UNIVERSITY OF SOUTHAMPTON

FACULTY OF PHYSICAL SCIENCES AND ENGINEERING

Electronics and Computer Science

Multi Stage Noise Shaping (MASH) Sigma Delta Modulator for Capacitive MEMS Inertial Sensors

by

Bader Almutairi

Thesis for the degree of Doctor of Philosophy

April 2015

UNIVERSITY OF SOUTHAMPTON

ABSTRACT

School of Electronics and Computer Science

Doctor of Philosophy

Multi Stage Noise Shaping (MASH) Sigma Delta Modulator for Capacitive MEMS Inertial Sensors

by *Bader Almutairi*

This research discusses the theoretical investigation, simulation and hardware implementation of the ElectroMechanical Multi-stAge noise SHaping (EM-MASH) sigma delta modulator (ΣΔM). The potential advantages of an EM-ΣΔM MASH compared to single-loop high-order ΣΔMs applied to inertial MEMS sensors are its inherent stability and high overload input level due to the use of lower order ΣΔMs in its individual stages. Furthermore, MASH has the advantages of high dynamic range and high noise shaping performance because of its overall high-order ΣΔM architecture. So far, the EM-MASH has not been sufficiently explored. This study is expected to serve as solid basis for the application of EM-MASH. In this research, various EM-MASH architectures (MASH21, MASH22, MASH211, MASH221 and MASH222) were theoretically studied, and the results were validated with simulations. A fourth order EM-MASH22-ΣΔM was theoretically examined and successfully implemented with a capacitive MEMS accelerometer, which includes a second order EM-ΣΔM loop cascaded with a purely electronic second order ΣΔM. The quantization noise from the first loop is digitised by the second loop and then cancelled by digital filters, whereas the quantization noise from the second loop is shaped by the second loop filter and a digital filter, which together provide fourth order noise shaping. The performance of EM-MASH22 was compared with that of a single-loop fourth order EM-ΣΔM (SD4). Both architectures were investigated by system level modelling and hardware implementation using surface-mount PCB technology. The results show that (a) both architectures achieve the same noise floor level of 19 μg/√Hz; (b) MASH22 is unconditionally stable, whereas SD4 is only conditionally stable; and (c) MASH22 achieves a higher overload input level and a higher dynamic range than does SD4. Furthermore, the research presents a novel EM-MASH-ΣΔM that employs a dual quantization technique and adopts a 2-0 structure (EM-MASH20). With a simpler and configurable composition, MASH20 is aimed at exhibiting a performance higher than that achieved with the MASH22 structure. The MASH20 does not require a second-stage ΣΔM, which reduces the complexity of the digital filters compared to those required for the MASH22; thus, the digital filter matching is more easily achievable. The study shows that the MASH20, like the MASH22, has an inherent stability, high overload input level, and high dynamic range compared to single-loop ΣΔM. However, the MASH20, with its simpler implementation, achieved a higher dynamic range and signal-to-noise ratio than the MASH22 and the SD4. A capacitive MEMS accelerometer was designed and employed with MASH20. Within a bandwidth of 1 kHz, the sensor achieves a noise floor level of 15 μg/√Hz, a full-scale acceleration of ±20 g and a bias instability of 20 μg for a period of three hours. The EM-MASH-ΣΔM is sensitive to the variation of the sensing element parameters and other analogue parameters, both of which are subject to manufacturing tolerance and imperfections. This causes a leakage of the quantization noise in the final output and degrades the modulator performance. The research explored a calibration method to solve this problem by utilizing the digital domain capabilities. The method is based on the optimization algorithm which was investigated using MATLAB. The research confirms the concept of the EM-MASH structure and proves that it is applicable as a closed-loop interface for high-performance capacitive MEMS inertial sensors.

Table of Contents

List of Tables

List of Figures

DECLARATION OF AUTHORSHIP

I, **Bader Almutairi**, declare that the thesis entitled "**Multi Stage Noise Shaping (MASH) Sigma Delta Modulator for Capacitive MEMS Inertial Sensors**" and the work presented in it are my own and has been generated by me as the result of my own original research. I confirm that:

1. This work was done wholly or mainly while in candidature for a research degree at this University;
2. Where any part of this thesis has previously been submitted for a degree or any other qualification at this University or any other institution, this has been clearly stated;
3. Where I have consulted the published work of others, this is always clearly attributed;
4. Where I have quoted from the work of others, the source is always given. With the exception of such quotations, this thesis is entirely my own work;
5. I have acknowledged all main sources of help;
6. Where the thesis is based on work done by myself jointly with others, I have made clear exactly what was done by others and what I have contributed myself;
7. Parts of this work have been published as:

1- Almutairi, B., Alshehri, A., Kraft, M., "**Design and implementation of a MASH2-0 electromechanical sigma-delta modulator for capacitive MEMS sensors using dual quantization method**" (Submitted to *Journal of Microelectromechanical Systems* – under review).

2- Almutairi, B., Alshehri, A, Kraft, M., "**MASH2-0 electromechanical sigma-delta modulator for capacitive MEMS sensors using dual quantization method**", *IEEE Sensors 2014*, pp. 1780-1783, 2-5 Nov. 2014.

3- Almutairi, B., Kraft, M., "**Multi stage noise shaping sigma–delta modulator (MASH) for capacitive MEMS accelerometers**," *Sensors and Actuators A: Physical*, vol. 186, p. 169–177, 2012.

4- Kraft, M., Wilcock, R., Almutairi, B., "**Innovative control systems for MEMS inertial sensors**," *IEEE International Frequency Control Symposium (FCS) 2012*, pp. 1-6, 21-24 May 2012.

5- Almutairi, B., Kraft, M., "**Experimental study of single loop sigma-delta and multi stage noise shaping (MASH) modulators for MEMS accelerometer**," *IEEE Sensors Conference*, pp. 520-523, 28-31 Oct. 2011.

6- Almutairi, B., Kraft, M., "**Multi stage noise shaping sigma-delta modulator (MASH) for capacitive MEMS accelerometers**," *Procedia Engineering, Eurosensors XXV*, vol. 25, p. 1313–1316, 4-7 Sept. 2011.

7- Almutairi, B., Kraft, M., "**Comparative study of multi stage noise shaping and single loop sigma–delta modulators for MEMS accelerometers**" *Procedia Engineering, Eurosensors XXIV*, vol. 5, pp. 512-515, 2010.

Signed:.....

Date: .....

Acknowledgements

I am very grateful to my supervisor, Professor Michael Kraft, who gave me the opportunity to do this research. I would like to thank him for his support and encouragement throughout this entire process. His guidance helped me tremendously during my research and the writing of this thesis.

I also would like to thank my brother-in-law, Ali Alshehri, for his valuable help and support during my PhD journey. I would like to express my gratitude for the beneficial discussions and collaboration with the friendly PhD fellows in Mountbatten Building and thereby thank Reuben, Ioannis and Pejwaak for their help and advice.

I will always be thankful to my family for making the necessary sacrifices to allow me to study in the United Kingdom so that I could have the opportunity to pursue higher education. I offer special thanks to my brother, Ali Almutairi, for taking some family responsibilities off of my shoulders and for his support during my study. I would like to express my deepest gratitude to my wife for her encouragement, support and patience, despite the fact that she is also a student and raising our lovely daughters: Yara and Mayar.

I am also deeply grateful to the King Abdulaziz City for Science and Technology (KACST[1]) for offering me a scholarship to complete my PhD study at the University of Southampton.

[1] www.kacst.edu.sa

Abbreviations

ADC	analogue to digital convertor
AM	amplitude modulation
ASIC	application-specific integrated circuits
BGA	ball gate array
BOX	buried oxide layer
CDS	correlated double sampling
CFC	capacitance to frequency conversion
CIDF	cascaded integrator with distributed feedback
CMOS	complementary metal oxide semiconductor
CPLD	complex programmable logic device
CT	continues time
CVC	capacitance-to-voltage conversion
DAC	digital to analogue converter
DC	direct current
DRIE	deep reactive ion etching
EM	electromechanical
EMI	electromagnetic interference
ENTF	electronic noise transfer function
FEM	finite element method
FIR	finite impulse response

FPGA	field programmable gate array
GA	genetic algorithm
HDL	hardware description language
HF	hydrofluoric acid
IIR	infinite impulse response
MASH	multi-stage noise shaping
MEMS	micro-electro-mechanical systems
LMS	least mean square
MSB	most-significant-bit
NRZ	non-return-to-zero
NTF	noise transfer function
OLA	overload acceleration input
OLF	overload factor
Op-Amp	operational amplifier
OSR	oversampling ratio
PC	personal computer
PCB	printed circuit board
PIC	peripheral interface controller
PSD	power spectral density
QNTF	quantization noise transfer function
RMS	root mean square

ROM	read only memory
SC	switched capacitance
SD2	second order single loop electromechanical sigma delta modulator
SD4	fourth order single loop electromechanical sigma delta modulator
SEM	scanning electron microscope
SMD	surface mount device
SNR	signal-to-noise
SOI	silicon-on-insulator
SPI	serial peripheral interface
STF	signal transfer function
TNEA	total noise equivalent acceleration
UART	universal asynchronous receiver/transmitter
UDP	user datagram protocol
VHDL	very high speed integrated circuit hardware description language
USB	universal serial bus
ΣΔM	sigma delta modulator

Symbols

a(s)	sensor's frame acceleration in Laplace domain
A	area of the overlap between electrodes
Amp	amplitude of the input acceleration signal
Area	electrode overlap area
b	damping coefficient
C	lead compensator
$C_{a\text{-}bot}$	bottom actuation capacitor
$C_{a\text{-}top}$	top actuation capacitor
$C_{s\text{-}bot}$	bottom sense capacitor
$C_{s\text{-}top}$	top sense capacitor
C_0	nominal sense capacitor
D_i	digital filter of the i^{th} stage of the MASH structure
d_0	gap between electrodes
e	maximum quantization error
E(f)	noise spectral density function in frequency
ε_r	relative permittivity of the material between the capacitor plates
ε_0	permittivity of free space
F_{elec}	electrostatic force applied on the proof mass
f	frequency in Hz
F_n	Noise force
f_r	resonance frequency
f_s	sampling frequency
f_0	upper limit of the bandwidth of interest
g	gravity acceleration (9.81 m/s^2)

Gi	feedback gain of the ΣΔM i^{th} internal loop
H	electrode height
Hi	single loop ΣΔM forward path i^{th} integrator
k	spring constant
K_B	Boltzmann constant
K_{bst}	boost gain
K_{elec}	electrostatic force spring
K_{fb}	feedback loop gain
Ki	single loop ΣΔM forward path i^{th} integrator gain
K_{po}	pickoff circuit gain
K_{qi}	quantization noise gain of the i^{th} quantizer
KR	scaling gain R of the quantization error in MASH
KS	scaling gain S of the quantization error in MASH
K2	second stage input gain
L	electrode length
m	proof mass
M	sensing element second order transfer function
Nqi	quantization noise signal of the i^{th} quantizer
Q	quality factor
Q_{n1}	quantization noise signal of the i^{th} quantizer in the MASH structure
s	Laplace parameter
S	accelerometer sensitivity
T	Kelvin temperature
V_{bot}	voltage signal applied on the bottom actuator capacitor
Vc	carrier signal

V_{fb}	feedback voltage signal
V_{fs}	quantizer full scale level range
V_{top}	voltage signal applied on the top actuator capacitor
ω_n	natural frequency of the undamped accelerometer
ω_r	resonance frequency of the damped accelerometer
x	proof mass displacement with respect to the sensor's frame
$\dot{x}$	proof mass velocity with respect to the sensor's frame
$\ddot{x}$	proof mass acceleration with respect to the sensor's frame
y	proof mass displacement
$\dot{y}$	proof mass velocity
$\ddot{y}$	proof mass acceleration
z	sensor's frame displacement
$\dot{z}$	sensor's frame velocity
$\ddot{z}$	sensor's frame acceleration
Z	digital domain parameter
ζ	damping ratio
Δ	space between two adjacent levels
ΔC	change in capacitance

.

"In the name of Allah, the Most Gracious, the Most Merciful."

Chapter 1: **Introduction**

1.1 Introduction

Micro-electro-mechanical systems (MEMS) research has been steadily expanding since the inception of the vibrating beam resonator [1] in 1967, which introduced semiconductor microfabrication to the field of MEMS. Beside the miniature size of MEMS, the ability to manufacture MEMS products in high-volume batches lowers of the cost of the units for consumers [2] [3]. Furthermore, advanced microfabrication techniques allow monolithic integration of MEMS devices with application-specific integrated circuits (ASIC) in a single chip [4].

The accelerometer is one of the most important MEMS sensors; its importance arises from its ability to measure the acceleration of a moving object. A number of different systems utilize the accelerometer in their operations. For example, military and aerospace systems require extremely accurate acceleration measurements to perform accurate navigation [5]. On the other hand, low-cost MEMS accelerometers are in high demand for use in cost-effective applications due to their good specifications and miniature size. The list of these applications is uncountable, as every day new systems are introduced. Some of these applications include inertial navigation [6] [7] [8], biomedical applications (i.e. pacemaker, patient monitoring) [9] [10], consumer electronics (i.e. smart phones, three-dimensional computer mice, electronic toys) [11], sports activities [12], robotic systems and industrial monitoring [13].

Inertial MEMS accelerometers convert mechanical energy into electrical energy (transduction). The transduction mechanism used in MEMS varies, with the most common transduction techniques being piezoresistive, capacitive, piezoelectric, optical, resonant, thermal and magnetic [14] [15]. Each transduction technique has its own field and research; it is beyond the research scope of this work to discuss each technique. Capacitive transduction and actuation will be the primary topics in the present research. The capacitive MEMS sensor benefits from high sensitivity, good DC response, good noise performance, low drift, low temperature sensitivity, low power dissipation, and simple design and fabrication. However, they suffer from Electromagnetic Interference (EMI), which can be minimized by the use of good packaging and shielding [16].

The capacitive MEMS accelerometers are classified by the operation mode, either open or closed loop mode. The closed loop mode is achieved by controlling the proof mass using the actuation capacitors with the appropriate electronic circuit, to produce an electrostatic force that keeps the proof mass very close to its nominal position. The closed loop mode has more advantages compared with the open loop mode. It has wider bandwidth, higher dynamic range and increased sensitivity [17] [18]. Since the proof mass in closed loop mode has small deflection, thus, nonlinear effects associated with the sense and actuator capacitors, squeeze film damping and the beam elasticity can be considered very insignificant. There are two approaches for the closed loop accelerometer, analogue and digital [19]. The former requires a bias voltage to be applied with the feedback signal to achieve linear electrostatic force; nevertheless, this approach is limited with very small proof mass deflection, and for large deflection a nonlinear electrostatic force which will lead to pull-in situation. The digital closed loop accelerometer is achieved by applying digital pulses on the actuation electrodes, and it does not encounter the pull-in problem. Also, it has one more advantage that is the output of the accelerometer is in digital form, which can be interfaced to a digital system. One way to construct the digital closed loop accelerometer is to incorporate it with the sigma-delta modulator (ΣΔM).

1.2 Motivation and Contribution

In this research, multi-stage noise shaping (MASH) ΣΔMs are considered as control structures for interfacing with inertial capacitive MEMS sensors, which deploy a closed loop control on the inertial sensor and emit a measurement in the form of a digital signal.

The concept of the electromechanical (EM) ΣΔM was initially introduced by *Henrion* et al. [20] with a second order EM-ΣΔM in which only the micromechanical accelerometer-sensing element provided the loop dynamics. It was obvious that quantization noise was a key concern in such second order control architecture. Therefore, recent research has focused on the ability of high-order EM-ΣΔMs to minimize quantization noise and improve the linearity, dynamic range and bandwidth of capacitive MEMS sensors while providing digital output in the form of a pulse-density modulated signal. High order EM-ΣΔMs with single loop architecture have successfully been applied to capacitive MEMS accelerometers [21-23]. However, increasing the order of these modulators in single loop

architecture comes with stability constraints, and can only operate with a lower maximum acceleration input level compared to second order EM-ΣΔMs [24] [25]. Otherwise, the system will become overloaded and unstable. On the other hand, a high order EM-ΣΔM can be achieved by means of MASH. The potential advantages of an EM-MASH-ΣΔM compared to single loop, high order ΣΔMs applied to inertial MEMS sensors are its inherent stability and high overload input level due to the use of lower order ΣΔMs in its individual stages. Furthermore, MASH has the advantages of high dynamic range and high noise shaping performance because of its overall high-order ΣΔM architecture.

This research discusses the theoretical investigation, simulation and hardware implementation of the EM-MASH. As part of this research, a fourth order EM-MASH22 was theoretically investigated [26] and successfully implemented with a MEMS accelerometer [27-30], which included a second order EM-ΣΔM loop cascaded with a purely electronic second order ΣΔM. The quantization noise from the first loop is digitized by the second loop and then cancelled by digital filters, while the quantization noise from the second loop is shaped by the second loop filter and a digital filter, which together provide fourth order noise shaping.

The dual quantization technique represents another approach to the design of a ΣΔM [31-33] that benefits from both the reduced error of a multi-bit quantizer and the inherent linearity of a single-bit quantizer in a single modulator. However, a MASH version of this technique has not been explored for use in EM-ΣΔMs despite its advantages compared to the single-loop approach. Such a version would rely on digitizing the signal of the forward path by a multi-bit analogue to digital convertor (ADC), which allows the signal to be processed in the digital domain. The structure uses the most-significant-bit (MSB) to close the first ΣΔM loop, while the remaining bits represent the quantization error of the single-bit quantizer, which can be then removed by a cancellation logical unit.

The research presents a novel EM-MASH that employs the dual quantization technique and adopts an electromechanical 2-0 multi-stage noise shaping structure (EM-MASH20).

The EM-MASH is sensitive to the sensing-element parameters and other analogue parameters, both of which are subject to manufacturing tolerance and imperfections. This causes a leakage of the quantization noise in the final output and degrades the modulator

performance. The research explored a calibration method to solve this problem by utilizing digital domain capabilities. The method is based on the genetic algorithm (GA) which was investigated and verified using MATLAB.

It should be mentioned that the higher order MASH can be implemented for both MEMS accelerometers and gyroscopes. The accelerometers usually require low-pass filtering, as they acquire their measurement within the bandwidth up to their resonant frequency. Gyroscopes, in contrast, require band pass filters to measure rotation rates within the drive mode oscillation frequency. To prove the EM-MASH concept a capacitive MEMS accelerometer was used in this research.

1.3 Document Structure

The thesis is dedicated to research on the EM-MASH, and it is structured in eight chapters as follows:

Chapter 2 provides theoretical background, briefly discussing the basic topics in capacitive MEMS accelerometers. These include mechanical lumped modelling and performance analysis of inertial MEMS accelerometers, capacitive MEMS technology, operation mode as open-loop and closed-loop accelerometer, and the principles of ΣΔMs in terms of quantization, modulation and noise shaping, as well as further discussion on higher order ΣΔMs.

Chapter 3 is a literature review discussing recent research on closed-loop EM-ΣΔMs. It covers the two architectures of these modulators: single-loop EM-ΣΔM and EM-MASH.

Chapter 4 presents the design and simulation of the EM-MASH. It presents theoretical and simulation investigations of the EM-MASH with various architectures with the use of MATLAB and the Simulink software package. The design procedure of a successful MASH system is also presented. This chapter goes on to discuss the stability, the electrostatic feedback force modelling, and the maximum acceleration input level of the system.

Chapter 5 is a theoretical comparative study between two fourth order ΣΔM structures, single-loop and MASH. Both of them are modelled using Simulink and incorporate over-damped and under-damped accelerometers. The comparison focuses on the noise shaping

and the signal-to-noise (SNR), stability, input signal power vs. SNR, and parameter sensitivity.

Chapter 6 discusses the hardware implementation and the measurements of the electronic circuit that accommodates either 4^{th} order single loop (SD4) ΣΔM or MASH22 modulator architectures. It first presents a detailed approach to characterising the sensing element and the pickoff circuit. Then, each subunit of the electronic circuit is explained. The first and second stages of the MASH22 are individually investigated. The post-processing and digital filtering of the data using MATLAB is discussed. Finally the MASH22 and SD4 are presented and experimentally compared.

Chapter 7 presents the design and implementation of the novel MASH20 structure for MEMS accelerometer. It includes a discussion of the design and fabrication process of the high-gee accelerometer designed for the ΣΔM system. The hardware implementation of the MASH20 will be discussed, along with the experimental results. A digital calibration method to solve the quantization leakage problem will be discussed. The method is based on the genetic algorithm (GA) which will be investigated and verified using MATLAB

Chapter 8 is the conclusion, in which future work based on the research is also proposed.

Chapter 2: **Theoretical Background**

2.1 Introduction

This chapter briefly discusses the theoretical background of the capacitive micro-electro-mechanical system (MEMS) accelerometer. First, the mechanical lumped model of the inertial accelerometer sensor will be discussed. Then, a brief discussion is presented about the capacitive MEMS technology used within such accelerometers. The principle of the ΣΔM will be discussed, in terms of construction, operation and quantization noise shaping; a further discussion about higher ΣΔM will follow.

2.2 Mechanical Lumped Model of an Accelerometer

The basic structure of a capacitive MEMS accelerometer is depicted in Figure 2-1(a). It comprises a proof mass anchored by fixable beams (springs) to a fixed frame (anchors). The proof mass can be equipped with a differential sense capacitor $C_{s\text{-}top}$ and $C_{s\text{-}bot}$ to detect the proof mass movement. In the closed-loop mode, the proof mass will be controlled by an electrostatic force generated by a set of differential capacitive actuators $C_{a\text{-}top}$ and $C_{a\text{-}bot}$. Due to the inertial force, the proof mass movement will encounter dissipative forces generated by the spring reaction and the squeeze film damping effect between the capacitors' electrodes [34-38].

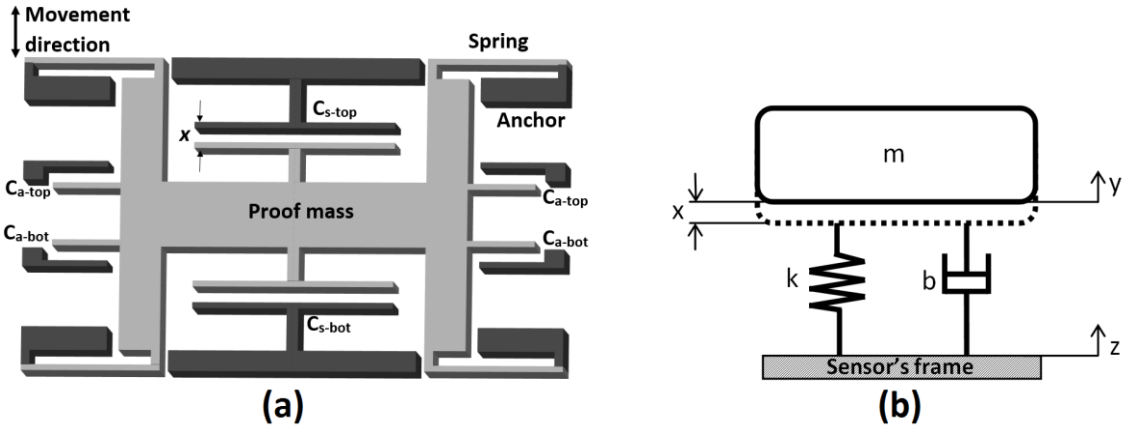


Figure 2-1: Capacitive MEMS accelerometer (a) basic structure and (b) lumped model.

The capacitive accelerometer can be represented with the mass-spring-damper system with the lumped parameter, as shown in Figure 2-1(b). The proof mass *m* is physically

attached to the sensor's frame with elastic beams of a spring constant *k*. The squeeze film damping effect originating between the electrodes of the sense and actuator capacitors is modelled with a dashpot with damping coefficient *b*. The sensor's frame and the proof mass displacements are denoted by *z* and *y*, respectively. Thus, the proof mass relative displacement *x* is equal to the difference between the displacement *z* and *y*, i.e., *x* = *z* - *y*.

If the first derivative of the displacements ($\dot{x}, \dot{y}\ and\ \dot{z}$) represent the velocity, and the second derivative of the displacements ($\ddot{x}, \ddot{y}\ and\ \ddot{z}$) represent the acceleration, the equation of motion of the mass-spring-damper system can be derived by applying Newton's second law as follows [39]:

$$m\ddot{y} = b\dot{x} + kx \tag{2.1}$$

By using the assumption of $x = z - y$ the equation in 2.1 can be rewritten as follows:

$$m\ddot{z} = m\ddot{x} + b\dot{x} + kx \tag{2.2}$$

If the equation of motion in 2.2 is represented in the Laplace domain, the accelerometer transfer function that shows the frequency response of the accelerometer *x* with respect to the acceleration of the sensor's frame ($a=\ddot{z}$) is given by [15]:

$$\frac{x(s)}{a(s)} = \frac{1}{\mathrm{s}^2 + \frac{\mathrm{b}}{\mathrm{m}}\mathrm{s} + \frac{\mathrm{k}}{\mathrm{m}}} \tag{2.3}$$

The natural frequency ω_n of the undamped accelerometer is given by:

$$\omega_n = \sqrt{\frac{k}{m}} \tag{2.4}$$

If the quality factor Q is defined as follows:

$$Q = \frac{\omega_n m}{b} = \frac{\sqrt{km}}{b} \tag{2.5}$$

The transfer function in 2.3 can be expressed in term of the quality factor Q and the natural resonance frequency ω_n as follows:

$$\frac{x(s)}{a(s)} = \frac{1}{s^2 + \frac{\omega_n}{Q}s + {\omega_n}^2} \qquad 2.6$$

Based on the quality factor, the frequency response of the accelerometer can be classified into three types, as shown in Figure 2-2. The red line shows the response when Q > 0.5. It can be seen that the system will be under-damped where, at the resonance frequency, the magnitude shows a resonance peak and fast phase change from 0° to 180°. If Q =0.5, as shown with green line, the system is classified as critically damped. The resonance peak is flattened and the phase is smoothly transited. At Q < 0.5, the accelerometer will be over-damped (as shown in blue line), so there is no resonance peak and at low frequency the response will exhibit a phase lag.

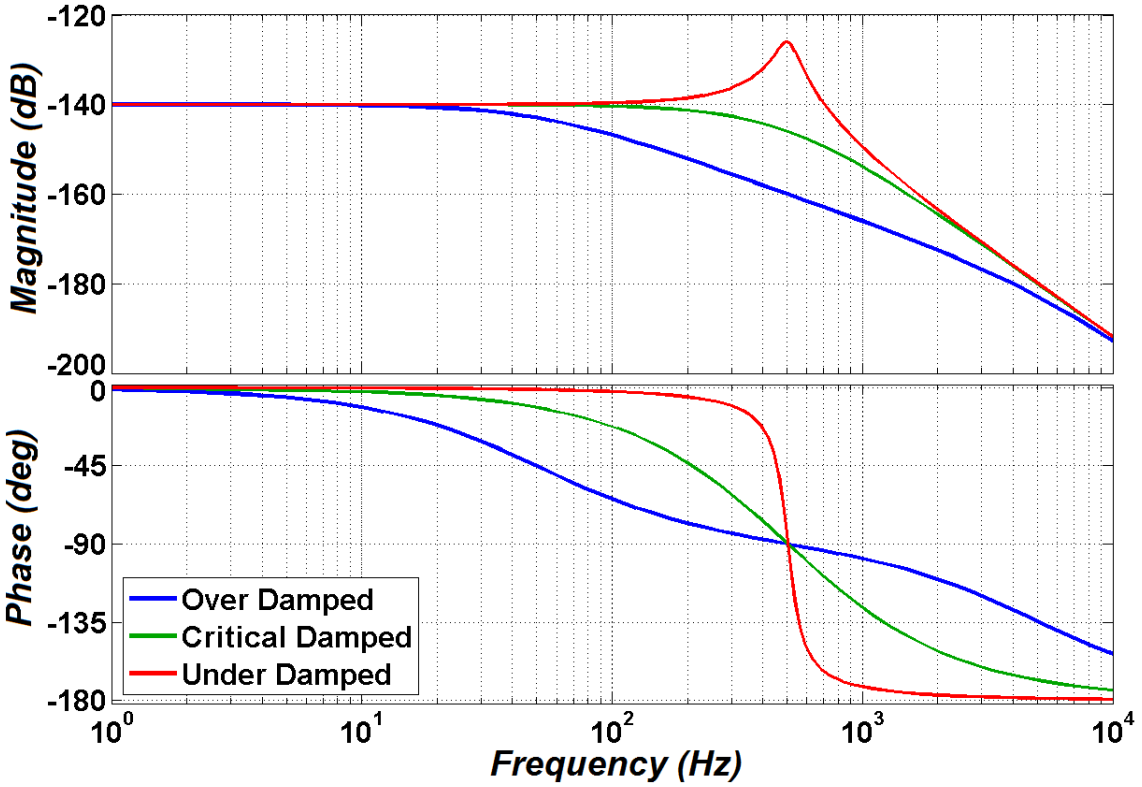


Figure 2-2: Frequency response of the accelerometer transfer function (equation 2.6) with Q=5 for under-damped (blue), Q=0.5 for critically-damped (green) and Q=0.1 for over-damped (red) response.

The resonant frequency ω_r of the damped accelerometer is given by:

$$\omega_r = \omega_n\sqrt{1 - \zeta^2} \qquad 2.7$$

Where ζ is the damping ratio, and is equal to:

$$\zeta = \frac{b}{2m\omega_n} \qquad 2.8$$

If equation 2.3 is analysed at low frequency ($s \ll \omega_r$) the accelerometer sensitivity (*S*) is given by [40]:

$$S = \frac{m}{k} = \frac{1}{{\omega_n}^2} \qquad 2.9$$

The operating bandwidth for the open-loop inertial MEMS accelerometer is usually limited by its natural frequency. It can be seen in equation 2.4 and 2.9 that by increasing the spring constant 'k' and/or decreasing the proof mass 'm', the bandwidth is increased but the sensitivity is decreased and vice versa. This trade-off between the sensitivity and the natural frequency (the bandwidth) can be solved if a closed-loop accelerometer is considered [19].

