Ultra-high resolution A/D converters
Analysis, modeling and optimization tools

THÈSE N^{O} 6174 (2014)

PRÉSENTÉE LE 27 JUIN 2014

À LA FACULTÉ DES SCIENCES ET TECHNIQUES DE L'INGÉNIEUR
LABORATOIRE D'ÉLECTRONIQUE GÉNÉRALE 1
PROGRAMME DOCTORAL EN MICROSYSTÈMES ET MICROÉLECTRONIQUE

ÉCOLE POLYTECHNIQUE FÉDÉRALE DE LAUSANNE

POUR L'OBTENTION DU GRADE DE DOCTEUR ÈS SCIENCES

PAR

Sylvain MARÉCHAL

acceptée sur proposition du jury:

Dr J.-M. Sallese, président du jury
Prof. M. Kayal, Dr F. Krummenacher, directeurs de thèse
Dr Ph. Deval, rapporteur
Prof. F. Maloberti, rapporteur
Dr A. Schmid, rapporteur

ÉCOLE POLYTECHNIQUE
FÉDÉRALE DE LAUSANNE

Suisse
2014

Abstract

This thesis addresses the design of Sigma-Delta A/D converters over 20 bits. The main limitation in the design of ΣΔ ADCs is the validation of the circuits before production. As transistor-level simulators are too slow, models are required to validate the circuits.

The main contribution of this thesis is a software to simulate switched capacitors ΣΔ converters. The simulator distinguish itself by a direct description of the components - amplifiers, switches and capacitors - unlike existing high-level simulators, that use functional blocks: integration, subtraction, etc.
Modeling on the components level offers manifold benefits. The validation of the circuit is improved, as the model of the circuit is close to the final version: the sequence of the phases is identical, the common-mode is taken into account and the transfer functions for the signal and for the reference do not suffer from any approximation related to the architecture of the converter.

The new features provided by the simulator speed up the design of ΣΔ modulators. The implementation with switched capacitors is taken into account from the early design phases. It makes possible to compare the effects of components imperfections in several architectures and implementations in order to select a suited topology. Moreover, the sensitive capacitive parasitic couplings are easily identifiable to guide the layout, and the layout can finally be validated, simulating an extracted view in the high-level simulator.

As the acquisition of an electrical signal through an acquisition chain is not limited by the sole ΣΔ modulator, a fully digital solution - optimal filtering - is investigated to improve the resolution. Pre-amplification of the signal is also studied to optimize the power consumption and the noise level.

The theoretical contributions of this thesis are approved with the design of an integrated circuit including a front-end, an incremental ΣΔ modulator and digital filters. The converter exhibits a resolution of 20 bits with a sample rate of 64 Sps. The power consumption is lower than 400μW.

Keywords - Analog-to-Digital Converter; Sigma-Delta Modulation; Switched-Capacitors; Ultra-high resolution; Modeling; High-level Simulator; Incremental Converter; Optimal Filter; System-on-Chip.

Résumé

Cette thèse traite du développement de convertisseurs A/D Sigma-Delta atteignant des résolutions supérieures à 20 bits. La principale limitation dans la conception des ADC $\Sigma\Delta$ provient de la validation du circuit avant l'intégration. Les simulateurs de bas-niveau étant inadaptés car trop lents, il est nécessaire de valider ces circuits avec des modèles.

Le principal apport de cette thèse est un simulateur de convertisseurs $\Sigma\Delta$ à capacités commutées. L'aspect novateur du simulateur est une représentation directe des composants - amplificateurs, interrupteurs et capacités - contrairement aux simulateurs de haut-niveau existants où des blocs fonctionnels sont utilisés : intégration, soustraction, etc.
Les intérêts d'une telle modélisation sont multiples ; la validation est améliorée, car la représentation du circuit est proche de sa version finale : le séquençage des phases est identique, le mode-commun est pris en compte et les fonctions de transfert pour le signal et le bruit ne souffrent d'aucune approximation liée à l'architecture du convertisseur.

La conception de modulateurs $\Sigma\Delta$ est accélérée avec les nouvelles fonctionnalités offertes par le simulateur. Une sélection de l'implémentation avec des capacités commutées est prise en charge, dès les premières phases de design. Différentes architectures et implémentation peuvent ensuite être comparées vis-à-vis des imperfections des composants. Finalement, les couplages parasites sensibles peuvent être identifiés aisément afin de guider le layout, et le layout validé en simulant une vue extraite directement dans le simulateur de haut-niveau.

Comme l'acquisition d'un signal électrique à travers une chaîne d'acquisition complète ne se limite pas au seul modulateur $\Sigma\Delta$, une solution purement digitale - le filtrage optimal - est investiguée afin d'augmenter la résolution. La pré-amplification du signal est aussi étudiée pour optimiser la consommation et le niveau de bruit.

Les contributions théoriques de cette thèse sont approuvées avec la réalisation d'un circuit intégré incluant un front-end, un modulateur $\Sigma\Delta$ incrémental et les filtres digitaux. Le convertisseur atteint une résolution de 20 bits lorsqu'il acquiert 64 échantillons par seconde. La puissance dissipée est inférieure à 400μW.

Mots-clefs - Convertisseur Analogique-Numérique ; Modulation Sigma-Delta, Capacités-Commutées ; Ultra-haute résolution ; Modélisation ; Simulateur de haut-niveau ; Convertisseur Incrémental ; Filtre optimal ; SoC.

Remerciements

Je voudrais remercier premièrement ceux qui ont permis à ce projet de voir le jour et m'ont donné la chance de faire cette recherche en partenariat avec l'industrie : mon directeur de thèse, Prof. Maher Kayal, Dr. Jean-Paul Bardyn et François Salchli. J'aimerais aussi remercier ceux qui ont accepté de se pencher sur ce travail avec un regard intéressé et critique : le président du jury, Dr. Jean-Michel Sallese, et les experts Prof. Franco Maloberti, Dr. Philippe Deval et Prof. Alexandre Schmid pour avoir conclu cette thèse sur une discussion utile et intéressante.

Tout au long de ce travail, j'ai eu la chance d'être encadré au niveau technique par trois personnes sans qui cette recherche n'aurait pas été ce qu'elle est. Merci à mon co-directeur de thèse, François Krummenacher, Olivier Nys et Michel Chevroulet pour m'avoir guidé, conseillé et avoir partagé leur savoir lors de toutes nos discussions.

Je tiens à remercier tous mes collègues du laboratoire pour entretenir une atmosphère de travail plaisante et agréable. Merci aux secrétaires du labo, Isabelle et Karin, sans qui la vie de doctorant serait bien plus compliquée.
Merci à Cédric, Didier, François et Guillaume et plus particulièrement Andrea, mes collègues de bureau, pour les innombrables discussions techniques ponctuées de franches rigolades. Sans vous, le plaisir de venir chaque jour au bureau n'aurait pas été le même.

Ma gratitude va finalement à ma famille et à mes parents pour leur soutien inconditionnel qui, sans forcément comprendre mon travail, ont toujours cru en moi. A ceux qui contribuent aussi à faire jour après jour de moi celui que je suis, mes amis, Merci pour tous les bons moments, jeux, débats, ripailles et aventures plus ou moins dangereuses. Merci P. & N., F., S. & S., M., A.P. & M., J. & X., M. & Y., M. & S., M., C. & S., M. & D., M. & T., N. & J.B., et à celle qui partage ma vie depuis de nombreuses années, merci Estelle.

Lausanne, 28 Mai 2014 Sylvain Maréchal

Contents

Abstract (English/Français) **iii**

Remerciements **vii**

List of figures **xii**

List of tables **xviii**

1 Introduction **1**
- 1.1 Problems and Imperfections 3
- 1.2 Selected approach 3
- 1.3 Thesis outline 4

2 Basic concepts **7**
- 2.1 Sampling and oversampling 7
- 2.2 Quantization error and noise shaping 8
- 2.3 Feedback and feedforward modulators 9
  - 2.3.1 Feedback structure 10
  - 2.3.2 Feedforward 12
- 2.4 Cascaded modulators 13
- 2.5 Sigma-Delta or Incremental converters 14
- 2.6 Single/Multi-bit quantization 15
- 2.7 Conclusions 15

3 Optimal decoding **17**
- 3.1 First order 18
  - 3.1.1 Optimal filter – thresholds 21
- 3.2 Second-order – MASH 1-1 26
- 3.3 Sweep input 36
  - 3.3.1 Optimal decoder 39
- 3.4 Stochastic input 44
- 3.5 Conclusion 51

4 ADC Input stage - A comparative study 53

4.1 Two-phases architectures . . . 55
4.2 Symmetrical architectures . . . 59
4.3 Comparative study . . . 61
4.4 Conclusion . . . 63

5 Models 65

5.1 Integrator . . . 66
5.1.1 Ideal Integrator . . . 69
5.1.2 Amplifier - DC Gain and Offset . . . 69
5.1.3 Amplifier - Voltage Saturation . . . 71
5.1.4 Amplifier - Non-linear Gain . . . 71
5.1.5 Amplifier - Transient - SR . . . 74
5.1.6 Amplifier - Transient - SR Hyperbolic Tangent . . . 80
5.1.7 Switches - Transient . . . 81
5.1.8 Switches - Clock Feedthrough . . . 85
5.1.9 Non-linear Capacitors . . . 87
5.1.10 Thermal Noise . . . 92
5.1.11 Quantizer . . . 96
5.1.12 DAC . . . 98
5.2 Linear Circuit . . . 99
5.2.1 Matrix Representation . . . 100
5.3 Conclusion . . . 101

6 Simulator 103

6.1 Design steps and simulators . . . 103
6.1.1 Definition of the architecture . . . 104
6.1.2 Implementation with switched capacitors . . . 104
6.1.3 Implementation of the digital blocks . . . 105
6.1.4 Design of the analog blocks . . . 106
6.1.5 Post-layout validation . . . 106
6.1.6 Summary of the design steps . . . 107
6.2 Specifications . . . 108
6.3 Simulator Core . . . 108
6.3.1 Simulator - Top Classes . . . 109
6.3.2 Simulator - Architecture Classes . . . 109
6.3.3 Simulator - Integrator Classes . . . 111
6.3.4 Simulator - Linear Circuit . . . 112
6.4 Applications . . . 115
6.4.1 Testbenches . . . 115

6.4.2 Mismatch Analysis 116
6.4.3 Parasitic Coupling 118
6.4.4 Comparison of Architectures 120
6.4.5 Conclusion 127

7 ADC 129

7.1 Specifications 129
7.2 Selection of the architecture 130
7.2.1 Structure 131
7.2.2 OSR - C_{in} 131
7.2.3 Topology - Order 131
7.3 Implementation 133
7.3.1 Protection for the reference input 134
7.3.2 Clock feedthrough 136
7.3.3 Parasitic coupling 137
7.3.4 2^{nd} and 3^{rd} integrators 139
7.3.5 Incremental and $\Sigma\Delta$ modes 140
7.3.6 Coefficients 141
7.3.7 Sizing 141
7.4 Digital controls and filters 145
7.4.1 Controls 145
7.4.2 Filters 147
7.4.3 Mathematical relationships 148
7.5 Conclusion 152

8 PGA 153

8.1 Auto-Zero Amplifier 154
8.1.1 Offset and Flicker Noise Reduction Techniques 154
8.1.2 Proposed Circuit 155
8.1.3 Implementation 158
8.2 Noise Analysis 160
8.2.1 1/f noise - Auxiliary chain 161
8.2.2 1/f noise - Main amplifier 161
8.2.3 Thermal noise - Amplifiers 163
8.2.4 Thermal noise - Input chopper 164
8.2.5 Thermal noise - Resistors 164
8.2.6 Common-mode control circuits 165
8.2.7 Power Consumption 171
8.2.8 Noise Contributions - Summary 172
8.3 Common-mode rejection 174

8.3.1 Proposed solution 175
8.3.2 Nonideal effects 177
8.3.3 Implementation 178
8.4 Output buffers 179
8.4.1 Passive resistors 179
8.4.2 Active buffers 180
8.5 Conclusion 181

9 Characterization 183
9.1 Functional tests - ADC 184
9.2 Functional tests - PGA 187
9.3 Performances 189
9.4 Conclusions 191

10 Conclusion 193
10.1 Main contributions 193
10.2 Future perspectives 194

Bibliography 204

List of Figures

2.1 Block diagram of a Sigma-Delta A/D converter. 7
2.2 Transfer functions and quantization error of (left) multi- and (right) single-bit quantizers. 8
2.3 Linear representation of the first-order ΣΔ modualtor. 9
2.4 Second-order unstable modulator. 9
2.5 Second-order CIFB modulator. 10
2.6 Second-order CIFB modulator with input feedforward. 11
2.7 Second-order CIFB modulator with input feedforward. 11
2.8 Second-order CIFF modulator with input feedforward. 12
2.9 MASH 1-1-1 modulator. 13

3.1 Comparison of the counter and optimal filters for an OSR of two bits. . 19
3.2 First order modulator. 19
3.3 Thresholds for the 16 first bits of the first order modulator. 22
3.4 Example of an update of the input space. After each bit, the possible input space (blue) is compared to the bit[k] thresholds (red boxes) and updated accordingly 23
3.5 Transfer function using counter filter (dash-dot line) and optimal filter (solid line) for $nbits = 4$. 24
3.6 Mean resolution using counter filter (dotted line) and optimal filter (solid line). The dashed lines are referenced to the upper logarithmic scale. 25
3.7 Block diagram of a MASH11 ΣΔ modulator with a constant input. . . . 26
3.8 Linear filter relative coefficients for the first (left) and second (right) stage for an over-sampling ratio of $nbits = 6$. 29
3.9 Optimal filter decoding flow. 30
3.10 Transfer function of the MASH 1-1 optimal filter (solid), and of the linear filter (dash dots). 34
3.11 Quantization error using the MASH 1-1 optimal filter (top), and the linear filter (bottom) for $nbits = 8$. 35

3.12 Mean resolution of the MASH 1-1 optimal filter (solid line), and of the linear filter (dashed line). 36
3.13 Sweep input: mean value and slope. 37
3.14 Block diagram of first-order ΣΔ modulator with sweep input. 37
3.15 Iterative process to generate a polygon using thresholds, $nbits = 3$. Top left figure is the starting polygon, inside which the modulator is not saturated. Axes are α (horizontal) and x (vertical). Input has a slope $\alpha = 0.05$ and a normalized mean $x = 0.65$ (star). 40
3.16 Optimal filtering procedure. 41
3.17 Mean resolution of the optimal filter with sweep input (solid line), and of the counter filter (dashed line). 45
3.18 First order modulator with a noisy input. 45
3.19 Graphical representation of the probability domains (integration surfaces) for the two-bits OSR. The input x is a parameter, shifting the blue lines. The axis u and v represent the uncorrelated error $err(0)$ and $err(1)$. 47
3.20 Probability functions for the 8 output codes of a three-bits output. The considered noise function has a Gaussian shape and a deviation $\sigma = 0.05$. 49
3.21 Probability functions of the code 100 for values of σ from 0.17 to 0.01. The left plot corresponds to $\sigma = 0.17$ while the narrow one, around 0.5 was generated with $\sigma = 0.01$. 50

4.1 High-level representation of a first-order modulator. 53
4.2 Control signals for the architectures #1 to #9. 54
4.3 Architecture #1. 55
4.4 Architecture #2. 55
4.5 Architecture #2b. 56
4.6 Architecture #3. 56
4.7 Architecture #4. 56
4.8 Architecture #5. 57
4.9 Architecture #6. 58
4.10 Architecture #7. 58
4.11 Architecture #8. 59
4.12 Architecture #9. 60
4.13 Architecture #10. 60
4.14 Specific control signals for the architecture #10. 61

5.1 High-level view of a first-order ΣΔ modulator. 66
5.2 Implementation of the integrator with switched capacitors in a fully-differential topology. 67

5.3 Representation of a two-phases integrator (a) into two similar subcircuit for phase 1 (b) and phase 2 (c). 68

5.4 Saturation transfer function. The saturated output cannot reach voltages beyond the V_{sat-} and V_{sat+} limits. 71

5.5 Representation of the model of the integrator considering an amplifier with a non-linear gain. The parasitic capacitors taken into account in the model are displayed. 72

5.6 Graphical representation of the normalized gain functions (5.15) of the amplifier. 74

5.7 Model of the amplifier with a limited output current. 75

5.8 Normalized amplifier transconductances with a maximal slew rate. The model of the amplifier with a constant transconductance g_m and a current limit I_0 is described with the dark blue piecewise linear function. The red curve is the continuous hyperbolic tangent representation, section 5.1.6. 75

5.9 Ouput transient voltage with a slewing phase between $t = 0$ and $t = T_0$ followed by a settling phase for $t > T_0$. 76

5.10 Equivalent linear circuit during initialization. 77

5.11 Equivalent linear circuit during the slewing phase. 78

5.12 Equivalent model of the integrator with non-ideal switches. 82

5.13 Graphical representation of the equivalent conductance of a transmission gate. The equivalent conductance is the sum of the conductance g_n of the NMOS and of the conductance g_p of the PMOS. 83

5.14 Illustration of the clock feedthrough for one of the switches on high-impedance nodes. While the switch is opened, the variation of the gate voltage V_g forces a charge transfer of the charges held in the Gate-Drain and Gate-Source capacitors. 86

5.15 Simplified representation of the charge distribution test-bench. The line and sensor impedances are modeled with a serial resistance R_a. 86

5.16 (a) MIM capacitance created between two metal layers. (b) Fingers capacitance, interleaving both electrodes on each metal layer. 88

5.17 Equivalent model for an integrator with non-linear capacitors. 88

5.18 (a) Equivalent model of one of the input branches during pre-charge. (b) Small-signal representation including noise sources. 93

5.19 Simplified (single-ended) representation of the equivalent input of the integrator including parasitic capacitors. 95

5.20 (a) Equivalent simplified model of the integrator during the charge transfer. (b) Small-signal representation including noise sources. 96

5.21 Linear and non-linear transfer functions. 97

5.22 Non-linear transfer function with hysteresis. 97
5.23 DAC implementation with switched-capacitors. 98
5.24 Splitting of a circuit into linear (gray) and non-linear (white) subcircuits. 100

6.1 Analog and digital fundamental elements of a ΣΔ converter. 103
6.2 IOs and main processing functions of the simulator. 108
6.3 Classes of the simulator. Connections are inheritance from parent class. Colors are used to distinguish the two simulators (blue - green), the definition of the topology (orange) and common files (gray). 110
6.4 Simulation flow of the linear simulator. 113
6.5 Block view of the communication with the simulator. 115
6.6 Definition of gain error, offset error, INL error from the transfer function (left) and Peak-Peak and RMS resolution (right). 116
6.7 Monte-Carlo simulation flow. 117
6.8 Schematic of a MASH 1-1 ΣΔ modulator. 117
6.9 Statistical distributions of gain and offset error, P-P and RMS resolution for a cascaded ΣΔ ADC. Set of 500 samples. 118
6.10 Top-down design flow and contribution of the tool for parasitic coupling.119
6.11 Parasitic coupling simulation flow. 119
6.12 INL error of a ΣΔ ADC with a parasitic coupling problem. Left: using proposed tool. Middle: using a transistor level simulator with macro blocks for amplifiers, comparators, etc. Right: measured INL. The asymmetry in the middle graph is due to a lower number of points. 120
6.13 Quantization noise limitation of the four selected architectures. 121
6.14 Evolution of the resolution with a finite DC gain in the first amplifier. . 122
6.15 Simulated INL of the MASH 1-1 architecture with a finite DC gain in the first amplifier, 80dB. 122
6.16 Evolution of the resolution with a finite DC gain in the second amplifier. 122
6.17 Evolution of the resolution with a constant integrated capacitor ($C_{in}OSR$ product). The right plot shows a thermal noise 8 times lower as the capacitors are 64 times larger. 123
6.18 Impact of the settling time on the linearity and gain error. 124
6.19 Simulation example with 1^{st} and 2^{nd} order nonlinearity 124
6.20 Variation of the first-order (left) and of the second-order (right) voltage coefficient of the non-linear capacitors. 125
6.21 Distortions of the INL due to parasitic capacitors. 125

7.1 Acquisition chain. 129
7.2 Second-order feedback stage. 132
7.3 First-order stage. 133

7.4 Evolution of the resolution with the OSR in noiseless ideal modulators. 133
7.5 Switched capacitors input with switching scheme #5. 134
7.6 Left: Voltage limitation implemented with capacitors. Right: implementation with serial resistors. 135
7.7 Distribution of the charges of the switch into the input and the integration capacitances. 136
7.8 Modification of the structure #5 to avoid critical capacitive parasitic coupling. 138
7.9 Charge transfer between the two first integrators. Top: direct connection. Bottom alternate connection. 139
7.10 High-level view of the implemented architecture. 141
7.11 Implementation of the first stage. 143
7.12 Implementation of the second stage. 144
7.13 General view and connections of the two dedicated digital blocks. . . . 146
7.14 Blocks view of the digital filters. 148

8.1 Acquisition chain. 153
8.2 Architecture of the implemented front-end. 154
8.3 Classic continuous time auto-zeroed amplifier. 156
8.4 Flicker noise rejection using chopper modulation. 156
8.5 Auto-zero amplifier principle. 157
8.6 Switched-capacitors resistor using left: standard implementation, right: symmetrical implementation with choppers. 158
8.7 Auto-zero amplifier: step 1. 159
8.8 Auto-zero amplifier: step 2. 159
8.9 Auto-zero amplifier: final implementation. 160
8.10 Complex representation of the nulling amplifier in the signal baseband (low frequencies). 161
8.11 Parasitic capacitors in the integrator. 164
8.12 Resistors noise. 165
8.13 Gain stage input CMFB. 166
8.14 Input-referred modulated noise. 168
8.15 Input-referred demodulated noise. 168
8.16 Input-referred noise contribution of the gain stage CMFB circuit. . . . 168
8.17 Integrator input CMFB. 170
8.18 Simplified (single ended) continuous-time representation of the noise contribution of the CMFB circuit in the integrator. 171
8.19 Auto-zero main noise sources. 172
8.20 Classic three-op-amp instrumentation amplifier. 175

8.21 Amplification of input signal with different common-mode voltages. Left: Common-mode centered. Right: Common-mode close to V_{dd}. . . 175
8.22 Instrumentation amplifier with a controlled common-mode 176
8.23 Amplification of input signal with different common-mode voltages using the output common-mode rejection circuit. Left: Input common-mode equal to the reference common-mode. Right: Input common-mode close to V_{dd} 176
8.24 Input common-mode range. 177
8.25 Equivalent output load, including the input capacitors of the ADC. . . 179
8.26 PGA output current limitation with serial resistors. 180
8.27 PGA output structure using buffers. 180
8.28 Output buffer. 181

9.1 Layout of the integrated circuit. 183
9.2 Layout of the front-end (left) and of the modulator (right). 184
9.3 Measured noise distribution caused by a reset phase too short. 185
9.4 Measured noise distribution of the ADC with an extended reset phase. 186
9.5 Measured INL of the topology #5 around mid-scale. 186
9.6 Modified version of the topology #5. The four extra switches, named '1' and '2', are the ones connected to the virtual ground of the amplifier. . 187
9.7 Output noise distribution of the PGA without the Auto-Zero (left) and with the auto-zero enabled (right). The upper plots show the flicker noise rejection with a gain of 8, while a gain of 64 was set in the lower plots. 188

List of Tables

3.1 Estimates of the three-bits codes. 1^{st} order modulator and constant input. 50

4.1 Advantages and drawbacks of the selected implementations 63

6.1 Main design steps of a ΣΔ ADC with a top-down procedure. 107
6.2 Comparison of four architectures. 126

7.1 Operating conditions. 130
7.2 Block specifications. 130
7.3 Gain error coefficients. 151

8.1 Auto-zero amplifier: noise summary and key characteristics 174

9.1 Performance comparison. 190

1 Introduction

This thesis is dedicated to the analysis, modeling and optimization of discrete-time analog-to-digital converters implemented with switched capacitors.

The quality of an A/D converter can be estimated with a limited set of factors, starting with the resolution. The notion of resolution may be confusing. On one hand, there is the number of bits on the digital output and, on the other hand, the noise level and the linearity of the converter. As it is possible to artificially increase the width of the digital output bus, the effective resolution is preferred in this thesis. The Effective Number Of Bits (ENOB) is taking into account the noise level of the converter.

The second main characteristic of an ADC is the sampling frequency. In most of the data acquisition devices, the user is free to exchange time for precision, i.e. select between a high resolution and a slow sampling rate or a lower resolution with a faster refresh rate.

The power consumption is the third main characteristic of the quality of an ADC. Its importance is occasionally underestimated as it is often invisible for the end user, which can neither measure nor optimize the power consumption of most electronic devices. The power consumption is nevertheless a key characteristic of a converter, as it is possible to enhance the resolution or the frequency increasing the power consumption.

Apart the three previously cited criteria, other ones influence the design. They are either dependent on fabrication costs - process, silicon area, trimming, production yield; or related to external environmental constraints - temperature range, input impedance... All these parameters strongly depend on the end use for which the ADC is sized. It is thus inconvenient to use these parameters to measure and compare the performances of data converters.

It is convenient to aggregate the three main quality factors in: resolution (ENOB or

SNR) and energy per conversion, common measure of the frequency and of the power consumption.

The elaboration of a common figure of merit including the power P and the sampling frequency f_s is subject to controversy. Two FOM are nevertheless widely spread. The first one is:

$$FOM_1 = \frac{P}{2^{ENOB} f_s} \tag{1.1}$$

Historically the oldest one, it is not suited for converters in which the thermal noise dominates (i.e. $NSD \propto kT/C$ or $NSD \propto 4kTR$), as for each extra bit the power consumption is increased by a factor 4.

The second figure of merit, mainly used to compare high-resolution converters is:

$$FOM_2 = \frac{P}{2^{2ENOB} f_s} \tag{1.2}$$

Obviously, the best way to agree with both definitions would be to compare only converters with identical resolutions.

The resolution of oversampling data converters is limited by three main factors:

- The quantization noise: Determined by the order of the converter. There is no direct relationship between the quantization noise and the sampling frequency or the power consumption.
- The thermal and the flicker noises: The thermal noise is proportional to the power consumption (or to the frequency if an averaging is possible). The noise power of 1/f noise sources is inversely proportional to the current consumption and to the gate area of active transistors. As the spectrum of the 1/f noise is not white, circuit techniques are commonly implemented to reduce its level in the signal baseband.
- Circuit imperfections: Non-ideal effects of the analog part of the converter. Fundamentally, these effects are not directly related to the frequency or to the power consumption.

While the first two limitations - quantization and thermal/flicker noises - are well known and are easily estimable, it is more fastidious to take the last category into account.

Reducing the non-ideal effects requires a special care and a very good knowledge of the architectures and implementations of the converters. Most of the design effort should thus be invested to guarantee a negligible degradation of the resolution and a minimal consumption overhead.

1.1 Problems and Imperfections

The designer of a $\Sigma\Delta$ modulator should be aware of the design issues limiting the resolution of the converters. The non-ideal effects may be classified into two main groups: first the ones related to the base components and then the ones related to the architecture. The best-known limitations for the blocks are:

- Amplifiers: Offset, finite gain, non-linear gain, limited bandwidth and slew-rate.
- Switches: On-resistance, clock feedthrough.
- Quantizers: linearity, hysteresis.
- DACs: Mismatch, non-linearity.

Furthermore, the problems related to the signal dynamic range and to the closed loop architecture of the $\Sigma\Delta$ modulator have also to be taken into account:

- Saturation of the amplifiers.
- Excessive loop delay.
- Noise shaping and noise rejection.
- Loop stability.
- Insufficient common-mode rejection.
- Excessive signal amplitude (loss of charges through the power supplies).

In addition to problems related to the analog modulator, it is also necessary to check the analog-digital interfaces to ensure that the control signal are driving in a correct order the analog switches (in particular if the driving signals are passing through level shifters, or if non-overlap conditions should be guaranteed). Parasitic couplings between analog signals, and from digital to analog signals, have also to be taken into account.

Most of the architectural problems are related to the implementation of the integrators with switched capacitors. E.g., excessive signal amplitude at the output of amplifiers can be reduced increasing the integration capacitors. Loop delay may also be adjusted using appropriate switching phases.

1.2 Selected approach

The selected approach in this thesis to study, design and optimize high-resolution converters is the following:

- **Modeling**: Two key elements are missing to fulfill the design toolset: an efficient

validation of the implementation and a verification of the parasitic coupling in a reasonable time. As the implementation of ΣΔ modulators with switches and capacitors is not integrated into existing high-level simulation tools, models have to be developed and integrated into a dedicated simulator. The simulator has to account for commonly known non-ideal effects to precisely evaluate the quality of ΣΔ topologies.
- **Comparative study** based on a limited number of architectures. The aim of the comparison is firstly to validate the models and the simulator with existing circuits and secondly to compare accurately several implementations with switched capacitors.
- **Design** of a switched capacitors incremental converter following a top-down methodology. The choices of implementation and the optimization of the blocks are carried out with the developed simulator.

The modulator is the key component of any ΣΔ acquisition chain. However, it is possible to improve the resolution with pre- and post-processing.

- **Digital filters**: Classic linear filters perform a weighted average of the output stream of the modulator. The hypotheses on the quantization noise - considered as white noise - lead to a suboptimal estimation of the input signal. The gradual shift to submicron technologies reduces the costs (power consumption and silicon area) of the digital, while improving the available features. Non-linear filtering is studied to improve the decoding of the output of the modulator.
- **Front-end**: The best performances of data converters are obtained when the range of the input signal is close to full-scale. In order to be compatible with a large selection of sensors, pre-processing of the analog signal may be accomplished in a prior gain stage. The aim of the front-end is on one hand to amplify the signal and, on the other hand, to provide an input impedance for sensors with limited output current.

1.3 Thesis outline

The next chapter provides the necessary bases to understand ΣΔ modulators. The linear model is introduced and the most common architectures are discussed. Differences and tradeoffs of the main categories of architectures are briefly stated.

Chapter 3 studies optimal filters. It shows that useful information on the input signal is lost with linear filters and provides algorithms to enhance the resolution. Several architectures are covered assuming various hypotheses on the input signal: constant input, sweep input and constant with Gaussian noise.

Chapter 4 compares implementations with switched capacitors. Ten types of input connection achieving the same high-level functionality are analyzed. The main performances - noise, power consumption - are compared, as well as the sensitivity to non-ideal effects.

Chapter 5 models ΣΔ converters. The integrator is first modeled, including best known limitations in switched capacitors topologies. Secondly, a linear model is expressed to model the full modulator and account for capacitive parasitic coupling. In both cases, special care is taken to minimize the required mathematical complexity.

Chapter 6 introduces a simulator. A comparative review of existing tools discusses first the features of the simulator. An object-oriented software architecture is then defined to be compatible with a large selection of ΣΔ ADC topologies. The scope of operations of the simulator is finally exhibited through a comparative study of four existing converters.

Chapter 7 discusses the design of a ΣΔ ADC in a top-down methodology. This design example takes advantage of prior studies and of the simulator to define an architecture suited for ultra-high resolution converters. Each critical design issue, identified with the developed tools, is discussed.

Chapter 8 introduces a new architecture for a low-noise low-power front-end. A novel common-mode control for instrumentation amplifiers is discussed. A detailed analysis is then performed on an auto-zero amplifier with a fully differential cancellation path to optimize the noise and power consumption performances.

Chapter 9 provides the results measured on the implemented ADC and discusses the observed limitations.

Finally, chapter 10 concludes this thesis, highlights the main contributions and provides suggestions for future developments and improvements.

2 Basic concepts

This chapter briefly introduces the basic concepts of ΣΔ A/D conversion and reviews the common architectures.

This chapter is not intended to cover in details all aspects of ΣΔ modulation; instead, it provides the information necessary to understand the next chapters.

2.1 Sampling and oversampling

Figure 2.1 shows a block diagram of a ΣΔ A/D converter that includes an anti-aliasing filter, a modulator and a digital filter.

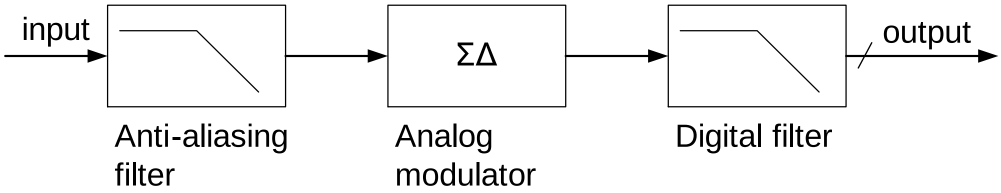

Figure 2.1: Block diagram of a Sigma-Delta A/D converter.

The anti-aliasing filters suppresses the signal spectral components over half of the sampling frequency. One of the main benefits of the oversampling converters is the relaxation of the requirements of this filter. Indeed, a sharp cut-off is not required, and a first-order filtering is usually sufficient.

The modulator samples and quantizes the signal. In addition to these two operations performed by any A/D converter, a ΣΔ modulator shifts most of the energy of the quantization noise out of the signal baseband. This effect is named noise shaping. The output of the modulator is a digital stream at the oversampling frequency with a limited number of levels.

The signal then enters in the digital filter. The aim of the filter is firstly to suppress the high-frequency components of the modulated signal and then to under-sample the signal to the Nyquist frequency.
The output of the converter is finally a representation of the input signal, coded in a large number of bits, at the Nyquist rate.

2.2 Quantization error and noise shaping

Amplitude quantization of a signal is required in any A/D conversion. In a $\Sigma\Delta$ modulator, a flash converter with a limited number of levels (usually between 1 and 5 bits) is used.

The transfer function of an ideal quantizer is shown in Fig. 2.2. In a multi-level comparator, an input range is defined, beyond which the quantizer is saturated at a constant value. If the quantizer is single-bit (comparator), the output is always saturated and toggles as soon as the comparison threshold is crossed. The input range is thus not defined in a comparator.

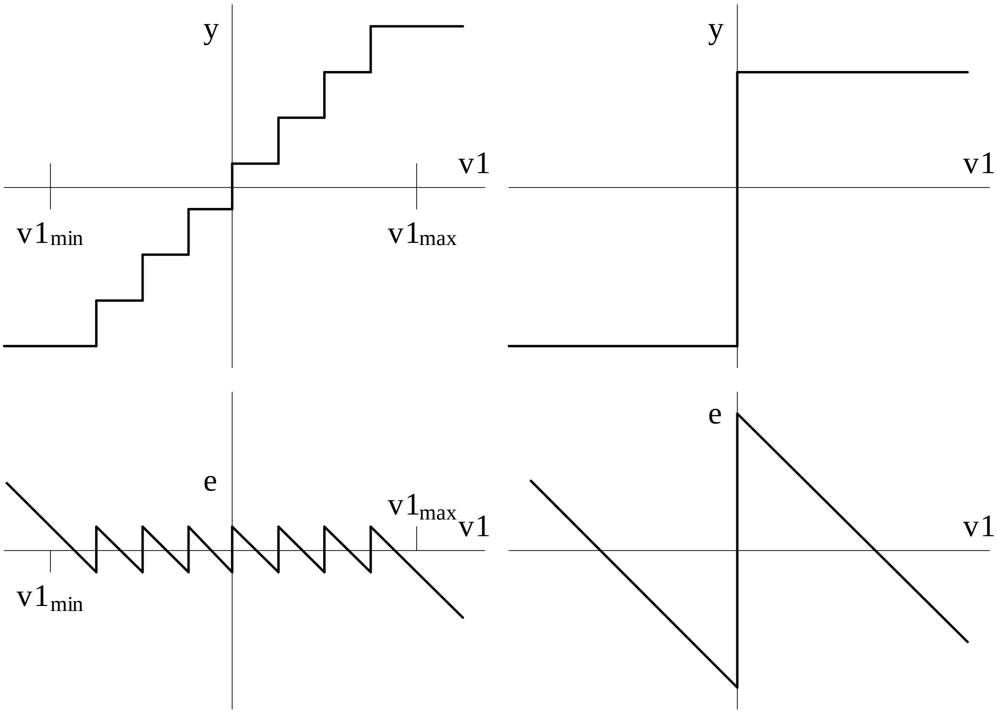

Figure 2.2: Transfer functions and quantization error of (left) multi- and (right) single-bit quantizers.

The difference between the input signal of the quantizer $v1[k]$ and its output $y[k]$ is called quantization error, $e[k]$.

$$y[k] = v1[k] + e[k] \tag{2.1}$$

The linear representation of the first-order ΣΔ modulator is illustrated in Fig. 2.3. Such a linear view allows deriving the transfer functions for the signal (STF) and for the quantization noise (NTF).

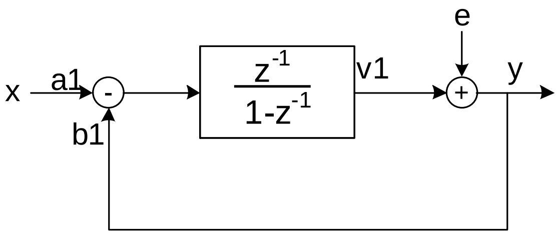

Figure 2.3: Linear representation of the first-order ΣΔ modualtor.

The analysis of the linear representation of the first-order system with unit coefficients leads to the following results:

$$\begin{aligned} STF &= \frac{y}{x} = z^{-1} \\ NTF &= \frac{y}{e} = 1 - z^{-1} \\ v1 &= z^{-1}x - z^{-1}e \end{aligned} \tag{2.2}$$

The STF exhibits an all-pass response and the NTF provides a first-order high-pass filtering.

2.3 Feedback and feedforward modulators

The most straightforward method to construct high order modulator is to cascade several integrators in the forward path. However, the direct implementation, Fig. 2.4, is unstable.

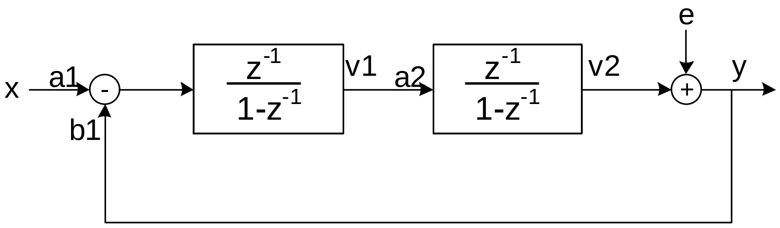

Figure 2.4: Second-order unstable modulator.

Two structures are commonly used to stabilize the modulator. In the first one called a Cascade of Integrator with distributed FeedBack (CIFB) [1, 2], each integrator receives feedback from the quantizer. Alternatively, weighted feedforward can be added in a Cascade of Integrators with weighted FeedForward summation (CIFF) [3, 4].

2.3.1 Feedback structure

The feedback topology is illustrated for a second-order modulator in Fig. 2.5.

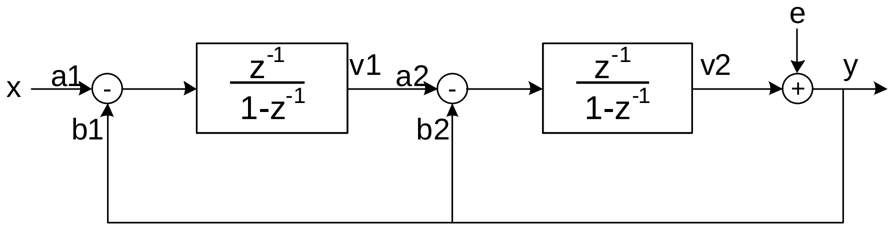

Figure 2.5: Second-order CIFB modulator.

The analysis of this topology with the coefficients a1=a2=b1, b2=2 leads to the following results:

$$\begin{aligned} STF &= \frac{y}{x} = z^{-2} \\ NTF &= \frac{y}{e} = \left(1-z^{-1}\right)^2 \\ v1 &= z^{-1}\left(1+z^{-1}\right)x - z^{-1}\left(1-z^{-1}\right)e \\ v2 &= z^{-2}x - z^{-1}\left(2-z^{-1}\right)e \end{aligned} \tag{2.3}$$

The STF is flat and the NTF is a second-order high-pass filter. The main disadvantage of this configuration is that the signals at the output of the amplifiers, $v1$ and $v2$, are a function of the input signal x. In this feedback structure, the design of the amplifiers is difficult, firstly because a large output dynamic range is required and secondly because the non-linearity of the amplifiers distorts the transfer function of the ADC.

The signal-dependent term in (2.3) may be suppressed injecting the signal at the output of the amplifiers. The generalized version of the CIFB structure, sometimes named CIFB with Input Feedforward is displayed in Fig. 2.6.

This structure with the coefficients a1=a2=a4=b1=1, a3=b2=2 leads to a flat STF and ensures that the output of the integrators $v1$ and $v2$ are not correlated with the input signal[1]. The NTF remains identical than in (2.3).

---

[1] Assuming that the quantization error is stochastic and thus not correlated with the input signal.

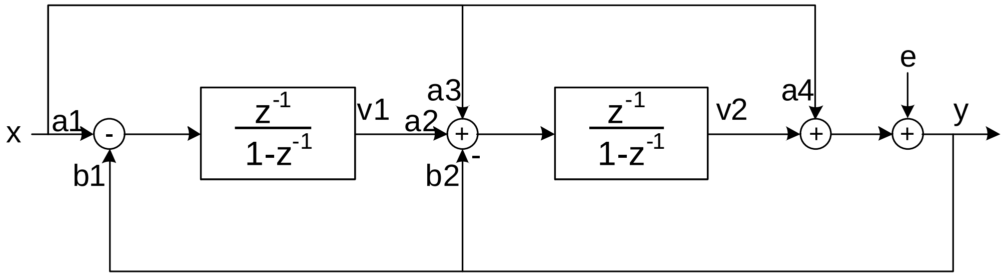

Figure 2.6: Second-order CIFB modulator with input feedforward.

Besides the extra load on the input and the increased complexity of the circuit, one of the main disadvantages of this structure is the delay-free path from the input, through the quantizer, and back to the input of the modulator [5].

The independence of $v1$ relatively to the input signal may however be partially guaranteed in an architecture without any direct path from the input to the quantizer. An example of implementation is shown in Fig. 2.7.

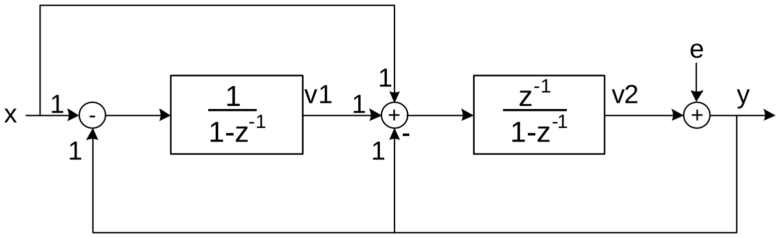

Figure 2.7: Second-order CIFB modulator with input feedforward.

The transfer functions of the structure, Fig. 2.7, are:

$$\begin{aligned}
STF &= \frac{y}{x} = z^{-1}\left(2-z^{-1}\right)\\
NTF &= \frac{y}{e} = \left(1-z^{-1}\right)^2\\
v1 &= \left(1-z^{-1}\right)x - \left(1-z^{-1}\right)e\\
v2 &= z^{-1}\left(2-z^{-1}\right)x - z^{-1}\left(2-z^{-1}\right)e
\end{aligned} \tag{2.4}$$

If the oversampling frequency is much larger than the input signal frequency, the output of the amplifier $v1$ is only related to the quantization error. The output of the second amplifier is still proportional to x, however, the design constraints on the second amplifier are less restrictive.

The drawback of this implementation is the non-constant STF. In particular, the STF exhibits a gain of 3 at half of the oversampling frequency. This amplification of out-of-band frequencies is not critical for converters with large oversampling ratio, but adds more constraints on the anti-alias filter in ADCs with limited OSR.

2.3.2 Feedforward

The second common way of ensuring stability in high-order ΣΔ modulators is to add forward paths to the quantizer. The generic structure of CIFF with input feedforward is illustrated in Fig. 2.8 for a second-order modulator.

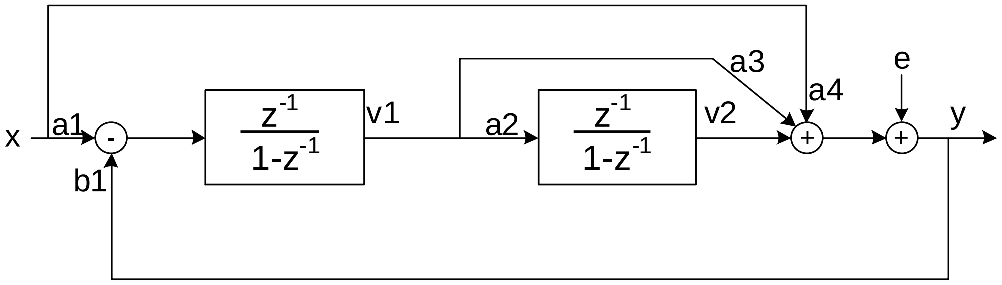

Figure 2.8: Second-order CIFF modulator with input feedforward.

The coefficients[2] a1=a2=b1=1, a3=2, a4=0 lead to the following results:

$$\begin{aligned} STF &= \frac{y}{x} = z^{-1}\left(2 - z^{-1}\right) \\ NTF &= \frac{y}{e} = \left(1 - z^{-1}\right)^2 \\ v1 &= \left(1 - z^{-1}\right)x - \left(1 - z^{-1}\right)e \\ v2 &= z^{-2}x - z^{-2}e \end{aligned} \tag{2.5}$$

The main characteristics, STF, NTF and output of the first integrator ($v1$) are identical to the ones of the CIFB structure, Fig. 2.7. The output of the second integrator is slightly less sensitive to the quantization error, but remains proportional to the input signal

The previous analysis showed that the basic characteristics of the feedback and of the feedforward structures are equivalent. The drawbacks of each topology emerge during the implementation. A CIFF modulator requires an extra adder before the quantization and loads more the output of the first amplifier. Both feedback paths in the CIFB structure must be properly matched to avoid a mismatch between the analog and the digital.

---

[2] Here as well, the a4 coefficient is zero to avoid a direct path from the input to the quantizer

2.4 Cascaded modulators

Single-stage modulators shift the quantization noise out of the signal base-band. Another technique to lower the quantization noise is to measure and subtract it. In cascaded, or MASH, topologies, each modulator (stage) measures the quantization noise of the previous stage. The bit-streams are then combined in the digital logic in order to cancel the quantization noise of all stages, excepting the last one.

A third-order cascaded ADC is shown in Fig. 2.9. This converter is named MASH 1-1-1 as each stage is a first order modulator.

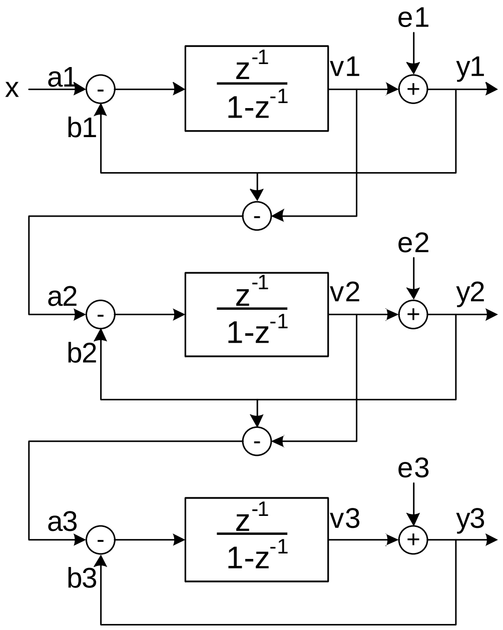

Figure 2.9: MASH 1-1-1 modulator.

Cascaded ADCs made of first- or second-order modulators are stable by construction. Moreover, for identical order and OSR, the quantization noise is lower than in single-loop topologies. The drawback of the cascaded architecture is the good matching required between the analog and digital circuits [6]. Particularly, for high-resolution ADCs, an excellent matching of the capacitors, a long enough settling time as well as a huge DC gain of the amplifiers are needed.

MASH converters are better suited for modulators with limited OSR. They are thus mainly used in fast ADCs with limited resolution [7].

2.5 Sigma-Delta or Incremental converters

The difference between $\Sigma\Delta$ converters and incremental converters is the reset of the integrators before each conversion in incremental modulators.

In incremental ADCs, the successive samples are not correlated. These data converters are thus suited for multiplexed inputs, when the input signal is delivered by distinct sensors. The second advantage of these converters is the suppression of the idle tones, visible in $\Sigma\Delta$ ADCs for static inputs[3].

The generation of the clocks is however more complicated as extra phases for the resets have to be created. Moreover, the sampling of the signal is not uniformly spread over time as no sample is acquired during the reset phases.

The analog implementation of $\Sigma\Delta$ and incremental converters is almost identical. Digital filters are on the other side fundamentally different. In $\Sigma\Delta$ converters, a sinc filter is applied. The order of the filter is equal to the order of the modulator plus one. E.g. in a second-order modulator with an OSR of 4, the filter is:

$$\begin{aligned} H_s(z) &= \left(1 + z^{-1} + z^{-2} + z^{-3}\right)^3 \\ &= 1 + 3z^{-1} + 6z^{-2} + 10z^{-3} + 12z^{-4} + 12z^{-5} + 10z^{-6} + 6z^{-7} + 3z^{-8} + z^{-9} \end{aligned} \tag{2.6}$$

In incremental converters, the filtering philosophy is different. After each cycle of the modulator, the output of the last integrator $res[k]$ is estimated, and a portion of the reference is added or subtracted. After OSR cycles, the residue within the last integrator is a weighted sum of the input and of the reference:

$$res[OSR] = \sum_{k=0}^{OSR-1} (OSR - k)\,(x[k] - bit[k]) \tag{2.7}$$

In a classic approach[4], the probability distribution of the residue is considered to be symmetrical and with a zero mean. The normalized input x is thus directly estimated with the weighted sum of the bit-steam. E.g. in a second-order modulator with an OSR of 4, the coefficients are:

$$H_s(z) = 1 + 2z^{-1} + 3z^{-2} + 4z^{-3} \tag{2.8}$$

The comparison of both filters (2.6) and (2.8) shows that the order of the digital filter is higher in $\Sigma\Delta$ converters. In incremental ADCs, the weighting of the coefficients is

---

[3] The effect is the most important in $\Sigma\Delta$ converters when the constant input is close to mid- or full-scale.

[4] In opposition to the optimal filtering, discussed in a subsequent chapter

asymmetric, but does not overlap with previous samples.

Incremental converters are mostly used in instrumentation, for high-resolution but low frequency applications [8, 9].

2.6 Single/Multi-bit quantization

Single bit architectures use simple comparators to estimate the output of the integrators. Multi-bit topologies quantize on more than two levels.

Historically, single-bit quantizers were used in a wide majority of converters [10, 11]. These architectures were popular as they are simple - only a comparator and a DAC element are needed - and as the size of the digital (DAC management), expensive in old CMOS processes, was limited.

Multi-bit topologies are nowadays widely spread [12, 13, 14]. As multi-bit quantizers further reduce the quantization noise, the order of the modulator may be lower. Moreover, the quantization with several levels stabilizes high-order loops.
The main advantage of multi-bit DACs is the reduction of the power consumption, in particular if the input capacitors are shared between the signal and the reference[5].
The mismatch of the DAC capacitors is restrictive. Dynamic matching techniques help improving the resolution, but the linearity of multi-bit converters is lower than in single-bit architectures.

The selection between a single- and a multi-bit topology is a crucial decision as two key parameters - the resolution and the power consumption - are involved. High-performance, medium- to high-resolution converters (14-18 bits) are mainly implemented with multi-bit quantizers [15, 16]. Single-bit architectures are the most frequently used in ultra-high resolution (> 18 bits) converters. [17, 18]

2.7 Conclusions

In this chapter, basic concepts of $\Sigma\Delta$ modulation were firstly introduced - quantization, linear model and noise shaping. An overview of the different architectures was then provided to the reader to get a first knowledge of the advantages and drawbacks of each topology.

---

[5] The implementation with switched capacitors is detailed in chapter 4.

3 Optimal decoding

In the previous chapter, various architectural techniques based on the linear model of the modulator were introduced to reduce the quantization noise. This chapter is dedicated to the study of non-linear filters to improve the estimation of the input signal.

If the ADC is a sole ΣΔ modulator (without any extra quantization of the residue), all the quantization noise is located in the last integrator, after the last modulator cycle. After the last analog measurement of the residue it is no more possible to obtain extra information (apart from increasing the OSR), and the digital filters have to decode the digital output fluxes to determine the input.
Classic linear filters do consider the residue on the last integrator only as noise and thus assume a mid-range value to minimize the error (see sections 3.2 and 7.4). Practically, if the thermal noise is not strongly dominating, the residue is not uniform and is related to the measured digital fluxes. As a portion of the 'quantization noise' is held in the digital outputs, it is possible to improve the filtering quality to estimate at best the input signal value. A filtering considered as optimal decodes the outputs of the quantizers at best, without considering the final quantization error as a uniformly spread white noise.

The development of optimal filters for ΣΔ A-to-D converters is not a trivial task, due to the nonlinear nature of such modulators. Therefore, there is no generic optimal decoding algorithm compatible with various topologies and a large input diversity. Previous work addressed thus mainly the design of optimal filters for specific topologies. Most of them focus of first-order modulator with a constant input [19, 20, 21], with or without additional Gaussian noise on the input [22, 23]. The simple-loop second-order architecture was also analysed and an optimal filter derived [24].

The first section of this chapter presents the design of an optimal filter for a single order modulator with a constant input. This known case is detailed to get to know optimal filtering and to introduce a design methodology for such filters. The work on the cascade second-order modulator is then presented [25]. Various input signal are then studied in the following sections. An algorithm for optimal decoding with a sweep input is first designed [26] while the last part assumes a extra Gaussian noise on the input.

This work focuses on incremental ΣΔ converters, with a reset of the integrators before each data conversion. Some other work dealt with the optimal decoding of the standard ΣΔ modulator [27]. The choice of incremental converters is motivated firstly by the development of such a converter (chapter 7) and secondly by the eased filter design: if the initial value is known at the beginning of a conversion, the decoding algorithms are simplified and the results do not suffer from an extra uncertainty.

3.1 First order

Before starting the design of the optimal filter, a first very simple example is detailed to understand the interest of the optimal decoding compared to classic counter filters. In the case of a single bit modulator with a constant normalized input x between 0 and 1, the first output bit is equal to 0 if the input is between 0 and 0.5 and is set to 1 if x is between 0.5 and 1.

The value of the second bit is also set by the input x. As shown in Fig. 3.1, the code 00 is obtained if x is between 0 and 0.25, 01 if $x \in]0.25;0.5[$, 10 if $x \in]0.5;0.75[$ and 11 if $x \in]0.75;1[$.

The decoding of the fluxes 01 and 10 by the counter filter shows clearly a loss of information. The counter filter assigns a unique output value $\hat{x}$=0.5 for all x between 0.25 and 0.75, while it is known, after the first bit, if x is higher or lower than 0.5.

The aim of optimal filtering is not to loose information located in the history of the output flux and to determine the admissible input range of all output fluxes. The estimates $\hat{x}$ of the input x of this simple example are also shown in Fig. 3.1.

It is possible to describe any sigma-delta modulator with a system of recursive equations, including the initial conditions: one equation for each integrator and one comparison function for each quantizer.

The first-order incremental modulator with a constant input, Fig. 3.2, is described using the following system:

$$\left[\begin{array}{lcl} y(0) & = & x, \\ y(k) & = & x + y(k-1) - bit0(k-1), \\ bit0(k) & = & \begin{cases} 1 & \text{if } y(k) \geq 1/2, \\ 0 & \text{if } y(k) < 1/2. \end{cases} \end{array}\right. \tag{3.1}$$

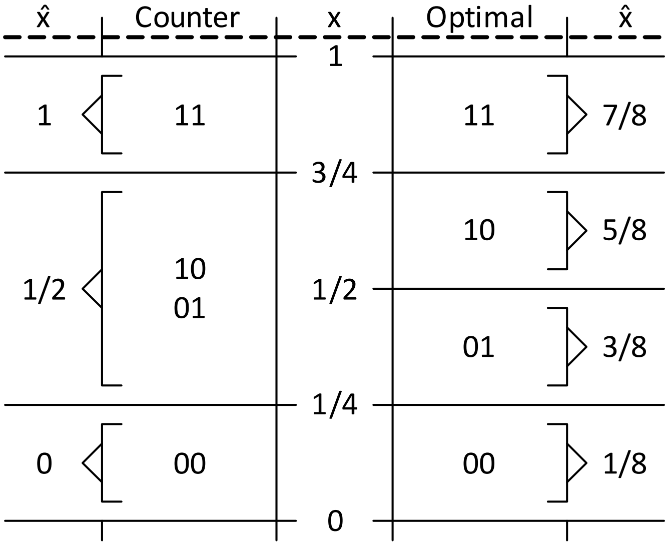

Figure 3.1: Comparison of the counter and optimal filters for an OSR of two bits.

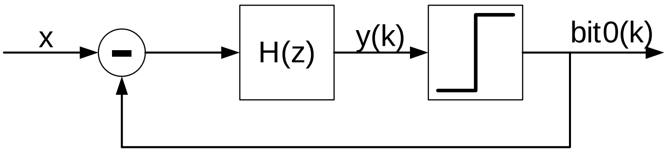

Figure 3.2: First order modulator.

It is first important to show that, if the modulator is not saturated (i.e. $x \in]0;1[$), the output of the integrator $y[k+1]$ is bounded between -0.5 and 1.5:

$$\begin{aligned} &\text{i) } y(k) \in \,]\text{-}0.5;0.5[\\ &\quad => bit0(k) = 0, \\ &\quad => y(k) - bit0(k) \in \,]\text{-}0.5;0.5[, \\ &\quad => y(k+1) \in \,]\text{-}0.5;1.5[\qquad\qquad \forall x \in \,]0;1[. \end{aligned} \tag{3.2}$$

$$\begin{aligned} &\text{ii) } y(k) \in \,]0.5;1.5[\\ &\quad => bit0(k) = 1, \\ &\quad => y(k) - bit0(k) \in \,]\text{-}0.5;0.5[, \\ &\quad => y(k+1) \in \,]\text{-}0.5;1.5[\qquad\qquad \forall x \in \,]0;1[. \end{aligned} \tag{3.3}$$

And thus $y(k) \in \,]\text{-}0.5;1.5[\ \forall k$.

Another term, used in later mathematical developments, is also bounded:

$$\begin{aligned}
&\text{as } y(k) - bit0(k) \in \,]\text{-}0.5;0.5[,\\
&=> y(k) - bit0(k) + 0.5 \in \,]0;1[,\\
&=> y(k) - bit0(k) + 0.5 = \left[y(k) - bit0(k) + 0.5\right]\%1,
\end{aligned} \tag{3.4}$$

where % is the modulus operator. $[a]\%b$ is a modulus b. In particular $[a]\%1$ is the decimal part of a.

To find a non-recursive equation for $bit0(k)$ it is possible to proceed as follow:

$$\begin{aligned}
&\text{As } y(k) \in \,]\text{-}0.5;1.5[,\\
&bit0(k) = round\left\{y(k)\right\},\\
&\qquad\quad = round\left\{x + y(k-1) - bit0(k-1)\right\},\\
&\qquad\quad = round\left\{x + \left[y(k-1) - bit0(k-1) + 0.5\right] - 0.5\right\},\\
&\qquad\quad = round\left\{x + \left[y(k-1) - bit0(k-1) + 0.5\right]\%1 - 0.5\right\},\\
&\qquad\quad = round\left\{x + \left[kx - \sum_{i=0}^{k-1} bit0(i) + \frac{1}{2}\right]\%1 - \frac{1}{2}\right\},\\
&\qquad\quad = round\left\{x + \left[kx + \frac{1}{2}\right]\%1 - \frac{1}{2}\right\}.
\end{aligned} \tag{3.5}$$

Starting with this first non recursive equation, it is possible to develop alternative notations, to have a better view on the involved terms:

$$\begin{aligned}
bit0(k) &= round\left\{x + \left[kx + \frac{1}{2}\right]\%1 - \frac{1}{2}\right\},\\
&= x + \left[kx + \frac{1}{2}\right]\%1 - \left[x + \left[kx + \frac{1}{2}\right]\%1\right]\%1,\\
&= x + \left[kx + \frac{1}{2}\right]\%1 - \left[(k+1)x + \frac{1}{2}\right]\%1.
\end{aligned} \tag{3.6}$$

$$\begin{aligned}
bit0(k) &= x + \left[kx + \frac{1}{2}\right]\%1 - \left[(k+1)x + \frac{1}{2}\right]\%1,\\
&= kx - kx + \frac{1}{2} - \frac{1}{2} + x - \left(-\left[kx + \frac{1}{2}\right]\%1\right) - \left[(k+1)x + \frac{1}{2}\right]\%1,\\
&= (k+1)x + \frac{1}{2} - \left[(k+1)x + \frac{1}{2}\right]\%1 - \left(kx + \frac{1}{2} - \left[kx + \frac{1}{2}\right]\%1\right),\\
&= round\{(k+1)x\} - round\{kx\}.
\end{aligned} \tag{3.7}$$

3.1.1 Optimal filter – thresholds

In a simple configuration, first-order with a constant input, the first goal of the optimal decoding is to determine the possible input range corresponding to a given output flux. Starting with a full covering of the input dynamic, it is possible to reduce the possible input range with each new bit of the digital output stream.
The particular values of the input x for which the state of the comparator changes are named thresholds. Knowing the value of the thresholds allows to narrow the possible input set, knowing at each iteration the output bit $bit0[k]$.

From (3.7) it is possible to calculate the inputs levels x_{th} corresponding to the boundaries between two output codes.

$$\begin{aligned}
&\text{a) } kx + 0.5 = m\,, \quad m \in \{1, 2, \dots, k\} \\
&\quad x_{th(k,m)} = \frac{m - 0.5}{k} \\
&\text{b) } kx + 0.5 = n\,, \quad n \in \{1, 2, \dots, k+1\} \\
&\quad x_{th(k,n)} = \frac{n - 0.5}{k+1}
\end{aligned} \tag{3.8}$$

The state of the comparator is modified only if each threshold is unique. A threshold is double is a solution of (3.8-a) is also a solution of (3.8-b):

$$\begin{aligned}
&\frac{m - 0.5}{k} = \frac{n - 0.5}{k+1} \\
&(2m - 1)(k+1) = (2n - 1)(k).
\end{aligned} \tag{3.9}$$

The equation (3.9) has no solution as the left part is always odd while the right term is always even.

