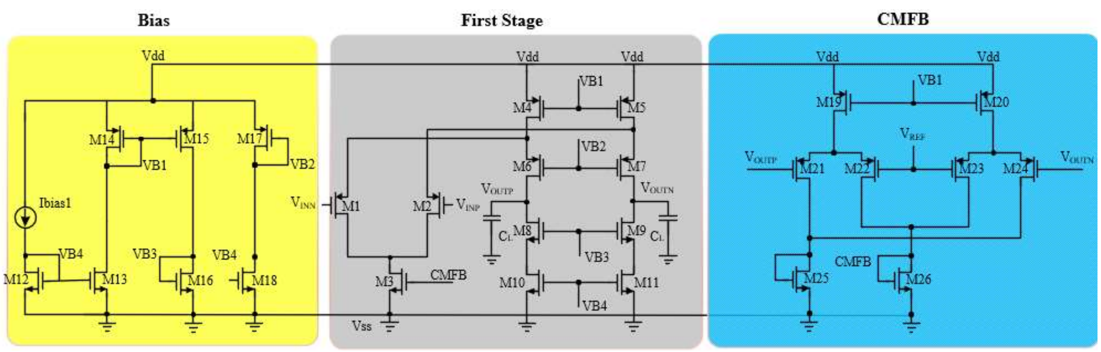


Figure 3.22. Schematic of the first stage real OPA.

Table VII. Specification Of First transistor.

	1st Order OPA - Folded cascode opamp
	Pre-sim
loading (pF)	2
DC Gain (dB)	54.15
Ft(KHz)	11.18
PM (degree)	92.35
SR+(us/sec)	2.15
SR-(us/sec)	1.60
CM(vol)	0.9
CMRR (dB)	87.7
power (uW)	225.128
UBM (MHz)	6.2

Table VIII. Size Of First transistor and biasing circuit.

Folded Cascade OPA							
Transistor	W/L (um)	Finger	Multiplier	Transistor	W/L (um)	Finger	Multiplier
M1	400/300	2	1	M7	600/400	6	1
M2	400/300	2	1	M8	400/300	2	1
M3	400/300	10	1	M9	400/300	2	1
M4	600/400	17	2	M10	400/300	2	1
M5	600/400	17	2	M11	400/300	2	1
M6	600/400	6	1				

Biasing Circuit							
Transistor	W/L (um)	Finger	Multiplier	Transistor	W/L (um)	Finger	Multiplier
M12	400/300	4	1	M16	400/300	1	1
M13	400/300	4	1	M17	600/400	4	1
M14	600/400	22	1	M18	400/300	4	1
M15	600/400	22	1				

CMFB Circuit							
Transistor	W/L (um)	Finger	Multiplier	Transistor	W/L (um)	Finger	Multiplier
M19	600/400	10	1	M23	600/400	4	1
M20	600/400	10	1	M24	600/400	4	1
M21	600/400	4	1	M25	400/300	2	1
M22	600/400	4	1	M26	400/300	2	1

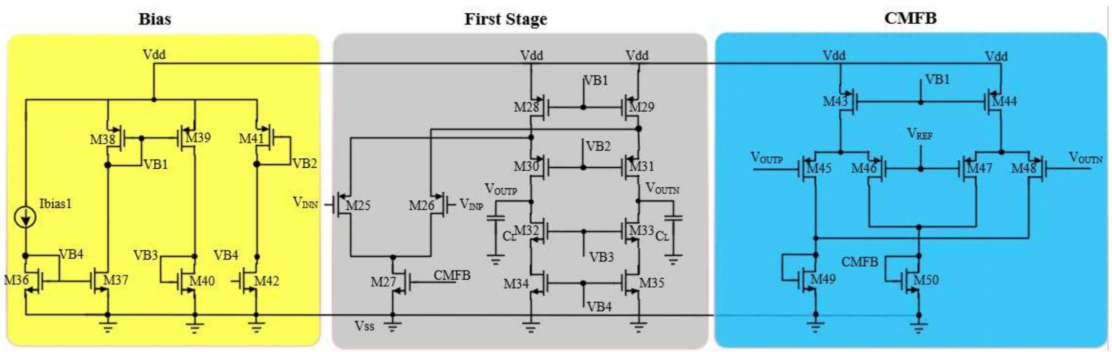


Figure 3.23 Schematic of the second stage real OPA.

Table IX. Specification of the second transistor.

	2nd Order OPA - Folded cascode opamp
	Pre-sim
loading (pF)	2
DC Gain (dB)	63.77
Ft(KHz)	3.69
PM (degree)	89.19
SR+(us/sec)	2.067
SR-(us/sec)	1.423
CM(vol)	0.9
CMRR (dB)	95.79
power (uW)	247.038
UBM (MHz)	5.6

Table X. Size of the second transistor and biasing circuit.

Folded Cascade OPA							
Transistor	W/L (um)	Finger	Multiplier	Transistor	W/L (um)	Finger	Multiplier
M25	250/500	2	2	M31	470/500	6	1
M26	250/500	2	2	M32	260/500	3	1
M27	250/500	16	3	M33	260/500	3	1
M28	400/500	17	2	M34	260/500	3	1
M29	400/500	17	2	M35	260/500	3	1
M30	470/500	6	1				

Biasing Circuit							
Transistor	W/L (um)	Finger	Multiplier	Transistor	W/L (um)	Finger	Multiplier
M36	250/500	10	1	M40	250/500	2	1
M37	250/500	10	1	M41	400/500	4	1
M38	400/500	20	1	M42	250/500	10	1
M39	400/500	20	1				

CMFB Circuit							
Transistor	W/L (um)	Finger	Multiplier	Transistor	W/L (um)	Finger	Multiplier
M43	500/500	9	1	M47	400/500	9	1
M44	500/500	9	1	M48	400/500	9	1
M45	400/500	9	1	M49	250/500	4	2
M46	400/500	9	1	M50	250/500	4	2

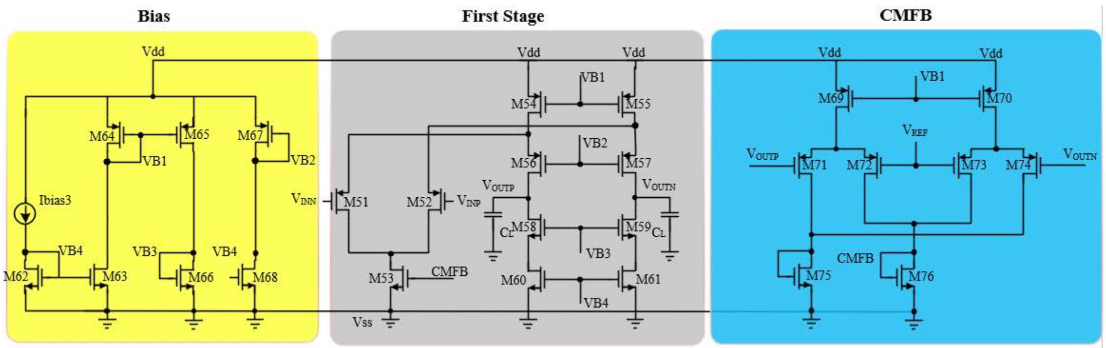


Figure 3.24. Schematic of the third stage real OPA.

Table XI. Specification of the third transistor.

	3rd Order OPA - Folded cascode opamp
	Pre-sim
loading (pF)	2
DC Gain (dB)	62.63
Ft(KHz)	25.76
PM (degree)	85.89
SR+(us/sec)	16.06
SR-(us/sec)	8.458
CM(vol)	0.9
CMRR (dB)	93.24
power (uW)	287.993
UBM (MHz)	34.05

Table XII. Sizes of the third transistor and biasing circuit.

Folded Cascade OPA							
Transistor	W/L (um)	Finger	Multiplier	Transistor	W/L (um)	Finger	Multiplier
M51	400/300	2	1	M57	600/400	12	2
M52	400/300	2	1	M58	400/300	4	1
M53	400/300	10	2	M59	400/300	4	1
M54	600/400	17	4	M60	400/300	4	1
M55	600/400	17	4	M61	400/300	4	1
M56	600/400	12	2				

Biasing Circuit							
Transistor	W/L (um)	Finger	Multiplier	Transistor	W/L (um)	Finger	Multiplier
M62	400/300	4	1	M66	400/300	1	1
M63	400/300	4	1	M67	600/400	4	1
M64	600/400	22	1	M68	400/300	4	1
M65	600/400	22	1				

CMFB Circuit							
Transistor	W/L (um)	Finger	Multiplier	Transistor	W/L (um)	Finger	Multiplier
M69	600/400	10	1	M73	600/400	4	1
M70	600/400	10	1	M74	600/400	4	1
M71	600/400	4	1	M75	400/300	2	1
M72	600/400	4	1	M76	400/300	2	1

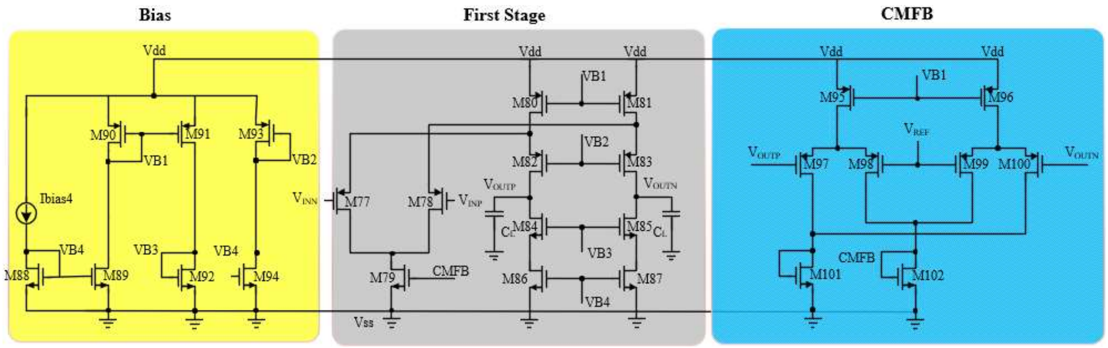


Figure 3.25 Schematic of the fourth stage real OPA.

Table XIII. Specification of the fourth transistor.

	4th Order OPA - Folded cascode opamp
	Pre-sim
loading (pF)	2
DC Gain (dB)	72.48
Ft(KHz)	11.18
PM (degree)	80.05
SR+(us/sec)	18.05
SR-(us/sec)	4.46
CM(vol)	0.9
CMRR (dB)	100.81
power (uW)	262.893
UBM (MHz)	40.9

Table XIV. Size of the fourth transistor and biasing circuit.