2.3 Brownian Noise

Due to the small size of the MEMS inertial sensors, the measurement signal power has a low value, which could easily be degraded by noise signals. The most commonly encountered type of noise signal here is thermal noise, which could indicate the lowest measurable value in MEMS inertial sensors. Because of the thermal variation, gas molecules fluctuate and affect the mass and spring with unwanted mechanical motion, which is known as Brownian noise. The noise frequency spectrum depends on the damping coefficient, which is expressed as a noise force F_n as follows [35]:

$$F_n = \sqrt{4K_BTb} \qquad 2.10$$

The total noise equivalent acceleration (TNEA) is given by:

$$TNEA = \frac{F_n}{m} = \frac{\sqrt{4K_BTb}}{m} = \frac{\sqrt{4K_BT\omega_r}}{mQ} \qquad 2.11$$

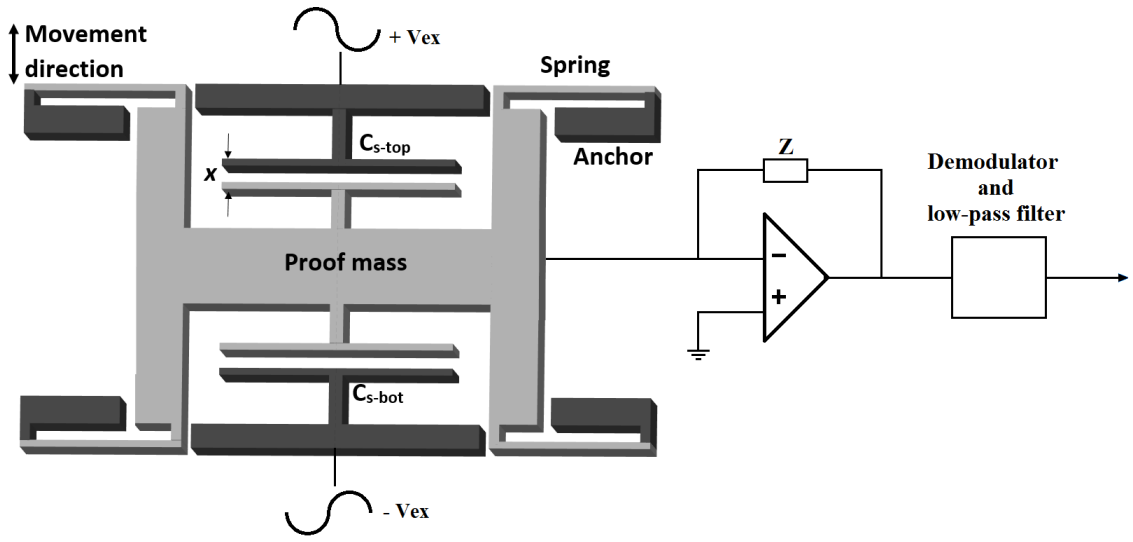
where K_B is the Boltzmann constant, T is the Kelvin temperature, b is the damping coefficient, ω_r is the resonant frequency of the inertial sensor, m is the mass and Q is the quality factor.

As can be seen in equation 2.11, the effect of Brownian noise can be reduced by increasing the mass and the quality factor of the inertial MEMS sensor [41].

2.4 Capacitive MEMS Control Strategies

2.4.1 Open-Loop Accelerometer

Capacitive MEMS inertial sensors are operated in either open- or closed-loop mode. Most low-cost commercial MEMS inertial sensors are open loop, where the proof mass movement is counteracted by the spring and damping forces. An open-loop accelerometer is shown in the conceptual block diagram in Figures 2-3, where the accelerometer is interfaced to a pickoff circuit that outputs a voltage proportional to the input acceleration. The proof mass is equipped with differential sense capacitors $C_{s\text{-}top}$ and $C_{s\text{-}bot}$ to detect proof mass movement. Due to inertial force, movement of the proof mass will change the sense capacitors $C_{s\text{-}top}$ and $C_{s\text{-}bot}$. Moreover, the proof mass will encounter dissipative forces generated by the spring reaction and the squeeze film damping effect between the capacitors' electrodes [42] [43].



Figures 2-3: Basic structure for a capacitive MEMS accelerometer with a differential sense capacitor configuration and a simple pickoff circuit.

The capacitive transduction technique is realized by gap variation. If the top electrode is considered, when the proof mass moves toward the electrode, the gap in between will decrease and the capacitance will increase. The capacitance is then given by [40]:

$$C_{s-top} = \frac{\varepsilon_0 \varepsilon_r A}{d_0 - x} \qquad 2\text{-}12$$

where ε_0 is the permittivity of free space, ε_r is the relative permittivity of the material between the plates, A is the area of overlap between electrodes, d_0 is the gap between electrodes and *x* is the proof mass displacement. Equation 2-12 shows that the capacitance changes with respect to the separation distance. The differential capacitance can be expressed as a change in capacitance (ΔC) [40]:

$$\Delta C = C_{s-top} - C_{s-bot} = \varepsilon A \left(\frac{1}{d_0 - x} - \frac{1}{d_0 + x} \right) = 2\varepsilon A \frac{x}{d_0^2 - x^2} \qquad 2\text{-}13$$

Assuming very short mass deflection *x* compared to the gap between the electrodes d_0, that is, $x << d_0$, the change in capacitance of the differential sense capacitor ΔC can be assumed to be linear. Therefore, equation 2-13 can be rewritten as follows [40]:

$$\Delta C \approx 2\varepsilon A \frac{x}{d_0^2} \quad [\text{F}] \qquad 2\text{-}14$$

Typically, the pickoff circuit can be realized by connecting the proof mass to a charge integrator forming a capacitive half-bridge, as shown in Figures 2-3. The pickoff circuit converts the change in capacitance due to acceleration into a proportional output voltage (V_f), which can be expressed as [19]:

$$V_f = K_{po} x \quad [\text{V}] \qquad 2\text{-}15$$

where K_{po} is a function relating the proof mass deflection to the output voltage, which can be given by [19]:

$$K_{po} = 2\varepsilon A \frac{x}{d_0^2 - x^2} K_c \quad [\text{V/m}] \qquad 2\text{-}16$$

where K_c is the pickoff circuit gain relating the differential change in capacitance to the output voltage. For small mass deflection, K_{po} can be considered constant. The linear approximation for equations 2-14 and 2-16 only holds true where very short mass deflection is assumed. In general, this assumption is not always valid for open-loop accelerometers [14] [19] [44]. Moreover, the squeeze film damping and the beam elasticity can no longer be considered linear with the larger proof mass deflection in open-loop accelerometer [45] [46]. Consequently, the major disadvantage of the open-loop

operation mode is its non-linear effects, which are caused by the possibility of the proof mass experiencing a large displacement. For a precision accelerometer, where the linearity requirement is crucial, the maximum allowable deflection of the proof mass is limited [14], and in some application it cannot exceed 10% of the nominal gap [19]. This poses a severe restriction on the dynamic range of the sensor.

2.4.2 Closed-Loop Accelerometer

Closed-loop accelerometers have the advantage of small and controlled proof mass displacement, which improves the linearity of the sensor. The closed-loop mechanism is achieved by applying electrostatic feedback force on the proof mass, which adds another dissipative force on top of the spring and damping forces. Consequently, this force extends the dynamic range of the sensor compared with the open-loop approach [14] [17] [19] [40] [45].

As discussed in section 2.2, the operating bandwidth for the open-loop accelerometer is limited by its natural frequency and there is a trade-off between sensitivity and bandwidth. However, when the accelerometer is operated in closed-loop mode, dependency on the spring constant for bandwidth and sensitivity is reduced [44], as the bandwidth of the system is determined by the bandwidth of the loop, which extends beyond the natural frequency of the sensor.

There are two ways to close the loop in a capacitive MEMS accelerometer: analogue or digital, which will be briefly discussed in the following sections.

2.4.2.1 Analogue Feedback Force

Analogue electrostatic force is one technique for closing the control loop in capacitive MEMS accelerometers. Figure 2-4 shows a general analogue closed-loop force feedback accelerometer. The sense electrodes of the accelerometer are interfaced to a pickoff circuit and followed by a compensator circuit for loop stability. The output is a measure of the input acceleration, which is an analogue voltage signal that is applied to the top and bottom actuator electrodes to generate an analogue electrostatic force that keeps the proof mass close to its nominal position.

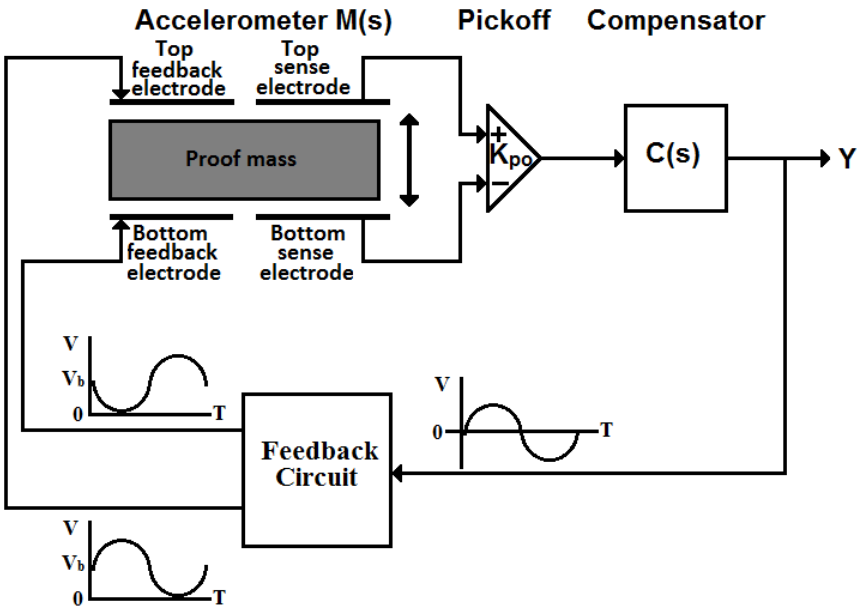


Figure 2-4. General block diagram of analogue closed-loop accelerometer.

Analogue electrostatic feedback force encounters some drawbacks. First, the relationship between the resulting electrostatic feedback force with the voltage signal (V_{fb}) and the proof mass displacement (x) are nonlinear, as indicated in equation 2-17. Second, the electrostatic force is always attractive; hence, it is not possible to achieve negative feedback force with a single actuator. The electrostatic force acting between the proof mass and one actuator electrode is given by [46]:

$$F = -\frac{\varepsilon A}{2}\frac{V_{fb}^2}{(d_0 - x)^2} \qquad 2\text{-}17$$

A common method for constructing a linear and negative electrostatic feedback force is to employ a differential actuator with a fixed bias voltage (V_b) additional to the top electrode and with equal magnitude but opposite polarity to bottom electrode [46]. The net electrostatic force (F) generated by this configuration is given by:

$$F_{ele} = F_{top} - F_{bot} = -\frac{\varepsilon A}{2}\left[\frac{(V_b - V_{fb})^2}{(d_0 - x)^2} - \frac{(V_b + V_{fb})^2}{(d_0 + x)^2}\right] \qquad 2\text{-}18$$

Assuming very short proof mass displacement (i.e. $d_o >> x$), equation 2-18 can be simplified to:

$$F_{ele} \approx -2\varepsilon A\left[\frac{V_b V_{fb}}{{d_0}^2}\right] \qquad 2\text{-}19$$

The net electrostatic force F_{ele} in the above equation is linear with respect to the applied feedback voltage. It is desirable to maximize the bias voltage V_b compared with the feedback voltage V_{fb} to enhance the linearity; however, too large a bias voltage will reduce the sensitivity, so a design trade-off needs to be considered [19].

When the proof mass displacement is large, the feedback force becomes nonlinear with respect to the applied feedback voltage ($V_{fb} \rightarrow F_{ele}$) and proof mass displacement ($F_{ele} \rightarrow x$). Furthermore, the feedback voltage is derived from the output voltage of the pickoff circuit, which also encounters nonlinearity with respect to the displacement ($x \rightarrow V_{fb}$). Equation 2-18 can now be rewritten to include the nonlinearity introduced by the pickoff circuit in equation 2-16, as follows [19]:

$$F_{ele} = -\frac{\varepsilon A}{2}\left[\frac{(V_b - K_{po})^2}{(d_0 - x)^2} - \frac{(V_b + K_{po})^2}{(d_0 + x)^2}\right] \qquad 2\text{-}20$$

For a typical sensors parameters shown in Table 2-1, Figure 2-5 shows the analogue electrostatic force in equation 2-20 with respect to the proof mass displacement and with the bias voltage ranging from 5 to 7.5 V. The negative feedback electrostatic force is negative and linear only for small proof mass displacement. For large proof mass displacement, it can be seen that the electrostatic force becomes nonlinear and is reduced, and for much larger proof mass displacement ($x > 2d_0/3$) [46-49], the electrostatic force changes its polarity to positive, which drives the proof mass towards the close fixed electrode. This phenomenon is called pull-in, where, at this particular point ($x > 2d_0/3$), the electrostatic force generated by the voltage value and the small gap between the proof mass and the close electrode overpowers the opposite electrostatic force generated by the other electrode. The pull-in instability problem is the main disadvantage of the analogue electrostatic feedback force. One solution is to add mechanical stoppers about 1/3 the length of the nominal gap. This prevents the proof mass from collapsing with the fragile electrodes. The disadvantage of this solution is that it increases the gap between the proof mass and the sense/actuator electrodes, which need to be as small as possible to maximize the capacitance and the sensitivity. There are other more complex solutions in the literature to solve the pull-in problem [48], which are beyond the scope of this research, such as changing the electrostatic force profile by modifying the plate structure of the actuator [50], using a leveraged bending technique and nonlinear mechanical strain-stiffening [51], or other control strategies [52-54].

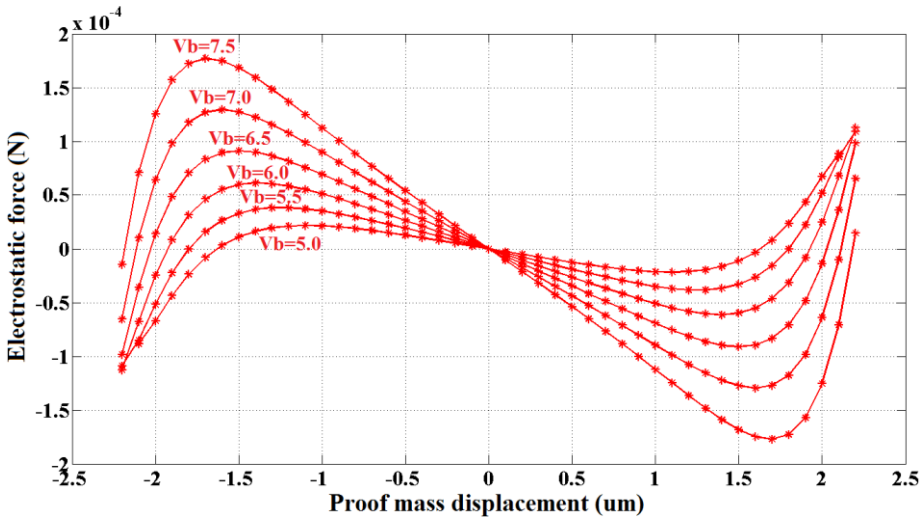


Figure 2-5. Analogue electrostatic feedback force function with respect to proof mass displacement and various bias voltages.

Parameter	value
Proof mass (m)	1.2×10^{-6} kg
Damping coefficient (b)	6×10^{-3} N.s/m
Spring constant (k)	5 N/m
Nominal capacitance (C_0)	16×10^{-12} F
Nominal gap	3×10^{-6}
Kc	10^{10} V/F

Table 2-1: Parameters of a typical capacitive MEMS accelerometer [19].

2.4.2.2 Digital Feedback Force

Digital feedback electrostatic force is based on ΣΔM architecture. Figure 2-6 shows a typical block diagram of a MEMS accelerometer interfaced to a pickoff circuit followed by a compensator circuit for loop stability. A single-bit comparator is used to provide two-level digital output. Based on the sign of the output, an electrostatic force is applied to the opposite direction of the proof mass movement. Digital force feedback control has several advantages over the analogue approach [19]:

1- The digital closed loop approach eliminates the pull-in problem that occurs in the analogue closed-loop approach. Since the electrode closer to the proof mass is grounded, while the other one is energized, the proof mass will always move towards the energized electrode, even with shocks in acceleration.

2- It provides a digital output signal in the form of a pulse density modulated bitstream.
3- It has a simple feedback circuit consisting of an analogue switch with two-level output.
4- The amount of the electrostatic force is based on the width of the voltage pulse, while in the analogue approach, it is based on the magnitude of the feedback voltage.

This control approach is the focus of this research and will be explored in more detail in the chapters that follow.

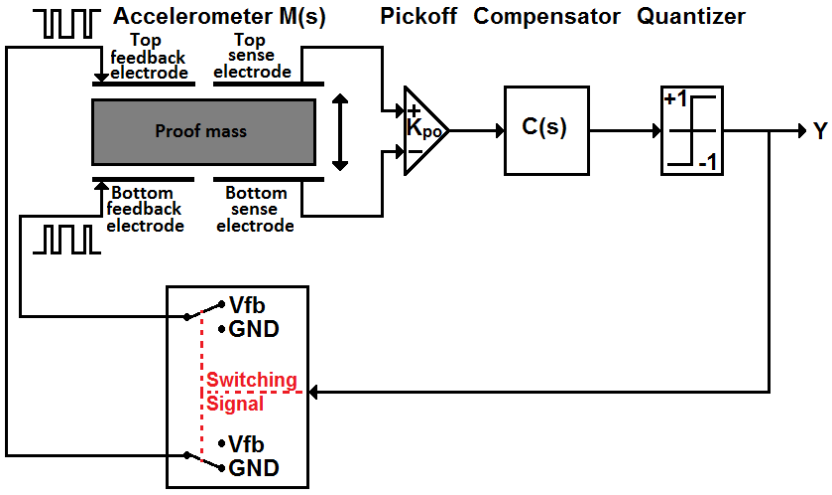


Figure 2-6. Typical block diagram of a digital closed-loop accelerometer using the ΣΔM technique.

2.5 Sigma-Delta Principle

2.5.1 Quantization & Modulation Noise

Analogue to digital conversion is performed by two main operations. The first is sampling of the analogue signal at a certain frequency f_s. The second is quantizing the sampled signal by rounding its value to one of the quantization levels. The quantization process clearly introduces a rounding error, which is designated the quantization error. The quantization is usually uniform; i.e., the space Δ between two adjacent levels is fixed, hence, the maximum quantization error 'e' is $\frac{\Delta}{2}$ [55], as shown in Figure 2-7.

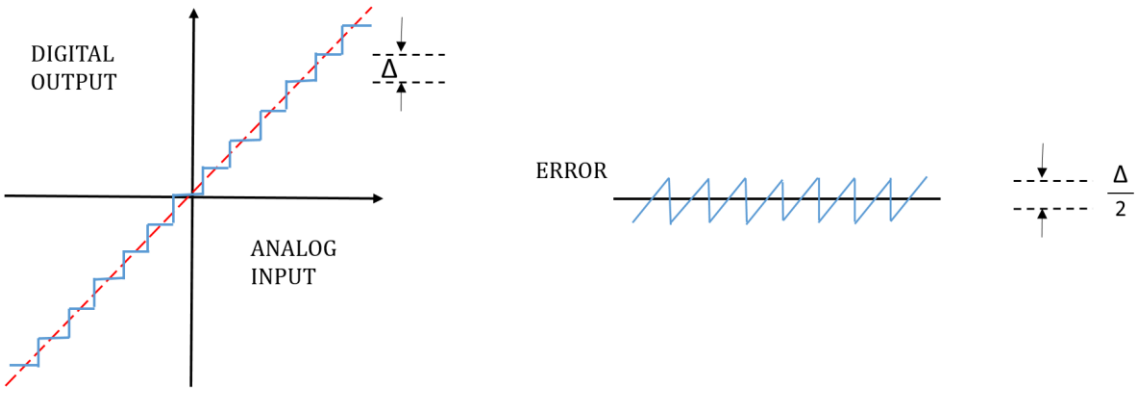


Figure 2-7: Quantization levels and error.

The quantizer error is the dominant error source that affects any modulator performance; therefore, it is essential to have a linear model for the quantizer in order to characterize its behaviour and then to minimize its quantization error 'e'. The output of the quantizer can be approximated by the following linear function [56]:

$$y = G.x + e \tag{2.21}$$

where y is the quantizer output, x is the analogue input signal to the quantizer, G is the quantizer gain and e is the added error. For an N-bit quantizer, if the input x remains within the quantizer full scale level range (V_{fs}), and changes randomly, then the quantization error e can be treated as a white noise that is independent of the input and has equal probability to lie between $-\frac{\Delta}{2}$ and $\frac{\Delta}{2}$. Hence, its mean square value can be given by [55]:

$$e_{rms}^2 = \frac{1}{\Delta}\int_{-\Delta/2}^{\Delta/2} e^2\, de = \frac{\Delta^2}{12} \tag{2.22}$$

$$e_{rms} = \frac{\Delta}{\sqrt{12}} \tag{2.23}$$

where

$$\Delta = \frac{V_{fs}}{2^N} \tag{2.24}$$

In general, when the quantization noise signal is uniformly distributed over the interval $\pm\frac{\Delta}{2}$, the rms quantization noise power equals $\frac{\Delta}{\sqrt{12}}$ and it is inversely proportional to the number of bits N.

2.5.2 Noise Shaping

2.5.2.1 Conventional Converters

Conventional analogue to digital converters are also known as Nyquist-Rate converters due to their use of the Nyquist sampling theorem, i.e., the sampling frequency f_s must be greater than twice the input signal bandwidth [57-59].

If the quantized input signal is sampled at frequency f_s, then all of its power will be folded into the band $0 \leq f < \frac{f_s}{2}$. Assuming the quantization noise is white, we can recall the result in equation 2.23 to derive the noise spectral density [56]:

$$E(f) = e_{rms}\left(\frac{2}{f_s}\right)^{\frac{1}{2}} \qquad \left(\frac{V}{\sqrt{Hz}}\right) \tag{2.25}$$

The noise power can be calculated by integrating $E^2(f)$ over the bandwidth of interest f_0 as follows:

$$n_0^2 = \int_0^{f_0} E^2(f)\, df = \mathrm{e}_{\mathrm{rms}}^2\left(\frac{2\mathrm{f}_0}{\mathrm{f}_\mathrm{s}}\right) \qquad (V^2) \tag{2.26}$$

$$n_0 = \mathrm{e}_{\mathrm{rms}}\left(\frac{2\mathrm{f}_0}{\mathrm{f}_\mathrm{s}}\right)^{\frac{1}{2}} \qquad (V) \tag{2.27}$$

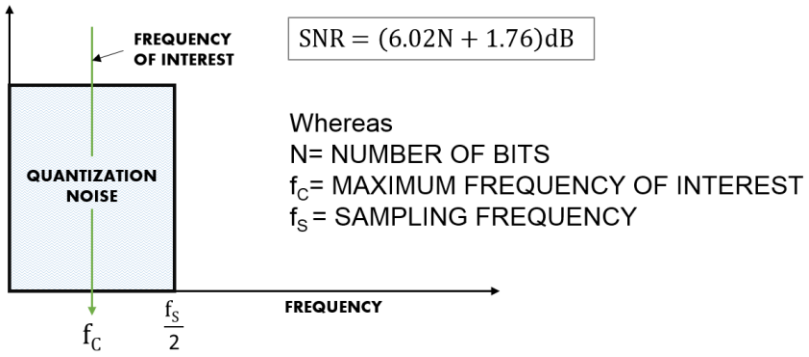


Figure 2-8 Quantization noise spectrum in Nyquist converter, reproduced from [55].

Figure 2-8 shows typical noise spectral for Nyquist converters, where most of the quantization noise appears in the bandwidth of interest [55]. The only way to increase the

SNR is to increase the effective number of bits, N or to increase the sampling frequency, as will be discussed in the next section.

2.5.2.2 Oversampling Converters

Oversampling converters, as their name suggests, use a sampling frequency that is much higher than the Nyquist frequency. These converters can achieve a higher effective number of bits and a higher SNR value compared with conventional converters, through the use of oversampling and digital filtering. They have relaxed analogue circuitry requirements compared to conventional converters [55]. The oversampling ratio OSR is defined as the ratio of the sampling frequency f_s to the Nyquist frequency $2f_0$. Recalling equation 2.27, the quantization noise can be written in terms of OSR, as follows:

$$n_0 = \mathrm{e_{rms}}\left(\frac{1}{\mathrm{OSR}}\right)^{\frac{1}{2}} \qquad (V) \qquad 2.28$$

Equation 2.28 shows that the in-band quantization noise is reduced by the square root of the OSR; in other words, each doubling of the OSR will reduce the in-band noise by 3 dB.

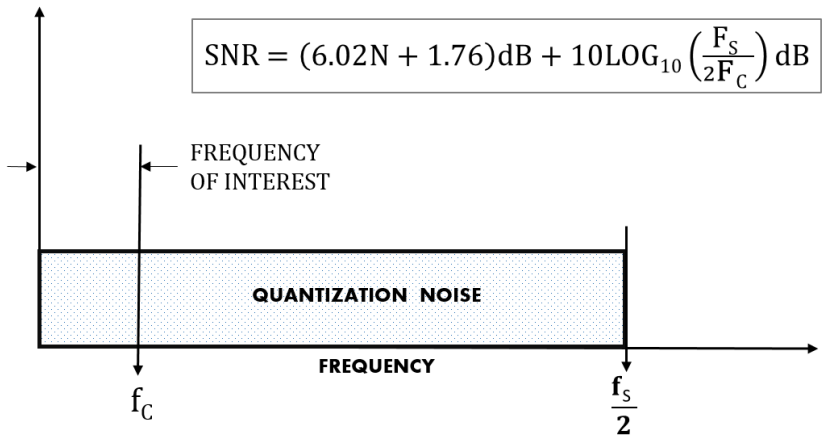


Figure 2-9 Quantization noise spectrum in oversampling converters, reproduced from [55].

As can be seen in Figure 2-9, the quantization noise is spread over a wider spectrum. Therefore, it is greatly reduced in the bandwidth of interest, and hence, the remaining noise outside the bandwidth can be filtered out with relaxed filter requirements.

2.5.2.3 Sigma-Delta Modulators

Like oversampling converters, ΣΔMs operate at a sampling frequency much higher than the Nyquist frequency $2f_0$, but with better noise shaping performance. A simple structure

of these types of converters is the first order modulator shown in Figure 2-10. In this case, the analogue input signal X enters the quantizer via an integrator and the digital output Y from the quantizer is fed back through a digital to analogue converter (DAC). This feedback structure assures that the average value of the output signal Y follows the average value of the input signal X.

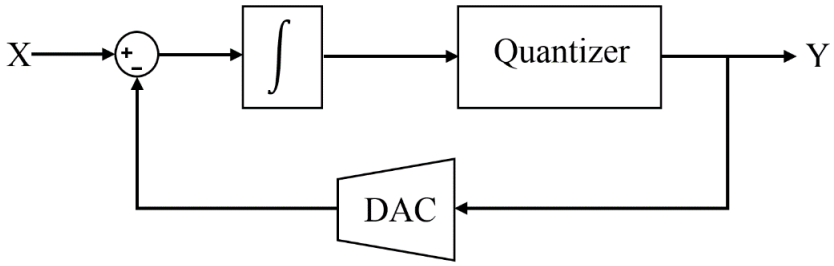


Figure 2-10: First order sigma-delta modulator.

To understand how the noise shaping is carried out by ΣΔM, a linear model is presented in Figure 2-11. With the assumption of a busy input signal, we can treat the quantization noise as white.

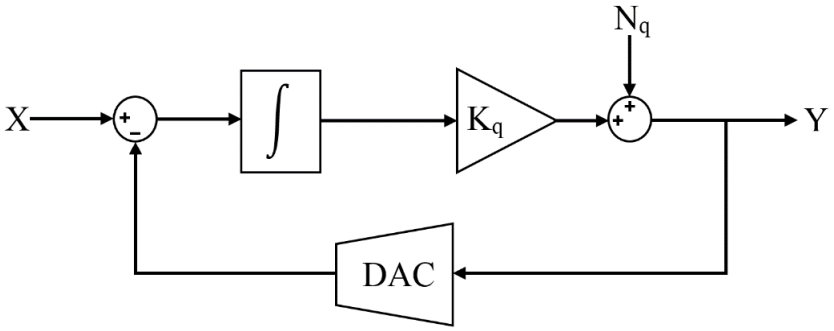


Figure 2-11: ΣΔM linear model.

The main cause of the noise shaping in a ΣΔM is the existence of the integrator in the feedback path of the quantization noise signal N_q; hence, the integrator acts as a high pass filter for the noise signal N_q. As a result, most of the quantization noise is pushed into a higher frequency band. In contrast, the integrator acts as a low pass filter for the input signal X, because it is located in its forward path. It is clear that adding more integrators will result in sharper cut-off filters, hence better noise shaping. This separation between the input signal X and the noise signal N_q, together with the oversampling, provide improved noise shaping without affecting the input signal within the bandwidth of interest.