The Fig. 3.3 is a graphical representation of the thresholds for the first-order modulator. The thresholds for the first 16 bits are displayed. Each column is divided in light and dark areas, according to the value of $bit0[k]$.

For a given output code, it is possible to find the corresponding input range with the following algorithm:

- a) For the first output bit, initialize the two boundaries for x: $x_{min} = 0$, $x_{max} = 1$.
- b) Determine the thresholds for the current bit.
- c) If there is any threshold between the current boundaries, update x_{min} and x_{max}.
- d) Back to point b) with the next bit.

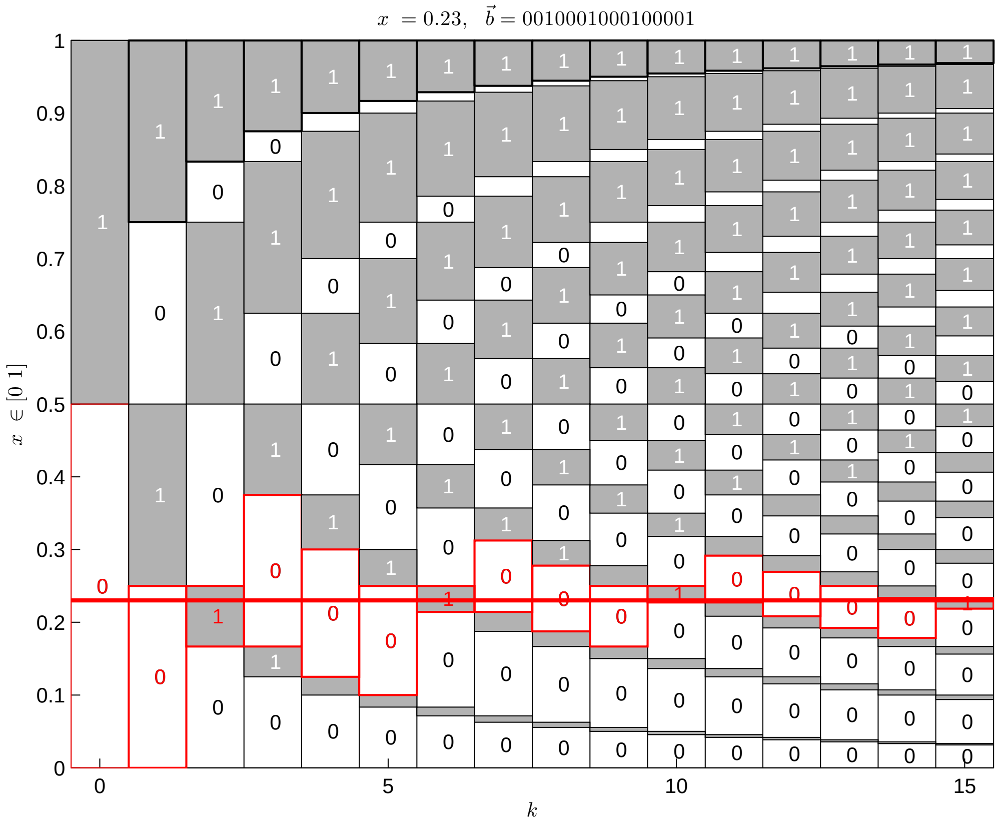

Figure 3.3: Thresholds for the 16 first bits of the first order modulator.

To update the input space, (item c), it is firstly necessary to determine if the threshold $x_{th(k,n)}$ crosses the input space. The intersection occurs if the following condition is fulfilled:

$$\left(x_{max} - x_{th(k,n)}\right)\left(x_{min} - x_{th(k,n)}\right) < 0 \tag{3.10}$$

If the threshold $x_{th(k,n)}$ splits the set of solutions in two parts, an update is required. According to the value of $bit0(k)$, the new admitted input space will be either from $x_{th(k,n)}$ to x_{max} or from x_{min} to $x_{th(k,n)}$.

An example of the update of the input range is provided in Fig. 3.4. The lower bound x_{min} and upper bound x_{max} for an input signal x are closer than the one obtained with a classic filter (counter). In this example, the optimal decoder locates the input between 0.22727 and 0.23333 while the classic filter provides a value between 0.21875 and 0.28125.

Once the space of all possible inputs is defined, the estimation of the best input in this set has to be computed. A common method is to minimize the squared error on

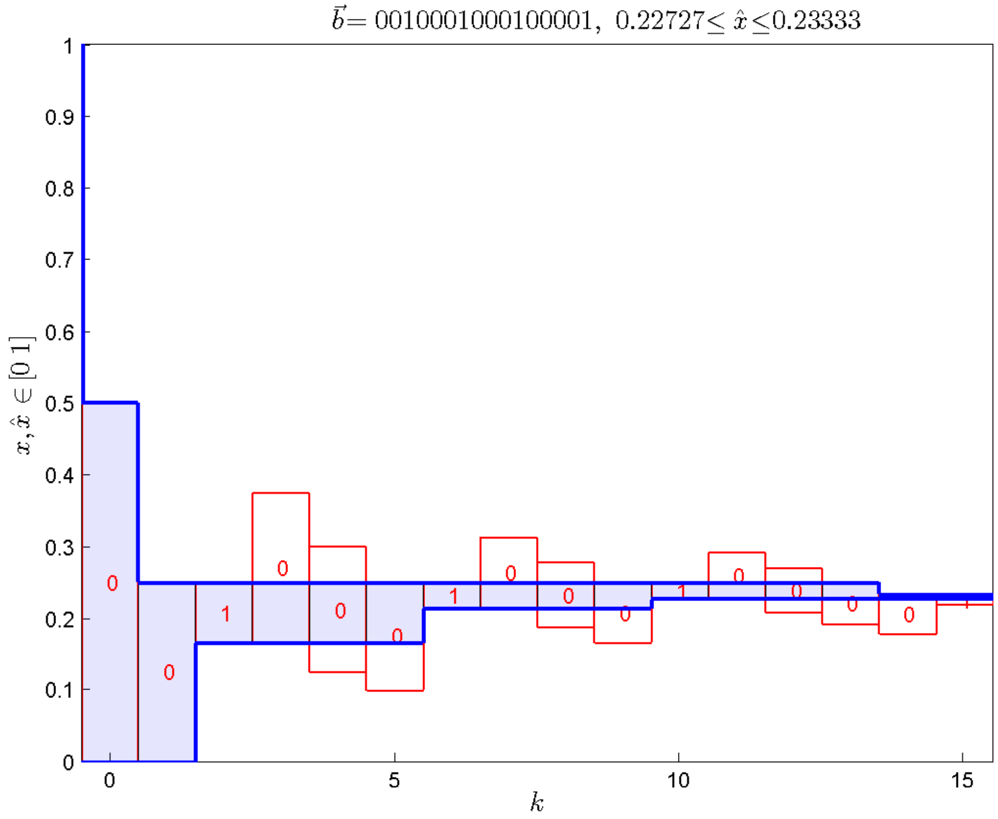

Figure 3.4: Example of an update of the input space. After each bit, the possible input space (blue) is compared to the bit[k] thresholds (red boxes) and updated accordingly

the whole surface (RMS criterion). The result for a continuous one-dimension range is simply the algebraic mean:

$$\hat{x} = \frac{x_{min} + x_{max}}{2} \tag{3.11}$$

The equation (3.11) assumes a uniform probability distribution of the input x on the whole range between 0 and 1.

The transfer function of the first-order modulator for an output code with a length $nbits = 4$ is displayed in Fig. 3.5. The transfer function using the counter filter is also displayed to compare.

<u>Practical considerations:</u> The aforementioned algorithm is based on the comparison of thresholds contained between two extreme values, given by the saturation of the modulator; in a normalized case, 0 and 1. Besides the necessity of storing decimal values, some operations (3.10) require a computation of differences between two floating-point numbers. Theses differences can be very small, especially for high over

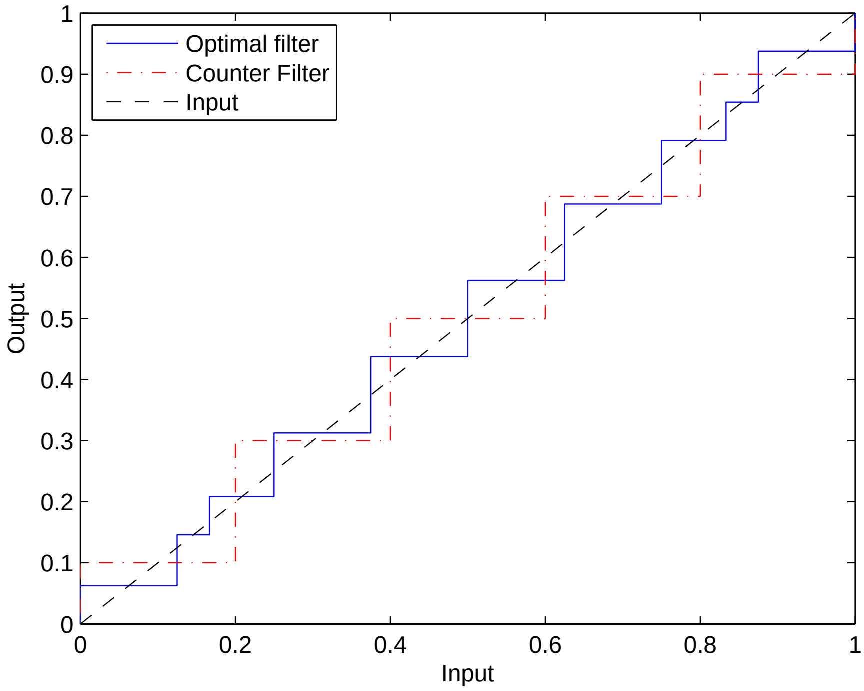

Figure 3.5: Transfer function using counter filter (dash-dot line) and optimal filter (solid line) for $nbits = 4$.

sampling ratios (OSR). A floating point computing unit is thus required, as well as a good numerical precision for the value of each threshold.

These last factors being expensive in computing time and registers, it is advantageous to work with integer numbers. Each threshold described in (3.23), as well as the two initial limits 0 and 1, are fractional numbers. Therefore, they can be substituted by two integers nu and de. The thresholds are fractional numbers because k, n and p are integers.

$$x_{th} = \frac{nu}{de} \tag{3.12}$$

The problematic computation of the differences (3.10) can also be substituted by an comparison of two integers. The condition to update the input space becomes, for a threshold i:

$$(nu_{max}de_i - nu_i de_{max})(nu_{min}de_i - nu_i de_{min}) < 0 \tag{3.13}$$

The estimate of the final value of the input $\hat{x}$ (3.11) can also be expressed with a fraction. This last result is of lesser importance as the final numerical precision is restricted by the width of the converter digital output bus.

FOM: In order to be able to evaluate the quality of this algorithm, it is necessary to define a figure of merit, which will be, in this case, a mean resolution (3.14). It is necessary to compute a mean resolution as, unlike the counter filter, the resolution is not constant. An RMS criterion is applied:

$$\mathrm{FOM} = \mathrm{ENOB} = -log_2\left(\sqrt{12\sum_{int} err_{int}^2}\right) \tag{3.14}$$

where err_{int} is the mean error of each interval int.

$$err_{int} = \sqrt{\int_{int_{min}}^{int_{max}} (x - \hat{x}_{int})^2\, dx} \tag{3.15}$$

The normalization factor 12 is added to guarantee that this computation of the ENOB, applied on the classic linear filter provides a resolution of $1/nbits$ for a code of length $nbits$.

The evolution of the resolution with the OSR is shown in Fig. 3.6. The resolution of the counter filter is also displayed.

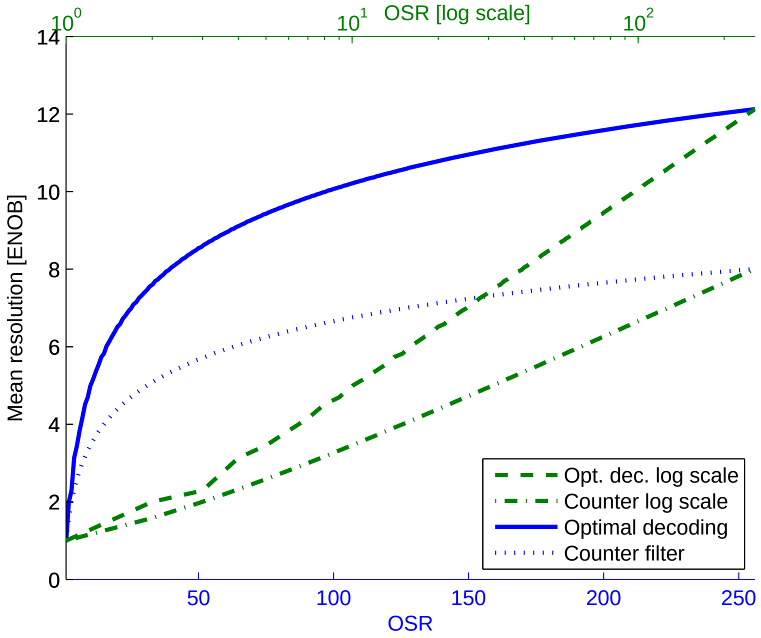

Figure 3.6: Mean resolution using counter filter (dotted line) and optimal filter (solid line). The dashed lines are referenced to the upper logarithmic scale.

To generate this result, all the thresholds have been computed for each OSR, using a dedicated C++ code. After the computation of the equivalent resolution, Matlab® was used for plotting.

3.2 Second-order – MASH 1-1

As the study of the optimal filtering of the single-loop second-order modulator is already covered by another work [24], this section only covers the cascade MASH 1-1 structure.

A MASH 1-1 modulator is made of two first-order loops, as shown in Fig. 3.7. The aim of the second loop is to estimate the quantization error of the first one.

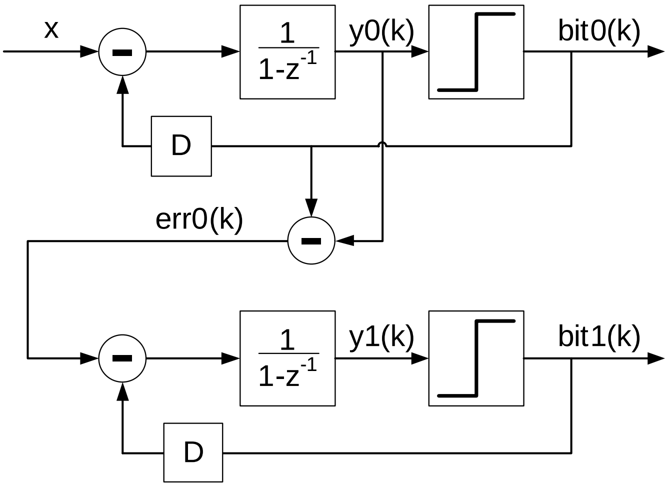

Figure 3.7: Block diagram of a MASH11 ΣΔ modulator with a constant input.

The design of the optimal filter for the MASH 1-1 follows the same approach than for the first-order ΣΔ ADC, starting with the recursive equations of the modulator, defining then the thresholds and finally estimating the input value. The system of recursive equations for the MASH 1-1 modulator is:

$$\left[\begin{array}{lcl} y0(0) & = & x, \\ y0(k) & = & x + y(k-1) - bit0(k-1), \\ bit0(k) & = & \begin{cases} 1 & \text{if } y0(k) \geq 1/2, \\ 0 & \text{if } y0(k) < 1/2, \end{cases} \\ err(k) & = & bit0(k) - y0(k) + 1/2, \\ y1(k) & = & err(k) + y(k-1) - bit1(k-1), \\ bit1(k) & = & \begin{cases} 1 & \text{if } y1(k) \geq 1/2, \\ 0 & \text{if } y1(k) < 1/2. \end{cases} \end{array}\right. \tag{3.16}$$

As well as $y0[k] - bit0[k]$ is between -0.5 and 0.5, $y1[k] - bit1[k]$ is also bounded between -0.5 and 0.5. This is true because the error $err[k]$ is bounded between 0 and 1 for all k.

It is possible to developp the system (3.16) to express $bit1(k)$ with a non-recursive notation:

$$
\begin{aligned}
&\text{As } y1(k) \in \,]\text{-}0.5;1.5[,\\
bit1(k) &= round\left\{y1(k)\right\},\\
&= round\left\{err(k) + y1(k-1) - bit1(k-1)\right\},\\
&= round\{y0(k) - bit0(k) + 0.5\\
&\qquad + y1(k-1) - bit1(k-1) + 0.5 - 0.5\},\\
&= round\{[y0(k) - bit0(k) + 0.5]\,\%1\\
&\qquad + [y1(k-1) - bit1(k-1) + 0.5]\,\%1 - 0.5\},\\
&= round\left\{\left[(k+1)x - \sum_{i=0}^{k} bit0(i) + \frac{1}{2}\right]\%1\right.\\
&\qquad \left. + \left[\sum_{i=0}^{k-1}\left(y0(i) + \frac{1}{2} - bit0(i) - bit1(i)\right) + \frac{1}{2}\right]\%1 - \frac{1}{2}\right\},\\
&= round\left\{\left[(k+1)x + \frac{1}{2}\right]\%1 + \left[\sum_{i=0}^{k-1}\left(y0(i) + \frac{1}{2}\right) + \frac{1}{2}\right]\%1 - \frac{1}{2}\right\},\\
&= round\left\{\left[(k+1)x + \frac{1}{2}\right]\%1 + \left[\sum_{i=0}^{k-1}\left(y0(i)\right) + \frac{k+1}{2}\right]\%1 - \frac{1}{2}\right\},\\
&= round\left\{\left[(k+1)x + \frac{1}{2}\right]\%1\right.\\
&\qquad \left. + \left[\sum_{i=0}^{k-1}\left((i+1)x - \sum_{j=0}^{i-1} bit0(j-1)\right) + \frac{k+1}{2}\right]\%1 - \frac{1}{2}\right\},\\
&= round\left\{\left[(k+1)x + \frac{1}{2}\right]\%1 + \left[\sum_{i=0}^{k-1}(i+1)x + \frac{k+1}{2}\right]\%1 - \frac{1}{2}\right\},\\
&= round\left\{\left[(k+1)x + \frac{1}{2}\right]\%1 + \left[\frac{k(k+1)}{2}x + \frac{k+1}{2}\right]\%1 - \frac{1}{2}\right\}
\end{aligned}
\tag{3.17}
$$

to have a better view of the important terms, this last expression for $bit1(k)$ can be written with the following alternative notations:

$$\begin{aligned}
bit1(k) = round&\left\{\left[(k+1)x+\frac{1}{2}\right]\%1+\left[\frac{k(k+1)}{2}x+\frac{k+1}{2}\right]\%1-\frac{1}{2}\right\},\\
=&\left[(k+1)x+\frac{1}{2}\right]\%1+\left[\frac{k(k+1)}{2}x+\frac{k+1}{2}\right]\%1\\
&-\left[\left[(k+1)x+\frac{1}{2}\right]\%1+\left[\frac{k(k+1)}{2}x+\frac{k+1}{2}\right]\%1\right]\%1,\\
=&\left[(k+1)x+\frac{1}{2}\right]\%1+\left[\frac{k(k+1)}{2}x+\frac{k+1}{2}\right]\%1\\
&-\left[(k+1)x+\frac{1}{2}+\frac{k(k+1)}{2}x+\frac{k+1}{2}\right]\%1,\\
=&\left[(k+1)x+\frac{1}{2}\right]\%1+\left[\frac{k(k+1)}{2}x+\frac{k+1}{2}\right]\%1\\
&-\left[\frac{(k+1)(k+2)}{2}+\frac{k}{2}\right]\%1
\end{aligned}\tag{3.18}$$

$$\begin{aligned}
bit1(k) =&\left[(k+1)x+\frac{1}{2}\right]\%1+\left[\frac{k(k+1)}{2}x+\frac{k+1}{2}\right]\%1\\
&-\left[\frac{(k+1)(k+2)}{2}x+\frac{k}{2}\right]\%1,\\
=&-\left[\frac{(k+1)(k+2)}{2}x+\frac{k+1}{2}+\frac{1}{2}\right]\%1\\
&+\left[\frac{k(k+1)}{2}x+\frac{k+1}{2}\right]\%1+\left[(k+1)x+\frac{1}{2}\right]\%1,\\
=&\frac{(k+1)(k+2)}{2}x+\frac{k+1}{2}+\frac{1}{2}-\left[\frac{(k+1)(k+2)}{2}x+\frac{k+1}{2}+\frac{1}{2}\right]\%1\\
&-\left(\frac{k(k+1)}{2}x+\frac{k+1}{2}-\left[\frac{k(k+1)}{2}x+\frac{k+1}{2}\right]\%1\right)\\
&-\left((k+1)x+\frac{1}{2}-\left[(k+1)x+\frac{1}{2}\right]\%1\right),\\
=&\,round\left\{\frac{(k+1)(k+2)}{2}x+\frac{k+1}{2}\right\}\\
&-round\left\{\frac{k(k+1)}{2}x+\frac{k}{2}\right\}-round\left\{(k+1)x\right\}
\end{aligned}\tag{3.19}$$

As the first stage is a first order modulator, its output is identical to the one developed in the last section (3.7):

$$bit0(k) = round\{(k+1)x\} - round\{kx\} \tag{3.20}$$

Linear filter

Before designing the optimal filter, a succinct reminder of the classic filtering of incremental ADCs with linear filters is provided here.

At the end of a conversion, the information of the value of the input x is mainly held in the two output codes ($bit0/1$), and, to a lesser extent, in the second stage integrator. The residual quantization error is given by (3.21):

$$err1(k) = bit1(k) - y1(k) + 1/2. \tag{3.21}$$

In a classic scheme, this error is considered as a quantization noise. The statistical distribution of this error after the last cycle is thus assumed to be constant in the allowed error range (between 0 and 1). When substituting this error by its mean value (0.5), it is possible to compute an estimate of the input, performing a weighted sum of the output $bit0$ and $bit1$. The relative coefficients of this sum are shown in fig. 3.8.

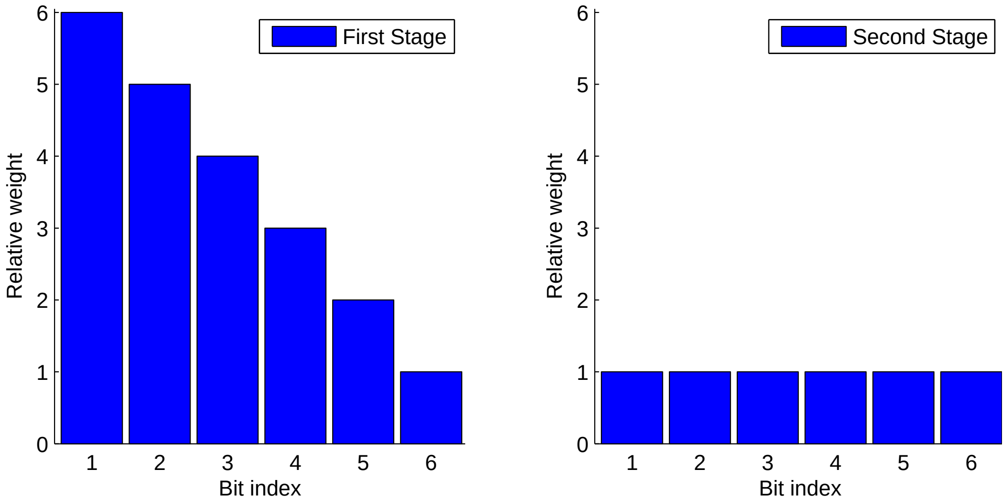

Figure 3.8: Linear filter relative coefficients for the first (left) and second (right) stage for an over-sampling ratio of $nbits = 6$.

Optimal filter

This section follows the same procedure than the one for the first order modulator. The aim of the optimal decoder is, here as well, to determine the set of possible inputs, corresponding to given output fluxes $bit0$ and $bit1$. In a $\Sigma\Delta$ incremental ADC with a constant input, this set of inputs is contained in an 1D space bounded by two limits x_{min} and x_{max} (i.e. x_{min}, resp. x_{max}, is the minimal, resp. maximal, value of the constant input x for which the given output code is obtained).

The algorithm is sequentially processing each bit of each stage to update the input range.

The comparison between the first order and the second order topologies clearly shows that, for a same input x, the outputs codes $bit0$ are identical. It is thus possible to perform the decoding in two steps:

- Decode the first stage as developed in section 3.1.
- Process the output flow of the second stage to update the two bounds provided by the first stage.

The decoding procedure is summarized in fig. 3.9.

As the processing of the first stage was previously described and as the result, given by the two limits x_{min} and x_{max} after the processing of the first stage, can be used to initialize the decoding of the second stage, the description of the optimal filter for the MASH 1-1 modulator is more detailed for the second stage than for the first one.

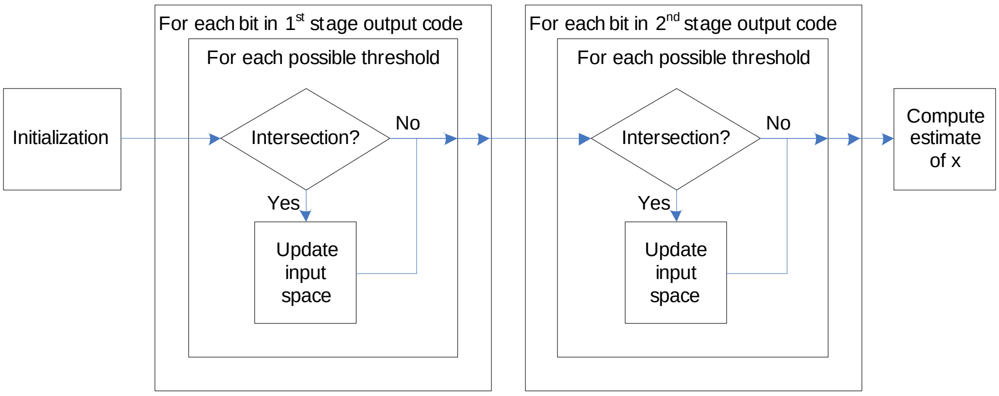
Initialization
For each bit in 1st stage output code
For each possible threshold
Intersection?
No
Yes
Update input space
For each bit in 2nd stage output code
For each possible threshold
Intersection?
No
Yes
Update input space
Compute estimate of x

Figure 3.9: Optimal filter decoding flow.

Initialization

The set of possible inputs is first initialized with the whole set of inputs which are not saturating the modulator. In a normalized case, $x_{min} = 0$ and $x_{max} = 1$.

Thresholds

As the thresholds and the processing of the first stage were already defined in the last section, there is no need to rewrite them here. The new thresholds, introduced with the second stage of the MASH 1-1 structure, are crossed when the decimal part of one of the three terms in the round functions of equation (3.19) is equal to one half.

$$\begin{aligned}
&\text{a) } \frac{1}{2} = [(k+1)x_{th}]\,\%1\\
&\text{b) } \frac{1}{2} = \left[\frac{k(k+1)}{2}x_{th} + \frac{k}{2}\right]\%1\\
&\text{c) } \frac{1}{2} = \left[\frac{(k+1)(k+2)}{2}x_{th} + \frac{k+1}{2}\right]\%1.
\end{aligned} \tag{3.22}$$

The thresholds are obtained, inverting the equations of (3.22):

$$\begin{aligned}
&\text{a) } x_{th} = \frac{2m-1}{2k+2}, \quad m \in \{1,2,\dots,k+1\}\\
&\text{b) } x_{th} = \begin{cases} \dfrac{2n}{k(k+1)}, & n \in \{1,2,\dots,k(k+1)/2-1\}, \quad \text{odd } k\\ \dfrac{2n-1}{k(k+1)}, & n \in \{1,2,\dots,k(k+1)/2\}, \quad \text{even } k \end{cases}\\
&\text{c) } x_{th} = \begin{cases} \dfrac{2p-1}{(k+1)(k+2)}, & p \in \{1,2,\dots,(k+1)(k+2)/2\}, \quad \text{odd } k\\ \dfrac{2p}{(k+1)(k+2)}, & p \in \{1,2,\dots,(k+1)(k+2)/2-1\}, \quad \text{even } k \end{cases}
\end{aligned} \tag{3.23}$$

As done for the single stage structure, the existence of double thresholds has to be checked. One has to check if a threshold of a) is a threshold of b), if a threshold of a) is a threshold of c) and the same for b) and c). The next six equations are comparing the threshold of a), b) and c) for odd and even k.

- odd k, i) solution of a is a solution of b:

  $$\begin{aligned}
  &\frac{2m-1}{2k+2} = \frac{2n}{k(k+1)}\\
  &(2m-1)k = 4n
  \end{aligned} \tag{3.24}$$

  no solution, left term is odd and right one is even.

- odd k, ii) solution of a is a solution of c:

  $$\begin{aligned}
  &\frac{2m-1}{2k+2} = \frac{2p-1}{(k+1)(k+2)}\\
  &(2m-1)(k+2) = 2(2p-1)
  \end{aligned} \tag{3.25}$$

  no solution, left term is odd and right one is even.

- odd k, iii) solution of b is a solution of c:

  $$\begin{aligned} \frac{2p-1}{(k+1)(k+2)} &= \frac{2n}{k(k+1)} \\ (2p-1)k &= 2n(k+2) \end{aligned} \tag{3.26}$$

  no solution, left term is odd and right one is even.

- even k, i) solution of a is a solution of b:

  $$\begin{aligned} \frac{2m-1}{2k+2} &= \frac{2n-1}{k(k+1)} \\ (2m-1)k &= 2(2n-1) \\ n &= mk/2 + 1/2 - k/4 \end{aligned} \tag{3.27}$$

  solutions only if $k \in \{2, 6, 10, \ldots\}$,

  moreover, $n \in \{1, 2, \ldots, k(k{+}1)/2\}, \forall m \in \{1, 2, \ldots, k{+}1\}$.

- even k, ii) solution of a is a solution of c:

  $$\begin{aligned} \frac{2m-1}{2k+2} &= \frac{2p}{(k+1)(k+2)} \\ 4p &= (2m-1)(k+2) \\ p &= mk/2 + m + 1/2 - k/4 \end{aligned} \tag{3.28}$$

  There is a solution only if $k \in \{2, 6, 10, \ldots\}$,

  moreover, $p \in \{1, 2, \ldots, (k{+}1)(k{+}2)/2-\}, \forall m \in \{1, 2, \ldots, k{+}1\}$.

- even k, iii) solution of b is a solution of c:

  $$\begin{aligned} \frac{2n-1}{k(k+1)} &= \frac{2p}{(k+1)(k+2)} \\ 2pk &= (2n-1)(k+2) \\ pk/2 &= nk/2 + n - k/4 - 1/2 \end{aligned} \tag{3.29}$$

  solutions only if $k \in \{2, 6, 10, \ldots\}$.

We can see that the set of solutions for (3.28) is identical to the set of solutions for (3.29). This mean that all bounds given by (3.22-a) are also bounds of (3.22-b) and (3.22-c). In order not to have double solution, but only triple solution, one have to prove that the set of solution for (3.29) is the same than the set of solution of (3.27).

First, one can replace k by j: $k = 4j-2,\ j \in \{1,2,3,...\}$. From (3.29):

$$\begin{aligned}
&\frac{2n-1}{k(k+1)} = \frac{2p}{(k+1)(k+2)}\\
&\Rightarrow (2n-1)(k+2) = 2pk\\
&\Rightarrow n(2j-1)+n-j = p(2j-1)\\
&\Rightarrow 2jn = 2jp-p+j\\
&\Rightarrow n = p-p/2j+1/2\\
&\text{solutions only if } p \in \left\{j,3j,5j,...,(2k+1)j\right\},\\
&\Rightarrow p = (2i-1)j,\ i \in \{1,2,...,k+1\}\\
&\Rightarrow n = (2i-1)j-(2i-1)/2+1/2\\
&\qquad = (2j-1)i-j+1\\
&\qquad = ik/2-k/4+1/2\\
&\text{which is exactly the set of solutions obtained in 3.27.}
\end{aligned} \tag{3.30}$$

Finally, all the bounds are unique except the one given by (3.22-a) which are also present in (3.22-b) and (3.22-c).

As the thresholds defined in (3.22-a) and (3.22-b) are subsets of the threshold of (3.22-c), only the last equation is kept. The threshold $x_{th(k,p)}$ are obtained inverting (3.22-c). The whole set of thresholds (3.31) related to the second stage is thus given, for the position k of the stream $bit1$, by:

$$x_{th(k,p)} = \begin{cases} \dfrac{2p-1}{(k+1)(k+2)}, & p \in \{1,2,...,(k+1)(k+2)/2\}, \quad \text{odd } k\\ \dfrac{2p}{(k+1)(k+2)}, & p \in \{1,2,...,(k+1)(k+2)/2-1\}, \quad \text{even } k \end{cases} \tag{3.31}$$

Update of the input set and computation of the estimate $\hat{x}$

Once the new thresholds are computed, the input space is updated, similarly than in the first order modulator, to determine the bounds x_{min} and x_{max}. The estimate $\hat{x}$ is, here as well, the average of the two bounds (3.11).

Figures 3.10 and 3.11 show the transfer functions and the quantization errors of the optimal and of the counter filters.

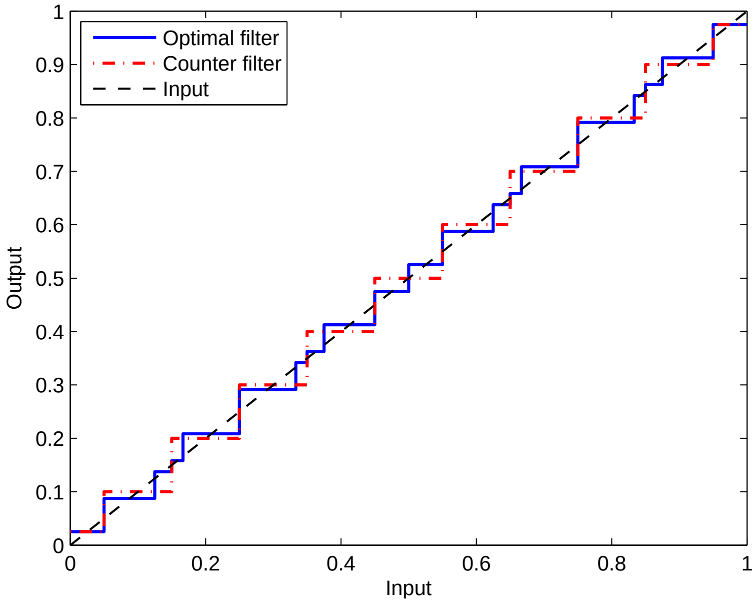

Figure 3.10: Transfer function of the MASH 1-1 optimal filter (solid), and of the linear filter (dash dots).

Specific practical considerations

In order to lighten the explanation and to ease the understanding of this paper, the presented algorithm for the MASH 1-1 structure processes first all the bits from the first stage, and then the bits from the second one. A better solution is to process alternately the bits from the two stages, i.e. process the bits in the following order: $bit0(0)$, $bit1(0)$, $bit0(1)$, $bit1(1)$, ...
In this implementation, the whole stream given by the output $bit1$ is not stored in memory and the computation load is spread on the whole acquisition period.

ENOB and results

The computation of the equivalent resolution is, here as well, normalized with 12 to guarantee a resolution of $1/nbits^2$ with the output of the linear filter (section 3.2).

$$\text{ENOB} = -log_2\left(\sqrt{12}\, MSE_x\right). \tag{3.32}$$

$$MSE_x = \frac{1}{I_{tot}}\sqrt{\int_{I_{tot}} (x-\hat{x}_x)^2 dx}. \tag{3.33}$$

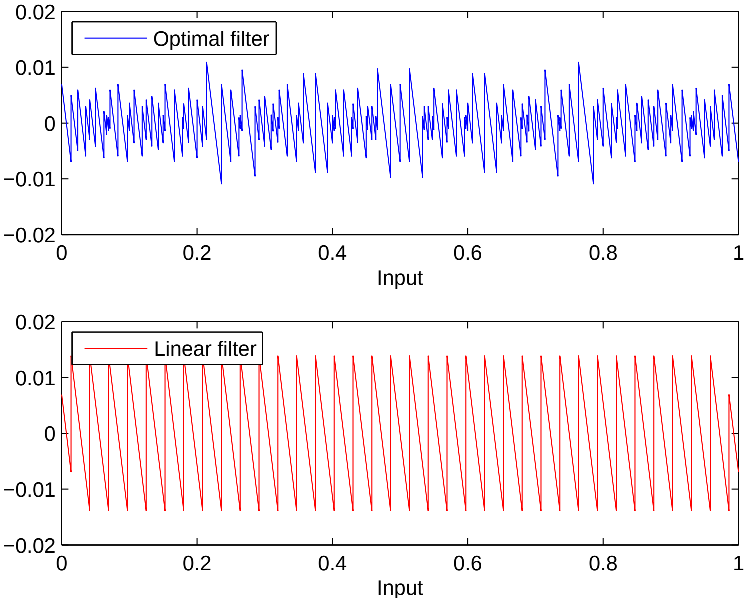

Figure 3.11: Quantization error using the MASH 1-1 optimal filter (top), and the linear filter (bottom) for $nbits = 8$.

As the full interval I_{tot} is a collection of smaller intervals I_i, the integral on I_{tot} can be split in a finite sum:

$$MSE_x = \frac{1}{I_{tot}} \sqrt{\sum_i \int_{I_i} (x - \hat{x}_i)^2 dx}. \tag{3.34}$$

The computation of each interval I_i can then be simplified, as $\hat{x}$ is uniquely defined for each i, where int_{min} and int_{max} are the lower and upper bounds of the interval I_i.

$$\begin{aligned} \int_{I_i} (x - \hat{x}_i)^2 dx &= \int_{int_{min}}^{int_{max}} \left(x - \frac{int_{min} + int_{max}}{2} \right)^2 dx \\ &= \frac{(int_{max} - int_{min})^3}{12}. \end{aligned} \tag{3.35}$$

In order to evaluate the quality of the provided optimal decoder, the mean resolution (3.32) as a function of the OSR is shown in Fig. 3.12. The mean resolution of the linear filter is also displayed to evaluate easily the resolution enhancement.

To conclude this section on optimal decoding of incremental modulators with a constant inputs, one can notice that the obtained mean resolution is much better than the one with linear filters. Even if the optimal decoding is harder to elaborate than a simple linear filter, it is still implementable in an embedded technology, as it

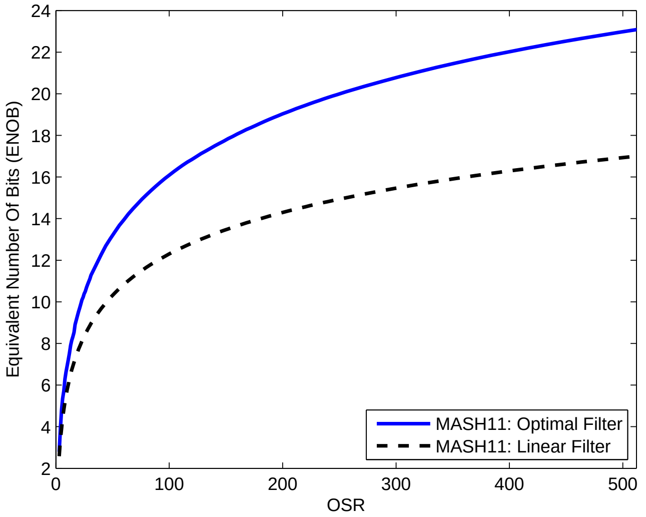

Figure 3.12: Mean resolution of the MASH 1-1 optimal filter (solid line), and of the linear filter (dashed line).

is only computing integer numbers. The main disadvantage of the aforementioned optimal filters is their poor robustness towards the fluctuations of the input signal. A small variation of the input or an extra noise will probably produce a code considered as invalid by the algorithm, that considers a strictly constant input.

3.3 Sweep input

An input signal varying linearly (Fig. 3.13) is describable using two parameters: its mean value x and its slope α.

The slope α has no physical unit; it is computed dividing the variation of the normalized signal amplitude by a discrete time unit. The particular values of x and α displayed in the graphs of this section were added to illustrate the decoding.

If a first-order modulator is driven by a time-varying input (fig. 3.14), it is no longer possible to use the previous optimal filters. Indeed, they operate perfectly only for well-suited codes (i.e. the codes provided by ideal modulators, fed with constant inputs).

The goal of this section is to develop an algorithm to determine a set of solutions for each given output code. The procedure developed to design optimal filters for constant inputs is reused, starting from the general equations defining the behavior of the modulator.

The starting point is the block schematic of the modulator, fig 3.14, and its description

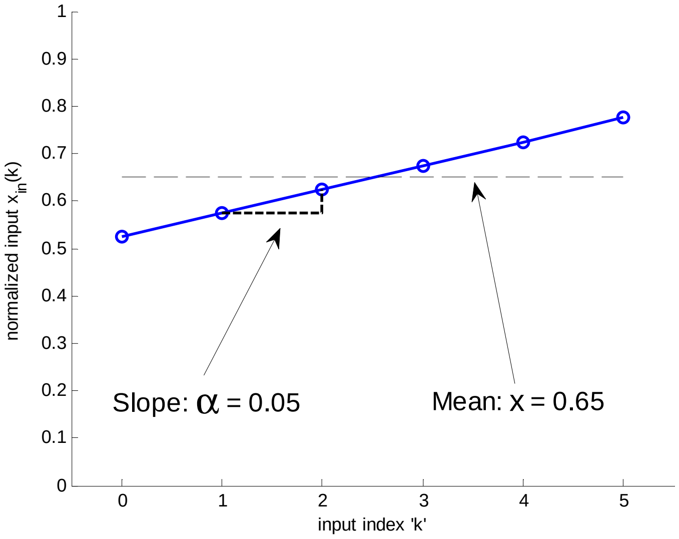

Figure 3.13: Sweep input: mean value and slope.

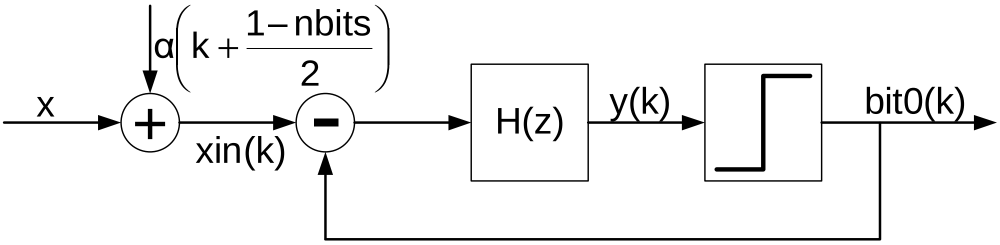

Figure 3.14: Block diagram of first-order ΣΔ modulator with sweep input.

using a system of equations (3.36).

$$\begin{cases} y(0) &= x_{in}(0), \\ y(k) &= x_{in}(k) + y(k-1) - bit0(k-1), \\ x_{in}(k) &= x + \alpha\left(k + \frac{1-nbits}{2}\right), \\ bit0(k) &= \begin{cases} 1 & \text{if } y(k) \geq 1/2, \\ 0 & \text{if } y(k) < 1/2. \end{cases} \end{cases} \tag{3.36}$$

In order to decode properly the output, the assumption that the modulator is not saturated is done, and that the input is bounded: $x_{in}(k) \in \,]0;1[\ \forall k$; thus:

$$\begin{cases} 0 < x < 1, \\ -\frac{1-|2x-1|}{nbits+1} < \alpha < \frac{1-|2x-1|}{nbits+1}. \end{cases} \tag{3.37}$$

Identically to the constant input case (3.3), $y(k)$ is also bounded: $y(k) \in\ ]\text{-}0.5;1.5[\ \ \forall k$.

For a first-order modulator, it is also possible to get the non-recursive equations for the system (3.36):

$$
\begin{aligned}
&\text{As } y(k) \in\ ]\text{-}0.5;1.5[,\\
bit0(k) &= round\left\{y(k)\right\},\\
&= round\left\{x_{in}(k) + y(k-1) - bit0(k-1)\right\},\\
&= round\left\{x_{in}(k) + \left[y(k-1) - bit0(k-1) + 0.5\right] - 0.5\right\},\\
&= round\left\{x_{in}(k) + \left[y(k-1) - bit0(k-1) + 0.5\right]\%1 - 0.5\right\},\\
&= round\left\{x_{in}(k) + \left[\sum_{i=0}^{k-1} x_{in}(i) - \sum_{i=0}^{k-1} bit0(i) + \frac{1}{2}\right]\%1 - \frac{1}{2}\right\},\\
&= round\left\{x_{in}(k) + \left[kx + \alpha\frac{k(k-nbits)}{2} + \frac{1}{2}\right]\%1 - \frac{1}{2}\right\},\\
&= round\left\{x + \alpha\left(k + \frac{1-nbits}{2}\right)\right.\\
&\qquad \left. + \left[kx + \alpha\frac{k(k-nbits)}{2} + \frac{1}{2}\right]\%1 - \frac{1}{2}\right\}.
\end{aligned}
\tag{3.38}
$$

In order to lighten the notation, the following substitution is used:

$$
z = x + \alpha\left(-\frac{nbits}{2} + \frac{1}{2}\right) \Leftrightarrow x = z + \alpha\left(\frac{nbits}{2} - \frac{1}{2}\right)
\tag{3.39}
$$

thus, with alternative notations

$$
\begin{aligned}
bit0(k) &= round\left\{z + \alpha k + \left[kz + \alpha\frac{k(k-1)}{2} + \frac{1}{2}\right]\%1 - \frac{1}{2}\right\},\\
bit0(k) &= z + \alpha k + \left[kz + \alpha\frac{k(k-1)}{2} + \frac{1}{2}\right]\%1\\
&\qquad - \left[(k+1)z + \alpha\frac{k(k+1)}{2} + \frac{1}{2}\right]\%1,\\
bit0(k) &= round\left\{(k+1)z + \alpha\frac{k(k+1)}{2}\right\} - round\left\{kz + \alpha\frac{k(k-1)}{2}\right\},\\
bit0(k) &= round\left\{\left[kz + \alpha\frac{(k-1)k}{2} + \frac{1}{2}\right]\%1\right.\\
&\qquad \left. - \left[(k+1)z + \alpha\frac{k(k+1)}{2} + \frac{1}{2}\right]\%1 + \frac{1}{2}\right\}.
\end{aligned}
\tag{3.40}
$$

The thresholds are then computed, as it is possible to calculate the inputs levels corresponding to the boundaries between two output codes with (3.40).

$$\begin{aligned}
&\text{a) } kz+\alpha\frac{k(k-1)}{2}+\frac{1}{2}=m\,, \quad m\in\{1,2,\dots,k\}\\
&z=\frac{2m-1}{2k}-\alpha\frac{k-1}{2}\\
&\text{b) } (k+1)z+\alpha\frac{k(k+1)}{2}+\frac{1}{2}=n\,, \quad n\in\{1,2,\dots,k+1\}\\
&z=\frac{2n-1}{2k-2}-\alpha\frac{k}{2}.
\end{aligned} \tag{3.41}$$

The existence of dual thresholds has to be checked $\forall x, m, n$, i.e. if a solution of (3.41-a) is also a solution of (3.41-b):

$$\frac{2m-1}{2k}-\alpha\frac{k-1}{2}=\frac{n-0.5}{k+1}. \tag{3.42}$$

Equation (3.42) has no solution because the lines corresponding the left and right solution in the αz plane never have the same slope[1].

Previously, with a constant input, each input set, that corresponded to an output code, could be represented by a bounded interval. In the case of a sweep input, the two independent parameters x and α must be taken into account. The whole input set is then described as a 2D plane with axes α and x.

3.3.1 Optimal decoder

The thresholds given by (3.41) are lines in this αx plane, as well as the limits of saturation of the modulator. Each input set can thus be described by a polygon. Moreover, from (3.41), a line j is entirely defined using its two parameters k_j and n_j.

Separation of the αx plane is illustrated in Fig. 3.15 for an output code with length three.

Each polygon is usually described using a list of coordinate representing the summits. As here the representation of the lines constituting the polygon is simpler, it is described by a set a lines.

The interest of describing any polygon with the equations of the sides, is that each side or line is only function of two integers n and k[2]. The order of the lines in the polygon is arbitrary chosen clockwise.

---

[1] The lines (thresholds) in the 2D plane are illustrated in Fig. 3.15.

[2] This statement is only valid in this particular case of decoding.

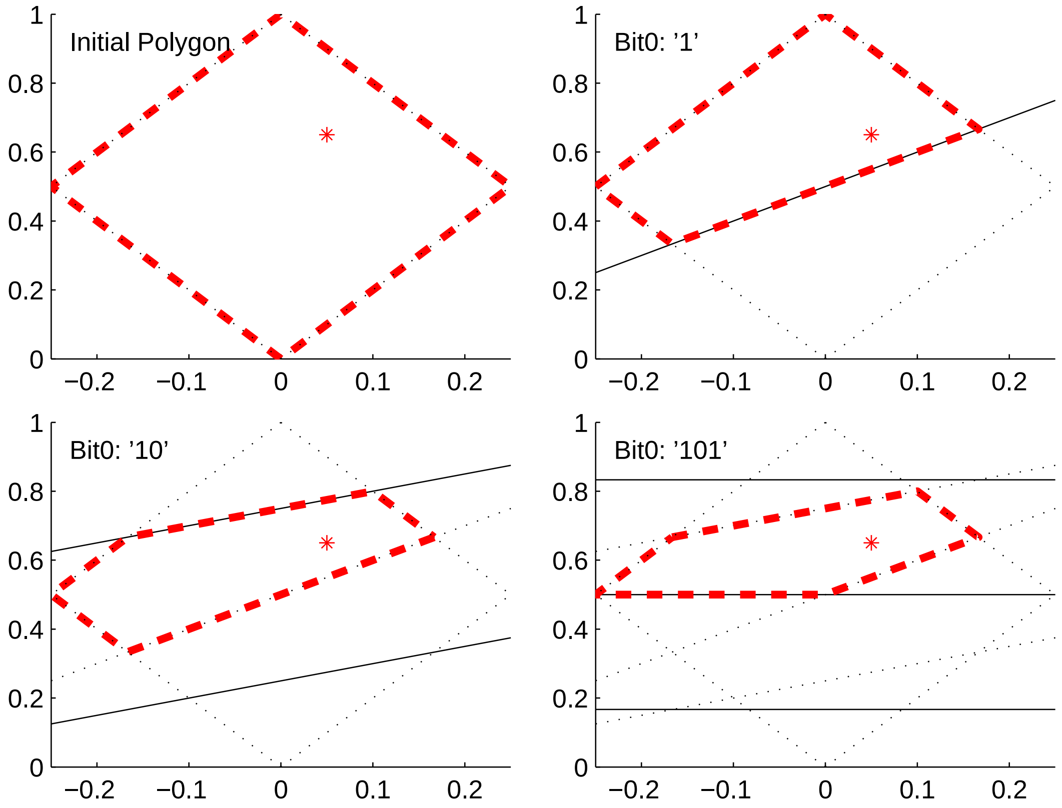

Figure 3.15: Iterative process to generate a polygon using thresholds, $nbits = 3$. Top left figure is the starting polygon, inside which the modulator is not saturated. Axes are α (horizontal) and x (vertical). Input has a slope $\alpha = 0.05$ and a normalized mean $x = 0.65$ (star).

Remark: In 3.41, the boundaries between two output codes were represented using two integer number, m and n. But, as each set of solution using m for a given bit k is the same set using n for the previous bit $k-1$, the line in the αz plane can be represented using only the n parameter.

The first step to decode the output bitstream is to determine the corresponding input range, described by a polygon. The procedure is summarized in Fig. 3.16.

Initialization

Before reading the first bit of the output code, all the inputs that do not saturate the modulator are accepted. The starting polygon is thus initialized with this set, represented in Fig. 3.15.

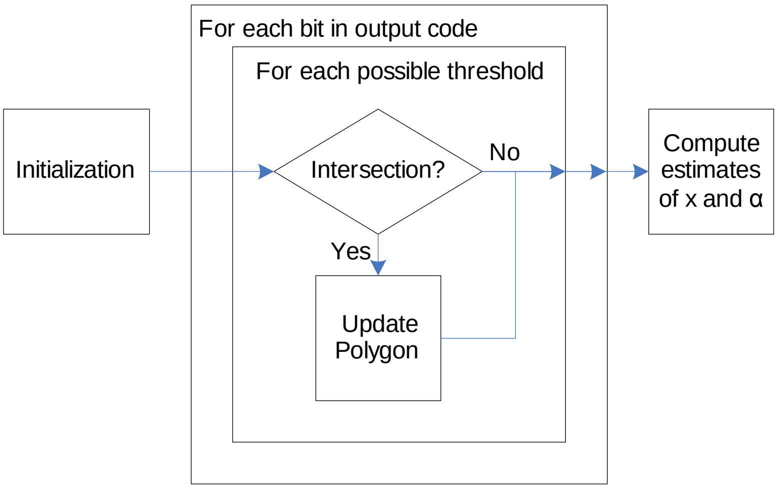
For each bit in output code
For each possible threshold
Initialization
Intersection?
No
Yes
Update Polygon
Compute estimates of x and α

Figure 3.16: Optimal filtering procedure.

Condition to update the polygon

Firstly, it is necessary to define the intersection point $(\alpha_{i,j}, x_{i,j})$ of a line i with another line j. The first coordinate of this point is:

$$\alpha_{ij} = \frac{(2n_j - 1)(k_i + 1) - (2n_i - 1)(k_j + 1)}{(k_i + 1)(k_j + 1)(k_j - k_i)}, \tag{3.43}$$

The second coordinate $x_{i,j}$ (or $z_{i,j}$), is easily obtained combining (3.41) and (3.43).

$$z_{ij} = \frac{(2n_i - 1)k_j(k_j + 1) - (2n_j - 1)k_i(k_i + 1)}{2(k_i + 1)(k_j + 1)(k_j - k_i)}, \tag{3.44}$$

where n_i and k_i are the corresponding value of the parameters n and k from (3.41) used to produce the line i.

Crossing of a line and a polygon: If the current threshold (line j), crosses the i^{th} segment of the polygon, the following condition is met:

$$(\alpha_{i-1,i} - \alpha_{i,j})(\alpha_{i,i+1} - \alpha_{i,j}) \leq 0. \tag{3.45}$$

Note: As the differences $\alpha_{i-1,i} - \alpha_{i,j}$ and $\alpha_{i,i+1} - \alpha_{i,j}$ can be very small when implemented on a real system, it is more interesting to replace this comparison of fractional numbers with an comparison of integers. It is possible because $2n-1$ and k are integer numbers.

Polygon update

If the threshold j crosses the lines p and n of the polygon, the update is done according to the k^{th} value of the output code $bit0(k)$

- either from p to n adding the line j at the end
- either from n to p adding the line j at the end

Note: It is possible to know if the line j crosses the polygon on one or more summits, i.e. if the condition (3.45) is satisfied respectively three or four times. In this particular case, an additional test is necessary to determine which polygon lines have to be kept.

Computation of the estimates $\hat{x}$ and $\hat{\alpha}$

The preceding points have determined the polygon including the full set of possible solutions in the αx plane. To find the estimate $(\hat{\alpha}, \hat{x})$, the mean square error MSE, which is computed on the whole surface of the polygon S_p, is minimized.

$$\mathrm{MSE}_{min} = min\left(\frac{1}{S}\sqrt{\int_S (x-\hat{x})^2 + (\alpha-\hat{\alpha})^2 dxd\alpha}\right) \tag{3.46}$$

As x and α are independent variables, the solution of the previous minimization is also the solution of the following system:

$$\left[\begin{array}{l} \dfrac{d}{d\hat{x}}\displaystyle\int_S (x-\hat{x})^2 + (\alpha-\hat{\alpha})^2\, dxd\alpha = 0 \\ \dfrac{d}{d\hat{\alpha}}\displaystyle\int_S (x-\hat{x})^2 + (\alpha-\hat{\alpha})^2\, dxd\alpha = 0 \end{array}\right. \tag{3.47}$$

and thus

$$\left[\begin{array}{l} \hat{x} = \dfrac{\int_S x\; dxd\alpha}{\int_S dxd\alpha} \\ \hat{\alpha} = \dfrac{\int_S \alpha\; dxd\alpha}{\int_S dxd\alpha} \end{array}\right. \tag{3.48}$$

As a polygon can be represented by a finite number of lines, and as the j^{th} line of the polygon is defined by its two parameters i_j and n_j, the previous expressions can be developed as follow (using the notation $N_j = 2n_j - 1$):

$$
\begin{aligned}
\int_S dx d\alpha &= \sum_j \int_{\alpha_{j-1,j}}^{\alpha_{j,j+1}} \int_0^{\frac{nbits-i_j}{2}\alpha+\frac{N_j}{2i_j}} dx d\alpha \\
&= \sum_j \int_{\alpha_{j-1,j}}^{\alpha_{j,j+1}} \frac{nbits-i_j}{2}\alpha + \frac{N_j}{2i_j} d\alpha \\
&= \sum_j \frac{nbits-i_j}{4}\left(\alpha^2_{\alpha_{j,j+1}} - \alpha^2_{\alpha_{j-1,j}}\right) + \frac{N_j}{2i_j}\left(\alpha_{\alpha_{j,j+1}} - \alpha_{\alpha_{j-1,j}}\right) \\
&= \sum_j \frac{i_{j+1}-i_j}{4}\alpha^2_{j,j+1} - \frac{1}{2}\left(\frac{N_{j+1}}{i_{j+1}} - \frac{N_j}{i_j}\right)\alpha_{j,j+1} \\
&= -\frac{1}{4}\sum_j \left(i_{j+1} - i_j\right)\alpha^2_{j,j+1}
\end{aligned}
\tag{3.49}
$$

$$
\begin{aligned}
\int_S x\, dx d\alpha &= \frac{1}{8}\sum_j \int_{\alpha_{j-1,j}}^{\alpha_{j,j+1}} \left((nbits-i_j)\alpha + \frac{N_j}{i_j}\right)^2 d\alpha \\
&= \frac{1}{24}\sum_j \left[2\left(i^2_{j+1}\text{-}i^2_j\right) - nbits\left(i_{j+1}\text{-}i_j\right)\right]\alpha^3_{j,j+1} - 3\left(N_{j+1}\text{-}N_j\right)\alpha^2_{j,j+1}
\end{aligned}
\tag{3.50}
$$

$$
\begin{aligned}
\int_S \alpha\, dx d\alpha &= \frac{1}{2}\sum_j \int_{\alpha_{j-1,j}}^{\alpha_{j,j+1}} (nbits-i_j)\alpha^2 + \frac{N_j}{i_j}\alpha\, d\alpha \\
&= -\frac{1}{12}\sum_j \left(i_{j+1} - i_j\right)\alpha^3_{j,j+1}
\end{aligned}
\tag{3.51}
$$

Figure of merit

Since one of the key points of the quality of a comparator is its resolution, the imprecision factor expressed as the MSE is first computed on the whole surface S_{tot}, which is the full set of possible input (fig. 3.15). As this surface is paved with polygons p, the integral on S_{tot} can be split into a finite sum (3.52).

$$
\begin{aligned}
MSE_x &= \frac{1}{S_{tot}}\sqrt{\int_{S_{tot}} \left(x - \hat{x}_{\alpha,x}\right)^2 dx d\alpha} \\
&= \frac{1}{S_{tot}}\sqrt{\sum_p \int_{S_p} \left(x - \hat{x}_p\right)^2 dx d\alpha}
\end{aligned}
\tag{3.52}
$$

The computation of the error on each polygon p can then be simplified, as $\hat{x}_p$ is uniquely defined for each polygon. The error contribution of each polygon is:

$$\begin{aligned}&\int_{S_p} \left(x-\hat{x}_p\right)^2 \, dx d\alpha = \\ &\sum_j \int_{\alpha_{j-1,j}}^{\alpha_{j,j+1}} \int_0^{\frac{nbits-k_j-1}{2}\alpha+\frac{N_j}{2k_j+2}} \left(x-\hat{x}_p\right)^2 \, dx d\alpha.\end{aligned} \tag{3.53}$$

The mean resolution, expressed in bits can then be defined as:

$$\text{FOM} = \text{ENOB} = -log_2\left(\sqrt{\frac{8}{nbits+1}}\, MSE_x\right) \tag{3.54}$$

The normalization factor $\sqrt{8/(nbits+1)}$ was added to guarantee that this FOM applied with the counter filter gives a resolution of $log_2(nbits+1)$.

Results

To evaluate qualitatively the optimal decoder for sweep inputs, the mean resolution (3.54) was computed for code lengths, corresponding to the oversampling ratio OSR, from 3 to 156, Fig. 3.17 .

The mean resolution of the output of the optimal filter for a constant input is also displayed. Therefore, overall performances of both cases can be compared.

The main improvement of this new algorithm is the larger range of possible input signals. Moreover, a precise estimation of the slope of the signal is available.

The development of an optimal filter suitable for sweep input allowed an improvement of the decoding filter algorithm for the first-order modulator. In addition to the estimation of a more complex input signal, the algorithm for sweep inputs is more robust as it covers a larger proportion of output codes.

3.4 Stochastic input

Even if the previous algorithms are relatively simple to implement, their main drawbacks are their sensitivity to noise and their robustness (i.e. an incomplete coverage of the whole set of the available output fluxes).

In this section, the optimal decoding of an incremental modulator with a noisy input is addressed. The modulator is shown in Fig. 3.18.