Folded Cascade OPA							
Transistor	W/L (um)	Finger	Multiplier	Transistor	W/L (um)	Finger	Multiplier
M77	250/500	30	1	M83	470/500	9	1
M78	250/500	30	1	M84	260/500	3	1
M79	250/500	16	3	M85	260/500	3	1
M80	400/500	38	1	M86	260/500	3	1
M81	400/500	38	1	M87	260/500	3	1
M82	470/500	9	1				

Biasing Circuit							
Transistor	W/L (um)	Finger	Multiplier	Transistor	W/L (um)	Finger	Multiplier
M88	250/500	10	1	M92	250/500	2	1
M89	250/500	10	1	M93	400/500	4	1
M90	400/500	20	1	M94	250/500	10	1
M91	400/500	20	1				

CMFB Circuit							
Transistor	W/L (um)	Finger	Multiplier	Transistor	W/L (um)	Finger	Multiplier
M95	500/500	9	1	M99	400/500	9	1
M96	500/500	9	1	M100	400/500	9	1
M97	400/500	9	1	M101	250/500	4	2
M98	400/500	9	1	M102	250/500	4	2

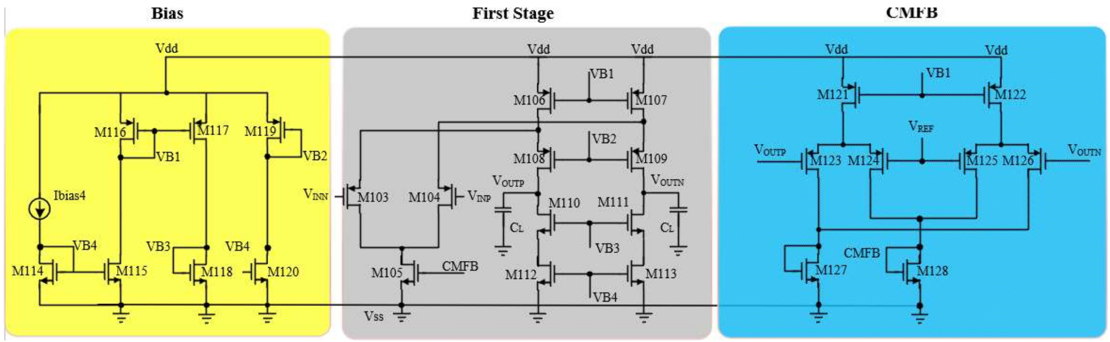


Figure 3.26 Schematic of the fifth stage real OPA.

Table XV. Specification of the fifth transistor.

	5th Order OPA - Folded cascode + gain bosted opamp
	Pre-sim
loading (pF)	2
DC Gain (dB)	90.91
Ft(KHz)	1.22
PM (degree)	84.63
SR+(us/sec)	23.56
SR-(us/sec)	3.61
CM(vol)	0.9
CMRR (dB)	128.82
power (uW)	319.103
UBM (MHz)	43.52

Table XVI. Size of the fifth transistor and biasing circuit.

Folded Cascade OPA							
Transistor	W/L (um)	Finger	Multiplier	Transistor	W/L (um)	Finger	Multiplier
M103	250/500	30	1	M109	470/500	9	2
M104	250/500	30	1	M110	260/500	3	2
M105	250/500	16	6	M111	260/500	3	2
M106	400/500	38	2	M112	260/500	3	2
M107	400/500	38	2	M113	260/500	3	2
M108	470/500	9	2				

Biasing Circuit							
Transistor	W/L (um)	Finger	Multiplier	Transistor	W/L (um)	Finger	Multiplier
M114	250/500	10	1	M118	250/500	2	1
M115	250/500	10	1	M119	400/500	4	1
M116	400/500	20	1	M120	250/500	10	1
M117	400/500	20	1				

CMFB Circuit							
Transistor	W/L (um)	Finger	Multiplier	Transistor	W/L (um)	Finger	Multiplier
M121	500/500	9	2	M125	400/500	9	2
M122	500/500	9	2	M126	400/500	9	2
M123	400/500	9	2	M127	250/500	4	4
M124	400/500	9	2	M128	250/500	4	4

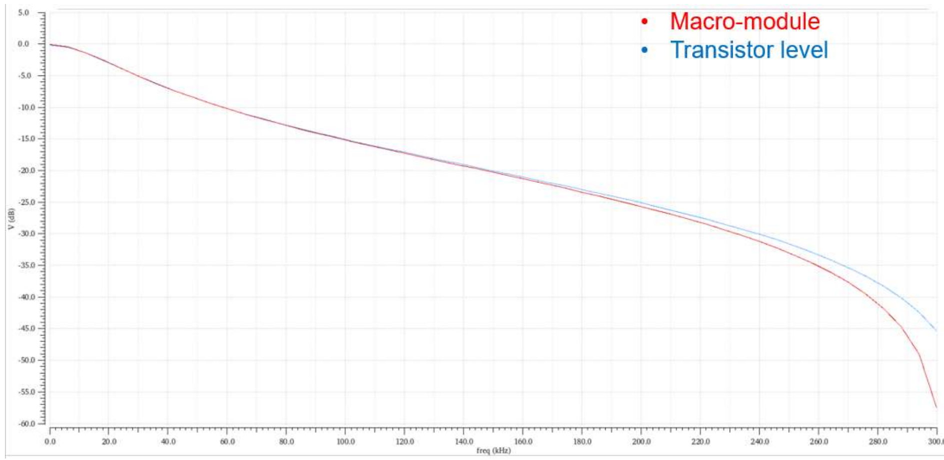


Figure 3.27 Frequency response of the linear section between the ideal and transistor levels.

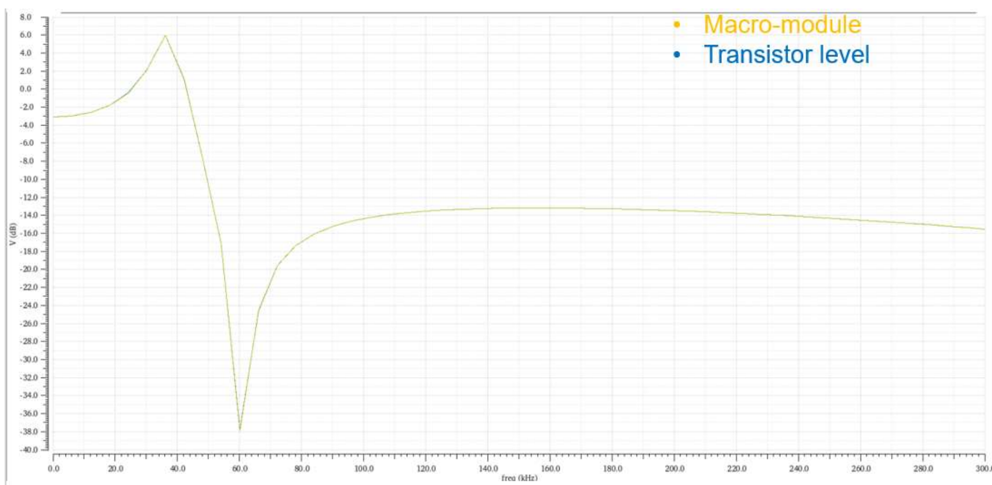


Figure 3.28 Frequency response of the high-Q section between the ideal and transistor levels.

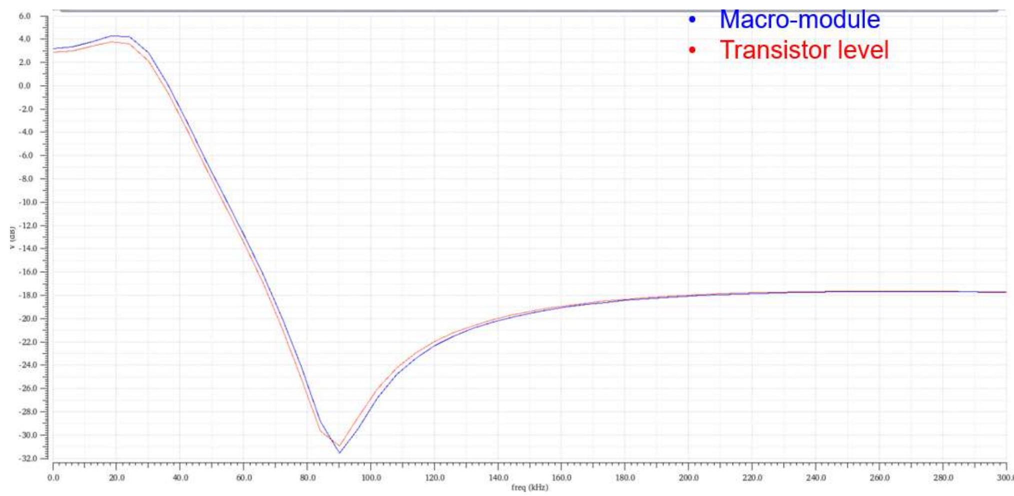


Figure 3.29. Frequency response of the low-Q section between the ideal and transistor levels.

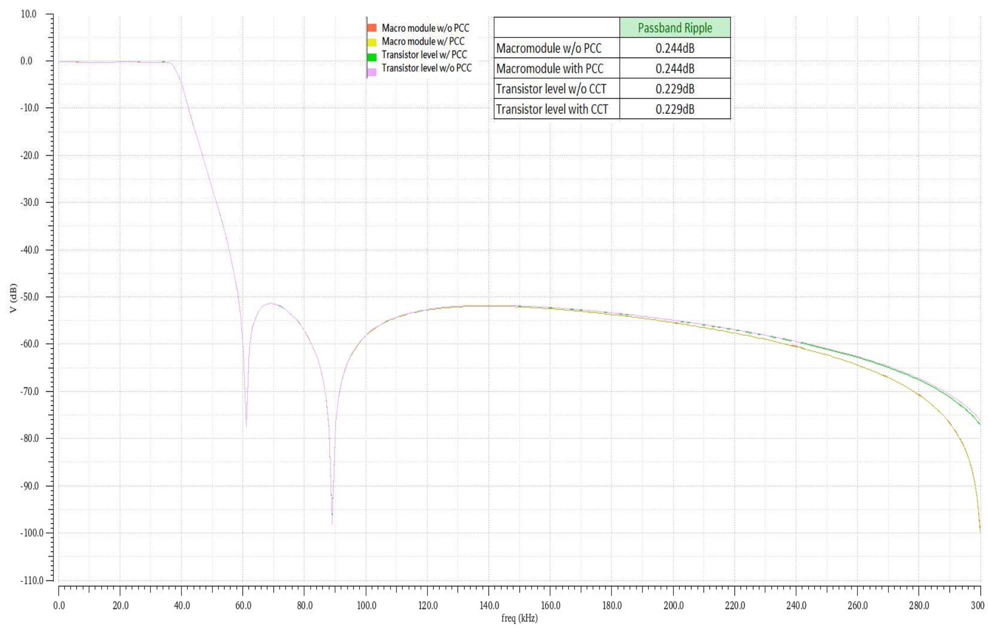


Figure 3.30. Frequency response of the LPF between the ideal and transistor levels.

In this work, a 5th order elliptic LPF was analyzed and designed. To ensure the high performance of the LPF, the non-idealities of the OPA and sampling switches were simulated. The TSMC 0.18 µm CMOS technology was employed to verify this design. The transistor level simulation results are summarized in Table XVII.

Table XVII. Specification for the LPF.

Parameter	Value
Sampling Frequency	600KHz
DC Gain	0 dB
Passband	0-36 KH
Ripple in passband	0.229dB
Stopband	72-240 KHz
Gain in Stopband	- 51dB
Minimum Capacitor size	100 fF
Total Capacitance	4560.5fF

3.3 PCC Technique [3]

The linearity, power, and area efficiencies of a filter design are critical for wireless applications with low hardware complexity and cost. These integrators in filter application should have high linearity to achieve a high SNDR and to avoid the input referred noise, which affects the overall linearity performance [9].