For the above linearized model, with the assumption that quantization noise gain $K_q = 1$ and the DAC is replaced with a unity gain, the signal transfer function (STF) and the noise transfer function (NTF) of a first order modulator are given by:

$$STF = \frac{Y}{X} = \frac{1}{S+1} \qquad 2.29$$

$$NTF = \frac{Y}{Nq} = \frac{S}{S+1} \qquad 2.30$$

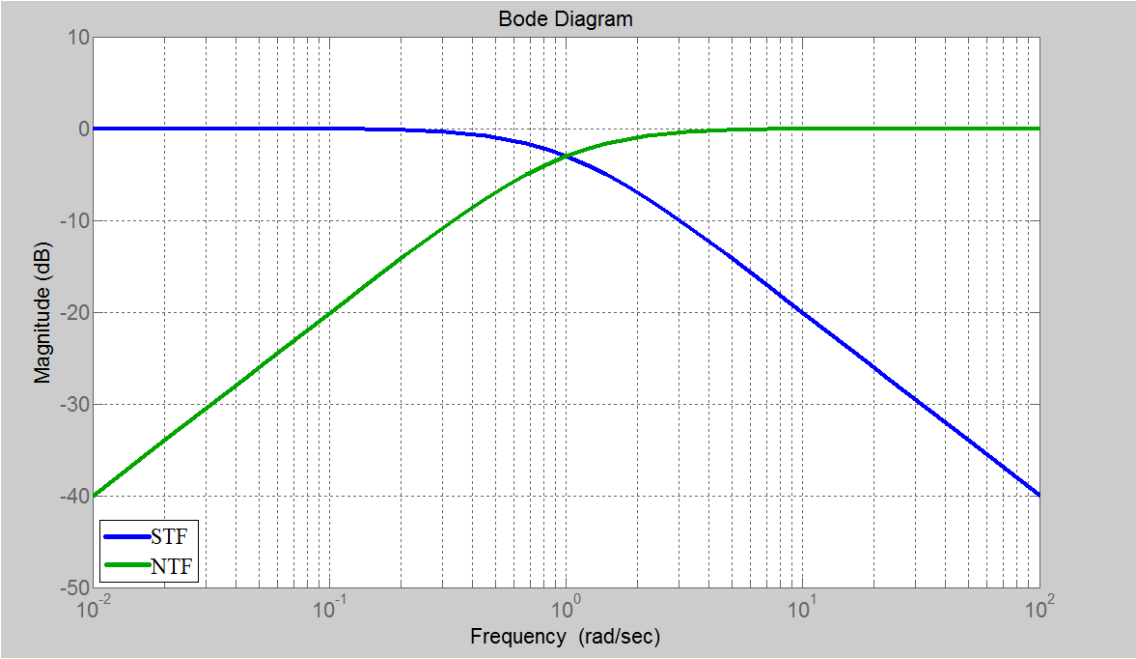


Figure 2-12 : Noise shaping of 1st order ΣΔM.

Figure 2-12 shows the noise shaping for the 1st order ΣΔM. The noise clearly is highly degraded in the bandwidth of interest and pushed into higher frequency.

To understand the origin of the terms Σ and Δ, it is better to represent the modulator in its sampled data equivalent diagram, as shown in Figure 2-13. The Δ can be seen to represent the negative feedback part while the Σ represents the functionality of the accumulator (integrator in the S-domain).

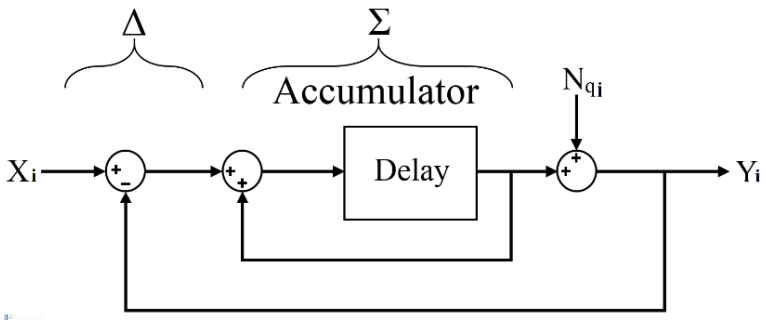


Figure 2-13 : Sampled data equivalent diagram for ΣΔM.

The difference equation for the output of the modulator shown in Figure 2-13 is given by:

$$y_i = x_{i-1} + (\mathrm{N_{qi}} - \mathrm{N_{qi-1}}) \qquad 2.31$$

Let $e_i = \mathrm{N_{qi}}$, the spectral density of the noise $n_i = \mathrm{N_{qi}} - \mathrm{N_{qi-1}}$ is given by:

$$N(f) = E(f) * \left|1 - e^{\frac{-j\omega}{f_s}}\right| = 2\mathrm{e_{rms}}(\tfrac{2}{f_s})^{\frac{1}{2}}\ \mathrm{sine}(\tfrac{\omega}{2f_s}) \qquad \left(\tfrac{V}{\sqrt{Hz}}\right) \qquad 2.32$$

The noise power can be calculated by integrating $E^2(f)$ over the bandwidth of interest f_0 and is given by:

$$n_0^2 = e_{rms}^2 \frac{\pi^2}{3}\left(\frac{1}{OSR}\right)^3 \qquad (V^2) \qquad 2.33$$

$$n_0 = e_{rms} \frac{\pi}{\sqrt{3}}\left(\frac{1}{OSR}\right)^{\frac{3}{2}} \qquad (V) \qquad 2.34$$

This means that in a 1st order ΣΔM, increasing the oversampling ratio OSR by a factor of 2 will decrease the in-band noise by 9 dB.

Equation 2.34 can be extended into a more generic formula for application to a higher order modulator (order > 2), as follows:

$$n_0 = e_{rms} \frac{\pi^L}{\sqrt{2L+1}}\left(\frac{1}{OSR}\right)^{L+\frac{1}{2}} \qquad (V) \qquad 2.35$$

where L is the order of the modulator. Equation 2.35 proves that each doubling of the OSR will decrease the in-band noise by $3(2L+1)$ dB.

2.5.3 Limit Cycle and Dithering

When the input to the ΣΔM is DC or a slowly varying signal, the feedback structure will make the output of the modulator oscillate between two levels in order to maintain its average value equal to the input. This oscillation produces a repetitive output sequence with a specific frequency. When this frequency lies within the input signal bandwidth, the modulation becomes noisy and the signal to noise ratio SNR will be highly degraded. This quantization error is variously referred to as *idle tones*, *noise pattern* or *limit cycles* [56]. When the input is sinusoidal, the quantization error may also become periodic, which causes the output spectrum to contain harmonic tones that are dependent on the amplitude and frequency of the input signal [60]. Some ADCs and DACs have treated this tone problem by shifting these tones to a higher frequency. As a result, the tones became less destructive when a small DC signal is injected in parallel to the input signal. Dithering is a technique used to overcome tone problems. It is a way to add a random signal or pseudo-random signal with a few bits of resolution to the quantizer input. Usually this random signal is assumed to be white noise; hence, it will have similar shape to that of the quantization noise. The main purpose of dithering is to whiten the quantization noise and to make it independent of the input signal [60]. In DAC, dithering is implemented by cascading a digital adder with a feedback shift register; this configuration generates a random number at each clock. Dithering in ADC is achieved using the same technique as for DAC, but it needs simple DAC in front of the shift register to convert the digital random number to an analogue signal. Therefore, dithering in ADC is relatively more complex in term of implementation [60].

2.5.4 Higher Order Single-Bit ΣΔ Modulators

In this section, two different architectures for implementing higher order single-bit ΣΔMs will be briefly discussed. The first is the single stage higher order ΣΔM, in which more integrators and feedback loops are added before a single quantizer. The second architecture is the multi-stage modulator, where several lower order (1^{st} or 2^{nd}) ΣΔMs are cascaded to form one higher order ΣΔM.

2.5.4.1 Single-Stage 1-Bit Higher Order ΣΔ Modulators

One obvious way to increase the SNR and the bit resolution of a ΣΔM is to increase the loop order, as depicted in equation 2.35. Unfortunately, as with any feedback non-linear

system, increasing the loop order by more than second order will result in loop instability and the system will become conditionally stable [60]. For a 1-bit modulator, the quantizer gain varies strongly with the input value, which makes the linearized stability analysis a difficult task. One way to maintain loop stability is to use root locus methods to examine the pole locations for various possible quantizer gains [55]. Another way to establish loop stability is to use extensive simulation, by carefully choosing the modulator coefficients and restricting the maximum input amplitude [60].

Several loop structures are available for implementing a single-stage high order ΣΔM, which provide trade-off between structure complexity and the freedom of choosing the shape of the STF and the position of the NTF zeroes [24]. The simplest and the most commonly used structure is the Cascaded Integrator with Distributed Feedback (CIDF), depicted in Figure 2-14. It contains a cascade of L delaying integrators with feedback signals that are scaled by factors a_i and fed to each integrator.

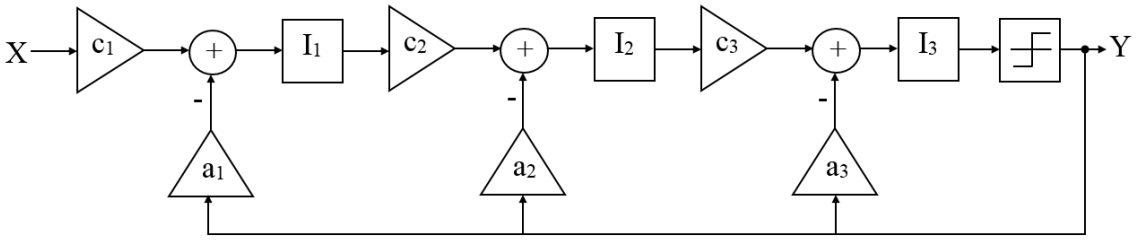


Figure 2-14: The cascaded integrators with distributed feedback (CIDF), reproduced from [24].

The single stage higher order ΣΔMs are resistant to analogue component mismatch. In addition, the quantization noise is randomized by the high order loop; therefore, the limit cycle tones are less likely to occur. Moreover, the higher order ΣΔMs have a higher dynamic range but also have a smaller overload threshold.

2.5.4.2 Multi-Stage 1-Bit Higher Order ΣΔ Modulators (MASH)

In the multi-stage higher order ΣΔM, several lower order (1st or 2nd) single stage ΣΔMs are cascaded to construct a higher order ΣΔM as shown in Figure 2-15. In the ideal case, the quantization noise Nq1 from the first stage is digitized by the second stage and then cancelled by the digital filters D1 and D2. The only noise that appears in the overall modulator output is the quantization noise Nq2 of the second stage, which will be shaped by an order equal to the sum of all stages orders. An advantage of the MASH modulator is that the quantization noise signals for all of the stages other than the first one are very

similar to a true white noise. This is because the input signal of each stage is a random noise signal by itself, hence, linear approximation for these stages is effective [55].

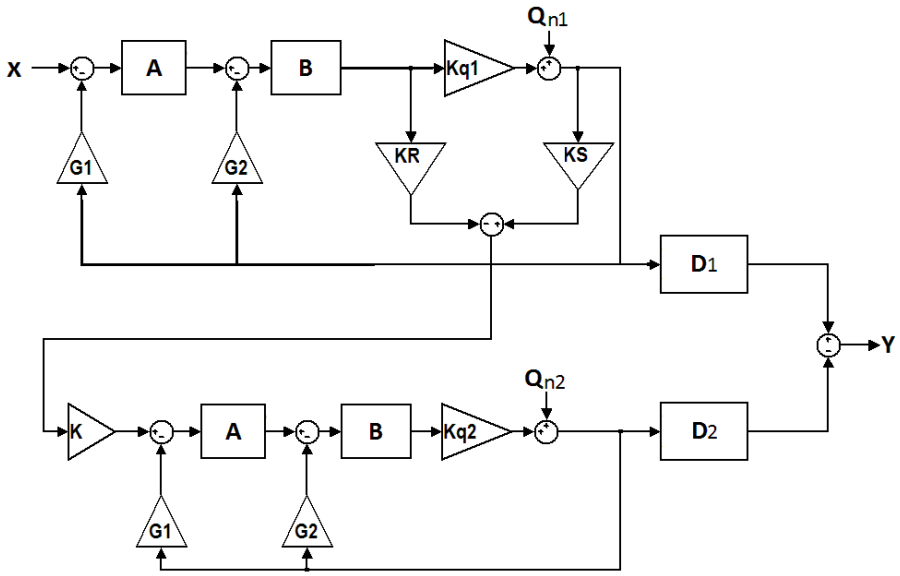


Figure 2-15: Multi-stage high order ΣΔM.

The MASH modulators display excellent stability properties and a high no-overload input threshold for their lower order stages. They also take the advantage of the high dynamic range and high noise shaping performance of the higher order ΣΔMs [24].

The drawback of the MASH modulators is the need for precise filter matching between the digital filters and the analogue components of the modulators. A mismatch in these will lead to a quantization error leakage from the first stage, which will substantially degrade the overall performance of the modulator.

2.6 Summary

In order to design closed loop MEMS accelerometer, the mechanical modelling of the inertial accelerometer was discussed using Newton's second low. The design of the MEMS accelerometer was focusing on the capacitive transduction and actuation technique, where the related equation that models the sense and electrostatic force were addressed.

The chapter addressed the basic concept of the ΣΔM, the design and the noise shaping of the single loop and the multi-stage (MASH) ΣΔM. One technique to apply digital closed loop accelerometer is to use the ΣΔM method which will be discussed in the next chapter.

Chapter 3: **Literature Review**

3.1 State of the Art

Significant improvements in microfabrication technology have enabled MEMS accelerometers with low to medium sensitivity to be commercialized at low cost [14]. However, some applications, such as inertial navigation systems, oil and gas exploration, space systems and earthquake detection, require highly sensitive and stable accelerometers. Table 3-1 shows the typical accelerometer requirements of automotive and inertial navigation applications.

Parameter	Automotive	Navigation
Range	± 50 g (airbag) ± 2 g (stability)	± 1 g
Frequency range	DC-400 Hz	DC-100 Hz
Resolution	< 100 mg (airbag) < 10(stability)	< 4 µg
Off-axis sensitivity	< 5%	< 0.1%
Nonlinearity	< 2%	< 0.1%
Max. shock in 1msec	> 2000 g	> 10 g
Temperature range	-40 ^{o}C to 85 ^{o}C	-40 ^{o}C to 80 ^{o}C
Temp. coeff. offset	< 60 mg/^{o}C	< 50 µg/^{o}C
Temp. coeff. sensitivity	< 900 pmm/^{o}C	± 50 pmm/^{o}C

Table 3-1. Typical accelerometer requirements of automotive and inertial navigation applications (reproduced from [14]).

In order to achieve high performance and obtain the most from capacitive MEMS accelerometers, the mechanical and electronic noise has to be minimized as much as possible. Therefore, the research about the capacitive MEMS accelerometer has been focussed on three different issues.

The first area for improvement is the design of the accelerometer itself, where different aspects in the design of the accelerometer are aimed to improve the sensitivity and minimize the mechanical noise:

1- The higher the sense capacitance, the higher the sensitivity and the lower the required pickoff circuit gain, which yields less electronic noise. Also, the higher the actuation capacitance, the higher the electrostatic force generated, which

results in a high dynamic range; thus, low feedback voltage is required, which, in turn, reduces the required power. In order to increase the sense and actuation capacitors, the electrodes have to be designed with a large overlap area and with minimum gap. An out-of-plane accelerometer benefits from a large area for the sense and actuation capacitors, using the whole proof mass area. Also in such designs, the gap between the proof mass and electrode is very small (around 2 μm) which results in large capacitive per mass displacement. These two requirements set the microfabrication limits for the in-plane accelerometers, where a large overlap area is obtained using bulk micromachining or SOI technology and the sensor benefits from the whole wafer thickness, which is usually in the range of 25-100 μm and the gap between two electrodes sets the maximum aspect ratio.

2- Increasing the proof mass will increase the sensitivity and reduce the influence of Brownian noise [61-63]. Again, the out-of-plane sensors are favourable; such sensors utilise the whole wafer thickness for proof mass [64] [65]. Nevertheless, the increase in the size of the proof mass comes at the cost of mechanical design complexity, where unwanted vibration modes could easily affect the measurement in the frequency band of interest. The finite element analysis is commonly used to estimate the dynamics of the sensor and prevent such problems from occurring.

3- Vacuum packaging is one method of increasing the Q-factor that also reduces the Brownian noise effect [15], but adds extra cost to the sensor [66].

The second area of MEMS accelerometer development is directed at the front-end electronic circuit. In most capacitive MEMS accelerometers, the electronic noise can dominate the overall noise, which has a significant effect on the output signal. Therefore, an electronic circuit designed with a noise floor lower than the mechanical noise is desirable [66] [67]. It is important to note that increasing the gain of the front-end electronics will also increase the electronic noise [61]. Capacitive MEMS front-end circuits can be categorized into three groups [67]: AC-bridge with voltage amplifier, transimpedance amplifier and switched-capacitor circuits. The choice of the appropriate circuit depends on the sensor fabrication technology and packaging. For example, surface micromachined technology offers the opportunity for monolithic integration of the accelerometer sensor along with the interface electronic circuit on the same chip, which

benefits from low electronic noise and less parasitic capacitance. However, due to the small proof mass, the resolution is limited and mechanical noise becomes an important degrading factor [61] [68]. On the other hand, bulk micromachining and SOI technology produce much greater proof mass, so high resolution can be achieved and low mechanical noise is induced. Nevertheless, the inability to achieve monolithic integration of the electronic circuit on same chip with bulk micromachined accelerometers is the main disadvantage; hence, electronic noise and parasitic capacitance impose their effect.

The third development area addresses the control-loop strategy, which can be either an open- or closed-loop system. Most commercial accelerometers with low to medium sensitivity employ the open-loop approach. High performance accelerometers are commonly designed to operate with closed-loop control. The latter benefit from wide bandwidth, better linearity and a large dynamic range when compared with the open-loop approach. However, these advantages come at the cost of circuit complexity. Two main closed-loop methodologies are employed in capacitive MEMS accelerometers. The first type is for analogue feedback accelerometers, which are usually prone to the pull-in problem (discussed in section 2.4.2.1). The second type is the digital feedback accelerometer, which employs the ΣΔM method. This type is immune to the pull-in problem (discussed in section 2.4.2.2) and is becoming increasingly popular. Nevertheless, the digital feedback approach encounters quantization noise, which elevates the noise floor above the mechanical noise. Recent research has focussed on the single-loop high-order EM-ΣΔM, which aims to decrease the quantization noise, but this approach comes at the cost of low dynamic range and instability problems when compared with the 2^{nd} order EM-ΣΔMs. The MASH EM-ΣΔM is constructed by cascading the low order ΣΔM (1^{st} or 2^{nd} order), so it benefits from the inherent stability. It also aims to reduce the quantization noise and retain the high dynamic range of the 2^{nd} order EM-ΣΔM.

The focus in this research is on the control-loop strategy, and in particular, the implementation of the MASH EM-ΣΔM. In the following literature, the EM-ΣΔM will be discussed, starting with a second order EM-ΣΔM as a simple form of implementation, where in such systems the quantization noise becomes a performance limiter; therefore, the need for high order EM-ΣΔM will be addressed to minimise the quantization noise. The implementation of the high order EM-ΣΔM will be discussed in single loop architecture and in MASH architecture.

3.2 Second Order Electromechanical Sigma-Delta Modulator

The first use of the second order ΣΔM to control a capacitive MEMS accelerometer was reported by *Henrion* et al. in 1990 [20]. The inertial sensor is usually modelled as a second order lumped-model, as discussed in section 2.2, which could be considered as two cascaded integrators within the inertial sensor's operating frequency band. Several publications were reported [69-73] on the use of the second order ΣΔM to control the capacitive MEMS inertial sensors. Figure 3-1 shows a block diagram of an accelerometer embedded in a second order ΣΔM control loop [70]; the output of the sensor is fed to a charge integrator and then to a comparator, and the digital output is fed back to the electrostatic force electrodes. This sensor was fabricated with surface micromachined technology, which has a device thickness of 2 μm. The on-chip electronic circuit detected the out-of-plan displacement when the sensor was put under acceleration, and achieved a noise floor of 1.6 $mg/\sqrt{Hz}$.

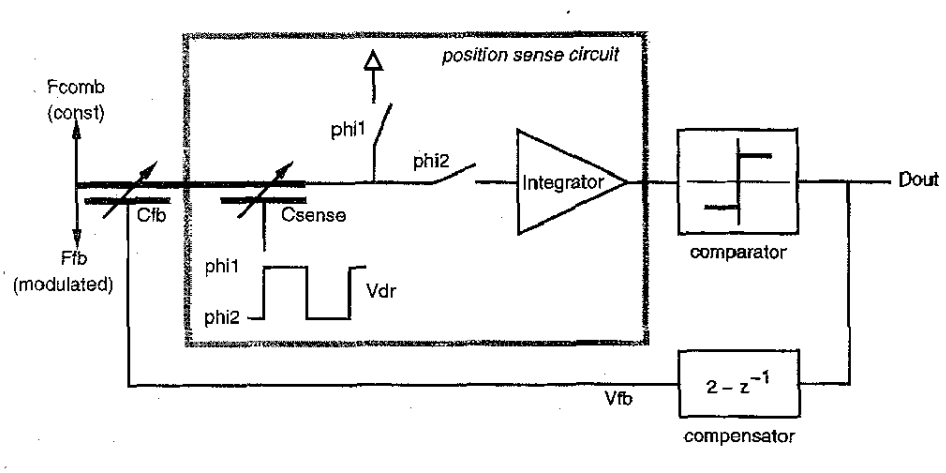


Figure 3-1: Digital force feedback accelerometer block diagram, reproduced from [70].

The digital output advantage of the second order EM-ΣΔM comes at the cost of quantization noise. Therefore, the main objective of the modulator is to provide a measurement in a controllable and stable fashion with minimum quantization noise, i.e. the quantization noise must be lower than other noise sources, e.g. electronic noise. The simple approach to further reduce the quantization noise is to increase the OSR, however, the interaction between the quantization noise and the electronic noise increases the overall noise floor of the second order EM-ΣΔM [22] [74]. Another approach is to employ a multi-bit EM-ΣΔM with a multi-bit feedback force. However, this approach comes with

two main disadvantages. First, using a multi-bit feedback force is categorized under the analogue electrostatic feedback force, which comes across some disadvantages such as the pull-in problem, as discussed in section 2.4.2.1. Second, to achieve lower quantization noise shaping from the multi-bit feedback structure, linear feedback actuation voltage levels are required, which requires more accurate electronics and adds more complexity to the design [75-77]. Furthermore, it is considerably easier to build a single-bit quantize than a multi-bit quantizer both in terms of power consumption and quantization levels [77]. A single-bit quantizer is intrinsically more linear than a multi-bit quantizer as it deals with only two feedback voltage levels [78] [79]. Later in chapter 7, a dual quantization method will be presented that takes advantage of the reduced quantization error of the multi-bit quantizer in the forward path and the inherent linearity of the single-bit quantizer in the feedback path.

The third approach is to employ a high order EM-ΣΔM which will be discussed in the following sections.

3.3 Higher Order Single Loop EM-ΣΔM

Single loop higher order EM-ΣΔMs are realised by adding electronic integrators with their associated gains after the second order sensing element. As the order of EM-ΣΔM increases, the loop filtering and the noise shaping performance increase, leading to strong suppression of the quantization noise floor within the bandwidth of interest, compared to the second order EM-ΣΔM. However, as in the electronic ΣΔM the increased performance of the high order EM-ΣΔM encounters the closed loop instability problem and a low overload input level [25] compared with the second order EM-ΣΔM.

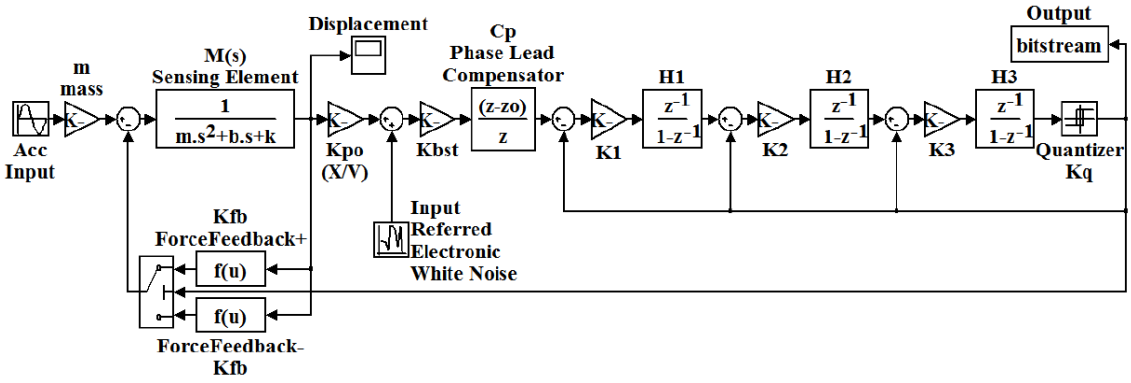


Figure 3-2: Fifth order EM-ΣΔM Simulink model using the multi feedback approach, reproduced from [74].

For example, to construct a single loop fifth order EM-ΣΔM, three electrical integrators (H_1, H_2 and H_3) with their associated gains (K_1, K_2 and K_3) are added in the forward path [74] as depicted by the Simulink model in Figure 3-2. Using the same linearization assumption that was discussed in section 2.5, the signal transfer function (STF), the quantization noise transfer function (QNTF) and the electronic noise transfer function (ENTF), can be derived as follows [74]:

$$STF = \frac{mk_m \prod_{i=1}^{3} K_i H_i / K_{fb}}{1 + k_m \prod_{i=1}^{3} K_i H_i + K_q \sum_{i=1}^{3} \prod_{j=i}^{3} K_j [H_j]} \quad 3.1$$

$$QNTF = \frac{1}{1 + k_m \prod_{i=1}^{3} K_i H_i + K_q \sum_{i=1}^{3} \prod_{j=i}^{3} K_j [H_j]} \quad 3.2$$

$$ENTF = \frac{k_m \prod_{i=1}^{3} K_i H_i / (K_{fb} M K_{po})}{1 + k_m \prod_{i=1}^{3} K_i H_i + K_q \sum_{i=1}^{3} \prod_{j=i}^{3} K_j [H_j]} \quad 3.3$$

where $k_m = K_{fb} M K_{po} K_{bst} C_p K_q$ which is function of the feedback conversion gain K_{fb}, the sensing element transfer function M, the pick-off and boost gain K_{po} and K_{bst}, the compensator transfer function C_p and the quantization gain K_q.

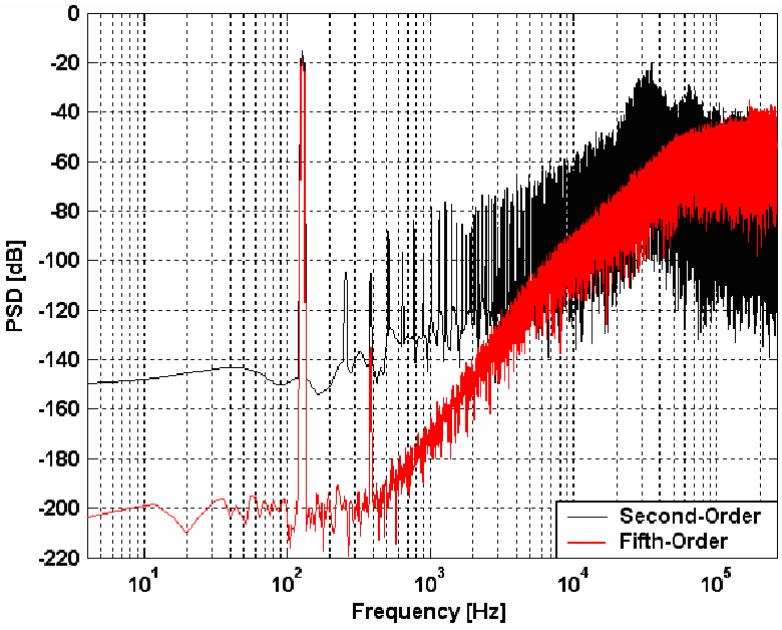


Figure 3-3: A comparison in term of the noise shaping between second and fifth order EM-ΣΔM, the OSR = 256, and bandwidth = 1024 Hz, reproduced from [74].

The Simulink results in Figure 3-3 show the noise shaping comparison between the second order (black) and fifth order (red) EM-ΣΔM. This diagram demonstrates that the

fifth order has improved the noise floor of the accelerometer by about -50 dB, compared with second order architecture.

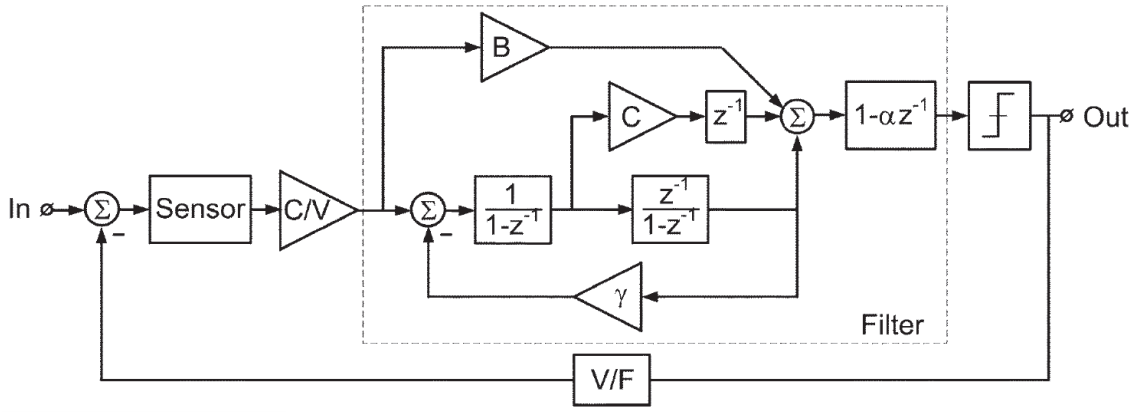


Figure 3-4: Fourth order EM-ΣΔM, reproduced from [22]

Several publications for high order EM-ΣΔM have been reported. In 2005 [22], a fourth order EM-ΣΔM was reported, which can be configured for a lateral accelerometer with 6 kHz resonant frequency, resulting in a resolution of $150\ \mu g/\sqrt{Hz}$. The fourth order prototype was fabricated in a standard 0.5 µm complementary metal oxide semiconductor (CMOS) process. Figure 3-4 shows a block diagram for the fourth order EM-ΣΔM, which was designed for both a gyroscope and an accelerometer. By simply assigning $\gamma = 0$, the architecture can operate as a fourth order low pass ΣΔM accelerometer.