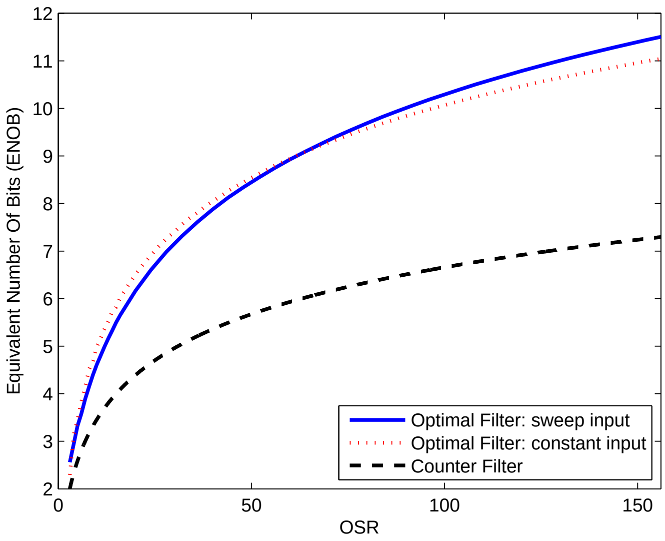

Figure 3.17: Mean resolution of the optimal filter with sweep input (solid line), and of the counter filter (dashed line).

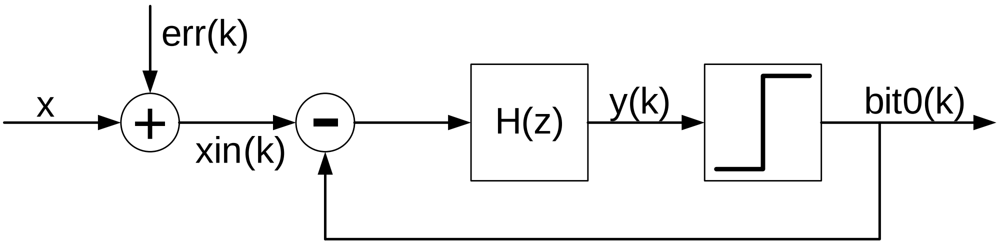

Figure 3.18: First order modulator with a noisy input.

Preliminary remark: The goal of the optimal decoding is to lower the ‘quantization noise’. If, in a given system, the thermal noise is greatly dominating the quantization noise, reducing the quantization noise will not improve the resolution of the converter. In this case, linear and optimal filters converge to comparable results [28], promoting the cheaper implementation of classic filters.

The signals and specific notations used in this section are listed thereafter:

- x is the constant input.
- $y(k)$ is the internal node at iteration k.
- $bit(k)$ is the k^{th} output bit, also written $\vec{b}_k$.
- $err(k)$ represents the noise, by convention added at the input of the modulator. By definition, $err(k)$ is a stochastic signal, so $err(k)$ is not correlated with $err(j)$, $\forall j,k$.
- $\varphi(s)$ is the probability density function of err. If err has a Gaussian distribution

with a zero mean, it can be written $err_\sigma(k)$, where σ is the standard deviation of the probability density function.

- $\vec{b}$ is a vector containing the n first output bits of the modulator.
- $P(A,B,C)$ represent the probability of an event A, function of parameters B and C.
- The symbol $\cap$ in the probability function is the 'AND' function. $P(A\cap B)$ is the the probability to have A and B at the same time.
- The symbol | is the 'knowing that' symbol. Especially if A and B are correlated, $P(A|B=r)$ is the probability to have A knowing the value of B.

Starting with the first bit of the output flux or the modulator, the probability that the bit is equal to zero for a constant input signal x and a noise distribution φ is:

$$\begin{aligned} P(bit(0)=0,x,\varphi) &= P\left(y(0)<0.5\right) \\ &= P\left(x+err(0)<0.5\right) \\ &= P\left(err(0)<0.5-x\right) \\ &= \int_{-\infty}^{0.5-x}\varphi(u)du \end{aligned} \tag{3.55}$$

If particular, for a Gaussian distribution

$$\begin{aligned} P(bit(0)=0,x,\sigma) &= P\left(err_\sigma(0)<0.5-x\right) \\ &= \frac{1}{\sigma\sqrt{2\pi}}\int_{-\infty}^{0.5-x} e^{\frac{-u^2}{2\sigma^2}}du. \end{aligned} \tag{3.56}$$

Then for the two first bits:

$$\begin{aligned} P(\vec{b}{=}00,x,\varphi) &= P\left(bit(0)=0\cap bit(1)=0,x,\varphi\right) \\ &= P\left(bit(0)=0\cap bit(1)=0|bit(0)=0,x,\varphi\right) \\ &= P\left(y(0)<0.5\cap y(1)<0.5|bit(0)=0,x,\varphi\right) \\ &= P\left(x+err(0)<0.5\cap 2x+err(0)+err(1)<0.5\right) \\ &= P\left(err(0)<0.5-x\cap err(0)+err(1)<0.5-2x\right). \end{aligned} \tag{3.57}$$

The probabilities for the other three possibilities are given by:

$$\begin{aligned} P(\vec{b}{=}01,x,\varphi) &= P\left(err(0)<0.5-x\cap err(0)+err(1)>0.5-2x\right), \\ P(\vec{b}{=}10,x,\varphi) &= P\left(err(0)>0.5-x\cap err(0)+err(1)<1.5-2x\right), \\ P(\vec{b}{=}11,x,\varphi) &= P\left(err(0)>0.5-x\cap err(0)+err(1)>1.5-2x\right). \end{aligned} \tag{3.58}$$

As $err(0)$ and $err(1)$ are not correlated, the last four results can be displayed in one graph, Fig. 3.19.

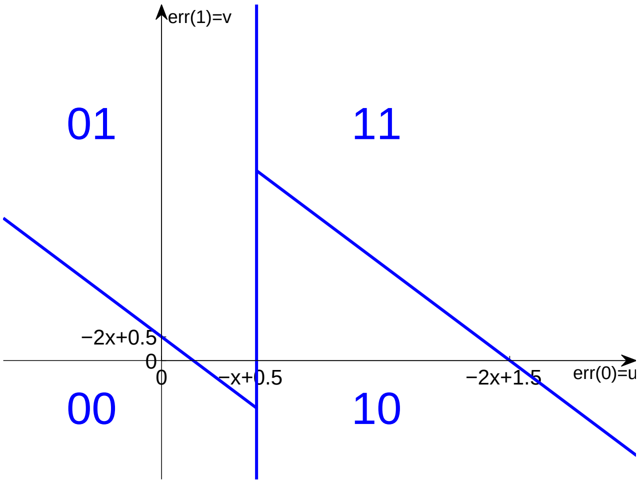

Figure 3.19: Graphical representation of the probability domains (integration surfaces) for the two-bits OSR. The input x is a parameter, shifting the blue lines. The axis u and v represent the uncorrelated error $err(0)$ and $err(1)$.

The probabilities being given by the integral of the density function, it is now possible to express them.

$$\begin{aligned}
P(\vec{b}{=}00, x, \varphi) &= \int_{-\infty}^{0.5-x} \int_{-\infty}^{0.5-2x-u} \varphi(v)\varphi(u) dv du,\\
P(\vec{b}{=}01, x, \varphi) &= \int_{-\infty}^{0.5-x} \int_{0.5-2x-u}^{\infty} \varphi(v)\varphi(u) dv du,\\
P(\vec{b}{=}10, x, \varphi) &= \int_{0.5-x}^{\infty} \int_{-\infty}^{1.5-2x-u} \varphi(v)\varphi(u) dv du,\\
P(\vec{b}{=}11, x, \varphi) &= \int_{0.5-x}^{\infty} \int_{1.5-2x-u}^{\infty} \varphi(v)\varphi(u) dv du.
\end{aligned} \tag{3.59}$$

Using the same procedure, probabilities for an output code $\vec{b}$ of length n are given by:

$$P(\vec{b}, x, \varphi) = \int_{tdo_1}^{tup_1} \int_{tdo_2}^{tup_2} \cdots \int_{tdo_n}^{tup_n} \varphi(s_n) \cdots \varphi(s_2)\varphi(s_1)\; ds_n \cdots ds_2 ds_1. \tag{3.60}$$

with, $\forall i \in \{1;2;...;n\}$,

$$\begin{aligned}
tup_i &= \begin{cases} \infty & \text{if } \vec{b}_i = 1 \\ 0.5 - ix + \sum_{k=1}^{i-1} \left(\vec{b}_k - s_k\right) & \text{if } \vec{b}_i = 0 \end{cases}\\
tdo_i &= \begin{cases} 0.5 - ix + \sum_{k=1}^{i-1} \left(\vec{b}_k - s_k\right) & \text{if } \vec{b}_i = 1 \\ -\infty & \text{if } \vec{b}_i = 0 \end{cases}
\end{aligned} \tag{3.61}$$

If the noise has a Gaussian distribution:

$$P(\vec{b},x,\sigma)=\left(\frac{1}{\sigma\sqrt{2\pi}}\right)^n\int_{tdo_1}^{tup_1}\int_{tdo_2}^{tup_2}\cdots\int_{tdo_n}^{tup_n}e^{\frac{s_1^2+s_2^2+\cdots+s_n^2}{-2\sigma^2}}\,ds_n\cdots ds_2ds_1. \tag{3.62}$$

An example is shown in Fig. 3.20 for a three-bits code (i.e. 8 possible codes) and a Gaussian noise with a standard deviation $\sigma=0.05$. The axes Fig. 3.20 are the input x (horizontal axis) and the probability to obtain the output codes (vertical axis).

As a probability density function is assigned to each output code, it is possible to get an estimate $\hat{x}$ of the input x, if the output code $\vec{b}$ and the noise distribution function φ are known. The minimization of the mean square error is selected to compute $\hat{x}$.

$$\begin{aligned}
&\frac{d}{d\hat{x}}\int_0^1 P(\vec{b},x,\varphi)(\hat{x}-x)^2\,dx=0\\
\Rightarrow&\int_0^1 P(\vec{b},x,\varphi)\frac{d}{d\hat{x}}(\hat{x}-x)^2\,dx=0\\
\Rightarrow&\int_0^1 P(\vec{b},x,\varphi)2(\hat{x}-x)\,dx=0\\
\Rightarrow&\ \hat{x}(\vec{b},\varphi)=\frac{\int_0^1 P(x,\vec{b},\varphi)x\,dx}{\int_0^1 P(x,\vec{b},\varphi)\,dx}.
\end{aligned} \tag{3.63}$$

Robustness: If the probability distribution φ is always positive in $\Re$, there is a non-zero probability for each code to be generated.
If this condition is fulfilled, it is possible to compute an estimate $\hat{x}$ with (3.63) for all output codes $\vec{b}$. An example of probability functions for code with a length $nbits=3$ is displayed in Fig. 3.20. A Gaussian noise distribution with a zero mean and a normalized standard deviation $\sigma=0.05$ is considered.

Generalized optimal decoding of a noiseless input: Comparing the precision and the robustness of the counter filter and of the noiseless optimal filter, one can notice that each filter has specific disadvantages.

- The counter filter has a complete coverage of all output codes, assigning a value to every code, but the mean resolution is poor.
- The optimal filter for a noiseless input shows a good resolution, but cannot deal with every output code. Particularly, if the OSR is 3 bits, it fails to decode the outputs $\vec{b}=100$ and $\vec{b}=011$.
- The optimal filter for a noisy input requires a prior knowledge of the shape and width of the dispersion.

If the case of the constant input is considered as a particular case of the noisy case (i.e.

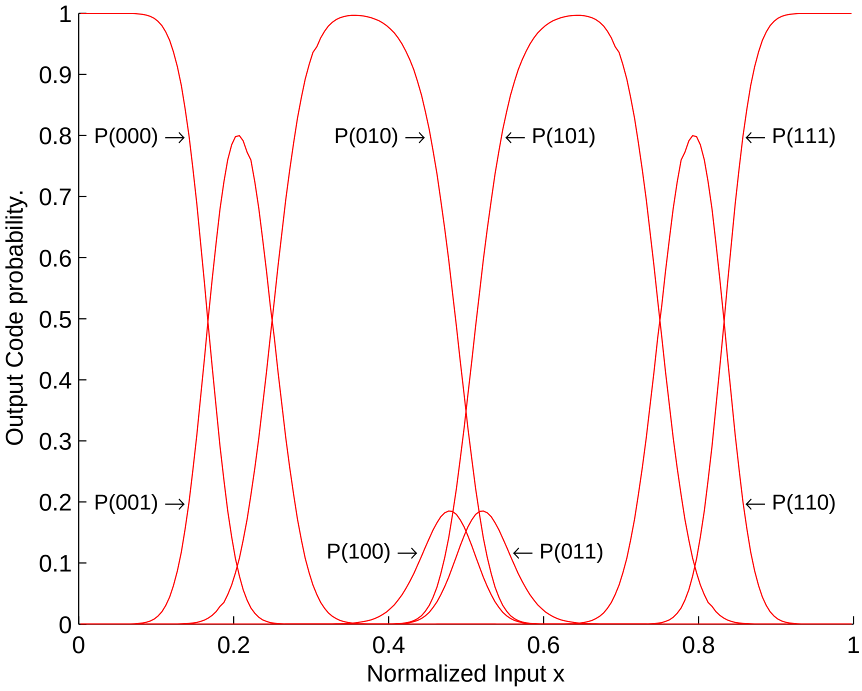

Figure 3.20: Probability functions for the 8 output codes of a three-bits output. The considered noise function has a Gaussian shape and a deviation $\sigma = 0.05$.

with noise equal to zero), it is possible to get the same results than with the simple optimal decoding (section 3.1).

The advantage of such an approach, with the noise power converging to 0, is the possibility to get an estimate for all output codes. If the distribution is Gaussian, the estimation of a noiseless signal x is given by (from (3.63)):

$$\hat{x}(\vec{b}) = \lim_{\sigma \to 0} \frac{\int_0^1 P(x, \vec{b}, \sigma) x \, dx}{\int_0^1 P(x, \vec{b}, \sigma) \, dx} \tag{3.64}$$

Note: In the computation of a noiseless input (3.64), it may be possible to obtain similar results whatever the dispersion function is, as long as this function is greater than zero in $\Re$, continuous and monotonic (i.e. the probability of a larger perturbation is smaller). Nevertheless, this assumption has not been proved mathematically. The substitution of the Gaussian distribution by a more exotic probability density may simplify the processing of the estimate $\hat{x}$. Indeed, the computation of $\hat{x}$ with a Gaussian has no analytical solution and requires a lot of numerical resources to compute all the successive integrations of (3.62).

The plots Fig. 3.21 visually show the convergence of $\hat{x}$ towards 0.5 for the codes $\vec{b} = 100$

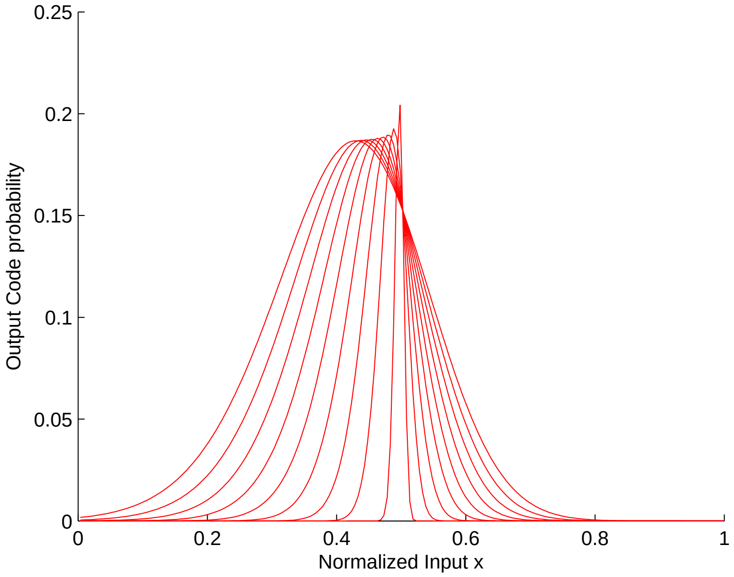

Figure 3.21: Probability functions of the code 100 for values of σ from 0.17 to 0.01. The left plot corresponds to $\sigma = 0.17$ while the narrow one, around 0.5 was generated with $\sigma = 0.01$.

and $\vec{b} = 011$ for a lower noise.

Table 3.1 summarizes the results of the three filters showing, as an example, the decoding of the three first bits of a flux $\vec{b}$.

Table 3.1: Estimates of the three-bits codes. 1st order modulator and constant input.

Code	Counter	Opt. Dec.	Gen. Opt. Dec.
111	1	11/12	11/12
110	2/3	19/24	19/24
101	2/3	5/8	5/8
011	2/3	?	1/2
100	1/3	?	1/2
010	1/3	3/8	3/8
001	1/3	5/24	5/24
000	0	1/12	1/12

3.5 Conclusion

It has been shown that the optimal decoding is able to reduce the quantization error of the sigma-delta converters, and this, for various architectures and various input signals.

The goal of the filters is not to compensate the internal imperfection of the modulators. Optimal filtering is thus sensitive to non-ideal effects. It has to be noticed that a similar limitation exists with the linear filters, when the transfer function of the filter is not anymore representative of the analog transfer function. For instance, MASH converters are especially sensitive to the DC gain of the amplifiers and the multi-bits converters suffer of the mismatch of the basic components of the DAC.

The selected approach of the optimal filers introduced in this chapter covered several architectures, and was also focused on non-constant inputs. The use of a filter for a sweep input delivers a better precision are moreover quantizes information on the variation of the input signal. The main drawback of the algorithms for constant and sweep inputs is their poor robustness. Indeed, these optimal filters fail to decode unexpected bit-streams, e.g. induced by non-ideal components in the modulator.

The analysis of input signals with Gaussian noise generalized the initial filter to have a full coverage of all output codes and thus provided a robust decoder. Nevertheless the increase of the mathematical complexity excludes its integration on chip.

To conclude, this chapter demonstrated the existence of a digital alternative to reduce the quantization error of ΣΔ modulators. Unlike classic methods to reduce the quantization noise (high-level, multi-bits, or hybrid modulators to quantize the residue after a conversion) no extra analog component is necessary. The benefits of optimal filters is however very limited if the thermal noise is much larger than the quantization noise.

4 ADC Input stage - A comparative study

In nowadays literature, emphasis is often put on the selection of the architecture and on the measured results. The implementation is sometimes mentioned, most of the time without any explanation about the prior reasoning or any justification of the choices made. While designing a converter, any designer has nevertheless to transform the high-level view of the modulator into an implementation with amplifiers, switches and capacitors.

The aim of this chapter is to provide a comparison of many implementations of a very simple topology of an integrator: one input for the signal and one input for the reference with an identical gain in a single-bit configuration. This simple integrator, whose high-level schematic is shown in Fig. 4.1, is typically used as the first integrator of ADCs.

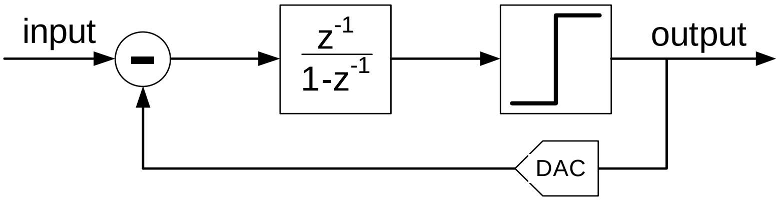

Figure 4.1: High-level representation of a first-order modulator.

A common basis has been defined to compare the architectures:

- After one cycle, the difference of charges on the integration capacitors should be equal to $C(Signal \pm Reference)$, depending on the value of the feedback bit.
- The delay between the integration of the reference and the quantization of the output should, whenever possible, be maximized. E.g., in the architecture #2, Fig. 4.4, if the comparator is active at the end of phase 1, the reference is integrated during phase 2.

- The load of the amplifier is distributed at best between the two phases.

The control signals enabling the switches are displayed in each schematic. A summary of all controls used in the architectures 1 to 9 is displayed in Fig. 4.2. The star in the signal name indicates that the signal depends on the feedback bit. I.e. in a set of 4 switched named 2*, during phase 2, either the direct connection, either the crossed connection is used. During phase 1, the four switches are left open.
The compact notation with chopper in the schematics is used only when the switches configuration is direct or crossed exclusively (i.e. the configuration with the 4 switches open is not used - apart from non-overlapping).

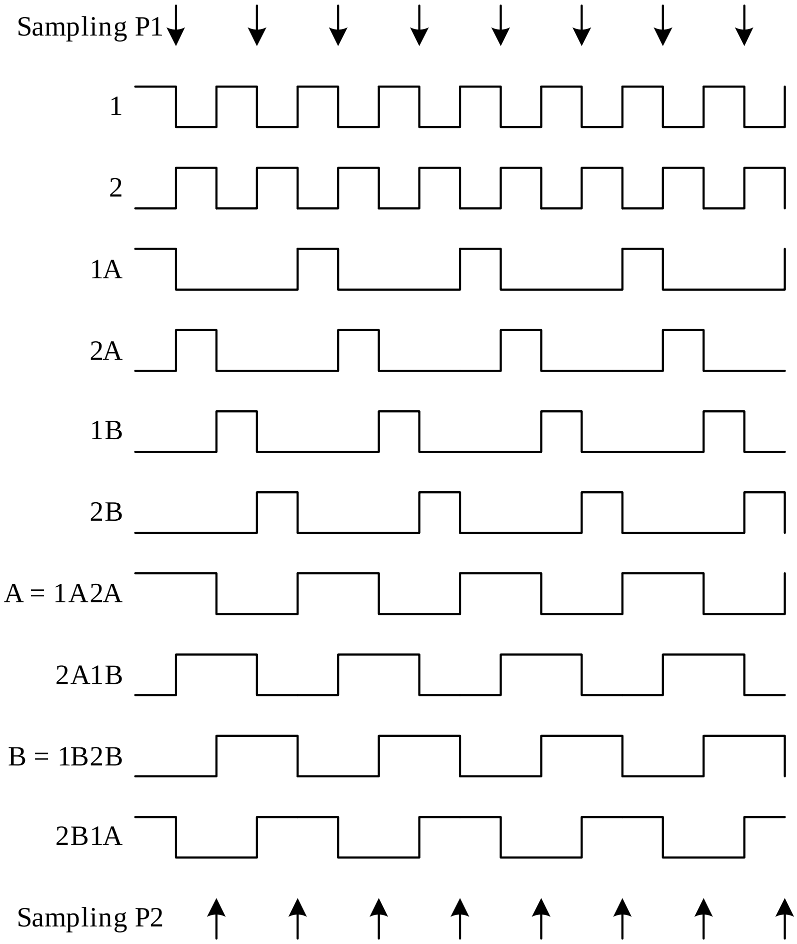

Figure 4.2: Control signals for the architectures #1 to #9.

In order not to overload the schematics, the delay between the signals has not been indicated. The controls of the switched on the high-impedance nodes (i.e. between the input capacitors and the amplifier) are always swapping firstly, and the source sided switches secondly. The goal of this shift, common to all architectures, is to reduce the injection of parasitic charges.

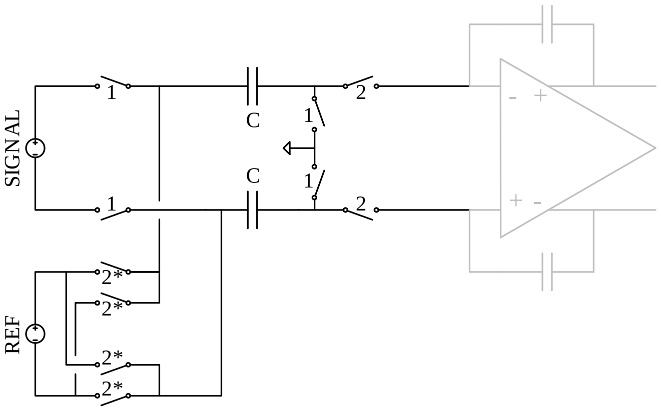

Figure 4.3: Architecture #1.

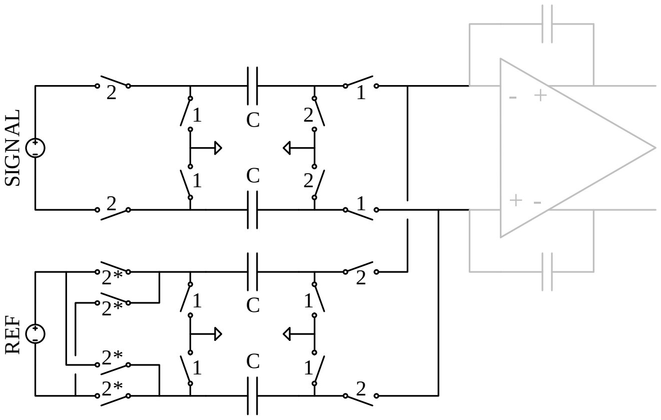

Figure 4.4: Architecture #2.

4.1 Two-phases architectures

The first architecture proposed, Fig. 4.3, is the most compact and probably the most used [29, 30, 31, 32, 33, 34] for high-resolution converters. The same input capacitors are used to transfer the signal and the reference. The precision of the gain is very good, but the gain is inevitably unitary. If the common-mode of the signal is different than the one of the reference, charges are transferred from the signal to the reference.

The second architecture, Fig. 4.4, has two distinct paths for the signal and the reference [35, 36, 37]. The gain can thus be adjusted, changing the value of the input capacitors. A variant of this architecture, Fig. 4.5, gets rid of the common-mode currents between the signal, reference and common-mode sources [38].

In the third topology, Fig. 4.6, the management of the feedback bit is done on the high-impedance nodes instead of being done on the reference itself.

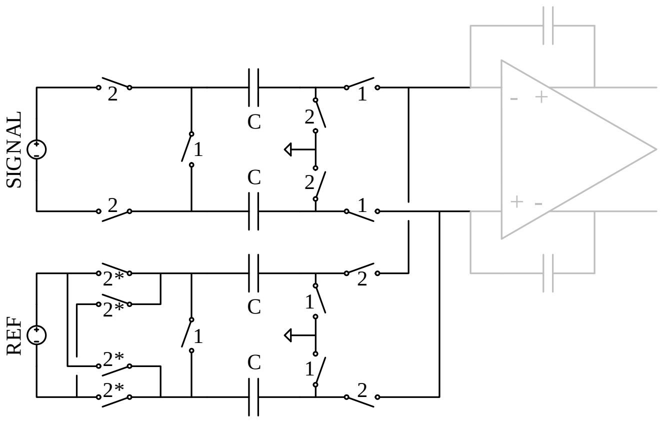

Figure 4.5: Architecture #2b.

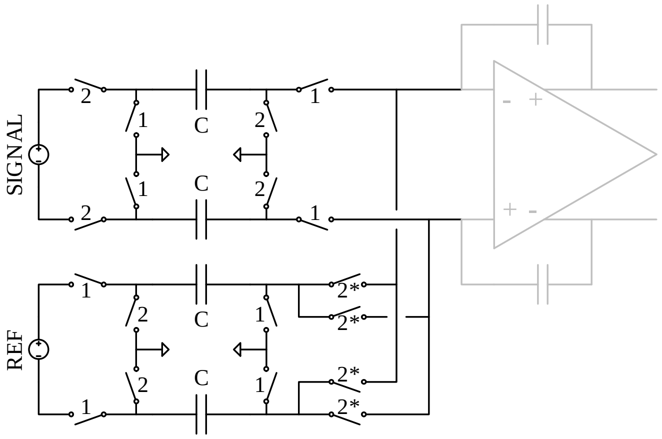

Figure 4.6: Architecture #3.

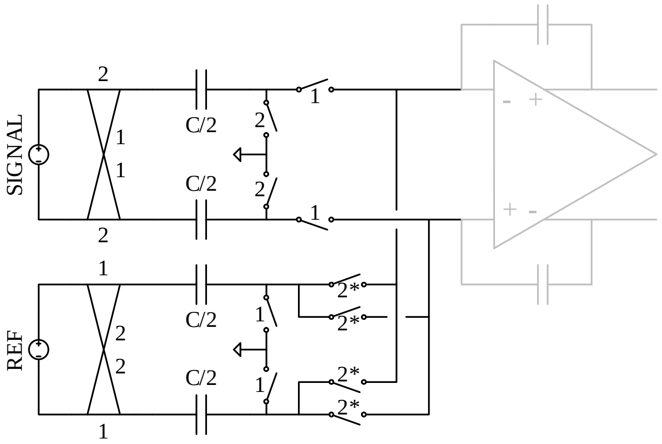

Figure 4.7: Architecture #4.

The input capacitors for the reference are connected alternatively to the reference source and to the amplifier of the integrator (i.e. the phases of pre-charge and of charge transfer are clearly separated). In this configuration, the settling errors of the reference and of the amplifier are not added. Practically, this topology is rarely used, but, in this document, it details the transition between the architecture 2 and 4.

In the architecture #4, Fig. 4.7, the signal and the reference are crossed to perform the charge transfer. The size of the input capacitors is thus divided by two to keep an identical transfer function, common to all architectures. The main enhancement compared to the architectures 2 and 3 is the diminution of the thermal noise on a complete cycle.

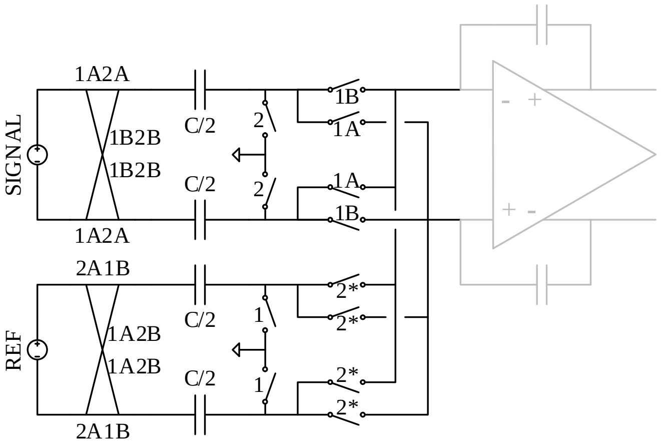

Figure 4.8: Architecture #5.

The fifth structure, Fig. 4.8, is a modified version of the topology #4 with lower dynamic power consumption. No charges are theoretically transferred between the end of a charge transfer phase and the beginning of a pre-charge phase. The switches connected to the common-mode are uniquely useful to control the common-mode of the virtual ground of the amplifier.

The architecture 4 and 5 allowed a diminution of the thermal noise in architectures with distinct path for the signal and for the reference, but with a sampling during both phases. In certain cases, the signal may not be available during both phases, e.g. if the integrator is preceded by a switched-capacitors system. The output of integrators or front-ends using a two-phase operation mode is usually correct in only one phase. The architecture 6, Fig. 4.9, stores the signal during one phase and performs the charge transfer during the next cycle. In order to guarantee a charge transfer during each cycle, two sets of capacitors are connected alternatively.

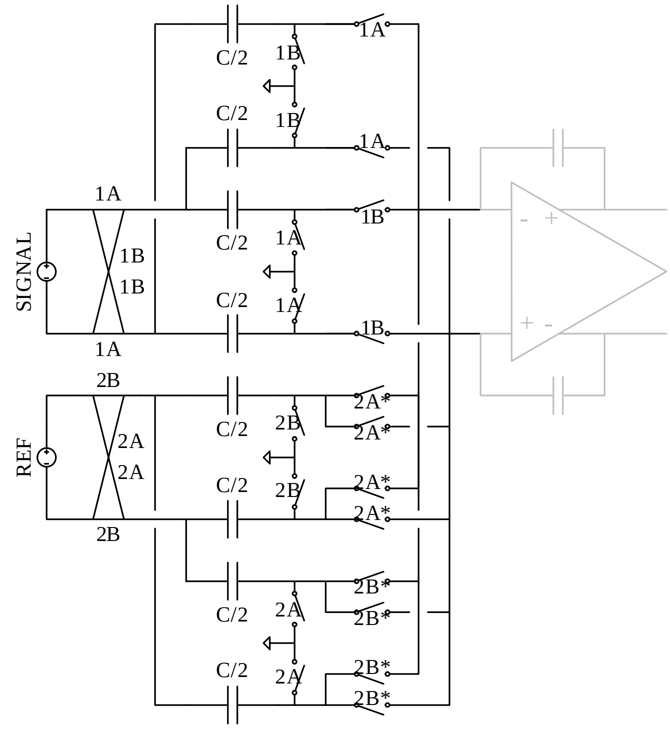

Figure 4.9: Architecture #6.

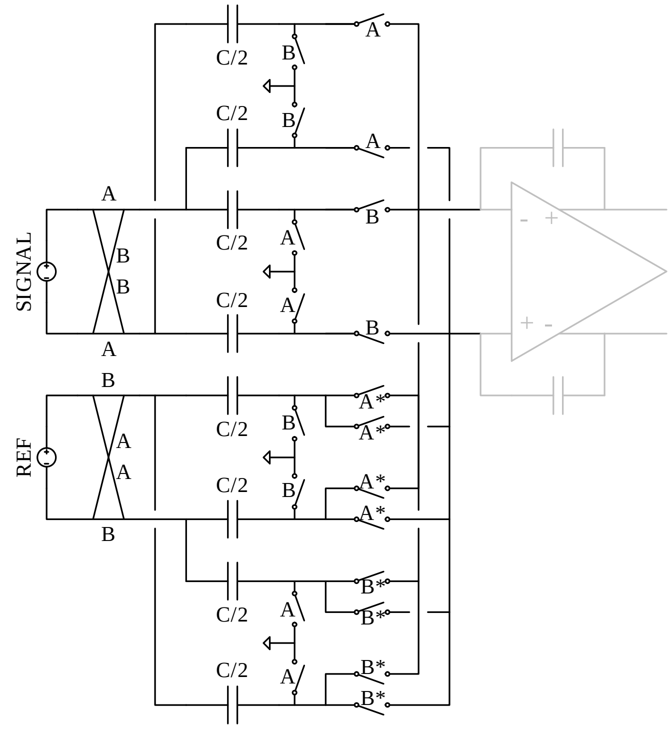

Figure 4.10: Architecture #7.

4.2 Symmetrical architectures

The power consumption of the architectures is determined with multiple factors. There is on one hand the dynamic power consumption, sum of all charges transferred during one cycle and, on the other hand, the static power consumption of the amplifiers and sources. If the current sources are class-A structures, the static power consumption of the blocks is determined with the maximal load they have to deliver during one phase.

In order to distribute the load on the blocks, especially the amplifiers, double-sampling structures may be used [39]. These topologies, Fig. 4.10 to 4.13 are not using two phases per cycle, but are transferring both signal and reference during a unique phase twice longer.

The major drawbacks of such structures are the short decision time left to the quantizer to evaluate the output, the increase of the area and a notable risk of introducing sensitive parasitic coupling degrading the resolution.

The architecture #7, implemented in [40], is the symmetrical version of the topology #4. The hardware is identical than the one of the structure #6, but with different controls and twice longer phases.

A first symmetrical version of the topology #5 is displayed in Fig. 4.11. This architecture #8 is extremely compact. The input capacitors are connected between two choppers to emulate resistors [40, 39]. For the reference, an almost identical structure is used. The only difference being the connection of the high-impedance chopper, determined according to the feedback bit.

Despite its apparent simplicity, this implementation requires a control of the virtual ground of the amplifier. This extra circuit is not shown in Fig. 4.11.

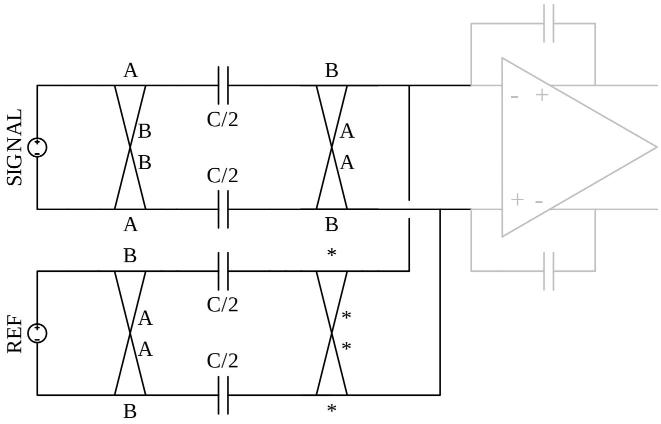

Figure 4.11: Architecture #8.

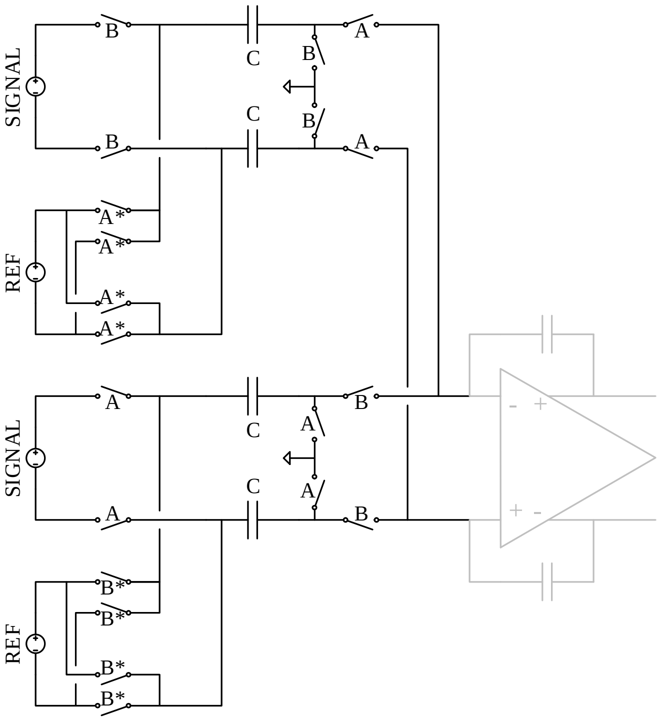

Figure 4.12: Architecture #9.

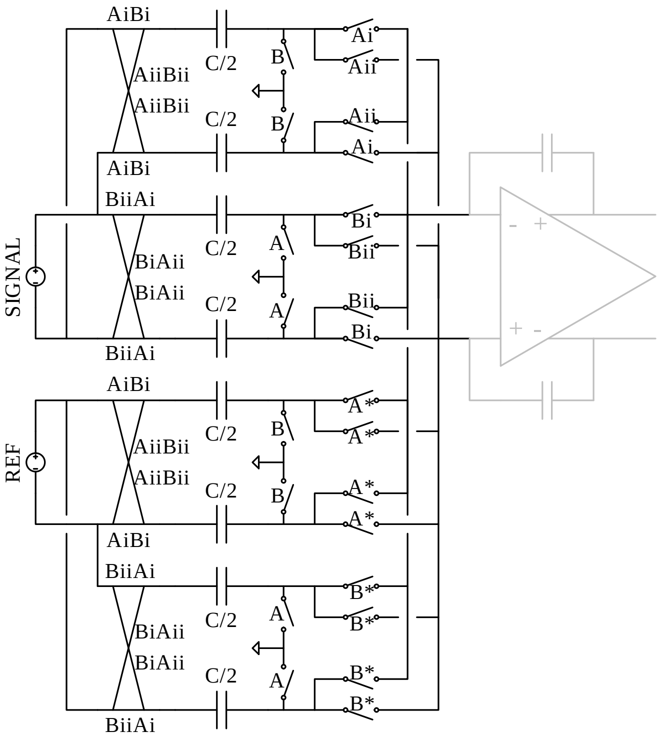

Figure 4.13: Architecture #10.

The architecture #9, Fig. 4.12, is the symmetrical version of the #1, with a better partitioning of the charges on the signal, the reference and the amplifier [41].

The tenth and last architecture in this comparison, Fig. 4.13, is a second symmetrical implementation of the topology #5. Compared to the architecture number 8, this structure implements a control of the common-mode of the virtual ground. The total capacitance is however larger. The specific controls of this architecture are shown in Fig. 4.14.

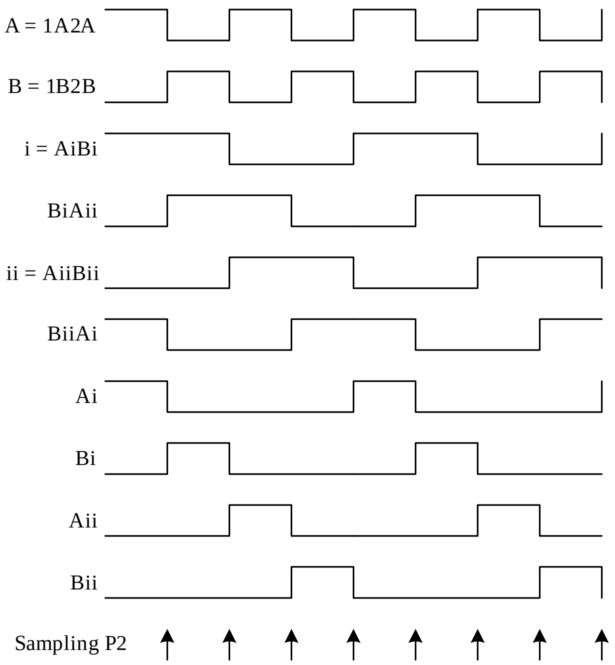

Figure 4.14: Specific control signals for the architecture #10.

4.3 Comparative study

The quality of the shown architectures, Fig. 4.3 to 4.13, is evaluated with the following criteria. The comparison, summarized in table 4.1, allows an efficient selection of the suited architecture, according to the specifications.

- Noise: The noise comparison of each topology is only focused on the thermal noise on the input capacitors. The noise of the amplifier (thermal and 1/f) strongly depends of the structure of the amplifier. The charge error on the input capacitors is mainly related to the number of sampling, to the size of the input capacitors and of the type of the input connection.
- Maximal load: The power consumption of the sources - input signal, reference, common-mode and amplifier - is evaluated supposing class-A sources with

constant power consumption. The comparison criterion is the mean current delivered by the source during any phase: $I = \Delta Q/T$. The current I is computed during the phase during which the charge to deliver ΔQ is the most important. Remark on the common-mode source: Theoretically, if the integrator is perfectly symmetrical around the common-mode source should not deliver any charge. During each phase, the charges pass directly from one capacitor to another, without loading the common-mode. The potential V_{cm} should thus remains constant during the transient. Practically, the amplifier is not symmetrical. While the value of the capacitors is almost constant, the on-resistance of the switches and the output resistance of the amplifier strongly depend on the applied voltage. During the transient, part of the charge is thus first absorbed by the common-mode source and then injected back.

- Dynamic power consumption: Sum of all the charges transferred in capacitors during one cycle of the integrator. While the previous item was dealing with the static power consumption of the sources, this one is only looking at the dynamic contribution.
- Common-mode current: Existence of a common-mode current between the sources if their respective common mode is not identical. E.g. in the architecture #1, between the input signal and the reference.
- Matching: Gain error related to the capacitive mismatch. The gain precision is mainly due to the size and to the connection mode of the capacitors.
- Silicon area: Determined with the overall size of the input capacitors.
- Programmable gain: Indicates if the gain is easily selectable with programmable capacitors.
- Sampling during one of the two phases: Some topologies are sampling the signal and the reference during each phase, while other ones are only sampling during one phase. Sampling during one phase may be required if the source is not stable during both phases. Especially true if the source - front end or integrator - is implemented with switched capacitors.
- Delay for the comparator / for the next stage: Indicates if a delay in the following integrator or in the comparator is compatible with the topology. If not, the quantization is done at the end of one phase and the reference is directly updated in the next one.
- Sensitivity to parasitic capacitors: Risk of parasitic coupling degrading strongly the linearity of the converter. The sensitivity was determined using a dedicated simulator. Refer to chapters 6 and 7.
- Sensitivity to the charge injection: The asymmetric clock feedthrough is detailed in chapter 5. The most sensitive architecture are the ones in which the switches on the high-impedance side are opened while the signal is connected to the

input capacitors.

- Overvoltage risk: Indicates if the high-impedance electrodes of the input capacitors may reach voltages beyond the power supplies during transients. The risk is evaluated for amplitudes (signal and/or reference) over half of the power supply voltage. If the risk exists, workaround have to be implemented in order to ensure a correct charge transfer, see chapter 7.

The comparison is summarized in table 4.1. The criteria are divided in four groups. The first one includes all the parameters related to the power consumption and to the thermal noise of the integrator. The second group contains the size criteria, matching and capacitive area. The third set regroups the features related to external constraints while the last one lists the risky points of the topologies.

Architecture	I	II	III	IV	V	VI	VII	VIII	IX	X
Noise	☺	☹	☹	☺	☺	☺	☺	☺	☺	☺
Max. load on amplifier	☹	☺	☺	☺	☺	☺	☺	☺	☺	☺
Max. load on signal	☹	☺	☺	☺	☺	☹	☺	☺	☺	☺
Max. load on reference	☹	☺	☺	☺	☺	☹	☺	☺	☺	☺
Max. load on Vcm	☹	☹	☹	😐	☺	😐	😐	☺	😐	☺
Dynamic power cons.	☺	☹	☹	☹	☺	☹	☹	☺	☺	☺
Common-mode current	☹	😐	😐	☺	☺	☺	☺	☺	☹	☺
Matching	☺	😐	😐	☹	☹	☹	☹	☹	☺	☹
Cap. area	☺	☹	☹	☺	☺	☹	☹	☺	☹	☹
Programmable gain	☹	☺	☺	☺	☺	☺	☺	☺	☹	☺
1 or 2 phases sampling	☺	☺	☺	☹	☹	☺	😐	😐	😐	😐
Delay for next stages	☺	☺	☺	☺	☺	☺	☹	☹	☹	☹
Parasitic coupling	☺	☺	😐	😐	☹	☹	☹	?	?	☹
Clock feedthrough	☺	☺	☺	☹	☹	☹	☹	☹	☺	☹
Overvoltage risk	☹	☺	☺	☹	☹	☹	☹	☹	☹	☹

Table 4.1: Advantages and drawbacks of the selected implementations

4.4 Conclusion

This succinct comparison showed the existence of a large diversity for implementing an integrator with switched-capacitors. The results synthesized in table 4.1 should help the designer selecting an implementation according to the constraints imposed on the modulator. Nevertheless, it is very important to validate the use of the selected architecture in the full modulator with simulations.

5 Models

Modeling of a physical phenomenon is done, through an abstractive procedure, to represent it mathematically. The main goal of a model is to predict a phenomenon, whether it is an apple falling, tomorrow's weather or, in the present case, a switched-capacitors integrated circuit.

An important choice in the model is the abstraction level. Is it better to have a complex model describing at best the reality or on the contrary an intuitive and easily understandable simple model?

A key parameter to select the abstraction level is, in addition to the required precision, the effort to use the model. The most abstract models are often used to predict a phenomenon with hand calculations. With the advent of modern computing, the most complex calculations are executed on computers. Therefore, the quantization of the effort to provide changed to become a required number of operations and finally a computation or simulating time.

The necessary time to predict the behavior of a circuit is related to two parameters:

- The abstraction level of the model. A complex model requires more effort to be used than a simple one.
- The size of the circuit: A small circuit, with only a few nodes, is very easy to simulate. Furthermore, the fragmentation of a large circuit in many smaller entities allows a significant gain of time, as the required number of operations to perform is rarely linear with the number of components.

This chapter deals with different approaches, modeling first a small circuit (an integrator) with a lot of details. Secondly, a simpler model, but containing the whole modulator is presented. Both approaches are complementary and allow to apprehend distinct effects in a reasonable time.

This chapter is dedicated to the models of the switched-capacitors ΣΔ modulator. Special care is taken on each model to reduce the number of operations required on their application.

The models are implemented in a dedicated simulator, in the next chapter, with the design of Computer-Aided Design (CAD) tools and their use in the conception of an ADC.

5.1 Integrator

In an abstract view of a ΣΔ modulator, Fig. 5.1, a few main components are discernible. There is a subtractor, an integrator, a quantizer and a DAC.

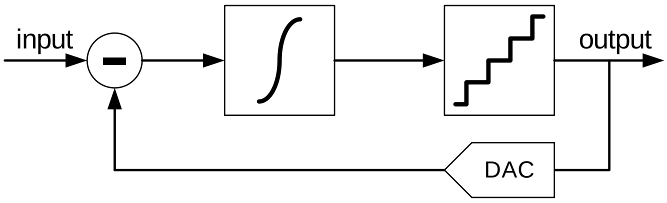

Figure 5.1: High-level view of a first-order ΣΔ modulator.

This block representation is used in many high-level simulators, the best known are the ’Delta Sigma Toolbox’ [42], ’SIMSIDES’, [43], ’Daisy’ [44] and the ’SD Toolbox’ [45]. This high-level representation is easily understandable by the human, as each block function is visually displayed.

Looking at a lower level, a switched-capacitors ΣΔ modulator is made of capacitors, switches, amplifiers, voltage references and quantizers, Fig. 5.2. The disposition of the components is different: the integrator has several phases, has a dual input, and integrate the signal and the reference (if the DAC is made of switched capacitors, which is the most common case).

In a switched-capacitors model, the behavior of the modulator is no more defined using high-level function as a subtraction or an integration. Equations such as charge conservation are used, closer to the electrical nature of the circuit.

In this model, a complete integrator has several inputs, at least one for the signal, and usually one for the reference. The main idea of the proposed modeling is to consider a single model whatever the current state of the modulator is. Solving the current state of an integrator is nothing else than evaluating the charge transfers between input capacitors and integration capacitors, knowing the initial state of the integrator (i.e. the charge in each capacitor). This decomposition is detailed in Fig. 5.3 for an

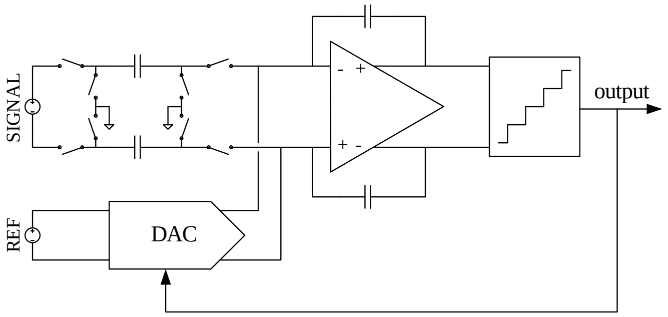

Figure 5.2: Implementation of the integrator with switched capacitors in a fully-differential topology.

integrator with two inputs sequentially connected.

This section is dedicated to the modeling of the integrator, Fig. 5.3b. Imperfections of the amplifier, of the switches and of the capacitors are taken into account. To cover all operation modes of the integrator[1], individual (Fig. 5.3b) and multiple inputs (Fig. 5.3c) configurations are analyzed.

The addressed problem, in a discrete-time integrator, is to find the voltages at the end of a conversion knowing the initial charges in the capacitors (at the beginning of a conversion). In most cases, for a single-input integrator, four potential have to be found, the virtual ground nodes W_{irtn} and W_{irtp} and the output voltages, V_{outp} and V_{outn}. The basic equations of the charge transfer are the charge conservation equations on the inputs of the amplifier:

$$\begin{aligned}
&C_{inup}(V_a - W_{irtn}) + C_{up}(V_{outp} - W_{irtn}) - C_{aup}(W_{irtn}) - C_{cw}(W_{irtn} - W_{irtp}) \\
&= Q_{inup(t=0)} + Q_{up(t=0)} - Q_{aup(t=0)} - Q_{cw(t=0)} \\
&C_{indo}(V_b - W_{irtp}) + C_{do}(V_{outn} - W_{irtp}) - C_{ado}(W_{irtp}) + C_{cw}(W_{irtn} - W_{irtp}) \\
&= Q_{indo(t=0)} + Q_{do(t=0)} - Q_{ado(t=0)} + Q_{cw(t=0)}
\end{aligned} \tag{5.1}$$

The extra capacitors C_{aup}, C_{ado} and C_{cw} are the parasitic capacitors connected on the virtual ground nodes, as shown in Fig. 5.5.

As the integrator is fully-differential, the output common-mode has to be set to a

---

[1] The pre-charge of the input capacitors, the reset of the integrator, and the integrator without input capacitors are usually not detailed, as their modeling is easier than the charge transfer case.

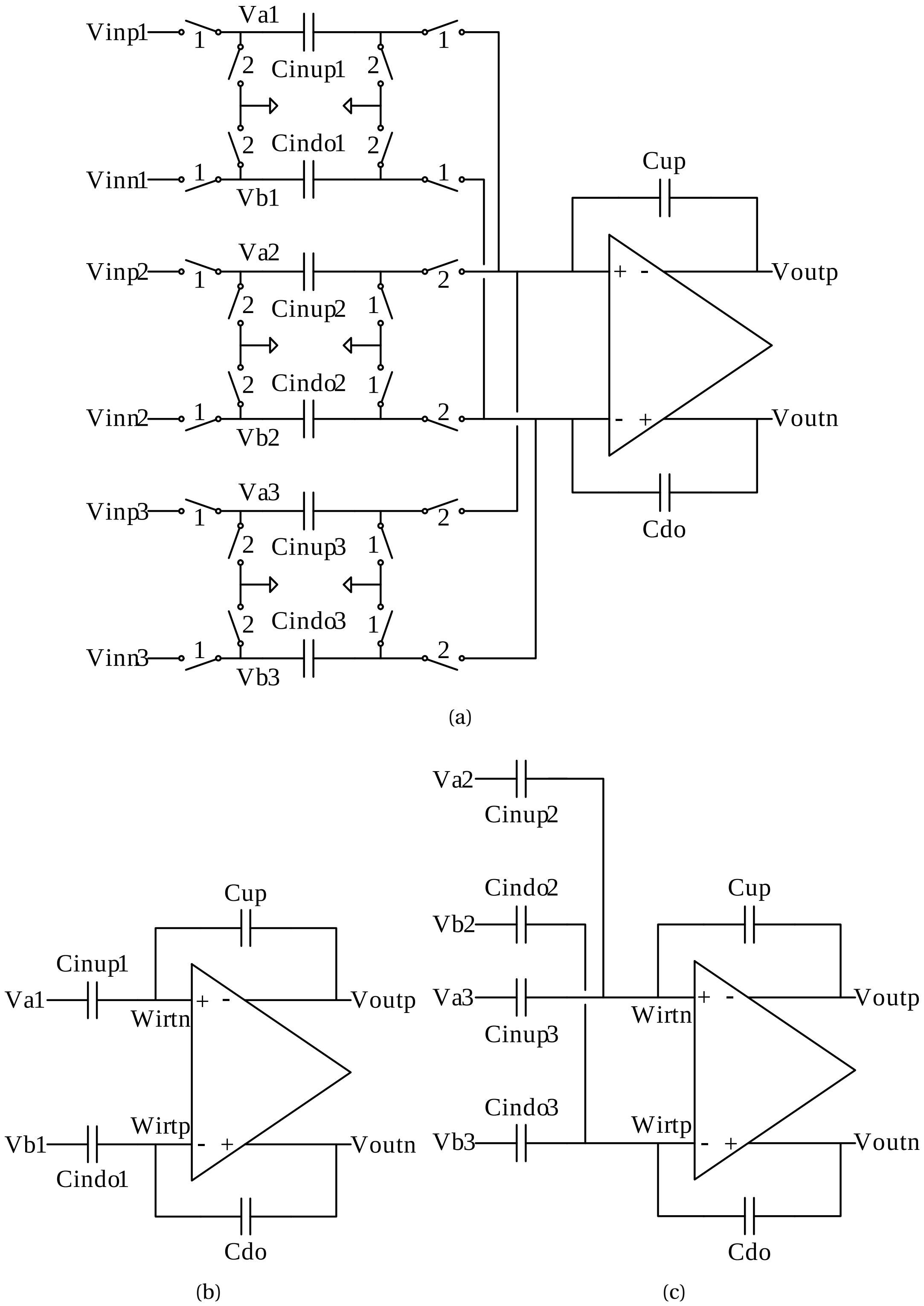

Figure 5.3: Representation of a two-phases integrator (a) into two similar subcircuit for phase 1 (b) and phase 2 (c).

common-mode potential:

$$V_{outp} + V_{outn} = 2V_{cm} \tag{5.2}$$

The last equation is for the amplifier in a negative feedback configuration. If the amplifier is ideal, W_{irtp} is equal to W_{irtn}.

5.1.1 Ideal Integrator

The ideal integrator (i.e. with ideal components) is linear. It is thus possible to compute the voltages at equilibrium using a matrix description: $A\vec{v} = \vec{b}$:

$$\begin{pmatrix} 1 & 1 & 0 & 0 \\ 0 & C_{up} & C_{cw} & -C_{inup} - C_{up} - C_{aup} - C_{cw} \\ C_{do} & 0 & -C_{indo} - C_{do} - C_{ado} - C_{cw} & C_{cw} \\ 0 & 0 & 1 & -1 \end{pmatrix} \begin{pmatrix} V_{outn} \\ V_{outp} \\ W_{irtp} \\ W_{irtn} \end{pmatrix} = \begin{pmatrix} 2V_{cm} \\ \Sigma Q_{up} - V_a C_{inup} \\ \Sigma Q_{do} - V_b C_{indo} \\ 0 \end{pmatrix} \tag{5.3}$$

The first equation of (5.3) is the control of the output common-mode, the 2nd and 3rd equations are the charge conservation equations for the two nodes of the virtual ground. As the amplifier in the integrator is ideal, both inputs are equal (last equation).

The four unknown nodes V_{outn}, V_{outp}, W_{irtp} and W_{irtn} are computed according to the capacitors in the circuit, the input voltages $V_{a/b}$ and the initial charges of the integrator. The two terms ΣQ_{up} and ΣQ_{do} contains the initial charges:

$$\begin{aligned} \Sigma Q_{up} &= Q_{Cup} + Q_{Cinup} - Q_{Caup} - Q_{Ccw} \\ \Sigma Q_{do} &= Q_{Cdo} + Q_{Cindo} - Q_{Cado} + Q_{Ccw} \end{aligned} \tag{5.4}$$

5.1.2 Amplifier - DC Gain and Offset

The considered non-ideal integrator is shown in Fig. 5.3b. The capacitors are linear and the switches are substituted with wires. It is possible to include the parasitic capacitor of the input gates of the amplifier into the capacitors C_a. The amplifier has a finite gain A_0 and an input offset V_{off}. Its transfer function is:

$$\begin{aligned} V_{outp} &= V_{cm} + \frac{A(V_{in})}{2} = V_{cm} + \frac{A_0(W_{irtp} - W_{irtn} - V_{off})}{2} \\ V_{outn} &= V_{cm} - \frac{A(V_{in})}{2} = V_{cm} - \frac{A_0(W_{irtp} - W_{irtn} - V_{off})}{2} \end{aligned} \tag{5.5}$$

The other equations for this circuit (charge conservation) are identical than the ones for the ideal integrator (5.1).

If the integrator has many inputs connected at the same time, the equations (5.1) are generalized:

$$\begin{aligned} &\sum_i C_{inup,i}(V_{a,i} - W_{irtn}) + C_{up}(V_{outp} - W_{irtn}) - C_{aup,i}(W_{irtn}) - C_{cw}(W_{irtn} - W_{irtp}) \\ &= \Sigma Q_{up(t=0)} \\ &\sum_i C_{indo,i}(V_{b,i} - W_{irtp}) + C_{do}(V_{outn} - W_{irtp}) - C_{ado,i}(W_{irtp}) + C_{cw}(W_{irtn} - W_{irtp}) \\ &= \Sigma Q_{do(t=0)} \end{aligned} \tag{5.6}$$

The equations (5.5) and (5.1) form a linear system with four equations. It is thus possible to write this system using a matrix form: $A\vec{v} = \vec{b}$

$$\begin{aligned} &\begin{pmatrix} 1 & 1 & 0 & 0 \\ 0 & C_{up} & C_{cw} & -C_{inup} - C_{up} - C_{aup} - C_{cw} \\ C_{do} & 0 & -C_{indo} - C_{do} - C_{ado} - C_{cw} & C_{cw} \\ 1/A_0 & -1/A_0 & 1 & -1 \end{pmatrix} \begin{pmatrix} V_{outn} \\ V_{outp} \\ W_{irtp} \\ W_{irtn} \end{pmatrix} \\ &= \begin{pmatrix} 2V_{cm} \\ Q_{inup} + Q_{up} + Q_{aup} - Q_{C_{cw}} - V_a C_{inup} \\ Q_{indo} + Q_{do} + Q_{ado} + Q_{C_{cw}} - V_b C_{indo} \\ V_{off} \end{pmatrix} \end{aligned} \tag{5.7}$$

As the matrix A only contains constant inputs (i.e. not related to the state, voltages and charges of the modulator), it is possible to pre-compute the invert matrix A^{-1}. Thereby, during each modulator cycle, only the matrix product $\vec{v} = A^{-1}\vec{b}$ is computed.

Once the voltages V_{out} and W_{irt} are computed, it is easy to get the charges Q of the capacitors.

5.1.3 Amplifier - Voltage Saturation

Every amplifier has a maximal output range V_{sat-}, V_{sat+}, voltages beyond which it cannot operate.

The considered model is simple: either the amplifier is not saturated, with output voltages between V_{sat-} and V_{sat+}, or it is saturated and its outputs are equal to V_{sat-} and V_{sat+}.

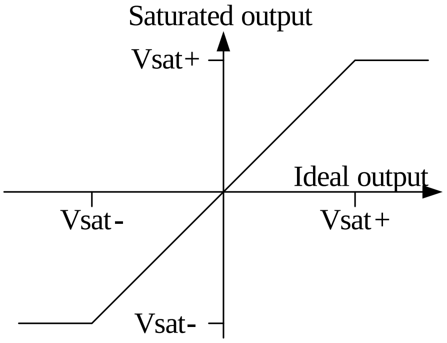

Figure 5.4: Saturation transfer function. The saturated output cannot reach voltages beyond the V_{sat-} and V_{sat+} limits.

The implementation of the saturation is straightforward. The voltages that would be obtained without saturation (section 5.1.1) are compared to the saturation thresholds. If the amplifier is saturated, the output voltages are set to V_{sat-} and V_{sat+} and the voltage of the nodes of the virtual ground are recomputed with the charge conservation equations (5.1).

For continuity reasons in the model, it is preferable that the potentials V_{sat-} and V_{sat+} are centered around the output common-mode of the amplifier: $V_{sat-} + V_{sat+} = 2V_{cm}$. If this is not the case in the desired implementation, a pessimistic case should be considered for modeling.