Several techniques have been proposed to improve the slew rate of OTAs in switched capacitor circuits [10][11]. These techniques involve the designing of an active component and auxiliary amplifier to share the redundant current flows, resulting in the settling down of the fully-differential OPA circuit to the common ground level with enough current flows. These approaches enhance the linearity by improving the slew rate, but at the expense of chip efficiency. This study focuses on the slew rate enhancement, which would reduce the settling time with an additional passive charge compensation path adding at the output of the OTA [3]. The proposed technique is shown in Figure 3.31. During phase S_1, the input is sampled onto C_1, while the charge compensation capacitor C_3 holds the previous output voltage. During the charge transfer phase S_2, the OTA needs to provide charge equal to C_1V_{in} to the top plate of C_2 in addition to charging the load capacitor. In the charge compensation technique, an additional charge proportional to the input voltage is also provided onto the top plate of C_2 provided through C_3. Ideally, if V_{in} between S_1 and S_2 does not change, and if the optimum value of C_3 is chosen, the OTA does not need to provide any charge, hence power can be saved in biasing the OTA. If the input varies slowly (i.e., it is oversampled) the charge provided by the OTA can still be greatly reduced as only the charge proportional to the difference between the previous input voltage and current input voltage needs to be provided. Considering the effect of C_L, C_3 can be expressed as (3.31).

$$C_3 = C_1 \frac{1+({C_L}/{C_2})}{1-({C_1}/{C_2})} = C_1 \frac{C_2+C_L}{C_2-C_1}, since\ C_2/C_1 < 1 \tag{3.31}$$

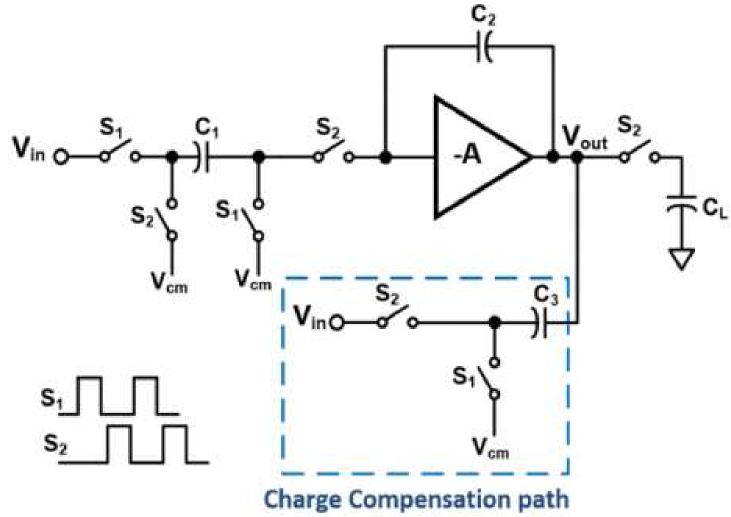


Figure 3.31. A charge-compensated integrator [3].

3.3.1 PCC Technique Design and Implementation

In the LPF design, the DC-gain loops of each OTA were less than 1, we derived the capacitance needed at the output of the OTA of the linear , high-Q, and low-Q sections. We added PCC paths at the output of the 1st integrator in the high-Q and low-Q sections during the evaluation because the output of the 2nd integrator connected the unknown capacitor load.

1. Designing the PCC capacitance of the linear section

The linear section with a PCC path is shown in Figure 3.32.

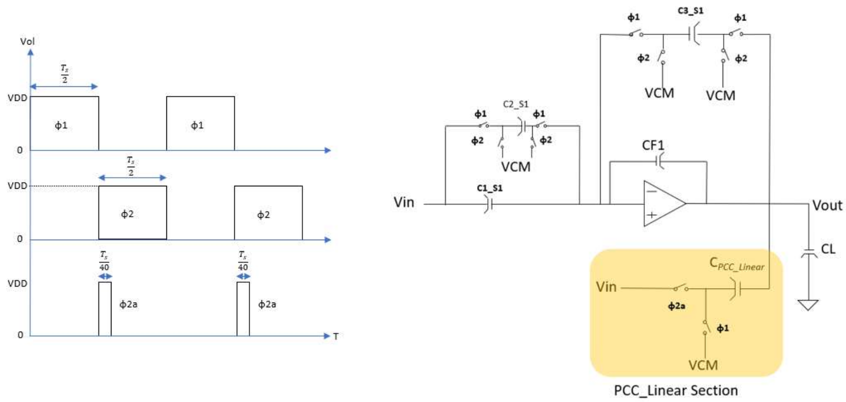


Figure 3.32. Linear section with a PCC path.

During the Φ2 phase, by KCL, we can derive equations (3.21) – (3.24), to determine the C_{PCC_Linear}.

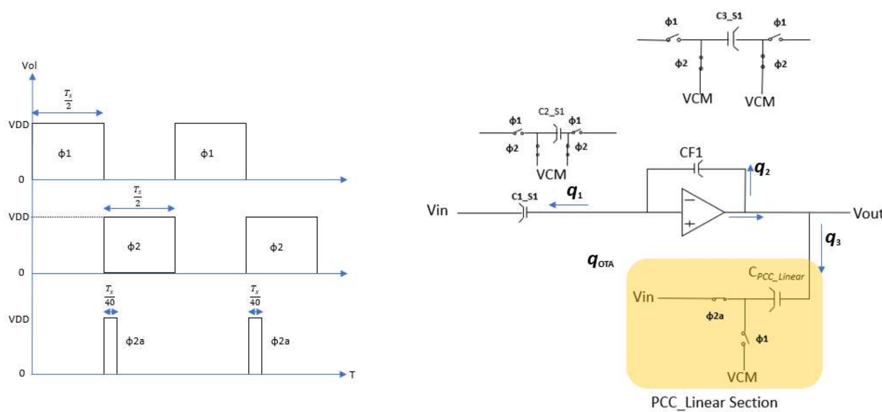


Figure 3.33. Linear section with a PCC path during the Φ2 phase operation.

$$q_1[n] = C_{1\_S1} * V_{in}[(n - 1/2)] \tag{3.32}$$

$$q_2[n] = C_{CF1} * \{V_{out}(n) - V_{out}(n - 1/2)\} \tag{3.33}$$

As we know, q_2 *and* q_3 are discharged to ground.

Then, q_3 =[$V_{out}[(n - 1/2))$-0 –($V_{out}[(n - 1) - 0)]$

So, $V_{out}[n - 1/2] = V_{out}[n - 1]$ (3.34)

Put (3) into (3.22), $q_2[n] = C_{CF1} * \{V_{out}(n) - V_{out}[(n - 1))]$

$V_{out}(n) - V_{out}(n - 1) = \frac{C_{1\_S1}}{C_L} V_{in}(n - 1/2)$

$q_L[n] = C_L * V_{out}[n] - 0 - \{V_{out}(n - 1/2) - 0\}$

$$= C_L * V_{out}[n] - V_{out}(n - {}^{1}/_{2})$$

$$= C_L \frac{C_{1\_S1}}{C_L} V_{in}(n - {}^{1}/_{2}) \qquad (3.35)$$

$$C_{PCC\_Linear} = C_{1\_S1} \frac{1+\frac{C_L}{C_2}}{1-\frac{C_1}{C_2}} = C_{1\_S1} \frac{C_{2\_S1}+C_L}{C_{2\_S1}-C_{1\_S1}} \approx 264fF \qquad (3.36)$$

2. Designing the PCC capacitance of the high-Q section

The high-Q section with a PCC path is shown in Figure 3.34.

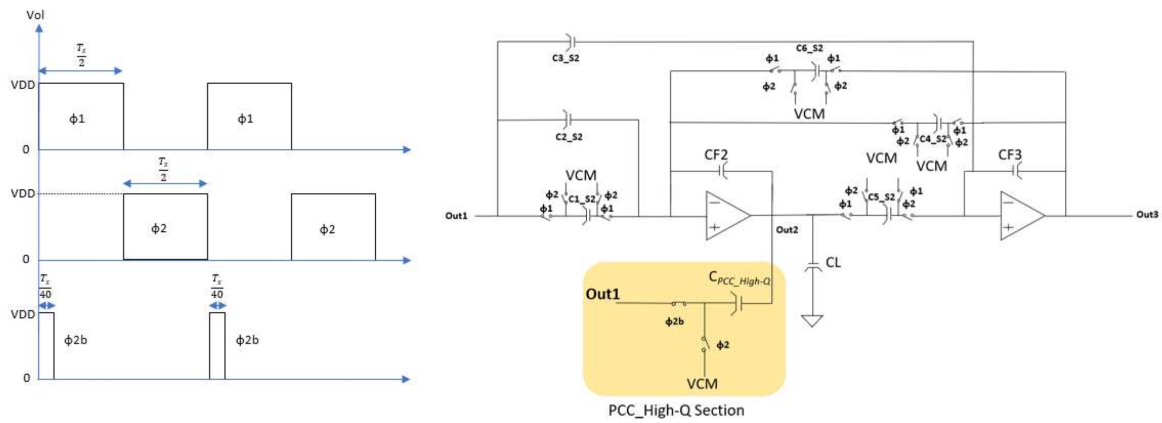


Figure 3.34. High-Q section with a PCC path.

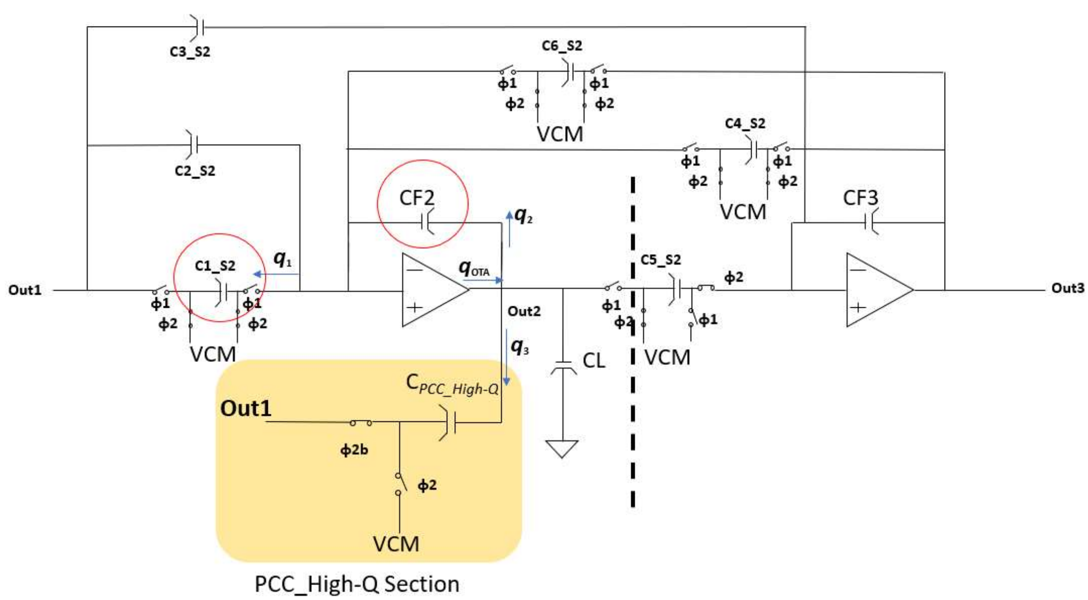


Figure 3.35. High-Q section with a PCC path at the Φ2 phase.

During the Φ2 phase, by KCL, we derived equations (3.32) – (3.35) to derive $C_{PCC\_High\text{-}Q}$,

$$\mathrm{q}_1[\mathrm{n}] = C_{1\_S2} * V_{out1}[(n - {}^1/_2)] \tag{3.37}$$

$$\mathrm{q}_2[\mathrm{n}] = C_{CF2} * \{V_{out2}(n) - V_{out2}(n - {}^1/_2)\} \tag{3.38}$$

As we know, if q_2 *and* q_3 are discharged to ground.

Then, q_3 =[$V_{out2}[(n - {}^1/_2))$-0 –($V_{out2}[(n-1) - 0)]$

So, V_{out} $[\mathrm{n} - {}^1/_2] = V_{out2}[n-1]$ (3.39)

Put (3) into (3.22), $\mathrm{q}_2[\mathrm{n}] = C_{CF2} * \{V_{out2}(n) - V_{out2}[(n-1))]$

$V_{out2}(n) - V_{out2}(n-1) = \frac{C_{1\_S2}}{C_L} V_{in}(n - {}^1/_2)$

$$\begin{aligned}\mathrm{q}_\mathrm{L}[\mathrm{n}] &= C_L * V_{out2}[n] - 0 \ - \{V_{out2}(n - {}^1/_2) - 0\} \\ &= C_L * V_{out2}[n] \ - V_{out2}(n - {}^1/_2) \\ &= C_L \frac{C_{1\_S2}}{C_L} V_{in}(n - {}^1/_2)\end{aligned} \tag{3.40}$$

$$C_{PCC\_High-Q} = C_{2\_S2} \frac{1+\frac{C_L}{C_{F2}}}{1-\frac{C_{2\_S2}}{CF}} = \ C_{2\_S2} \frac{CF2+C_L}{CF2-C_{2\_S2}} \ \approx 186.6fF \tag{3.41}$$

3. Designing the PCC capacitance of the low-Q section

The low-Q section with a PCC path is shown in Figure 3.36.