In 2005 [21], *Dong* et al. implemented a fifth order EM-ΣΔM with an in-plane accelerometer. The experimental results presented showed a noise floor of -80 dB, as shown in Figure 3-5. The performance of this accelerometer was degraded by the electronic noise floor injected by the pickoff circuit. An improvement for the previous fifth order e EM-ΣΔM was reported in 2006 [80]. The author analysed the nonlinearity of the applied electrostatic force with respect to the mass position during the feedback operation. An effective linearization scheme was proposed to increase the SNR. The circuit was implemented on a PCB with the fifth order configurations and a measured noise floor of -110 dB was achieved.

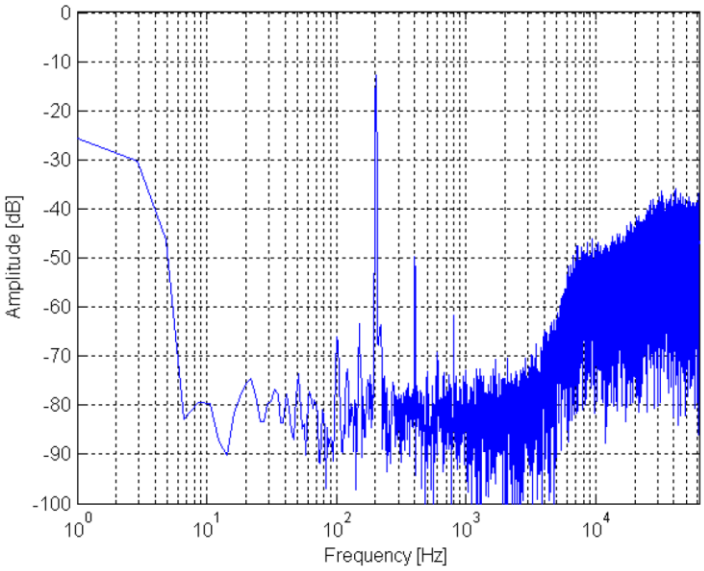


Figure 3-5: Post process noise shaping of the fifth order electromechanical ΣΔ modulator, with input acceleration of -10 dB amplitude and 200 Hz frequency, reproduced from [21].

A dual quantization technique for the purely electrical ΣΔM [31-33] has been applied for high order EM-ΣΔM [81-86]. Such systems are comprised of an analogue and digital unit [82], as shown in Figure 3-6. In the analogue unit, the output of the sensing element is acquired and then digitised using a multi-bit ADC. The multi-bit digital output is then fed to the digital unit to perform high order filtering and to complete the EM-ΣΔM structure. The output of the digital unit is generated by a single bit quantizer. The digital part is implemented using the field programmable gate array (FPGA). Therefore, traditional analogue op-amp integrator circuits are replaced with digital integrators. The FPGA offers the designer extra flexibility to tailor the high order EM-ΣΔM.

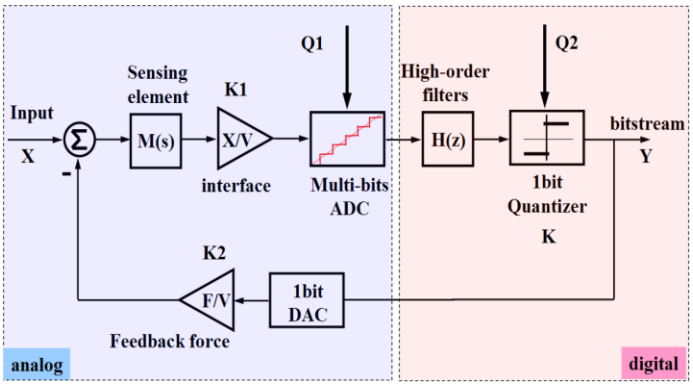


Figure 3-6: Fifth order ΣΔM with FPGA technology, it shows the analogue and digital units with dual quantization architecture, reproduced from [82].

A fifth order ΣΔM accelerometer implemented with the above FPGA configuration [82] has 11 g full scale input for a bandwidth up to 300 Hz. The system was experimentally tested in a quiet room and the measured noise spectrum, as shown in Figure 3-7, has a noise floor down to -120 dBg/√Hz. Moreover, the FPGA is utilised to implement an overload recovery mechanism [83], which switches to a second order EM-ΣΔM architecture in case of high-g shock.

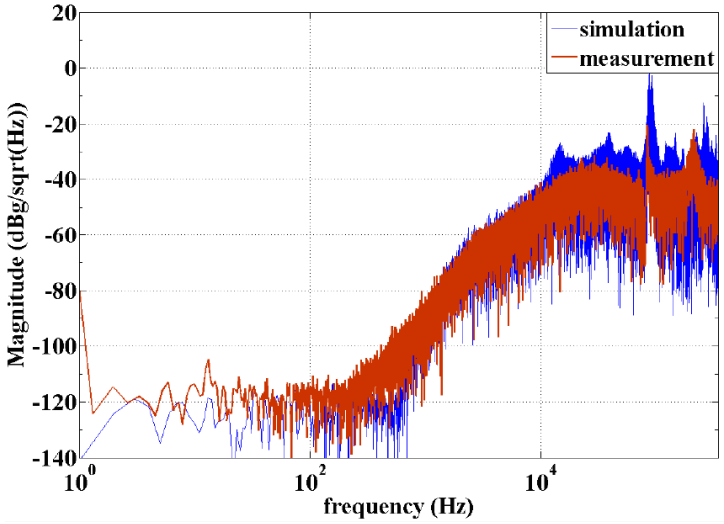


Figure 3-7: Noise shaping measurement of a fifth order ΣΔM accelerometer implemented using FPGA, reproduced from [82].

Amini et al. reported in [87] the results of a study involving a hybrid switched-capacitor (SC) charge amplifier with a fourth-order ΣΔM CMOS accelerometer. The structure of this accelerometer type is shown in Figure 3-8. The change in capacitance of a low-Q accelerometer is directly measured by the front-end SC. The back-end ΣΔ is used to apply digital feedback force and close the loop. The system achieved a dynamic range of 95 dB, a noise floor of -87 dBV/√Hz and a bias stability of 8 μg for a period of 12 hours.

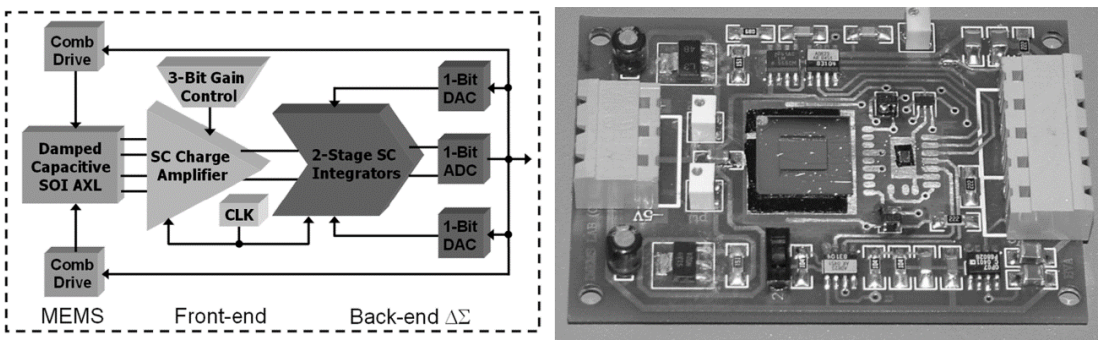


Figure 3-8: Block diagram of a fourth-order ΣΔ CMOS SOI accelerometer (left) and a hardware implementation of the system (right), reproduced from [87].

3.4 Single-Loop High Order Electromechanical ΣΔM Design Methodologies

Designing a high order EM-ΣΔM is not a trivial task. One reason for this is that the ΣΔMs with an order higher than two are only conditionally stable [88]. Moreover, the second integrator of the EM-ΣΔM is not accessible; this introduces a phase lag, hence, maintaining a stable control loop is difficult [89]. Several design methodologies for the high order EM-ΣΔM are reported, some of which take advantage of a well-established method in the electronic ΣΔM, such as the root locus approach [55], and apply the same principles in the EM-ΣΔM [21] [74] [80] [90]. *Dong* described this method in [74]. The main design procedure can be summarised as follows:

1- Find the optimal coefficient for specific N^{th} order electronic ΣΔM topology to ensure maximum stability and performance based on a stability criteria, i.e. NTF gain < 1.5 [91].
2- A second order EM-ΣΔM has to be designed in parallel for maximum performance and stability; a lead compensator may be needed in this process.
3- The first integrator in the purely electronic ΣΔM is replaced with the sensing element and compensator, and the DAC is replaced with the electrostatic feedback force, to build $(N+1)^{th}$ order EM-ΣΔM.

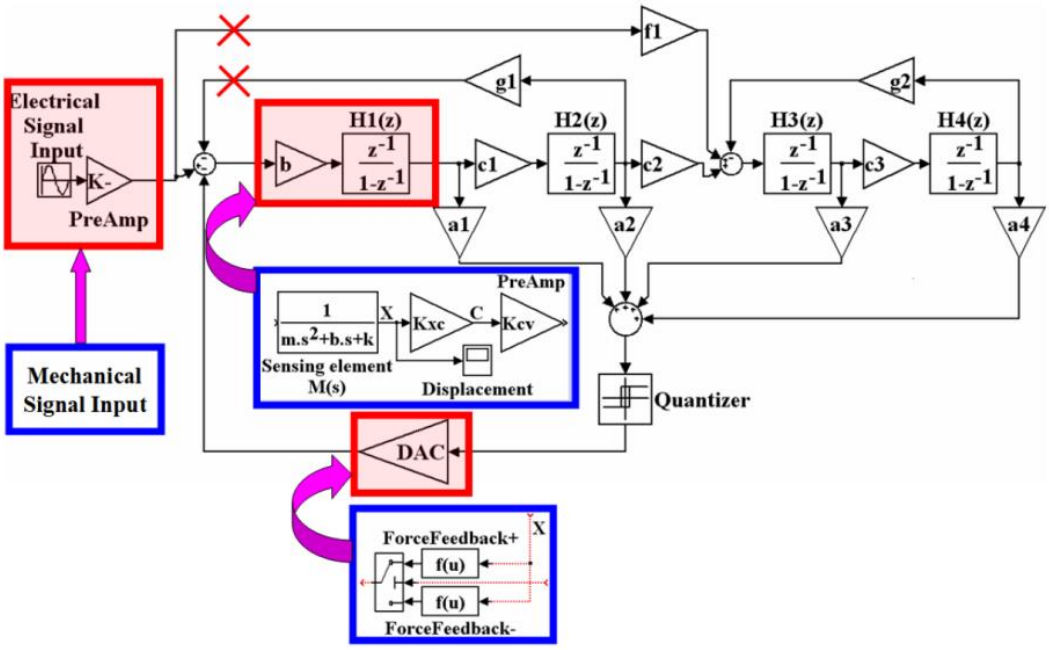


Figure 3-9: High order EM-ΣΔM design as described in [74].

Figure 3-9 shows the conversion of a fourth order electronic ΣΔM to a fifth order EM-ΣΔM. A fifth order ΣΔM accelerometer has been successfully implemented using this method in [90]. Figure 3-10 shows the measured frequency spectral of the system. It was excited with a 1g acceleration input, and the performance achieved a noise floor of -120 dBV/√Hz.

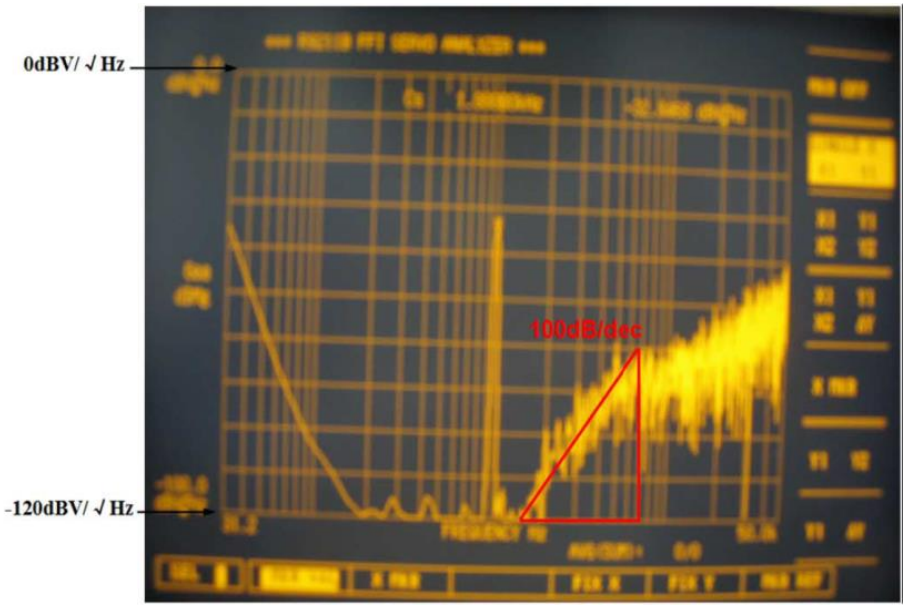


Figure 3-10: A fifth order ΣΔM accelerometer measurement, the PSD shows a 1 g response at 1 kHz, reproduced from [90].

In the previous method, one integrator is replaced with two mechanical integrators that have an inaccessible inner node. Therefore, the overall structure misses one degree-of-freedom to have full control of the loop roots. A systematic method to design high order EM-ΣΔM is reported in [89] [92] [93]. The author suggested an unconstrained architecture so that the inner node of the mechanical sensing element is not required. This method provides sufficient degrees-of-freedom, allowing the NTF poles to be placed at arbitrary positions. In accordance with Figure 3-11, this method can be summarised as follows [89]:

1- The design starts with an unconstrained purely electrical N^{th} order ΣΔM architecture, as shown in Figure 3-11 (a).
2- By applying block diagram reconstruction, the feedback path of the inner node (after the first integrator) is converted to be a feedforward path, as shown in Figure 3-11 (b).

3- Now, the feedforward path can be shifted to the right as shown in Figure 3-11 (c). This shift procedure results in two important features: First, it still preserves the equivalence between (b) and (c), and second, it eliminates the need for the feedback path at the inner node.

4- The first two electrical integrators can be replaced with two mechanical integrators, as shown in Figure 3-11 (c). The resulting structure maintains the same order as the original purely electrical ΣΔM order, and does not need compensation circuit for loop stability.

5- Finally, the root locus approach, proposed by *Schreier* [94], used for the purely electrical ΣΔM can now be followed to design the high order EM-ΣΔM.

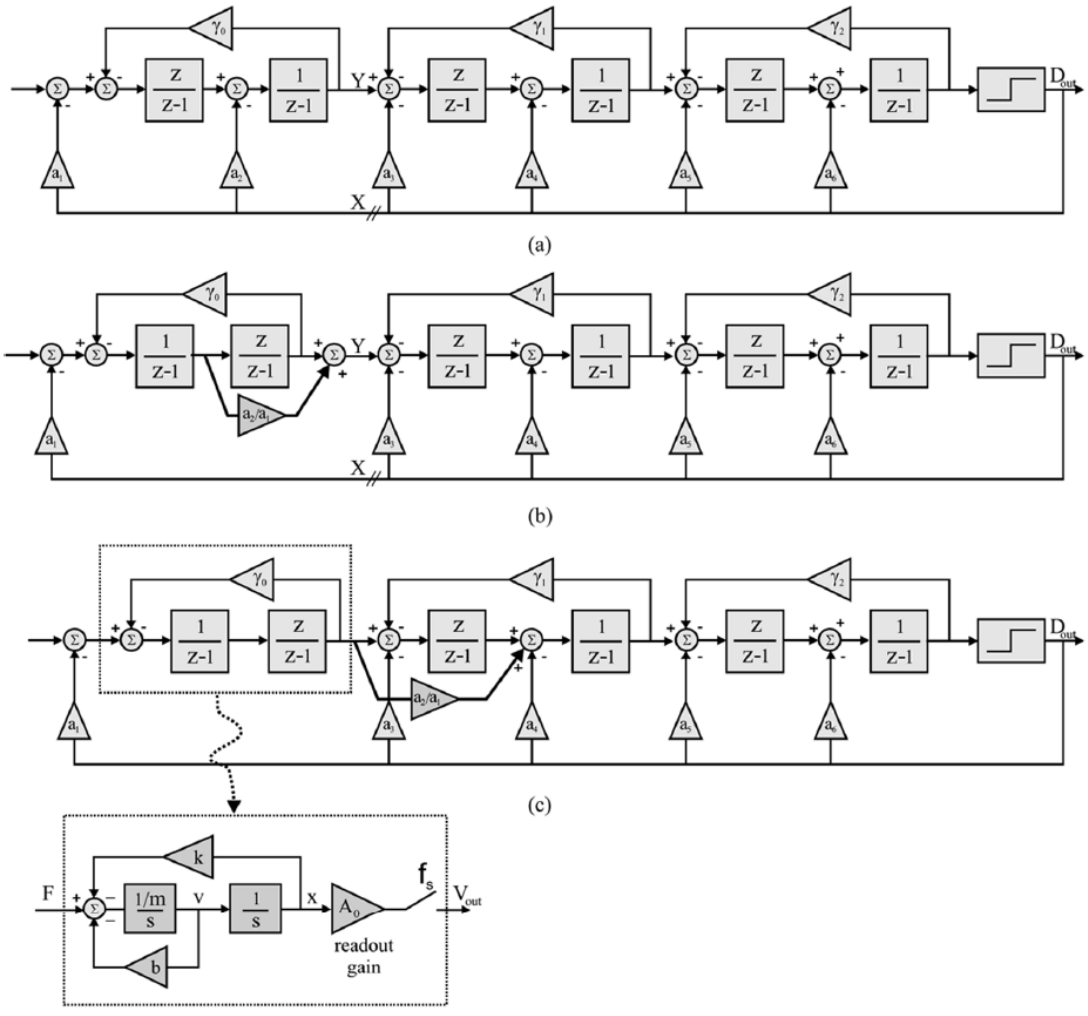


Figure 3-11: Converting a purely electrical ΣΔM to an unconstrained EM-ΣΔM. (a) Starting with a purely electrical ΣΔM. (b) After the first integrator, performing a conversion of the inner feedback path to a feedforward path. (c) Shifting the feedforward path to the right and replacing the two electrical integrators with two mechanical integrators, reproduced from [89].

In 2011, this method was reported in [23] to construct an accelerometer that is hybrid integrated with fourth order unconstrained ΣΔM CMOS readout chip. The accelerometer was micromachined with a structure thickness of 35 µm, and the electronic circuit comprised a reconfigurable readout circuit that had the ability to fulfil the required interface for a specific accelerometer. The readout circuit with the ΣΔM was implemented using ASIC technology, as shown in Figure 3-12 (top). The accelerometer achieved a maximum full scale ratio of ±40.3 g, and a noise floor of 11.3 µg/√Hz as shown in Figure 3-12 (bottom).

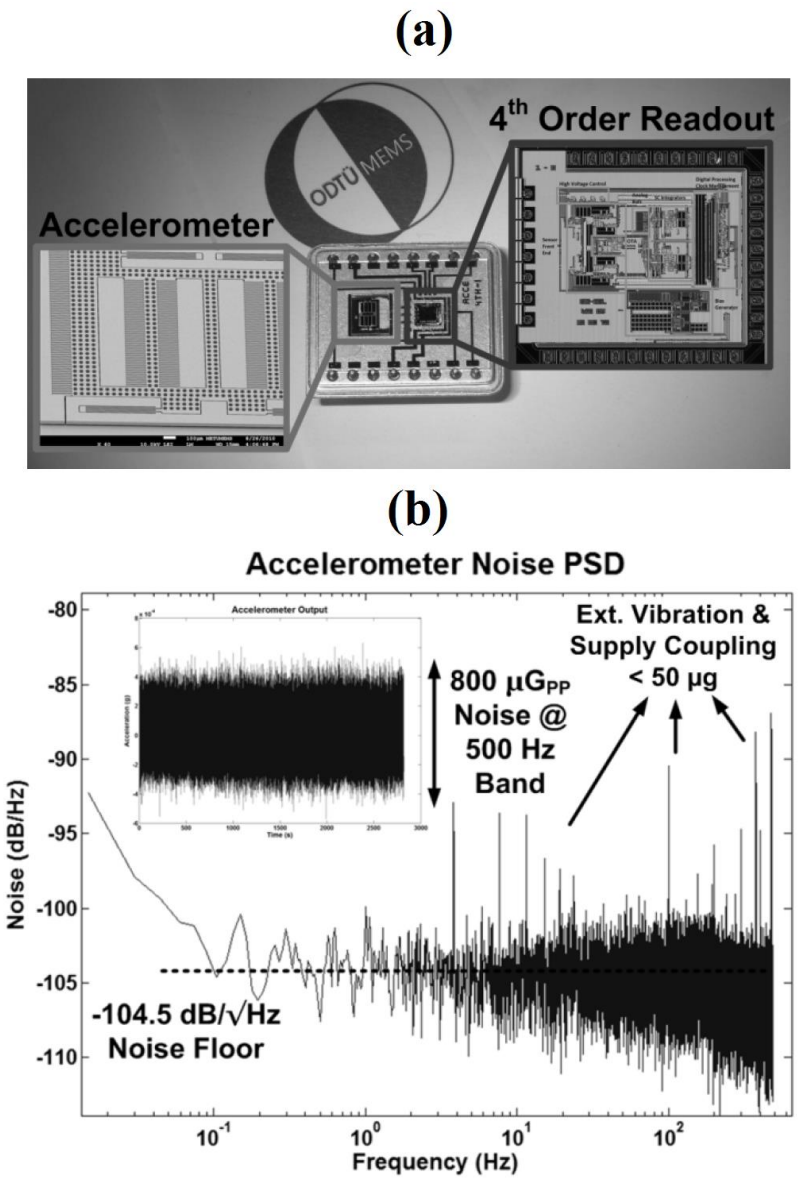


Figure 3-12: (Top) Photo of the micromachined accelerometer with fourth order ΣΔM readout implemented in ASIC and (Bottom) frequency spectral measurement of the system, reproduced from [23].

Another method for designing a high order EM-ΣΔM system is reported in [95], it is also based on the NTF analysis. It works by equating the NTF of the EM-ΣΔM with a reference NTF of a purely electrical ΣΔM. The design starts in the discrete-time domain using a linear model of both modulators. Using the systematic transformation approach from the discrete-time to the continuous-time domain [96] and following the root locus approach suggested by *Schreier* [94], the equivalence between the two NTFs is achieved. Because it is not possible to modify the mechanical parameters, any difference between the two NTFs will be balanced using the electronic part of the EM-ΣΔM.

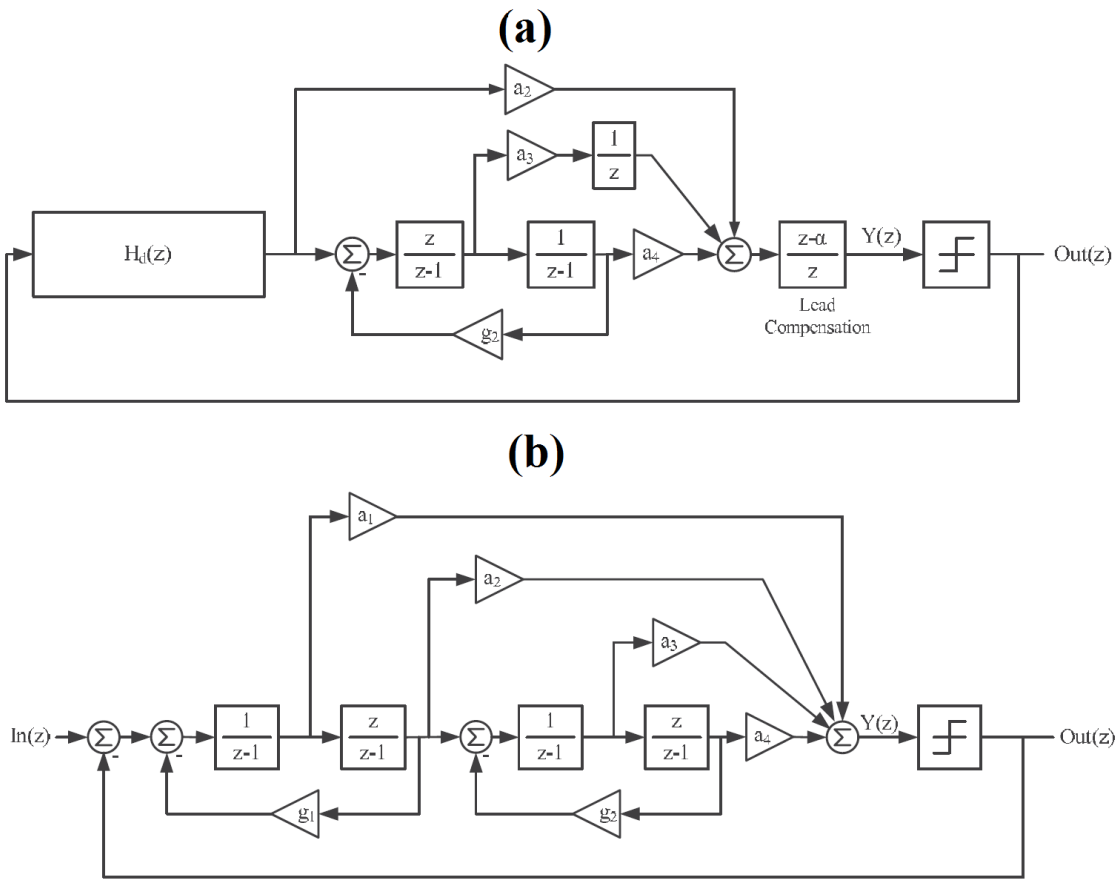


Figure 3-13: (a) 4th order EM-ΣΔM, and (b) 4th order electrical ΣΔM, reproduced from [95].

Figure 3-13 (a) shows the desired 4th order EM-ΣΔM of the feed-forward topology, which has a lead compensator for loop stabilisation. (b) is the reference purely electrical 4th order ΣΔM with the feed-forward topology. The simulation analysis using this method shows promising results in Figure 3-14, where the EM-ΣΔM achieves high SNR as conventional electrical modulator and a comparable input stable range.

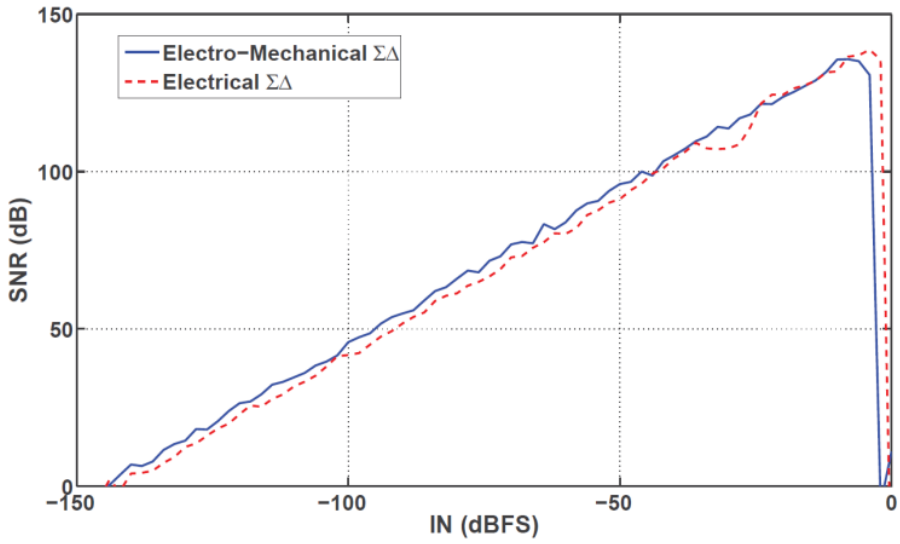


Figure 3-14: Simulation analysis between SNR and input amplitude for 4th order EM-ΣΔM (in blue line) and purely 4th order electrical ΣΔM (in red/dashed line), reproduced from [95].

All previous design methodologies rely on the model linearization of the high order EM-ΣΔM systems. The linearization approach in the EM-ΣΔM has some degree of inaccuracy in terms of the performance and the control loop stability, especially when designing a high order EM-ΣΔM [97]. Moreover, the linear model does not include the effect of the nonlinear behaviour between the proof mass displacement and the applied electrostatic feedback force [98]. These two nonlinear implications are addressed in a novel design methodology of high order EM-ΣΔM with a nonlinear model suggested by *Wilcock* and *Kraft* [98]. The design is based on the genetic algorithms (GA) and Monte Carlo simulations [99]. The design is implemented in MATLAB using predefined functions that can be customised to fit specific applications. This design methodology is applicable for any topology with any order, either an accelerometer low-pass or a gyroscope band-pass EM-ΣΔM.