5.1.4 Amplifier - Non-linear Gain

The integrator including an amplifier with a non-linear gain is described in Fig. 5.5. To lighten the writing of the equations in this section, all voltages are referenced to the output common-mode of the amplifier V_{cm} (i.e. $V_{outpl} = V_{outp} - V_{cm}$). The suffix 'l' was added in order to avoid any confusion with the ground referenced voltages. The expression of the output common-mode (5.2) is simplified:

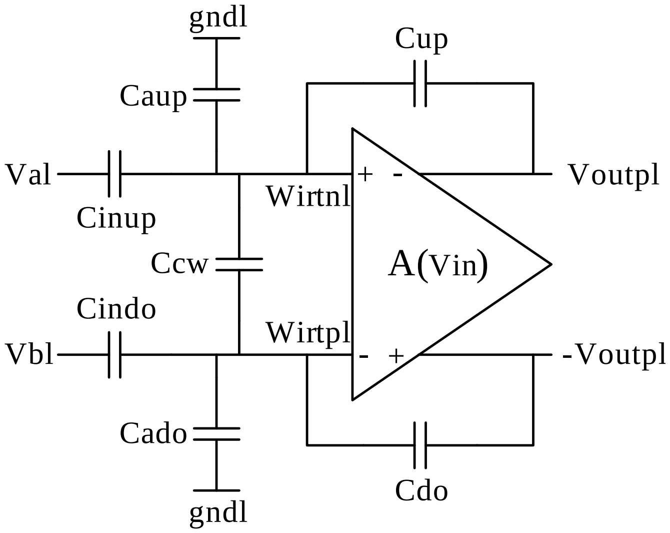

Figure 5.5: Representation of the model of the integrator considering an amplifier with a non-linear gain. The parasitic capacitors taken into account in the model are displayed.

$$V_{outpl} + V_{outnl} = 0 \tag{5.8}$$

To reduce the number of variables, V_{outnl} is systematically replaced by $-V_{outpl}$ in the subsequent equations.

The output voltage of an integrator with an amplifier with a generic gain function $V_{out} = A(V_{in})$ is:

$$V_{outpl} = A(W_{irtpl} - W_{irtnl} - V_{off})/2 \tag{5.9}$$

The virtual ground potential voltages are obtained inverting the charge conservation equations (5.1) on the nodes W_{irtp} and W_{irtn}:

$$\begin{aligned} W_{irtnl} &= \frac{V_{outpl}C_{up} + V_{al}C_{inup} + gndlC_{aup} - \Sigma Q_{up}}{C_{up} + C_{inup} + C_{aup}} \\ W_{irtpl} &= \frac{-V_{outpl}C_{up} + V_{bl}C_{indo} + gndlC_{ado} - \Sigma Q_{do}}{C_{do} + C_{indo} + C_{ado}} \end{aligned} \tag{5.10}$$

Particularly, the input of the amplifier $W_{irtpl} - W_{irtnl}$ is obtained subtracting the equations (5.10):

$$W_{irtp} - W_{irtn} = aV_{outpl} + b \tag{5.11}$$

with

$$\begin{aligned} a &= -\frac{C_{up}}{C_{up} + C_{inup} + C_{aup}} - \frac{C_{do}}{C_{do} + C_{indo} + C_{ado}} \\ b &= -\frac{V_{al}C_{inup} + gndlC_{aup} - \Sigma Q_{up}}{C_{up} + C_{inup} + C_{aup}} - \frac{V_{bl}C_{indo} + gndlC_{ado} - \Sigma Q_{do}}{C_{do} + C_{indo} + C_{ado}} \end{aligned} \tag{5.12}$$

The combination of the equations (5.9) and (5.11) provides a single equation for V_{outpl}:

$$V_{outpl} = A(aV_{outpl} + b - V_{off}) \tag{5.13}$$

As the equation (5.13) has generally no analytical solution, a digital iterative method, such as the Newton's method (5.14), has to be implemented:

$$x_{n+1} = -\frac{f(x_n)}{f'(x_n)} + x_n \tag{5.14}$$

with $f(x_n) = f(V_{outpl}) = -x_n + A(ax_n + b)$ and $f'(x_n) = -1 + aA'(ax_n + b)$

Implemented functions

The choice of the function modeling the gain A is not trivial. On one side, the function A should be close to the transfer function of the amplifier implemented with transistors, and on the other side it has to respect a few mathematical properties to be compatible with Newton's method. As it is necessary for the function A and A' to be continuous, it is preferable to select a function A which is not piecewise defined.

The implemented functions to model the amplifier are the following, with x being the normalized input, A the normalized output and α a coefficient set to 1.5:

$$\begin{aligned} A_{linear} &= x \\ A_{HypTan} &= \tanh(x) \\ A_{xGaussSech} &= \tanh\left(\frac{x}{2}\right) + x\left(e^{-x^2} - \frac{1}{\cosh(2x)}\right) \\ A_{sinGauss} &= \tanh(x) + \sin(\alpha x)\, e^{-\alpha^2 x^2} - \sin(x) e^{-x^2} \\ A_{sinSech} &= \tanh(x) + \frac{\sin(\alpha x)}{\cosh(\alpha x)} - \frac{\sin(x)}{\cosh(x)} \end{aligned} \tag{5.15}$$

All these classic and exotic functions, shown on Fig. 5.6, have been selected for:

- Linear: To test the model and the numerical convergence method, comparing the results with section 5.1.2.
- Hyperbolic tangent: A simple continuous function with output voltages saturated by the power supplies.
- The three other functions: $xGaussSech$, $sinGauss$ and $sinSech$ are trying to reproduce the alteration of the gain in the middle of the range of real amplifiers.

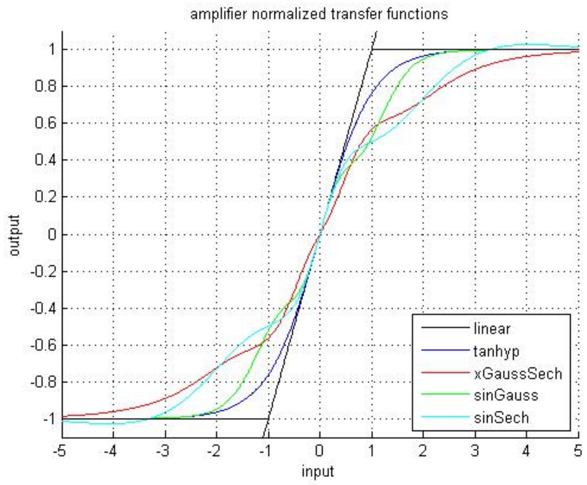

Figure 5.6: Graphical representation of the normalized gain functions (5.15) of the amplifier.

Multiple Inputs

The modeling of an integrator with multiple inputs is easily generalizable: it is sufficient to substitute, as in (5.6), the terms V_aC_{inup}, V_bC_{indo}, ΣQ_{up} and ΣQ_{do}.

5.1.5 Amplifier - Transient - SR

The transient phenomena of the integrators in switched-capacitors circuits are mainly due to the Slew-Rate of the amplifier and to the on-resistance of the switches. It has been decided to analyze these two effects separately as they are independent. The SR of the amplifier is limited by its biasing and by the load on the output nodes, while the speed limitation of the charge transfer on the resistive lines is limited by the serial resistances (on-resistance of the switches) and by the load on the lines.

The distinct analysis of both phenomena provides a decent tool of analysis and comprehension, even if, for the design, extra margin have to be taken into account. This, especially if the design margins are small, for the slew-rate AND for the RC.

This section and the following one are modeling the limitation of the amplifier while a specific section ,5.1.7, is dedicated to the switches.

The integrator view for the models of the SR of the amplifier is shown in Fig. 5.7. The capacitors C_{cw}, C_{co}, C_{Lup} and C_{Ldo} are parasitic and output capacitors on the input and output nodes of the amplifier. In this section, ideal switches are considered, with a zero on-resistance. The switches have been substitutes by wires in Fig. 5.7.

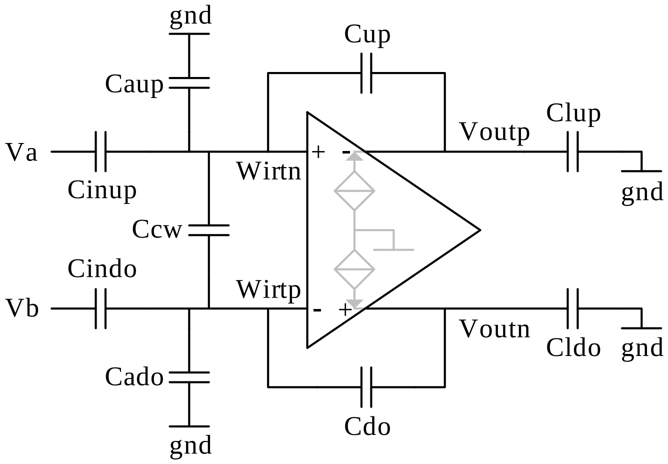

Figure 5.7: Model of the amplifier with a limited output current.

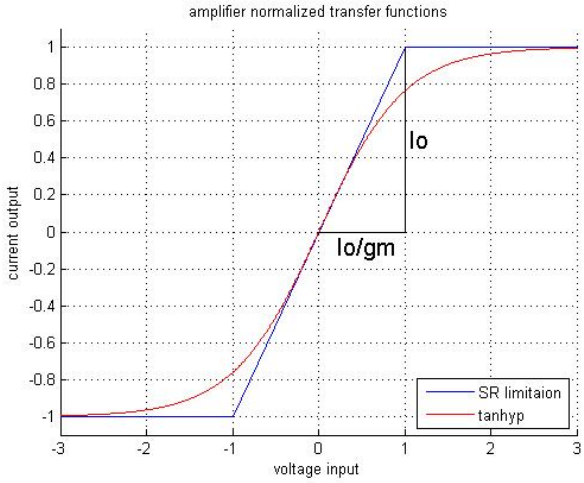

Figure 5.8: Normalized amplifier transconductances with a maximal slew rate. The model of the amplifier with a constant transconductance g_m and a current limit I_0 is described with the dark blue piecewise linear function. The red curve is the continuous hyperbolic tangent representation, section 5.1.6.

A first way to model the current limitation of the amplifier is to force a maximal output current I_0 in the model. The transfer function between the input differential pair and the output current is displayed in Fig. 5.8.

During a conversion, if an important charge has to be transferred in the integrator, a first slewing phase is visible, limited by the current I_0. It is followed by a settling phase, where the amplifier is acting in its linear regime. The schematic Fig. 5.9 shows the evolution over time of the output voltage and introduces the parameters used.

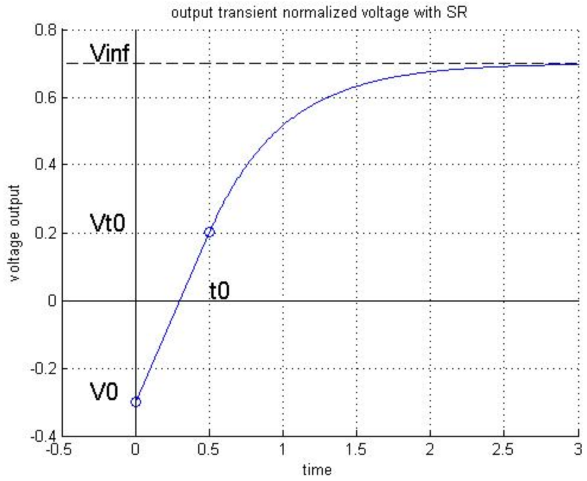

Figure 5.9: Ouput transient voltage with a slewing phase between $t = 0$ and $t = T_0$ followed by a settling phase for $t > T_0$.

One can notice in Fig. 5.9 that, to compute the output of the modulator, it is not necessary to process the whole transient. The computation of the voltages on three specific moments is sufficient:

- t = 0. Initialization
- t = T_0. End of the SR and beginning of the linear mode
- t = T_{conv}. End of the conversion

Specific models are used in each step to determine the voltages of the modulator.

Initialization

In the initialization phase, at time $t = 0$, the amplifier has not yet delivered any charges. The equivalent circuit to analyze, Fig. 5.10. is completely passive. To ensure continuity with the model during the slewing phase, the output common-mode is assumed to be constant and equal to V_{cm}. As a consequence, an extra common-mode charge Q_{cm} was added on the output nodes of the integrator.

The circuit shown in Fig. 5.10 is linear. A matrix representation is thus suitable, $A\vec{v} = \vec{b}$.

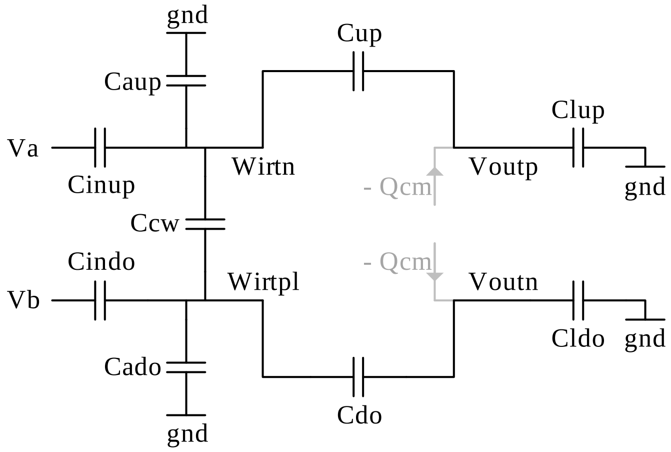

Figure 5.10: Equivalent linear circuit during initialization.

$$\begin{pmatrix} 1 & 0 & C_{lup}+C_{up} & 0 & -C_{up} \\ 1 & C_{ldo}+C_{do} & 0 & -C_{do} & 0 \\ 0 & 1 & 1 & 0 & 0 \\ 0 & C_{do} & 0 & \text{-}C_{do}\text{-}C_{indo}\text{-}C_{ado} & 0 \\ 0 & 0 & C_{up} & 0 & \text{-}C_{up}\text{-}C_{inup}\text{-}C_{aup} \end{pmatrix} \begin{pmatrix} Q_{cm} \\ V_{outn} \\ V_{outp} \\ W_{irtp} \\ W_{irtn} \end{pmatrix}$$
$$= \begin{pmatrix} Q_{lup}+Q_{up}+V_{cm}C_{lup} \\ Q_{ldo}+Q_{do}+V_{cm}C_{ldo} \\ 2V_{cm} \\ \Sigma Q_{do} \\ \Sigma Q_{up} \end{pmatrix} \tag{5.16}$$

As the matrix A only contains constant entries (in this model, the value of the capacitors is not varying), its inverse A^{-1} is pre-computable when the architecture is created. The only operation left for each cycle is the matrix product computation $\vec{v} = A^{-1}\vec{b}$.

Slewing

In the slewing phase, the amplifier is saturated and delivers a maximal current I_0. An equivalent representation of the circuit is shown in Fig. 5.11. In addition to the capacitors of the integrator, two current sources are added to model the amplifier. The output common-mode regulation is supposed to be ideal, and is modeled adding an identical charge Q_{cm} on both output nodes. The system of equation corresponding to the schematic Fig. 5.11 contains six equations. Four equations establish the charge conservation of the input and output nodes of the amplifier. One equation (5.2) set the output common-mode. The last equation $(W_{irtp} - W_{irtn})g_m = I_0$ defines the transition point between the slewing and the settling phases. The system of equations is written hereafter:

$$\begin{pmatrix} 1 & \mp I_0 & 0 & C_{lup}{+}C_{up} & 0 & -C_{up} \\ 1 & \pm I_0 & C_{ldo}{+}C_{do} & 0 & -C_{do} & 0 \\ 0 & 0 & 1 & 1 & 0 & 0 \\ 0 & 0 & 0 & C_{up} & C_{cw} & \text{-}C_{inup}\text{-}C_{up}\text{-}C_{aup}\text{-}C_{cw} \\ 0 & 0 & C_{do} & 0 & \text{-}C_{indo}\text{-}C_{do}\text{-}C_{ado}\text{-}C_{cw} & C_{cw} \\ 0 & 0 & 0 & 0 & 1 & -1 \end{pmatrix}
\begin{pmatrix} Q_{cm} \\ t_0 \\ V_{outn} \\ V_{outp} \\ W_{irtp} \\ W_{irtn} \end{pmatrix} = \begin{pmatrix} Q_{lup} + Q_{up} \\ Q_{ldo} + Q_{do} \\ 2V_{cm} \\ \Sigma Q_{up} - V_a C_{inup} - gndC_{aup} \\ \Sigma Q_{do} - V_b C_{indo} - gndC_{ado} \\ \pm I_0/g_m \end{pmatrix} \tag{5.17}$$

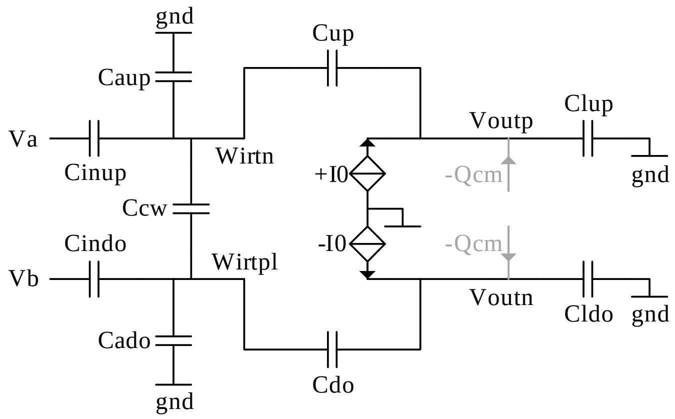

Figure 5.11: Equivalent linear circuit during the slewing phase.

As the four last equations are independent of the two first ones, it is possible to solve first the following subsystem:

$$\begin{pmatrix} 1 & 1 & 0 & 0 \\ 0 & C_{up} & C_{cw} & \text{-}C_{inup}\text{-}C_{up}\text{-}C_{aup}\text{-}C_{cw} \\ C_{do} & 0 & \text{-}C_{indo}\text{-}C_{do}\text{-}C_{ado}\text{-}C_{cw} & C_{cw} \\ 0 & 0 & 1 & -1 \end{pmatrix} \begin{pmatrix} V_{outn} \\ V_{outp} \\ W_{irtp} \\ W_{irtn} \end{pmatrix} = \begin{pmatrix} 2V_{cm} \\ \Sigma Q_{up} - V_a C_{inup} - gndC_{aup} \\ \Sigma Q_{do} - V_b C_{indo} - gndC_{ado} \\ \pm I_0 / g_m \end{pmatrix} \tag{5.18}$$

It is, there as well, possible to compute the inverse of the capacitance matrix in order not to have to perform this operation for each cycle of the modulator. The two unknowns left, T_0 and Q_{cm}, are easily obtainable with the voltages $V_{outp/n}$ and $W_{irtp/n}$.

Settling

In the last step, the voltages of the nodes V_{outp} and V_{outn} have to be determined at the end of the settling phase, at time T_{conv}. As shown in Fig. 5.9, the settling phase is the convergence of an exponential toward a limit to infinity $V_{out(t=\infty)}$. The potential $V_{outp/n(t=\infty)}$ are the static solutions of the integrator, and are thus computed using the model with a full charge transfer (section 5.1.1).

The slope ∂V_{out} at time T_0 is given by the slew-rate of the amplifier:

$$\frac{\partial V_{outp(t=T_0)}}{\partial t} = SR = \frac{V_{outp(t=T_0)} - V_{outp(t=0)}}{T_0} \tag{5.19}$$

The time constant τ of the time response is easily computable:

$$\tau = \frac{V_{outp(t=\infty)} - V_{outp(t=T_0)}}{SR} \tag{5.20}$$

and the voltage at time t is defined by:

$$V_{outp(t)} = V_{outp(t=T_0)} + \left(V_{outp(t=\infty)} - V_{outp(t=T_0)}\right)\left(1 - e^{\frac{T_0 - t}{\tau}}\right) \tag{5.21}$$

Remark: Both slewing and settling phases do not necessarily appear. If the conversion time is too short, there is only the slewing phase, while the compensation of a weak perturbation is done in the settling mode only. Once the potential V_{outp} and V_{outn} are defined, the computation of the virtual ground voltages is straightforward.

Here as well, the model is compatible with multiple inputs. As previously, the only modification is the substitution of the factors $C_{in}V_a$ and $C_{in}V_b$.

5.1.6 Amplifier - Transient - SR Hyperbolic Tangent

A second model for the analysis of the transient in the integrator is using a hyperbolic tangent representation for the output current, Fig. 5.8. The main advantage of the $tanh$ model is the continuity of the function, without rough transition between the slewing and the settling phases.

The representation of the equivalent circuit is identical than the one for the SR (Fig. 5.11) replacing the constant current sources $\pm I_0$ by variable current sources I, according to the input voltage:

$$I = \pm I_p \tanh(\frac{V_e}{V_x}) \tag{5.22}$$

with $V_e = W_{irtp} - W_{irtn}$, $I_p = I_0$ and $V_x = I_0 / g_m$.

The equations of the charge conservation on the output nodes are defined, integrating the current I between the beginning ($t = 0$) and the end ($t = T_{conv}$) of the conversion:

$$\begin{aligned} \int_0^{T_{conv}} I_p \tanh\left(\frac{V_e}{V_x}\right) dt - Q_{cm} =& C_{lup}(V_{outp} - V_{cm}) - Q_{lup} + C_{up}(V_{outp} - W_{irtn}) \\ & - Q_{up} + C_{co}(V_{outp} - V_{outn}) - Q_{C_{co}} \\ - \int_0^{T_{conv}} I_p \tanh\left(\frac{V_e}{V_x}\right) dt - Q_{cm} =& C_{ldo}(V_{outn} - V_{cm}) - Q_{ldo} + C_{do}(V_{outn} - W_{irtp}) \\ & - Q_{do} - C_{co}(V_{outp} - V_{outn}) + Q_{C_{co}} \end{aligned} \tag{5.23}$$

The integrator is thus described with six equations, four for the charge conservation (5.1), (5.23), one for the common mode (5.2) and one condition on the output current (5.22). The first derivative of this system of six equations can be simplified, substituting the variables $\partial V_{outp/n}$, $\partial W_{irtp/n}$ and ∂Q_{cm} to get a single differential equation for V_e:

$$-I_p \tanh(\frac{V_e}{V_x}) = C_{eq} \frac{\partial V_e}{\partial t} \tag{5.24}$$

with the equivalent capacitor C_{eq}. The details of the computation of the capacitor are not provided as this computation it is not sophisticated. The variables are simply substituted to reduce the number of equations.

$$
\begin{aligned}
C_{eq} =& \frac{(C_{lup}+C_{ldo}+4C_{co})\left(C_{3do}C_{3up}+C_{cw}(C_{3do}+C_{3up})\right)}{2\left(C_{3up}C_{up}+C_{3do}C_{do}\right)} \\
&+\frac{C_{cw}\left((C_{up}+C_{do})(C_{3up}+C_{3do})-(C_{up}-C_{do})^2\right)}{2\left(C_{3up}C_{up}+C_{3do}C_{do}\right)} \\
&+\frac{(C_{inup}+C_{aup})C_{3do}C_{up}+(C_{indo}+C_{ado})C_{3up}C_{do}}{2\left(C_{3up}C_{up}+C_{3do}C_{do}\right)}
\end{aligned}
\tag{5.25}
$$

with $C_{3up} = C_{up} + C_{inup} + C_{aup}$ and $C_{3do} = C_{do} + C_{indo} + C_{ado}$. The analytical solution of the differential equation (5.24) is:

$$
V_e(t) = \frac{I_p}{g_m}\operatorname{asinh}\left(\sinh\left(\frac{V_{e0}g_m}{I_p}\right)e^{-\frac{tg_m}{C_{eq}}}\right)
\tag{5.26}
$$

The initial potential V_{e0} is obtained solving an initialization system, similar to the one described in section 5.1.5. Once the potential $V_e(t)$ is computed, the voltage of the other nodes is processed solving the following linear system:

$$
\begin{aligned}
&\begin{pmatrix}
1 & 1 & 0 & 0 \\
0 & C_{up} & C_{cw} & \text{-}C_{inup}\text{-}C_{up}\text{-}C_{aup}\text{-}C_{cw} \\
C_{do} & 0 & \text{-}C_{indo}\text{-}C_{do}\text{-}C_{ado}\text{-}C_{cw} & C_{cw} \\
0 & 0 & 1 & -1
\end{pmatrix}
\begin{pmatrix}
V_{outn} \\ V_{outp} \\ W_{irtp} \\ W_{irtn}
\end{pmatrix} \\
&=\begin{pmatrix}
2V_{cm} \\
\Sigma Q_{up} - V_a C_{inup} \\
\Sigma Q_{do} - V_b C_{indo} \\
V_e(t)
\end{pmatrix}
\end{aligned}
\tag{5.27}
$$

The computation of the voltages for an integrator with multiple inputs is similar.

5.1.7 Switches - Transient

This third section dedicated to the study of the transient in the integrator is focused on the settling time of the voltage of the internal nodes. The settling times are mainly limited by the switches, modeled by equivalent resistors in Fig. 5.12.

This analysis assumes the following hypothesis:

- The amplifier is ideal. It is not current limited, there is no offset and its gain is infinite. The current through the inputs is negligible and its inputs W_{irtp} and

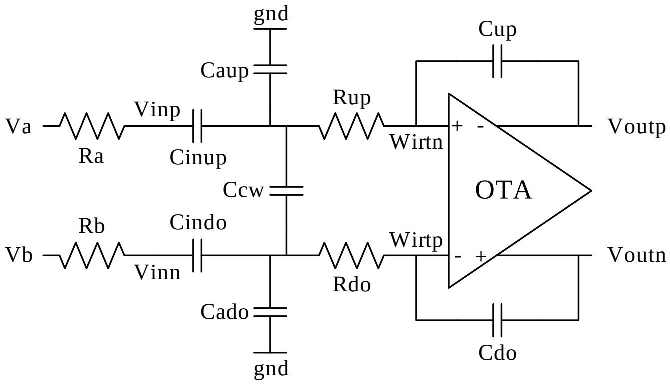

Figure 5.12: Equivalent model of the integrator with non-ideal switches.

W_{irtn} are at the same potential.

$$W_{irtp} = W_{irtn} = V_m \tag{5.28}$$

- The leakage is not significant and the resistance of the disconnected switches is infinite. These switches are thus not shown in Fig. 5.12.
- The on-resistance of the switches is constant during the transient. The computation or the input resistances is based on the potential V_a and V_b. The resistance of the central switches R_{up} and R_{do} is computed with the corresponding common-mode voltage.

Computation of the on-resistance

The on-resistance of the switches is computed considering the conductance of a PMOS in parallel with the one of an NMOS. The conductance of each transistor is given by:

$$G_{on} = wK_P \frac{V_g - V_{T0} - n(V_d + V_s)/2}{l} = wK_P \frac{V_g - V_{T0} - nV_f}{l} \tag{5.29}$$

where w and l are the dimensions of the switch, V_g is the gate voltage, $K_p = \mu C_{ox}$, n and V_{T0} are technology parameters and V_f is the analog voltage of the switch (V_f is the drain and the source voltage).

The computation of the equivalent resistor of the transmission gate is graphically summarized in Fig. 5.13.

In practice, to compute the on-resistance of the switches, simulations on the transistor level including process corners were run. The values of the parameters in (5.29) were extracted from these simulations considering a pessimistic case (more resistive).

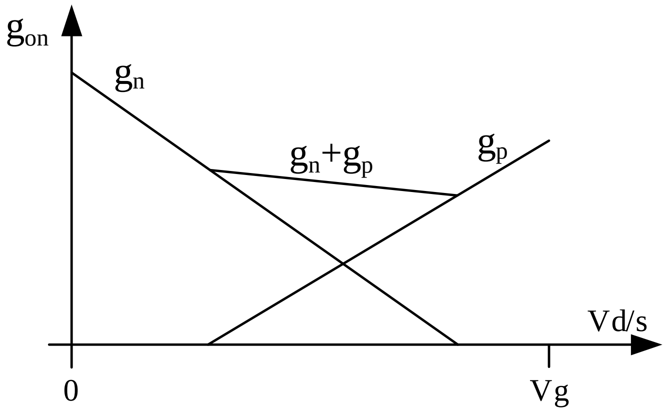

Figure 5.13: Graphical representation of the equivalent conductance of a transmission gate. The equivalent conductance is the sum of the conductance g_n of the NMOS and of the conductance g_p of the PMOS.

Transient

The sum of the currents in nodes $V_{inp/n}$ and $V_{p/n}$ provides the following differential equations:

$$\begin{aligned}\frac{V_a - V_{inp}}{R_a} &= C_{up}\left(\frac{\partial V_{inp}}{\partial t} - \frac{\partial V_p}{\partial t}\right) = C_{aup}\frac{\partial V_p}{\partial t} + C_{cw}\left(\frac{\partial V_p}{\partial t} - \frac{\partial V_n}{\partial t}\right) + \frac{V_p - V_m}{R_{up}}\\ \frac{V_b - V_{inn}}{R_b} &= C_{do}\left(\frac{\partial V_{inn}}{\partial t} - \frac{\partial V_n}{\partial t}\right) = C_{ado}\frac{\partial V_n}{\partial t} - C_{cw}\left(\frac{\partial V_p}{\partial t} - \frac{\partial V_n}{\partial t}\right) + \frac{V_n - V_m}{R_{do}}\end{aligned} \tag{5.30}$$

The differential equations for the potentials of the virtual ground nodes are:

$$\begin{aligned}C_{up}\left(\frac{\partial V_{outp}}{\partial t} - \frac{\partial V_m}{\partial t}\right) &= -\frac{V_p - V_m}{R_{up}}\\ C_{do}\left(\frac{\partial V_{outp}}{\partial t} - \frac{\partial V_m}{\partial t}\right) &= -\frac{V_n - V_m}{R_{do}}\end{aligned} \tag{5.31}$$

The equations (5.31) are simplified, introducing the common-mode relationship (5.2) to get a single equation (5.32):

$$V_m\left(\frac{1}{C_{up}R_{up}} + \frac{1}{C_{do}R_{do}}\right) - \frac{V_p}{C_{up}R_{up}} - \frac{V_n}{C_{do}R_{do}} = -2\frac{\partial V_m}{\partial t} \tag{5.32}$$

The equations (5.30) and (5.32) are written with a matrix form considering the substitutions for x, y, $\partial V_M/\partial t$: $x = V_p - V_m$, $y = V_n - V_m$ and $\frac{\partial V_m}{\partial t} = \frac{x}{2C_{up}R_{up}} + \frac{y}{2C_{do}R_{do}}$.

$$R\frac{\partial\vec{v}}{\partial t} = S\vec{v} + \vec{e} \tag{5.33}$$

with

$$\begin{aligned}
R &= \begin{pmatrix} C_{aup}+C_{cw} & -C_{cw} & 0 & 0 \\ -C_{cw} & C_{ado}+C_{cw} & 0 & 0 \\ -C_{inup} & 0 & C_{inup} & 0 \\ 0 & -C_{indo} & 0 & C_{indo} \end{pmatrix} \\
S &= \begin{pmatrix} -\frac{1}{R_{up}} - \frac{C_{aup}}{2R_{up}C_{up}} & -\frac{C_{aup}}{2R_{do}C_{do}} & -\frac{1}{R_a} & 0 \\ -\frac{C_{ado}}{2R_{up}C_{up}} & -\frac{1}{R_{do}} - \frac{C_{ado}}{2R_{do}C_{do}} & 0 & -\frac{1}{R_b} \\ \frac{C_{inup}}{2R_{up}C_{up}} & \frac{C_{inup}}{2R_{do}C_{do}} & -\frac{1}{R_a} & 0 \\ \frac{C_{indo}}{2R_{up}C_{up}} & \frac{C_{indo}}{2R_{do}C_{do}} & 0 & -\frac{1}{R_b} \end{pmatrix} \\
\vec{e} &= \begin{pmatrix} V_a/R_a \\ V_b/R_b \\ V_a/R_a \\ V_b/R_b \end{pmatrix} \\
\vec{v} &= \begin{pmatrix} x \\ y \\ V_{inp} \\ V_{inn} \end{pmatrix}
\end{aligned} \tag{5.34}$$

As the matrix R is constant, its inverse is pre-computable during the initialization. For each cycle, the product $A = R^{-1}S$ and $\vec{b} = R^{-1}\vec{e}$ is processed to get the ordinary differential equation $\partial\vec{v} = A\vec{v} + \vec{b}$.

A analytical exists at time t if the matrix A, the vector $\vec{b}$ and the initial conditions V_0 are known. Part of the operations to find the solution at time t requires a lot of operations. In particular, the decomposition of A into eigenvalues and eigenvectors is time consuming, as well as the computation of the inverse of A.

The decomposition of S into two matrices $S = CG$ reduces the number of operations for each cycle. G is a diagonal matrix with the conductances $1/R_{up}$, $1/R_{do}$, $1/R_a$ and $1/R_b$. As the entries of the C matrix are constant capacitors, the matrix C^{-1} is pre-computable. With this decomposition, the computation of $A^{-1} = G^{-1}C^{-1}R$ for each cycle is very fast as C^{-1} is already available and G is a diagonal matrix.

Particular cases

While the matrix S is always invertible, whatever the values of $C_{up/do}$, $R_{a/b}$, $C_{inup/do} > 0$ and $C_{aup/do} \geq 0$. The matrix R is sometimes singular. The matrix R has at least one zero eigenvalue in the following cases:

- $C_{cw} = 0$ and C_{aup} or C_{ado} is zero (one zero eigenvalue)
- C_{aup} and C_{ado} are equal to zero (one zero eigenvalue)
- $C_{aup} = 0$, $C_{ado} = 0$ and $C_{cw} = 0$ (two zero eigenvalues)

If R is singular, the system with four unknowns $R\partial\vec{v} = S\vec{v} + e$ has to be simplified into a system with three or two unknowns, according the number of eigenvalues of R. in particular, if C_{aup}, C_{ado} and C_{cw} are equal to zero, the matrix equation (5.34) is simplified into:

$$\begin{pmatrix} \partial x/\partial t \\ \partial y/\partial t \end{pmatrix} = \begin{pmatrix} -\left(\frac{1}{C_{inup}} + \frac{1}{2C_{up}}\right)\frac{1}{R_a+R_{up}} & -\frac{1}{2C_{do}}\frac{R_{up}}{R_{do}}\frac{1}{R_a+R_{up}} \\ -\frac{1}{2C_{up}}\frac{R_{do}}{R_{up}}\frac{1}{R_b+R_{do}} & -\left(\frac{1}{C_{indo}} + \frac{1}{2C_{do}}\right)\frac{1}{R_b+R_{do}} \end{pmatrix}\begin{pmatrix} x \\ y \end{pmatrix} \tag{5.35}$$

Multiple Inputs

The resolution of a modulator with 'n' input is not trivial in this case. Indeed, the number of nodes, and thus the number of unknowns, increases consequentially. The system of equations of a modulator with n inputs contains $4n$ unknown, instead of four V_p, V_n, V_{inp} and V_{inn}. The resolution of the system with multiple inputs is not detailed, but follows a similar procedure, with the writing of a system of ordinary differential equations:

$$\frac{\partial\vec{v}}{\partial t} = A\vec{v} + \vec{b} \tag{5.36}$$

The size of A is $4n$ x $4n$, and $\vec{b}$ is $4n$ x 1.

As well as for the single input case, if the matrix A is singular, the system has to be simplified to reduce the number of equations. Particularly, if the system is not including any capacitors C_{ai} or C_{cwi} (equal to zero), the size of A is $2n$ x $2n$.

5.1.8 Switches - Clock Feedthrough

When the switches are opened, the potential of the gate changes, injecting the charges of the capacitors C_{gd} and C_{gs} in the drain and source of the transistor, Fig. 5.14. The distribution of the charges between the drain and source nodes is not obvious, as some charges are still crossing the switch while it is being opened.

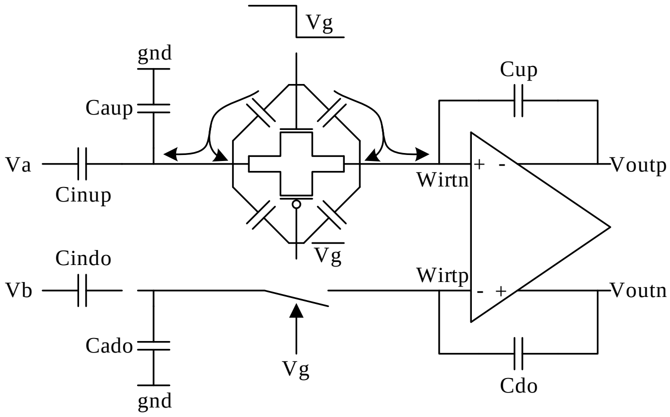

Figure 5.14: Illustration of the clock feedthrough for one of the switches on high-impedance nodes. While the switch is opened, the variation of the gate voltage V_g forces a charge transfer of the charges held in the Gate-Drain and Gate-Source capacitors.

In order to determine a model for the coupling with the clock, simulations were run with a transistor-level simulator. The simulation bench, Fig. 5.15, includes a transmission gate and an amplifier connected as an integrator. A resistance R_a is added serially with the input capacitance to model the line resistance (line, switches...).

The modeling of the opening of a transmission gate made of a PMOS and of an NMOS is not trivial. The charge injected by the PMOS can be partially absorbed by the NMOS if the gates are perfectly synchronized. If the gate controls are sequentially generated, the charges of the first opened transistor can still go through the second one. The considered model for the clock feedthrough is pessimistic, simulating a single transistor switch, without possible charge compensation.

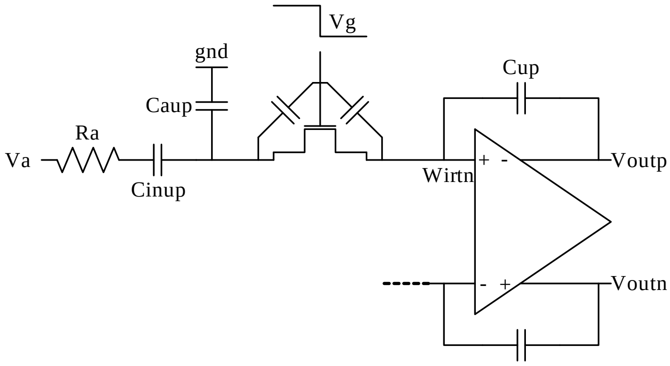

Figure 5.15: Simplified representation of the charge distribution test-bench. The line and sensor impedances are modeled with a serial resistance R_a.

Transistor level simulations showed that the overall charge of the switch Q_{tot} is divided between the source and the drain in function of the input resistance R_a:

$$\begin{aligned} Q_d &= Q_{tot}\frac{R_a}{R_a+R_c} \\ Q_s &= Q_{tot}\frac{R_c}{R_a+R_c} \end{aligned} \tag{5.37}$$

The R_c resistance is a constant depending of the amplifier. To provide an order of magnitude, the resistance of R_c of the simulated circuit was equal to 2kOhms. The charge of the switch is related to the gate voltage, to the dimensions of the switch and to the oxide capacitance.

$$Q_{tot} = wlC_{ox}V_g \tag{5.38}$$

The model of the integrator with multiple inputs assumed that the injections on each input branches were independent phenomena that could be processed sequentially.

5.1.9 Non-linear Capacitors

Several types of capacitors exist in integrated circuits. The most common capacitors in a CMOS technology are the MOS capacitor and the capacitors between metals.
The MOS capacitors (also sometime named moscap) use the gate oxide as the dielectric to create a capacitance between the gate and the other accesses of the transistor: source, drain and bulk. A separate well is usually required (unless one of the pin is connected to a power supply), promoting the use of PMOS transistors. The moscap have a fair capacitance per square-micrometer, but are voltage limited and are highly non-linear [46].
The capacitors between metals (MIM for Metal Insulator Metal) use a larger chip area, as the dielectric in metal layers is thicker. The MIM capacitors are either planar, between two or more metal layers (Fig. 5.16-a), with a top and a bottom electrode, or created with fingers, as shown in Fig. 5.16-b. In some technologies, there is a specific layer between the two standard top metal layers to create higher density MIM capacitors with a thinner dielectric.

As the MOS capacitors are not linear, they are usually not well suited in the switched-capacitors circuits, when the control of the gain is important. The capacitors between two metal layers are often characterized by the silicon foundries and are modeled

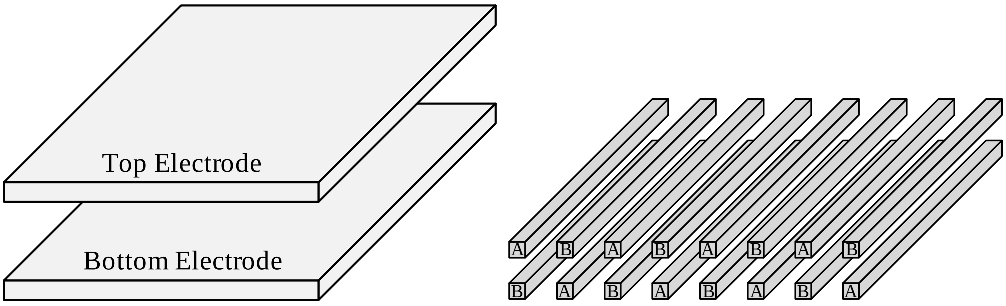

Figure 5.16: (a) MIM capacitance created between two metal layers. (b) Fingers capacitance, interleaving both electrodes on each metal layer.

with a second-order equation:[2]

$$C(v) = C_0(1 + \alpha v + \beta v^2) \tag{5.39}$$

The finger capacitors are, by nature, symmetrical. They do not have any top or bottom electrode and thus do not have odd order coefficients.

This section is focused on the integrator with non-linear capacitors (Fig. 5.17). Just as in all other sections of this chapter, the goal is again to compute efficiently (i.e. with a minimum of operations) the charge transfer. As there is generally no analytical solution with nonlinear capacitors, this section is divided into specific analyses, depending on the complexity level of the required computations.

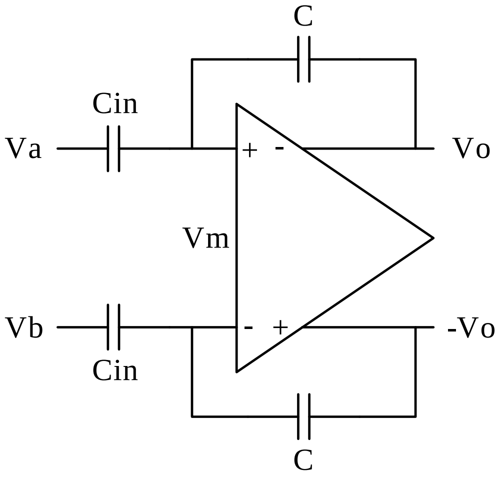

Figure 5.17: Equivalent model for an integrator with non-linear capacitors.

In this section, the non-ideal effects of the real amplifier are not taken into account.

---

[2] Several coefficients are also added for the temperature dependency, but the gain of the integrator is not affected by the temperature if all capacitors are matched and if the temperature is uniform.

The amplifiers are supposed ideal, with infinite gain, no offset and an output common-mode perfectly controlled:

$$\begin{aligned}
&V_{vitrual\,ground\,p} = V_{virtual\,ground\,n} = V_m \\
&V_{outp} = -V_{outn} = V_o
\end{aligned} \tag{5.40}$$

Moreover, the gain control of the integrator requires matched capacitors, of the same type and preferably created parallelizing many unit capacitors. As all these unit capacitors are of the same nature and have the same dimensions, their nonlinear coefficients are the same. I.e. if a capacitor $C1$ has a voltage dependency $C1(v) = C1_0 f(v)$, a matched capacitor $C2$ will also have the same coefficients: $C2(v) = C2_0 f(v)$. The voltage dependency function $f(v)$ is assumed to be identical for the two capacitors.

The charge transfer equations for the nodes W_{irtp} and W_{irtn} are:

$$\begin{aligned}
&(V_a - V_m)\,C_{in}\left(1 + \alpha\,(V_a - V_m) + \beta\,(V_a - V_m)^2\right) + \\
&(V_o - V_m)\,C\left(1 + \alpha\,(V_o - V_m) + \beta\,(V_o - V_m)^2\right) = \sum Q_{up} \\
&(V_b - V_m)\,C_{in}\left(1 + \alpha\,(V_b - V_m) + \beta\,(V_b - V_m)^2\right) - \\
&(V_o + V_m)\,C\left(1 - \alpha\,(V_o + V_m) + \beta\,(V_o + V_m)^2\right) = \sum Q_{do}
\end{aligned} \tag{5.41}$$

where ΣQ_{up} and ΣQ_{do} are the sum of the initial charges stored in the capacitors of the upper and lower halves of the differential integrator. It is possible to rewrite the two above equations, computing their sum and their difference:

$$\begin{aligned}
&C_{in}\left(V_a + V_b - 2V_m + \alpha\left(V_a^2 + V_b^2\right) - 2\alpha\,(V_a + V_b)\,V_m + 2\alpha V_m^2\right) \\
&+ C_{in}\beta\left(V_a^3 + V_b^3 - 3(V_a^2 + V_b^2)V_m + 3(V_a + V_b)V_m^2 - 2V_m^3\right) \\
&- 2C\left(V_m - \alpha(V_o^2 + V_m^2) + \beta V_m(3V_o^2 + V_m^2)\right) = \Sigma Q \\
&\\
&C_{in}\left(V_a - V_b + \alpha\left(V_a^2 - V_b^2\right) - 2\alpha\,(V_a - V_b)\,V_m\right) \\
&+ C_{in}\beta\left(V_a^3 - V_b^3 - 3(V_a^2 - V_b^2)V_m + 3(V_a - V_b)V_m^2\right) \\
&+ 2CV_o\left(1 - 2\alpha V_m + \beta(V_o^2 + 3V_m^2)\right) = \Delta Q
\end{aligned} \tag{5.42}$$

where $\Sigma Q = \Sigma Q_{up} + \Sigma Q_{do}$ and $\Delta Q = \Sigma Q_{up} - \Sigma Q_{do}$.

As the computation complexity depends on the order of the capacitor non-linearity as well as on the circuit operating conditions, various situations are handled. The separation into specific cases allows, if the system (5.42) has an analytical solution, to compute the solution much faster than using an iterative method to converge to the solution.

First-Order Nonlinearity

A capacitor with a first-order voltage dependency is of the form: $C(V) = C_0(1+\alpha V)$.

The system of equations (5.42) becomes in this case:

$$\begin{aligned}
&C_{in}\left(V_a + V_b - 2V_m + \alpha\left(V_a^2 + V_b^2\right) - 2\alpha\left(V_a + V_b\right)V_m + 2\alpha V_m^2\right)\\
&+2C\left(-V_m + \alpha V_o^2 + \alpha V_m^2\right) = \Sigma Q\\
&C_{in}\left(V_a - V_b + \alpha\left(V_a^2 - V_b^2\right) - 2\alpha\left(V_a - V_b\right)V_m\right)\\
&+2CV_o\left(1 - 2\alpha V_m\right) = \Delta Q
\end{aligned} \tag{5.43}$$

The output voltage V_0 can be expressed as a function of the virtual ground V_m:

$$V_o = \frac{\Delta Q - C_{in}\left(V_a - V_b + \alpha\left(V_a^2 - V_b^2\right) - 2\alpha\left(V_a - V_b\right)V_m\right)}{2C\left(1 - 2\alpha V_m\right)} \tag{5.44}$$

A 4^{th}-order equation is obtained for V_m, substituting V_0 in the first equation of (5.43). This last equation has a computable analytical solution (not detailed here). If this equation has more than one real solution, the voltages V_m and V_o are compared to those obtained solving the linear system (Section 5.1.1). The closest solution is then selected. This approach is reasonable if the non-linearity of the capacitor is small ($|\alpha| \ll 1$), which is the case for metal-metal capacitors.

Second-Order Nonlinearity

Finger capacitors created on one or several metal layers do not have odd order nonlinearities as they are symmetrical by construction. If the capacitors are created between two metal layers, with a top and a bottom electrode, it is possible to remove the first order effect by the parallel cross-connection of two identical capacitors.

The system of equations (5.42) becomes in this case:

$$\begin{aligned}
&C_{in}\left(V_a + V_b - 2V_m\right) + C_{in}\beta\left(V_a^3 + V_b^3 - 3(V_a^2 + V_b^2)V_m + 3(V_a + V_b)V_m^2 - 2V_m^3\right)\\
&-2CV_m\left(1 + \beta\left(3V_o^2 + V_m^2\right)\right) = \Sigma Q\\
&C_{in}\left(V_a - V_b + \beta\left(V_a^3 - V_b^3 - 3(V_a^2 - V_b^2)V_m + 3(V_a - V_b)V_m^2\right)\right)\\
&+2CV_o\left(1 + \beta\left(V_o^2 + 3V_m^2\right)\right) = \Delta Q
\end{aligned} \tag{5.45}$$

In a general case, this system of equation does not have an analytical solution. An iterative convergence method, as described in section 5.1.9, has to be used. Nevertheless, in practice, the input voltages V_a and V_b are often symmetrical (i.e. $V_b = -V_a$). The most common cases of symmetry are:

- The charge transfer is done by connecting the input capacitors to the common-mode.
- The common-mode of the reference voltage ($V_{refp}/2 + V_{refn}/2$) and the output common-mode are identical. The use of the power supply as the reference is the most common case, as the output common-mode of the integrators is often set to half of the power supply to maximize the output range.
- The signal is balanced with respect to the common-mode. It is the case if the integrator is driven by another integrator or by a front-end with the same output common-mode.

If the integrator was also symmetrical during the previous phases, there is a symmetry in the charges stored into the capacitors (initial state of the integrator, at the beginning of any phase). This fulfills the condition $\Sigma Q = 0$.

If this condition is valid, the first equation of (5.45) has an analytical solution for $V_m = 0$. The second equation of (5.45) becomes, substituting V_m with 0 and V_b with $-V_a$:

$$2C_{in}V_a\left(1+\beta V_a^2\right)+2CV_o\left(1+\beta V_o^2\right)=\Delta Q \tag{5.46}$$

As this third-order equation for V_o has an analytical solution, it is possible to compute efficiently the response of the integrator.

First- and Second-Order Nonlinearity

If the system of equations (5.42) does not have an analytical solution (i.e. none of the previously described cases applies), a numerical method is used to find the voltages of the integrator at the end of the charge transfer. It is possible to rewrite the system (5.42) to have a single 9th order equation for V_m:

$$f\left(V_m\right)=V_m^9+c_8V_m^8+\ldots+c_2V_m^2+c_1V_m+c_0=0 \tag{5.47}$$

As this function f is analytically differentiable and as its first derivative f' is easily computable, the Newton's iterative convergence method is well suited.

$$x_{n+1}=x_n-\frac{f\left(x_n\right)}{f'\left(x_n\right)} \tag{5.48}$$

The convergence speed of this method is quadratic but it can diverge, depending upon the set of initial conditions. An initial guess V_{m0}, close to the expected solution, is sometimes necessary for the method to converge rapidly. The Newton's method is first initialized with $V_{m0} = 0$ (input common-mode equal to the output common-mode).

The advantage of this first guess is that no extra computation is required and that it is relatively close to the solution. If the iterative algorithm diverges with $V_{m0} = 0$, a second trial is initialized using the linear solution (Section 5.1.1). As the nonlinear coefficients are small for metal-metal capacitors ($|\alpha| \ll 1, \left|\beta\right| \ll 1$), the linear solution is close to the expected one. If the iterative method still diverges, a combination of both dichotomy and Newton's method is applied to ensure convergence.

Multiple Inputs

The generalization for an integrator with multiple inputs is done without increasing the complexity of the resolution of the equations as the number of nodes (i.e. of unknowns) is identical. For an integrator with n inputs simultaneously connected, it is possible to reuse all the algorithms developed in this section, substituting:

$$\begin{aligned}
&C_{in}V_a \text{ with } \sum_i^n C_{in,i}V_{a,i} \\
&C_{in}V_b \text{ with } \sum_i^n C_{in,i}V_{b,i} \\
&C_{in}V_a^2 \text{ with } \sum_i^n C_{in,i}V_{a,i}^2 \\
&C_{in}V_b^2 \text{ with } ...
\end{aligned} \tag{5.49}$$

where $V_{a,i}$, $V_{b,i}$ and $C_{in,i}$ are respectively the voltage potentials and the capacitors of the input i.

5.1.10 Thermal Noise

Unlike the other sections of the modeling chapter, this one, dedicated to the thermal noise analysis in the integrator, starts with the non-trivial model for the pre-charge (without the amplifier) and continues with the familiar representation for the charge transfer.

The main thermal noise sources in the integrator are the switches and the amplifier. The noise of the switches, when active with an on-resistance R_{on}, is identical to the one of regular resistors:

$$S_{vt}(f) = 4kTR_{on} \tag{5.50}$$

The input-referred noise of the amplifier is, considering that the differential pair is the main noise contributor (gain g_m):

$$S_{vt}(f) = \alpha \frac{kT}{g_m} \tag{5.51}$$

In a simple or telescopic OTA, considering a noise model for transistors in strong inversion, the noise is [42]:

$$S_{vt}(f) = \frac{16}{3}\frac{kT}{g_m}\left(1 + \frac{g_{m,cm}}{g_m}\right) \cong \frac{16}{3}\frac{kT}{g_m} \tag{5.52}$$

with $g_{m,cm}$ the gain of the transistors in the current mirror (active load).

In the noise analysis provided in this section, the single-ended integrator is considered for simplicity reasons. In the differential structure, the noise of the two branches is not correlated and the overall noise is thus simply the sum of both contributions (the power is summed).

Pre-charge

The common case of a single input capacitor C has already been widely covered in the literature. The equivalent sampled input noise is the well-known kT/C.

The first model includes an extra capacitor C_a on the virtual ground of the amplifier. An equivalent representation of the noise (Fig. 5.18-b) is displayed with the schematic for the pre-charge (Fig. 5.18-a).

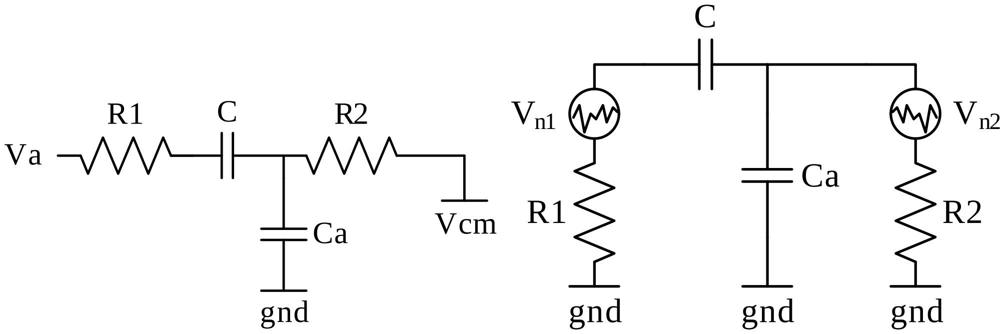

Figure 5.18: (a) Equivalent model of one of the input branches during pre-charge. (b) Small-signal representation including noise sources.

The noise induces an error on the charges stored in the capacitors, when the switched are being opened. The charge error due to the noise corresponds to the charge that should be injected into the node V_{ca} to come back to an ideal situation. The charge error Δ_Q is given by:

$$\Delta_Q = C_a V_{ca} - C\Delta_{V_c} \tag{5.53}$$

Substituting the capacitors with their complex counterpart $1/j\omega C$, this charge error is:

$$\Delta_Q = V_{n1}\frac{-C}{1+j\omega a+(j\omega)^2 d} + V_{n2}\frac{C+C_a+j\omega R_1 C C_a}{1+j\omega a+(j\omega)^2 d} \tag{5.54}$$

with $a = R_2 C + R_2 C_a + R_1 C$ and $d = R_1 R_2 C C_a$.

The voltage power of the equivalent noise referred to the input capacitor C is given by:

$$\begin{aligned}
\overline{V^2_{in,eq}} &= \overline{V^2_{in1}} + \overline{V^2_{in2}} \\
&= \int_0^{\infty} S_{vt1}(f)\left|H_1(j\omega)\right|^2 df + \int_0^{\infty} S_{vt2}(f)\left|H_2(j\omega)\right|^2 df \\
&= \int_0^{\infty} \frac{4kTR_1}{C^2}\left|\frac{-C}{1+j\omega a+(j\omega)^2 d}\right|^2 df + \int_0^{\infty} \frac{4kTR_2}{C^2}\left|\frac{C+C_a+j\omega R_1 C C_a}{1+j2\pi f a+(j2\pi f)^2 d}\right|^2 df \\
&= \frac{kT}{\pi C^2 R_2}\int_{-\infty}^{\infty} \frac{s+w^2 d^2}{1+\omega^2(a^2-2d)+\omega^4 d^2} d\omega
\end{aligned} \tag{5.55}$$

with $s = R_1 R_2 C^2 + R_2^2 C^2 + R_2^2 C_a^2 + 2R_2^2 C C_a$.

Short explanation: the sum of the powers is first computed in the first line of (5.55). The sampled noise is defined by the integral on the whole spectrum of the noise $S_{vt}(f)$ multiplied by the transfer function H; the square of the norm of H is used, as the equation is for the powers ((5.55), line 2). The noise sources $S_{vt}(f)$ are then substituted by the $4kTR$ model in line 3 and the transfer functions are replaced by the ones defined in (5.54). A division by C^2 is also done to convert a charge equation to a voltage equation. The result of (5.55) is obtained after the recombination of the left and right terms.

The computation of (5.56), using the theorem of the residues in the complex plane, is not detailed in order not to overload this section with formulas. The result is:

$$\int_{-\infty}^{\infty} \frac{s+\omega^2 d^2}{1+\omega^2(a^2-2d)+\omega^4 d^2} d\omega = \pi\frac{s+d}{a} \tag{5.56}$$

The equivalent charge is obtained, after the replacement of the temporary variables a,

s and d in the equations (5.55) and (5.56):

$$\overline{V_{in,eq}^2} = kT\frac{C+C_a}{C^2} \tag{5.57}$$

The same result is obtained if an extra parasitic capacitance C_f is added on the low impedance node, Fig. 5.19.

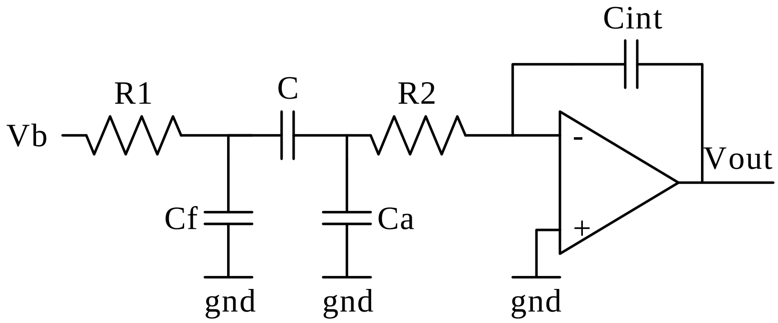

Figure 5.19: Simplified (single-ended) representation of the equivalent input of the integrator including parasitic capacitors.

Charge Transfer

During the charge transfer, the input capacitance is connected to the amplifier, Fig. 5.20-a. The noise of the amplifier is modeled adding an equivalent noise source V_{no} on one of the inputs. The gain of the amplifier is $I_{out}/V_{in} = g_m$. The equivalent model for the noise is displayed in Fig. 5.20-b. The switches are modeled with on-resistances and serial noise sources.

The charge error Δ_Q is given by:

$$\begin{aligned}\Delta_Q &= C_a V_{ca} - C\Delta_{V_c}\\ &= V_{n1}\frac{-C}{1+j\omega a+(j\omega)^2} + (V_{no}+V_{n2})\frac{C+C_a+j\omega R_1 C C_a}{1+j\omega a+(j\omega)^2 d}\end{aligned} \tag{5.58}$$

with $a = R_1C + R_mC + R_mC_a$, $d = R_1R_mCC_a$ and $R_m = R_2 + 1/g_m$.

The input-referred voltage error is computed using the same approach than for the pre-charge:

$$\overline{V_{in,eq}^2} = \frac{kT}{C^2}\left(C + C_a + \left(\frac{\alpha}{4} - 1\right)\frac{1}{g_m R_m}\frac{R_mC^2 + R_mC_a^2 + 2R_mCC_a + R_1CC_a}{R_1C + R_mC + R_mC_a}\right) \tag{5.59}$$

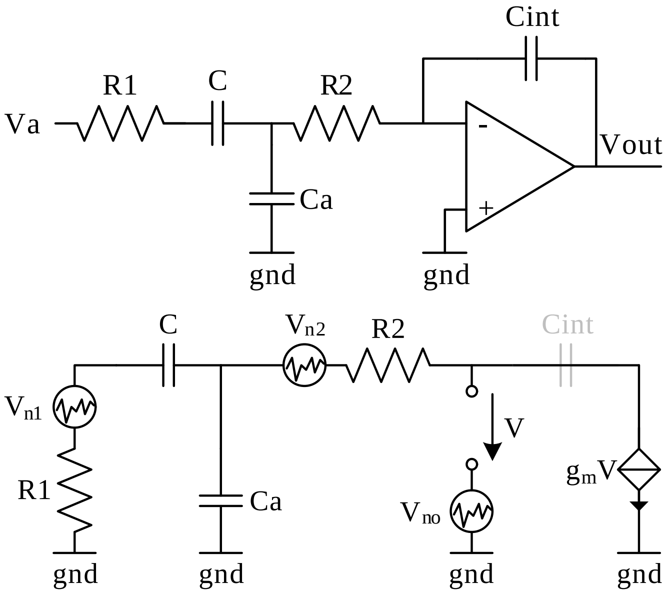

Figure 5.20: (a) Equivalent simplified model of the integrator during the charge transfer. (b) Small-signal representation including noise sources.

This equivalent input voltage is bounded by:

$$\begin{aligned}\overline{V^2_{in,eq}} &< kT\frac{C+C_a}{C^2}\left(1+\left(\frac{\alpha}{4}-1\right)\frac{1}{g_m R_m}\right) = kT\frac{C+C_a}{C^2}\left(1+\left(\frac{\alpha}{4}-1\right)\frac{1}{1+R_2 g_m}\right)\\ &< \frac{\alpha}{4}kT\frac{C+C_a}{C^2}\end{aligned} \tag{5.60}$$

To summarize, the overall noise during a full clock cycle (pre-charge and charge transfer) can be related to the input of the integrator summing the equations (5.57) and (5.59). Considering a simple model for the amplifier ($S_{vt} = 16/3kT/g_m$), the input referred noise for a full modulator cycle is:

$$\overline{V^2_{in,eq,fullcycle}} < \frac{7}{3}kT\frac{C+C_a}{C^2} \tag{5.61}$$

5.1.11 Quantizer

After the integrator, the second block of the ΣΔ modulator is the quantizer, also named comparator in a binary case. A quantizer with a non-binary output is named multi-levels or multi-bits quantizer.