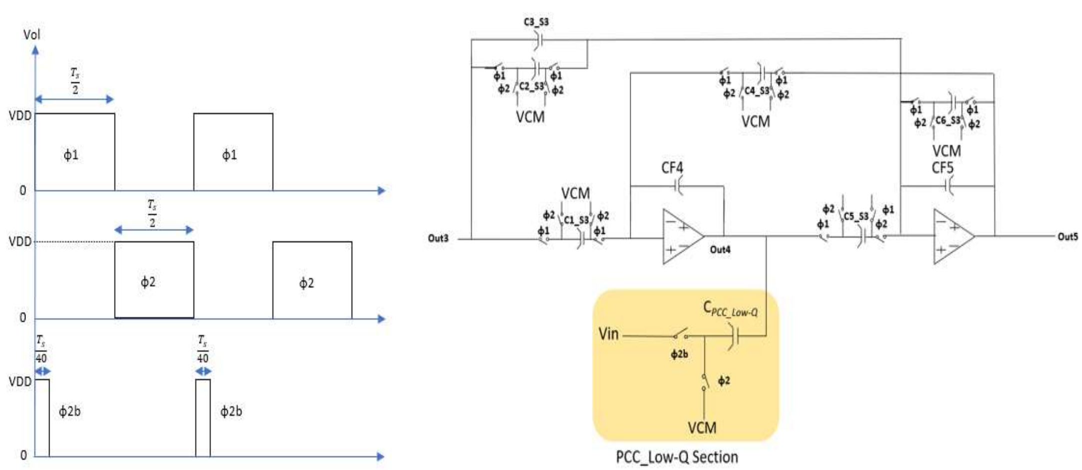


Figure 3.36. Low-Q section with a PCC path.

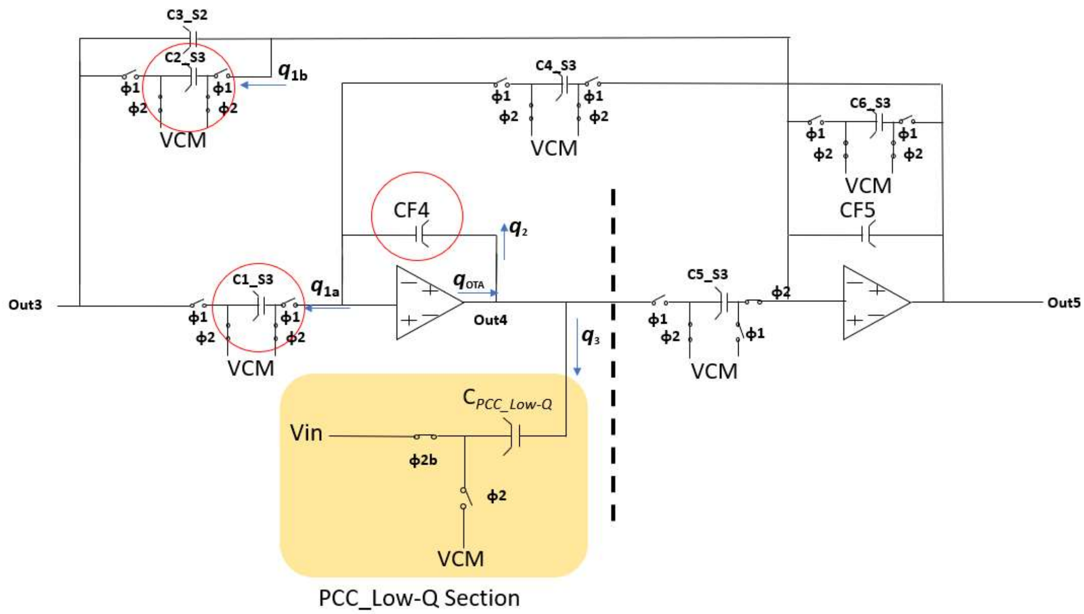


Figure 3.37. Low-Q section with a PCC path at the Φ2 phase.

During the Φ2 phase, by KCL, we derived equations (3.32) – (3.35) to determine $C_{PCC\_High\text{-}Q}$,

$$q_{1a}[n] = C_{1\_S3} * V_{out3}[(n - 1/2)] \tag{3.42}$$

$$q_{1b}[n] = C_{2\_S3} * V_{out3}[(n - 1/2)] \tag{3.43}$$

$$q_1[n] = q_{1a}[n] + q_{1b}[n] = (C_{1\_S3} + C_{2\_S3}) * V_{out3}[(n - 1/2)] \tag{3.44}$$

$$q_2[n] = C_{CF} \ * \{V_{out4}(n) - V_{out4}(n - 1/2)\} \tag{3.45}$$

As we know, q_2 *and* q_3 are discharged to ground.

Then, q_3 =$[V_{out4}[(n - 1/2))$-0 –$(V_{out4}[(n - 1) - 0)]$

$$\text{So, } V_{out4}[n - 1/2] = V_{out4}[n - 1] \tag{3.46}$$

Put (3) into (3.22), $q_2[n] = C_{CF4} * \{V_{out4}(n) - V_{out4}[(n - 1))]$

$$V_{out4}(n) - V_{out4}(n - 1) = \frac{C_{1\_S3} + C_{2\_S3}}{C_L} V_{in}(n - 1/2)$$

$$q_L[n] = C_L * V_{out4}[n] - 0 \ - \{V_{out4}(n - 1/2) - 0\}$$

$$= C_L * V_{out4}[n] \ - V_{out4}(n - 1/2)$$

$$= C_L \frac{C_{1\_S3}+C_{2\_S3}}{C_L} V_{in}(n - {}^{1}\!/_{2}) \quad (3.47)$$

$$C_{PCC\_LOW-Q} = (C_{1\_S3} + C_{2\_S3}) \frac{1+\frac{C_L}{C_{F4}}}{1-\frac{(C_{1\_S3}+C_{2\_S3})}{C_{2\_S2}}} = (C_{1\_S3} + C_{2\_S3}) \frac{C_{2\_S3}+C_L}{C_{2\_S3}-C_{1\_S3}-C_{2\_S3}} \approx 56fF \quad (3.48)$$

The completed LPF circuit with the proposed PPC technique is shown in Figure 3.38.

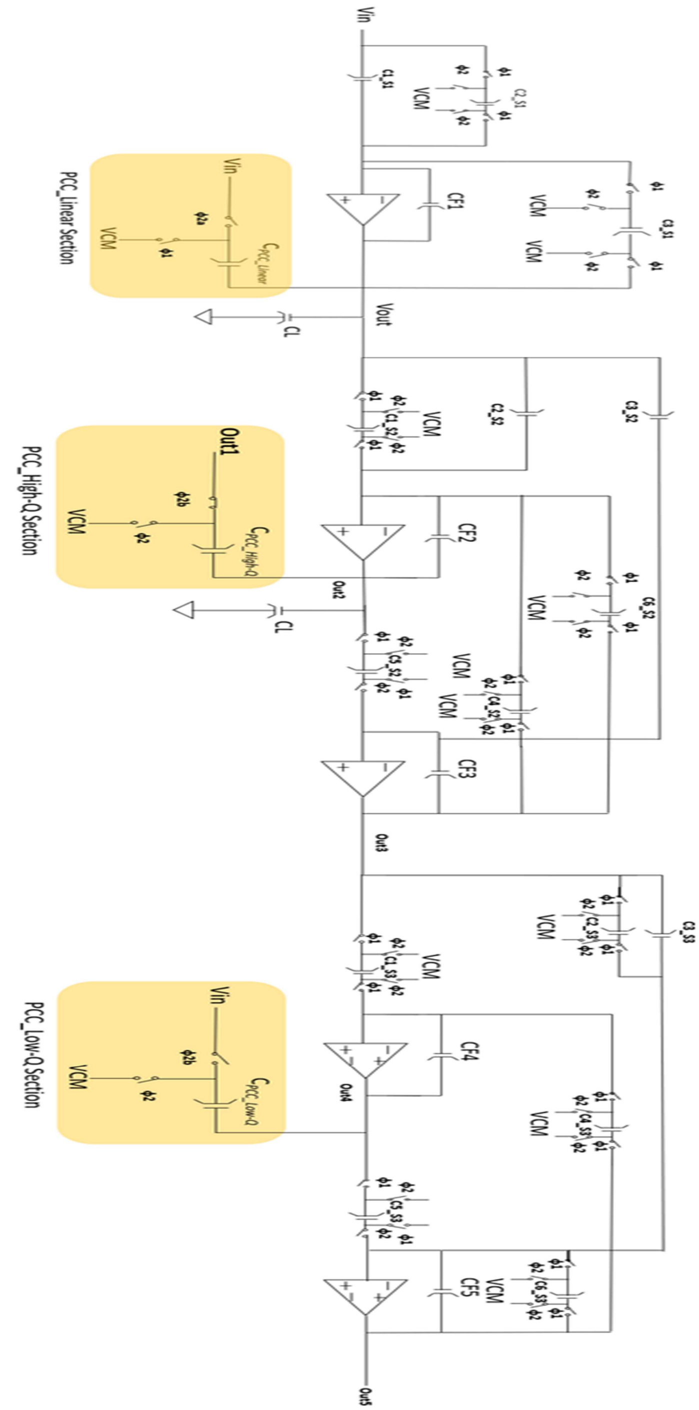


Figure 3.38. Completed work with proposed PCC technique.

3.3.2 Simulation Result

Among the three stages, the linear stage had the highest requirement for slew rate (SR), so the SR requirements of this stage are discussed. The output harmonics with different SR when the input signal frequency was 30 KHz with V_{pp} sine signal, ± 1.8 vol, are shown in Figure 3.39, and the comparison of the transient response is shown in Figure 3.40. Table XVIII lists the amplitude values of the 3rd and 5th harmonics at the conventional LPF circuit. The PCC result shoed that the noise improvement (12.34 dB) noise improvement at the 3rd harmonic distortion was reduced from -105.79dBc to -118.13dBc & improve 13.15dB at 5th harmonic distortion was reduced from -115.25dBc to -128.40dBc.

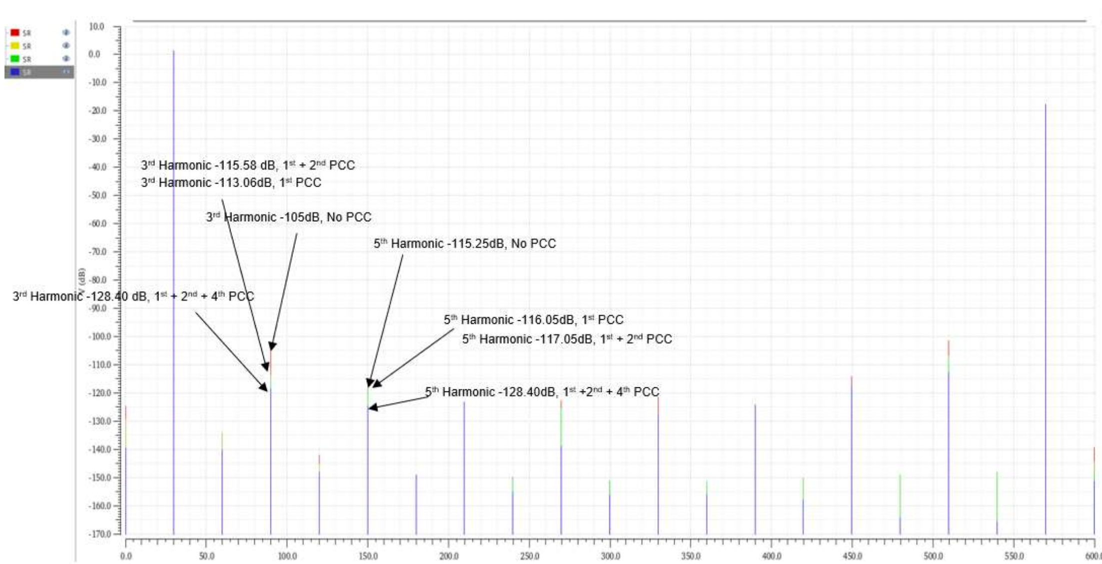


Figure 3.39. Distortions of the conventional LPF.