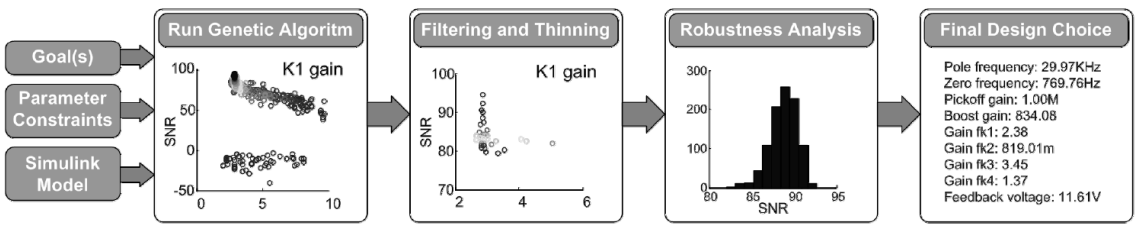


Figure 3-15: Generic progression flow for the nonlinear GA design methodology of the EM-ΣΔM, reproduced from [98].

Figure 3-15 shows the process data flow of the GA design methodology using a nonlinear model. Initially, the user has to set up and provide the GA algorithm using the following: i) The parameterised Simulink model of the EM-ΣΔM, ii) Model parameters and constraints and iii) Goal parameters in order to discriminate valid results and output the optimised ones. Goals can be the maximum SNR value calculated from the power spectral density of the output bitstream, or the minimum RMS value of the proof mass displacement in a closed loop operation.

Since the output result of the GA step consists of a large number of equally optimal solutions, the filtering and thinning procedure is used to discard the solutions that do not satisfy the goals specified by the user. These solutions are processed based on a robustness analysis step using the Monte Carlo technique. This step performs a statistical simulation in order to sort out the elite solutions based on the maximum performance and control stability in the presence of sensing element parameter variations.

A fourth order ΣΔM accelerometer is designed using the GA algorithm and hardware implemented in [28]. The fourth order ΣΔM accelerometer model is shown in Figure 3-16. The GA algorithm was set to find the optimal design parameter that includes the compensator, pickoff and boost gain, and the inner feedback path gain constants.

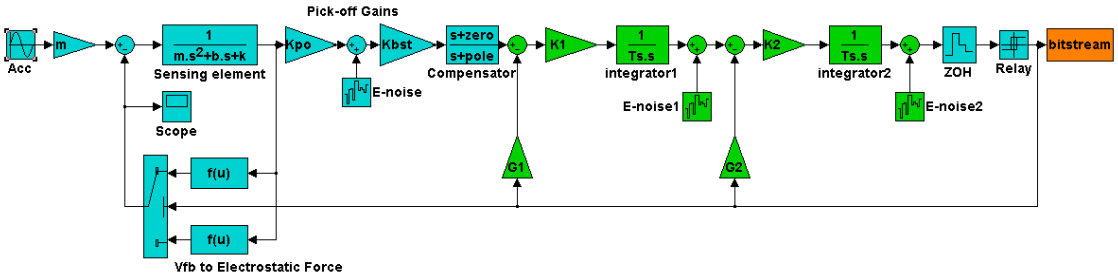


Figure 3-16: Fourth order ΣΔM accelerometer Simulink model, reproduced from [28]

The measured noise shaping of the fourth order ΣΔM accelerometer is shown in Figure 3-17. The system was tested with 0.6g acceleration input at 550 Hz. It achieved a bandwidth of 1 kHz and a noise floor level of -110 dB (equivalent to 19 μg/√Hz).

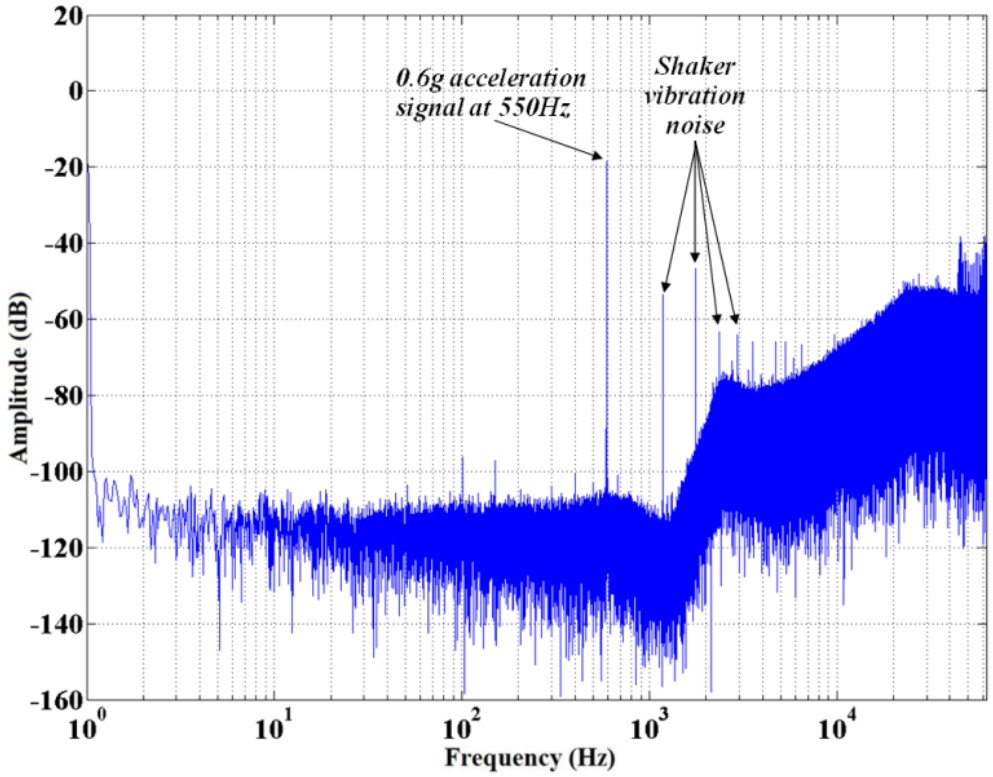


Figure 3-17: Measured noise shaping of the fourth order ΣΔM accelerometer with 0.6 g acceleration at 550 Hz, reproduced from [28].

3.5 Multi Stage Noise Shaping (MASH) Sigma Delta Modulator

Electromechanical MASH inferred from electronic ADC for which the MASH concept was first introduced. It typically employs cascaded ΣΔMs of first and second order stages along with digital filters [100-104]. The MASH structure attains the performance of the high order single loop ΣΔM and also overcomes the potential instability issue. However, MASH structure requires precise digital parameters matching with the analogue components, in order to cancel the quantization noise [105-108].

The implementation of a MASH ΣΔ modulator is usually achieved by employing switched-capacitor circuits, which demonstrate good matching between the digital filter parameter and the analogue ones. However, the switched-capacitor technique uses a sampler at the modulator input; thus, it requires an anti-alias filter [109]. On the other hand, a continuous time ΣΔ modulator benefits from the anti-alias filtering introduced by the loop filter, which in turn simplifies the electronic design and reduces power dissipation [110]. Breems et al. [111] have reported an analogue-to-digital convertor (ADC) with a continuous-time MASH22 ΣΔ modulator, as shown in the block diagram

in Figure 3-7. It also provides anti-alias filtering with over 50 dB suppression in bandwidths between 150 and 170 MHz, and a dynamic range of 67 dB. Breems et. al. also implemented an adaptive calibration technique to solve the problem of matching the analogue and digital filter coefficients [112]. By calculating the variance of the bitstream, after the decimation, a measure of the in-band noise power can be obtained. During the calibration phase, when a mismatch between the digital filter and the analogue components occurs, the quantization noise power increase is reflected by high variance value. Therefore, by adjusting the calibration gain value, matching can be achieved and the quantization noise power can be reduced. At this stage, the variance reaches its minimum and the calibration phase ends.

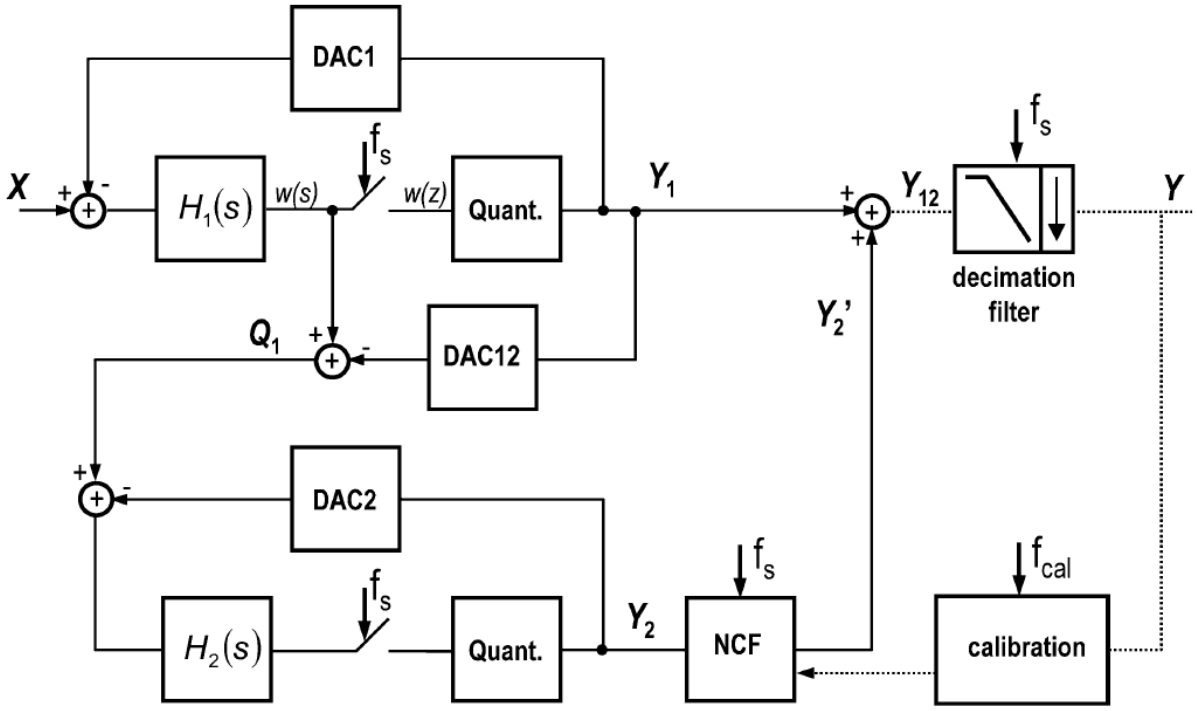


Figure 3-18. Continuous-time MASH22 ΣΔ modulator with digital filter calibration, reproduced from [111].

Kawahito et al. [113] reported a 2-D CMOS microfluxgate sensor embedded in a MASH11 ΣΔM interface and control circuit to measure weak magnetic fields and provide digital output. Figure 3-19 shows a block diagram of the system, where two orthogonal fluxgate sensors measure magnetic induction components parallel to the sensor's surface. A set of pickoff coils in the sensor is designed to output a voltage signal proportional to the change in the external magnetic field. The signal is then amplified and filtered by the front-end circuit. A switched-capacitor integrator is added to perform the first order loop

filtering of the ΣΔM. Next, a comparator generates feedback loop pulses that are fed to the feedback coils of the sensor to close the loop in the magnetic field domain, which counteracts the external magnetic change. A second loop is added to the first order ΣΔM to construct a MASH11, thus, a second-order noise-shaping performance is achieved and the pattern noise is reduced as a result of the MASH structure. The system achieved nonlinearity of less than 1.5 μT within a range of ±50 μT, and the angular resolution for the 2-D vector sensor was less than 4^{o} for the 50 μT magnetic induction measurement.

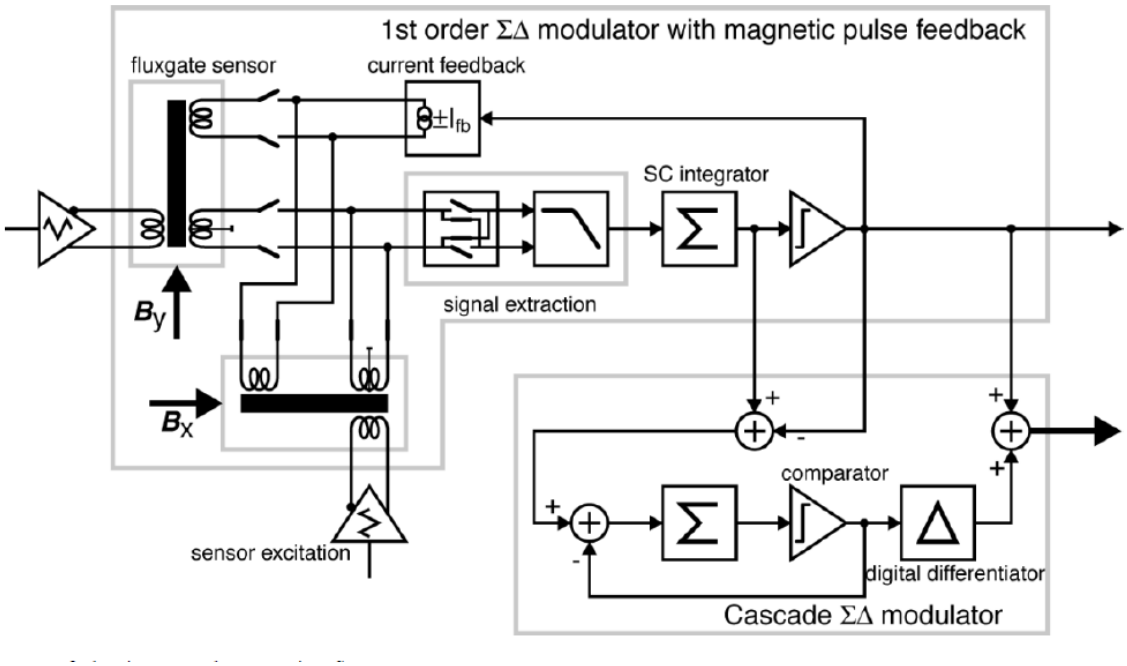


Figure 3-19. Block diagram of the 2-D microfluxgate with MASH11 ΣΔM interface and control circuit (reproduced from [113]).

In contrast to the EM-ΣΔM, where the mechanical sensor contributes to the loop filter, this fluxgate structure does not contribute to the ΣΔM loop filter; hence, an integrator is added to the first loop to form the first-order ΣΔM.

The Sturdy MASH (S-MASH) is another ΣΔM structure that retains the MASH performance and does not require digital filters, thus, the necessity for analogue components and digital filters matching is eliminated [114-116]. Since the S-MASH requires access to the second integrator of the first stage, the implementation of this structure is not possible for electromechanical ΣΔM, as the first stage has an inaccessible node within the sensing element. Therefore, the use of the conventional MASH structure for the MEMS devices will be addressed in this work.

The EM-MASH was first proposed by *Kraft* et al. in 2001 [117]. The idea is to cascade the 2nd order EM-ΣΔM that includes the sensing element with a purely electronic ΣΔM. The quantization noise is fed into the second modulator, which constructs a two stage MASH21.

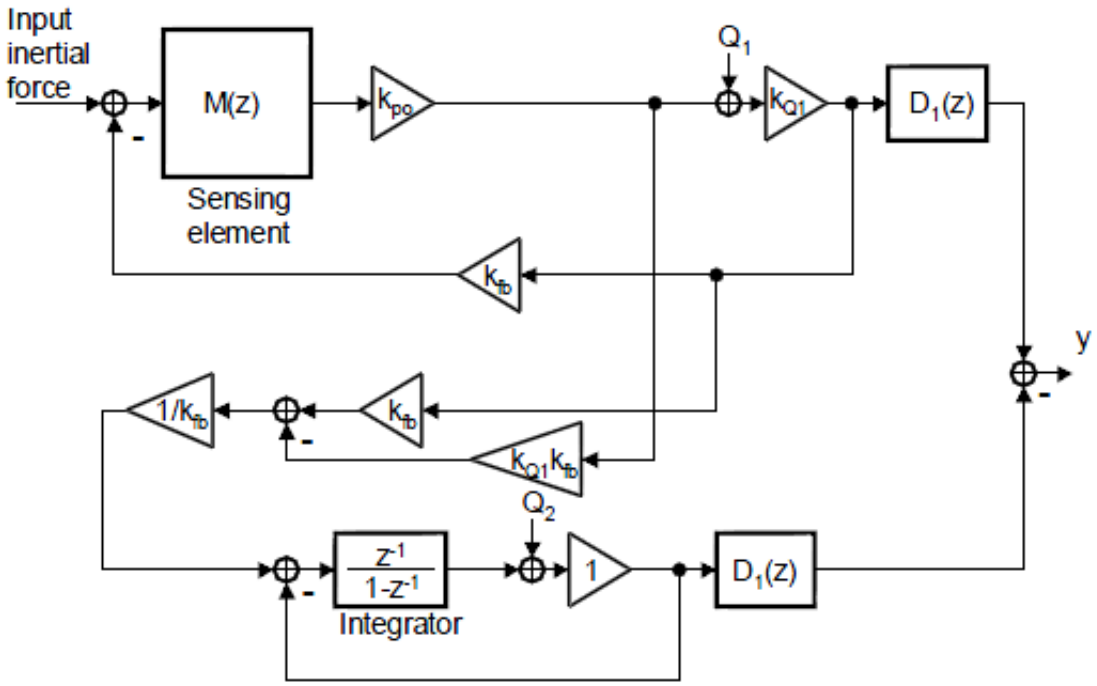


Figure 3-20: Linearized block diagram of an electromechanical MASH, reproduced from [117].

Figure 3-20 shows the linear model for the MASH21 that was suggested by *Kraft* et al. The digital filters are designed to cancel the quantization noise from the first stage. The noise transfer function of the first stage can be given by:

$$NTFQ_1 = \left(\frac{KQ_1D_1}{1 + M\,K_{po}KQ_1K_{fb}} - KQ_1Z^{-1}D_2\right) \qquad 3.4$$

If the digital filters are chosen as:

$$D_1 = z^{-1} \quad \text{and} \quad D_2 = \frac{1}{(1+M\,K_{po}KQ_1K_{fb})}$$

The noise transfer function $NTFQ_1$ will equal zero, hence the quantization noise Q_1 from the first stage will be removed.

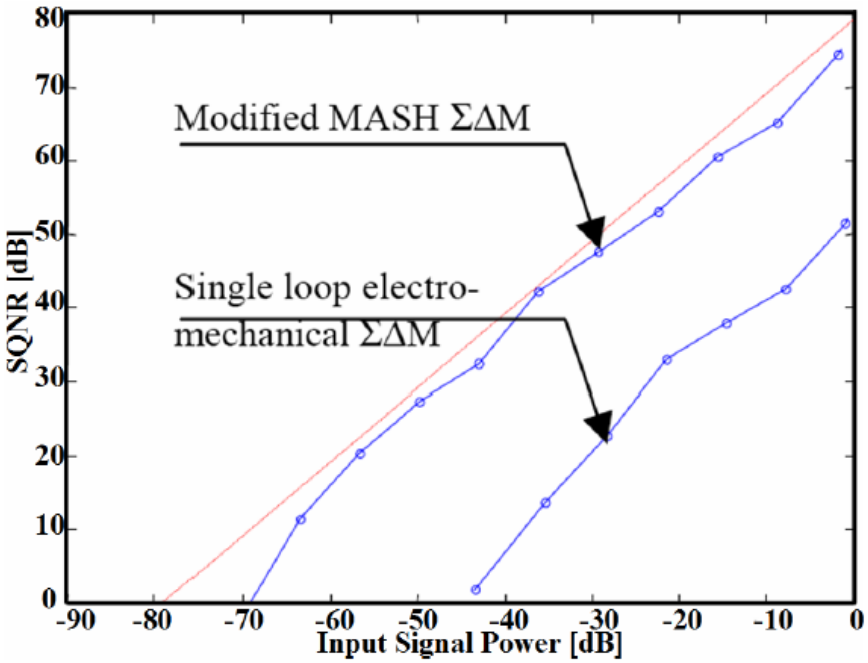


Figure 3-21: Input power versus SNR for the modified MASH compared with a single loop 2nd order mechanical ΣΔM and ideal 2nd order ΣΔM, reproduced from [117].

The simulation results are shown in Figure 3-21. *Kraft* demonstrated that, the MASH21 architecture provides a noise shaping similar to the ideal second order ΣΔM and improves the SNR of the 2nd order EM-ΣΔM by nearly 30 dB. However, the structure is sensitive to parameters variations which causes a leakage of the quantization noise and degrades the modulator performance.

Several MASH orders, including MASH21, MASH22 and MASH211 were examined by *Mokhtari* [118]. Their performance reached a maximum SNR of 75 dB with an OSR of 64, as shown in Figure 3-22, which is similar to the performance of an ideal second order ΣΔM. However, *Mokhatri* concluded that the performance of higher order models (MASH22 and MASH211) does not exceed that of the simple MASH21, as shown in Figure 3-22. He stated that the lack of the performance was due to the low frequency gain of the electromechanical sensor and the presence of AC-tones in the baseband. However, it will be shown by means of simulation and theoretical results presented in chapter 4 that the performance of the MASH-ΣΔM will be enhanced as the order is increased, whether by increasing the number of stages or by increasing the order of each stage. The design of electromechanical MASH based on the linear analysis will be addressed in the next chapter. It will be shown that the linear approximation works effectively for the MASH as it benefits from its low order stages.

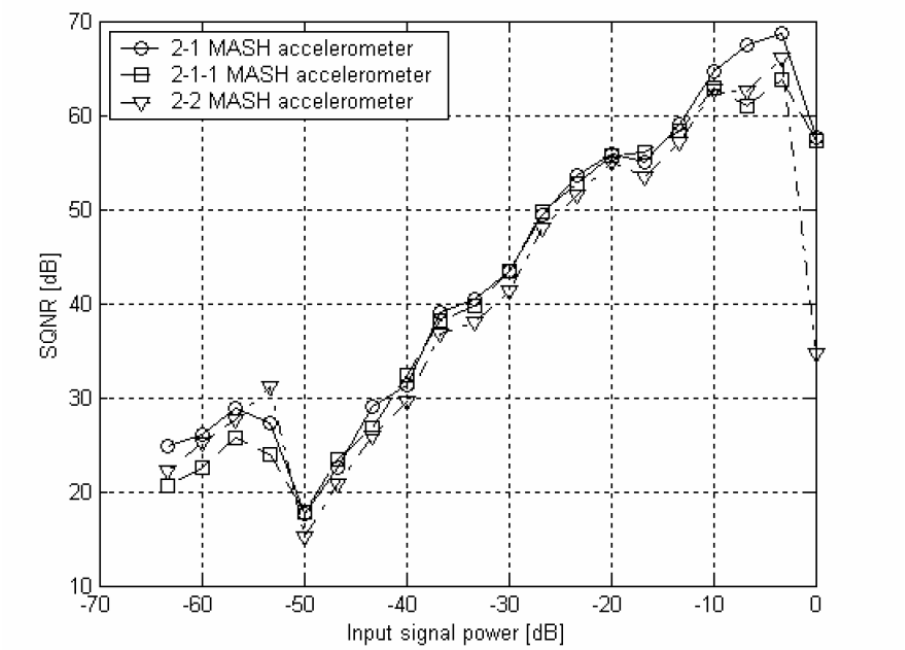


Figure 3-22: Performance comparison of MASH21, MASH22 and MASH211 for various amplitudes of the input signal, reproduced from [118].

3.6 Summary

Inertial MEMS sensors operated in the closed loop mode are realised by either the analogue or digital approach. Both approaches offer the inertial MEMS sensor important advantages, such as enhanced linearity and an increase in the dynamic range and operation bandwidth. The ΣΔM technique is initially applied to construct the analogue to digital convertors; however, this technique has been successfully employed in inertial MEMS sensors. The EM-ΣΔM offers direct digital output and does not suffer from the pull-in issue encountered in analogue feedback sensors.

In this chapter, the single loop EM-ΣΔM was discussed, starting with the second order EM-ΣΔM in a simple form. It was clear that the quantization noise was a key concern in high performance accelerometers; therefore, high order EM-ΣΔMs were proposed that offer suppressed quantization noise. Several design methodologies were suggested for high order EM-ΣΔM. One method [74] uses the purely electrical ΣΔM design technique to optimise the design and then replaces the first integrator with the electromechanical second order model. This method requires a compensation circuit in order to stabilise the loop. Another method [89] is to reconstruct the purely electrical ΣΔM to have two first integrators with no inner node access, which yields to a compatible replacement of the

two first electrical integrators with the electromechanical second order model; thus, stability and enhanced performance can be achieved. The previous design methodologies use classical control theory, which requires linearized model of the high order EM-ΣΔM. However, the linearization approach in the EM-ΣΔM has some degree of inaccuracy in terms of the performance and the control loop stability, especially when designing a high order EM-ΣΔM. A novel method for the high order EM-ΣΔM design proposed by *Wilcock* [98] was discussed, which uses the genetic algorithms (GA) and Monte Carlo simulations to obtain the optimum design parameters of the nonlinear Simulink model of the high order EM-ΣΔM.

Finally, a discussion on the MASH ΣΔM was addressed. The EM-ΣΔM with such architecture has seldom been investigated, despite having the considerable advantages of inherent stability, a high overload input and a high dynamic range when compared with the single loop ΣΔM. Nevertheless, the MASH is sensitive to component and parameter tolerances.

Chapter 4: Design and Simulation of MASH ΣΔ Modulators for Inertial MEMS Capacitive Accelerometer

4.1 Introduction

In this chapter an illustration on the MASH ΣΔM operation concept and the analytical description of the system will be discussed. Furthermore, an explanation of the design procedure of the MASH and the signal to noise ratio (SNR) estimation are addressed.

The theoretical analysis of the MASH with different orders is performed using MATLAB and Simulink. The analysis will also be carried for the noise shaping, overload point and parameter sensitivity for MASH architecture.

4.2 Analytical Investigation

The structure of the EM-MASH-ΣΔM is divided into analogue and digital circuits as shown in Figure 4-1. The analogue circuit contains the first stage ΣΔM; this includes the sensing element and it is analogue interface parts, i.e. pickoff circuit, low pass filter, and feedback force circuits. It also contains the pure electronic ΣΔMs stages.

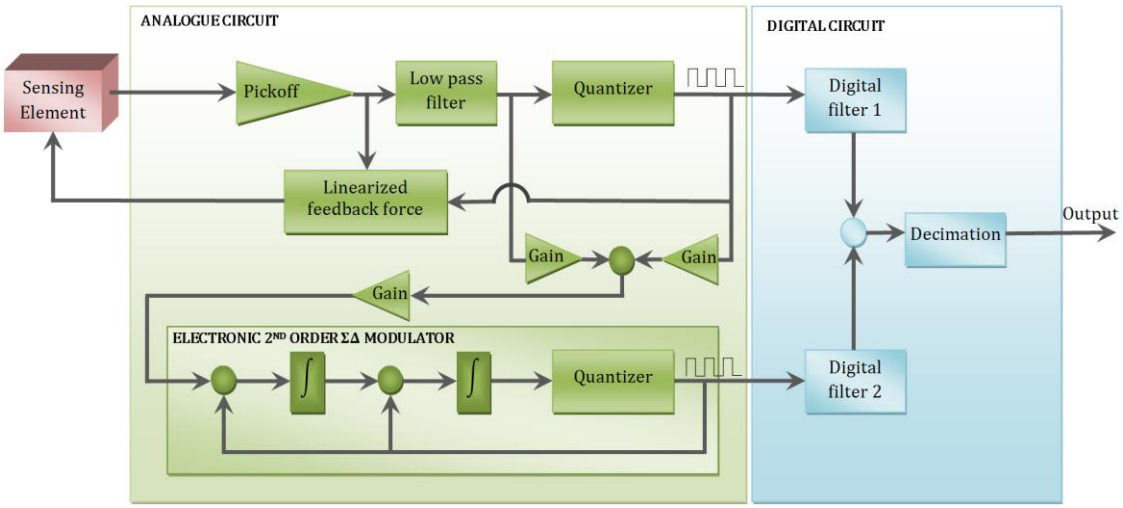


Figure 4-1: Block diagram of an electromechanical MASH22 ΣΔM for an inertial MEMS capacitive sensor.

The digital circuit (in Figure 4-1) applies the digital filters (D_1, D_2), which are responsible for cancelling the quantization noise from all stages but the last, as well as the decimation filter, which down-samples and low-pass filters the output signal from the high frequency noise.

4.2.1 Second Order Electromechanical ΣΔ Modulator

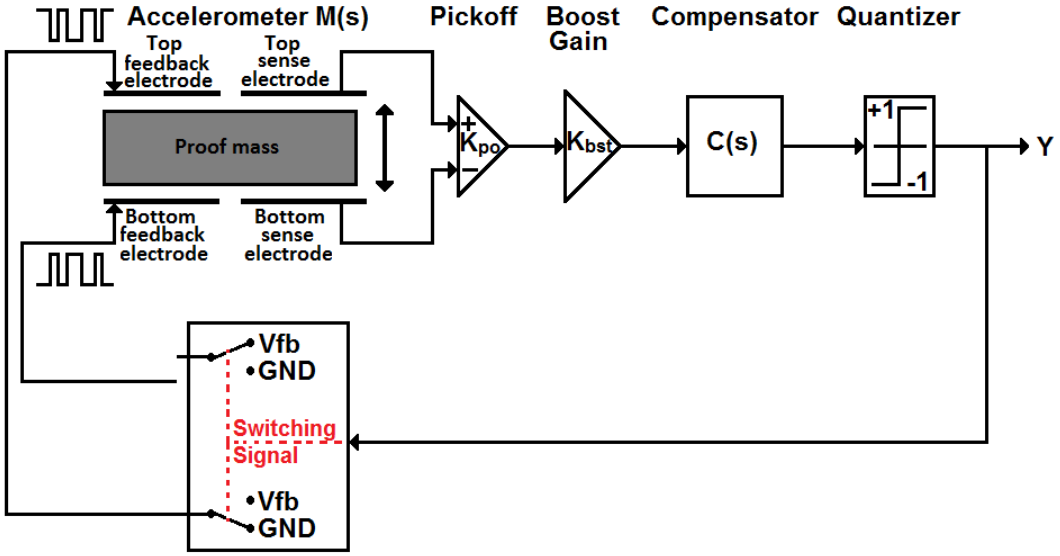


Figure 4-2: Second order EM-ΣΔM block diagram.