The use of multi-bits quantizers increases the precision of ΣΔ modulators, reducing the quantization error. They are frequently used in single-loop high-order modulators for stability reasons [47].

The main limitations of the converters, besides the assumed sufficient response time, are the linearity and the hysteresis.

Linearity

As a quantizer is made of real components, the levels are not perfectly distributed, Fig. 5.21. The linearity error is simply modeled, adding to any ideal level n_{idl} an error generated with a probability distribution sigma:

$$n = n_{idl} + err(\sigma) \tag{5.62}$$

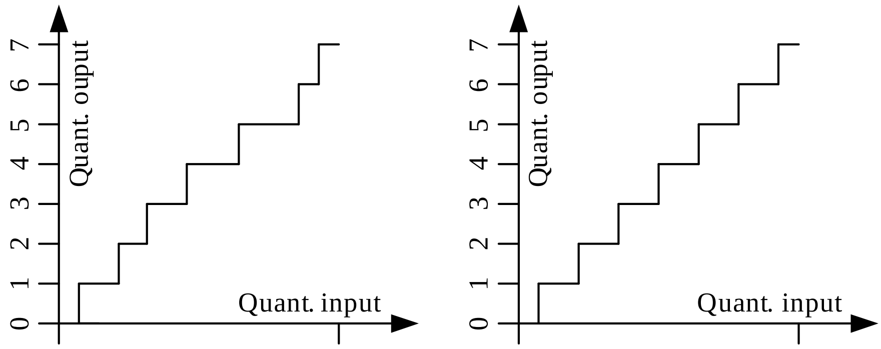

Figure 5.21: Linear and non-linear transfer functions.

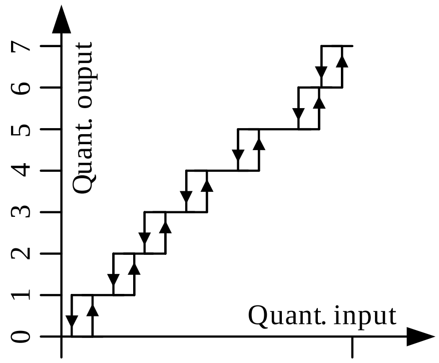

Figure 5.22: Non-linear transfer function with hysteresis.

Hysteresis

A way of modeling the hysteresis of the comparator is to define two sets of levels instead of a single one. The higher levels are defined adding half of the hysteresis to the original levels (the ones without hysteresis). Similarly, the lower levels are created subtracting the same value, Fig. 5.22.

During the quantization, input of the quantizer is compared to the input of the previous quantization (in a discrete time comparator, to the value one clock cycle earlier). The higher or the lower levers are selected accordingly.

5.1.12 DAC

The last component in the ΣΔ modulator is the Digital-to-Analog Converter. In a switched-capacitors circuit, the DAC is usually made of programmable capacitors whose value depend of the input code, Fig. 5.23.

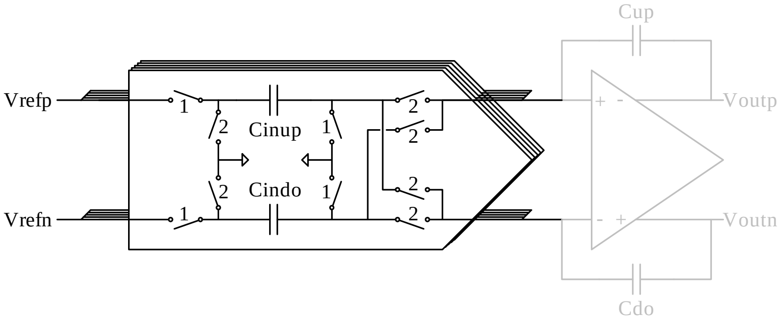

Figure 5.23: DAC implementation with switched-capacitors.

As the DAC is a set of capacitors connected to voltage sources (voltage reference) with switches, there is no specific model for the DAC.

The mismatch between the capacitors of the DAC induces a linearity error. For this reason, the DAC is one of the most sensitive blocks in multi-level ΣΔ modulators.

To restrict the impact of the mismatch of the capacitors, several methods to scramble the capacitors exist, to average the errors over several clock cycles. These techniques are grouped under the Dynamic Element Matching (DEM) name.

Ideally, in a DEM, the capacitors should be scrambled randomly. Practically, a random number generator is hard to implement, and thus, simpler methods have to be used. The most-common DEM encoders are:

- Barrel shifter: In this method, the capacitors are used in turns, always in the same order [48].
- Butterfly shuffler: The DEM encoder is made of unit switching cells, cross connected [49][50].
- Vector feedback [51][52].
- Tree structure encoder: Generic structure with a minimal number of switching blocks [53][54].

5.2 Linear Circuit

In the previous sections, a classic approach was introduced to model the sigma-delta modulator with basic blocks. In the classic approach, sub-circuits are computed sequentially. E.g, a second-order two-stages modulator (MASH 1-1) would be simulated, computing for each clock cycle firstly the output of the first stage and secondly the output of the second stage. The computation of the first stage is completely independent of the one of the second stage, as the 2^{nd} is not driving the 1^{st} one.

Such an approach is effective to analyze the local non-ideal effects, essentially inside the integrator. On the other side, it cannot simulate external perturbations, with digital signals or between distinct blocks. The aim of this section is to built a model for the whole modulator, including all nodes and all interconnections between nodes. This model has to: simulate coupling between nodes, easily estimate the impact of a capacitive mismatch and validate the local digital logic (i.e. all the gates in the analog block and in the same layout area, but that are not implemented using a synthesized code).

The first reason of a higher level modeling is either abstraction to ease the comprehension, either the mathematical simplification to reduce the computation time of a simulator. The goal of the model described in this section is to speed up the computation. In switched-capacitors circuits, an approach in the charge-voltage (QV) regime is possible, when the useful information in exclusively held in the voltages at the end of the modulator cycles (stable state). The hypothesis on the circuit are:

- Discrete time: the transient phenomena are not taken into account. The state at the end of the charge conversion is assumed to be stable.
- Only ideal components are considered: the two states of the switches are either on ($R = 0$) or off ($R = \infty$), the capacitors are linear, the amplifiers are linear or ideal, and the resistances are not taken into account.

The main idea of the proposed model is to split the circuit into two distinct parts, Fig. 5.24. The first one includes the voltage sources, the capacitors, the switches, the operational amplifiers and all other linear components. The second one only includes the quantizers and the high-level (synthesized) logic of the circuit: digital controls, decimation filters... In a charge-voltage representation, the voltages at the end of a sequence are related to the initial conditions of the sequence with a system of linear equations. The complexity to solve the system is O(n3) for classic algorithms (Gauss-Jordan).

A circuit with switches is generally non-linear. The main idea of this model is to consider, instead of a non-linear circuit with switches, a collection of linear circuits

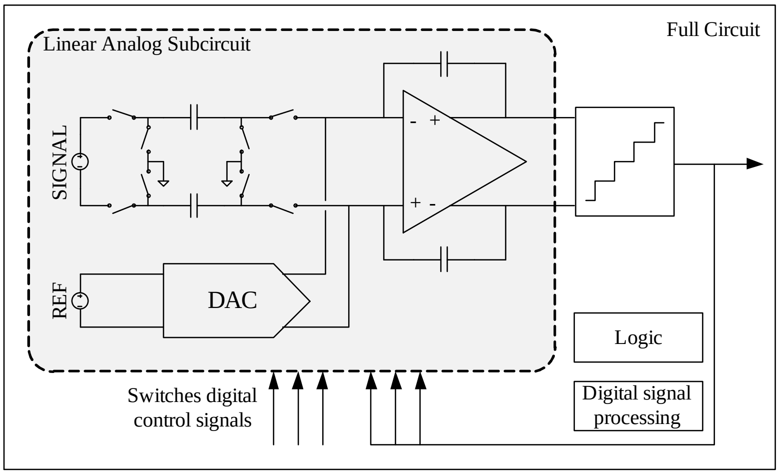

Figure 5.24: Splitting of a circuit into linear (gray) and non-linear (white) subcircuits.

without switches. Each linear sub-circuit is named state and correspond to a particular configuration of the switches. The number of states of a system is much smaller than the overall number of cycles of a ΣΔ modulator (typically 3 states for a single-bit first-order modulator: pre-charge, reference subtracting and reference addition). It is thus preferable to work with a restricted number of optimized states to minimize the number of operations to perform during each cycle instead of naively solving a huge linear system in each cycle.

5.2.1 Matrix Representation

The circuit can be described, as every system of linear equations, with a matrix form. The system depends of the initialization variables (initial charges/voltages, $\vec{b_n}$), and of the unknowns are the voltages at equilibrium, $\vec{a_n}$. The number of lines n in the system is equal to the number of nodes in the circuit.

$$L_{n_x n}\vec{a}_n = R_{n_x n}\vec{b}_n \tag{5.63}$$

Each line represents either a voltage source, a closed switch, an amplifier or a charge conservation equation. A matrix line for a source is empty, except a unique entry equal to 1. A closed switch has two non-zero entries, 1 and -1. Two lines are necessary to describe an ideal differential amplifier: the first one defines a symmetry for the output common-mode (two non-zero entries equal to 1 in the matrix), while the second line sets the same potential for the two inputs (1 and -1 like the switch). Finally, in a line for a charge conservation equation, the number of non-zero entries is determined by the number of capacitors on the node (two capacitors in parallel are viewed as a unique component).

Optimization

As stated previously, it is more efficient to optimize the few states (set of linear systems) of the modulator instead of solving 'from scratch' a huge linear system during each cycle. The under-mentioned steps optimize the resolution of the system with a pre-processing of the states.

Reduction of the number of nodes: Each state is characterized with two matrices L and R. The size of both matrices is n x n. The first part of the optimization reduces the matrices size, assigning only one variable per node on the same potential, mainly active switches and amplifiers inputs. The disconnected nodes are also eliminated at that time.

$$L_{m_x m} \vec{a}_m = R_{m_x m} \vec{b}_m, \; m \leq n \tag{5.64}$$

Inversion of the matrices: the system is rewritten in order to have the output voltages as a direct function of the inputs.

$$\vec{a}_m = L^{-1}_{m_x m} R_{m_x m} \vec{b}_m = S_{m_x m} \vec{b}_m \tag{5.65}$$

Before the inversion, the consistency of the matrix has to be checked. The matrix may not be consistent if there is a wrong connection in the netlist, resulting in too many or not enough constraints. It is possible to accomplish the inversion only once in a pre-processing phase as all entries of the matrices L and R are constant. Indeed, these matrices are filled with unit entries and capacitors values.

Fragmentation: if the resulting matrix is sparse, which is the case when the nodes are only connected to a few components, it is more efficient to store and compute only the non-zero entries of the matrix. The matrix is transformed into a linear system of equations, removing the zero entries, in order to minimize the number of operations to be performed.

5.3 Conclusion

This chapter detailed the modeling of switched-capacitors ΣΔ converters on the implementation level. The proposed modeling features a description of the component that is not based on functional blocks, but based on circuit components.

The main challenge of this modeling was to provide accurate models while targeting a low mathematical complexity, the final goal being to integrate the models in a simulating software.

The required computing power to solve a system grows with the size of the circuit. As the complexity is not increasing linearly with the number of nodes, the most accurate models cannot be applied directly on the full circuit. A prior partitioning of the circuit is thus necessary.

A dual approach was considered to overcome this constraint. The integrator was first analyzed and models including common imperfections were derived. In order to guarantee a reasonable computing power, the non-ideal effects are addressed individually.
Global models of modulators were then integrated in a linear simulator. The key concept to speed up the computations is based on the highly repetitive nature of $\Sigma\Delta$ converters. As each phase is repeated many times, its simulation can thus be optimized in a prior processing.

6 Simulator

Mathematical models were developed in the previous chapter. Their implementation in a simulator dedicated to switched-capacitors ADCs is detailed in this chapter. The first section reviews the design flow of ΣΔ ADCs and identifies the time consuming design steps. The software architecture of the simulator is then described, and two tools - for statistical simulations and capacitive coupling analysis - are introduced. The last sections of this chapter describe the environment around the simulator (test-benches) and the use of the simulator in a comparison of architectures.

6.1 Design steps and simulators

While designing a ΣΔ A/D converter, the engineer is usually validating his work through several design steps. This section details a standard top-down approach. The goal of each step is described, in the analog and digital domains, and a list of the available tools is provided.

This section is divided into five portions, from the high-level definition of the architecture up to the validation of the layout of the converter. The design strategy is separated for the analog and digital aspects of the modulator, whose fundamental elements are shown in Fig. 6.1.

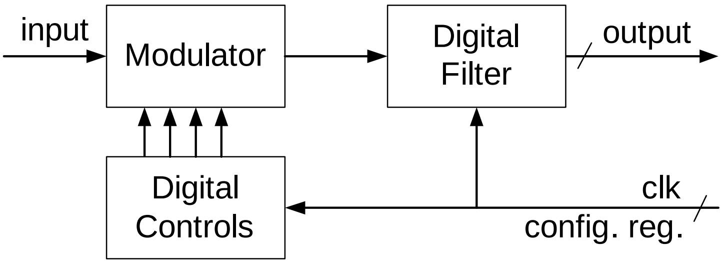

Figure 6.1: Analog and digital fundamental elements of a ΣΔ converter.

6.1.1 Definition of the architecture

Analog: The first step in a top-down design is to roughly determine the architecture of the modulator with the specifications. The topology is mainly determined by the quantization noise, setting the order. The available options are:

- Order of the modulator: Determined with the sampling frequency and with the oversampling frequency. The order should be large enough to lower the quantization noise below the desired resolution within a number of cycles determined by the OSR.
- Single-loop / Multiple-loop topology: A structure with multiple stages - MASH - is more effective to reduce the quantization noise and is stable if the order of each stage is equal or lower than 2. A single-loop structure is on the other side more compact, as it only requires one quantizer and lightens significantly the design constraints on some base blocks, e.g. the DC gain of the amplifiers.
- Feedback or Feedforward structure: A feedback topology requires a good matching between the DACs of each integrator, but save the extra adder of feedforward structures.
- Number of levels of the quantizer: A quantizer with several levels has many advantages, as lower quantization noise, better loop stability and reduced dynamic power consumption. However, multi-bit architectures require dynamic matching of the DAC capacitors to compensate the non-linearity introduced by the mismatch of the unit components.

Digital: Definition and validation of the transfer function of the digital filters in a high-level language (Matlab code, C code...).

The selection and the validation of an architecture are two relatively complex steps, due to the non-linear nature of the ΣΔ modulators. Many simulators and toolboxes were created to help the designer making the good choices. The most popular are probably Richard Schreier's toolbox [42, 55] and SIMSIDES [43, 56], a simulator developed in a Matlab Simulink environment by José M. de la Rosa et al. These simulators, coded in high-level programming languages, provide quickly first results and are thus well-suited to compare easily various topologies.

6.1.2 Implementation with switched capacitors

Analog: The next step, after the definition of the architecture of the ΣΔ modulator is its implementation with switched capacitors. The architectural comparison of chapter 4 showed that all implementations are not equivalent in terms of noise and power consumption. The sensitivity to imperfections should not be neglected either.

Digital: In order to control the modulator, the sequence of the controls driving the analog block has to be determined. Defining the digital controls may be a trivial task for classic $\Sigma\Delta$ converters with a limited number of phases. The control of incremental converters is, in contrast, much more complex, especially if it is combined with special implementations. The complexity is increasing with the number of independent digital commands. E.g. the incremental ADC implemented in the end of this thesis has more than fifty different states, while a $\Sigma\Delta$ modulator may only have two.

In order to validate the switched capacitors implementation, transistor level simulators are most of the time used. To speed up the simulation and as the sub-circuits are not yet designed with transistors, macroblocks are used instead of amplifiers, voltage sources, current sources, quantizers...

The simulation time of a converter modeled with macroblocks is much more important than the simulation time with a high-level simulator. The main reason is the continuous-time simulation of a discrete-time converter. During each cycle of the ADC, and depending of the defined precision for the voltages and currents, transistor-level simulators compute many points. The density of points is especially high during the transients, when the signals are strongly varying due to the charge transfer. These simulators are thus not well-suited to compare and select implementations with switched capacitors. Moreover, it is often not possible to estimate easily the impact of non-ideal effects using such simulators.

6.1.3 Implementation of the digital blocks

The aim of this step is to validate the low-level coding and the synthesis of the digital function with digital simulators.

Analog: In order to be compatible with the digital simulators, a model of the modulator is written in a low-level programming language such as vhdl, vhdl-ams, verilog...

Digital: The filters and digital control are written with a code synthesizable with logic gates. The digital part is validated using the analog models written in the same language.

The classic simulation and synthesis tool are used, e.g. Modelsim, RTL compiler, Encounter, Olympus... The digital simulators are fast as they are event driven, which allows an efficient and quick validation of the digital implementation.

6.1.4 Design of the analog blocks

Analog: The basic blocks constituting the modulator are created and validated independently according to their respective specifications. The main blocks are the amplifiers, the quantizers, the current sources for the biasing and the voltage sources for 1) the common-mode, 2) the cascode transistors biasing and 3) eventually for the reference.

Digital: If necessary and depending on the design, additional delays due to the implementation with transistors should be evaluated. If the digital is powered by a supply voltage lower than the voltage of the gates, the level shifters or bootstrap circuits introduce a delay. It is often required to ensure that the controls reach the switches in a correct order. Local non-overlap circuits inside the blocks may be implemented.

The simulators used in this step are transistor level simulators as Spectre, Eldo... The validation of the individual blocks is fast as the sub-circuits only contain a limited number of transistors. Moreover, as most of the elementary blocks are not discrete time, a first sizing and validation iteration is efficiently performable with very fast AC simulation. The slower transient simulation is then kept for the final validation of the blocks.

6.1.5 Post-layout validation

During the drawing of the layout, the performances of each element of the modulator are evaluated individually. The required simulations are fast as long as each block contains a small number of nodes.

Analog: A problematic step, as it requires a very important simulation time, is the validation of the full modulator including the parasitic components extracted from the layout. Mixed-mode simulators such as ADVanceMS are used, simulating both analog and digital part with specific simulators. This approach is particularly efficient if many transistors are included in the digital part, which is more and more frequent with recent processes. The main disadvantage of the validation of the parasitic coupling from the layout with a low-level simulator is the delay between successive iterations. A few days up to several weeks are often required to get only a few points of the transfer function.

Digital: The last action for the digital is the validation of the A/D and D/A interfaces of the full modulator (i.e. the interfaces between the synthesized blocks and the modulator). A simulation of a few clock cycles of the modulator with an analog simulator allows checking after the synthesis and the layout that the sequence of the digital controls on the gates of the switches is correct.

6.1.6 Summary of the design steps

The table 6.1 summarizes the design steps of a ΣΔ A/D converter with switched capacitors, including the commonly used tools in each step.

View	Analog Task	Digital Task	Tools
High level	Definition of the architecture	Definition and verif. of the digital filter	Matlab / Simulink toolboxes
Analog Implementation	Definition of the implementation	Definition and verification of the control signals	Low-level simulator with macroblocks
Digital Implementation	Digital models of the analog blocks	Coding / validation of the VHDL/verilog implementation	Digital simulators and synthesizers
Transistor level	Design and validation of the blocks	Integration of the levelshifters and bootstrap circuits	Mixed-mode simulator
Layout	Post-layout validation of the sub-blocks	Check of the D/A and A/D interfaces	Transistor-level simulator

Table 6.1: Main design steps of a ΣΔ ADC with a top-down procedure.

A fast design is guaranteed if each step is either using a fast simulator, or including a minimal number of iteration (validation only). With this definition, two steps are problematic in the simulation flow, table 6.1.

- The validation of the implementation. A low-level continuous-time simulator is used. It is thus not possible to evaluate the quality of the selected implementation using many iterations.
- The verification of the parasitic coupling. It is especially important for switched capacitors circuits in which parasitic capacitors may directly create charge errors and degrade the resolution of the converter. The classic validation of the full layout of the modulator is only performed at the end of the design and requires an enormous validation time after each enhancement of the layout to reduce parasitic coupling.

6.2 Specifications

The simulator has to fulfill the following specifications, on top of the previously defined features:

- Implementation of the models defined in the previous chapter
- Easy definition of a new architectures. The architecture and the simulator core have to be located in separate files
- The simulator has to compute, for a given configuration, an INL or an FFT, depending on the modulator operating mode (incremental, sigma-delta).
- The simulator configuration options are located in a text file. The simulation results are also exported to text files, Fig. 6.2.
- The simulator has to be designed for fast simulations. In addition to the optimizations defined in the modelling chapter, a smart management of both memory and IOs has to be used.

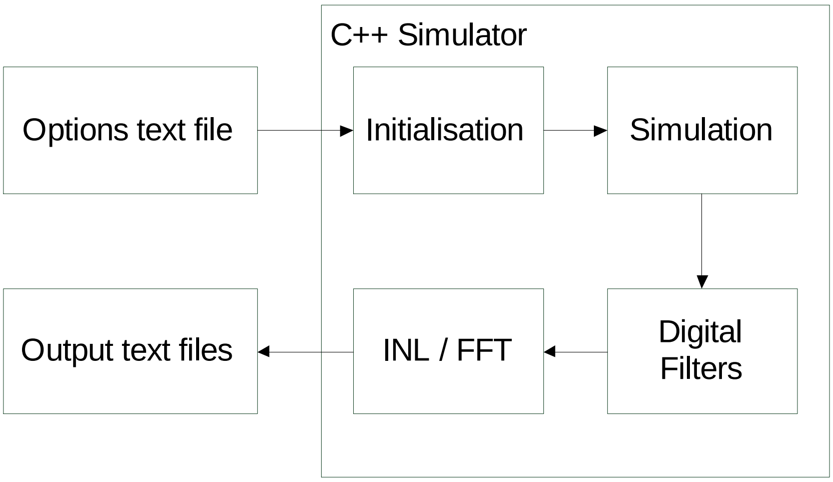

Figure 6.2: IOs and main processing functions of the simulator.

The C++ programming language has been selected as it is object oriented (to define easily a new architecture), compiled (for speed) and with a strong variable casting (for the memory management).

6.3 Simulator Core

The simulator is made of three relatively distinct major parts:

- The upper part includes the IOs with the configuration and output files, the management of the successive conversions (stair-shaped input) for the INL, and the computation of the INL and of the FFT.

- The integrator-based simulator.
- The simulator based on a linear view of the entire modulator.

A graphical representation of the objects (classes) of the simulator is shown in Fig. 6.3, where the separation of the three main parts is specified. The integrator-based simulator is in blue, the linear simulator in green and the common objects in gray. The specific classes of a generic architecture XYZ (orange) are easily discernible. An object inside an other one inherits the functions and variables from the parent class.

6.3.1 Simulator - Top Classes

In this part, the configuration file is first read, and stored into memory (option class). The simulator instantiates then a modulator for the selected architecture and, depending on the desired simulation (incremental or sigma-delta) creates a stair signal or a sine wave to drive the modulator. Once the conversion performed, the simulator exports into text files the ADC digital output, the INL or the FFT and several high-level parameters (ENOB, STD, Best Fit slope, offset...). These parameters are summarized in Fig. 6.6, in the application section (6.4).

The classes of this part are:

- **Options**: The internal representation of the configuration text file
- **Techno**: The process specific parameters are stored in this object
- **TSignal**: A sampled signal object. The signals are described by two variables, amplitude and time. The class includes functions to compute the INL and the FFT of a signal.
- **Main**: The program starts here. This is not an object, but the main instantiations are created and the ADC conversions are driven from here.

6.3.2 Simulator - Architecture Classes

For each architecture XYZ, a dual view is created. If the integrator-based models are used, the ADC is made of one or several stages (classes **Stage_XYZ_1**, **Stage_XYZ_2**...), and each stage instantiates one or several integrators, DACs and one quantizer. The stages are connected together in a modulator (class **ModXYZ**). The digital filter of the XYZ architecture is in an upper class (**ModGen_XYZ**) because it is common to the integrator-based model and to the linear circuit model.
The architecture definition class for a linear circuit (**LCG_ModXYZ**) defines the components of the circuit: analog nodes, digital signals, capacitors, switches, amplifiers, sources and logic gates. The states of the linear circuit as well as the sequence of the digital controls are defined.

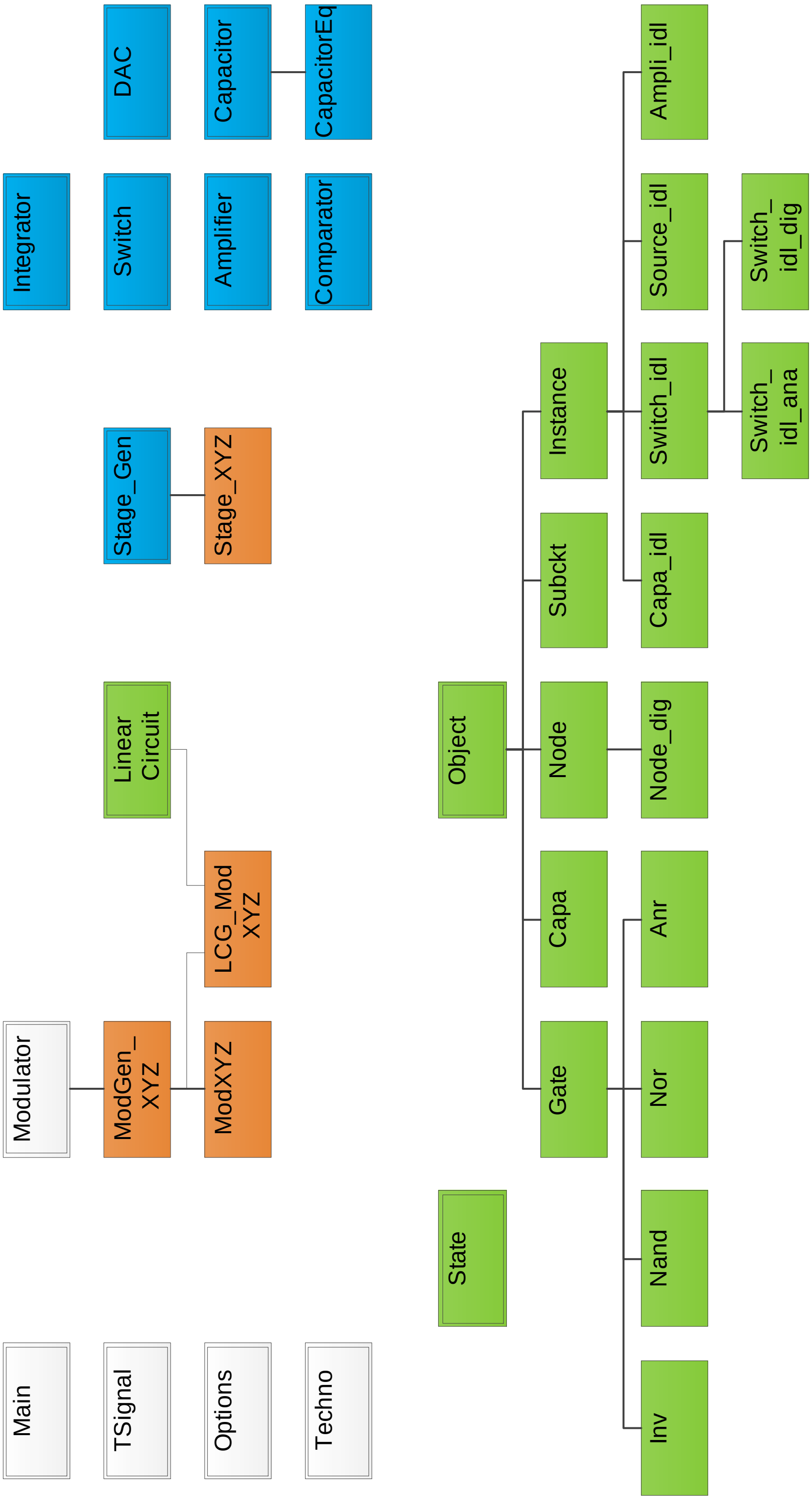

Figure 6.3: Classes of the simulator. Connections are inheritance from parent class. Colors are used to distinguish the two simulators (blue - green), the definition of the topology (orange) and common files (gray).

The architecture specific classes are:

- **Stage_XYZ**: A stage of the modulator. The switches, capacitors, amplifiers and integrators are instantiated. The conversion sequence for the stage is also defined.
- **ModXYZ**: Instantiation of the stages and connection of the stages together.
- **LCG_ModXYZ**: Creation of the components of the linear circuit, of the states and of the digital sequence.
- **ModGen_XYZ**: Definition of the digital filter.

The abstract class Modulator is not related to a specific architecture XYZ, but it provides a template to create the architectures and defines a common interface for all architecture to communicate with the classes of the upper part.

6.3.3 Simulator - Integrator Classes

This subsection describes the behavior of the core of the simulator based on the integrator. When the architecture is created, each stage (class **Stage_XYZ**) instantiates the basic components: switches, capacitors, DACs and amplifiers. The model of the integrator defined in the previous chapter only includes closed switches. As shown in the model definition, an integrator with several phases can be replaced with a set of integrators without any phases (Fig. 5.3). Several instances of the integrator class are thus created per real integrator, one for each phase. The base components of the integrator are not copied when the integrators are instantiated, as some elements are shared between the instances (typically the feedback capacitors and the integrator). Implementation is done with pointers.

Three states are defied in the integrator: pre-charge, charge transfer and reset. The class of the integrator includes thus three functions to solve these three states. In the two first states, the input variables are the potential V_a and V_b. When one of these functions is called, the integrator applies a model of the previous chapter. The selection of the model to use is defined in the configuration file (see section 6.4.1). The voltages at the end of the phase are computed and the load of the capacitors are updated.

The specific classes of this section are:

- **Comparator**: Compares the differential input signal to defined levels. During the creation of the comparator, the value of the thresholds is generated, according to the number of levels, to the hysteresis and to a standard deviation for the internal components mismatch.
- **Switch**: The switches (transmission gates) are initialized with their physical

dimensions W_n, W_p, L_n and L_p. Functions are included to compute the on-resistance and the charge distribution when the clock feed-through model is used.
- **Amplifier**: The amplifier is defined with a DC gain, an offset, a biasing current, an output common-mode and saturation voltages. Several gain functions (section 5.1.4) and the corresponding derivative are available.
- **Capacitor**: A capacitor is defined with its value and with the non-linearity coefficients, alpha and beta. The base value is altered during initialization with a dispersion sigma, taking into account the number of unit components.
- **CapacitorEq**: In this particular class of capacitors (inherits of the **Capacitor** class), the capacitance value is editable. This class is used in the multi-level DACs to implement DEM algorithms.
- **DAC**: This class is defined as an integrator with editable input capacitors (with **CapacitorEq**). The previous state of the DAC is stored into memory for the DEM algorithms.
- **Integrator**: Implements the charge transfer, pre-charge and reset functions. During the creation of an integrator, pre-computed matrices are stored in the program memory (see chapter 5).

6.3.4 Simulator - Linear Circuit

The steps to create a linear circuit are: the analog and digital nodes are firstly created, each analog node is named. The digital signals are named and are referenced to the two digital power potentials, corresponding to the logic levels. The instances of the components are then created for the sources, capacitors, switches and amplifiers. Each instance is named and the name of the nodes are used to connect the ports of the instances.
The last step in the definition of an architecture is the creation of a list containing the states (for each state, the active controls are listed). The conversion sequence is then created, indicating in which order the states are used. This sequence is included in the function called from the upper part (section 6.3.2) to perform a conversion.

In addition to theses mandatory components in the linear circuit, other elements have been put in the simulator to ease the coding of an architecture.
Objects for digital gates are included. The main advantage of using digital gates is the definition of internal digital signals. Nodes are easily added and above all, the definition of the states is simplified, defining only the digital inputs of the modulator. The status of the digital internal nodes of the modulator is computed while the states are initialized; the digital nodes are connected to the digital supplies through switches. The second major feature to facilitate the definition of an architecture is the introduc-

tion of sub-circuits. With sub-circuits, the view is hierarchical and existing blocks are reused. A sub-circuit is defined identically to an architecture, adding nodes, components, logic gates, sub-circuits and ports for the connections.

The last feature in the definition of an architecture is the import of parasitic capacitors, extracted from the layout of the modulator. The import requires identical naming of the nodes and of the hierarchy in the transistor level simulator netlist and in the high-level linear circuit netlist. If a node is misnamed, the related capacitors are not created. This feature adds readily many capacitors (typically more than 10k capacitors in the designed ADC, chapter 7). Sort function have been implemented to speed up the identification of problematic capacitors. This feature is used in the layout validation of the ADC in the next chapter.

Once all the components are added into the circuit, the simulator is creating, during initialization, an internal structure connecting the components together (pointers are used, e.g to list in an instance of a node, the components connected to this node). The optimized matrices of the states are then created to alleviate the computation load for each cycle. A conversion is driven with the modulator input signal, as well as in the integrator-based simulator.

A simplified view of the simulation flow of a full conversion using the linear simulator is shown in Fig. 6.4.

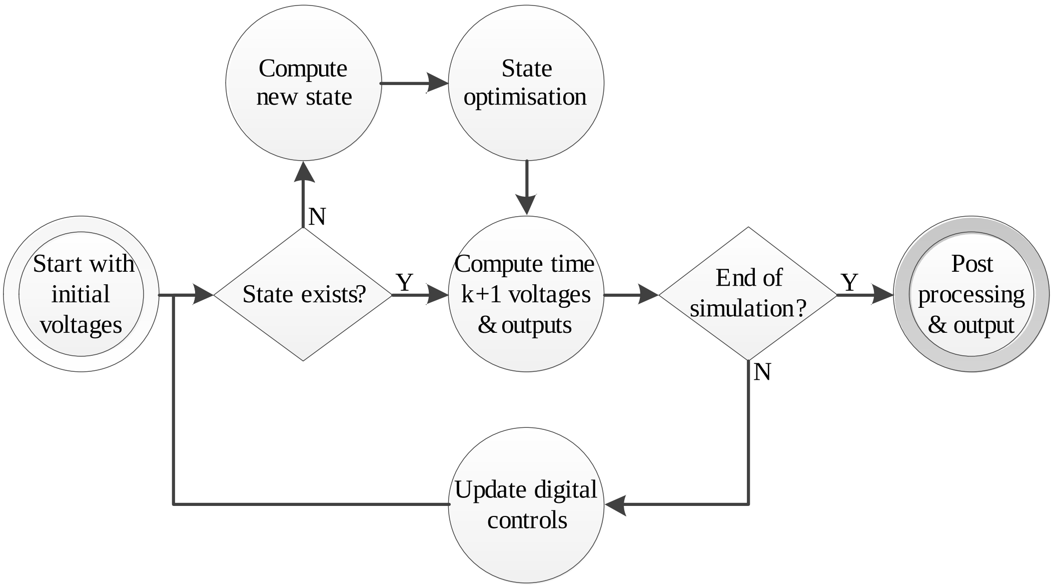

Figure 6.4: Simulation flow of the linear simulator.

The classes for the simulation of linear circuits are:

- **LinearCircuit**: Abstract class for the definition of a specific architecture. It holds the instantiated components. Several functions are available to simplify the creation of the states ant to add sub-circuits.

- **State**: When a state is created, the matrices are created from the configuration of the switches. For each state, the matrix is optimized (section 5.2.1) and stored in the memory. At each iteration, the current state is called to compute the new voltages of the circuit.
- **Node**: Node of the circuit (analog). It is characterized with a voltage and is connected to components (instances). Each line of a matrix of a state is computed from a node.
- **Node_dig**: Digital node. In addition to the Node properties, the digital node has a logic state and is driven by two digital switches. The first one being connected to the ground, the second one to the digital power supply.
- **Source_Idl**: Instance of an ideal voltage source, with two pins.
- **Ampli_Idl**: Instance of a ideal differential amplifier, five ports with the common-mode.
- **Cap_Idl**: Ideal capacitor, characterized with a capacitance and a charge. During initialization, a statistical distribution is applied to model component mismatch.
- **Switch_Idl**, **Switch_Idl_ana** and **Switch_Idl_dig**: Classes for the transmission gates (analog switches) and digital signal drivers (digital switches). The analog switches have three ports (in-out-gate) while the digital one only have two ports (in-out) and a logic state.
- **Instance**: Generic class for the capacitors, sources, amplifiers and switches. The common attributes (list of ports, information of the type of instance) are stored here.
- **Inv**, **Nand**, **Nor**, **Anr** and **Gate**: Definition of logic gates, there is one class for each sort of gate. Each gate has a list of digital input signals and one output, these variables are stored in the parent class, **Gate**.
- **Subckt**: Structure used to define sub-circuits, in order to reuse some blocks from an architecture to another. A sub-circuit may contain instances (capacitors, switches...) logic gates and sub-circuits. Local analog and digital nodes are also definable.
- **Object**: Parent class of all named classes. Finds an instance in a list with its name.
- **Capa**: Interface class used for the import of parasitic capacitors from a spice netlist. This class is not mandatory in the operation of the linear simulator, but imports capacitors from a text file containing the extracted layout.

6.4 Applications

6.4.1 Testbenches

Several testbenches were defined to provide a test environment for the simulator, all testbenches were coded with Matlab. The aim of the benches is usually to vary a single parameter and to ask the C++ simulator to compute the transfer function, the INL or the FFT of the ADC with a specific set of parameter. The simulator is called for each iteration of the bench parameter(s).

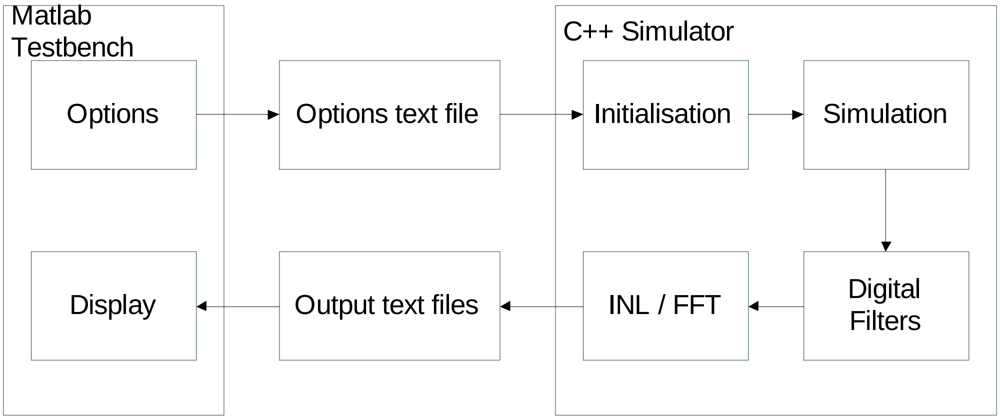

Figure 6.5: Block view of the communication with the simulator.

The Matlab testbenches communicate with the simulator through text files, Fig. 6.5. For each iteration in the testbench, an option file is created. This file is then read by the simulator and the output of the ADC is computed along with a few high-level parameters (ENOB, gain error, offset, std...) and stored is a simulation database where each file is solely defined by the option used. Once all the conversions are performed, the testbench displays the result (usually the ENOB) according to the bench parameter.

Remark: The ENOB is usually defined with the SNR or SNDR. This work is however focused on ultra-high resolution converters for instrumentation. Static characteristics and in particular the linearity of the converters are the key targets. In this chapter, another definition of the ENOB is used, based on the static transfer function of the ADC, Fig. 6.6.
A converter is reaching a given linearity if its output is located within the $\pm 0.5LSB$ range. The static definition of the ENOB based on the linearity of the converter is thus based on the maximal error in the INL:

$$ENOB = log_2\left(\frac{V_{fullscale}}{V_{error,max}}\right) \tag{6.1}$$

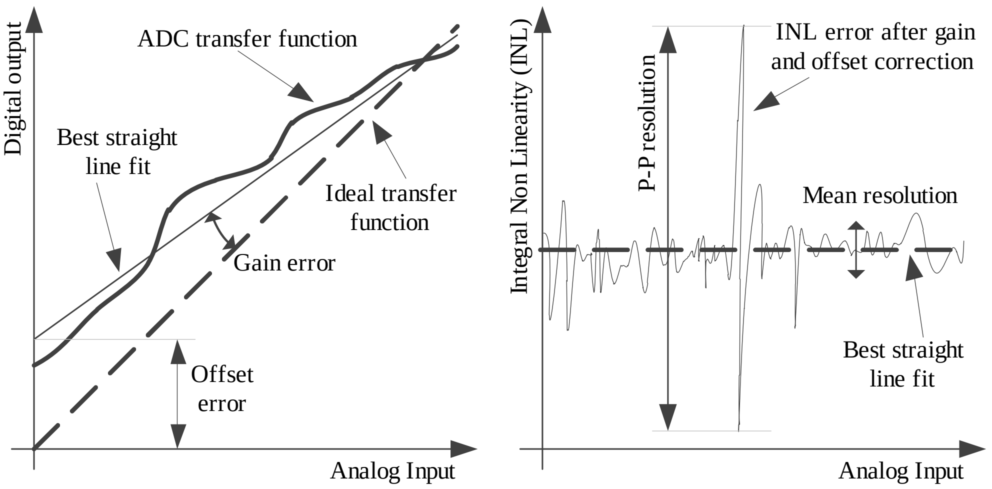

Figure 6.6: Definition of gain error, offset error, INL error from the transfer function (left) and Peak-Peak and RMS resolution (right).

The full-scale is proportional to the reference voltage. A value for $V_{fullscale} = 2V_{ref}$ is firstly supposed before evaluating the architectures. The maximal error is extracted from the INL and is named 'Peak-Peak resolution' in Fig. 6.6.

For the simulations including noise, the ENOB is based on the standard deviation of the INL, named 'Mean resolution' in Fig. 6.6. The computation is similar to the standard based on the SNR. The noise level is based on the mean noise power of the INL on the full input range.

6.4.2 Mismatch Analysis

The first application example of the linear simulator is the evaluation of the resolution of an ADC withcapacitive mismatch. Component mismatch is one of the main limitations of the resolution of ADC. Successive approximation and flash converters are very sensitive, while dual/multi slopes or ΣΔ structures are less as they use a reduced number of components. In a switched-capacitors ΣΔ circuit, the main mismatch sources are the capacitors. Depending on the capacitor dimensions, the standard deviation of the statistical distribution is given, for two identical capacitors by: $\sigma_{cap} = \sigma_0 / \sqrt{WL}$, where σ_0, expressed in $[\% \cdot \mu m]$, is a technology dependant mismatch factor.

The simulation flow of the testbench and of the simulator to perform a statistical analysis on a full system, is shown in Fig. 6.7. The core of this tool is the aforementioned simulator. It includes the following steps:

- User specifications of the mismatch coefficient and of the ideal circuit
- Generation, for each run, of a circuit with a statistical distribution for the capac-

itors

- Simulation and outputs processing. In an ADC, the outputs can be the transfer function, resolution (ENOB), INL, DNL, gain error...
- Histograms generation and computation of dispersion values (mean and variance) of each characteristic of the circuit

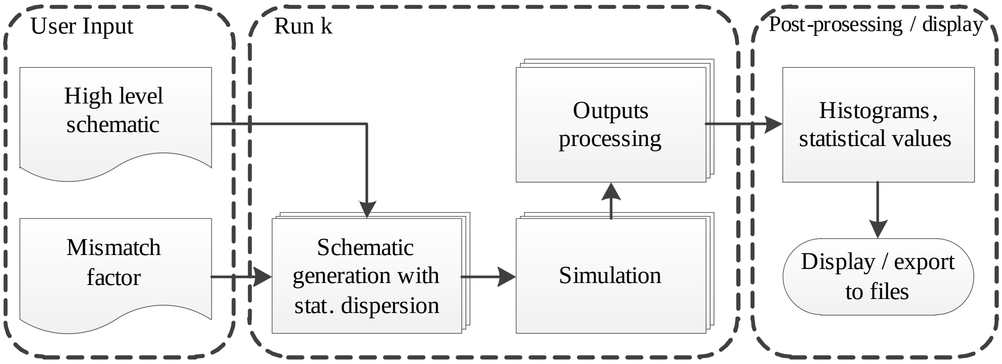
User Input
High level schematic
Mismatch factor
Run k
Schematic generation with stat. dispersion
Simulation
Outputs processing
Post-prosessing / display
Histograms, statistical values
Display / export to files

Figure 6.7: Monte-Carlo simulation flow.

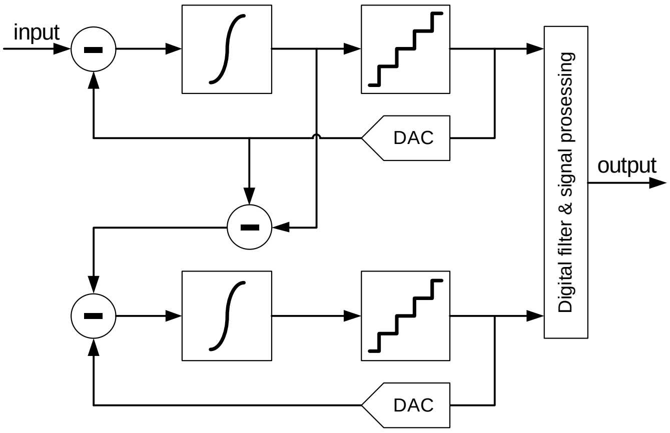

Figure 6.8: Schematic of a MASH 1-1 ΣΔ modulator.

For example, a statistical analysis is performed on an incremental second-order cascaded ΣΔ, Fig. 6.8. This fast simulator is suitable for high-resolution ΣΔ ADCs as a large number of clock cycles is necessary, especially to run Monte-Carlo simulations. The following modulator characteristics are extracted after each run: the gain and offset errors are given by the transfer function, Fig. 6.6 left; mean (RMS) and Peak-to-Peak (P-P) resolutions (ENOB) are extracted from the INL, which is the error after gain and offset correction, Fig. 6.6 right.

Figure 6.9 shows simulated results for the MASH11 converter with a mismatch factor $\sigma_{cap} = 2\%$. The asymmetry in the RMS resolution is due to the 20 bits quantization noise limitation.

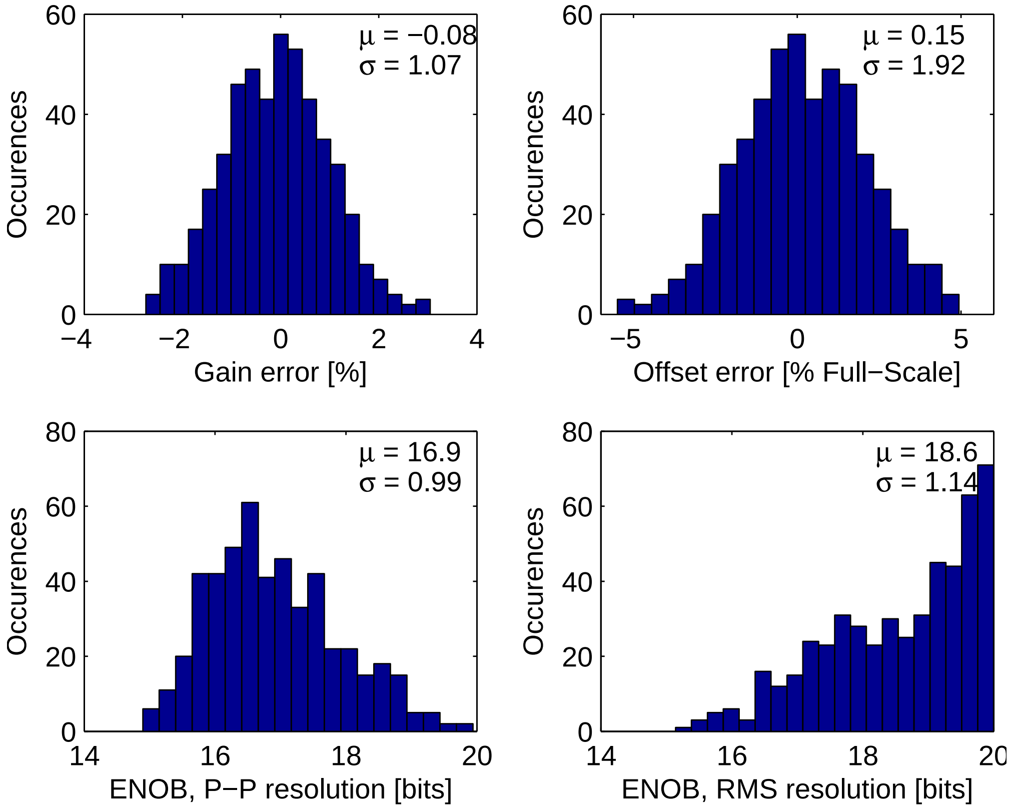

Figure 6.9: Statistical distributions of gain and offset error, P-P and RMS resolution for a cascaded ΣΔ ADC. Set of 500 samples.

6.4.3 Parasitic Coupling

A classic design scheme, Fig. 6.10, starts with a high-level topology to size the blocks and then draws the circuit layout. Once all these steps are finished and the blocks are validated, a last verification of the full system using an extracted netlist is performed, as long as the size of the circuit allows it. As the layouts of the sub-circuits are tested individually, the basic use of such a simulation is to test interconnections between blocks. The extracted netlist provides information on line resistances and on parasitic coupling between the circuit nodes.

As these simulations are extremely slow, sometimes lasting for several days, any problem caused by a parasitic coupling triggers a significant delay in the circuit design, or even worse, if the error is not seen or identified, a problem in the integrated circuit.

This section introduces a CAD tool to: I) automatically identify the sensitive parasitic coupling from a high-level schematic, thereby providing advice for the layout; II) identify in a fast and efficient way a parasitic coupling-related problem from the

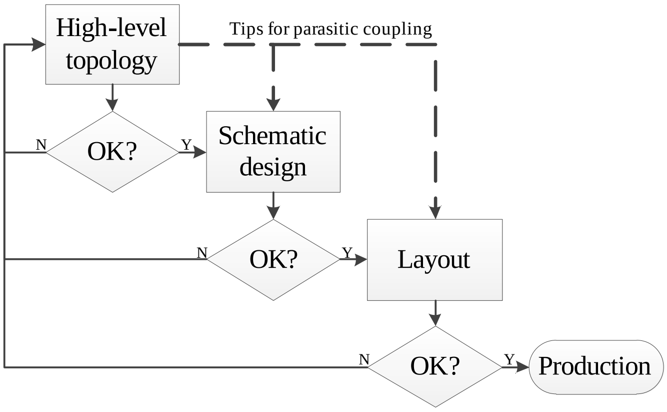

Figure 6.10: Top-down design flow and contribution of the tool for parasitic coupling.

layout extracted netlist.

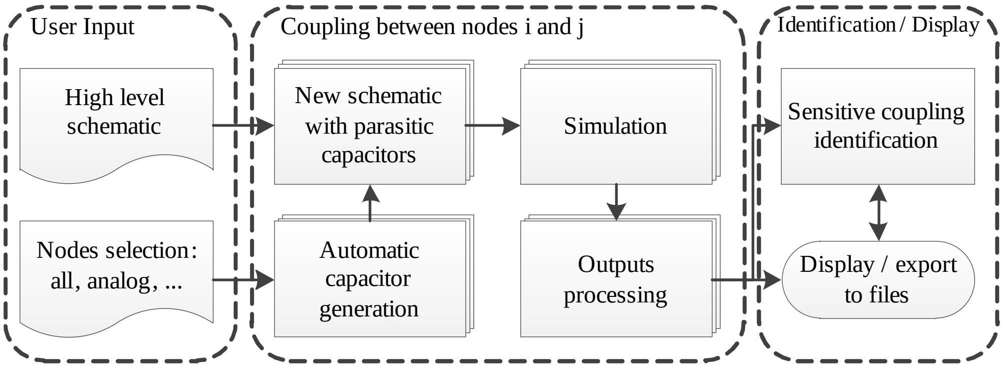
User Input
High level schematic
Nodes selection: all, analog, ...
Coupling between nodes i and j
New schematic with parasitic capacitors
Automatic capacitor generation
Simulation
Outputs processing
Identification / Display
Sensitive coupling identification
Display / export to files

Figure 6.11: Parasitic coupling simulation flow.

<ins>Automatic identification of sensitive nodes</ins>: the identification procedure, Fig. 6.11, takes a high-level schematic as a parameter, and then simulates the circuit for each possible coupling between two nodes. A classification is then performed to determine the most sensitive couplings. As the number of simulations is increasing in $O(n^2)$, the user can specify a node subset to be analysed: i) all nodes, ii) excluding coupling between digital signals (controls), iii) exclusively analog or iv) a custom set.

<ins>Efficient layout coupling analysis</ins>: An interface is provided to import the parasitic capacitors from a layout-extracted Spice file. As in the automatic coupling identification, a new circuit including these capacitors is generated. A set of sorting functions is available, allowing the exclusion or inclusion of some nodes. Compared to a classic simulation, this set of features, combined with the speed of the simulator, provides a fast and efficient debugging tool. Fig. 6.12 shows an example of parasitic coupling restricting the resolution of a MASH11 incremental converter (Fig. 6.8).
The results obtained with the implemented simulator are compared to a measure-

ment and to a classic simulator. The simulation setup included 50 nodes and over 500 capacitors. It required 2s and 1MB RAM to perform the 2^{21} clock cycles on a single 3GHz CPU core, including initialization and post-processing. The transistor level simulator with macro blocks took several hours for 2^{18} clock cycles and would take more than one month using transistors (1h/2^8 clock cycles).

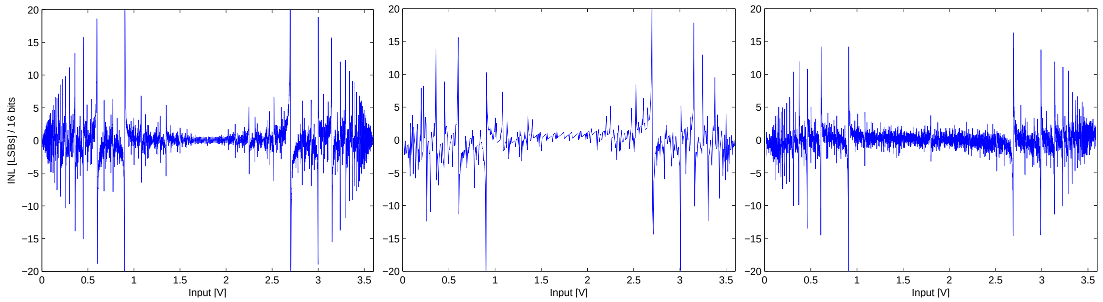

Figure 6.12: INL error of a ΣΔ ADC with a parasitic coupling problem. Left: using proposed tool. Middle: using a transistor level simulator with macro blocks for amplifiers, comparators, etc. Right: measured INL. The asymmetry in the middle graph is due to a lower number of points.

6.4.4 Comparison of Architectures

The use of a fast software allows to readily compare various architectures. Such a comparison, in addition to provide utilization examples of the simulator, is pointing the weaknesses of each topology. The final goal is to combine the advantages of the architectures to design a new one, in the next chapter.

The architectures to compare were selected according to their differences. While the order of the four topologies is comparable (2^{nd} and 3^{rd} order), the connections of the input and between the stages are dissimilar. The phases and digital sequences are also different. Two published academic architectures and two commercial ones are compared. The architectures are:

- **2^{nd} Order**: Architecture from the industry with a single second-order loop and distinct capacitor for the signal and for the reference.
- **2^{nd} Order Boser**: This published architecture [57] is also a single second-order structure. Its particularity is the sharing of the input capacitors (in both integrators) between the input signal and the reference. This architecture is named 2ndOrderBos in the figures.
- **MASH 1-1**: Second-order cascaded commercial architecture. This architecture is named MASH2 in the code and in the figures.
- **MASH 2-1**: Third-order published architecture [58] with a first loop including

two integrators and a first-order second stage. The 2nd stage has three input: for the signal from the first stage and for DACs (reference) of both first and second stages.

Remark: Reset switches were added in all architectures to simulate in sigma-delta as well as in incremental mode.

OSR

The first criterion of comparison between the architectures is the quantization noise and the OSR. In this simulation, the four architectures are compared with the linear simulator and the integrator-based simulator. The results are displayed in Fig. 6.13. As expected, the results are identical with both simulation approaches as the modulator is not including any imperfection.

In the legend of the figure, the plots generated with the linear circuit simulator are referenced with the prefix LC.

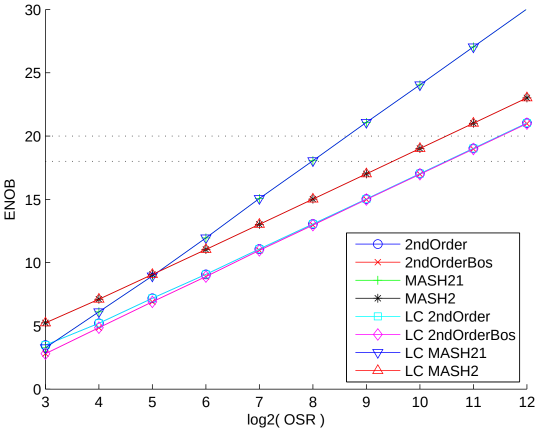

Figure 6.13: Quantization noise limitation of the four selected architectures.

DC Gain

Simulation results with finite DC gain of the first amplifier are plotted in Fig. 6.14. In all architectures a finite DC gain has an impact on the gain error (Fig. 6.14-right), while the linearity of the converter suffers from an insufficient DC gain only in cascaded architectures. The plot 6.15 shows that the degradation of the INL in the MASH 1-1 architecture is the most critical in the mid-scale region. The DNL of the MASH 1-1 is thus heavily impacted by a poor DC gain as the peaks in the INL are here the consequence of flat area in the transfer function.

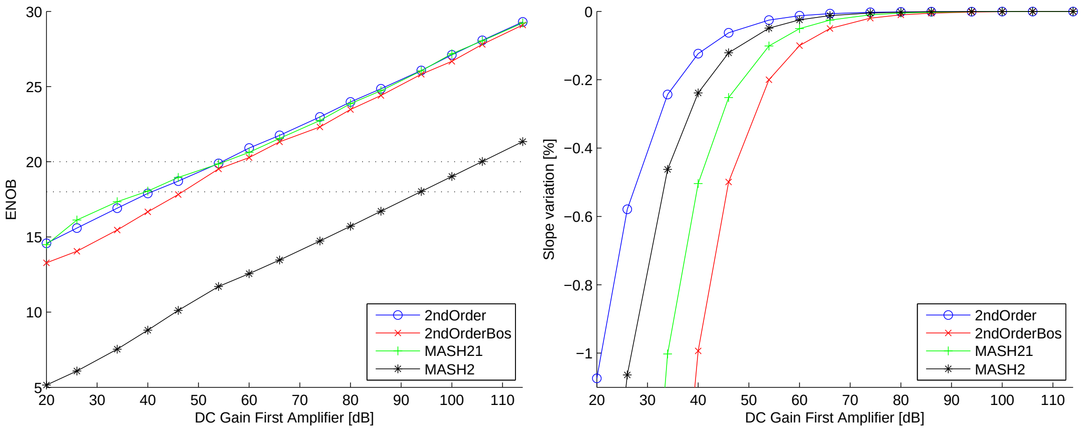

Figure 6.14: Evolution of the resolution with a finite DC gain in the first amplifier.

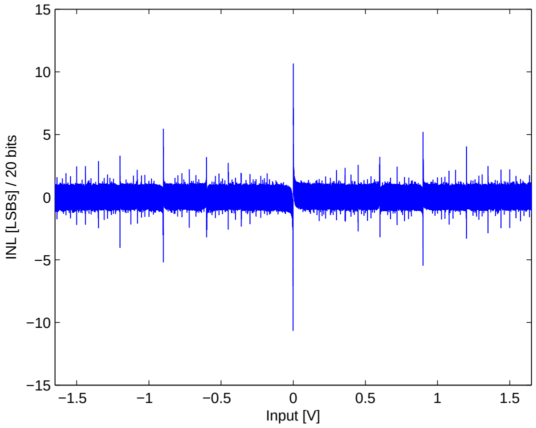

Figure 6.15: Simulated INL of the MASH 1-1 architecture with a finite DC gain in the first amplifier, 80dB.

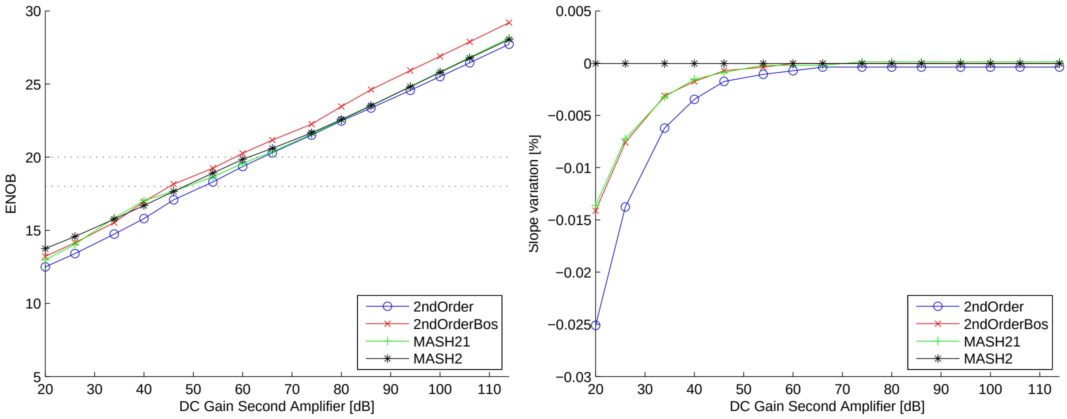

Figure 6.16: Evolution of the resolution with a finite DC gain in the second amplifier.