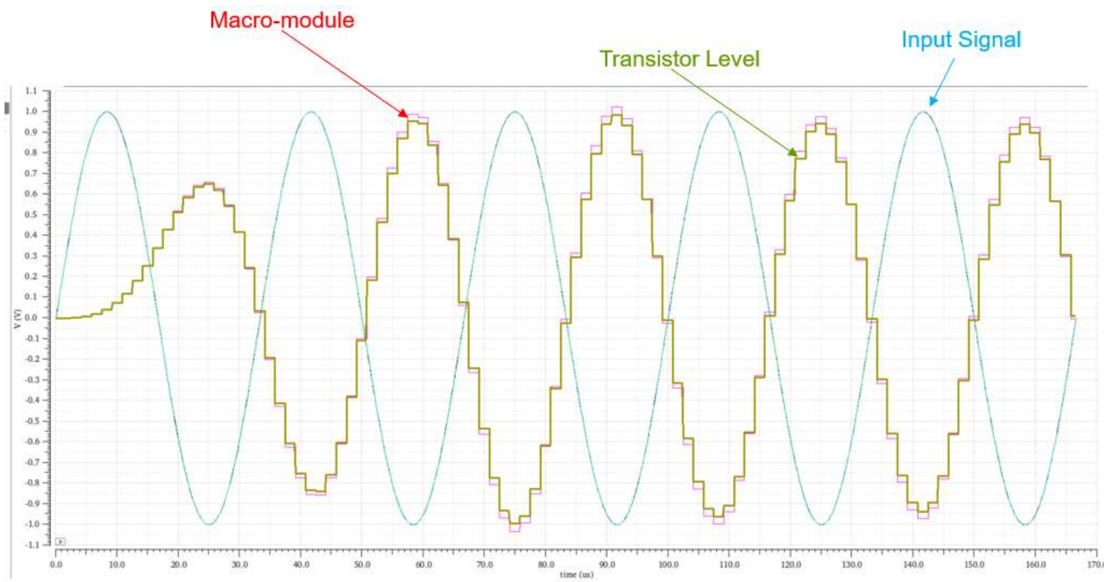


Figure 3.40. LPF transient time responses of the ideal and transistor levels.

Table XVIII. Simulation results for the macro-module and transistor level.

Macro-module

Harmonic Amplitude (Unit: dB)	3rd Harmonic	5th Harmonic
5th order macro-module without CCT	-110.25	-121.42
5th order macro-module with 1st CCT	-117.40	-118.15
5th order macro-module with 1/2 CCT	-119.71	-125.9
5th order macro-module with 1/2/4 CCT	-125.12	-131.78

Transistor level

Harmonic Amplitude(Unit: dB)	3rd Harmonic	5th Harmonic
5th order transistor level without CCT	-105.79	-115.25
5th order transistor level with 1st CCT	-113.06	-116.05
5th order transistor level with 1/2 CCT	-112.58	-117.05
5th order transistor level with 1/2/4 CCT	-118.13	-128.40

The dynamic powers at each SC circuit and the individual PCC techniques are given in Table XIX, whereas the comparison between with and without the PCC techniques is given in Table XX. The corresponding dynamic power consumption with the PCC is 0.363 uW as low as expected.

Table XIX. Simulation results of dynamic power at the transistor level circuit.

		Bilinear Section Caps value table			
		DR Unit (fF)	Area	Dynamic Power	Vrms (Vol)
	CF1	825	825		
	C1_S1	100	100 (Unit Cap)		
SC	C2_S1	200	200	1.07E-07	0.943
SC	C3_S1	200	200	1.02E-07	0.922
SC	C_PCC_Linear	264		1.89E-07	1.093

		High-Q Section Caps value table			
		DR Unit (fF)	Area	Dynamic Power	Vrms (Vol)
	CF2	463	100 (Unit Cap)		
SC	C1_S2	133	348	6.92E-08	0.931
	C2_S2	0	363		
	C3_S2	100	100 (Unit Cap)		
SC	C4_S2	184.5	0	1.00E-07	0.952
	CF3	374	97		
SC	C5_S2	141	179	6.90E-08	0.903
	C6_S2	103	100 (Unit Cap)		
SC	C_PCC_HighQ	186.6		1.34E-07	1.093

		High-Q Section Caps value table			
		DR Unit (fF)	Area	Dynamic Power	Vrms (Vol)
	CF4	530.5	530		
SC	C1_S3	141.7	141	8.13E-08	0.978
SC	C2_S3	0	0	0.00E+00	0.976
SC	C3_S2	100	100 (Unit Cap)	5.13E-08	0.925
SC	C4_S2	100	100 (Unit Cap)	5.65E-08	0.97
	CF5	500	500		
	C5_S3	302.5	302		
SC	C6_S2	169	169	9.23E-08	0.954
SC	C_PCC_HighQ	56		4.01E-08	1.093

Table XX. Simulation results of the dynamic power comparison.

	Unit(uW)
Grand Total of Dynamic Power w/o PCC (5 OTAs excluded)	0.728
Grand total of Dynamic Power w/ PCC (5 OTAs excluded)	1.091
Dynamic Power_Linear_PCC	0.189
Dynamic Power_HighQ_PCC	0.134
Dynamic Power_lowQ_PCC	0.04

Chapter 4: Conclusion and Future Work

Designing analog LPF circuits with good linearity, low cost, and area efficiency is critical for the present day mixed-signal designs. In this study, we designed a fully-differential 5th-order elliptic low-pass SC filter with a sampling frequency of 600 kHz, a corner frequency of 36 kHz using the PCC technique. This technique improved the linearity of the LPF with a noise reduction of 12.3 dB at the 3rd harmonic distortion and 13.1 dB at the 5th harmonic distortion without using the active blocks. This indicates that the PCC technique is an efficient approach for designing LPFs.

The future research will focus on the utilization of the PCC technique for designing FIR filters with low power consumption.

BIBLIOGRAPHY

[1] Y. Zhao, P. Mak, M. Law and R. P. Martins, "A 0.127-mm^2, 5.6-mW,5th-order SC LPF with +23.5-dBm IIP3 and 1.5-to-15-MHz clockdefined bandwidth in 65-nm CMOS," 2013 IEEE Asian Solid-State Circuits Conference (A-SSCC), Singapore, 2013, pp. 361-364.

[2] M. H. Naderi, S. Prakash and J. Silva-Martinez, "Operational Transconductance Amplifier With Class-B Slew-Rate Boosting for Fast High-Performance Switched-Capacitor Circuits," in IEEE Transactions on Circuits and Systems I: Regular Papers, vol. 65, no. 1, pp. 3769-3779, Nov. 2018

[3] Kareppagoudr, M.; Caceres, E.; Kuo, Y.; Shakya, J.; Wang, Y.; Temes, G.C. Passive slew rate enhancement technique for Switched-Capacitor Circuits. In Proceedings of the 2019 IEEE 62nd International Midwest Symposium on Circuits and Systems (MWSCAS), Dallas, TX, USA, 4–7 August 2019; pp. 913–916.

[4] Kerry Lacanette, "A Basic Introduction To Filter Devices-Active, Passive And Switched-Capacitor", National Semiconductor Application Note 779, April 1991.

[5] John G. Prokis. Dimitris G. Manolakis. Digital Signal Processing. Fourth Edition. 2007

[6] David A.Johns and Ken Martin, "Analog Integrated Circuit Design", John Wiley & Sons, Inc., 1997.

[7] Behzad Razavi, "Design of Analog CMOS Integrated Circuits", Edition 2002.

[8] Dr. Gabor Temes, Oregon State University, ECE626 lecture notes, Winter 2011.

[9] Temes, G.C., and J. Silva. "Simple and Efficient Noise Estimation Algorithm." Electronics Letters, vol. 40, no.11, 2004, p. 640., doi:10.1049/el:20040431.

[10] M. H. Naderi, S. Prakash and J. Silva-Martinez, "Operational Transconductance Amplifier With Class-B Slew-Rate Boosting for Fast High-Performance Switched-Capacitor Circuits," in IEEE Transactions on Circuits and Systems I: Regular Papers, vol. 65, no. 11, pp. 3769-3779, Nov. 2018

[11] Y. Zhao, P. Mak, M. Law and R. P. Martins, "A 0.127-mm2 , 5.6-mW, 5th-order SC LPF with +23.5-dBm IIP3 and 1.5-to-15-MHz clockdefined bandwidth in 65-nm CMOS," 2013 IEEE Asian Solid-State Circuits Conference (A-SSCC), Singapore, 2013, pp. 361-364.