The second order EM-ΣΔM (SD2) is the first loop in the EM-MASH. A block diagram of the SD2 for a capacitive accelerometer is shown in Figure 4-2. The accelerometer (M) is embedded inside a second order ΣΔM control loop, which creates a mechanical loop filter. Due to the inertial force (i.e., acceleration), the proof mass experiences a displacement with reference to the sensor frame. This displacement induces a capacitance change, which is detected by the pick-off circuit and is converted to a proportional voltage signal. Although the sensing element is a second order mechanical filter, the input to the second integrator is not accessible; thus, the stability of the loop may require additional circuitry. Therefore, a lead compensator (C) is usually employed to stabilise the control loop. The quantizer converts the analogue voltage to a digital signal in the form of a pulse density modulated bitstream. The output is either +1 or -1, and the conversion is performed with a sampling frequency higher than the inertial sensor bandwidth; thus, an over-sampling is realised. Based on the sign of the bitstream, a negative feedback force is achieved by means of electrostatic force with the application of the required voltage pulse on one actuator (e.g., the top electrode), while the other actuator (e.g., the bottom

electrode) is grounded, and vice versa. The electrostatic feedback force keeps the mass, on average, at its nominal position. This ensures that the pull-in situation, which is encountered in the analogue closed loop, is avoided.

As discussed in section 2.5.1, the model can be linearized using the usual assumption, whereby the quantizer is modelled as a simple gain (K_{q1}) with additive white noise (Q_{n1}). For small mass deflection, the pick-off and feedback circuits are modelled as simple gain constants (K_{po}, K_{bst} and K_{fb}), respectively [19], as shown in Figure 4-4. The signal transfer function (STF_1) and the quantization noise transfer function (NTF_1) can be derived as follows:

$$STF_1 = \frac{MK_{po}K_{bst}CK_{q1}}{1 + MK_{po}K_{bst}CK_{q1}K_{fb}} \quad 4.1$$

$$NTF_1 = \frac{1}{1 + MK_{po}K_{bst}CK_{q1}K_{fb}} \quad 4.2$$

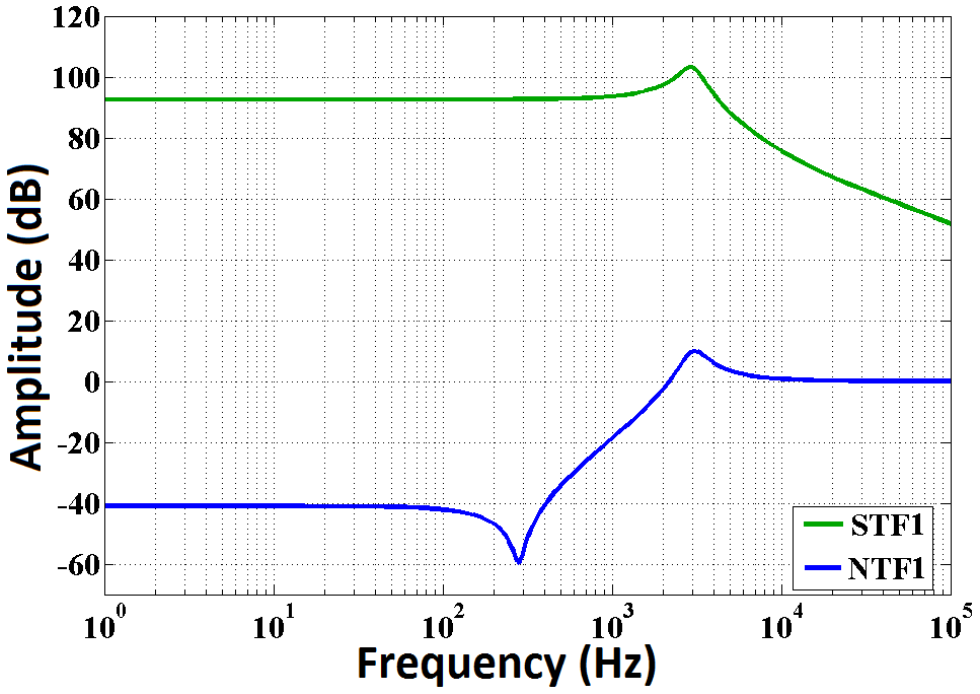


Figure 4-3: Frequency responses of the quantization noise and the input signal transfer functions in the SD2.

The frequency response of the STF_1 and NTF_1 are shown in Figure 4-3. The figure demonstrates how the quantization noise is shaped by the electromechanical closed loop filter. The digital output advantage of the SD2 comes at the cost of quantization noise. Therefore, the main objective of the modulator is to provide a measurement in a controllable and stable fashion with minimum quantization noise. Thus, a higher order EM-ΣΔM is needed, which can be achieved by the MASH structure.

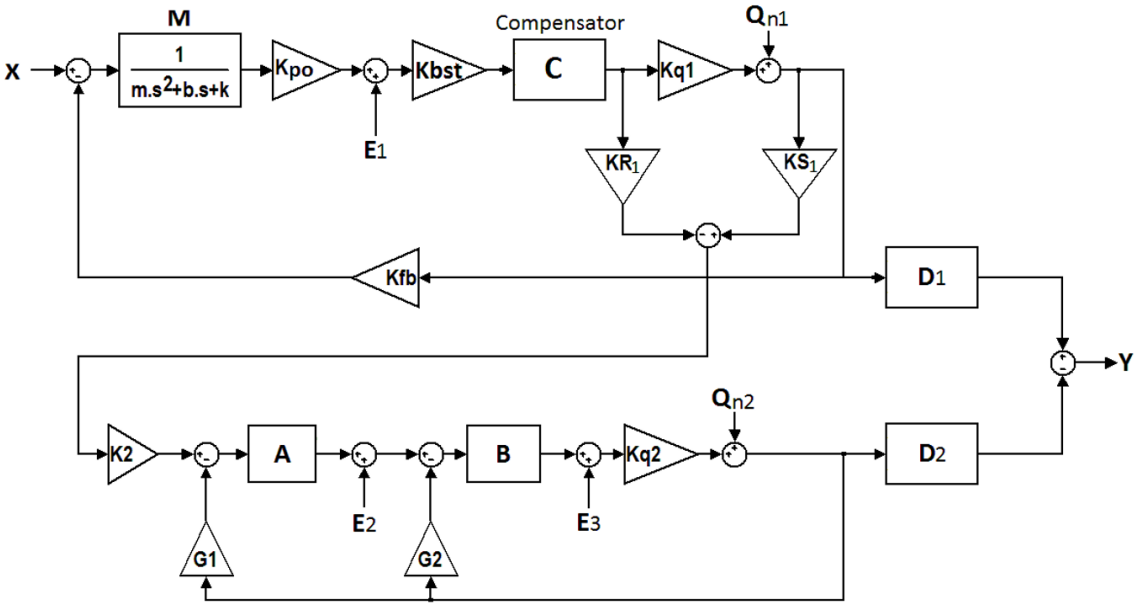


Figure 4-4: Linear model of an EM-MASH22 for an inertial MEMS capacitive sensor.

4.2.2 Electromechanical MASH22-ΣΔM

The fourth order MASH22 is constructed by cascading a second order, purely electronic ΣΔM to the first loop, as shown in the linear model in Figure 4-4. The quantization noise from the first stage is scaled by the three gain constants (KS_1, KR_1 and K2) and fed to the second stage for conversion to a digital bitstream. It is then cancelled by the digital filters D_1 and D_2. The signal transfer function of the MASH22 (STF) derived from the linear model is given by:

$$STF = \frac{MK_{po}K_{bst}CK_{q1}D_1}{1 + MK_{po}K_{bst}CK_{q1}K_{fb}} - \left[\left(\frac{A\ B\ K_{q2}\ D_2}{1 + G_1ABK_{q2} + G_2BK_{q2}}\right)\left(\frac{\left(KS_1K2K_{q1} - KR_1K2\right)K_{po}MK_{bst}C}{1 + MK_{po}K_{bst}CK_{q1}K_{fb}}\right)\right] \quad 4.3$$

The electronic noise transfer functions (E_1NTF, E_2NTF and E_3NTF) introduced by the pick-off circuit in the first stage and the integrators in the second stage are given by:

$$E_1NTF = \frac{K_{q1}D_1}{1 + MK_{po}K_{bst}CK_{q1}K_{fb}} - \left[\left(\frac{A\ B\ K_{q2}\ D_2}{1 + G_1ABK_{q2} + G_2BK_{q2}}\right)\left(\frac{\left(KS_1K2K_{q1} - KR_1K2\right)}{1 + MK_{po}K_{bst}CK_{q1}K_{fb}}\right)\right] \quad 4.4$$

$$E_2NTF = -\frac{K_{q2}B\ D_2}{1 + G_1ABK_{q2} + G_2BK_{q2}} \quad 4.5$$

$$E_3NTF = -\frac{K_{q2}\ D_2}{1 + G_1ABK_{q2} + G_2BK_{q2}} \quad 4.6$$

The quantization noise transfer functions from the first stage (Q_1NTF) and the second stage (Q_2NTF) are given by:

$$Q_1NTF = \frac{D_1}{1 + MK_{po}K_{bst}CK_{q1}K_{fb}} - \left[\left(\frac{A\ B\ K_{q2}\ D_2}{1 + G_1ABK_{q2} + G_2BK_{q2}}\right)\left(\frac{\left(KS_1K2 - \frac{KR_1K2}{K_{q1}}\right)}{1 + MK_{po}K_{bst}CK_{q1}K_{fb}} + \frac{KR_1K2}{K_{q1}}\right)\right] \quad 4.7$$

$$Q_2NTF = -\frac{D_2}{1 + G_1ABK_{q2} + G_2BK_{q2}} \quad 4.8$$

Equation 1.7 can be rewritten as follows:

$$Q_1NTF = NTF_1D_1 - \left\{STF_2D_2\left[NTF_1\left(KS_1K2 - \frac{KR_1K2}{K_{q1}}\right) + \frac{KR_1K2}{K_{q1}}\right]\right\} \quad 4.9$$

Where STF_2 is the signal transfer function of the electronic ΣΔM in the second stage, which is given by:

$$STF_2 = \frac{A\ B\ K_{q2}}{1 + G_1 ABK_{q2} + G_2 BK_{q2}} \qquad 4.10$$

To cancel the quantization noise (Q_{n1}) introduced in the first stage from the final output, the Q_1NTF in equation 4.9 must equal zero; therefore, the digital filter D_2 is given by:

$$D_2 = D_1 \frac{\mathrm{NTF_1}}{\mathrm{STF_2}\left[\mathrm{NTF_1}\left(\mathrm{KS_1K2} - \frac{\mathrm{KR_1K2}}{\mathrm{K_{q1}}}\right) + \frac{\mathrm{KR_1K2}}{\mathrm{K_{q1}}}\right]} \qquad 4.11$$

The digital filter D_1 is usually designed to introduce a delay in the path of the first stage to compensate for the time mismatch between the loops. In an ideal case, the quantization noise Q_{n1} introduced by the first stage will be cancelled; only the quantization noise Q_{n2} from the second stage will appear at the modulator output. The value of Q_{n2} is shaped by the second stage loop filters and the digital filter D_2, such that it has a lower noise floor than the electronic noise sources. Moreover, equation 4.11 shows that, if we know the NTF_1 of the first stage and STF_2 of the second stage, we can easily determine the digital filter D_2. However, in practice, the precise modelling of these two transfer functions is not possible because NTF_1 is function of the sensing element and STF_2 is function of analogue electronics, both of which are subject to manufacturing tolerances and imperfections. This leads to a mismatch between the digital filter D_2 and the analogue components. This mismatch causes a leakage of the quantization noise Q_{n1} in the final output and degrades the modulator performance.

Equation 4.11 can be written in a generic form to design the digital filters for higher stages (i.e., second, third and fourth stages) as follows:

$$D_n = D_{n-1} \frac{NTF_{n-1}}{STF_n\left[NTF_{n-1}\left(KS_{n-1}Kn - \frac{KR_{n-1}Kn}{K_{qn-1}}\right) + \frac{KR_{n-1}Kn}{K_{qn-1}}\right]} \qquad 4.12$$

where $n > 2$ is an integer representing the stage number. Higher order MASH structures (e.g., MASH221 and MASH222) that use this equation to design their digital filters are presented in section 4.7.2.

Using the system parameters listed in Table 4-1, the frequency responses of the transfer functions STF, E_1NTF, E_2NTF, E_3NTF and Q_2NTF for the EM-MASH22 are presented in Figure 4-5. It shows a flat and amplified response of the STF throughout the bandwidth of interest (1 kHz) for the acceleration measurement. The quantization noise Q_{n2} is shaped such that it has a lower noise floor than the electronic noises. It is clear that the electronic noise E_1 introduced by the pick-off circuit is the dominant noise that affects the performance of the modulator.

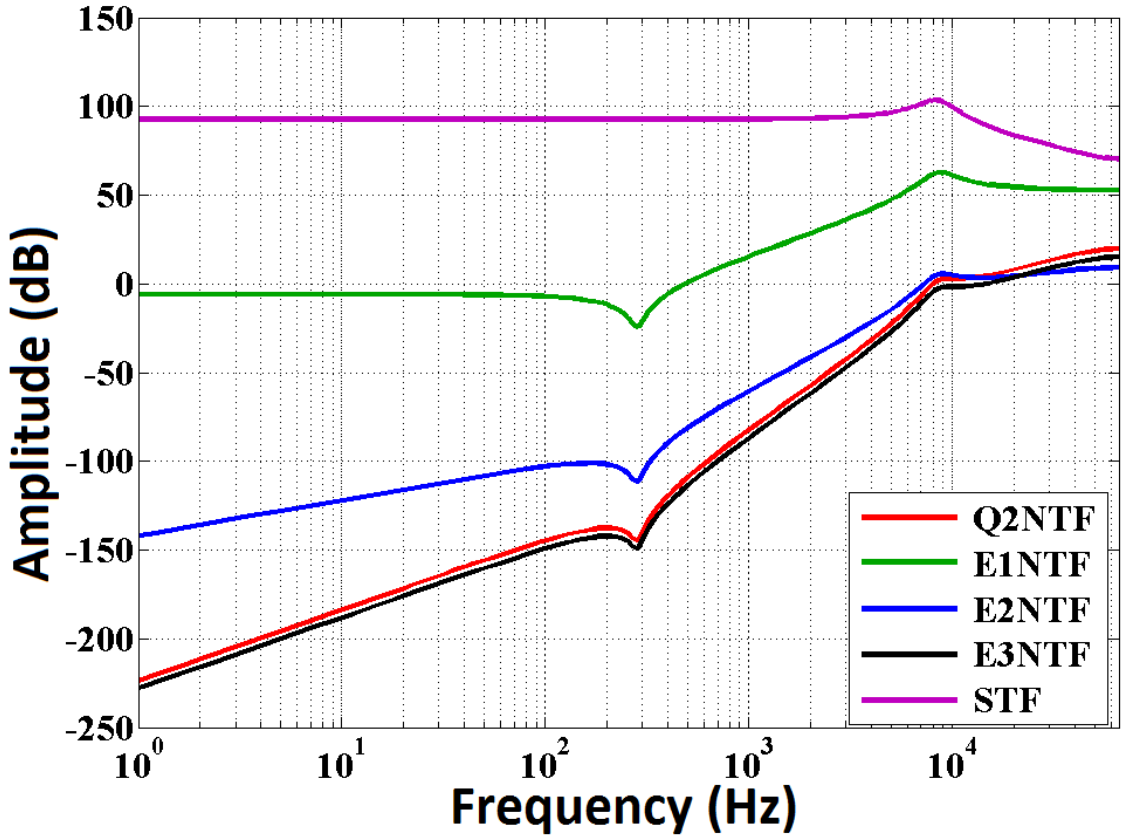


Figure 4-5: Frequency responses for different noise sources and the input signal in an EM-MASH22 model.

4.3 Electromechanical MASH-ΣΔM Stability Analysis

Since the MASH concept is based on cascading inherently stable first and second order ΣΔMs, the stability of the MASH system is guaranteed [55] [56].The same concept is valid for the EM-MASH. However, the stability of the first stage, the second order EM-ΣΔM, depends on the dynamic of the sensing element. For an underdamped sensor, a phase of 180° is caused by two complex poles at the resonance frequency; therefore, the stability of the second order EM-ΣΔM in the first stage of the MASH22 is assured by the

design of the lead compensator to provide a sufficient phase margin. For an over-damped sensor, the two poles are separated farther apart, such that one is within the bandwidth of interest and the other is much higher [25] [64]. Thus, the 180° phase is far beyond the sensor's bandwidth. This eliminates the necessity of a compensator.

As an example, an underdamped sensor is used to investigate the stability of the SD2. The open loop filter transfer function of the SD2 can be given by:

$$L = \frac{1}{NTF_1} - 1 = MK_{po}K_{bst}CK_{q1}K_{fb} \qquad 4.13$$

The root locus of the open loop filter is shown in Figure 4-6. The pair of complex poles at the resonance frequency is contributed by the sensing element, while the single pole at 300 kHz is contributed by the lead compensator. All poles are contained within the left-hand side region for all values of the quantizer gain (K_{q1}). This indicates unconditional stability of the modulator.

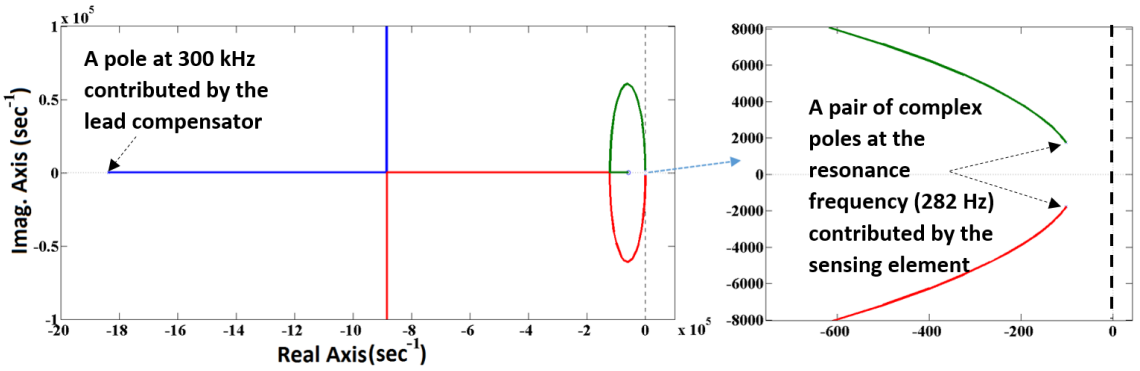


Figure 4-6: Root locus plot of the open loop filter of the SD2.

4.4 Electrostatic Feedback Force and the Maximum Acceleration Input Level

The method of closing the loop in EM-ΣΔM comprises the application of feedback pulses on the top and bottom electrodes based on the sign of the bitstream. This produces a negative feedback electrostatic force, which keeps the proof mass at its nominal position. If ε_o is the air permittivity, *Area* is the overlap area between the feedback electrodes and V_{fb} is the applied feedback voltage, then, the feedback electrostatic force is given by [19]:

$$K_{fb} = \frac{\varepsilon_o \, Area}{2}\left(\frac{V_{fb}}{d_o+x}\right)^2 \qquad 4.14$$

During the closed loop operation, the proof mass displacement is considered very small when compared with the nominal gap (i.e., x << d0), and the applied electrostatic force K_{fb} can be assumed linear [35] and is given by:

$$K_{fb} = \frac{\varepsilon_o \, Area}{2}\left(\frac{V_{fb}}{d_o}\right)^2 \qquad 4.15$$

The above linear assumption disregards the residual motion of the proof mass, which has a noticeable nonlinear effect in high-performance EM-ΣΔM and reduces the SNR. The effective electrostatic force increases as the proof mass gets closer to the energized electrode. This phenomenon produces a third harmonic distortion signal at three times the input acceleration frequency, as shown in the simulation result in Figure 4-7(a). However, a linearization technique proposed in [80] can be used to linearize the feedback force as follows:

$$K_{fb} = \frac{\varepsilon_o * Area}{2}\left(\frac{\left(1 \pm \frac{x}{d_o}\right)V_{fb}}{d_o \pm x}\right)^2 \qquad 4.16$$

This means that the amplitude of the feedback voltage V_{fb} must be reduced as the proof mass gets closer to the energized feedback electrode. The result of the feedback linearization is shown in Figure 4-7(b), where the third harmonic peak is highly suppressed.

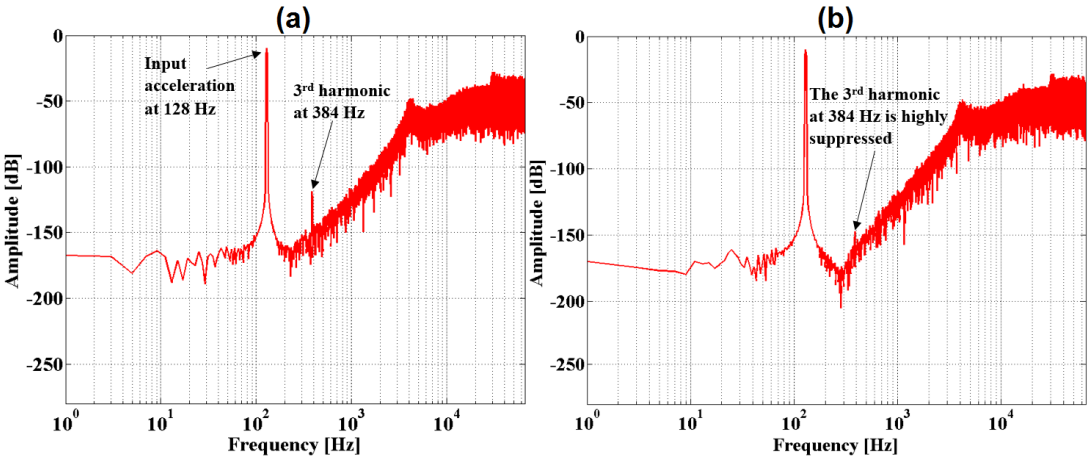


Figure 4-7: Simulation output spectrum of MASH22 (a) with conventional feedback force and (b) with linearized feedback force. FFT settings: sample rate = 131 kHz, number of samples = 128k, with Hanning window.

The overload acceleration input (OLA) is an important criterion in the EM-ΣΔM; the closed loop sensor is stable if operated within the OLA. If the modulator's order increases, the maximum acceleration input, wherein the system remains stable, correspondingly decreases [90]. During the closed loop operation, where the proof mass deflection is very small (x << d0), the dominant reacting force applied to the proof mass is the electrostatic force represented by (K_{fb}) (i.e., the spring force is negligible); therefore, the full-scale acceleration input can be approximated by K_{fb}/m (m/s^2). In EM-ΣΔM, the value of the OLA is less than the full-scale input, and they are related by the overload factor denoted by OLF, which is a number between zero and one. The overload input acceleration in 'g' unit is given by:

$$OLA = \frac{OLF \times K_{fb}}{m \times 9.81} \qquad 4.17$$

The OLA in equation 4.17 is proportional to the inverse of the proof mass m and to the feedback loop gain K_{fb}, which is a function of: i) the feedback voltage applied to the proof mass and ii) the parallel plates capacitive actuator parameters (e.g., the nominal gap [d_0] and the electrode overlap area [*Area*]).

4.5 SNR Estimation

For a given noise transfer function (Q_2NTF) and input signal power, the SNR for the modulator can be calculated. As discussed in section 2.5.1, if the quantization noise signal

is uniformly distributed over the interval $\pm\frac{\Delta}{2}$ (white noise), then its root mean square (RMS) can be given by [55]:

$$e_{RMS} = \frac{\Delta}{\sqrt{12}} \tag{4.18}$$

where Δ is the level spacing of the quantizer and for 1-bit quantizer $\Delta = 2$. The power spectral density (PSD) for the quantization noise signal is given by [55]:

$$E^2(f) = e_{RMS}^2 \frac{2}{f_s} \tag{4.19}$$

The PSD at the output of the modulator can be calculated by [55]:

$$N_{Q,out}^2 = E^2(f)|Q_2 NTF\,|^2 \tag{4.20}$$

The noise power n_0^2 can now be calculated by integrating $N_{Q,out}^2$ over the bandwidth of interest B as follows [55]:

$$n_0^2 = \int_0^B \left|N_{Q,out}^2\right| df \tag{4.21}$$

The RMS for sinusoidal signal is given by [55]:

$$RMS\ of\ input\ signal = Amp/\sqrt{2} \tag{4.22}$$

where Amp is the amplitude of the input acceleration signal. The SNR of the modulator can be given by [55]:

$$SNR = 10\log\left(\frac{RMS\ of\ input\ signal}{n_0^2}\right) \tag{4.23}$$

The above equations can be numerically evaluated using MATLAB (please refer to appendix A.1).

4.6 Design Procedure for Electromechanical MASH

Electromechanical MASH-ΣΔMs can be designed using a systematic approach to ensure stability and maximum performance. The model of the EM-MASH22, as shown in Figure 4-8, will be used for demonstration, but these design rules can be used for higher

stages with greater than two (i.e., MASH221 and MASH222). The design procedure is as follows:

1- Design each stage individually to ensure stability and maximum SNR.
    a. Design the first stage SD2 as follows:
        i. The acceleration input signal must be less than the full scale, which is approximated by $\frac{K_{fb}}{m}$, to avoid overloading the modulator.
        ii. Check whether a lead compensator is needed to stabilise the loop, as discussed in section 4.3.
        iii. Under the maximum acceleration according to (i), increase the forward path gain constants K_{po} and K_{bst}, such that the input level of the quantizer is smaller than its full-scale input (e.g., one volt). This step ensures that the quantizer gain (K_{q1}) has a small value. It also ensures high attenuation of the quantization noise, according to equation 4.2
        iv. The expected SNR value of this stage is greater than 55 dB.
    b. Design the second stage and beyond using purely electronic ΣΔMs.
        i. Use first and second order purely electronic ΣΔMs, as these are inherently stable.
        ii. The estimated SNR value of the first order electronic ΣΔM is 52 dB and the second order is 72 dB with an OSR of 64.
2- Interface the second stage to the first stage, such that the quantization noise signal level is less than the overload level of the second stage. This can be done by adjusting the scaling gains KR_1, KS_1, and K2. The estimated OLA value of the first-order electronic ΣΔM is 0.95 and the second order is 0.9 [55].
3- Design the digital filters as follows:
    a. Choose the digital filter D_1 as a delay that equals the order of the following stage (i.e., $D_1 = Z^{-1}$ for MASH21 $D_1 = Z^{-2}$ for MASH22, $D_1 = Z^{-4}$) for MASH222, and so on.
    b. Use the generic equation 4.12 to design the digital filter D_2.

4.7 MATLAB and Simulink Modelling

In this section, the EM-MASH22 is simulated using MATLAB and Simulink to study the performance in terms of noise shaping and SNR. The effect of the sensing element parameters tolerances is also investigated using Monte Carlo simulation with varied input power.

Different EM-MASH structures will be discussed at the end of this section to prove that the performance is enhanced as the order of the modulator is increased, whether by increasing the number of stages or the order of each stage.

4.7.1 Electromechanical MASH22

The Simulink model of the EM-MASH22 is presented in Figure 4-8. The system was constructed according to the design procedure described in section 4.5, with the parameter values summarized in Table 4-1 using typical accelerometer parameters [19].

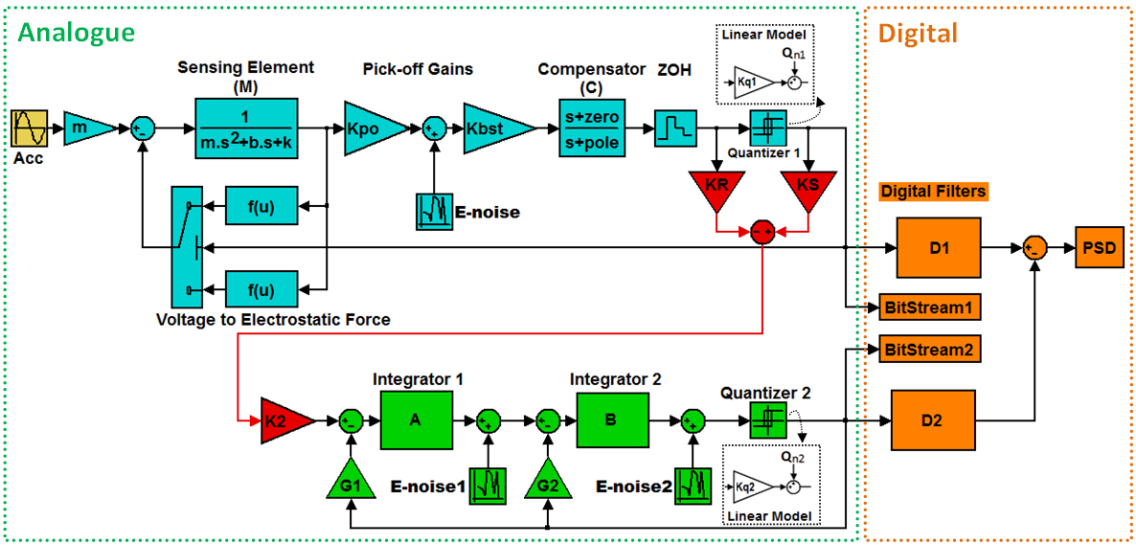


Figure 4-8: Electromechanical MASH22 Simulink model for an inertial MEMS capacitive sensor.