The second amplifier is less critical. While the linearity is almost the same than in the first amplifier, Fig. 6.16-left, the gain error is almost not affected, Fig. 6.16-right.

If the linearity is the main target and if the gain error of the ADC is compensated by calibration, it appears that a DC gain lower than 60 dB may be sufficient to reach a resolution of 20 bits in single loop architectures. One should however be aware that any variation of the gain of the amplifier may degrade the resolution.

Noise

The thermal noise is mainly related to the size of the input capacitors ($V \cong \sqrt{kT/C}$ for a simple input). It is possible to average this noise increasing the number of clock cycles, either with a higher OSR or by averaging several samples. The measurement of the resolution of the ΣΔ ADC according the integrated capacitor $OSR \cdot C$ determines the limit between the quantization noise and the thermal noise. Moreover, if the power consumption is directly proportional to the input capacitors, a measurement with a constant $OSR \cdot C$ is like a measurement with a constant energy.

The evolution of both quantization noise and thermal noise with the architecture is shown in Fig. 6.17. The $OSR \cdot C$ product is constant for each curve (i.e. the capacitors in a curve for and OSR equal to 2^{10} are 4 times smaller than the capacitors in the same curve with an OSR equal to 2^{8}).
Two curves are displayed in Fig. 6.17. In the right plot, the capacitors are 64 times larger than in the left one. The dimensions of the input capacitors of the first stage are identical in all the architectures.

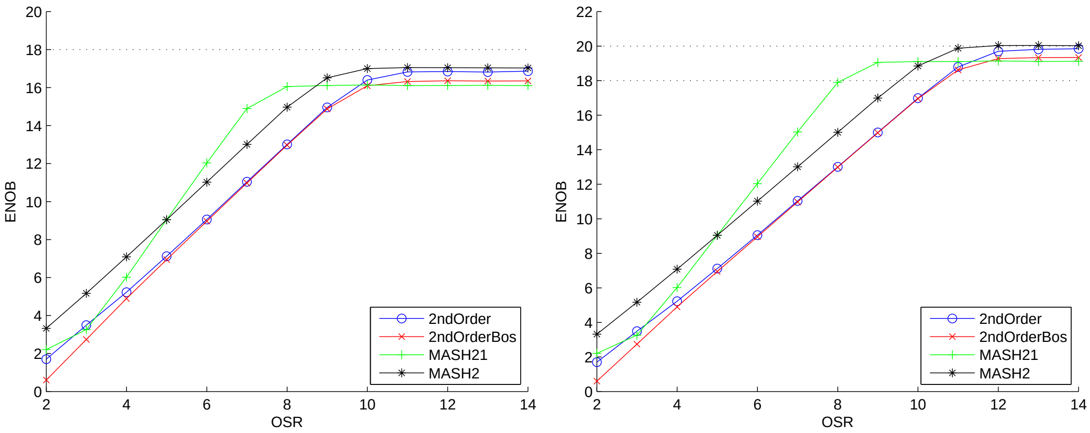

Figure 6.17: Evolution of the resolution with a constant integrated capacitor ($C_{in}OSR$ product). The right plot shows a thermal noise 8 times lower as the capacitors are 64 times larger.

Settling time

The settling of the integrators is related to the oversampling frequency, the size of the capacitors and the biasing current of the amplifiers.

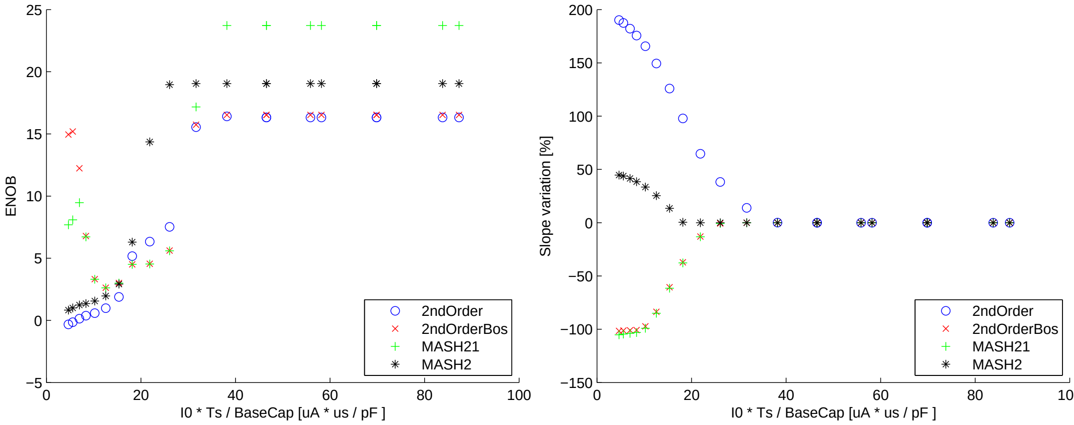

Figure 6.18: Impact of the settling time on the linearity and gain error.

The settling conditions are thus displayed, Fig. 6.18, according to a voltage, $V = t_s I/C$.

Non-linear Capacitors

Simulations were driven with a variation of the non-linear coefficients of the capacitors. The non-linearity of the capacitors induces smooth non-linear effect in the INL, without any consequences on the DNL. An example is shown in Fig. 6.19. The impact of the first-order parameter α is displayed, for the four architectures, in Fig. 6.20-left while the second-order coefficient β is shown in Fig. 6.20-right. In both plots, one of the input was constant (set to V_{cm}), while a sweep was applied on the second one. The results are thus sensitive as well to the variations of the input common-mode.

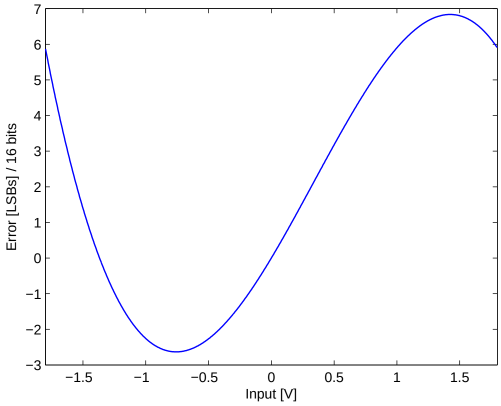

Figure 6.19: Simulation example with 1^{st} and 2^{nd} order nonlinearity

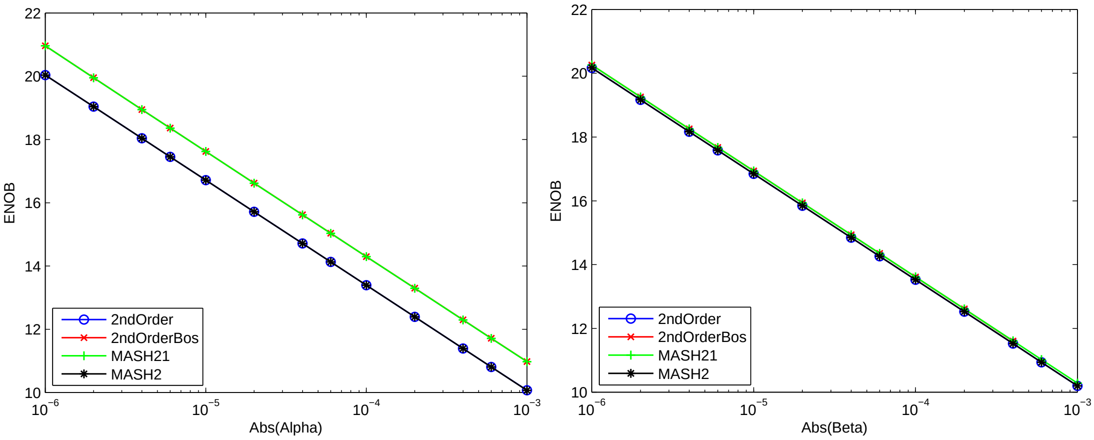

Figure 6.20: Variation of the first-order (left) and of the second-order (right) voltage coefficient of the non-linear capacitors.

Parasitic Capacitors

A comparison of the sensibility of the four topologies to the parasitic capacitors was conducted, adding a 1pF capacitor between each pair of nodes. The testbench follows the simulation flow defined in the section 6.4.3, Fig. 6.11. As an example, observed effects are displayed in Fig. 6.21.

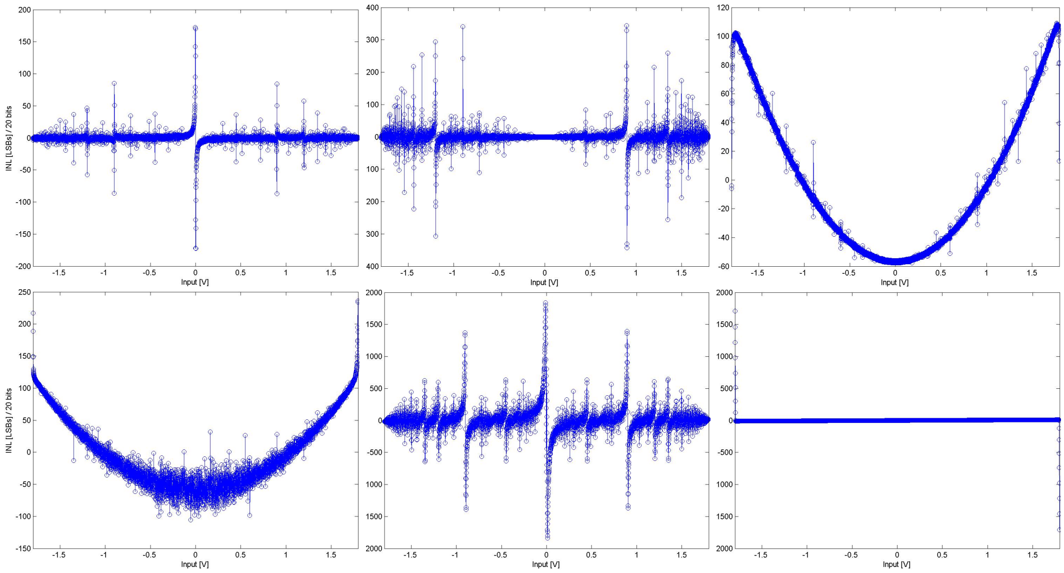

Figure 6.21: Distortions of the INL due to parasitic capacitors.

These examples show a large variety of impacts on the INL. The extra capacitor may have a strong impact on the DNL, visible with spikes in the INL or induce a smooth variation of the INL or even create a gain error as in the bottom-right plot. The unrealistic scale, up to several kLSB/20b is due to the artificially large parasitic capacitor (1pF) added in the bench. Such a large value was used in order not to miss a sensitive coupling, shadowed by the quantization noise.

In order to allow a comparison, in all architectures the input capacitors have the same value. Depending on the simulated resolution the coupling is reported to be sensitive or not (threshold sort). The number reported in Table. 6.2 is given without any correlation with the number of nodes in the circuit. It is important to notice that, most of the time, it is possible to avoid the parasitic couplings with a proper layout. Nevertheless, the effort to check and fix the layout is more important if the amount of sensitive coupling is larger.

Summary

The comparison of the four architectures is summarizes in Table 6.2. The required performances are in each case reported for resolutions of 18 and 20 bits.

Table 6.2: Comparison of four architectures.

Topology Input S-R[1]	2nd Order #4-#4		2nd Order Bos. #1		MASH11 #6-#4		MASH21 #2-#2	
	18bits	20bits	18bits	20bits	18bits	20bits	18bits	20bits
log2(OSR)	11	12	11	12	10	11	8	9
C_{in}OSR [pF]	83	1330	168	2690	65	1050	222	3550
$I_0 T_s / C$ $[\mu A \mu s / pF]$	46		37		32		37	
Sensitive coupling	9		3		68		29	
DC Gain 1 $[dB]$	41	53	45	57	94	106	40	56
DC Gain 2 $[dB]$	52	64	45	58	48	60	49	63
Capa, β	4E-6	1E-6	4E-6	1E-6	4E-6	1E-6	4E-6	1E-6
Capa, α	4E-6	1E-6	2E-6	8E-6	4E-6	1E-6	2E-6	8E-6

[1] Refer to the input configurations defined in chapter 4. The first one is for the signal input and the second one is for the reference input.

This comparison shows that the MASH 1-1 architecture is the most efficient one in terms of energy per conversion as it has the lowest $C_{in}OSR$ value and is the fastest

($I_0 T_s / C$ value). This performance is reached as it implements an input configuration with crossed inputs.
This architecture is however very sensitive to the matching between the analog implementation and the digital filter. The effect is mainly visible for capacitive mismatch and for limited DC gain. The next drawbacks of this architecture are a higher sensitivity to the common-mode, visible in the first-order voltage coefficient of the capacitors, α and the worst sensitivity to parasitic capacitors. Both effects are related to the type #6-#4 implementation of the first stage.

The second most efficient (Energy/Conversion) topology is the single-loop second-order one '2nd Order'. Its type #4 inputs provide a fair thermal noise level with a well balanced load on the first amplifier in each phase. As the input #6 of the MASH 1-1, it is also sensitive to variations of the common-mode.

The middling power efficiency of the last second-order architecture, '2nd Order Boser' is mainly due to the implementation #1. While the noise performance is comparable to the one obtained with the previous implementations, the current consumption of the first amplifier is higher. The cause is the poor load balancing on this input type in a single-bit configuration.

The least efficient topology is the MASH 2-1. The main reasons are the type #2 implementation, that doubles the noise power, and the use of a higher order architecture. In a third-order architecture, the thermal noise power is approximately 1.4 times larger than in a second-order structure and 1.8 times larger than in a first-order topology.

6.4.5 Conclusion

The first part of this chapter described the implementation of the models previously developed with a high-level object oriented programming language, C++. Both simulation approaches - based on the integrator and on linear circuits - were coded and grouped into a single simulator.

Three main considerations were driving the coding effort:

- Separation of the core of the simulator and of the files describing the architecture. The massive use of objects and of inheritance allowed a minimization of the files to create in order to add a new architecture. A common environment for all architectures is defined into the parent classes. This separation simplifies a lot the creation of a new architecture and sets the bases of an external definition of the architecture: the topology could be created in an external graphical user interface or be imported through a text file.
- Common interface. Whatever the architecture or the simulation mode, the con-

figuration of the simulator is identical. The main benefit is the use of identical testbenches to compare readily various architectures and simulation modes. The testbenches were not integrated into the simulator. They were written with an executed language - Matlab - to be easily modified and to spare compilation time.

- Fast simulator. In addition to the mathematical optimizations defined in the previous chapter to reduce the complexity of the computations, special care was taken with the memory management to speed up the simulator.
  With this in mind, the post-processing (INL/FFT and computation of the high-level characteristics) is also computed into the compiled simulator.

The simulator was then tested and validated, comparing it with an existing circuit. Two tools, until now not available on this modeling level, were created. The first one runs statistical simulations to evaluate the matching of the capacitors into ΣΔ converters. The second one identifies, prior to the layout, the sensitive parasitic coupling and evaluates the quality of the layout, once this one drawn, with an import of an extracted netlist.

The last section of this chapter was dedicated to the comparative study of four architectures. For each non-ideal effect, the linearity and the gain error were evaluated. These simulations showed on one hand the scope of this simulator, and, on the other hand, evaluated the quality of each architecture. Modeling on the switched capacitor level showed - in addition to the differences related to the architectures - that the implementation contributes also a lot to the overall performance.

7 ADC

This chapter presents the design of the sigma-delta modulator. The architecture is first defined, the main components are then sized and the digital filters are finally detailed. The selection of the topology and the specifications of the blocks are guided by the comparison of architectures done in the previous chapter.

The ΣΔ modulator is part of a full acquisition chain, Fig. 7.1, including a front-end, digital controls and the digital filters required to decode the output streams of the modulator. The next chapter is dedicated to the design of the front-end while the last section of this chapter covers the design of the filters.

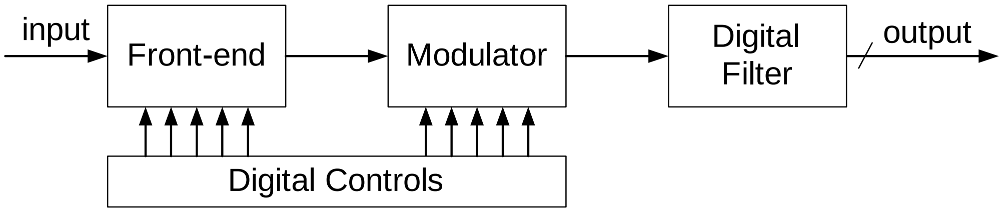

Figure 7.1: Acquisition chain.

7.1 Specifications

The process for this circuit is a standard CMOS 180nm - 1.8V. The additional options are the following:

- Process: TSMC018
- Transistors thick oxide: 3.3V P/NMOS
- Resistors: high-poly
- Six metal layers
- Capacitors: MIM plate capacitors on top metal layers
- Digital cells 3.3V: CSL6.1K library

The circuit is divided into two main parts, with two distinct supply voltage levels. The digital section is powered with a 1.8V supply while the analog blocks are 3.3V powered. The lower 1.8V supply reduces the size and the power consumption of the digital.

The modulator should operate with a power supply voltage V_{dda} between 1.8V and 3.6V and in an extended temperature range. When the power supply is low (<3.0V), an internal voltage source, V_{ddh}, is generated with a charge pump. This voltage source, sized to lower the on-resistance of analog transmission gates, cannot be connected to a static load as it does not deliver enough current.

Symbol	Description	Min	Max	Unit
Vdda	Analog power supply	1.8	3.6	V
Vddh	Internal power supply	3.0	3.6	V
Temp	Temperature	-40	125	˚C

Table 7.1: Operating conditions.

The specifications of the ADC are provided in table 7.2. The main ones are a 20 bits resolution at a sample rate of 64Sps and a current consumption of 120μA. The modulator should also meet a second set of requirements with a faster sample rate, 2kSps, but at a lower resolution of 16 bits.

Symbol	Description	Min	Typ.	Max	Unit
Res	ADC resolution	6		22	bit
Fs	Oversampling frequency	32	1000	1250	kHz
ENOB	Resolution @ 64 Sps	20			bit
	Resolution @ 2 kSps	16			
DNL	Differential Non Linearity			± 0.5	LSB
INL	Integral Non Linearity			± 15	LSB
Ipower	Current consumption		120		μA
Gain	Signal/Ref. amplification	1/8	1	16	-

Table 7.2: Block specifications.

The oversampling frequency, the resolution as well as the power consumption have to be programmable by the user. The programmability of the modulator is further detailed in the next sections.

7.2 Selection of the architecture

The selection of the topology is based on the comparative study of several architectures performed in the previous chapter.

7.2.1 Structure

One of the high-level decisions is the choice between a feedback and a feedforward structure. As this thesis is focused on switched capacitors implementations of the modulators, feedback and feedforward structures were not compared in the previous chapter analysis.

A feedback structure has been selected for the following reasons:

- The number of analog components is smaller. The modulator has no extra adder.
- High-level simulation with the designed tool estimated the required matching of the capacitors and validated the dimensions of the capacitors for the reference.
- The comparative study of the last chapter covered only feedback topologies. The design is thus safer for a better-known structure.

7.2.2 OSR - C_{in}

In a low-noise low-power design, the noise level of the modulator should be dominated by the thermal noise and not by the quantization noise, as the thermal noise power is inversely proportional to the power consumption.

The reduction of the quantization noise is usually done without increasing too much the power consumption, increasing the order of the modulator or using a quantizer with more levels.

The thermal noise of the modulator is mainly set by the product $C_{in}OSR$ of the first integrator. The choice of the OSR and of the size of the capacitors is determined with the available surface for the capacitive arrays of the first integrator. The size of the capacitors is a tradeoff between the surface of the first stage, the capacitor matching and the sensitivity to parasitic capacitors. Rough values may be estimated at this step, without knowing the architecture of the modulator. For a resolution of 20 bits, input capacitors of a few pF and an OSR between 2^{10} and 2^{13} are required.

7.2.3 Topology - Order

As the OSR is roughly known, the order of the converter can be estimated. A first-order converter is not suited as the quantization noise level would be too high compared the thermal noise. The second-order structure requires an OSR between 2^{10} and 2^{12} to reach a resolution of 20 bits, a third-order architecture is thus not necessary. In addition to the power consumption added by the extra stages, the thermal noise level of higher-order architecture is worse. This is a consequence of the non-uniform weighting of the samples (samples at the oversampling frequency).

The structures with one or several multi-level quantizers were also excluded in this design as the mismatch of the capacitors of the DAC for the reference strongly limits the resolution; the advantages of the multi-bit (lower quantization noise, lower stability and lower power consumption) do not compensate for the degradation of the linearity.

Optimal decoding was also not implemented. Indeed this design is dominated by the thermal noise. In such a situation, an optimal filter is not efficient as it firstly reduces only the quantization noise level and secondly would require a large computation power[1].

Second-order cascade modulators, with two first-order loops (MASH 1-1) do not require a very large OSR (2^{10}) to reach 20 bits and are unconditionally stable. On the other side, a good matching and a very large gain ($\gg$ 100db, DC) of the first amplifier are needed.

The selected topology is thus a second-order single-loop modulator with a single bit comparator. The minimal OSR for this architecture is around 2^{12} to reach the targeted resolution of 20 bits.

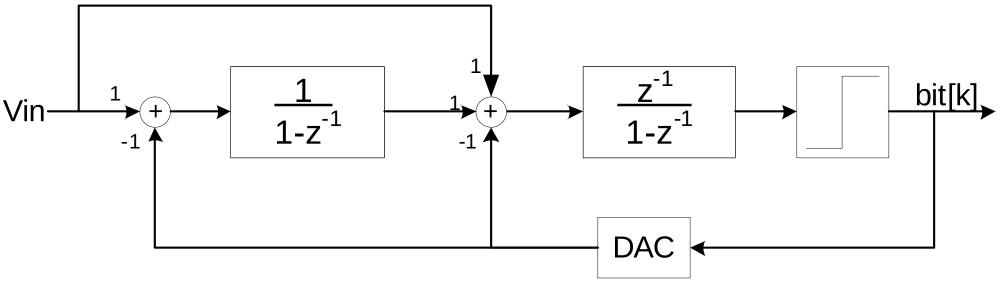

Figure 7.2: Second-order feedback stage.

This architecture is suited to reach the most cumbersome specification (i.e. 20 bits - 64Sps), but it cannot fulfill the 16 bits specification at 2kSps. The increase of the sample rate reduces the OSR and therefore increases the quantization noise.

A solution to overcome this limitation while keeping the 20 bits optimized architecture is to add an extra stage in a 3rd-order MASH 2-1 structure. Such a structure allows a significant reduction of the OSR while keeping a quantization noise much lower than 1 LSB. Moreover the stability of the modulator is guaranteed as the maximal order of each loop is equal or lower than 2.

---

[1] It was shown in the chapter dedicated to the optimal filtering that the mathematical complexity of an optimal filter for noisy signals was growing exponentially with the OSR.

The extra stage is enabled on-demand. In the 20 bits configuration, it is not connected to reduce the power consumption, lower the thermal noise level, simplify the structure and avoid potential matching problems between the stages.

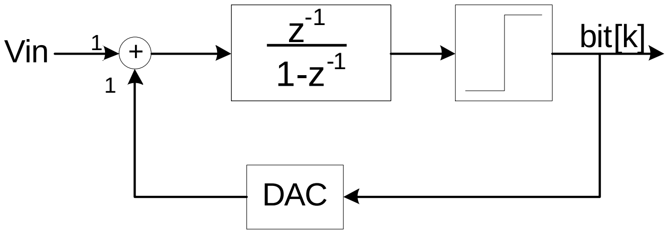

Figure 7.3: First-order stage.

A graphical representation of the resolution as a function of the OSR is provided in Fig. 7.4 for second- and third-order modulators. The resolution is given by: $ENOB = m\log_2(OSR) - k$, m being the order of the modulator and k a constant specific to the selected architecture.

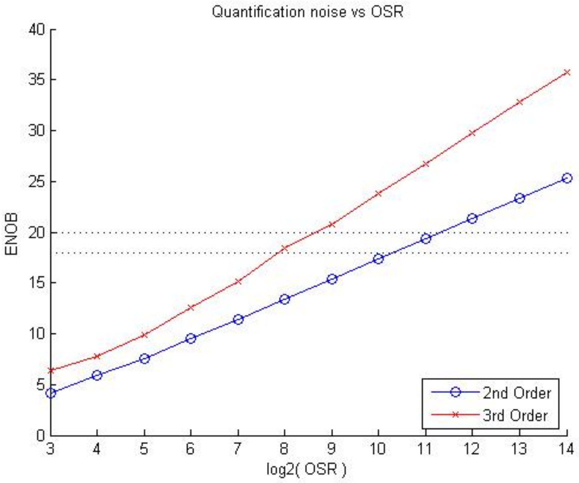

Figure 7.4: Evolution of the resolution with the OSR in noiseless ideal modulators.

7.3 Implementation

The implementation with switched capacitors was guided with the comparison of the input configuration in the chapter 4. The first integrator is the most critical as it dissipates most of the power and contributes to a major part of the noise budget. This integrator has two inputs, one for the signal and one for the reference from the DAC. The selection in the table 4.1 is thus direct as it is the same structure.

The topology #5, reported in Fig. 7.5 for the reader, has been selected as it has, with the differential structures #8-9-10, the best noise and power consumption performances. Despite the fact that the symmetrical structures are slightly better regarding the maximal load on the amplifier, theses structures were discarded to alleviate the requirement on the loop delay and on the comparator.

Although the noise and power consumption performances of the most spread structure #1 are better[2], it has not been selected as the gain of the integrator cannot be adjusted with programmable capacitors (i.e. to configure independently the gain of the signal and of the reference).

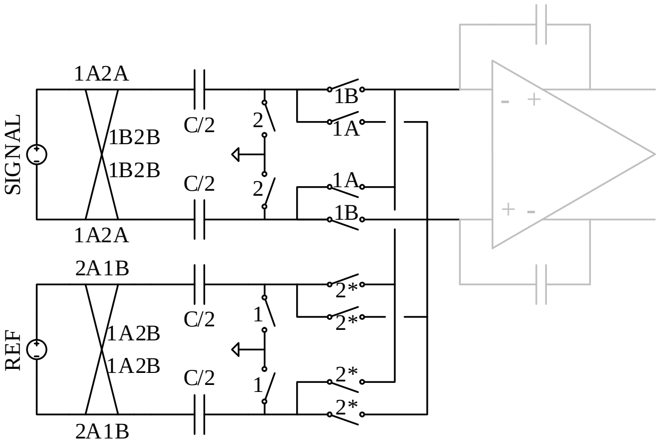

Figure 7.5: Switched capacitors input with switching scheme #5.

The drawbacks of this structure are: a slightly lower matching due to half-sized capacitors, a sampling of the signal and of the reference in both phases and potential problems of parasitic coupling, clock feedthrough and overvoltage. While the matching is acceptable[3] and the front-end is stable during both phases, special care must be taken for the three last items. Each limitation is detailed in the next sections and, if necessary, a workaround is provided.

7.3.1 Protection for the reference input

As the reference of the modulator is connected to a maximal voltage equal to the power supply, an overvoltage limitation has to be added on this input. If the topologies #4 or #5 are directly implemented with such a high reference voltage, the nodes on the high-impedance electrodes of the input capacitors[4] may reach a voltage beyond the power supply after the switching of the input chopper. This would induce a loss of

---

[2] Only in multi-bits topologies.

[3] Evaluation based on high-level simulation with the designed tool, capacitors of a few pF and a mismatch factor extracted from the process model files.

[4] Nodes named ’bhup’ and ’bhdo’ in Fig. 7.6.

charges through the substrate and/or, if the voltage is too high, even damage the transistors of the switches.

I) Extra capacitors to ground

As shown in Fig. 7.6-left, a first solution would be to add extra capacitors to ground of the same value than the input capacitors.

After the switching of the chopper, in the beginning of the transient, the charges are equally distributed between the input capacitors and the ones to ground. The high-impedance nodes of the capacitors are thus never exceeding the power supply.

The main drawback of this implementation is the contribution to the noise level of the extra capacitor. With identical capacitors, the noise power is doubled. This solution, easy to implement, is used in the 2^{nd} and 3^{rd} integrator as they do not contribute significantly to the noise level.

II) Serial resistors

In this second alternative, a serial resistor is added in order to limit the strong current to the input capacitor. As the transient is slowed down, time is left for the charges to be transferred to the integration capacitors (not shown in Fig. 7.6-right).

This solution is implemented in the first integrator. The resistances are bypassed for the end of the charge transfers, so they do not add any extra noise. The control of the bypass switches is generated in the digital domain. In order to guarantee the conservation of the charges during the transient, small extra capacitors (C/4) to ground are added.

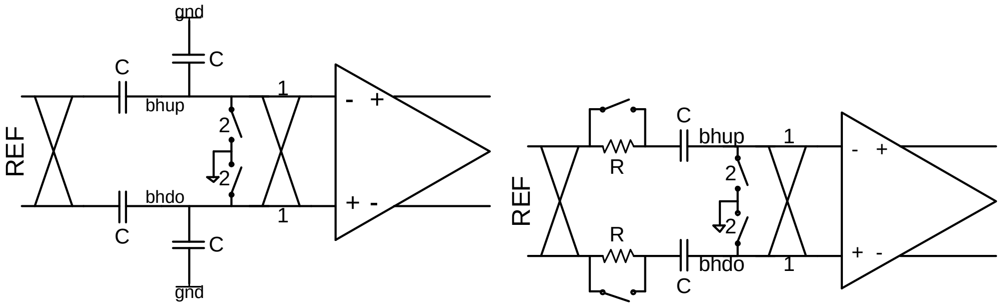

Figure 7.6: Left: Voltage limitation implemented with capacitors. Right: implementation with serial resistors.

The value of the serial resistor (a few kΩ) was determined with transistor-level simulation as:

- A resistance too small is not attenuating enough the transient voltage

- A resistance too large slows down the charge transfer and only delays the overvoltage after the opening of the bypass switch.

7.3.2 Clock feedthrough

When the switches connected between the input capacitors and the amplifiers are opened, the charges of the gate-source and gate-drain capacitors, as well as charges in the MOSFET channel, are released into the input capacitor and into the integration capacitor, Fig. 7.7.

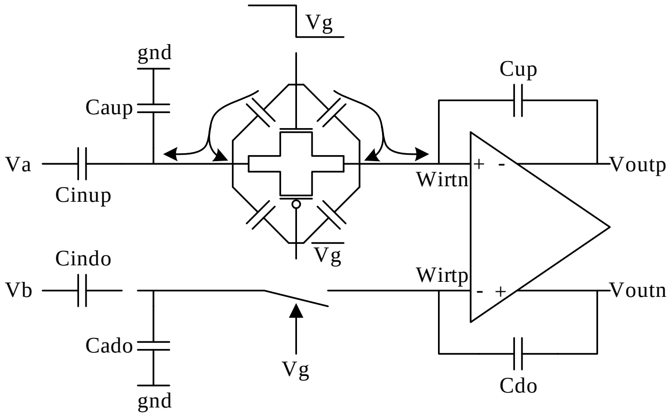

Figure 7.7: Distribution of the charges of the switch into the input and the integration capacitances.

In a differential structure, this injection of charges is problematic only if the distribution of the charges is different in the upper branch and in the lower branch. As the distribution is related to the source impedance, the error on the integrator varies with the amplitude of the input signal. The final consequence of the clock feedthrough is a non-linearity in the transfer function of the converter. Moreover, the non-linearity varies with the common-mode of the input signal.

There is also a clock feedthrough on the reference branch, but it does not impact the INL of the converter as the voltage reference is constant[5].

Several actions were taken to limit the clock feedthrough:

- The switches between the input capacitors and the amplifier are small and with symmetrical dimensions. Transmission gates with identical size PMOS and NMOS (i.e. with roughly the same gate capacitance) compensate partially the charges of the PMOS by the ones of the NMOS. The size of the switches is small

---

[5] However, it may introduce gain drift, as device characteristics vary with temperature and/or aging.

to minimize the amount of injected charges. The size is a tradeoff between the clock feedthrough and the settling time of the normal charge transfer.

- Minimization of the variation of the on-resistance switches on the input side. Transmission gates with larges PMOS than NMOS are used to have a constant conductance in the largest portion of the input range.
- Constant input common-mode and centered to the middle of the power supply voltage. The aim is to minimize the variation of the source impedances in the differential mode. As the sensor may not deliver an output with a constant common-mode, a feature is implemented in the front-end to control it, see next chapter.

These measures are not sufficient to guarantee an INL lower than 1 LSB / 20 bits, especially if the power supply voltage is minimal. They are nevertheless sufficient to fulfill the design specification (see table 7.2).

7.3.3 Parasitic coupling

Remark: The verification and modifications described in this section do not follow the chronological ADC design order. Some steps require the full design of the modulator with the layout. The analysis of the parasitic coupling is however done hereafter, as it is a consequence of the selection of the topology #5 for the input of the first integrator.

The architecture #5 is sensitive to the parasitic coupling with some control signal of the switches. The use of alternate phases (i.e. phases A and phases B) is a potential source of problems, if there is a slight asymmetry between the odd and even cycles. The largest degradation occurs for alternate bitstreams, typically 01010101...

Most sensitive parasitic coupling may be averted with a proper layout. The layout was guided with the two main following actions:

- Identification of the sensitive coupling: before the drawing of the layout, an exhaustive study of the capacitive coupling was carried out. The identification was done using the dedicated linear simulator, designed in the previous chapter.
- Simulations including the capacitance extracted from the layout: Once the layout drawn, simulations were run, importing the extracted view of the modulator in the dedicated simulator. This iterative procedure allowed a fast identification of the residual problems and a correction within a few iterations.

The sensitive parasitic couplings were all easily eliminated, except one. There is, by construction, parasitic capacitors between the control of the switches named '2*' in Fig. 7.5 and the nodes between these switches and the input capacitors. Simulations showed that a mismatch of the gate-source of these switches of 10aF degrades the INL of a few LSB / 20 bits. Special care was thus taken to draw the four switches named ’2*’ and to draw the routing of their command signals.

It is difficult to predict the mismatch of parasitic capacitances, especially when the mismatch is related to the process of the circuit. A feature has been implemented to toggle between topologies #5 and #4, the last one being less sensitive to parasitic coupling. A type #5 modulator is perfectly compatible with the type #4 as the analog components are the same. Only the control signals of the switches are different. In order to have more degrees of freedom, the branches for the signal and for the reference can be connected independently in type #4 or type #5.

Post-measurement observation: Practical results on the fabricated chip (chapter 9) show a degradation of the INL of ± 3 LSB / 20 bits with the topology #5. The perturbation can be avoided using a hybrid configuration with type #5 for the signal and type #4 for the reference.

A possible solution to avoid this particular parasitic coupling while keeping the lower dynamic power consumption of the structure #5 would be to modify this structure adding extra switches, Fig. 7.8.

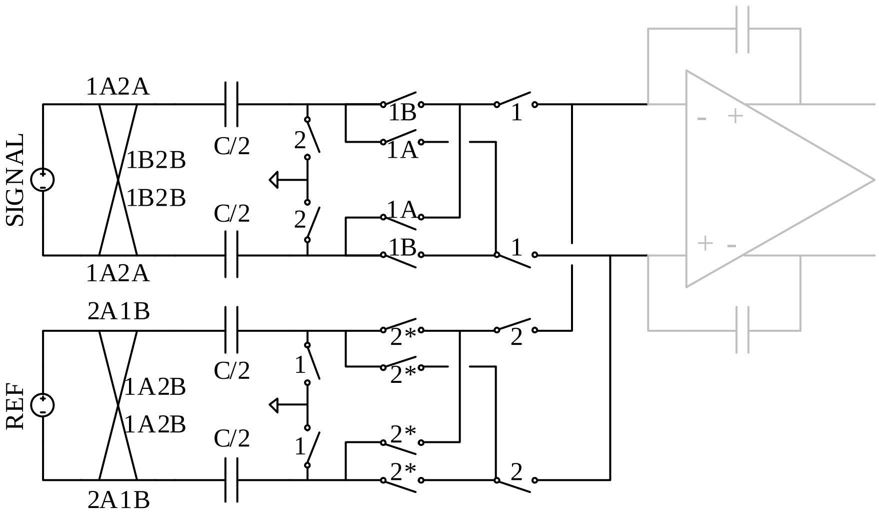

Figure 7.8: Modification of the structure #5 to avoid critical capacitive parasitic coupling.

In order to ensure that the parasitic capacitances from the control signals of the switches ’2*’ do not add undesired charges into the integrator and/or into the input capacitors, the switches ’2’ should be opened before the switches ’2*’ at the end of phase 2. At the end of phase 1, the switches ’2*’ should also be closed before the switches ’1’ are opened, in order to discharge the parasitic charges.

7.3.4 2$^{\text{nd}}$ and 3$^{\text{rd}}$ integrators

The topology used for the second integrator is similar to the one used for the first one (i.e. #5 for the reference and the input signal). As the output of the first integrator is not symmetrical during phases 1 and 2, the switching of the capacitors between the two stages is of the second type (standard alternate connection to the signal and to a constant potential, see Fig. 7.9).

The low-impedance electrode of the coupling capacitors between the two first integrators may be connected during either phase 1 or phase 2 without altering the loop delay. Both options have advantages and drawbacks; a configuration bit is thus reserved to toggle between both switching schemes:

- If the direct connection is enabled, the distribution of the load on the first amplifier during the two phases is better.
- If the alternate connection is selected, the two amplifiers are never connected together. The slewing/setting of the second amplifier is thus not dependent on the one of the first OTA.

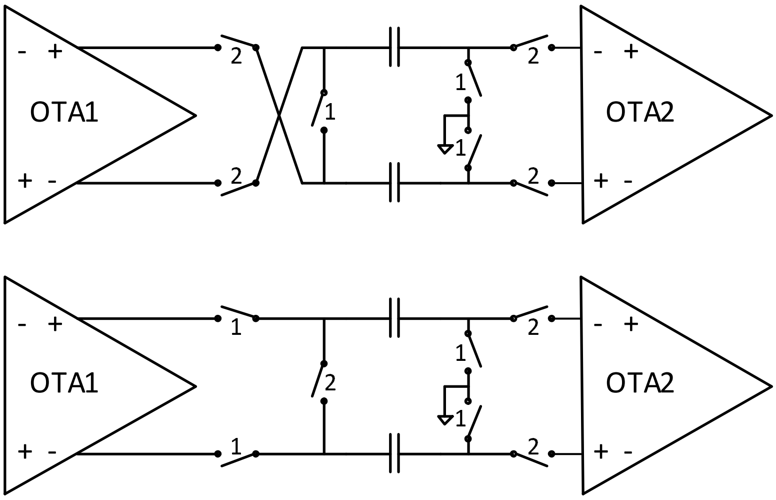

Figure 7.9: Charge transfer between the two first integrators. Top: direct connection. Bottom alternate connection.

The third integrator uses a type #5 input for the reference and a type #2 input for the signal from the first loop. The load on the amplifier is, here as well, distributed between both phases.

7.3.5 Incremental and ΣΔ modes

The decision of designing an incremental modulator rather than a sigma-delta ADC was motivated by the following reasons:

- Incremental converters are not sensitive to idle tones.
- Resetting the modulator removes previous information. Two successive samples are thus completely independent. It is a necessary condition if the ADC measures multiplexed signals, delivered by several sensors.
- It is possible to perform single conversions. It is useful if the converter is only used time to time to acquire a signal and is in stand-by mode most of the time to save power.
- Simplified implementation of the 1/f noise rejection.

The rejection of the flicker noise is traditionally done in the ΣΔ converters with correlated double sampling techniques or with choppers surrounding the amplifiers. Whatever technique is implemented, the complexity of the most critical block - the first amplifier - is increased, hardening the layout and increasing the parasitic coupling errors. Moreover, the compensation has to be done while the amplifier is working.

In the selected method to reject the flicker noise of the modulator, a half-conversion (i.e. of length OSR/2) is first done. The inverse of the input signal is then acquired during the second half-conversion[6]. The results of both conversions are finally subtracted in the digital filter to get the final output. In order not to loose any information, the last integrator is not reset between the two half-conversions in order not to delete the residual quantization error. Instead, it is subtracted to the second half-conversion, swapping the upper and lower integration capacitors of the last amplifier.

The aforementioned implementation rejects the offset of the whole modulator and not of the amplifiers only. Furthermore, the extra circuitry is active in a phase (reset phase) during which the modulator is idle and is located in an insensitive area - the last integrator. In return, the quantization noise is slightly increased and the digital filters are more complex, to take into account the recombination of the positive and negative signals.

Note: If the quantization noise is far below the thermal noise, it is possible to further reduce the 1/f noise, lowering the OSR of each unit conversion (i.e. positive and negative) and increasing proportionally the number of unit conversions. This operation does not modify the thermal noise level, but increases the quantization noise.

---

[6] The signal is easily inverted, toggling the polarity of the input choppers.

7.3.6 Coefficients

The gain coefficients are computed starting with the generic schematics of the second- and first-order loops, Fig. 7.2 and 7.3.

- In order to optimize the noise and the power consumption, all amplifiers are telescopic OTAs. As the output range of such structures is strongly limited, small gains of 1/8 and 1/16 are defined for the reference.
- In order not to change the gain of the initial loops, coefficients of 1/8 are also set on the signal path. (1/16 for the 2^{nd} integrator).
- The coefficient 1/2 between the two first integrators is defined to keep the same loop gain.

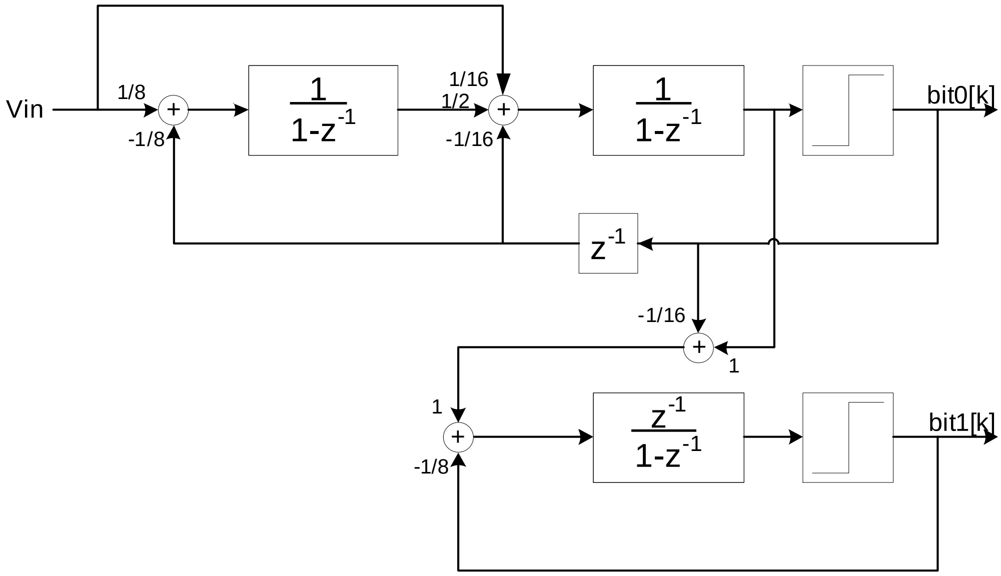

Figure 7.10: High-level view of the implemented architecture.

7.3.7 Sizing

The thermal noise determines a minimal value for the capacitors. A resolution over 20 bit is reached for a $C_{in}OSR$ product of 16nF. The thermal noise of the modulator alone should be slightly lower as the ADC and the front-end contribute both to the noise level[7]. Input capacitors of 2.56pF for the input of the first stage are selected, with an OSR of 2^{14}. This choice is a tradeoff between:

- The surface of the capacitors. The integration capacitors are already large (40pF).

---

[7] It is considered that the ADC and the front-end contribute equally to the thermal noise. This assumption is reasonable as the power budget of both blocks is comparable.

- The clock frequency used to generate the phases (8MHz). The maximal frequency of the digital system is equal to 16MHz.
- Minimal unit capacitors of 20fF in the second integrator for the signal and the reference. 20fF is the minimal size for mim capacitors in this process. The gain programmability in the second integrator is done selecting a variable sized capacitor.

The oversampling frequency is set at 1MHz, to reach 20 bits at 64Hz.

The values of the remaining capacitors of the first integrator are determined according to the gains in Fig. 7.10. The capacitors of the next integrators are smaller, as they contribute less to the noise budget. The minimal size of the capacitors is determined according to matching and process constraints.

Programmable gain: The gain of the ADC should be programmable between 1/4 and 16. In order not to increase the surface of the capacitive array, the typical configuration with a unity gain uses the largest capacitors. The positive gains 1 to 16 are set reducing the value of the capacitors connected to the reference, while the attenuations 1/4 to 1 are configured reducing the capacitors sampling the signal.

The attenuation of the reference is programmable between 1 and 1/16. Unit capacitors of 160fF are thus selected for the input of the first stage, for a total of 160fF*16=2.56pF on the reference.

The input capacitors on the signal branch are programmable between 320fF and 2.56fF in order to measure a large signal (e.g. the reference) without saturating the modulator. The attenuation is programmable between 1/8 and 1.

In order to reduce the number of unit capacitors forming the integration capacitors of the first integrator (40pF), the matching are the following:

- Matching of the integration capacitors of the first integrator with the coupling capacitors between the two first integrators.
- Matching of all input capacitors equal or over 160fF.

In order to reduce the power consumption of the second integrator and maximize the matching of the capacitors in the most critical case[8], input capacitors of 160fF are used for the inputs for the signal and the reference in the second integrator if a unity gain is selected.

As the same division factor is used in both first integrators, the smallest capacitors, to connect the reference to the 2nd integrator, are equal to 160fF/16=10fF. As the matching of the capacitors is less sensitive in the second integrator, these capacitors are not

---

[8] I.e. low gain in the PGA and important relative contribution of the ADC to the thermal noise

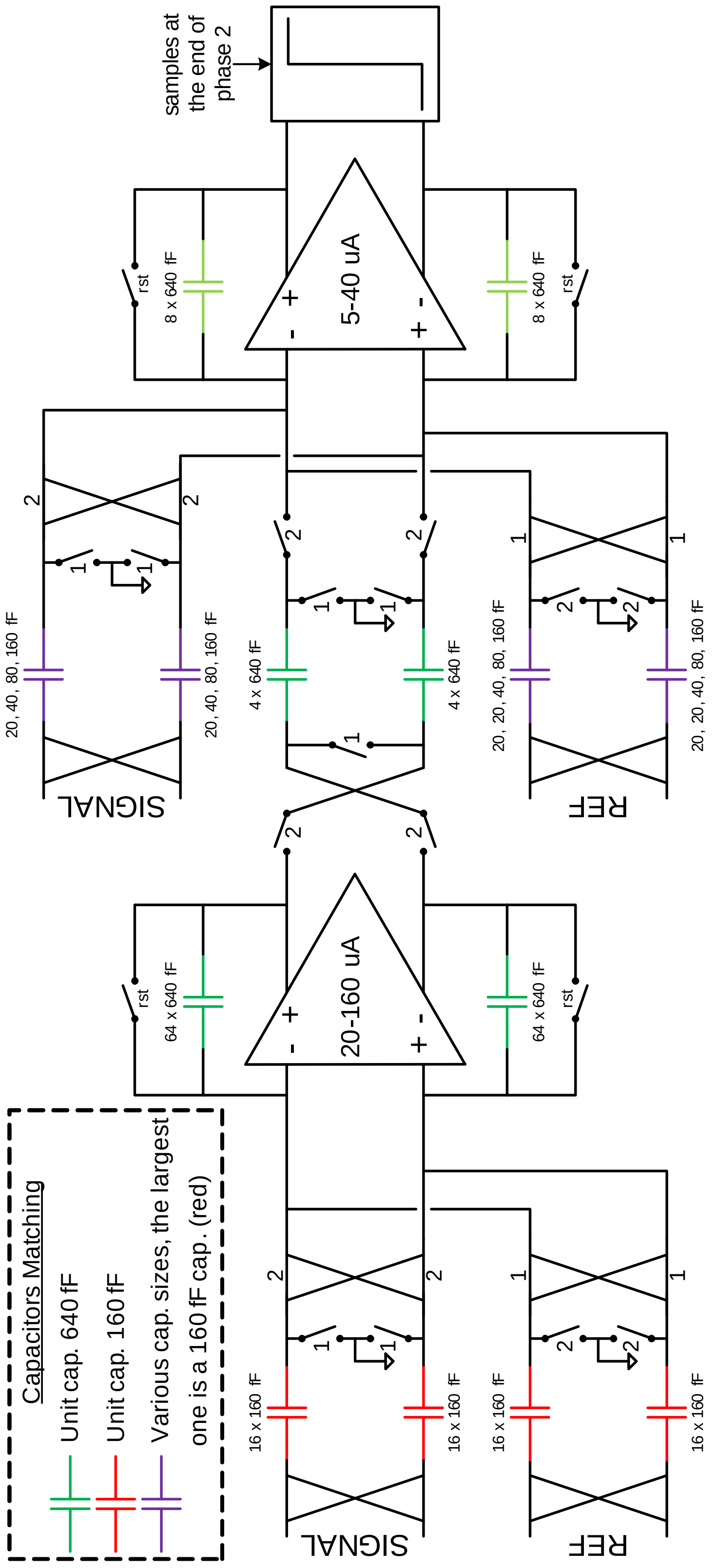

Figure 7.11: Implementation of the first stage.

matched to the capacitors of the first integrator. These programmable capacitors are made of a set of unit capacitors with various sizes (typ. 20fF, 40fF, 80fF not matched and 160fF matched with the first integrator). The smallest capacitors (10fF) are not implemented, but the pre-charge sequence is modified connecting these capacitors together instead of a connection to the inverted reference. The capacitors in the signal branch are implemented in a similar manner.

The third integrator uses capacitors from 20fF to 160fF, identical to the ones of the second integrator.

Remark: Due to the capacitive mismatch, the performances are degraded if the gain of the ADC is not unitary. The performances of the MASH 2-1 depend on the matching between the stages. The benefit of the 3rd integrator is thus not guaranteed if an attenuation of the signal or of the reference is done in the ADC. The attenuation on the reference should only be used when the gain of the PGA is not high enough. In this last case, the resolution of the whole acquisition chain is limited by the thermal noise of the PGA and the last stage is not useful.

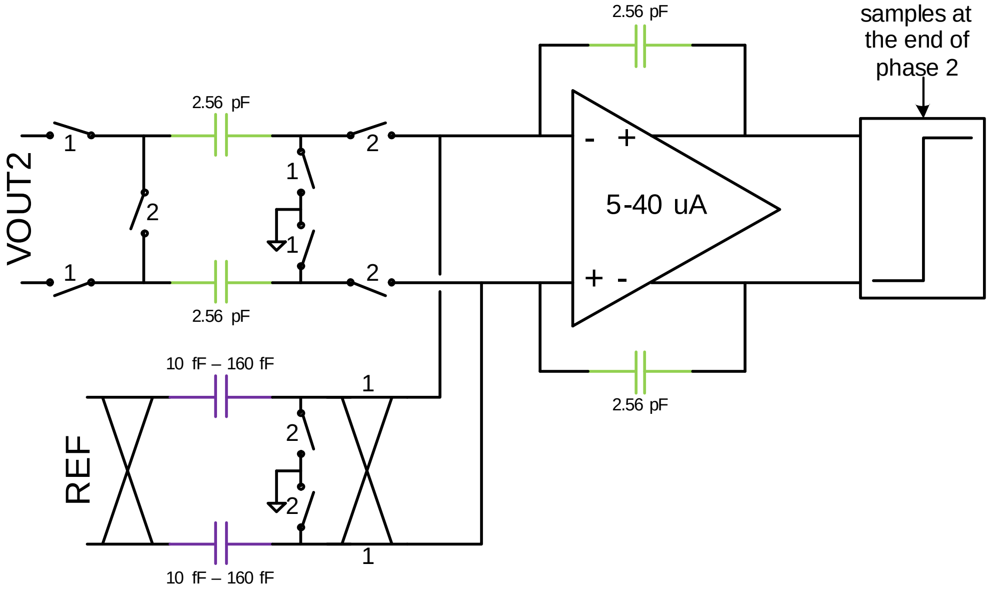

Figure 7.12: Implementation of the second stage.

Power consumption: The power consumption of the amplifier should be sufficient to cover a slewing and a settling phase during the integration phases. A current consumption of 80μA for the first amplifier and of 20μA for the next ones has been determined using the developed tool.

A safety margin has been added to the minimal power consumption computed with the high-level simulations. A fine programmability of the biasing of the amplifier is implemented in order to test the critical power consumption. The biasing of each amplifier is programmable from 0.25 to 2 time its nominal value.

Figures 7.11 and 7.12 summarize graphically the implementation of the modulator. The switching sequences, the capacitors sizes and matching, as well as the power consumption of the amplifiers are included.

7.4 Digital controls and filters

This section describes the synthesized blocks directly related to the analog to digital conversion chain. Some of the functionalities are implemented at 1.8V, as the digital filters or the generation of the ADC specific clocks, while a higher voltage (3.3V transistors) control block is integrated directly in the acquisition chain for a closer driving of the blocks.

The main aims of these two blocks are, in the 1.8V domain:

- To handle the whole acquisition chain frequency management, including the frequency division and the ADC duty cycle of the non overlapping signals. This block delivers the clock signals for the 3.3V block.
- To convert the digital bit-streams of the modulator 'bit0/1' to a 32 bits output through digital filters.
- Management of the beginning of the conversions in the 'conversion on demand' mode[9]n this mode, the ADC is not acquiring continuously the signal, in order to save power consumption..

and in the 3.3V domain:

- Generate the clocks for the voltage reference and voltage doubler.
- Generate the controls of the PGA.
- Generate the controls of the ADC.
- Generate the controls of the digital filters (i.e. so the filters know in which state is the modulator).

7.4.1 Controls

The 3.3V control block contains the following main function:

- Generation of the main clocks.
- State machine generating the different steps of the conversion algorithm.
- Generation of the control signals for the modulator combining the FSM state

---

[9] I

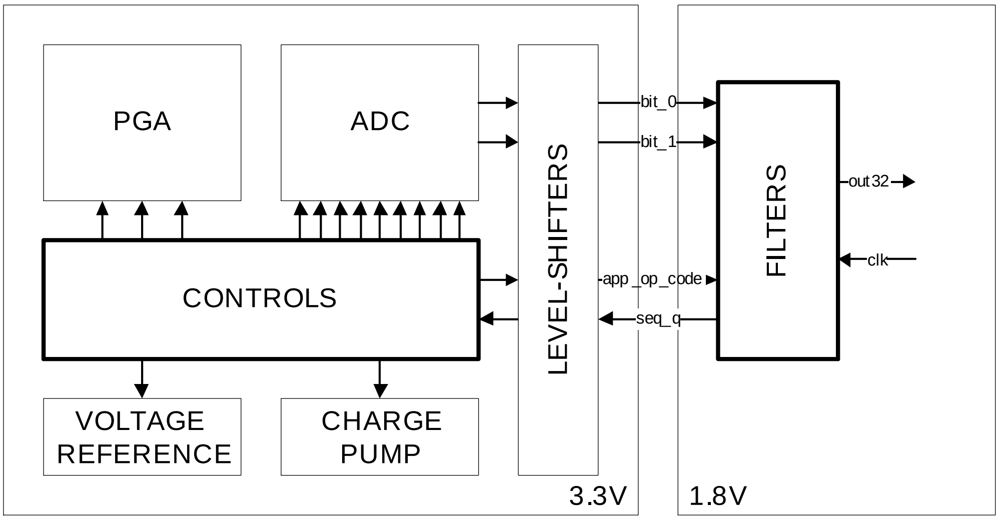

Figure 7.13: General view and connections of the two dedicated digital blocks.

and the clock signals.

- Digital filter controls generation.

The clocks generator is combining the clock signals from the 1.8V digital ’seq_q’ to generate two main internal non-overlapping clocks, driving the pre-charge and integration phases of the ADC. The duty cycle of these clocks is either 1/4, 3/8, 7/16 or 15/32, and is set in a specific user register.

The finite state machine is generating the basic instructions of the modulator. The main instructions are the integration (I) and reset (R). In addition to these, the operations of wait (W), and of last residue evaluation (L) are also implemented. If the circuit is not active, it is in a no-operation state (N).

The extra evaluation of the residue (L) quantizes the last integrator output after the end of the last integration of $+V_{in} \pm V_{ref}$. This reduces the quantization noise by 2.

The operation sequence is, for an OSR of 8 and an number of elementary conversions equal to two: N R I I I I I I I I R R I I I I I I I I R N N N for the first integrators and N N R I I I I I I I I W W I I I I I I I I L N N for the last integrator. The wait operation was implemented to keep the same conversion length for each elementary conversion. The sampling duration is thus $nelconv(osr+2)$, where $nelconv$ is the number of elementary conversion, implemented to reject the 1/f noise (see section 7.3.5). The extra unnecessary reset/wait phase between the two half-conversions was implemented to improve the synchronization with the front-end, as the front-end working frequency is generated dividing the A/D frequency by an integer number. If the ADC is continuously sampling, the no-operation steps are replaced by identical sequences.

7.4.2 Filters

The main aim of the digital filters is to convert the bitstream(s) bit_0 (and bit_1) of the modulator into a 32 bits output bus.

There are two main families of filters implemented, the ones for the incremental mode and the ones for the sigma-delta mode as the modulator can operate in both modes.

In the incremental mode, the content of the accumulating registers are first reset, then the bit-streams are integrated, normalized and the result is sent to the output registers. In this mode, the precision of a conversion (reduction of the quantization noise) is increased using larger osr and/or $nelconv$.

In the sigma-delta mode, the bit-streams are continuously accumulated. The integrated value goes then through a differentiator before being normalized and outputted. The $nelconv$ setting is not used at all is this mode as no 1/f noise rejection was implemented for the $\Sigma\Delta$ mode. The osr value is thus only determining the output code refresh rate.

The structure of the digital filters is summarized in Fig. 7.14. The description block by block of the accumulator is the following:

Control logic, operation code selection and timing generation: The output streams of the modulator are combined with the application operation code to tell the ALU to perform an addition or not.

ALU: The arithmetic and logic unit is incrementing or decrementing the $reg\_0/1/2$ values according to the bit_0 and bit_1 values. This block is also adding the $reg\_0/1/2$ registers together.

Inc. mux: The multiplexer for the incremental mode is selecting which output of the ALU should be used, depending of the modulator configuration. If the third order modulator is used, a multiplication by three of the reg_slave_2 value is also applied.

$\Sigma\Delta$ in mux: The input multiplexer for the sigma-delta mode selects the value to provide to the differentiator between the output of the 2^{nd} latch (reg_slave_1), for the one-stage modulator and the output of the 3^{rd} latch (reg_slave_2) for the multistage modulators.

$\Sigma\Delta$ differentiator: The differentiator for the sigma-delta mode is performing the multiplication by $(1-z^{-1})^2$ and by $(1-z^{-1})^3$. The $\Sigma\Delta$ output multiplexer is then selecting which value to use, according to the order of the modulator.

Normalization: This block normalizes the output code. The $nelconv$ and OSR setting are used to shift the output to the right or to the left. The output is truncated on 33

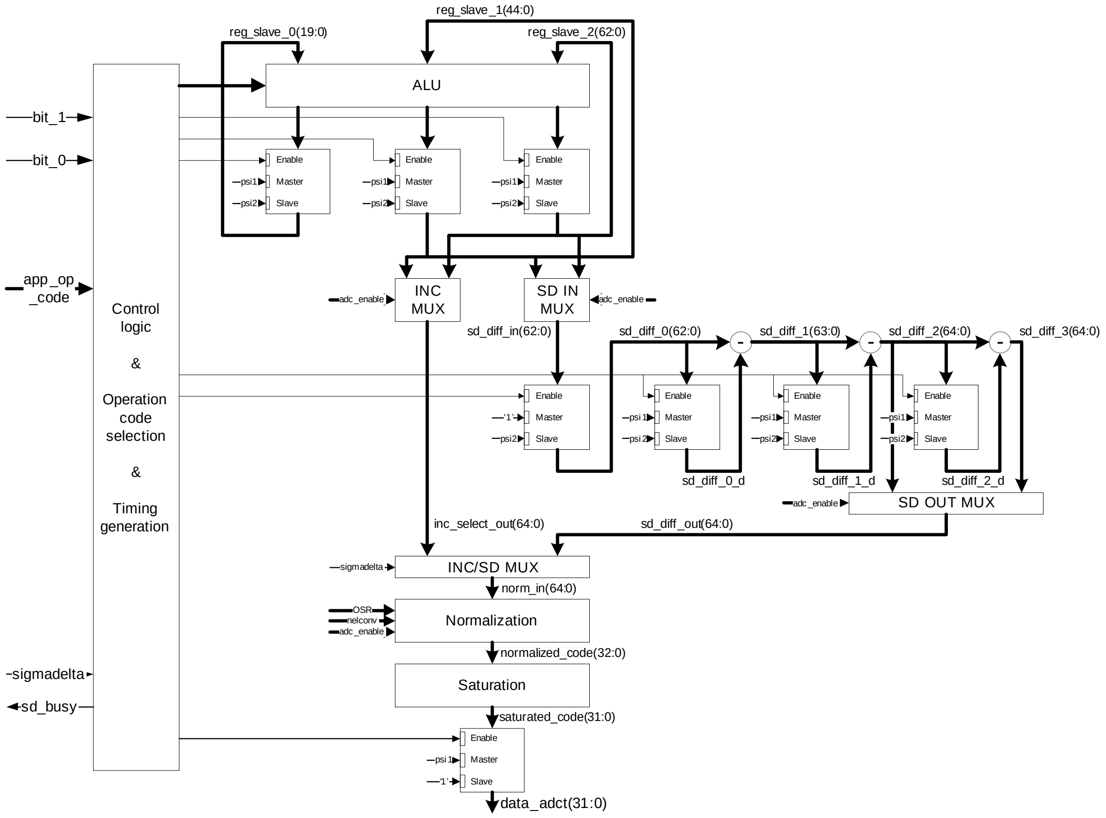

Figure 7.14: Blocks view of the digital filters.

bits

Saturation: The saturation of the modulator is evaluated, comparing the MSB and the 2^{nd} MSB. The output is limited to 32 bits.