APPENDIX I

Matlab script for the design of a 5th order elliptic LPF

```
% 626 filter - using James' values...



clear all;
%% Input specs
Fs = 600e3;              % 600kHz
fpass = 36e3;           % 36kHz
% fstop = [72e3 240e3];   % Hz   % why this is a range???..I think
that it is because at Fs/2 the filter goes to -inf and the ripple is
unbounded

fstop = 72e3 ;              % 72kHz
ripple_passband = 0.15 ;    % dB    -- peak to peak ripple inside
passband
gain_stopband = 51 ; %was 51       % dB    -- max gain in passband
minus max gain in stopband


%% STEP 1: Choosing lowest order implementation:
%{
%  { I will compare the different filters to see which has lowerst
order:
% filter_dig =  designfilt('lowpassiir','SampleRate', Fs);

% Normalize specs so they can be fed to matlab functions that give
order
% and cut-off (normalized) freq of each of the digital IIR filters:
Wp = fpass/(Fs/2);
Ws = fstop/(Fs/2);
Rp = ripple_pass;
Rs = ripple_stop;

% butterworth:
[n(1) , Wcut_off(1)] = buttord(Wp,Ws,Rp,Rs);
% chebyshev I:
[n(2) , Wcut_off(2)] = cheb1ord(Wp,Ws,Rp,Rs);
% chebyshev II:
[n(3) , Wcut_off(3)] = cheb2ord(Wp,Ws,Rp,Rs);
% elliptic:
[n(4) , Wcut_off(4)] = ellipord(Wp,Ws,Rp,Rs);

% denormalize the cut off freq
f_3db = Wcut_off * (Fs/2);
```

```
% CONCLUSION: the lowest order filter is achieved with an ELLIPTIC
%}
%% STEP 2: Get filter coefficients:

% Get order and cut-off:
[order , fcutoff_norm] =
ellipord(fpass/(Fs/2),fstop/(Fs/2),ripple_passband,gain_stopband);

% denormalize cutoff freq:
fcutoff = fcutoff_norm*(Fs/2);

% Get elliptic filter coeffs:
[Hdig.n , Hdig.d] = ellip(order , ripple_passband , gain_stopband ,
fcutoff_norm);

% define transfer function:
Hdig.tf = tf(Hdig.n , Hdig.d , 1/Fs);

%% STEP 3: verify that specs are matched using this TF:
% make 3 freq vector, gral, passband and stopband:
f = linspace(0 , Fs/2 , 1e3+1);
f = f(1:end-1);   % because at Fs/2 it goes to -Inf
f_pass = linspace(0 , fpass , 100e3);
f_pass2 = linspace(0 , fpass*1.2 , 100e3);   % Actually for the
passband freq vector I consider 20% more to have the peak well
inside the range
f_stop = linspace(fstop, 240e3 , 1e3);  % I use 240kHz because it is
on the specs!

[Hdig.mag , Hdig.ph] = bode(Hdig.tf, 2*pi*f)     ;  Hdig.mag =
squeeze(Hdig.mag);Hdig.ph = squeeze(Hdig.ph);
[Hdig.mag_pass , ~]  = bode(Hdig.tf, 2*pi*f_pass);  Hdig.mag_pass =
squeeze( Hdig.mag_pass );
[Hdig.mag_stop , ~]  = bode(Hdig.tf, 2*pi*f_stop);  Hdig.mag_stop =
squeeze( Hdig.mag_stop );

if ( max(20*log10(abs(Hdig.mag_pass)))-
min(20*log10(abs(Hdig.mag_pass)))  > ripple_passband )
    error("The passband ripple of the realized filter is larger than
the specified!");
end
%if ( max(20*log10(abs(Hdig.mag_stop)))-
min(20*log10(abs(Hdig.mag_stop)))  < gain_stopband )
 %   error("The stopband ripple of the realized filter is larger
than the specs!");
%end

%% STEP 4: split in second order sections:

% generates a matrix with 6 columns and the rows are 2nd order (or
1st
% order if the input TF is odd-order), and a gain.
% Each row has the format [num,den]=[n1 n2 n3]/[1 d2 d3 ]
%
```

```
% use 'down', to order the sections so the first row of sos contains
the poles
% closest to the unit circle  (highest Q first, lowest Q last).
% (according to the book there are other advantages/disadvatanges
% resulting from combining poles and zeros in other ways, but I
% experimented with them them, and this one seemed easier).

[ SecondOrdSections , gain_sos ] = tf2sos( Hdig.n , Hdig.d , 'down'
);

% separate the sections
% assign the gain_sos to the linear section arbitrarily,because
filters
% will be scaled later on.

[HiQ.n , HiQ.d] = sos2tf(SecondOrdSections(1,:))          ;
[LoQ.n , LoQ.d] = sos2tf(SecondOrdSections(2,:))          ;
[Lin.n , Lin.d] = sos2tf(SecondOrdSections(3,:),gain_sos);

% for the linear section drop the zero coefficients
Lin.n = Lin.n(1:end-1); Lin.d = Lin.d(1:end-1);

% generate the TF for each section

HiQ.tf = tf( HiQ.n , HiQ.d , 1/Fs );
LoQ.tf = tf( LoQ.n , LoQ.d , 1/Fs );
Lin.tf = tf( Lin.n , Lin.d , 1/Fs );

[HiQ.mag , HiQ.ph] = bode( HiQ.tf , 2*pi*f ) ; HiQ.mag =
squeeze(HiQ.mag) ; HiQ.ph = squeeze(HiQ.ph);
[LoQ.mag , LoQ.ph] = bode( LoQ.tf , 2*pi*f ) ; LoQ.mag =
squeeze(LoQ.mag) ; LoQ.ph = squeeze(LoQ.ph);
[Lin.mag , Lin.ph] = bode( Lin.tf , 2*pi*f ) ; Lin.mag =
squeeze(Lin.mag) ; Lin.ph = squeeze(Lin.ph);


%% STEP 5: Plot filter bode and zero-pole diagram:
plot_step_5 = 1;

if plot_step_5

    %plot bode mag
    figure();
    subplot(221);
    semilogx(f,dbv(Hdig.mag)); hold on
    semilogx(f,dbv(HiQ.mag),'r');
    semilogx(f,dbv(LoQ.mag),'g');
    semilogx(f,dbv(Lin.mag),'m');

    grid on;
    ylabel('Magnitude [dB]');
    xlabel('Freq [Hz]')
    title("Bode Plot")
```

```
    %plot bode phase
    subplot(223);
    semilogx(f,Hdig.ph); hold on;
    semilogx(f,HiQ.ph,'r');
    semilogx(f,LoQ.ph,'g');
    semilogx(f,Lin.ph,'m');

    grid on;
    ylabel('Phase [^o]');
    xlabel('Freq [Hz]')

    %plot pole-zero map
    subplot(2,2,[2 4]);
    pzmap(Hdig.tf); hold on
    pzmap(HiQ.tf,'r');
    pzmap(LoQ.tf,'g');
    pzmap(Lin.tf,'m');
    [p,z] = pzmap(Hdig.tf);% grid on;

    legend('Overall','High Q','Low Q','Linear');
end

%% STEP 6: Order of the sections and cap values

% % % % % % % % % % % % % % % % % % % % % % % % % % % % % % % % % %
% % % %
% Order of the sections:
%   IN ---> [Linear]---> [HiQ]---> [LoQ]---> OUT

% % % % % % % % % % % % % % % % % % % % % % % % % % % % % % % % % %
% % % %
% Normalized & un-scaled Cap values:
%
% I use a function that gets caps values from coefficients.
% These are unscaled.
% (The number of the caps follows that in John-Martins)
Lin.c = coef2caps(Lin.tf , "Linear");
HiQ.c = coef2caps(HiQ.tf , "HighQ");
LoQ.c = coef2caps(LoQ.tf , "LowQ");


%% Comments about dynamic range scaling from 626 notes:
%{
    in 626 notes  (Lec: advanced SC circ design techniques)an
argument is given
    to support why scaling a stage is convenient.
    Briefly a constant k is included in a stage, which increases the
output
    swing of the amplifier, but reduces the output current (e.g.
input branches
    capacitance is increased, and utput branches are reduced) that
goes to
    the next stage.
    From the current signal perspective nothing changes.
```

However from the current noise perspective, increasing k reduces the
output noise current, until an asymptote is reached at k=inf.

There is a limit to this, because more output swing will saturate the
amplifier eventually. Degrading the SNDR.
Therefore the optimal value for the constant k will make the amplifier
swing large enough (optimize SNR) but right before hitting
distortion (optimize SNDR).
That is why it is called "Optimal DR scaling".

That means that more k reduces the current noise that will be injected
into the next stage.
Note that this effect is due to the opamp noise (and assumes infinite
gain).
Also in the notes he mentions that SNR is improved, but then adds that
output noise depends on other factors as well (doesnt say which ones)
and the current might be only one of them (so stating that SNR improves
might be not be totally right??)

It would be interesting to see:
- the contribution of this noise when input referred
- this noise compared to other noise sources to see its relevance.
- see if this can be tested, in terms of SNR vs k.

Note that:
- This has nothing to do with kT/C noise.
- This is related with amplifier noise, since a noisy-er amp will inject
more noise.
- Note that the observtion is done on an infinite gain amp.

In slides 39 to 46, a specific scaling for SC is described.
The assumptions seem to be that (slide 39):
- all amplifiers have equal input noise,
- all have same maximum linear range.

It would be interesting to see what happens if that is not true.

I think that slide 41 has an error, because if both h and g are
multiplied by k the opamp output voltage doesnt change.