The first loop consists of a capacitive MEMS accelerometer embedded in a digitally controlled force feedback loop forming SD2. The compensator is required to maintain loop stability. The second loop is a purely electronic, second order sigma-delta modulator, where the quantization noise from the first loop is scaled by the interface gains (KR, KS,

and K2), then digitized by the second loop, and finally cancelled by the digital filters D_1 and D_2.

The digital filter D_1 is a simple double delay, while D_2 is calculated using equation 4.12 and they are given by:

$$D_1 = \frac{1}{Z^2}$$

$$D_2 = \frac{2.242\,Z^5 - 5.633\,Z^4 + 4.785\,Z^3 - 1.636\,Z^2 + 0.243\,Z - 0.0001957}{Z^5 - 1.757\,Z^4 + 0.9303\,Z^3 - 0.02838\,Z^2}$$

The electronic noises are included in the model to study their effect on the performance. In fact, the electronic noise sources are beneficial to the MASH modulator, as they act as dithering signals that randomize the quantization noises and make them independent of the input signals for both stages. The electrostatic force linearization method discussed in section 4.4 is also used to achieve maximum SNR.

Parameter	**Value**	**Parameter**	**Value**
First stage: second order EM-ΣΔM			
Acceleration input [m/s^2]	12.21	**Input frequency [Hz]**	256
OSR	64	**Pick-off gain 'K_{po}' [V/m]**	5×10^{6}
Bandwidth [Hz]	1024	**Boost gain 'K_{bst}**	200
Proof mass 'm' [kg]	1.745×10^{-6}	**Compensator zero [Hz]**	1388
Damping coeff. 'b' [N.s/m]	0.35×10^{-3}	**Compensator pole [Hz]**	148.5×10^{3}
Spring constant 'k' [N/m]	5.492	**Feedback voltage 'V_{fb}' [V]**	9
Second stage: second order electronic ΣΔ modulator			
G1	1	**G2**	2.5
Interface scaling gains			
KR	1	**KS**	0.5
K2	1.7		

Table 4-1: Design parameters for the EM-MASH22 used in the Simulink model.

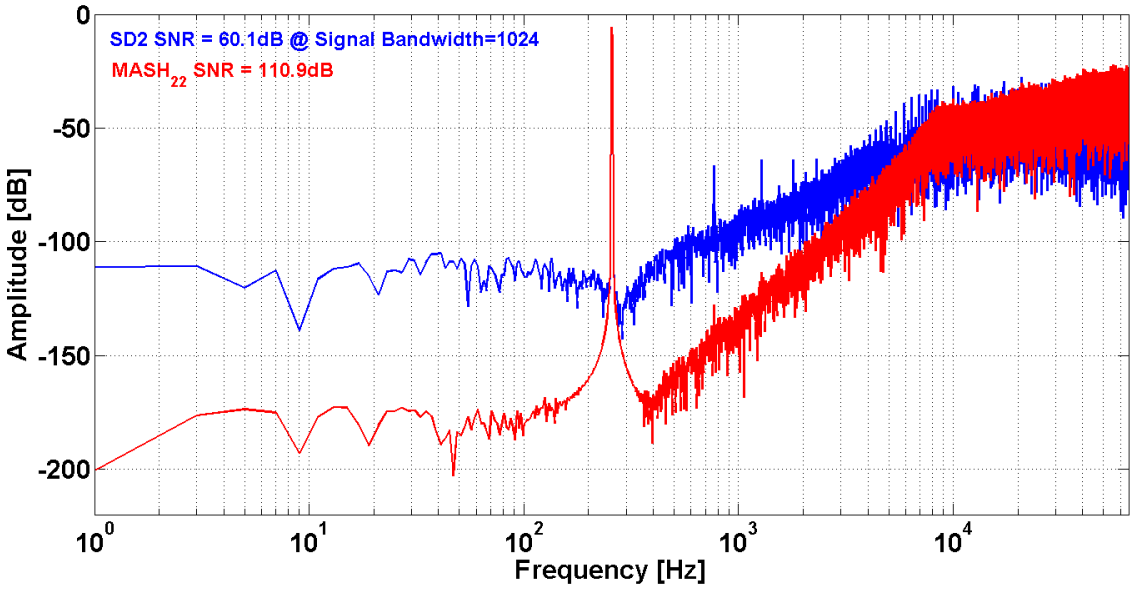


Figure 4-9: Electromechanical MASH22 noise-shaping characteristic. FFT settings: sample rate = 131 kHz, number of samples = 128k, with Hanning window.

The noise shaping characteristic of the MASH22 and the SD2 are illustrated in Figure 4-9. It can be seen that MASH22 enhanced the performance of the SD2 by about 50 dB and the noise floor of the MASH22 is considerably reduced in the bandwidth of interest. Using the noise transfer function in equation 4.8 and the method discussed in section 4.5, the estimated SNR for the MASH22 equals 111.4 dB, which closely agrees with the simulation result (110.9 dB).

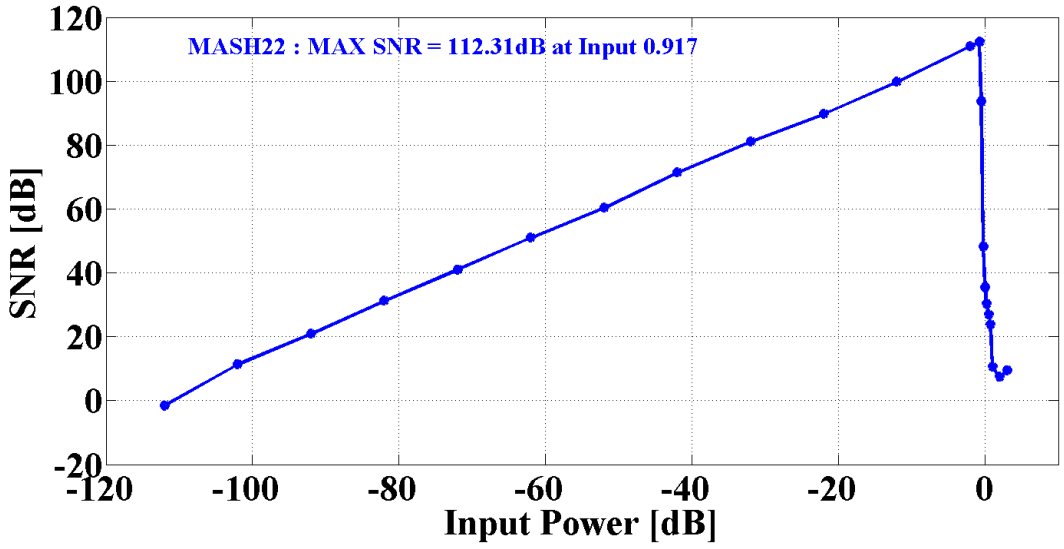


Figure 4-10: Electromechanical MASH22 input power versus SNR.

The performance of the EM-MASH22 is examined for various input signal powers. As shown in Figure 4-10, it shows a high overload input level of 0.917 of the full scale input; this is due to the characteristic of the first stage SD2. Furthermore, it shows a wide dynamic range of 110 dB.

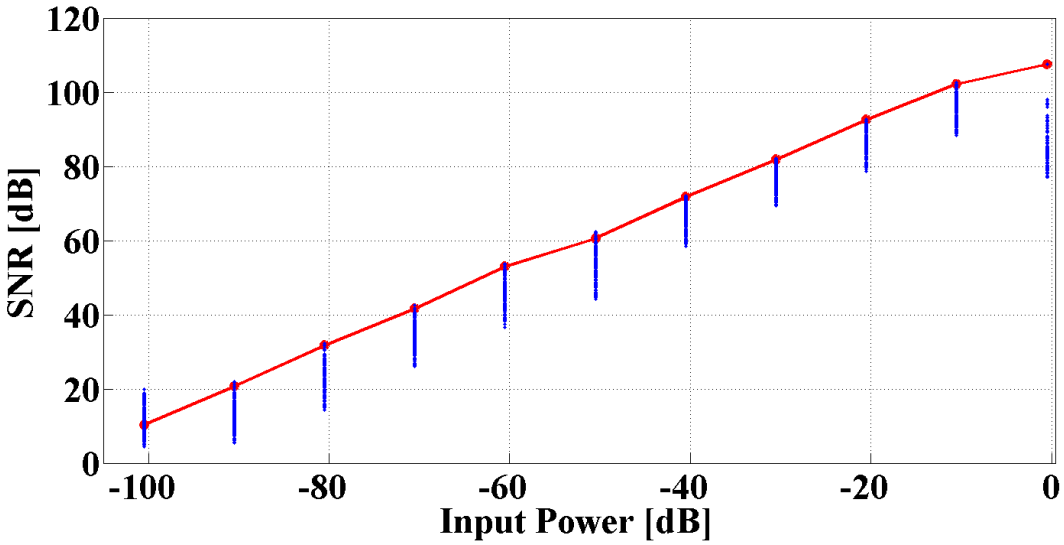


Figure 4-11: Electromechanical MASH22 ΣΔM sensitivity to inertial sensor parameter variation using Monte Carlo simulation with input power versus SNR. The SNR resulting from the nominal values is in (Red), while the (Blue) bars represent the SNR due to the parameter variation.

As discussed before, the mismatch between the digital filters and the analogue components, including the sensing elements, results in a leakage of the quantization noise $Q_{n1}$from the first stage to the final output. Figure 4-11 shows the Monte Carlo analysis resulting from varying the sensing element parameters (*m, b, and k*) with 15% of the nominal values for different input power. It can be seen that, the SNR is degraded by a maximum of 10%. As well, at the maximum input power level, the SNR is degraded by a maximum of 30%. This is a drawback of the MASH structure.

4.7.2 Higher Order Electromechanical MASH

This section discusses various EM-MASH structures. By means of theoretical analysis and simulation, third-, fourth-, fifth- and sixth-order MASH are compared in terms of noise shaping and SNR. All of these structures were designed using the procedure that is addressed in section 4.6.

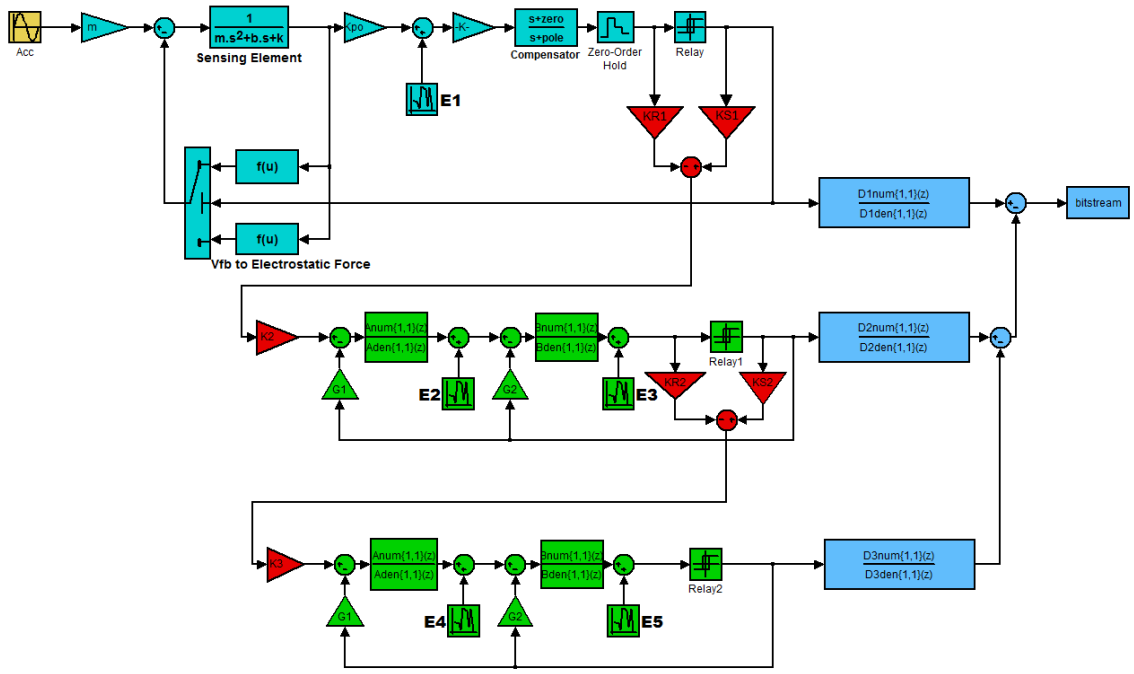


Figure 4-12: Electromechanical MASH222 Simulink model for an inertial MEMS capacitive sensor.

For example, by adding a third stage of the second order ΣΔ modulator to MASH22, a sixth-order ΣΔ modulator using MASH222 is constructed, as shown in Figure 4-12. As discussed previously, the digital filter D_1 is designed to compensate for the time mismatch between the first stage and the higher stages, while the digital filters D_2 and D_3 are calculated using the generic formula in equation 4.12, as shown below:

$$D_1 = \frac{1}{Z^4} \tag{4.24}$$

$$D_2 = D_1 \frac{\mathrm{NTF}_1}{\mathrm{STF}_2 \left[\mathrm{NTF}_1 \left(\mathrm{KS}_1 \mathrm{K2} - \frac{\mathrm{KR}_1 \mathrm{K2}}{\mathrm{K_{q1}}} \right) + \frac{\mathrm{KR}_1 \mathrm{K2}}{\mathrm{K_{q1}}} \right]} \tag{4.25}$$

$$D_2 = \frac{4.5\,Z^5 - 11.3\,Z^4 + 9.602\,Z^3 - 3.284\,Z^2 + 0.4876\,Z - 0.0003928}{Z^7 - 1.685\,Z^6 + 0.9095\,Z^5 - 0.03659\,Z^4}$$

$$D_3 = D_2 \frac{\mathrm{NTF}_2}{\mathrm{STF}_3\left[\mathrm{NTF}_2\left(\mathrm{KS}_2\mathrm{K3} - \frac{\mathrm{KR}_2\mathrm{K3}}{\mathrm{K_{q2}}}\right) + \frac{\mathrm{KR}_2\mathrm{K3}}{\mathrm{K_{q2}}}\right]} \qquad 4.26$$

$$D_3 = \frac{7.78\,Z^9 - 41.95\,Z^8 + 96.93\,Z^7 - 125.8\,Z^6 + 101.1\,Z^5 - 51.91\,Z^4}{Z^9 - 2.197Z^8 + 1.88Z^7} \ldots$$

$$\ldots \frac{+16.78\,Z^3 - 3.162\,Z^2 + 0.2778\,Z - 0.000223}{-0.6836Z^6 + 0.1164Z^5 - 0.00393Z^4} \qquad 4.27$$

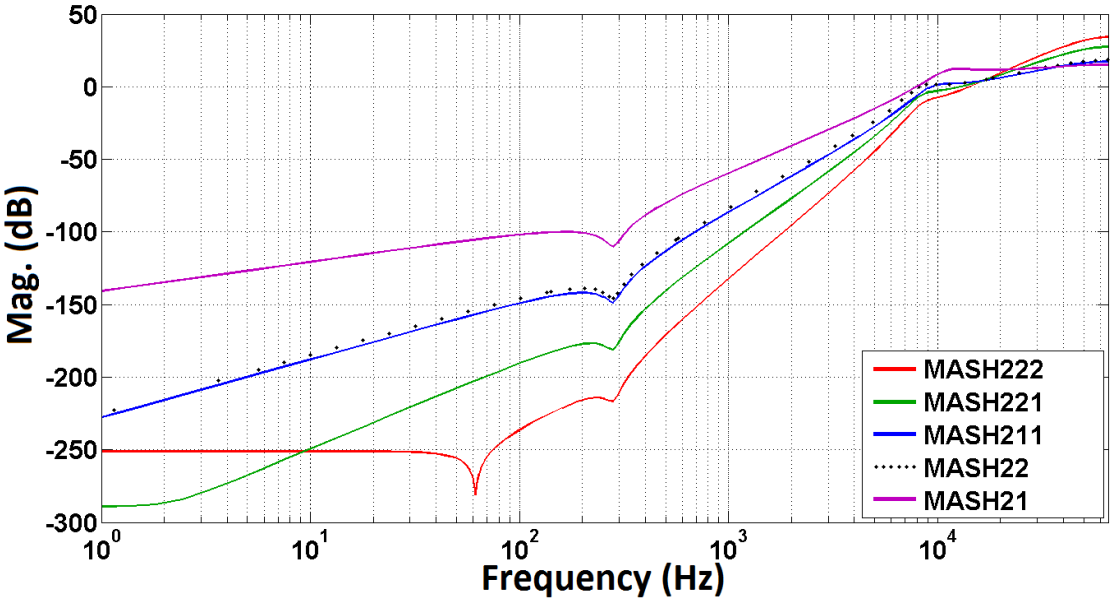


Figure 4-13: Theoretical noise-shaping characteristics of various EM-MASH.

A theoretical analysis was carried out for different MASH structures by employing a SD2 in the first stage. In addition to the previously described MASH22 and MASH222, the first loop was cascaded with i) a first-order electronic ΣΔM loop to form a MASH21, ii) two first-order electronic ΣΔM loops to form a MASH211 iii) and a second- and first-order electronic ΣΔM loop to form a MASH221. The theoretical noise-shaping characteristics for the MASH21, MASH22, MASH211, MASH221 and MASH222 are presented in Figure 4-13. It can be seen that as the order increases, the noise floor suppression increases. Furthermore, a fourth order noise shaping can be achieved by two different structures, MASH22 and MASH211.

The performances of the various MASH structures were simulated in MATLAB and Simulink, with and without the effect of the electronic noise sources. The SNR and the noise shaping were observed for both cases, as shown in Figure 4-14 and Figure 4-15. As expected, the electronic noise introduced by the pickoff circuit can be seen to have insignificant effect on the lower order MASH structures, while the higher order MASH structure performance degrade dramatically as they hit the noise floor of the electronic whit noise. In other word, the quantization noise is no longer the performance limiter and the electronic noise introduced by the pickoff circuit define the limit of the higher order EM-MASH structure.

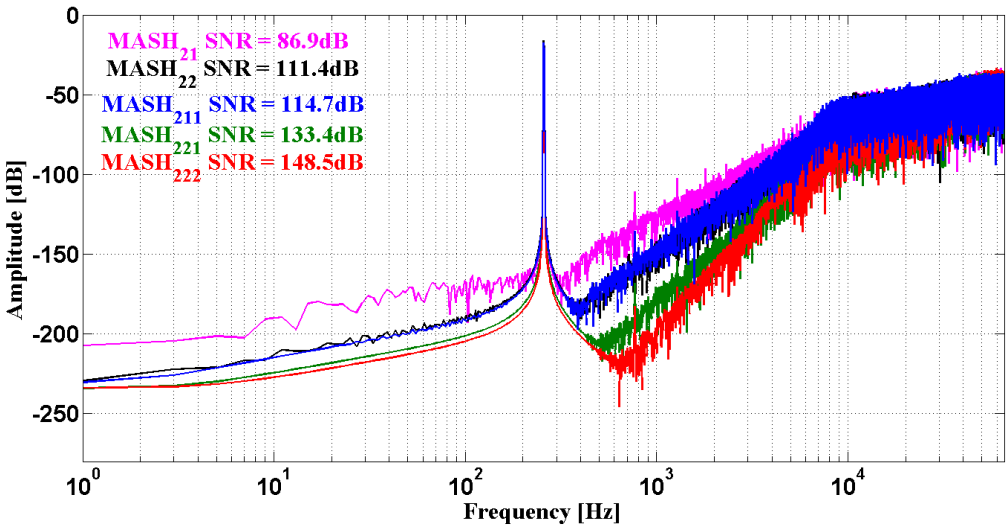


Figure 4-14: Noise-shaping characteristics of various EM-MASH without electronic noise. FFT settings: sample rate = 131 kHz, number of samples = 128k, with Hanning window.

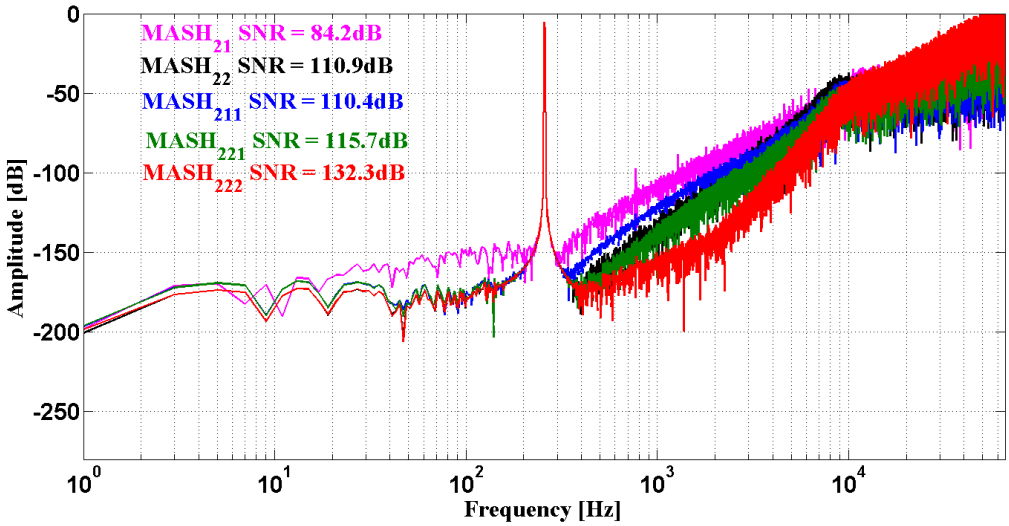


Figure 4-15: Noise-shaping characteristics of various EM-MASH with electronic noise. FFT settings: sample rate = 131 kHz, number of samples = 128k, with Hanning window.

The theoretical noise-shaping results shown in Figure 4-13 closely agreed with the simulation results in Figure 4-14. Moreover, the simulated SNR results were also in agreement with the estimated values, as shown in Table 4-2.

MASH structure	Theoretical estimated SNR (dB)	Simulated SNR (dB) without E-Noise	Simulated SNR (dB) with E-Noise
MASH21	86.5	86.9	84.2
MASH22	111.4	111.4	110.9
MASH211	114.5	114.7	110.4
MASH221	136.5	133.4	115.7
MASH222	155.7	156.0	132.2

Table 4-2: Theoretical and simulation comparison between different EM-MASH-ΣΔMs in term of SNR.

A further investigation was conducted, whereby the MASH structures were simulated with various input powers. Figure 4-16 shows the analysis of the relationship between the input signal power and the output signal-to-noise ratio (SNR). The OLA for all of these MASH structures is 0.9 of the full scale input, thus using equation 4.17 and assuming a feedback voltage of 9 V, the expected OLA is 1.6 g. It can be seen that the MASH structure maintains the same overload level as the order of the modulator increases. Furthermore, the SNR increases by approximately 20 dB as the order or the number of stages increases, and the dynamic range increases as the order of the modulator increases.

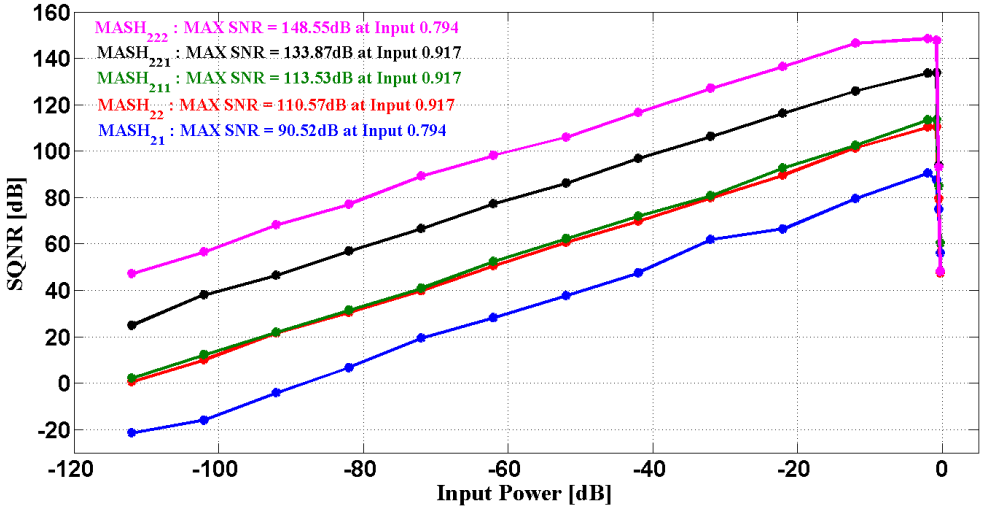


Figure 4-16: Comparison between input signal power and signal-to-quantization-noise ratio for various EM-MASH.

4.8 Summary

In this chapter, the operation concept and the design procedure of the EM-MASH was explained. The Simulink and MATLAB analysis of the fourth order electromechanical MASH architecture showed that the performance of the accelerometer increased by suppressing the quantization noise floor, such that it was no longer the dominant noise in the system. On the other hand, the analysis of MASH showed its sensitivity due to the system parameter variations.

The analysis was also carried out for higher-order EM-MASH architectures, and the study showed that the higher the order, the lower the noise floor. However, the quantization noise was no longer the performance limiter, and the electronic noise introduced by the pick-off circuit defined the limit of the higher-order EM-MASH structure. Therefore, the concept of the MASH is theoretically feasible and the performance of the capacitive accelerometer MEMS can be increased using MASH structure.

Chapter 5: System Level Comparative Study of Multi-Stage Noise Shaping and Single-Loop Sigma-Delta Modulators for MEMS Accelerometers

5.1 Introduction

Single-loop, higher-order EM-ΣΔM architectures have been applied successfully to capacitive MEMS accelerometers [21] [119]. In contrast, multi-stage (MASH) ΣΔM architectures have rarely been investigated for use as MEMS sensors [117], despite having the considerable advantage of inherent stability and simple construction. Both modulator architectures share a common feature whereby they improve the linearity, dynamic range and bandwidth of the MEMS sensor, while providing a digital output. In this chapter, two fourth order EM-ΣΔM are compared at the system level: 1) a single-loop (SD4) architecture and 2) a multi-stage (MASH22) architecture with a first loop comprising the second order transfer function of the micromachined accelerometer sensing element and the second loop consisting of a standard electronic second order ΣΔM. Both architectures were simulated and compared in terms of stability, signal-to-noise ratio (SNR), dynamic range and overload input level. The over-sampling ratio (OSR) for the comparison was set to 64 for both MASH22 and SD4. The comparison was performed for two different micromachined accelerometer sensing elements: one over-damped and the other underdamped, with the latter requiring an electronic phase compensator for loop stability [26].

The parameters of the micromachined sensors are presented in Table 5-1 with the corresponding performance comparisons. The first accelerometer was from QinetiQ and was fabricated using bulk micromachining, with resonant frequency of 1.28 kHz. The second sensor was fabricated by the Southampton Nanofabrication Centre with a resonant frequency of 282 Hz.

5.2 System Level Modelling

The Simulink models of the SD4 and MASH22 are illustrated in Figure 5-1 and Figure 5-2 respectively. The electromechanical sensing element (M) is a second order transfer

function modelling the mass-spring-damper system, and constitutes the mechanical filter part of the ΣΔM. Due to the inertial force (i.e., acceleration), the proof mass experiences a displacement with reference to the sensor's frame. This displacement induces a capacitance change, which is detected by the pick-off circuit (modelled by gain constants K_{po} and K_{bst}), converting it to a proportional voltage signal. A lead compensator (C) is usually used to stabilise the control loop with the underdamped sensor. The quantizer converts the analogue voltage to a digital signal in the form of a pulse density modulated bitstream. Based on the sign of the bitstream, a negative feedback force is achieved by electrostatic force, through the application of the required voltage pulse on one actuator (e.g., the top electrode), while the other actuator (e.g., the bottom electrode) is grounded, and vice versa. The electrostatic feedback force (K_{fb}) keeps the mass, on average, at its nominal position.

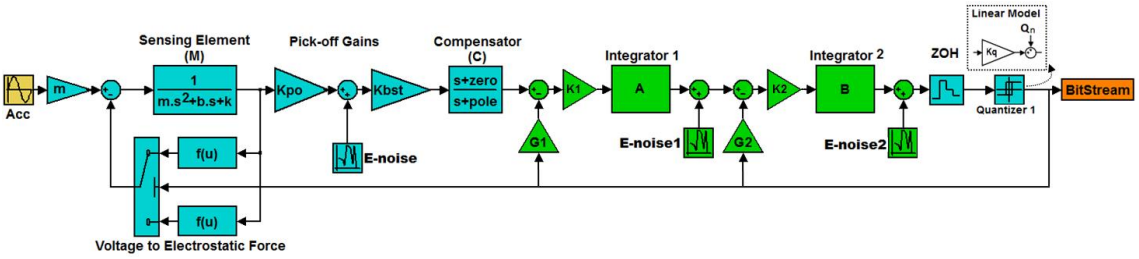


Figure 5-1: Electromechanical SD4 ΣΔM architecture. The micromachined accelerometer sensing element is cascaded with two electronic integrators to form a SD4.