Output latch: The result is sent through this last latch to the registers containing the 32 bits converted value.

7.4.3 Mathematical relationships

This section describes the modulator equations and the filter equations for the different available configurations.

In the incremental mode, there are five different configurations. Two for the third order integrator, with and without a direct connection between the two first stages (refer to section 7.3.4, Fig. 7.9), two for the second-order single loop-modulator and one for the MASH 1-1 converter[10] (when the first integrator is disabled).

---

[10] This mode mode was implemented in order to test individually the two last integrators.

In the sigma-delta mode, there are only three different digital configurations, one for the second-order single-loop modulator, one for the third order one and one for the mash 1-1 configuration. This section does not detail the implementation of the standard sinc filters for the $\Sigma\Delta$ mode, but focuses on the filters for the incremental mode.

The variables displayed in the equations are:

• x:	ADC input signal	Real value
• V_{ref}:	reference signal	Real value
• osr:	oversampling ratio	Integer, power of 2
• $nelconv$:	number of conv.	Integer, power of 2
• m:	signal attenuation	Integer (1-8 range, default:8)
• n:	reference attenuation	Integer (1-16 range, default:16)
• $Bit0[k]$	first stage output	Boolean, -1/1 value
• $Bit1[k]$	second stage output	Boolean, -1/1 value
• $Bit0\_end$	last bit 0 quantization	Boolean, -1/1 value
• $Bit1\_end$	last bit 1 quantization	Boolean, -1/1 value
• res	last integ. residual voltage	Real value

The indexing of the modulator outputs is not time based, but sample based. I.e, $bit0[k]$ and $bit1[k]$ refer to the k^{th} output of both streams and not to the output of the streams at time 'k'.
The input signal 'x' is assumed to be constant during the whole conversion.

Third-order modulator with direct connection between the two first OTA: [11]

$$x = \frac{3nV_{ref}}{2mOSR(OSR+1)(OSR+5)} \left\{ \sum_{k=0}^{osr\text{-}1} (OSR\text{-}k)(OSR\text{-}k+1)Bit0[k] + 4 \sum_{k=0}^{osr\text{-}1} Bit1[k] \right\} + \frac{768res}{mOSR(OSR+1)(OSR+5)} \tag{7.1}$$

Third-order modulator without a direct connection between the two first OTA:

$$x = \frac{3nV_{ref}}{2mOSR(OSR+1)(OSR+2)} \left\{ \sum_{k=0}^{osr\text{-}1} (OSR\text{-}k)(OSR\text{-}k+1)Bit0[k] + 4 \sum_{k=0}^{osr\text{-}1} Bit1[k] \right\} + \frac{768res}{mOSR(OSR+1)(OSR+2)}$$

---

[11] The implementation of the number of elementary conversions $nelconv$ as well as the management of the last bit quantization $Bit0/1\_end$ is not detailed in equations (7.1), (7.2), (7.4), (7.5) and (7.7) in order not to overload them.

(7.2)

In both third-order configurations, the digital filter is doing the following function:

$$out32 = 2^{31}\frac{3}{OSR^3 nelconv}\left\{\sum_{cnt=0}^{nelconv-1}(-1)^{cnt+1}\left(4\sum_{k=0}^{osr-1}Bit1[k+OSRcnt]\right.\right. \\ \left.\left. + \sum_{k=0}^{osr-1}(OSR-k)(OSR-k+1)Bit0[k+OSRcnt]\right) + 2Bit1\_end\right\} \tag{7.3}$$

Second-order single-stage modulator with direct connection between the two OTA:

$$x = \frac{nV_{ref}}{mOSR(OSR+3)}\sum_{k=0}^{osr-1}(OSR-k)Bit0[k] \\ + \frac{256res}{mOSR(OSR+3)} \tag{7.4}$$

Second-order single-stage modulator without direct connection between the OTA:

$$x = \frac{nV_{ref}}{mOSR(OSR+1)}\sum_{k=0}^{osr-1}(OSR-k)Bit0[k] \\ + \frac{256res}{mOSR(OSR+1)} \tag{7.5}$$

In both second-order single-loop configurations, the digital filter is doing the following function:

$$out32 = 2^{31}\frac{2}{OSR^2 nelconv}\left\{\sum_{cnt=0}^{nelconv-1}(-1)^{cnt+1}\sum_{k=0}^{osr-1}(OSR-k)Bit0[k+OSRcnt]\right. \\ \left. + Bit0\_end\right\} \tag{7.6}$$

Second-order two-stages modulator:

$$x = \frac{nV_{ref}}{mOSR(OSR+1)}\left\{\sum_{k=0}^{osr\text{-}1}(OSR-k)Bit0[k] + 2\sum_{k=0}^{osr\text{-}1}Bit1[k]\right\} \\ + \frac{256res}{mOSR(OSR+1)} \tag{7.7}$$

The digital filter in this configuration is doing the following function:

$$
\begin{aligned}
out32 = 2^{31}\frac{2}{OSR^2 nelconv}\Bigg\{\sum_{cnt=0}^{nelconv-1}(-1)^{cnt+1}\Bigg(2\sum_{k=0}^{osr-1}Bit1[k+OSRcnt] \\
+\sum_{k=0}^{osr-1}(OSR-k)Bit0[k+OSRcnt]\Bigg)+2Bit1\_end\Bigg\}
\end{aligned}
\tag{7.8}
$$

$$
\begin{aligned}
out32 = 2^{31}\frac{2}{OSR^2 nelconv}\Bigg\{\Bigg(2\sum_{k=0}^{osr-1}Bit1[k] \\
+\sum_{k=0}^{osr-1}(OSR-k)Bit0[k]\Bigg)+2Bit1\_end\Bigg\}
\end{aligned}
\tag{7.9}
$$

In all the architectures, the correspondence between the output of the digital filter $out32$ and the analog input x is:

$$x \cong \gamma\frac{V_{ref}}{2^{31}}\frac{n}{2m}out32 \tag{7.10}$$

where γ is a coefficient to correct the gain error. The gain error is introduced by the normalization block, which only performs divisions and multiplications by powers of two. E.g., in the first situation with the third-order modulator, a division by $OSR(OSR+1)(OSR+5)$ should be implemented (7.1). Instead, the digital filter (7.3) is dividing the sum by OSR^3. The gain error is thus $(OSR+1)(OSR+5)/OSR^2$. Table 7.3 summarizes the error coefficients γ for the five configurations.

Architecture	γ
Third-order Direct connection	$\frac{OSR^2}{(OSR+1)(OSR+5)}$
Third-order Indirect connection	$\frac{OSR^2}{(OSR+1)(OSR+2)}$
Second-order Direct connection	$\frac{OSR}{OSR+3}$
Second-order Indirect connection	$\frac{OSR}{OSR+1}$
Second-order MASH 1-1	$\frac{OSR}{OSR+1}$

Table 7.3: Gain error coefficients.

7.5 Conclusion

This chapter detailed the top-down design of a programmable gain ΣΔ converter targeting 20 bits. A configurable architecture - second-order or MASH 2-1 - has been selected to provide an ADC with configurable sample rate and resolution.

The architecture and the implementation with switched-capacitors were defined with the results provided by the simulator. High-level simulations showed then that the selected topology balances well the thermal noise and the power consumption, but also showed that it is sensitive to three non-ideal effects: 1) potential loss of charges during charge transfers, 2) asymmetrical clock feedthrough and 3) parasitic capacitors.

Each effect was analyzed, evaluated and, whenever necessary, the initial circuit was modified. Finally, the layout validation extensively benefited from the tools to identify parasitic capacitors and a cleaned version was obtained after a few iterations.

8 PGA

A full acquisition chain, Fig. 8.1 transforms the analog signal from the sensor into a digital representation. As the amplitude of the voltage delivered by the sensor may be very low, a front-end amplifying the signal before the ADC is commonly required. If the gain is selectable by the user, the front-end is a Programmable Gain Amplifier – PGA.

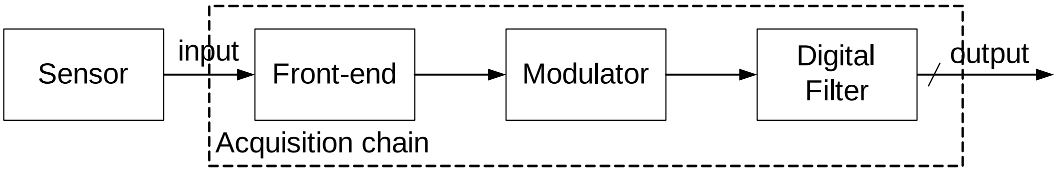

Figure 8.1: Acquisition chain.

In addition to the amplification of the signal, the second fundamental task of the front-end is to be a buffer between the sensor, with very limited output currents, and the ADC, periodically transferring charges.

The main specification of the front-end are summarized hereafter. In addition to the previously mentioned capabilities, a low power consumption and an extended input range are also required.

- Low noise: $15nV/\sqrt{Hz}$ input-referred in the $0-100Hz$ band.
- Low power: $< 200\mu A$
- Sensor output impedance range: 100Ω - $1M\Omega$
- Gain: programmable, 10 typ.
- Process: CMOS $0.18\mu m$, $1.8V-3.6V$
- Input range: $Vss+50mV$ - $Vdd-50mV$
- Controlled output common-mode.

The selected structure, Fig. 8.2 is based on the classic instrumentation amplifier. The programmable gain is implemented with the tuneable resistive feedback.

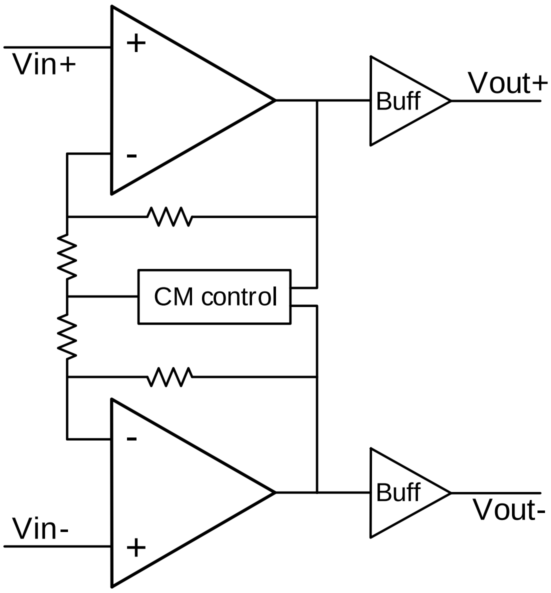

Figure 8.2: Architecture of the implemented front-end.

The amplifiers in the instrumentation structure are amplifiers with a flicker noise rejection circuit. The next section introduces the selected topology and provides a detailed analysis of the contribution to the noise level of each component. The final goal is to optimize the noise – power consumption factor.

Section 8.3 addresses the problem of the saturation of classic three-opamp instrumentation amplifiers with the common-mode. The new structure in Fig. 8.2, that controls the output common-mode is analyzed in details.

Finally, the last design section of this chapter introduces the output buffers of the PGA, to interface a continuous time front-end with the switched capacitor ADC. The output buffers are delivering efficiently most of the charges to the ADC and alleviate the design requirements on the PGA.

8.1 Auto-Zero Amplifier

8.1.1 Offset and Flicker Noise Reduction Techniques

In integrated circuits, the two main noise sources are the thermal noise and the flicker noise. The thermal noise, generated in the resistors and the transistors has a white noise spectrum, independent of the frequency. Lowering the white noise level is usually done increasing the power consumption. The flicker noise, also known as

1/f noise as its spectral density is inversely proportional to the frequency, is mostly generated in the active devices. The most straightforward way to reduce 1/f noise is to increase the device area.

There are several rejection techniques to reduce offset and 1/f noise. Even if most of them operate at circuit level, some physically reduce the noise generated in the transistors, de-correlating the samples [59]. The most common circuit techniques are briefly reviewed hereafter:

Correlated Double Sampling. CDS techniques sample during a first initialization phase the offset (and the 1/f noise) at the input of an amplifier, so that it can be further compensated during the active phase. These techniques are well-suited for discrete systems, such as switched-capacitors circuits where they are widely spread [60, 61, 62].

Another discrete time compensating method [63] is compensating the offset injecting a current at the output of an amplifier. During an initialization sequence, the offset is measured and compensated by successive approximations. This method requires the amplifier to be disconnected from the rest of the circuit during the measure phase.

Chopper Stabilization. In the CHS technique, the noise sources, usually amplifiers, are placed between a modulator and a demodulator to shift the 1/f noise outside the signal baseband frequency [62, 64, 65]. Due to their simplicity, choppers are usually preferred over pure single frequency (sinusoidal) modulators.

Auto-Zeroed Amplifiers. This method (Fig. 8.3), analysed in [66], stores a measurement of the offset in an auxiliary branch, in parallel with the main amplifier. The offset is then compensated through a secondary input located after the input differential pair of the main amplifier. Also called Ping-Pong auto-zeroing when two nulling paths are used in alternance, this method is well-suited for continuous-time circuits. Sometimes used alone is the past [67], is it nowadays often combined with chopper modulation techniques [65, 68, 69].

8.1.2 Proposed Circuit

The specifications for the auto-zero amplifier are almost identical than the ones of the full PGA. The main differences are a lower noise level ($10nV/\sqrt{Hz}$ instead of $15nV/\sqrt{Hz}$) and a lower power consumption ($100\mu A$ instead of $200\mu A$). Indeed, both current consumption and noise power are doubled in the dual structure.

Sample-data systems are not suited for this application due to the wideband noise aliasing occurring in the sampling process, e.g. in a switched-capacitor circuit, the noise level is set by the input sampling capacitors: $V_{n,RMS}^2 = kT/C$. As a result, an

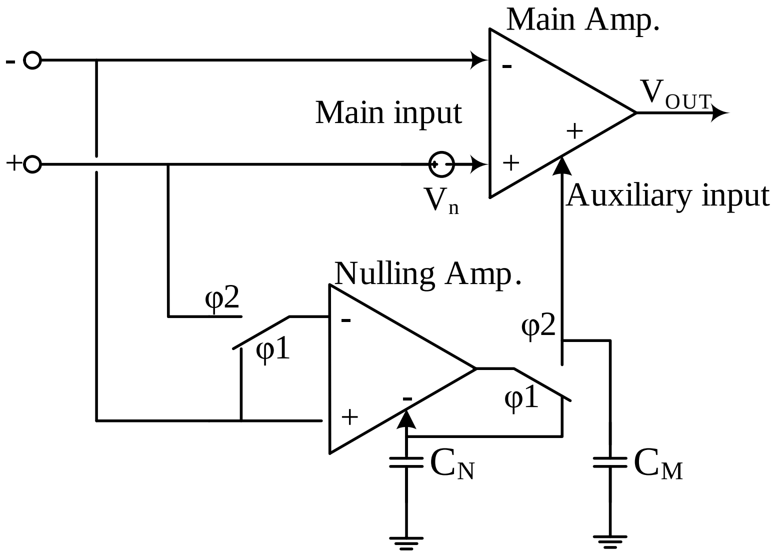

Figure 8.3: Classic continuous time auto-zeroed amplifier.

increased power consumption and large chip area can be expected.

Simple chopper stabilization circuits using switched capacitors as the input stage (Fig. 8.4) are only applicable for low impedance sensors.

Classic continuous-time auto-zero amplifiers (Fig. 8.3) are not suitable for very low noise circuits as the noise of the nulling amplifier is sampled and aliased down in the signal baseband during its own offset compensation phase.

Ping-Pong auto-zeroed amplifiers are neither suited for low-noise low-power applications as the duplicated nulling path is nearly doubling the power consumption.

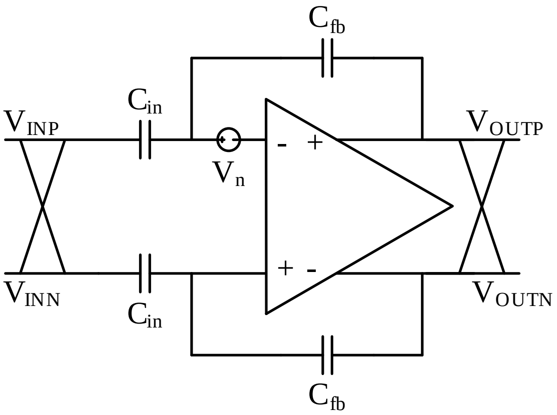

Figure 8.4: Flicker noise rejection using chopper modulation.

The PGA noise spectral density has to be lowered using a continuous-time topology

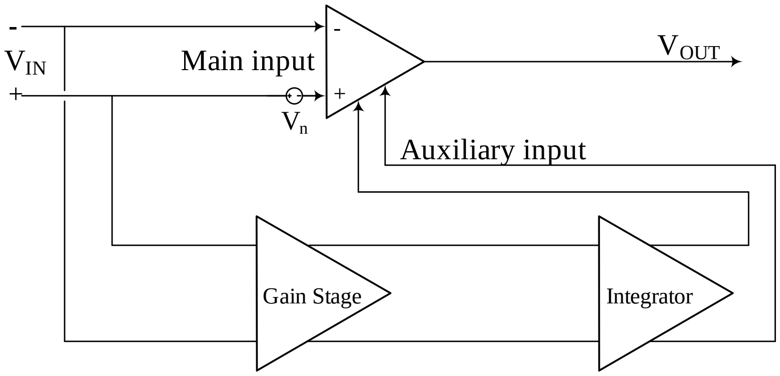

Figure 8.5: Auto-zero amplifier principle.

in order not to alias the thermal noise. Moreover, the instrumentation amplifier is used in an acquisition chain including a switched-capacitors ADC. The discrete time structure of the ADC samples at frequency f_s the signal as well as the thermal noise from the PGA. In order not to increase the noise level from the PGA, it is necessary to low-pass filter (below $f_s/2$) the thermal noise of the PGA before sampling in the ADC.

The architecture of the proposed circuit is inspired by the classic auto-zero amplifier (Fig. 8.3), using a continuous-time main amplifier and a nulling branch connected to an auxiliary input.

The offset and flicker noise of the main amplifier V_n are compensated with the auxiliary branch. The offset compensation principle, shown on Fig. 8.5, first amplifies the offset in a gain stage and then integrates it to reduce efficiently both offset and 1/f noise.

The continuous-time low-noise first gain stage amplifies the offset without aliasing the high-frequency noise components. As the gain stage has a finite bandwidth, it acts as a low-pass filter before sampling the 1/f noise signal in the switched capacitors integrator.

This structure allows, as the classic auto-zero, a rejection of the 1/f noise in a continuous time system. The fully differential architecture of the nulling branch has the following advantages:

- Very good rejection of the offset and 1/f noise of the main integrator, as the combined gain of the pre-amplification and of the integrator is used.
- Lower gain stage offset, compared with a single-ended structure. The offset is only created by the transistors mismatch. This lowers statistically the mean current delivered by the sensor.
- As the system is not having anymore two distinct phases (with different loads,

and operating conditions), stability is easier to achieve.

- The sensitivity to external references (necessary in [66], Fig. 8.3 for the auxiliary inputs) is lowered. The only required voltage reference left is the one generating the common-mode. It is used to properly bias the input and output common-modes of the amplifiers.

Moreover, the equivalent DC input bias current due to the switching operation of the input chopper is proportional to the residual input offset. This architecture is designed for input bias currents in the $1-100pA$ range.

8.1.3 Implementation

The implementation of the amplifier is split into several steps to help the understanding of the reader.

Step 1, Fig. 8.7. The gain stage is implemented using an amplifier with capacitive feedback and capacitive input. Modulation with choppers rejects the flicker noise of the amplifier. The capacitive gain guarantees a zero DC input current during each phase of the chopper. This structure also allows decoupling the common-modes of the input and of the virtual ground of the gain amplifier.
The input resistors of the integrator are emulated using switched-capacitors. In order to have symmetrical phases, the capacitors are crossed instead of being connected to a reference potential (Fig. 8.6).

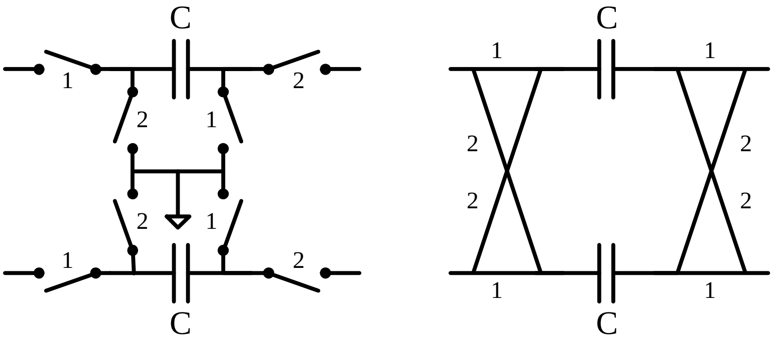

Figure 8.6: Switched-capacitors resistor using left: standard implementation, right: symmetrical implementation with choppers.

Step 2, Fig. 8.8. The flicker noise of the integrator has to be rejected. The amplifier of the integrator is thus surrounded by choppers to push the offset and 1/f noise outside the signal baseband.

If identical frequencies are selected for the modulation of the auxiliary chain amplifiers and for the switching of the integrator input capacitors, it is possible to reduce the

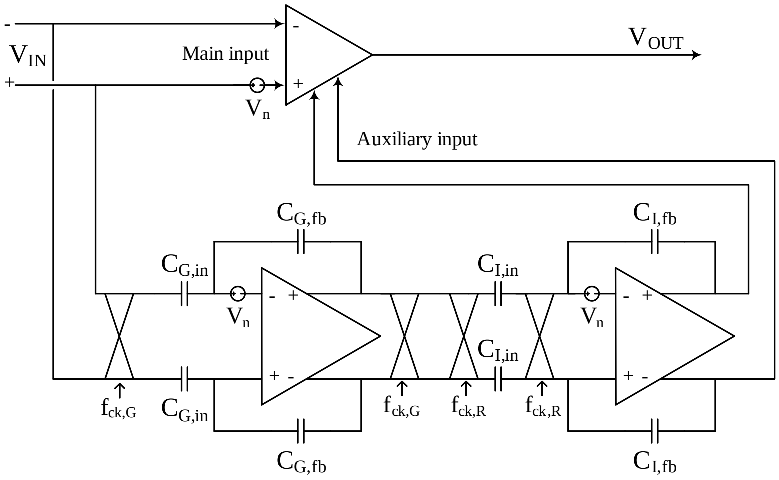

Figure 8.7: Auto-zero amplifier: step 1.

number of extra switches. The one demodulating the gain stage is compensated with the one on the left of the integrator's input capacitors. The chopper on the right of the switched capacitors is combined with the one modulating the input of the integrator's amplifier to have only one chopper left on the feedback path. The three remaining choppers are shown on Fig. 8.9.

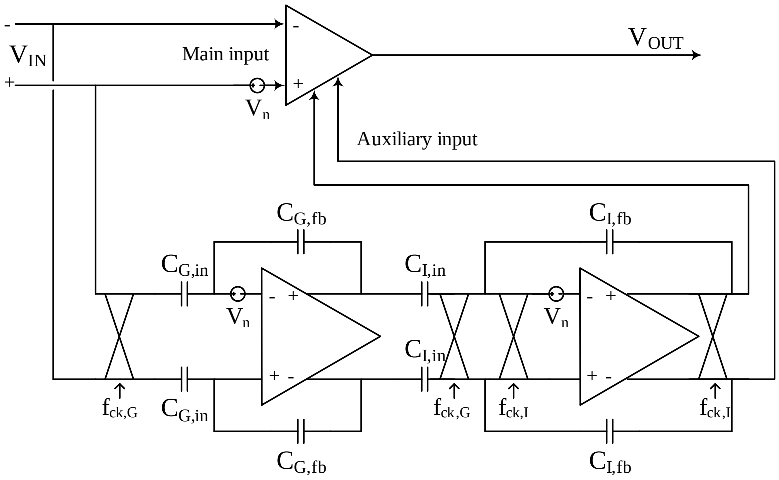

Figure 8.8: Auto-zero amplifier: step 2.

Step 3, Fig. 8.9. The architectures of the amplifiers are defined. A telescopic structure is selected for the OTA of the gain amplifier to optimize the noise - power consumption ratio and to have a sufficiently high open-loop gain. The reduction of the output voltage dynamic range induced by this cascode topology is acceptable as the input of the integrator is supposed to be low once the loop is settled.

The OTA of the integrator should have a high DC gain, a large dynamic range and a power consumption much lower than the previous stage. A folded-cascode structure is appropriate here, as the noise contribution of this amplifier is much lower than the noise level of the gain stage (detailed further in the next section).

The input capacitors of the amplifier and the gain of the first stage are finally sized to minimize the noise contribution of the integrator while keeping a reasonable silicon area for the capacitors. The sizing of the components and the definition of the biasing currents are further detailed in the next section.

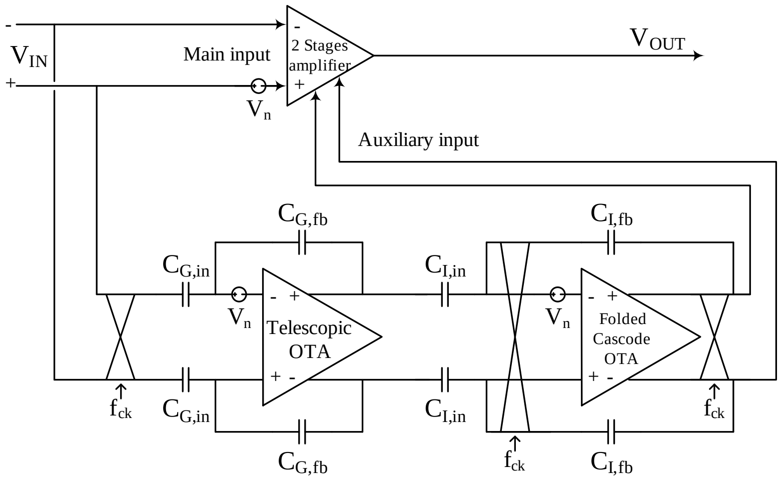

Figure 8.9: Auto-zero amplifier: final implementation.

Remark: For the sake of clarity, the circuits controlling the common-mode at the input and at the output of the amplifiers are not shown in Fig. 8.5 to 8.9. They are implemented using switched-capacitor techniques keeping in mind the noise constraints.

8.2 Noise Analysis

Two noise effects are considered in the noise analysis:

- offsets and flicker noise are placed in the first category. Rejection techniques, auto-zero and frequency shift, are implemented to reduce these noises,
- the second category is dedicated to the wideband thermal noise. As this noise is covering the whole spectrum, it cannot be reduced using modulation or correlation techniques.

8.2.1 1/f noise - Auxiliary chain

As the two amplifiers in the nulling path are working around the modulation frequency f_{ck}, their offsets as well as their 1/f noise have only little impact on the global output noise. The remaining noise contribution of the flicker noise around the modulating frequency is reduced by increasing the gate area of the input differential pairs.

8.2.2 1/f noise - Main amplifier

To analyse the noise at low frequencies (in the signal baseband), the following assumptions are considered:

- the main amplifier open-loop DC gains are finite, A_{M0} for the main input and A_{X0} for the auxiliary input.
- the OTA open-loop gain at the chopper frequency of the gain stage is $A_{Gf_{ck}}$.
- the OTA open-loop gain at the chopper frequency of the integrator is $A_{If_{ck}}$.

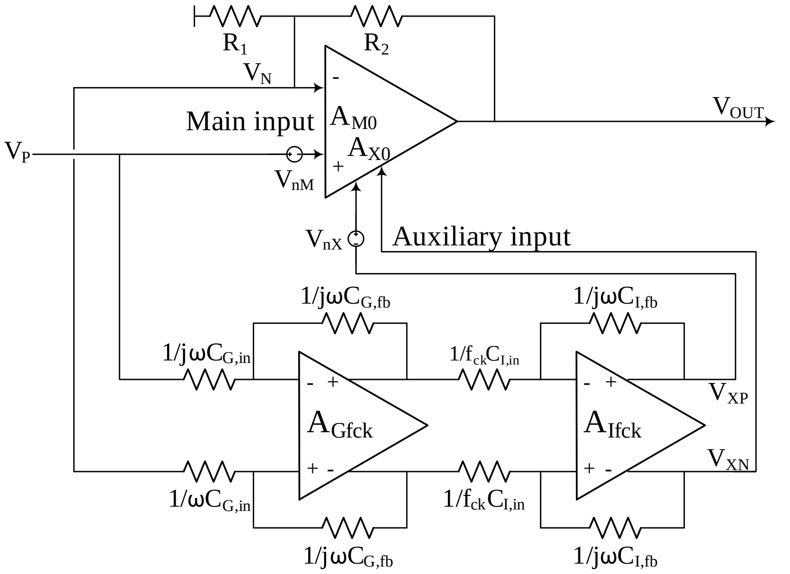

Figure 8.10: Complex representation of the nulling amplifier in the signal baseband (low frequencies).

The naming of the gains, nodes and noise sources are shown on the complex representation, Fig. 8.10. The transfer function of the gain stage is given by:

$$H(j\omega) = -\frac{C_{G,in}}{C_{G,fb}} \frac{A_{Gf_{ck}}}{A_{Gf_{ck}} - 1 - \frac{C_{G,in}}{C_{G,fb}}} \cong -\frac{C_{G,in}}{C_{G,fb}} \tag{8.1}$$

For the integrator:

$$\begin{aligned} H(j\omega) &= \frac{A_{If_{ck}}}{1 + j\omega \frac{1-A_{If_{ck}}}{f_{ck}} \frac{C_{I,in}}{C_{I,fb}}} \\ &\cong \frac{A_{If_{ck}}}{1 - j\omega \frac{A_{If_{ck}}}{f_{ck}} \frac{C_{I,in}}{C_{I,fb}}} \end{aligned} \tag{8.2}$$

At low frequencies, below the pole of the integrator, the transfer function of the integrator is the open-loop gain: $H(j\omega) = A_{If_{ck}}$.

Referencing the offsets and 1/f noise to the input V_P of the instrumentation amplifier, the Flicker Noise Rejection Rate is, for the main ($\mathrm{FNRR_M}$) and auxiliary ($\mathrm{FNRR_X}$) inputs:

$$\begin{aligned} \mathrm{FNRR_X} &= \frac{V_P}{V_{nX}} = \frac{C_{G,fb}}{C_{G,in}} \frac{1}{A_{If_{ck}}} \\ \mathrm{FNRR_M} &= \frac{V_P}{V_{nM}} = \frac{C_{G,fb}}{C_{G,in}} \frac{A_{M0}}{A_{If_{ck}} A_{X0}} = \frac{A_{M0}}{A_{X0}} \mathrm{FNRR_X} \end{aligned} \tag{8.3}$$

The previous equations show that it is possible to increase the signal over noise ratio, increasing the gain of the gain stage or the open-loop gain of the OTA in the integrator. The noise contribution of the main input can also be reduced, balancing the main and auxiliary open-loop gains of the main amplifier. The equations (8.3) being the offset rejection rate, the equivalent offset at the input of the amplifier is:

$$V_{off,eq} = V_{offM} \mathrm{FNRR_M} + V_{offX} \mathrm{FNRR_X} \tag{8.4}$$

with V_{offM} and V_{offX} the offsets at the inputs of the main amplifier. The mean DC current delivered by the sensor is:

$$\begin{aligned} I_{sensor} &= \frac{V_{off,eq}}{R_{in,eq}} = V_{off,eq} C_{G,in} f_{ck} \\ &= \frac{f_{ck} C_{G,fb}}{A_{If_{ck}}} \left(V_{offM} \frac{A_{M0}}{A_{X0}} + V_{offX} \right) \end{aligned} \tag{8.5}$$

where $R_{in,eq}$ is the equivalent input resistance of the auto-zeroed amplifier. The designed amplifier input current is $3pA$, with an equivalent input resistance of $320k\Omega$ and an estimated equivalent offset of $1\mu V$.

8.2.3 Thermal noise - Amplifiers

The thermal noise of the main amplifier is lowered in the signal baseband using the offset compensation provided by the auxiliary chain.

The thermal noise of the gain stage is not lowered. The thermal and flicker noise at the chopper frequency f_{ck} are shifted to the signal baseband. The 1/f noise can be attenuated increasing the size of the differential pair, while the thermal noise can only be reduced increasing the bias current I_0 of the OTA. For a single transistor, the noise spectral density is given by $NSD = (8/3)kT/g_m$. For telescopic and simple OTAs, the input referred noise is given by[1]:

$$NSD_{IN,OTA} = \frac{16}{3}\frac{kT}{g_{m,dp}}\left(1+\frac{g_{m,dp}}{g_{m,cm}}\right) \tag{8.6}$$

where $g_{m,dp}$ and $g_{m,cm}$ are the transconductances of the transistors of the differential pair and of the current mirror. If the differential pair is in weak inversion ($g_m \cong I_0/(2nU_t)$) and the current mirror is in strong inversion (assuming a reasonable ratio between $g_{m,dp}$ and $g_{m,cm}$ equal to 0.25 to keep some dynamic range at the output), it is possible to link the generated noise to the current consumption.

$$NSD_{IN,gain\_stage} = \frac{16}{3}\frac{kT}{g_{m,dp}}(1+0.25) = \frac{40}{3}\frac{nU_t}{I_0}kT \tag{8.7}$$

The input-referred noise power of the integrator is divided by the gain $C_{G,in}^2/C_{G,fb}^2$. The input-referred noise spectral density is thus given by:

$$NSD_{IN,integrator} = \frac{C_{G,fb}^2}{C_{G,in}^2}\frac{1}{f_{ck}}\frac{\alpha kT}{C_{I,in}}\left[\frac{V^2}{Hz}\right] \tag{8.8}$$

where α is a parameter depending of the amplifier topology. α is greater than one and lower than four for single stage OTAs.

Parasitic capacitors: The parasitic capacitors C_A, Fig. 8.11, are contributing to the noise, as they are switched at the input of the amplifier (as well as the attenuation capacitors on the Vref input of the ADC). The input-referred noise (8.8) becomes, including the parasitic capacitors:

$$NSD_{IN,integrator} = \frac{C_{G,fb}^2}{C_{G,in}^2}\frac{1}{f_{ck}}\frac{\alpha kT\left(C_{I,in}+C_A\right)}{C_{I,in}^2}\left[\frac{V^2}{Hz}\right] \tag{8.9}$$

---

[1] Assuming that the differential pair is in strong inversion. This is a pessimistic assumption, as the NSD for transistors in weaker inversion is lower than $(8/3)kT/g_m$.

As the capacitors C_A include the OTA differential pair capacitors, the integrator input capacitor $C_{I,in}$ should be at least of the same size than the differential pair gate capacitance.

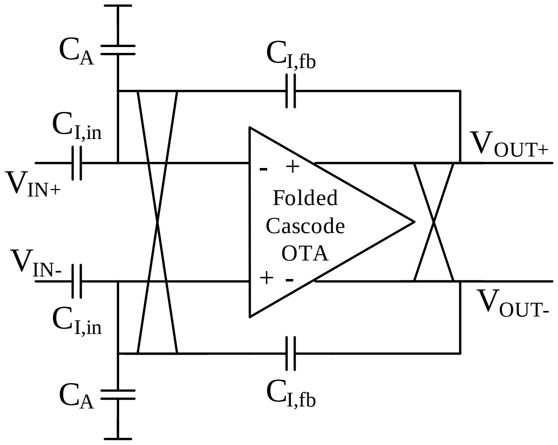

Figure 8.11: Parasitic capacitors in the integrator.

8.2.4 Thermal noise - Input chopper

The noise generated in the switches (with on-resistance R_{sw}) is first amplified by the gain stage and then sampled at the input of the integrator by the switched capacitors.

The sampling aliases the whole noise spectrum into the frequency band between DC and the sampling frequency f_{ck}. The total RMS noise at the input of the integrator is given by

$$N_{RMS}^2 = \int_0^{\infty} 4kTR_{sw} \left|A(f)\right|^2 df \left[V^2\right] \tag{8.10}$$

where $A(f)$ is the transfer function of the gain stage. If the transfer function $A(f)$ has a single pole located in f_{cutoff}, the input-referred NSD of the input chopper is:

$$NSD_{IN,chopper} = 2\pi kTR_{sw} \frac{f_{cutoff}}{f_{ck}} \left[\frac{V^2}{Hz}\right] \tag{8.11}$$

This noise is lowered decreasing the on-resistance R_{sw} of the switches of the chopper.

8.2.5 Thermal noise - Resistors

The thermal noise of the feedback resistors is not affected by the offset compensation. The equivalent circuit including the thermal noise of the resistors is shown in Fig. 8.12.

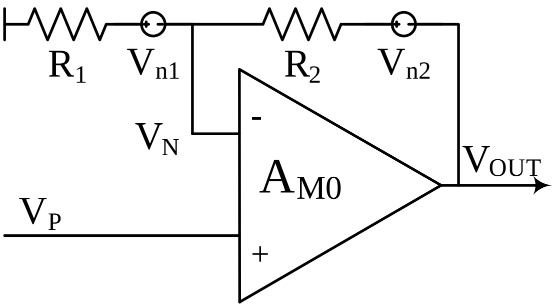

Figure 8.12: Resistors noise.

The closed loop gain G of the main amplifier is defined by the feedback resistors R_1 and R_2.

$$\frac{V_{OUT}}{V_P} = 1 + \frac{R_2}{R_1} = G \tag{8.12}$$

If the amplifier is ideal in the signal baseband, the output voltage is:

$$V_{OUT} = GV_P + (G-1)V_{n1} + V_{n2} \tag{8.13}$$

The output noise spectral density is, with a standard $4kTR$ model for the NSD of the resistors:

$$\begin{aligned} NSD_{R,OUT} &= (G-1)^2 4kTR_1 + 4kTR_2 \\ &= 4kTR(G-1)\left[\frac{V^2}{Hz}\right] \end{aligned} \tag{8.14}$$

where R is the sum of the two resistances R_1 and R_2.

In the signal baseband, an open-loop gain much larger than the closed loop gain G can be assumed. The input referred noise spectral density then becomes:

$$NSD_{R,IN,baseband} \cong 4kTR\frac{G-1}{G^2}\left[\frac{V^2}{Hz}\right] \tag{8.15}$$

8.2.6 Common-mode control circuits

The circuits to control the output common-mode of the gain stage and of the integrator are standard implementations with switched capacitors. The contribution to the noise level of such output common-mode control is negligible. This subsection is thus focused on the implementation of the virtual ground common-mode control of both gain stage and integrator. Such circuits are required as the selected fully differential implementation 8.6 does not emulate DC resistors between the input of the amplifier and a common-mode potential.

CM - Gain stage

The input common-mode of the gain stage is set using the structure detailed in Fig. 8.13. The proposed topology sets the input common-mode to V_{cm} and forces the DC gain to zero.

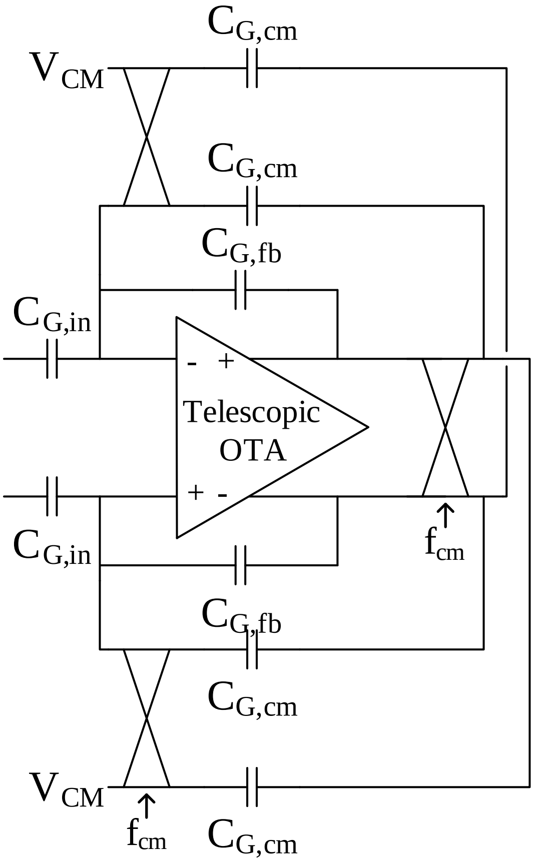

Figure 8.13: Gain stage input CMFB.

The switched-capacitors are modifying the gain stage transfer function to a high-pass filter with a cut-off frequency equal to:

$$f_{cut-off} = \frac{1}{2\pi R_{G,cm} C} = \frac{C_{G,cm}}{\pi C_{G,fb}} f_{cm} \tag{8.16}$$

where f_{cm} is the CMFB chopper frequency.

In order not to deteriorate the gain around the working frequency f_{ck}, the capacitors

$C_{G,cm}$ should be much lower than the feedback capacitors $C_{G,fb}$.

The noise related to the input common-mode capacitors is created, sampling the capacitors $C_{G,cm}$ in a modulated environment (gain stage), before demodulation and sampling (integrator). The noise analysis is decomposed in several sub-steps even if multiple operations are performed by a single component (e.g. the demodulation and the sampling are performed by a unique set of switches at the input of the integrator).

The mathematical steps are the following:

- A) Compute the RMS noise and the associated NSD without any (de)modulation
- B) Compute the demodulated noise
- C) Compute the sampled noise

A) The noise is first sampled on the capacitor $C_{G,cm}$:

$$N_{RMS}^2 = \int_0^{\infty} 4kTR_{sw} \left| H(j2\pi f) \right|^2 df = \frac{kT}{C_{G,cm}} \left[V^2 \right] \tag{8.17}$$

As the capacitors $C_{G,cm}$ are holding the value during half of the period, the input-referred noise spectral density of this noise is:

$$\begin{aligned} NSD_{IN,mod} &= \frac{kTC_{G,cm}}{C_{G,fb}^2} \frac{\pi}{2f_{cm}} \left(\frac{\sin(f\pi/2f_{cm})}{f\pi/2f_{cm}} \right)^2 \\ &= \frac{kTC_{G,cm}}{C_{G,fb}^2} \frac{\pi}{2f_{cm}} \mathrm{sinc}^2(f\pi/2f_{cm}) \left[\frac{V^2}{Hz} \right] \end{aligned} \tag{8.18}$$

Figure 8.14 shows the shape of the input-referred modulated noise. The displayed graphs are normalized with the chopping frequency f_{ck} (x axis) and with the noise power (y axis) to have an unity integrated noise power ($N_{RMS}^2 = 1$).

B) The noise is generated in a modulated environment (between the input chopper and the output chopper included in the integrator block). The demodulated noise is obtained multiplying the modulated one with the chopping function:

$$\begin{aligned} NSD_{IN,demod} = & \frac{4}{\pi^2} \frac{kTC_{G,cm}}{C_{G,fb}^2} \frac{\pi}{2f_{cm}} \sum_{k=0}^{\infty} \frac{1}{(2k+1)^2} \\ & \left(\mathrm{sinc}^2 \left(\frac{((2k+1)f_{ck} - f)\pi}{2f_{cm}} \right) + \right. \\ & \left. \mathrm{sinc}^2 \left(\frac{((2k+1)f_{ck} + f)\pi}{2f_{cm}} \right) \right) \end{aligned} \tag{8.19}$$

A graphical representation of (8.19) is displayed in Fig. 8.15.

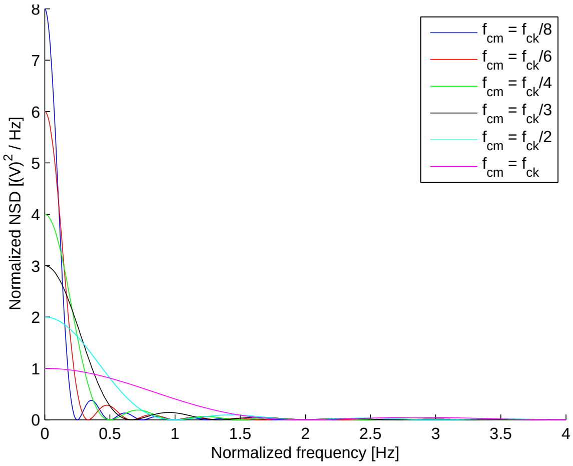

Figure 8.14: Input-referred modulated noise.

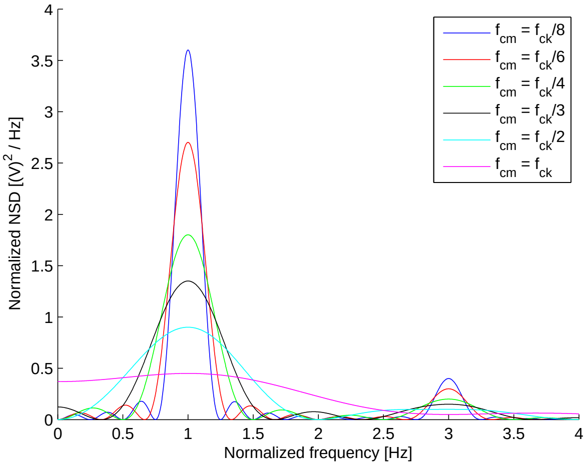

Figure 8.15: Input-referred demodulated noise.

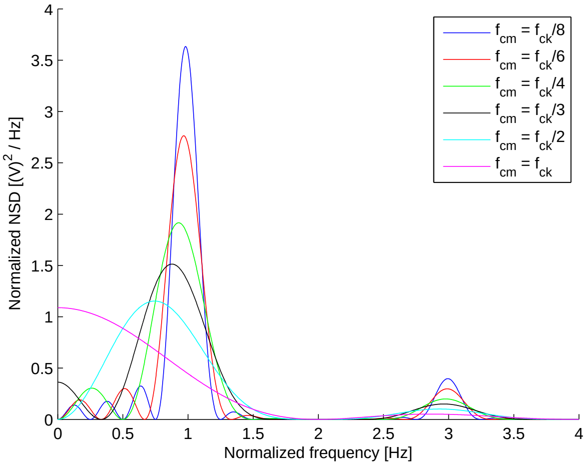

Figure 8.16: Input-referred noise contribution of the gain stage CMFB circuit.

C) The noise is finally sampled and held in the integrator. As the demodulation and the sampling are performed by the same set of switches at the input of the integrator, the sampling frequency is by construction twice the modulating one, $f_{sampling} = 2f_{ck}$.

$$\begin{aligned} NSD_{IN,SH} = & \frac{2kTC_{G,cm}}{\pi C_{G,fb}^2 f_{cm}} \mathrm{sinc}^2\left(\frac{f\pi}{2f_{ck}}\right) \sum_{k=0}^{\infty} \sum_{j=-\infty}^{\infty} \frac{1}{(2k+1)^2} \\ & \left(\mathrm{sinc}^2\left(\frac{((2k+1)f_{ck} - f + j2f_{ck})\pi}{2f_{cm}}\right) + \right. \\ & \left. \mathrm{sinc}^2\left(\frac{((2k+1)f_{ck} + f - j2f_{ck})\pi}{2f_{cm}}\right)\right) \end{aligned} \tag{8.20}$$

The noise contribution of the CMFB circuit (8.20) is displayed in Fig. 8.16. The parametric analysis in function of the common-mode frequency shows in particular that the f_{cm} frequency should be an even divider of the auto-zero frequency.
If the signal bandwidth is much lower than the chopping frequency, the most suitable value is half of the chopper frequency, $f_{cm} = f_{ck}/2$.

The $C_{G,cm}$ capacitors should also be minimized to reduce their contribution to the noise of the auto-zero amplifier. Compared to the global noise budget, the contribution of these capacitors can be made negligible.

CM - Integrator

The input common-mode of the integrator, Fig. 8.17, is set by a similar structure than the one for the gain stage, but the common-mode capacitors $C_{I,cm}$ are connected to the ground instead of a connection to the output nodes.

As in the gain stage, the noise is first sampled on the capacitors $C_{I,cm}$, and thus a continuous-time representation of the integrator can be derived, Fig. 8.18. As the noise is sampled, the noise power, $N_{RMS} = kT/C$, is folded into the choppers frequency band. The equivalent noise spectral density is thus, assuming an uniform distribution:

$$NSD_{V_{SNO}} = \frac{2kT}{C_{I,cm} f_{cm}} \tag{8.21}$$

The transfer function of the sampled noise source to the output of the integrator is:

$$H(j\omega) = \frac{V_X}{V_{SNO}} = \frac{2C_{I,cm} f_{cm}}{j\omega C_{I,fb}} \tag{8.22}$$

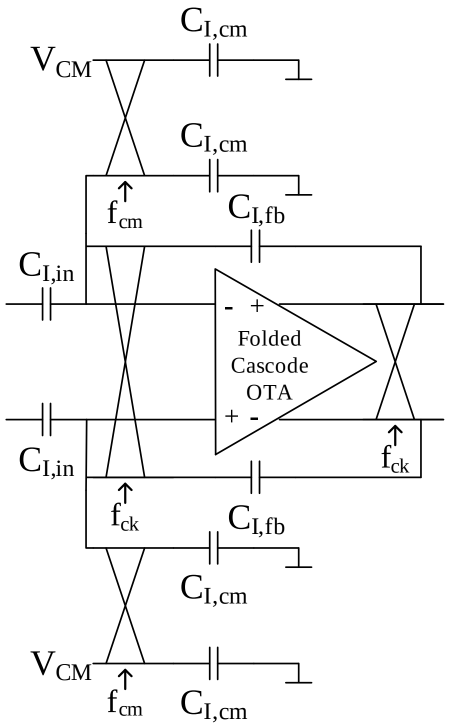

Figure 8.17: Integrator input CMFB.

and the output-referred noise spectral density is:

$$\begin{aligned} NSD_{OUT,I,CM} &= V_{SNO}\left|H(j2\pi f)\right|^2 \\ &= \frac{2kTC_{I,cm}f_{cm}}{(\pi f)^2 C_{I,fb}^2}\left[\frac{V^2}{Hz}\right] \end{aligned} \tag{8.23}$$

The input-referred noise spectral density is obtained, dividing the output-referred NSD by the transfer function of the integrator and by the gain of the first stage. In the signal baseband (around f_{ck}, as the signal is chopped), the input-referred noise spectral density is:

$$\begin{aligned} NSD_{IN,I,CM} &= NSD_{OUT,I,CM}\left(2\pi f R_{I,in} C_{I,fb}\right)^2 \frac{C_{G,fb}^2}{C_{G,in}^2} \\ &= \frac{2kTC_{I,cm}f_{cm}}{C_{I,fb}^2 f_{ck}^2}\frac{C_{G,fb}^2}{C_{G,in}^2}\left[\frac{V^2}{Hz}\right] \end{aligned} \tag{8.24}$$

This last equation shows that the noise contribution of the CMFB circuit of the integrator can be reduced without increasing the power consumption. The most straightforward way to minimize this noise contribution is to decrease the $C_{I,cm}$ capacitance.

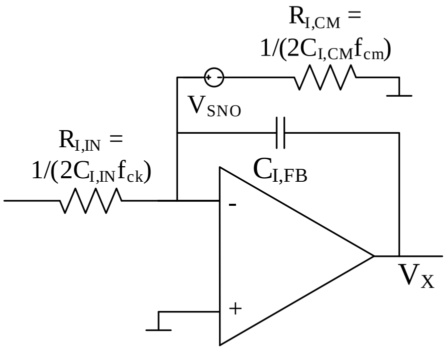

Figure 8.18: Simplified (single ended) continuous-time representation of the noise contribution of the CMFB circuit in the integrator.

8.2.7 Power Consumption

The previous analysis has shown that most of the power consumption should be used in the gain stage and in the feedback resistors. The output voltage of the main amplifier is proportional to the input signal and to the amplifier closed-loop gain. The full output range should be used to relax the requirements on a next stage (typically, an ADC). In order to explore the noise versus supply current trade-off, the current flowing into the resistors is evaluated here considering an output range covering half of the power supply voltage:

$$I_{R,max} = \frac{V_{DD,max}}{4R} \tag{8.25}$$

assuming a common-mode centered in the middle of the power supply. The input-referred noise spectral density of the resistors in the signal baseband is:

$$NSD_{R,IN,baseband} = \frac{4kTR(G-1)}{G^2} = \frac{kTV_{DD}}{I_R}\frac{G-1}{G^2} \tag{8.26}$$

This result, combined with the noise spectral density defined for the gain stage in section 8.2.3, equation (8.7), allows a sizing of the resistors and of the gain stage equalizing their respective noise-consumption ratio:

$$\frac{NSD_{R,IN,baseband}}{I_R} = \frac{NSD_{IN,gain\_stage}}{I_{0,gain\_stage}} \tag{8.27}$$

For the specific implementation simulated in the next section, a theoretical current consumption of $50-120\mu A$ is found for a noise spectral density of $75(nV)^2/Hz$. The noise contribution of the integrator (8.8) is $7(nV)^2/Hz$.

8.2.8 Noise Contributions - Summary

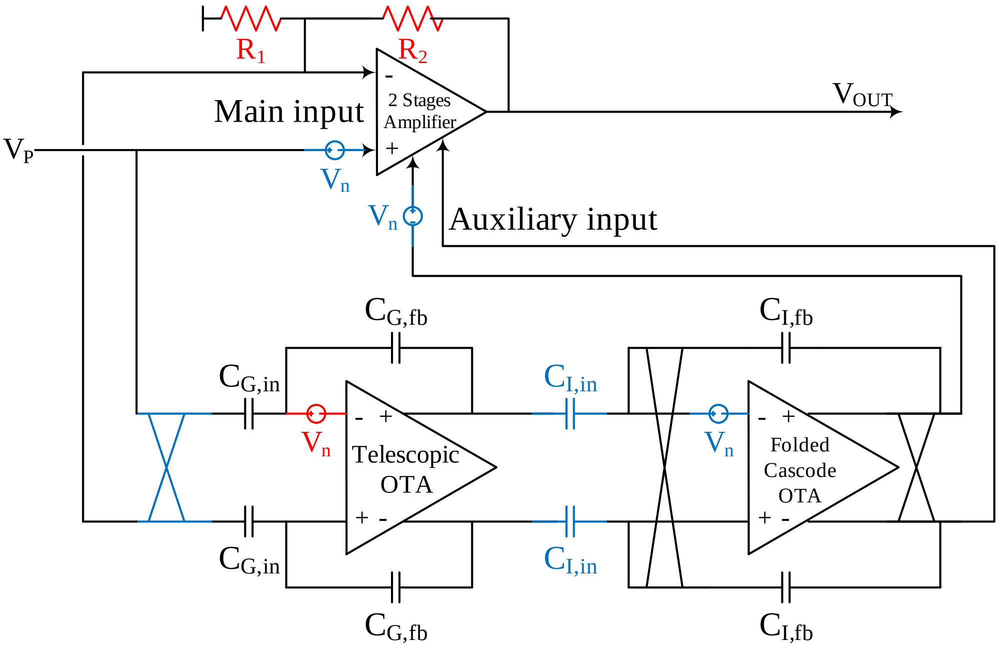

Figure 8.19: Auto-zero main noise sources.

To summarize, it is possible to sort the noise sources into two categories. The first ones, colored in red in Fig. 8.19 have a direct impact on the power consumption. These sources are:

- the thermal noise of the gain stage,
- the noise in the feedback resistors of the main amplifier.

The noise contribution of the second type of sources, in blue in Fig. 8.19, can be lowered without increasing the consumption:

- using an auxiliary chain for the main amplifier noise,
- lowering the on-resistance of the auxiliary chain input chopper,
- increasing the closed loop gain of the gain stage to reduce the sampled noise on the input capacitors $C_{I,in}$ of the integrator (a high gain in the first stage also lowers the thermal noise contribution of the integrator),
- using frequency modulation to shift the flicker noise of the amplifiers of the auxiliary chain outside the baseband.
- setting the frequency of the input common-mode control of the gain stage to half of the chopping frequency.
- minimizing the switched capacitors of the CMFB circuits in the gain stage and in the integrator.

Validation: Classically, AC simulations are used to check the noise level of a circuit. These simulations provide extremely rapidly a frequency response of a circuit with noise. AC simulations are based on the fact that the system to analyze is linear around a given operating point as long as the variation of the signal is not significant (small signal hypothesis). It is not possible to use such AC simulations to validate the auto-zero amplifier, because it uses a frequency shift and is non-linear.

Two different types of simulation are thus used to check the noise level. Firstly, AC simulations are used as much as possible inside the blocks of the auto-zero amplifier. It is possible to extract data on the noise level of the different blocks, as the low-frequency noise of the main amplifier or the noise level of the gain stage around the chopper modulation frequency.

Secondly, to simulate the full auto-zero amplifier, transient simulations are used. The noise contribution of each component (transistors and resistors) is added by the transistor level simulator. The spectrum of each generated noise source is continuous (the noise model used is not a finite sum of sine waves at discrete frequencies).

The noise spectral density in the baseband is extracted from the discrete Fourier transform of a noisy transient signal of length T_{sim}. As the front-end is expected to be followed by an analog to digital converter, the output noise is sampled to take aliasing into account.

$$NSD_{output} = FFT(V_{out(t),sampled})\frac{1}{\Delta f}\left[\frac{V^2}{Hz}\right] \tag{8.28}$$

The input-referred noise at the input of the auto-zero amplifier is given by:

$$NSD_{input} = FFT(V_{out(t),sampled})\frac{1}{G^2}\frac{1}{\Delta f}\left[\frac{V^2}{Hz}\right] \tag{8.29}$$

where Δf is the interval between each spectral line of the discrete Fourier transform. The spectral resolution is given by the inverse of the simulation time: $\Delta f = 1/T_{sim}$.

Table 8.1 shows the estimated and simulated noise levels in the baseband frequency of the auto-zero amplifier and provides a comparison with the noise levels of the main amplifier and of the gain stage of the auxiliary chain.

As expected, the flicker noise level is much lower than the thermal noise. The 1/f noise contribution to the global noise budget (Table 8.1) is negligible. The power consumption is higher than predicted, as the supply currents of the main amplifier, of the integrator and of the bias generation were not taken into account in the simplified analysis.

Table 8.1: Auto-zero amplifier: noise summary and key characteristics

Noise level	Theory	Simulated	Unit
Flicker	-	< 1 at 12 Hz	$(nV)^2/Hz$
Thermal	75	180 at 12 Hz	$(nV)^2/Hz$
Main Amplifier at 100Hz	-	> 60k	$(nV)^2/Hz$
Gain Stage at f_{ck}	45	47	$(nV)^2/Hz$
Power consumption	50	85	μA
Sensor output current	3	-	pA

8.3 Common-mode rejection

Instrumentation amplifiers can achieve large differential gains with high common-mode rejection and large input impedance. The classic three-op-amp configuration (Fig. 8.20) amplifies the differential signal in the first stage and rejects the common-mode in the differentiator. The gain of instrumentation amplifiers is usually set by the user, either by hardware, changing an external resistance ($2R_{in}$ in Fig. 8.20), or by software, if the instrumentation amplifier is fully integrated.

The classic implementation can achieve a good amplification of the differential signal while the input common-mode is far from the power supplies. The differential gain has to be limited if the input common-mode is important, in order not to saturate the outputs of the first stage [70]. Indeed, after the first stage, the internal nodes are given by (assuming ideal amplifiers):

$$V_{outp} = V_{in,diff}\frac{G}{2} + V_{in,cm} \tag{8.30}$$

$$V_{outn} = -V_{in,diff}\frac{G}{2} + V_{in,cm} \tag{8.31}$$

where $G = (1 + R_{fb}/R_{in})$ is the first stage differential gain and $V_{in,diff}$ and $V_{in,cm}$ are the differential-mode and the common-mode of the input signal. The saturation of the first stage in the presence of an input common-mode close to the power supply is illustrated in Fig. 8.21.

A first approach to get around this problem is to reduce the gain of the first stage while increasing that of the differentiator [71]. This technique is not suitable for low-noise low-power circuits as the constraints on the second stage are much more important without prior amplification.

Shifting the input range may also be achieved [72]; however, this technique is only

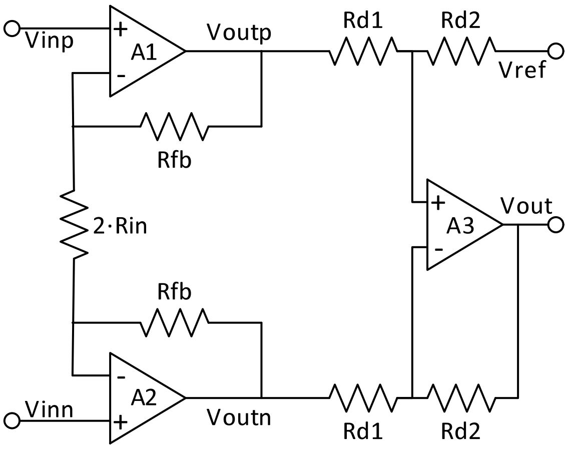

Figure 8.20: Classic three-op-amp instrumentation amplifier.

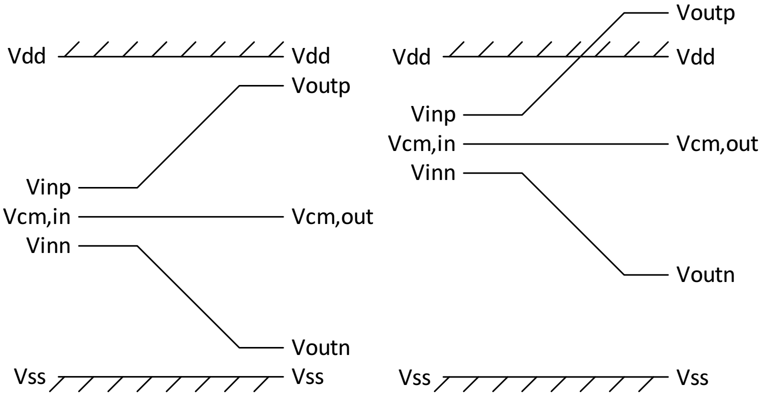

Figure 8.21: Amplification of input signal with different common-mode voltages. Left: Common-mode centered. Right: Common-mode close to V_{dd}.

appropriate for constant input common-mode and it is not possible for the input to go close to both V_{dd} and V_{ss} while ensuring a high differential gain.