- The area scaling consists in using the smallest coefficient as the
unit capacitance value for the design.

```
    -There is no mention on how the rounding affects.

    Mentions that the coefficients to mulyiply are
k=Vin,max/Vout.max.
    That all opamps should saturate with the same input.
    - The sampling caps are increased by k, this pushes more charge
into the
    integrating cap. But if the integrating cap is also increased
less
    - The integrating cap is reduced by the same factor

%}
%% STEP 7: Dynamic range scaling

%{
  For the scaling I need to see the internal realization of the
filters.
  Using the cap values obtained in STEP 6, and the flow diagrams in
  John-Martins it is possible to get them in simulink.

  Also in simulink it is possible to use an app called "Linear
Analysis
  Tool" (see help). This allows to extracts transfer functions form
a block diagram.

  By doing that I can extract the transfer functions at the output
of
  each integrator and scale them, before moving to cadence.

 This was done in simulink using the Linear Analysis tool. The
simulink
 model is
open("filter_626_simu.slx");
 And the Linear Analysis tool with the pre and post freq response
can be
 found in:
load("prescaling_freq_repsonses.mat");
%}


% It can also be done more automatic...

% define simulink the model name
model = 'filter_626_simu';

% open the simulink model
% open_system(model)

% set the scaling vector to 1 initially
scale = ones(1,5);


% define inputs and outputs to linearize, based on the output blocks
% defined in simulink
```

```
input     = linio('filter_626_simu/IN',1,'input');
output(1) = linio('filter_626_simu/S1_Linear/Int',1,'output');
output(2) = linio('filter_626_simu/S2_HighQ/Int1',1,'output');
output(3) = linio('filter_626_simu/S2_HighQ/Int2',1,'output');
output(4) = linio('filter_626_simu/S3_LowQ/Int1' ,1,'output');
output(5) = linio('filter_626_simu/S3_LowQ/Int2' ,1,'output');

for m =1 :length(scale)

    % runs the simulink model and extracts the TF in the indicated
    % input/output:
    linsys = linearize(model , [input output(m)]);
    % get the num and den
    [num , den ]  = ss2tf(linsys.A , linsys.B , linsys.C , linsys.D
);

    % get the mag response in the pass band
    tf_int = tf( num , den ,1/Fs);
    [mag , ~] = bode( tf_int , 2*pi*f_pass2 ) ; mag = squeeze(mag) ;

    % get the max magnitude
    [max_mag_db , max_index] = max(20*log10(abs(mag)));

    scale(m) =  10.^( max_mag_db-20*log10(abs(scale(m))));
%     fprintf("pause here -- for debugging\n");

end


% Get the values of the scaled coefficients and the resulting
transfer
% functions by inspection of the simulink model:

Lin.cs_dr = [Lin.c(1) Lin.c(2) Lin.c(3)*scale(1)];
Lin.cf_dr = [scale(1)];
[ax1,ax2] = caps2coef(Lin.cs_dr./Lin.cf_dr , "Linear");
Lin.c_dr_tf = tf(ax1,ax2,1/Fs);

HiQ.cs_dr = [HiQ.c(1)*scale(1) HiQ.c(2)*scale(1) HiQ.c(3)*scale(1)
...
             HiQ.c(4)*scale(3) HiQ.c(5)*scale(2) HiQ.c(6)*scale(3)];
HiQ.cf_dr = [scale(2) scale(3)];

[ax1,ax2] = caps2coef([HiQ.cs_dr(1)/HiQ.cf_dr(1)
HiQ.cs_dr(2)/HiQ.cf_dr(1) ...
                       HiQ.cs_dr(3)/HiQ.cf_dr(2)
HiQ.cs_dr(4)/HiQ.cf_dr(1) ...
                       HiQ.cs_dr(5)/HiQ.cf_dr(2)
HiQ.cs_dr(6)/HiQ.cf_dr(1) ] ,"HighQ");
HiQ.c_dr_tf = tf(ax1,ax2,1/Fs);

LoQ.cs_dr = [LoQ.c(1)*scale(3) LoQ.c(2)*scale(3) LoQ.c(3)*scale(3)
...
            LoQ.c(4)*scale(5) LoQ.c(5)*scale(4) LoQ.c(6)*scale(5) ];
```

```
LoQ.cf_dr = [scale(4) scale(5)];

[ax1,ax2] = caps2coef([LoQ.cs_dr(1)/LoQ.cf_dr(1)
LoQ.cs_dr(2)/LoQ.cf_dr(2) ...
                       LoQ.cs_dr(3)/LoQ.cf_dr(2)
LoQ.cs_dr(4)/LoQ.cf_dr(1) ...
                       LoQ.cs_dr(5)/LoQ.cf_dr(2)
LoQ.cs_dr(6)/LoQ.cf_dr(2) ] ,"LowQ");
LoQ.c_dr_tf = tf(ax1,ax2,1/Fs);



% merge coefficients into one matrix with 3 columns and 6 rows (each
% columns is one section of the filter and each row is each cap
value
Cap_s_dr = [[Lin.cs_dr zeros(1,3)]' HiQ.cs_dr' LoQ.cs_dr'];

% replace small coefficients smaller than 1e-10 with 0
Cap_s_dr( abs(Cap_s_dr) <= 1e-10 )= 0;

Cap_f_dr = [Lin.cf_dr HiQ.cf_dr LoQ.cf_dr];

% display scaled coefficients as columns from cap1 to cap6
disp(' After dynamic range is completed...')
disp('             S1:Linear     S2:High Q         S3:Low Q')
disp([ ["c_s1";"c_s2";"c_s3";"c_s4";"c_s5";"c_s6"]  Cap_s_dr])
disp([ ["c_f1";"c_f2";"c_f3";"c_f4";"c_f5"]  Cap_f_dr']);



%% STEP 8: Area scaling
% assume some minimum capacitor size (should be AKM's!)
% this assumes a minum capacitor of value 1...or that all caps are
% normalized to a unit cap of value 1.
C_min = 1;

% this describes which caps belong to each opamp in the filter (from
1 to 5)
ota_array = [ [1 1 1 0 0 0]' [2 2 3 2 3 2]' [4 5 5 4 5 5]' ];


% duplicate the coefficients matrix and replace the 0 for Inf so
they dont
% show up in the min search
Cap_s_dr_area = Cap_s_dr;
Cap_s_dr_area( Cap_s_dr == 0 )= NaN;

% do this for each opamp
for m = 1:5
    %finds the smallest absolute value of the caps connected to a
certain
    %opamp:
    min_cap = min(abs(Cap_s_dr_area( ota_array== m )));
    %then divides a
```

```
    Cap_s_dr_area = Cap_s_dr_area .* (ota_array== m )*C_min/min_cap
+ Cap_s_dr_area.* (ota_array~= m );
    Cap_f_dr_area(m) = Cap_f_dr(m) * C_min/min_cap;

end
Cap_s_dr_area( Cap_s_dr == 0)= 0;

% display scaled coefficients as columns from cap1 to cap6
fprintf(' After area scaling is completed - Assuming Cunit = %f
F\n',C_min)
disp('             S1:Linear     S2:High Q         S3:Low Q')
disp([ ["c_s1";"c_s2";"c_s3";"c_s4";"c_s5";"c_s6"]
Cap_s_dr_area(:,1) Cap_s_dr_area(:,2) Cap_s_dr_area(:,3)])
disp([ ["c_f1";"c_f2";"c_f3";"c_f4";"c_f5"]  Cap_f_dr_area']);


%% STEP 9: Getting layout-realizable caps
% After all the scaled cap values are not integers.
% We want to see the resulting filter frequency response if we round
them.
% It uses 92 unit caps (considering a fully diff implementation).
% Seems that the rounded version doesn't match well against the
ideal
% filter. The poles




% without any rounding :
[ id.all , id.s1 , id.s2 , id.s3 ] = cap_to_tf(Cap_s_dr_area ,
Cap_f_dr_area , ota_array , Fs);
verify_filter_specs(id.all , f_pass,f_stop, ripple_passband,
gain_stopband);

% with rounding:
[ rn.all , rn.s1 , rn.s2 , rn.s3 ] = cap_to_tf(round(Cap_s_dr_area)
, round(Cap_f_dr_area) , ota_array , Fs);
total_cap_round = (sum(abs(round(Cap_s_dr_area)),'all') +
sum(abs(round(Cap_f_dr_area)),'all'))*2;



within_spec = verify_filter_specs(rn.all , f_pass,f_stop,
ripple_passband, gain_stopband);
if within_spec
    disp("The rounded version is within specs.");
else
    disp("The rounded version is NOT within specs. Something needs
to be done!");
end

plot_step_9 = 0;

if plot_step_9
    % to get an idea of where the problem is coming from we plot
bode and
```

```
    % also poles and zeros
    figure();bode(Hdig.tf,'--gsq',  id.all ,'mo--',rn.all,'c');
    title(sprintf('Bode Diagram - %2.f C_{unit} required',
total_cap_round  ))
    legend('initial ideal','scaled ideal','round');

    hold on;
    bode(id.s1,'ro--', rn.s1,'b',id.s2,'ro--', rn.s2,'b',id.s3,'ro--
', rn.s3,'b');


    figure();
    subplot(221);
    pzmap(id.s1,'b'); hold on
    pzmap(rn.s1,'r');
    title('Linear');

    subplot(222)
    pzmap(id.s2,'b'); hold on
    pzmap(rn.s2,'r');
    title('High Q');

    subplot(223)
    pzmap(id.s3,'b'); hold on
    pzmap(rn.s3,'r');
    title('Low Q');

    subplot(224)
    pzmap(id.all,'b'); hold on
    pzmap(rn.all,'r');
    title('All');
    legend('Ideal','Rounded');


end
%{
%% STEP 9b: rounding the caps yields a filter that is off-
specs...need to solve this
% If within_spec is 0 then the rounded filter does not match the
specs, so
% something needs to be done in order to get a filter within specs.


% One option would be to look where poles and zeros are located in
the
% rounded version the ideal one. Also this can be done for each
section
% separately.

[id.s1_p , id.s1_z] = pzmap(id.s1);
[id.s2_p , id.s2_z] = pzmap(id.s2);
[id.s3_p , id.s3_z] = pzmap(id.s3);

[rn.s1_p , rn.s1_z] = pzmap(rn.s1);
```

```
[rn.s2_p , rn.s2_z] = pzmap(rn.s2);
[rn.s3_p , rn.s3_z] = pzmap(rn.s3);

d.s1_p = abs(rn.s1_p   -id.s1_p   );
d.s1_z = abs(rn.s1_z   -id.s1_z   );
d.s2_p = abs(rn.s2_p(1)-id.s2_p(1));
d.s2_z = abs(rn.s2_z(1)-id.s2_z(1));
d.s3_p = abs(rn.s3_p(1)-id.s3_p(1));
d.s3_z = abs(rn.s3_z(1)-id.s3_z(1));


% I want to see how the poles and zeros move when I choose an
integer value
% for the feedback cap and round the corresponding sampling cap.
% Intuitively if the feedback cap is larger (and integer) the
rounded
% sampling cap is larger too, and they ratio matches the ideal
coefficient
% better.

% These will hold the final result
Cap_s_round = Cap_s_dr_area;
Cap_f_round = Cap_f_dr_area;

cap_norm = Cap_s_dr_area.* (ota_array== 1 )./Cap_f_dr_area(1) + ...
           Cap_s_dr_area.* (ota_array== 2 )./Cap_f_dr_area(2) + ...
           Cap_s_dr_area.* (ota_array== 3 )./Cap_f_dr_area(3) + ...
           Cap_s_dr_area.* (ota_array== 4 )./Cap_f_dr_area(4) + ...
           Cap_s_dr_area.* (ota_array== 5 )./Cap_f_dr_area(5);

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%
%||||||||||||||||||||||||||||||SECTION
2||||||||||||||||||||||||||||||%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%
% Sweep the integrating caps in section 2 (both int2 and int 3) to
see how the poles and zeros
% distance to ideal location is affected :
% The sweeping range is given by the following variables:


% The search is done in this commented section:
%%{
target_distance_s2 = .003;
Cf2_vector = 1:30;
Cf3_vector = 1:30;
for m = 1 :length(Cf2_vector)
    % this is the x-variable  (columns)
    % round the feedback cap and then increase it
    Cap_f_round(2) = Cf2_vector(m);

    for n = 1:length(Cf3_vector)
        % this is the y-variable (rows)
        Cap_f_round(3) = Cf3_vector(n);
```

```
        % calculate the sampling cap associated with those
integrators and
        % round the caps. All other caps remain unchanged
        Cap_s_round = round(cap_norm*Cap_f_round(2))  .*
(ota_array== 2 ) +... % calculates and round the caps realted to 2nd
integrator only
                      round(cap_norm*Cap_f_round(3))  .*
(ota_array== 3 ) +... % calculates and round the caps realted to 3rd
integrator only
                      Cap_s_dr_area .* ~((ota_array== 2) |
(ota_array== 3));   % the remaining caps remain unchanged

        % round all caps(the recently modified ones shouldn't
change) and calcute the transfer function
        [ ~, ~, sta2 , ~] = cap_to_tf(round(Cap_s_round) ,
round(Cap_f_round) , ota_array , Fs);

        % calculate the total number of unit caps used:
        total_caps(n,m) =
(sum(abs(round(Cap_s_round)),'all')+sum(abs(round(Cap_f_round)),'all
'))*2;

        % calculate the pole and zero for this section:
        [poles , zeros] = pzmap(sta2);

        % there should be only 2 complex poles/zeros, and we want to
keep the ones
        % located in the first quadrant (because we are using that
one to
        % compare)
        if ( (~isempty(poles) && ~isempty(zeros)) &&
(length(poles)==2) && (length(zeros)==2) && ~isreal(poles) &&
~isreal(zeros))
            if imag(poles(1))>0
                %calc distance for pole
                d_pole(n,m) = abs(poles(1)-id.s2_p(1));
            else
%                 error("The chosen pole wasn't in the 1st
quadrant.")
                d_pole(n,m) = abs(poles(2)-id.s2_p(2));
            end
            if imag(zeros(1))>0
                %calc distance for pole
                d_zero(n,m) = abs(zeros(1)-id.s2_z(1));
            else
%                 error("The chosen zero wasn't in the 1st
quadrant.")
                d_zero(n,m) = abs(zeros(2)-id.s2_z(2));
            end

        else
            d_pole(n,m) = NaN;
            d_zero(n,m) = NaN;
```

```
        end

    end
end

plot_step_9b_s2 = 0;  % plots the 3d plots

if plot_step_9b_s2
    figure();
    surf( Cf2_vector , Cf3_vector , d_pole );colorbar;view(2)
    xlabel("OTA 2");ylabel("OTA 3");title("Distance to ideal pole")
%     zlim([0 d.s2_p/10])
    zlim([0 target_distance_s2])


    figure();
    surf( Cf2_vector , Cf3_vector , d_zero );colorbar;view(2)
    xlabel("OTA 2");ylabel("OTA 3");title("Distance to ideal zero")
%     zlim([0 d.s2_z/10])
    zlim([0 target_distance_s2])
end


%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%
% I will choose those points that for both pole and zero are 1/10
smaller
% than in the original rounded  solution.

% valid_sol_crit1 = (d_pole <= d.s2_p/10) & (d_zero <= d.s2_z/10);
valid_sol_crit1 = (d_pole <= target_distance_s2) & (d_zero <=
target_distance_s2);

% and from those the one with smallest total number of caps
total_caps_reduced = total_caps;
total_caps_reduced(~valid_sol_crit1)= Inf;  % if is not valid then
assign infinite area

% express the solution space only considering the total area
for k=1:length(Cf2_vector)*length(Cf3_vector)
    total_caps_reduced_lin(k) = total_caps_reduced(k);
end
figure();
plot(total_caps_reduced_lin,'ok','Linewidth',3);hold on;grid on
xlabel('index');
ylabel('Total area [C_U]');
title('Section 2: Area for solution space');
[min_area , index] = min(total_caps_reduced_lin);
[cf_3,cf_2] = ind2sub(size(total_caps_reduced) ,index);

% sanity check to see if the obtained points make sense:
[d_pole(cf_3,cf_2) , d.s2_p ; d_zero(cf_3,cf_2) d.s2_z ;
total_caps_reduced(cf_3,cf_2) total_cap_round]
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%
```

```
% Get the matrix with the chosen value:
Cap_f_round(2) = cf_2;
Cap_f_round(3) = cf_3;


% These are the chosen caps
% Cap_f_round(2) = 11;
% Cap_f_round(3) = 7;


Cap_s_round_temp = round(cap_norm*Cap_f_round(2))  .* (ota_array== 2
) +... % calculates and round the caps realted to 2nd integrator
only
              round(cap_norm*Cap_f_round(3))  .* (ota_array== 3 )
+... % calculates and round the caps realted to 3rd integrator only
              Cap_s_dr_area .* ~((ota_array== 2) | (ota_array== 3));
% the remaining caps remain unchanged

Cap_f_round_temp = Cap_f_round;

% round all caps(the recently modified ones shouldn't change) and
calcute the transfer function
[ ~, ~, sta2 , ~] = cap_to_tf(round(Cap_s_round_temp) ,
round(Cap_f_round_temp) , ota_array , Fs);

figure();pzmap(sta2 , 'r', rn.s2 ,'k' ,id.s2,'b');
legend('new','round','ideal')
title('Section 2');

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%
%||||||||||||||||||||||||||||||||SECTION
3||||||||||||||||||||||||||||||||%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%
% Sweep the integrating caps in section 3 (both int4 and int 5) to
see how
% the poles and zerosdistance to ideal location is affected :
% The sweeping range is given by the following variables:

% The search is done in this commented section:
%%{

target_distance_s3 = .003;
Cf4_vector = 1:60;
Cf5_vector = 1:60;
for m = 1 :length(Cf4_vector)

    % x-axis/columns
    Cap_f_round(4) = Cf4_vector(m);

    for n = 1:length(Cf5_vector)
        % y-axis/rows
        Cap_f_round(5) = Cf5_vector(n);
```

```
        % calculate the sampling cap associated with those
integrators and
        % round the caps. All other caps remain unchanged
        Cap_s_round = round(cap_norm*Cap_f_round(4))  .*
(ota_array== 4 ) +... % calculates and round the caps realted to 4nd
integrator only
                      round(cap_norm*Cap_f_round(5))  .*
(ota_array== 5 ) +... % calculates and round the caps realted to 5rd
integrator only
                      Cap_s_round_temp .* ~((ota_array== 4) |
(ota_array== 5));   % the remaining caps (after stage 2 remain
unchanged

        % round all caps(the recently modified ones shouldn't
change) and calcute the transfer function
        [ ~, ~, ~ , sta3] = cap_to_tf(round(Cap_s_round) ,
round(Cap_f_round) , ota_array , Fs);

        % calculate the total number of unit caps used:
        total_caps(n,m) =
(sum(abs(round(Cap_s_round)),'all')+sum(abs(round(Cap_f_round)),'all
'))*2;

        % calculate the pole and zero for this section:
        [poles , zeros] = pzmap(sta3);

        % there should be only 2 poles/zeros, and we want to keep
the ones
        % located in the first quadrant (because we are using that
one to
        % compare)
        if ( (~isempty(poles) && ~isempty(zeros)) &&
(length(poles)==2) && (length(zeros)==2) && ~isreal(poles) &&
~isreal(zeros))
            if imag(poles(1))>0
                %calc distance for pole
                d_pole(n,m) = abs(poles(1)-id.s3_p(1));
            else
%                 error("The chosen pole wasn't in the 1st
quadrant.")
                d_pole(n,m) = abs(poles(2)-id.s3_p(2));
            end
            if imag(zeros(1))>0
                %calc distance for pole
                d_zero(n,m) = abs(zeros(1)-id.s3_z(1));
            else
%                 error("The chosen zero wasn't in the 1st
quadrant.")
                d_zero(n,m) = abs(zeros(2)-id.s3_z(2));
            end

        else
            d_pole(n,m) = NaN;
```

```
            d_zero(n,m) = NaN;
        end

    end
end

plot_step_9b_s3 = 0;  % plots the 3d plots

if plot_step_9b_s3
    figure();
    surf( Cf4_vector , Cf5_vector , d_pole );colorbar;view(2)
    xlabel("OTA 4");ylabel("OTA 5");title("Distance to ideal pole")
%     zlim([0 d.s3_p/10])
    zlim([0 target_distance_s3])

    figure();
    surf( Cf4_vector , Cf5_vector , d_zero );colorbar;view(2)
    xlabel("OTA 4");ylabel("OTA 5");title("Distance to ideal zero")
%     zlim([0 d.s3_z/10])
    zlim([0 target_distance_s3])
end


%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%
% I will choose those points that for both pole and zero are 1/10
smaller
% than in the original rounded  solution.

% valid_sol_crit1 = (d_pole <= d.s3_p/10) & (d_zero <= d.s3_z/10);
valid_sol_crit1 = (d_pole <= target_distance_s3) & (d_zero <=
target_distance_s3);

% and from those the one with smallest total number of caps
total_caps_reduced = total_caps;
total_caps_reduced(~valid_sol_crit1)= Inf;  % if is not valid then
assign infinite area

% express the solution space only considering the total area
for k=1:length(Cf4_vector)*length(Cf5_vector)
    total_caps_reduced_lin(k) = total_caps_reduced(k);
end
figure();
plot(total_caps_reduced_lin,'ok','Linewidth',3);hold on;grid on
xlabel('index');
ylabel('Total area [C_U]');
title('Section 3: Area for solution space');

[min_area , index] = min(total_caps_reduced_lin);
[cf_5,cf_4] = ind2sub(size(total_caps_reduced) ,index);

% sanity check to see if the obtained points make sense:
[d_pole(cf_5,cf_4) , d.s3_p ; d_zero(cf_5,cf_4) d.s3_z ;
total_caps_reduced(cf_5,cf_4) min_area]
```