8.3.1 Proposed solution

The proposed solution to control the common-mode at the output of the first stage of the instrumentation amplifier is shown in Fig. 8.22. The output common mode is set, without altering the differential characteristics of the amplifier, the central resistor is split into two half-sized resistors and the central node is driven by an extra amplifier.

The voltage control of the central node V_{mid}, as well as the required feedback circuitry, are displayed on Fig. 8.22. The transfer functions for the differential-mode and for the

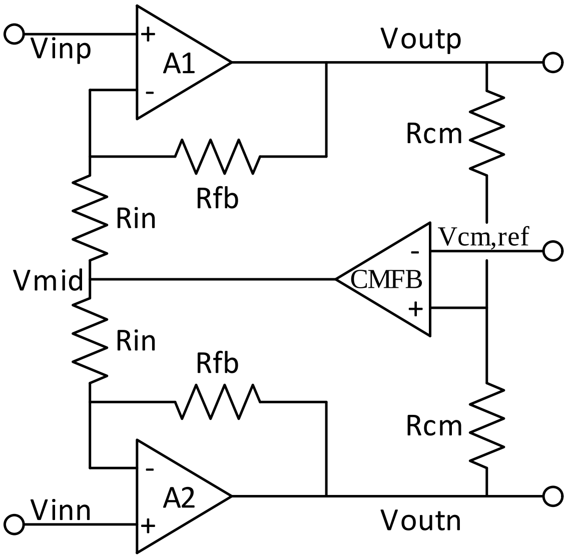

Figure 8.22: Instrumentation amplifier with a controlled common-mode

common-mode are:

$$V_{out,diff} = V_{in,diff}\left(1+\frac{R_{fb}}{R_{in}}\right) = V_{in,diff}G \tag{8.32}$$

$$V_{out,cm} = -V_{mid}(G-1) + V_{in,cm}G \tag{8.33}$$

The differential transfer function (8.32) is identical to that of the classic instrumentation amplifier (8.30 and 8.31), while the output common-mode is alterable, varying the V_{mid} voltage. A graphical representation of the new voltages is shown in Fig. 8.23.

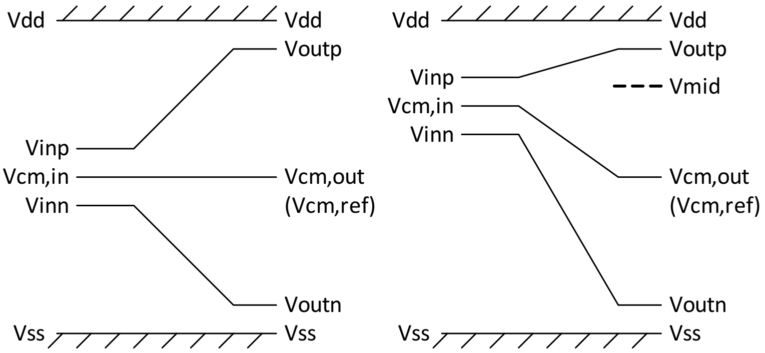

Figure 8.23: Amplification of input signal with different common-mode voltages using the output common-mode rejection circuit. Left: Input common-mode equal to the reference common-mode. Right: Input common-mode close to V_{dd}

8.3.2 Nonideal effects

Vmid saturation

In a closed-loop configuration, the $V_{cm,out}$ potential is set to $V_{cm,ref}$. The V_{mid} potential is (derived from (8.33)):

$$V_{mid} = -V_{cm,ref}\frac{1}{G-1} + V_{in,cm}\frac{G}{G-1} \tag{8.34}$$

If the input common-mode is very close to the power supplies and if a small differential gain G is set, the required potential V_{mid} which compensates perfectly the output common-mode may be located beyond the supplies. In practice, the common-mode feedback amplifier (Fig. 8.22) is saturated.

The common-mode rejection is incomplete, but still better than without the compensation. To lessen the V_{mid} saturation effect, the feedback amplifier should have a large output voltage range. The differential gain is not affected by the saturation of V_{mid}.

Figure 8.24 illustrates the relationship between the input common-mode and the saturation of the CMFB amplifier. It shows the input common-mode range for which the CMFB amplifier is not saturated. Theses limits are obtained considering an ideal CMFB amplifier (saturated at V_{dd} and V_{ss}) and a desired common-mode equal to the half of the power supply. This graph shows in particular that for a gain of 8, the output common-mode is centered on half of the power supply only if the input common-mode is within the 10% - 90% range. Beyond these limits, the control of the output common-mode is only partial due to the saturation of the CMFB amplifier.

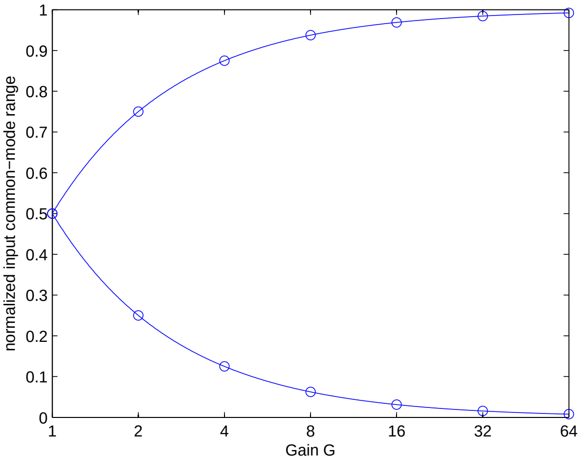

Figure 8.24: Input common-mode range.

Non-ideal feedback

The imprecision of the common-mode feedback is not altering the performance of the instrumentation amplifier. The output common-mode is, assuming a feedback amplifier having an offset V_{off} and a finite DC gain A and a mismatch of the resistors measuring the output common-mode of $R_{cm} \pm \Delta R_{cm}$:

$$V_{out,cm} = \frac{V_{cm,ref} - V_{off} - V_{in,diff} G \frac{\Delta R_{cm}}{2R_{cm}} + V_{in,cm} \frac{1}{A} \frac{G}{G-1}}{1 + \frac{1}{A(G-1)}} \tag{8.35}$$

The only aim of the feedback circuitry is to shift the output common-mode voltage to a centered value, in order not to saturate the auto-zero amplifiers. A feedback circuit with low static power consumption is appropriate as the output common-mode voltage does not need to be set precisely.

Mismatch of resistors

The classic instrumentation amplifier has a single input resistance $2R_{in}$. The division of this resistor into two distinct components R_{in} introduces a possible mismatch between them. A relative mismatch of $\pm\Delta R_{in}/R_{in}$ (i.e. the respective values of the top and bottom input resistors are $R_{in} + \Delta R_{in}$ and $R_{in} - \Delta R_{in}$) leads to the following transfer function, assuming ideal amplifiers:

$$V_{out,diff} = V_{in,diff} G + \left(V_{in,cm} - V_{cm,ref}\right) \frac{2\Delta R_{in}}{R_{in}} \tag{8.36}$$

The injection of the common-mode into the differential mode limits the precision of the instrumentation amplifier if the input differential signal is very small compared to the variation of the input common-mode. The classic two stages op-amp (Fig. 8.20) is similarly limited by the mismatch of the resistors in the output difference amplifier.

8.3.3 Implementation

The main constraints on the amplifier of the common-mode rejection are:

- Wide output range
- Resistive load
- Low static power consumption

A class AB amplifier is selected. The first stage is made of a simple OTA, the second stage being similar than the one used for the main amplifier in the auto-zero amplifier.

Typical specifications for this amplifier are a DC gain of $60dB$, a bandwidth of approximately $100Hz$, a wide input range $Gnd + 50mV/Vdd - 50mV$ and a low power consumption, $2\mu A$.

8.4 Output buffers

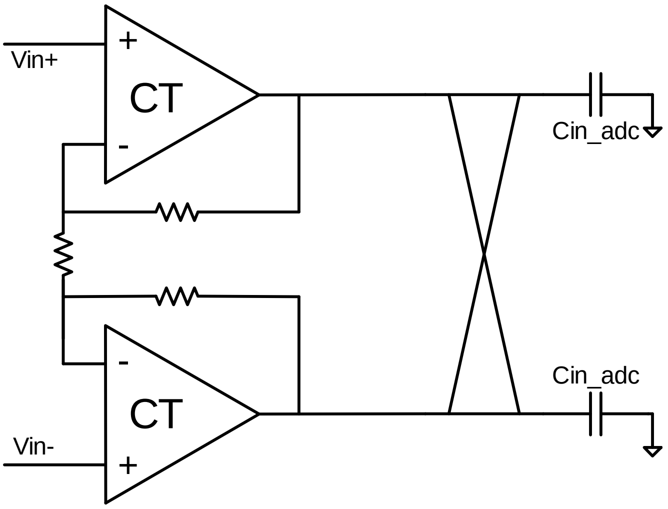

Figure 8.25: Equivalent output load, including the input capacitors of the ADC.

The continuous-time PGA is loaded by the switched capacitors of the $\Sigma\Delta$ modulator $C_{in,adc}$. Switching these capacitors introduces strong variation on the PGA output voltage and a large transient current. A representation of the equivalent load on the PGA is displayed in Fig. 8.25.

As these perturbations could couple with the PGA input or push temporally the PGA out of its linear regime, it is necessary to lessen them. Two approaches are discussed in this section to reduce the transient current delivered by the PGA. A passive implementation diluting the perturbation over time is first introduced, followed by an active solution injecting most of the charge to the load in an extra transient phase.

8.4.1 Passive resistors

A possible solution to limit the current and to time spread the perturbation is to add serial resistors at the output of the PGA, Fig. 8.26. As these resistors contribute to the thermal noise and are only useful during the transient, they are bypassed for the end of the charge transfer of the ADC input capacitors (i.e. end of the settling).

The test of this solution with a transistor-level simulator showed that the simple use of extra resistors is insufficient to keep the main amplifier into its linear regime. The

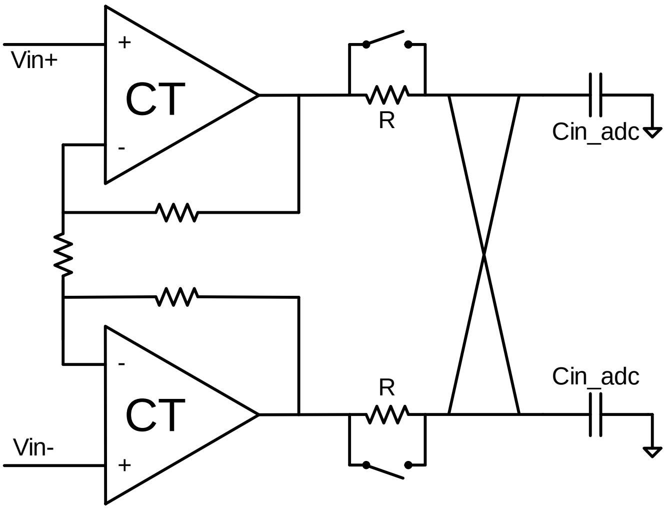

Figure 8.26: PGA output current limitation with serial resistors.

load effect created a gain error up to 0.5% and this error is not significantly reduced with the output resistors.

8.4.2 Active buffers

As the use of a passive solution is not efficient, two output buffers are added, Fig. 8.27. The aim of these buffers is to provide the high current during the transient phase after the swap of the loads. The buffers are only connected during a portion of the integrating phase, providing most of the energy to the load capacitors.

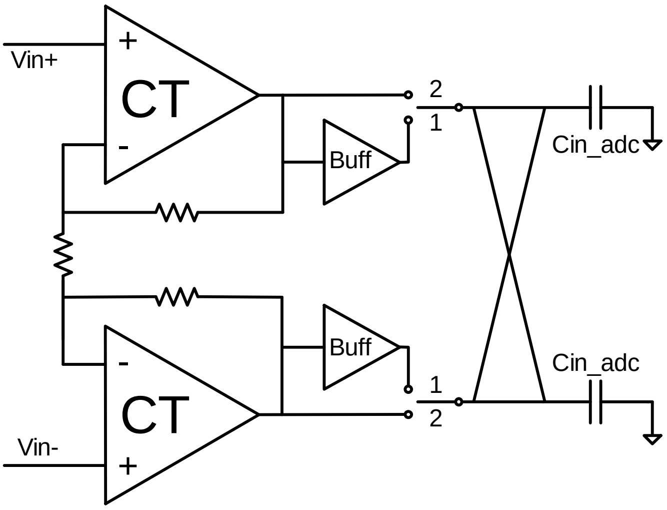

Figure 8.27: PGA output structure using buffers.

As the buffers are disconnected for the end of the charge transfer, it is not necessary to have neither a good linearity nor a low noise level. The static power consumption can thus be minimized. The implementation of the buffer is shown in Fig. 8.28

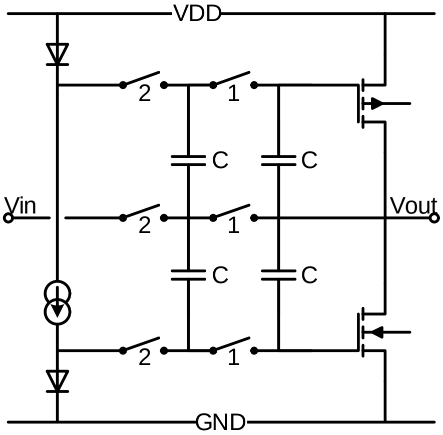

Figure 8.28: Output buffer.

In order to test the efficiency of the output buffer, the duty cycle of the active phase is programmable between 3% and 50%. It is also possible to completely disable the output buffer for test purposes. In this case, the buffers are bypassed and the auto-zero amplifiers are directly connected to the modulator.

8.5 Conclusion

Pre-amplifiers benefit to the acquisition of signal with limited output voltage. A programmable gain stage amplifies the input signal up to the input range of the ADC to get a maximal resolution.

A continuous-time structure was selected to reduce the power consumption and to limit the silicon area. Required input capacitors to reach an input-referred noise spectral density below $20nV/\sqrt{Hz}$ would have been too large.

The main contributions of this chapter are:

- Optimization of the power consumption: A new topology for low-noise low power continuous-time instrumentation amplifiers was introduced. Both auto-zeroing and chopper techniques were exploited to define an architecture with a symmetrical behavior in both phases, reducing constraints on amplifiers. The noise contribution of each block was analyzed in order to optimize both noise level and power consumption.
- Control of the output common-mode: The proposed topology, well suited for fully integrated circuits, solves the problem of the first stage saturation for large values of the input common-mode. The feedback circuit controls the output

common-mode within the first stage and, the second stage being unnecessary, a differential output is provided. The analysis of the non-ideal effects of the proposed architecture showed that the most critical matching is on the input resistors, while the instrumentation amplifier is not sensitive to the performance of the feedback circuit.

- Ripple limitation: Switched loads are a critical issue of continuous-time amplifiers as the transient current may temporarily push amplifiers out of their linear regime. The proposed extra buffers, with low static power consumption, provide most of the charges to the ADC and thus reduce the constraints of the continuous-time amplifier.

9 Characterization

The circuit designed in the previous chapters has been implemented in a CMOS 180nm process. The Fig. 9.1 shows the layout of the full circuit and of the acquisition chain.

Most of the silicon area is covered by the 1.8V digital that includes not only the filters for the ADC, but also a microcontroller, memories and all the interfaces to communicate with the external environment using standard protocols as UART, I2C, SPI and JTAG. Most of the pads are digital I/Os to communicate with the microcontroller.

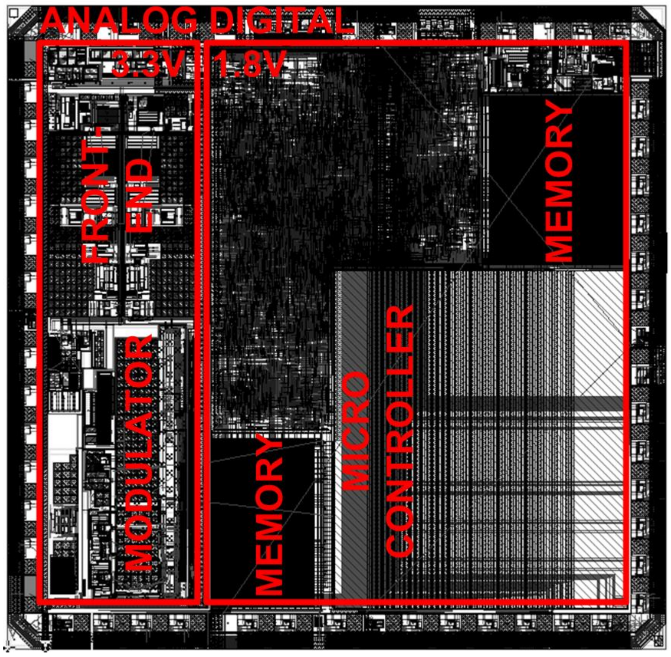

Figure 9.1: Layout of the integrated circuit.

The acquisition chain, Fig. 9.2 is mostly occupied by the PGA (40%) and by the ADC (50%). The remaining part is dedicated to services blocks such as input multiplexer,

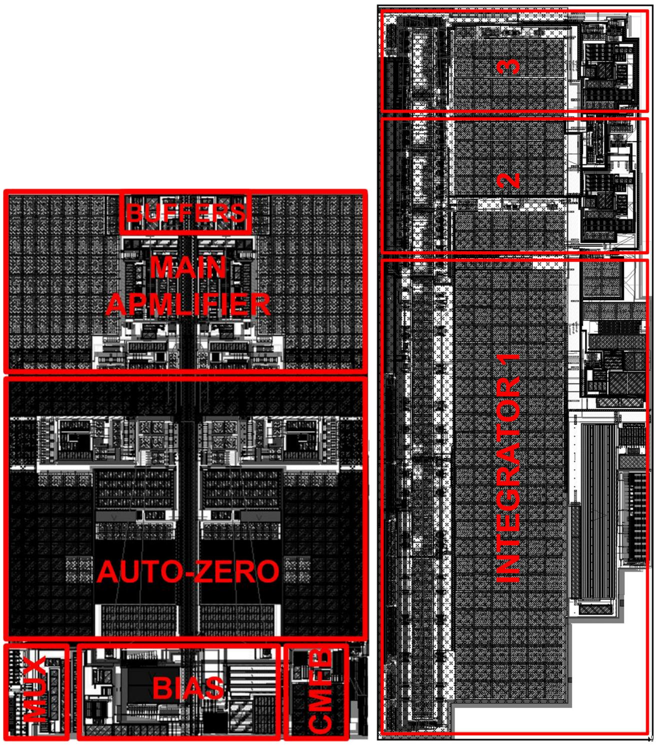

Figure 9.2: Layout of the front-end (left) and of the modulator (right).

current and voltage references and charge pump.

The first integrator of the modulator is clearly the largest one, in order to reduce the noise level of the converter. A large portion of the PGA is covered by the capacitors of the auxiliary path of the auto-zero amplifier. The largest OTA of the PGA is the gain stage amplifier of the auxiliary path, followed by the main amplifier. The relative size of the output buffers is negligible.

The characterization starts with the functional tests of the ADC, including the modulator and the filters. The limitations of the PGA are then evaluated through measurements of the whole acquisition chain. The performances of the circuit are finally compared to published work in the last section.

9.1 Functional tests - ADC

The first step in the characterization of an integrated circuit is the test of the functionality of the system. The known limitations and bugs of the ADC are briefly listed hereafter.

The likely cause of each observed problem is stated and, whenever possible with the chip programming, a solution is expressed.

Digital filters: The first problem identified is located in the selection of the digital filter. Whatever the selected topology of the modulator, second- or third-order, the connected filter is the one for the third-order architecture. As the size of the registers as well as the applied normalization function are completely different, it is definitively not possible to use the second-order modulator.
The following functional tests, as well as the evaluation of the circuit performance, are thus done using the third-order architecture.

Flicker noise: The second limitation is the expected poor noise performances of the modulator in the ΣΔ configuration. As no flicker noise rejection was implemented in this mode, the performances are far worse than the one of the incremental configuration.

Reset phase: An unexpected memory effect between two successive conversions limits the resolution of the converter. Measurements showed that the error is related to the last bit of the previous conversion and to the duration of the reset phase. As a consequence, the output distribution is not a typical Gaussian but a bi- or multi-polar distribution as displayed in Fig. 9.3. The width of the peaks Fig. 9.3 are noise related while the spacing between the peaks is linked to the poor reset. The ΣΔ mode is obviously not affected by this limitation.

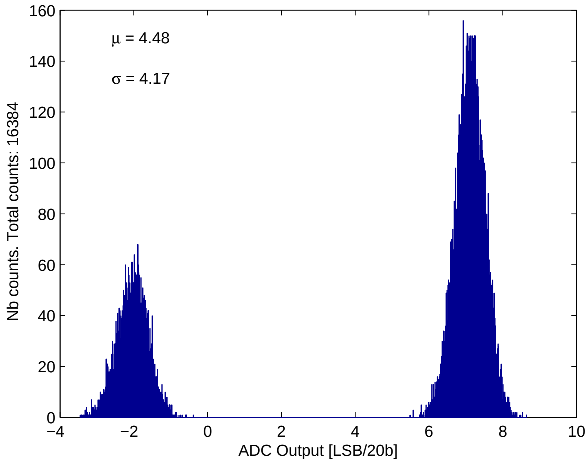

Figure 9.3: Measured noise distribution caused by a reset phase too short.

A workaround using the embedded microcontroller was implemented to add extra reset phases between each conversion. The extra phases lengthened the conversion time of less than 1%. While extending the reset phase, the multiple peaks converge to a single one with a Gaussian distribution, as shown in Fig. 9.4.

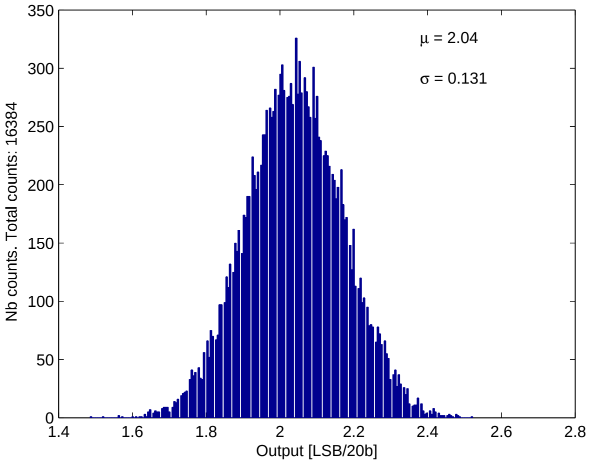

Figure 9.4: Measured noise distribution of the ADC with an extended reset phase.

Parasitic coupling: The last observable limitation is a central spike in the integral non-linearity, Fig. 9.5, while using the switching sequence with the lowest dynamic consumption (see Chapter 4, architecture V). The hybrid IV-V architecture[1], increasing slightly the power consumption of the reference branch, does not suffer of this degradation of the INL. The central spike is most probably due to a mismatch between parasitic capacitors.

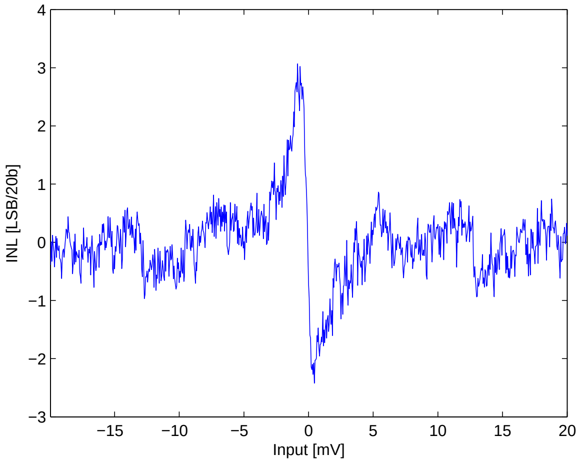

Figure 9.5: Measured INL of the topology #5 around mid-scale.

---

[1] The numbers IV and V refer to the implementation indexing defined in the comparison of the implementations, chapter 4. In both IV and V implementations, the inputs are swapped with choppers to perform the charge transfer. The fifth architecture uses a lower frequency control signal for the switches connected to the virtual ground of the amplifier. The called hybrid IV-V structure implements a type IV input for the reference and a type V input for the signal.

As suggested in the chapter 7, a possible solution to reduce the impact of the parasitic coupling in the type V structure would be to modify the end of the charge transfer as shown in Fig. 9.6. The switches connected to the amplifier are opened before the crossed switches named ’1A’, ’1B’ and ’2*’ to guarantee that the parasitic capacitances of these last switches are not adding any charges to the integration capacitors.

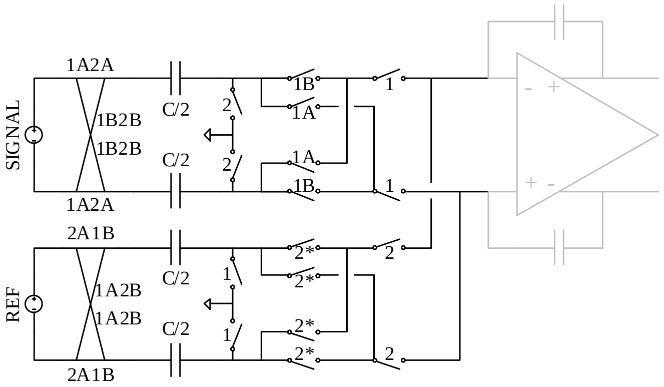

Figure 9.6: Modified version of the topology #5. The four extra switches, named ’1’ and ’2’, are the ones connected to the virtual ground of the amplifier.

Power consumption: The nominal power consumption of the circuit is slightly higher than the nominal expected value. Process variations may induce biasing shift, especially in an untrimmed circuit, such as the implemented prototype. The power consumption of the modulator was corrected, adjusting the bias current of each analog block independently.

9.2 Functional tests - PGA

Once the modulator and the filters were tested and validated independently, the whole acquisition chain could be tested. The gain selection is correct and the CMBF amplifier is effectively controlling the output common-mode of the front-end.

Output buffers: The main limitation of the implemented front-end is an insufficient driving capability of the output buffers. The cause was verified modifying the length of the active period of the output buffers. The consequences of this are manifold:

- The first obvious consequence is an improper slew rate of the PGA, creating charge transfer errors in the ADC.
- Distortion is added in the front-end static transfer function. The measured output non-linearity in a typical configuration (i.e. 64 Sps) is around 50ppm - while a few ppm were expected. The distortion is reduced for lower sampling rates.

- Ripple is added in the PGA, degrading strongly the noise performances. The ripple may be reduced, decreasing the operating frequency, but this first reduces the sample rate and then increases the noise level. Indeed, the flicker noise rejection is less efficient if the auto-zero operates at lower frequency.

Auto-Zero: Despite the previous limitations, the novel fully differential structure for the rejection of the offset and 1/f noise could be validated. Fig. 9.7 compares the output noise and offset of the front-end with and without the Auto-Zeroing. As the PGA was measured through the implemented ADC, the plots contains the noise and offset contribution of both blocks.

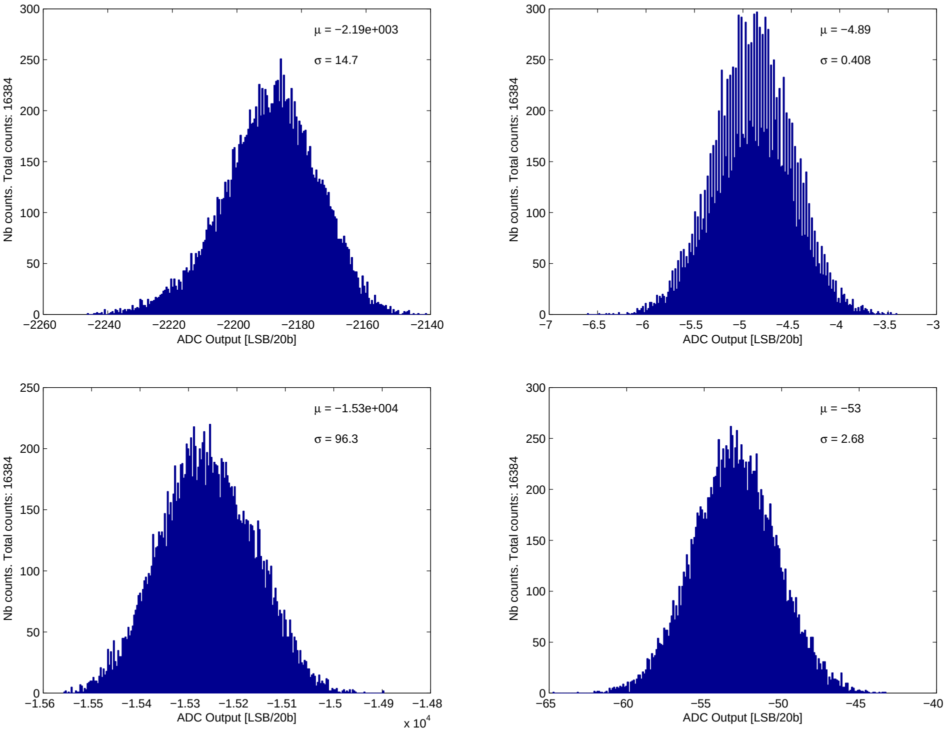

Figure 9.7: Output noise distribution of the PGA without the Auto-Zero (left) and with the auto-zero enabled (right). The upper plots show the flicker noise rejection with a gain of 8, while a gain of 64 was set in the lower plots.

Power Consumption: The typical power consumption of the front-end is around 200μA with shorted inputs. This value is, as the one of the modulator, slightly higher than the nominal expected value. The power consumption of both blocks (ADC and PGA) is correlated as the same biasing current generator is shared.

9.3 Performances

The measured ADC without the front-end reaches a resolution of 20.2 bits for a sample rate of 64Sps and a power consumption of 120μA [2]. The Effective Number Of Bits is given by:

$$ENOB = log_2\left(\frac{FullScale}{\sqrt{12}Noise_{RMS}}\right) \tag{9.1}$$

The correspondence of the ENOB and the Signal over Noise Ratio is:

$$ENOB = \frac{SNR_{max} - 1.76}{6.02} \tag{9.2}$$

One of the most commonly used Figure Of Merit to compare high-resolution converters is derived from Schreier's FOM [42].

$$FOM = DR_{dB} + 10log\left(\frac{BW}{P}\right) = SNR_{max} + 10log\left(\frac{1}{P2T_{conv}}\right) \tag{9.3}$$

where P is the power consumption, DR is the dynamic range, BW the bandwidth of the ADC and T_{conv} the conversion period.

Table 9.1 summarizes some of the most relevant publications of high- to ultra-high-resolution converters.

The order of the architectures is mainly related to the desired resolution and to the sampling rate. While all selected architectures below 20 bits are second-order, the order of the architectures with a higher resolution is between 3 and 5, depending on the requested bandwidth.

The distribution between the feedback and feedforward structures is similar and almost constant over years. All architectures in table 9.1 besides this work are single-loop topologies.

Multibit quantizers are implemented in a large majority of topologies. Except [78], all converters beyond 20 bits use single-bit quantization, in order to guarantee a better linearity.

The most common implementation with switched capacitors is the structure #1. In this configuration, the same capacitors are connected alternatively to the signal and to the reference. This structure, if combined with a multibit implementation, probably

---

[2] Power consumption of the modulator only. The power consumption of the digital filters could not be measured as the filters are included in the synthesized digital system, which includes the microcontroller and the system I/Os.

	Year	Architecture	Implementation	ENOB	Techno	Supply	Power	BW	SNR	FOM
					μm	V	μW	Hz	dB	dB
[73]	2009	2nd order - $\Sigma\Delta$ Feedforward - 4 bits	#1	16.3	0.18	0.7	870	25k	100	174.6
[74]	2007	2nd order - $\Sigma\Delta$ Feedback - 2 bits	-	17	0.18	3.3	14700	31.3k	104.1	167.4
[75]	2013	2nd order - Incremental Feedforward - 3 bits	#1	17	0.18	3.3	280	5k	105	177.5
[39]	2008	2nd order - $\Sigma\Delta$ Feedback - 4 bits	#2	17.2	0.13	3.3	9900	20k	105.5	168.5
[76]	1997	2nd order - $\Sigma\Delta$ Feedback - 3 bits	#1	19	2	5	2180	400	116.1	168.8
[77]	2006	3rd order - Incremental Feedforward - Single bit	#1	20.1	0.6	5	600	7.5	123	166.4
[78]	2008	5th order - $\Sigma\Delta$ Feedforward - 5 bits	#4	20.3	0.35	5	330000	20k	124	171.8
[79]	1994	4th order - $\Sigma\Delta$ Feedforward - Single bit	#2 - #4	21	3	10	25000	400	128.2	170.2
This work		MASH 2-1 - Incremental Feedback - Single bit	#5 - #4	20.2	0.18	3.3	360	32	123.1	172.3

Table 9.1: Performance comparison.

provides the best solution to reduce the power consumption of the amplifiers. Indeed, only the difference between the input signal and the reference is integrated.

The measured performances of the implemented circuit are comparable to state of the art comparators. The best FOM were reported in citations [73] and [75], followed by this work.

Nevertheless, one should consider that these two publications targeted much lower resolutions, of 16.3 and 17 bits respectively. The power consumption was optimized, combining a multibit quantizer with the switched capacitor implementation #1. The dynamic element matching in the multibit topologies significantly reduces the error induced by the mismatch of the DAC capacitors. As the compensation is not perfect, a residual degradation remains. The Signal-to-Noise-and-Distortion-Ratio is below the SNR by 5dB in [73] and by 13dB in [75].

Unlike the other architectures, a programmable gain of the modulator is available in this work. A major consequence of this design constraint is the need to separate the signal and reference paths with distinct sampling capacitors.

The combination of these two points - linearity and programmable gain - discarded the assembly of a multibit quantizer with the topology #1, as it was not suited at all for this specific design.

It should also be pointed that the MASH 2-1 topology used to characterize the proposed circuit in this work is not optimized for a 20 bits resolution. The expected results for the suited second-order architecture are a reduction of the power consumption of 15% and a lowering of the thermal noise of 13%, resulting in an improvement of the FOM of 2dB.

9.4 Conclusions

A complete design was carried out in a CMOS technology. The design methodology developed in this thesis allowed the realization of a prototype reaching the desired specifications at first integration. The flexibility of the modulator, implemented in the most sensitive parts, allowed a toggling to a slightly less efficient converter, but fully operational.

The resulting performances, resolution, power consumption and speed, were compared to the best published solutions. The comparison highlighted that two different topologies achieve high performance. On one side, the multi-bit topologies with shared input capacitors between the signal and the reference are perfectly suited for medium- to high-resolution converters and, on the other side, single-bit architectures with multiple inputs are better suited for ultra-high resolution converters. The proposed converter achieves the best performance in the latter category.

The observed limitations show an overview of the next generation converters, targeting more than 20 bits. The common-mode rejection is the first limitation. The static transfer function is impacted by the voltage non-linearity of the capacitors and by the asymmetrical clock feedthrough, even in a fully differential implementation. Secondly, the perturbation from the digital, and particularly from the clocks driving the modulator, must be taken into account.

10 Conclusion

The design of any A/D data converter is focused on three main characteristics: the resolution, the sample rate and the power consumption.

The linearity is the main challenge of high-resolution converters; it is altered by most imperfections of base components. As transistors level simulators are too slow to validate this characteristic, models have to be used in high-level views.

This work sought to enhance the design flow of ultra-high resolution ΣΔ ADCs with the modeling, simulation and design of switched capacitors circuits.

10.1 Main contributions

The key contributions of this thesis are:

- A comparative analysis of the input stage of the ADC. Several implementation structures of the ADC with switched capacitors were discussed. The noise and power consumption of each topology, as well as the sensitivity to non-ideal effect, were evaluated to help the designer to select the suitable structure.
- Modelling of the switched capacitors integrator. The selected level of modelling allows evaluating the impact of non-ideal effects in the integrator and to take into account the common-mode and the sequence of the digital signals driving the switches. The mathematical complexity of each model was estimated and, whenever possible, minimized. The model of the integrator is identical, whatever sequence of the phases and the number of inputs, to guarantee the compatibility with various architectures.
- A simulator of switched capacitors ADCs. Two simulators are included in a single software. The first one, based on the integrator, allows simulating most of the currently known non-ideal effects, while the second one is a linear simulator to account for the interactions between signals of different blocks. This

simulator is employed to define the sub-blocks specifications, to optimize the implementation with switches and capacitors, and to validate the switching sequences.
- Evaluation of the parasitic capacitors: the developed tool allows first to identify sensitive coupling and secondly to check the layout, importing in the simulator a list of parasitic capacitors.
- Filters. The study carried out showed that the common counter filters, based on a linear view of the modulator, are suboptimal. New filter were developed to reduce the quantization noise of the ADC. The advantages of such filters are however balanced with the required computing power, especially if the thermal noise strongly dominates.
- Front-end. The proposed architecture to control the output common-mode solves the saturation problem of internal nodes of the classic three-opamps instrumentation amplifier. A detailed noise analysis in auto-zero amplifiers allowed to define and to optimize a new fully differential cancellation path.
- Design and test. A full acquisition chain, including a programmable gain amplifier, an incremental ΣΔ modulator and the digital filters, was implemented in a 0.18μm CMOS technology. The test chip allowed to validate the simulated results and confirmed that the linearity of ADCs is influenced by the implementation with switched capacitors and also by the common-mode management.

10.2 Future perspectives

- It has been shown that the common-mode of the input signal limits the resolution of the converters. The performances are degraded due to asymmetrical charge injection and to non-linearity of capacitors. It would be valuable to control the common-mode or compensate the undesired effects with a low power consumption circuit. In this perspective, a control of the ADC constant voltage (ADC common-mode) used in the pre-charge of the input capacitors may be considered.
- Deep submicron technologies are more and more used to reduce the power consumption and add new features in digital blocks. A major restriction for the analog circuits is the increase of leakage currents, insignificant in the process selected for this work. An extra model of the integrator, taking the leakage currents into account, would thus be appreciable.
  Published ADCs with limited voltage supply are more and more inverter based [80, 81]. Static and dynamic models for these specific amplifiers are not yet integrated in the developed simulator.
- The linear simulator introduced in this work accounts for the digital signals

from the control blocks as well as the logic gates, added locally in the analog. It was supposed that the edges of the drivers of the switches were delay-free and perfectly synchronized with the clock. A delay model based on the current drive of the logic gates and on the capacitive load on the nodes would be useful to check the order of arrival of the control signal and to prevent unwanted glitches. These simulations are commonly used in digital synthesizers to identify critical paths, but are not extended to the gates included in the analog blocks.

- Model for the charge injection. In this work, the model used was based on results provided by a transistor-level simulator. Models for clock feedthrough and charge injection would be valuable to be compatible with several technologies. The main challenges are the accurate management of the transient of the control signals and partial compensation of the charges held in both transistors of analog transmission gates.
- Finally, a graphical user interface would help promoting the software and would speed up the design of data converters. A feature to import directly the architecture of the modulator from the low-level simulator netlist would also be attractive to perform a first validation of the schematics.

Bibliography

[1] M.F. Snoeij, O. Bajdechi, and J.H. Huijsing. A 4th-order switched-capacitor sigma-delta a/d converter using a high-ripple chebyshev loop filter. In *Circuits and Systems, 2001. ISCAS 2001. The 2001 IEEE International Symposium on*, volume 1, pages 615–618 vol. 1, May 2001.

[2] B. Nowacki, N. Paulino, and J. Goes. A 1.2 v 300uw second-order switched-capacitor sigma-delta modulator using ultra incomplete settling with 73 db sndr and 300 khz bw in 130 nm cmos. In *ESSCIRC (ESSCIRC), 2011 Proceedings of the*, pages 271–274, Sept 2011.

[3] Jun Wang, T. Matsuoka, and K. Taniguchi. A 0.5 v feedforward delta-sigma modulator with inverter-based integrator. In *ESSCIRC, 2009. ESSCIRC '09. Proceedings of*, pages 328–331, Sept 2009.

[4] S. Gambini and J. Rabaey. A 100ks/s 65db dr sigma-delta adc with 0.65v supply voltage. In *Solid State Circuits Conference, 2007. ESSCIRC 2007. 33rd European*, pages 202–205, Sept 2007.

[5] Ahmed Gharbiya. Architecture alternatives for time-interleaved and input-feedforward delta-sigma modulators. *PhD Thesis*, Jul 2008.

[6] K.A. O'Donoghue, P.J. Hurst, and S.H. Lewis. A digitally calibrated 5-mw 2-ms/s 4th-order sigma-delta adc in 0.25- um cmos with 94 db sfdr. In *ESSCIRC, 2010 Proceedings of the*, pages 422–425, Sept 2010.

[7] B.P. Brandt and B.A. Wooley. A 50-mhz multibit sigma-delta modulator for 12-b 2-mhz a/d conversion. *Solid-State Circuits, IEEE Journal of*, 26(12):1746–1756, Dec 1991.

[8] E. Bonizzoni, A.P. Perez, H. Caracciolo, D. Stoppa, and F. Maloberti. An incremental adc sensor interface with input switch-less integrator featuring 220-nvrms resolution with ±30-mv input range. In *ESSCIRC (ESSCIRC), 2012 Proceedings of the*, pages 389–392, Sept 2012.

[9] J. Liang and D.A. Johns. A frequency-scalable 15-bit incremental adc for low power sensor applications. In *Circuits and Systems (ISCAS), Proceedings of 2010 IEEE International Symposium on*, pages 2418–2421, May 2010.

[10] V. Peluso, P. Vancorenland, A.M. Marques, M.S.J. Steyaert, and Willy Sansen. A 900-mv low-power delta; sigma; a/d converter with 77-db dynamic range. *Solid-State Circuits, IEEE Journal of*, 33(12):1887–1897, Dec 1998.

[11] C.B. Wang, S. Ishizuka, and B.Y. Liu. A 113-db dsd audio adc using a density-modulated dithering scheme. *Solid-State Circuits, IEEE Journal of*, 38(1):114–119, Jan 2003.

[12] Kei-Tee Tiew and Minkyu Je. A 0.06-mm2 double-sampling single-ota 2nd-order sigma-delta modulator in 0.18-um cmos technology. In *Solid State Circuits Conference (A-SSCC), 2011 IEEE Asian*, pages 253–256, Nov 2011.

[13] H.A. Cubas and J. Navarro. Design of an ota-miller for a 96db snr sc multi-bit sigma-delta modulator based on gm/id methodology. In *Circuits and Systems (LASCAS), 2013 IEEE Fourth Latin American Symposium on*, pages 1–4, Feb 2013.

[14] Yang Shaojun, Tong Ziquan, Jiang Yueming, and Dou Naiying. The design of a multi-bit sigma-delta adc modulator. In *Measurement, Information and Control (ICMIC), 2013 International Conference on*, volume 01, pages 280–283, Aug 2013.

[15] A. Pena-Perez, E. Bonizzoni, and F. Maloberti. A 88-db dr, 84-db sndr very low-power single op-amp third-order $\sigma\delta$ modulator. *Solid-State Circuits, IEEE Journal of*, 47(9):2107–2118, Sept 2012.

[16] Liyuan Liu, Dongmei Li, Liangdong Chen, Yafei Ye, and Zhihua Wang. A 1-v 15-bit audio $\delta\sigma$ -adc in 0.18 μ m cmos. *Circuits and Systems I: Regular Papers, IEEE Transactions on*, 59(5):915–925, May 2012.

[17] F. Medeiro, B. Perez-Verdu, J.M. de la Rosa, and A. Rodriguez-Vazquez. Quick design of high-performance sigma; delta; modulators using cad tools: a 16.4 b 1.71 mw cmos sigma; delta;m for 9.6 ksample/s a/d conversion. In *Analog and Mixed IC Design, 1997. Proceedings., 1997 2nd IEEE-CAS Region 8 Workshop on*, pages 22–27, Sep 1997.

[18] D.B. Kasha, W.L. Lee, and A. Thomsen. A 16-mw, 120-db linear switched-capacitor delta-sigma modulator with dynamic biasing. *Solid-State Circuits, IEEE Journal of*, 34(7):921–926, Jul 1999.

[19] S. Hein and A. Zakhor. Optimal decoding for data acquisition applications of sigma delta modulators. *Signal Processing, IEEE Transactions on*, 41(2):602 –616, feb 1993.

[20] Nguyen T. Thao. Deterministic analysis of sigma–delta modulation for linear and non-linear signal reconstruction. *International Journal of Circuit Theory and Applications*, 25(5):369–391, 1997.

[21] M.A. Miled, M. Sawan, and E. Ghafar-Zadeh. A dynamic decoder for first-order modulators dedicated to lab-on-chip applications. *Signal Processing, IEEE Transactions on*, 57(10):4076 –4084, oct. 2009.

[22] P.W. Wong and R.M. Gray. Sigma-delta modulation with i.i.d. gaussian inputs. *Information Theory, IEEE Transactions on*, 36(4):784 –798, jul 1990.

[23] L.G. McIlrath. A robust o(n log n) algorithm for optimal decoding of first-order sigma;- delta; sequences. *Signal Processing, IEEE Transactions on*, 50(8):1942 –1950, aug 2002.

[24] S. Kavusi, H. Kakavand, and A.E. Gamal. On incremental sigma-delta modulation with optimal filtering. *Circuits and Systems I: Regular Papers, IEEE Transactions on*, 53(5):1004 – 1015, may 2006.

[25] S. Maréchal, F. Krummenacher, and M. Kayal. Optimal filtering of an incremental second-order mash11 sigma-delta modulator. In *Electronics, Circuits and Systems (ICECS), 2011 18th IEEE International Conference on*, pages 240–243, 2011.

[26] S. Maréchal, F. Krummenacher, and M. Kayal. Optimal filtering of incremental first-order sigma-delta modulators with sweep input. In *Electronics, Circuits, and Systems (ICECS), 2010 17th IEEE International Conference on*, pages 539 –542, dec. 2010.

[27] F. Dachselt and S. Quitzk. Structure and information content in sequences from the single-loop sigma-delta modulator with dc input. In *ISCAS (4)*, volume 4, pages 685–688, may 2004.

[28] Hossein Kakav and Abbas El Gamal. Bounds on distortion bit-cost function for first order sigma-delta analog-to-digital converter with input noise. In *44th Annual Allerton Conference on Communication, Control and Computing*, pages 454–550, 2006.

[29] Libin Yao, M.S.J. Steyaert, and Willy Sansen. A 1-v 140- mu;w 88-db audio sigma-delta modulator in 90-nm cmos. *Solid-State Circuits, IEEE Journal of*, 39(11):1809–1818, Nov 2004.

[30] Jeongjin Roh, Sanho Byun, Youngkil Choi, Hyungdong Roh, Yi-Gyeong Kim, and Jong-Kee Kwon. A 0.9-v 60-uw 1-bit fourth-order delta-sigma modulator with 83-db dynamic range. *Solid-State Circuits, IEEE Journal of*, 43(2):361–370, Feb 2008.

[31] Youngcheol Chae and Gunhee Han. Low voltage, low power, inverter-based switched-capacitor delta-sigma modulator. *Solid-State Circuits, IEEE Journal of*, 44(2):458–472, Feb 2009.

[32] Mu-Chen Huang and Shen-Iuan Liu. A fully differential comparator-based switched-capacitor $\delta\sigma$ modulator. *Circuits and Systems II: Express Briefs, IEEE Transactions on*, 56(5):369–373, May 2009.

[33] T. Musah, Sunwoo Kwon, H. Lakdawala, K. Soumyanath, and Un-Ku Moon. A 630μw zero-crossing-based $\delta\sigma$ adc using switched-resistor current sources in 45nm cmos. In *Custom Integrated Circuits Conference, 2009. CICC '09. IEEE*, pages 1–4, Sept 2009.

[34] Jian Xu, Xiaobo Wu, Hanqing Wang, Junyi Shen, and Bill Liu. Power optimization of high performance $\delta\sigma$ modulators for portable measurement applications. In *Solid State Circuits Conference (A-SSCC), 2010 IEEE Asian*, pages 1–4, Nov 2010.

[35] V. Peluso, M.S.J. Steyaert, and Willy Sansen. A 1.5-v-100- mu;w delta; sigma; modulator with 12-b dynamic range using the switched-opamp technique. *Solid-State Circuits, IEEE Journal of*, 32(7):943–952, Jul 1997.

[36] J. Sauerbrey, T. Tille, D. Schmitt-Landsiedel, and R. Thewes. A 0.7-v mosfet-only switched-opamp sigma; delta; modulator in standard digital cmos technology. *Solid-State Circuits, IEEE Journal of*, 37(12):1662–1669, Dec 2002.

[37] J. Sauerbrey, M. Wittig, D. Schmitt-Landsiedel, and R. Thewes. 0.65v sigma-delta modulators. In *Circuits and Systems, 2003. ISCAS '03. Proceedings of the 2003 International Symposium on*, volume 1, pages I–1021–I–1024 vol.1, May 2003.

[38] Huy-Binh Le, Sang-Gug Lee, and Seung-Tak Ryu. A regulator-free 84db dr audio-band adc for compact digital microphones. In *Solid State Circuits Conference (A-SSCC), 2010 IEEE Asian*, pages 1–4, Nov 2010.

[39] Min Gyu Kim, Gil-Cho Ahn, P.K. Hanumolu, Sang-Hyeon Lee, Sang-Ho Kim, Seung-Bin You, Jae-Whui Kim, G.C. Temes, and Un-Ku Moon. A 0.9 v 92 db double-sampled switched-rc delta-sigma audio adc. *Solid-State Circuits, IEEE Journal of*, 43(5):1195–1206, May 2008.

[40] Zhenglin Yang, Libin Yao, and Yong Lian. A 0.5-v 35- u w 85-db dr double-sampled delta-sigma modulator for audio applications. *Solid-State Circuits, IEEE Journal of*, 47(3):722–735, March 2012.

[41] P. Rombouts, J. De Maeyer, and L. Weyten. A 250-khz 94-db double-sampling sigma; delta; modulation a/d converter with a modified noise transfer function. *Solid-State Circuits, IEEE Journal of*, 38(10):1657–1662, Oct 2003.

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

[43] Jose M De la Rosa and Rocio del Rio. *CMOS sigma-delta converters: practical design guide; 3rd ed.* Wiley - IEEE. Wiley, Hoboken, NJ, 2013.

[44] K. Francken, P. Vancorenland, and G. Gielen. Daisy: a simulation-based high-level synthesis tool for /spl delta//spl sigma/ modulators. In *Computer Aided Design, 2000. ICCAD-2000. IEEE/ACM International Conference on*, pages 188–192, 2000.

[45] S. Brigati, F. Francesconi, P. Malcovati, D. Tonietto, A. Baschirotto, and F. Maloberti. Modeling sigma-delta modulator non-idealities in simulink(r). In *Circuits and Systems, 1999. ISCAS '99. Proceedings of the 1999 IEEE International Symposium on*, volume 2, pages 384–387 vol.2, 1999.

[46] J.L. McCreary. Matching properties, and voltage and temperature dependence of mos capacitors. *Solid-State Circuits, IEEE Journal of*, 16(6):608–616, 1981.

[47] R. Gaggl. *Delta-Sigma A/D-Converters: Practical Design for Communication Systems*. Springer Series in Advanced Microelectronics. Springer, 2012.

[48] R.T. Baird and T.S. Fiez. Linearity enhancement of multibit delta; sigma; a/d and d/a converters using data weighted averaging. *Circuits and Systems II: Analog and Digital Signal Processing, IEEE Transactions on*, 42(12):753–762, 1995.

[49] Tom Kwan, R. Adams, and R. Libert. A stereo multibit sigma; delta; dac with asynchronous master-clock interface. *Solid-State Circuits, IEEE Journal of*, 31(12):1881–1887, 1996.

[50] R. Adams and K.Q. Nguyen. A 113-db snr oversampling dac with segmented noise-shaped scrambling. *Solid-State Circuits, IEEE Journal of*, 33(12):1871–1878, 1998.

[51] R. Schreier and B. Zhang. Noise-shaped multibit d/a convertor employing unit elements. *Electronics Letters*, 31(20):1712–1713, 1995.

[52] A. Yasuda, H. Tanimoto, and T. Lida. A 100 khz 9.6 mw multi bit /spl delta//spl sigma/ dac and adc using noise shaping dynamic elements matching with tree structure. In *Solid-State Circuits Conference, 1998. Digest of Technical Papers. 1998 IEEE International*, pages 64–65, 1998.

[53] I. Galton. Noise-shaping d/a converters for delta; sigma; modulation. In *Circuits and Systems, 1996. ISCAS '96., Connecting the World., 1996 IEEE International Symposium on*, volume 1, pages 441–444 vol.1, 1996.

[54] I. Galton. Spectral shaping of circuit errors in digital-to-analog converters. *Circuits and Systems II: Analog and Digital Signal Processing, IEEE Transactions on*, 44(10):808–817, 1997.

[55] www.mathworks.com/matlabcentral/fileexchange/19-delta-sigma-toolbox.

[56] http://www2.imse-cnm.csic.es/simsides/.

[57] B.E. Boser and B.A. Wooley. The design of sigma-delta modulation analog-to-digital converters. *Solid-State Circuits, IEEE Journal of*, 23(6):1298–1308, 1988.

[58] Ge Binjie, Wang Xin'an, Zhang Xing, Feng Xiaoxing, and Wang Qingqin. Study and analysis of coefficient mismatch in a mash21 sigma-delta modulator. *Journal of Semiconductors*, 31(1):015007, 2010.

[59] E.A.M. Klumperink, S.L.J. Gierkink, A.P. van der Wel, and B. Nauta. Reducing mosfet 1/f noise and power consumption by switched biasing. *Solid-State Circuits, IEEE Journal of*, 35(7):994 –1001, july 2000.

[60] O. Oliaei. Noise analysis of correlated double sampling sc-integrators. In *Circuits and Systems, 2002. ISCAS 2002. IEEE International Symposium on*, volume 4, pages IV–445 – IV–448 vol.4, 2002.

[61] O. A. Hafiz, X. Wang, P. J. Hurst, and S. H. Lewis. Immediate calibration of operational amplifier gain error in pipelined adcs using extended correlated double sampling. *Solid-State Circuits, IEEE Journal of*, PP(99):1 –11, 2013.

[62] C.C. Enz and G.C. Temes. Circuit techniques for reducing the effects of op-amp imperfections: autozeroing, correlated double sampling, and chopper stabilization. *Proceedings of the IEEE*, 84(11):1584 –1614, nov 1996.

[63] M. Kayal, F. Burger, and R.S. Popovic. Magnetic angular encoder using an offset compensation technique. *Sensors Journal, IEEE*, 4(6):759 – 763, dec. 2004.

[64] Zhao Jing, Zhang Shuang, and Chen Shudong. A chopper stabilized pre-amplifier for magnetic sensor. In *Industrial Control and Electronics Engineering (ICICEE), 2012 International Conference on*, pages 501 –504, aug. 2012.

[65] M.A.P. Pertijs and W.J. Kindt. A 140 db-cmrr current-feedback instrumentation amplifier employing ping-pong auto-zeroing and chopping. *Solid-State Circuits, IEEE Journal of*, 45(10):2044 –2056, oct. 2010.

[66] I.G. Finvers, J.W. Haslett, and F.N. Trofimenkoff. Noise analysis of a continuous-time auto-zeroed amplifier. *Circuits and Systems II: Analog and Digital Signal Processing, IEEE Transactions on*, 43(12):791 –800, dec 1996.

[67] M. Degrauwe, E. Vittoz, and I. Verbauwhede. A micropower cmos instrumentation amplifier. In *Solid-State Circuits Conference, 1984. ESSCIRC '84. Tenth European*, pages 31 –34, sept. 1984.

[68] Peng Sun, Menglian Zhao, Xiaobo Wu, and Rui Fan. A novel capacitively-coupled instrumentation amplifier employing chopping and auto-zeroing. In *Circuits and Systems (APCCAS), 2012 IEEE Asia Pacific Conference on*, pages 156 –159, dec. 2012.

[69] J.F. Witte, J.H. Huijsing, and K.A.A. Makinwa. A chopper and auto-zero offset-stabilized cmos instrumentation amplifier. In *VLSI Circuits, 2009 Symposium on*, pages 210 –211, june 2009.

[70] . Three is a crowd for instrumentation amplifiers. *Maxim Engineering Journal*, 60:3–8, 2007.

[71] T Kugelstadt. Getting the most out of your instrumentation amplifier design. *Texas Instrument Analog Applications Journal*, 4Q:25–29, 2005.

[72] Kitchin C. and Counts L. A designer's guide to instrumentation amplifiers, 3rd edition. *Analog Devices Technical Resources*, 2006.

[73] Hyunsik Park, KiYoung Nam, D.K. Su, K. Vleugels, and B.A. Wooley. A 0.7-v 870-μw digital-audio cmos sigma-delta modulator. *Solid-State Circuits, IEEE Journal of*, 44(4):1078–1088, April 2009.

[74] Jian-Yi Wu, R. Subramoniam, Zhenyong Zhang, A. Djabbari, P. Holloway, F. Maloberti, M. Yousefi, M. Aslan, Hua Hong, and A. Bahai. Multi-bit sigma delta adc with reduced feedback levels, extended dynamic range and increased tolerance for analog imperfections. In *Custom Integrated Circuits Conference, 2007. CICC '07. IEEE*, pages 77–80, Sept 2007.

[75] Yao Liu, E. Bonizzoni, A. D'Amato, and F. Maloberti. A 105-db sndr, 10 ksps multi-level second-order incremental converter with smart-dem consuming 280 μw and 3.3-v supply. In *ESSCIRC (ESSCIRC), 2013 Proceedings of the*, pages 371–374, Sept 2013.

[76] O. Nys and R.K. Henderson. A 19-bit low-power multibit sigma-delta adc based on data weighted averaging. *Solid-State Circuits, IEEE Journal of*, 32(7):933–942, Jul 1997.

[77] V. Quiquempoix, P. Deval, A. Barreto, G. Bellini, J. Markus, J. Silva, and G.C. Temes. A low-power 22-bit incremental adc. *Solid-State Circuits, IEEE Journal of*, 41(7):1562–1571, July 2006.

[78] YuQing Yang, T. Sculley, and J. Abraham. A single-die 124 db stereo audio delta-sigma adc with 111 db thd. *Solid-State Circuits, IEEE Journal of*, 43(7):1657–1665, July 2008.

[79] D.A. Kerth, D.B. Kasha, T.G. Mellissinos, D.S. Piasecki, and E.J. Swanson. A 126 db linear switched-capacitor delta-sigma modulator. In *Solid-State Circuits Conference, 1994. Digest of Technical Papers. 41st ISSCC., 1994 IEEE International*, pages 196–197, Feb 1994.

[80] Hao Luo, Yan Han, R.C.C. Cheung, Xiaopeng Liu, and Tianlin Cao. A 0.8-v 230-μ w 98-db dr inverter-based $\sigma\delta$ modulator for audio applications. *Solid-State Circuits, IEEE Journal of*, 48(10):2430–2441, Oct 2013.

[81] T. Christen. A 15-bit 140- μ w scalable-bandwidth inverter-based $\delta\sigma$ modulator for a mems microphone with digital output. *Solid-State Circuits, IEEE Journal of*, 48(7):1605–1614, July 2013.

Curriculum Vitæ

Name : Sylvain MARÉCHAL
Date of birth : 13.11.1984
Nationality: Swiss

EDUCATION

- **2008-2014**
  Swiss Federal Institute of Technology, Lausanne, Switzerland
  PhD in Microsystems and Microelectronics
- **2003-2008**
  Swiss Federal Institute of Technology, Lausanne, Switzerland
  M.Sc. in Signal Processing
  B.Sc. in Electrical Engineering
- **2000-2003**
  Gymnase de Nyon, Nyon, Switzerland
  Maturité fédérale in Physics and Applied Mathematics

WORK EXPERIENCE

- **2008-2014**
  EPFL, Lausanne, Switzerland
  Research and teaching assistant
- **2012-2013**
  Semtech Neuchâtel Sàrl, Neuchâtel, Switzerland
  Analog IC Designer – 4 months work as a consultant
- **2008**
  Semtech Neuchâtel Sàrl, Neuchâtel, Switzerland
  Analog IC Designer – 5 months fellowship
- **2006**
  EPFL, Lausanne, Switzerland
  Analog Designer – 2 months fellowship

List of Publications

- Maréchal, S.; Nys, O.; Krummenacher, F.; Chevroulet, M.; Kayal, M., "Design and Noise Analysis of a Novel Auto-Zeroing Structure for Continuous-Time Instrumentation Amplifiers," *International Journal of Electronics and Telecommunications.* Volume 59, Issue 4, Pages 397-404

- Maréchal, S.; Krummenacher, F.; Kayal, M., "Simulating Nonlinear Capacitors in Sigma-Delta Modulators," *Electronics, Circuits and Systems (ICECS), 2013 20th IEEE International Conference on* , vol., no., pp., 8-11 Dec. 2013

- Maréchal, S. ; Nys, O. ; Krummenacher, F. ; Chevroulet, M. ; Kayal, M., "A continuous-time instrumentation amplifier employing a novel auto-zeroing structure," *Mixed Design of Integrated Circuits and Systems (MIXDES), 2013 Proceedings of the 20th International Conference* , vol., no., pp.157,162, 20-22 June 2013; **Best paper award**

- Maréchal, S.; Krummenacher, F.; Kayal, M., "CAD tools for fast analysis of parasitic coupling and mismatch effects in switched-capacitor circuits," *Electronics, Circuits and Systems (ICECS), 2011 18th IEEE International Conference on* , vol., no., pp.354,357, 11-14 Dec. 2011

- Maréchal, S.; Krummenacher, F.; Kayal, M., "Optimal filtering of an incremental second-order MASH11 sigma-delta modulator," *Electronics, Circuits and Systems (ICECS), 2011 18th IEEE International Conference on* , vol., no., pp.240,243, 11-14 Dec. 2011

- Maréchal, S.; Krummenacher, F.; Kayal, M., "Optimal filtering of incremental first-order sigma-delta modulators with sweep input," *Electronics, Circuits, and Systems (ICECS), 2010 17th IEEE International Conference on* , vol., no., pp.539,542, 12-15 Dec. 2010; **Best paper award**