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%
% Get the matrix with the chosen value:
Cap_f_round(4) = cf_4;
Cap_f_round(5) = cf_5;



% These are the chosen caps
% Cap_f_round(4) = 11;
% Cap_f_round(5) = 24;



Cap_s_round_final = round(cap_norm*Cap_f_round(4))  .* (ota_array==
4 ) +... % calculates and round the caps realted to 4nd integrator
only
                    round(cap_norm*Cap_f_round(5))  .* (ota_array==
5 ) +... % calculates and round the caps realted to 5rd integrator
only
                    Cap_s_round_temp .* ~((ota_array== 4) |
(ota_array== 5));   % the remaining caps remain unchanged

Cap_f_round_final = round(Cap_f_round);

% round all caps(the recently modified ones shouldn't change) and
calcute the transfer function
[ ~, ~, ~ , sta3] = cap_to_tf( Cap_s_round_final , Cap_f_round_final
, ota_array , Fs);

figure();pzmap(sta3 , 'r', rn.s3 ,'k' ,id.s3,'b');
legend('new','round','ideal')
title('Section 3');
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%
% Compare the overall rounded with the ideal and the rounded
originally:

[ overall, a, b , c] = cap_to_tf(round(Cap_s_round_final) ,
round(Cap_f_round_final) , ota_array , Fs);

figure();
subplot(121);
bode(overall , 'r', rn.all ,'k' ,id.all,'b');
legend('new','round','ideal')
title('Overall Bode'); grid on;

subplot(122);
pzmap(overall , 'r', rn.all ,'k' ,id.all,'b');
legend('new','round','ideal')
title('Overall pole zero');
```

```
within_spec = verify_filter_specs(overall , f_pass,f_stop,
ripple_passband, gain_stopband);


[a_p , a_z] = pzmap(a);
[b_p , b_z] = pzmap(b);
[c_p , c_z] = pzmap(c);

u.s1_p = abs(a_p   -id.s1_p   );
u.s1_z = abs(a_z   -id.s1_z   );
u.s2_p = abs(b_p(1)-id.s2_p(1));
u.s2_z = abs(b_z(1)-id.s2_z(1));
u.s3_p = abs(c_p(1)-id.s3_p(1));
u.s3_z = abs(c_z(1)-id.s3_z(1));


%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%
%% Other APPROACH : that is NOT what is described in the slides and
might have disadvantages

%{
%  express normalized cap as fractions optimizing number of caps and
absolute
% relative error sum for bunch of caps depending conencted to an
ota.

% Idea behind this:
%{
The algorithm is to assign a unit cap value to the integrating caps
and
then obtain the number of unit caps for the sampling caps by using
rounding.

Then calculate the resulting transfer function and calculate the
error of
all coefficients in the TF, and add them up (in magnitude).
Then increase the number of unit caps for the integrating caps.

At the end a map of the number of caps vs total error can be
generated
(for the biquads this has 2 dimensions, one for each feedback cap).
And
the minimum point can be obtained.

For the linear case I saw that there is correlation between the rms
error
of the gain and phase responses and the total error metric described
above. (haven't checked that for biquads).

NOTE 1: This optimizes area but not neccesarily in terms of noise.
Maybe some
other constraint can be added to that.
NOTE 2: This approach does not uses the suggestion in 626 slides
regarding
```

```
multiplying the smallest coefficient by ratio of the min cap value
divided the smallest coefficient. Not 100% sure how that works, but
could
be tested!
%}

% search range goes from 1 unit cap for the integrating cap up to
% Cf_unit_max
Cf_units = 10:10:10000;

% design region generation done in other function for compactness
(however
% the optimal point is not found yet)
[Lin.C.error_metric , Lin.C.total_caps] = explore_cap_ratio(Lin.c_dr
, "Linear" , Cf_units );
[HiQ.C.error_metric , HiQ.C.total_caps] = explore_cap_ratio(HiQ.c_dr
, "HighQ"  , Cf_units );
[LoQ.C.error_metric , LoQ.C.total_caps] = explore_cap_ratio(LoQ.c_dr
, "LowQ"   , Cf_units );

% The following plots show a general overview of the error metric vs
the
% total number of caps.
plot_this_2 = 0;

if plot_this_2
    figure();
    plot3(Cf_units, Lin.C.total_caps,Lin.C.error_metric,'--o' );grid
on
    xlabel("Int. Cap [C_U]");ylabel("Total caps
[C_U]");zlabel("error [%]");
    title("Linear Section");

    figure();
    surf(Cf_units , Cf_units , HiQ.C.error_metric , HiQ.C.total_caps
)
    xlabel("Int.2 Cap [C_U]");ylabel("Int.1 Cap
[C_U]");zlabel("error [%]");
    colorbar;
    title("High Q Section");

    figure()
    surf(Cf_units , Cf_units, LoQ.C.error_metric , LoQ.C.total_caps
)
    xlabel("Int.2 Cap [C_U]");ylabel("Int.1 Cap
[C_U]");zlabel("error [%]");
    colorbar;
    title("Low Q Section");
end

% The behavior for the HiQ and LoQ filters seems to be
% similar in that as the number of unit caps used in int1 increases
there
% is a point in which the error goes into a valley, and some more
increase
```

```
% makes it leave that valley into a peak. The cap used in int2
doesn't seem
% to have that behavior so much. One strategy could be to find the
valleys
% (downward concavity) to pick the cap_1 and then  move across cap_2
and
% find some reasonable point.

% Some points chosen visually...
% For Linear: Cint = 8 units;
% For High Q: Cint_2 = 11 units ; Cint_1 = 13 units;
% For Low Q : Cint_2 = 5 units  ; Cint_1 = 15 units;

% Minimum value possible for the givens earch range:
Lin_min_err = min(Lin.C.error_metric,[],'all');
HiQ_min_err = min(HiQ.C.error_metric,[],'all');
LoQ_min_err = min(LoQ.C.error_metric,[],'all');
best_error = max([Lin_min_err HiQ_min_err LoQ_min_err]);

%}

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%
%}
```