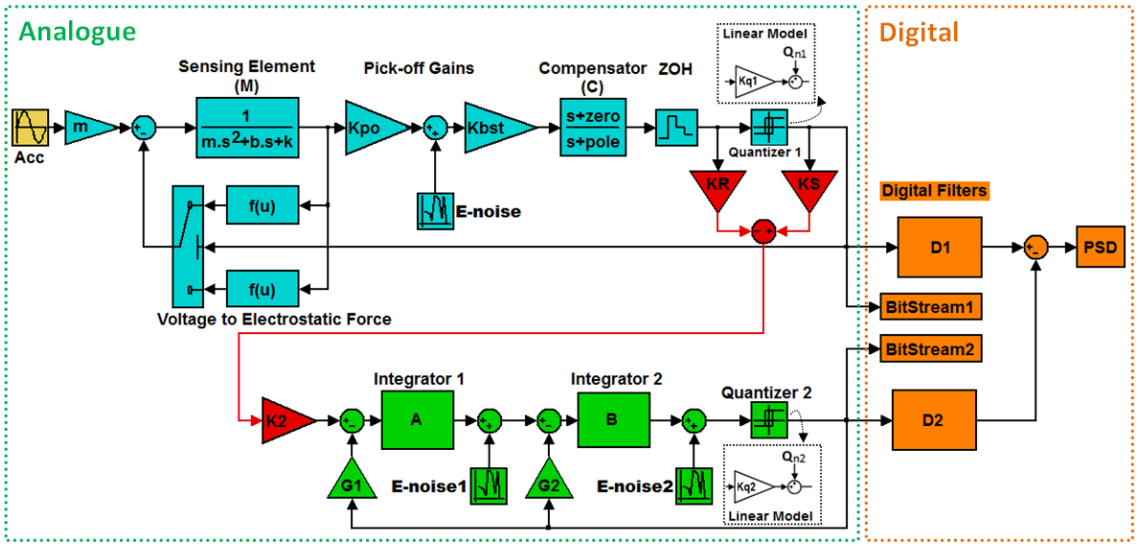


Figure 5-2: Electromechanical MASH22 architecture; the first loop comprises the micromachined accelerometer sensing element, whereas the second loop is purely electronic. The output bitstreams of the two loops are combined by digital filtering.

To construct a single loop SD4, two electronic integrators (A and B) with their associated gains (G1, K1, G2 and K2) are added in the forward path as shown in Figure 5-1 which provide additional loop filtering and noise shaping. The MASH22, on the other hand, is constructed by cascading a purely electronic second order ΣΔM as shown in Figure 5-2 where the quantization noise from the first stage is scaled by the three gain constants (KS, KR and K2), then digitized by the second stage and cancelled by the digital filters D_1 and D_2. Ideally, the only noise that appears in the overall modulator output is the quantization noise of the second stage along with the electronic noise, which will be shaped by an order equal to the sum of all stages orders, i.e. fourth order noise shaping.

To analyse both architectures mathematically, the usual assumption of modelling the quantizer as a simple gain with additive white noise is made. For small mass deflections, the pickoff and feedback circuits can also be modelled simply as gain constants (K_{po}, K_{bst} and K_{fb}), respectively. Using the above assumptions, the quantization noise transfer functions (QNTF and Q_2NTF) for SD4 and MASH22 are given by:

$$SD4\_QNTF = \frac{1}{1 + MK_{po}K_{bst}CK_1AK_2BK_qK_{kf} + G_1K_1AK_2BK_q + G_2K_2BK_q} \qquad 5.1$$

$$MASH\_Q_2NTF = \frac{D_2}{1 + G_1ABK_{q2} + G_2BK_{q2}} \qquad 5.2$$

The digital filter D_1 is a simple double delay, while D_2 is calculated using equation 4.12. By calculating the noise power of the above noise transfer functions (QNTF and Q_2NTF) and the RMS of the full scale input signal, the SNR can be estimated for both structures using the method discussed in section 4.5. Using the under-damped sensor parameters, the estimated SNR values are 99.5 dB and 111.5 dB for the SD4 and the MASH22, respectively.

5.3 Noise Shaping and SNR

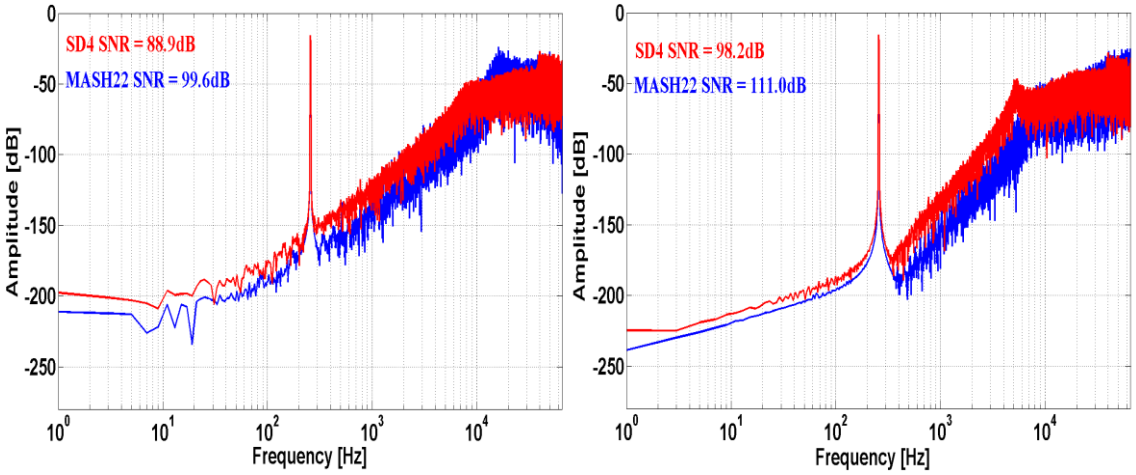


Figure 5-3: Noise-shaping and SNR comparison for the over-damped (left) and underdamped (right) sensors. FFT settings: sample rate = 131 kHz, number of samples = 128k, with Hanning window.

This simulation compared the noise shaping and the SNR using the two different sensors for the SD4 and MASH22. The Simulink results are presented in Figure 5-3, where the left figure shows the output spectra of the over-damped sensor for the two architectures, and the right shows the underdamped sensor. It is clear that both structures have similar noise shaping; however, the MASH22 shows a slightly lower noise floor, with SNRs of 99.6 dB and 111 dB for the over-damped sensor and the underdamped sensor respectively. On the other hand, the SD4 shows SNRs of 88.7 dB and 98.2 dB for the over-damped sensor and the underdamped sensor respectively. For both sensors, the SD4 cannot reach the SNR level of the MASH22 due to stability issues.

5.4 Stability

The stability analysis was carried out for both architectures with the under- and the over-damped sensors. In classical control theory, the closed loop system is stable if all poles' loci are within the left-hand side of the *s*-plane [88].

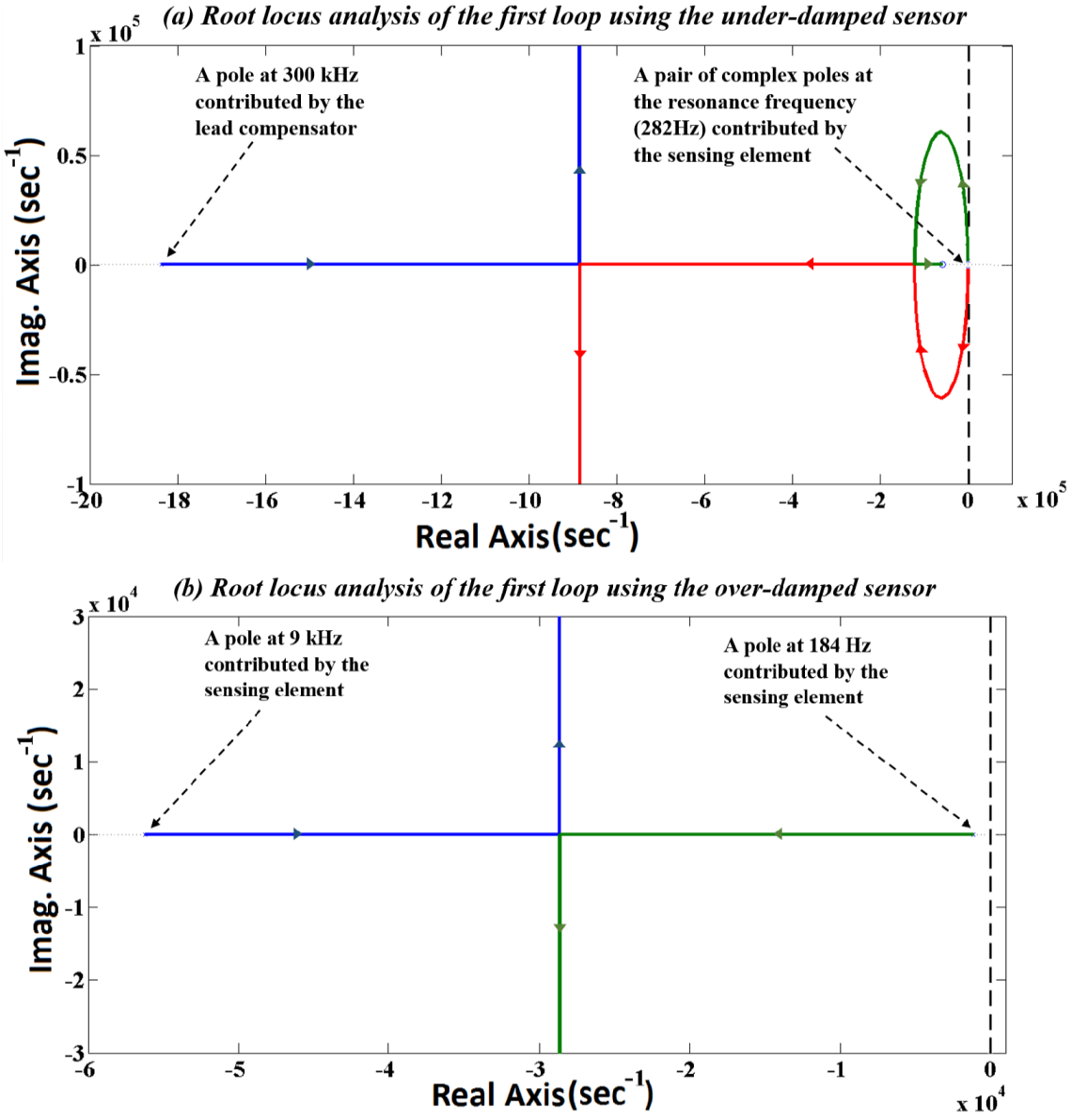


Figure 5-4: Root locus analysis of the open loop filter transfer function of the SD2 (a) underdamped and (b) over-damped sensors.

As discussed in section 4.3 the stability of the MASH system is guaranteed since it is based on cascading inherently stable first and second order ΣΔMs. The stability of the EM-MASH first stage depends on the dynamic of the sensing element. To perform the root locus analysis for the first stage, the open loop filter transfer function of the second order EM-ΣΔM (SD2) is given by:

$$L2 = MK_{po}K_{bst}CK_{q1}K_{fb} \qquad 5.3$$

The root locus stability analysis of the open loop control filter of the first stage is shown in Figure 5-4 (a) for the under-damped sensors and (b) for the over-damp one. The underdamped sensor provides pair of complex poles at the resonance frequency at 282 Hz. A lead compensator with a pole at 300 kHz is used to provide a sufficient phase margin and to insure the stability. The over-damped sensor provides a pole at 182 Hz which is within the sensor bandwidth and the other at 9 kHz. For both sensors, all poles are kept in the left-hand side for all values of K_{q1}. This indicates that the SD2 is unconditionally stable.

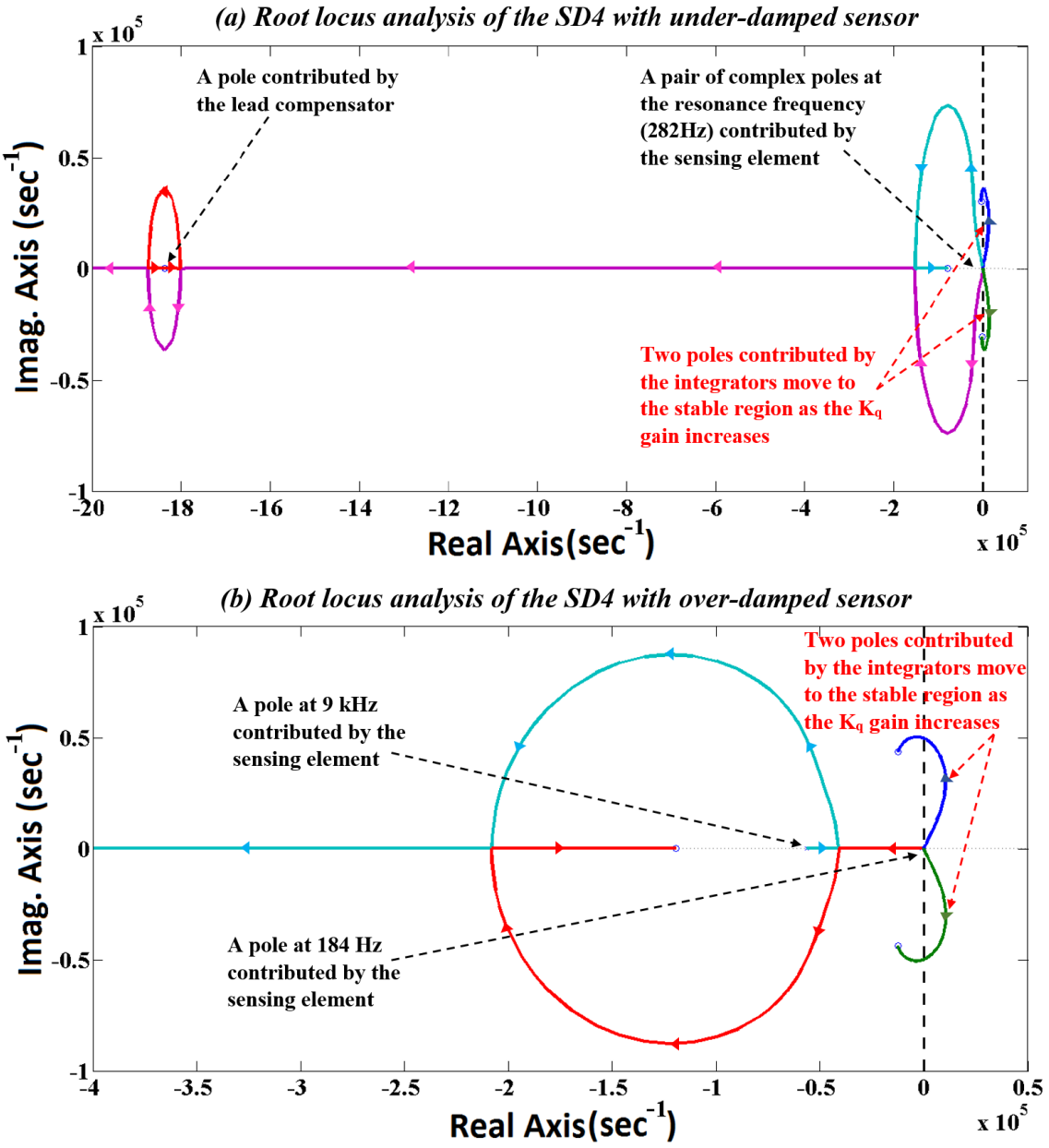


Figure 5-5: Root locus analysis of the open loop filter transfer function of the SD4 (a) underdamped and (b) over-damped sensor.

The same stability analysis is carried out for the SD4 for both sensors. The open loop filter transfer function of the SD4 is given by:

$$L4 = MK_{po}K_{bst}CK_1AK2BK_qK_{kf} + G_1K1AK2BK_q + G_2K2BK_q \qquad 5.4$$

Figure 5-5 shows the root locus analysis for the SD4 using (bottom) the underdamped sensor with a lead compensator and (top) the over-damped sensor. For both sensors, as the quantizer gain K_q increases, two poles that are contributed by the two integrators A and B, move into the stable region (left-hand side). Therefore, the SD4 system is conditionally stable with a minimum K_q value of 0.731 with the under-damped sensor and 0.143 with the over-damped sensor. In a practical implementation, in contrast to the SD2, the stability analysis for the high-order single-loop EM-ΣΔM obtained by the root locus is based on a linear model of the quantizer. Therefore, it cannot accurately predict the stability of a high-order single-loop EM-ΣΔM. Consequently, the design has to be verified with extensive and iterative simulations [55].

5.5 Input Signal Power versus SNR

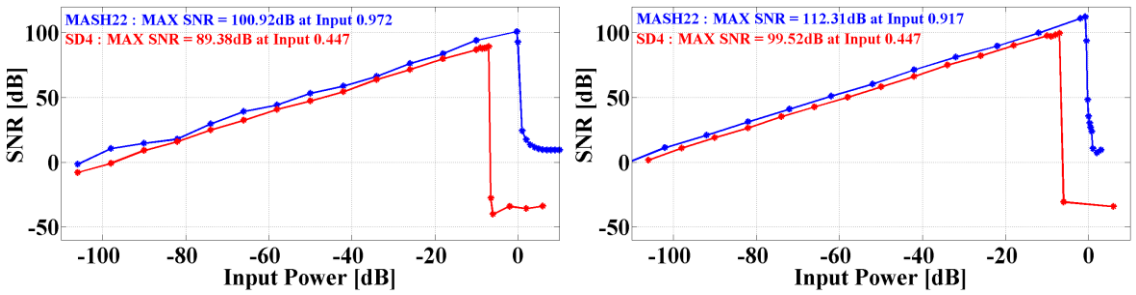


Figure 5-6: Input power versus SNR comparison for the over-damped (left) and underdamped (right) sensors

Using the over-damped sensor, the input signal power versus SNR plot shows the superior performance of the MASH22 over the SD4. Figure 5-6 (left) shows the SNR as a function of input signal power of the MASH22 with a maximum SNR of 100.9 dB. It can measure acceleration up to 1.1 g (0.97 of the full scale input). It also benefits from a wide dynamic range of 105 dB. In contrast, the SD4 shows lower performance than the MASH22, as it reaches a maximum SNR of 89.3 dB with an overload acceleration input of only 0.57 g. Due to its low overload input, the dynamic range for SD4 is only 89.4 dB. For the under-damped sensor, the MASH22 again proves its better performance, as it is able to

achieve a maximum SNR of 112.3d B with an overload acceleration input of 1.25 g as shown in Figure 5-6 (right). The SD4 performance is lower than that of the MASH22, where the maximum SNR is 99.5 dB. The dynamic range has also been reduced to 100 dB for the SD4.

5.6 Parameter Sensitivity

Monte Carlo analyses for both architectures were performed in this section using both sensors, in which the sensing element parameters were varied by 15% of their nominal values using a continuous uniform distribution function for 100 iterations. Further simulation was carried out to examine which of the sensing element parameters had the dominant effect on the performance.

5.6.1 Parameter Sensitivity Using Under-Damped Accelerometer

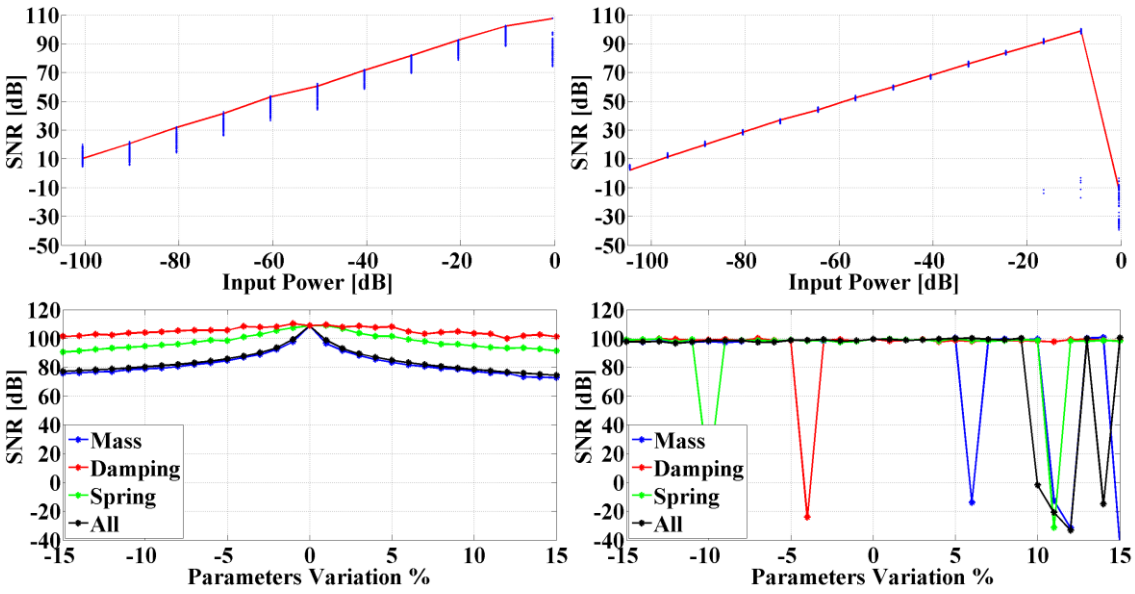


Figure 5-7: Monte Carlo analyses results (top) and parameter sensitivity analysis (bottom) for the MASH22 (left) and SD4 (right), using the under-damped sensor.

The Monte Carlo simulations for the MASH22 and SD4 are illustrated in Figure 5-7 (top). The top-left figure shows the performance of the MASH22, where the SNR is degraded by about 14% for the entire input power range except for the maximum acceleration input where it degraded by about 30%. The SD4 performance is presented in the top-right figure; where it shows a relative immunity to sensor parameter variations with a

maximum SNR degradation of only 5%. However, at the maximum input acceleration the SD4 became unstable and the performance hugely degraded.

Further SNR investigation is shown in Figure 5-7 (bottom), which depicts the three lumped model parameters of the inertial MEMS accelerometer: proof mass ‘m’, damping coefficient ‘b’, and spring constant ‘k’ are examined individually, to observe the parameter with the most critical effect when the maximum acceleration input is applied. The bottom-left panel in Figure 5-7 shows the MASH22 performance, where the proof mass parameter has the major degrading influence; this could reduce the SNR to -30% in the case of a ±15% variation of m, however, MASH22 remains stable for the whole variation percentages. It can also be seen that the existence of the lead compensator reduced the effect of the damping coefficient. The SD4 performance is presented on the bottom-right panel. In this case, the system could be assumed to be robust to parameter variation within the range of -3% to 5%, but the system clearly becomes unstable if the parameters exceed these variation percentages.

5.6.2 Parameter Sensitivity Using Over-Damped Accelerometer

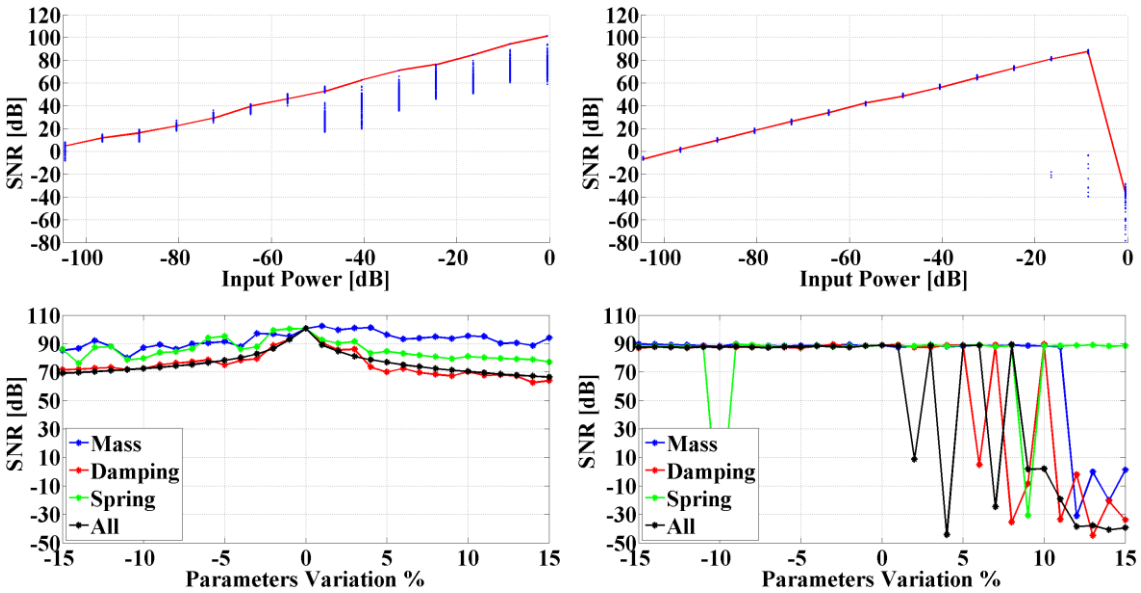


Figure 5-8 : Monte Carlo analyses results (top) and parameter sensitivity analysis (bottom) for the MASH22 (left) and SD4 (right), using the over-damped sensor

Same investigation was carried out on the over-damped sensor. The Monte Carlo results are shown in Figure 5-8 (top). Again, MASH22 shows a maximum of 25% degrading in the SNR, while the SD4 shows an immunity to sensor parameter variations with a maximum degradation of 5% at low input acceleration, however at acceleration input near the full scale, the SD4 exhibits instability due to large variation. The figure on the left shows the MASH22 performance, where the damping coefficient b has the major degrading influence; this could pull the SNR down to -30% in the case of a ±15% variation of b. The SD4 performance on the right shows that the system SNR sharply declines due to the instability of the SD4, which could be easily altered when parameters variation is introduced to the system.

5.7 Summary

In this chapter, two fourth order EM-ΣΔM were compared at the system level: a single-loop (SD4) architecture and a multi-stage (MASH22), using two different micromachined accelerometer sensing elements (one over-damped and the other underdamped). The comparison shows that the MASH22 has better noise shaping than the SD4 for the same over-sampling ratio (OSR = 64). It also shows that, after exhaustive simulations, SD4 was confirmed unable to reach the same SNR level as the MASH22 due to stability issues. For example, the simulation of the underdamped sensor showed the maximum SNR of the MASH22 at 111.2 dB with an overload acceleration of 1.25 g, while SD4 reached a maximum SNR of 98.7 dB with an overload acceleration of only 0.57 g.

	Sensor parameters	**Comparison criterion**	**MASH22**	**SD4**
Sensor 1	M = 1.5×10^{-6} kg	**Compensator**	No	No
	b = 0.007 N.s/m	**Max SNR**	101 dB	89.5 dB
	k = 98.1 N/m	**Over load**	1.1 g	0.47 g
	Q = 0.14 Over-damped	**Dynamic range**	105 dB	89.4 dB
Sensor 2	m = 1.7×10^{-6} kg	**Compensator**	Yes	Yes
	b = 3.6×10^{-4} N.s/m	**Max SNR**	112 dB	99.5 dB
	k = 5.5 N/m	**Over load**	1.25 g	0.57 g
	Q = 8.6 Under-damped	**Dynamic range**	110 dB	100 dB

Table 5-1: Comparison summary between MASH22 and SD4 for two different inertial accelerometer sensors.

In conclusion, the investigation reveals that the MASH22 is unconditionally stable and provides better noise shaping, higher SNR, higher dynamic range and a higher overload input level compared to the SD4, as summarized in Table 5-1. In addition, the comparison shows that the under-damped sensor has better performance for both structures compared with the over-damped sensor, due to the compensator. The only drawback of the MASH22 is its sensitivity to component and parameter tolerances, which leads to a leakage of the quantization noise to the input signal and a degradation of the performance of the modulator.

Chapter 6: MASH22 Hardware Implementation and Measurement Results

6.1 Introduction

The printed circuit board (PCB) is built using surface mount device (SMD) as shown in Figure 6-2; it is designed to accommodate either fourth order electromechanical single loop sigma-delta (SD4) or MASH22 architectures. The block diagram of the system is shown in Figure 6-1; it is designed with a differential line scheme, where the common noise along the path is eliminated. The system contains analogue and digital units, where extra care was taken to separate the ground layer between these two units to avoid digital signal coupling into the analogue circuit. The digital unit has the ability to transfer the acquired bitstreams to the PC via a USB link, and to receive controlling data from the PC to switch between SD2 or SD4. (Please refer to appendix D.1 for the circuit schematic)

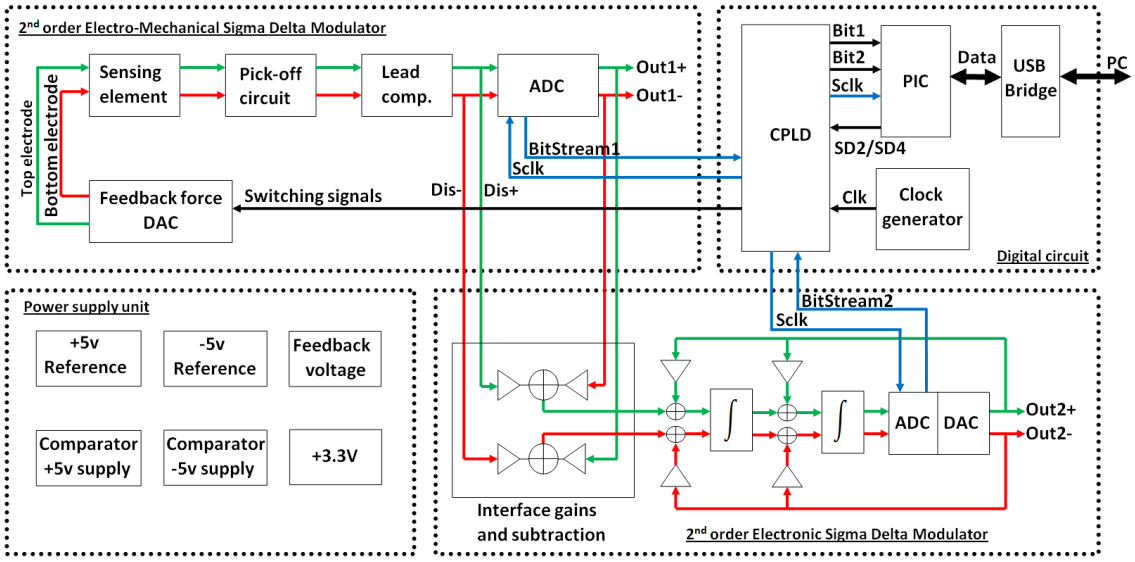


Figure 6-1: Electronic circuit block diagram.

The electronic circuit is divided to subunits. First is the SD2, which is composed of the sensing element, pickoff circuit, lead compensator, 1-bit ADC, and the feedback force circuit. Second is the 2nd order electronic ΣΔM, which is composed of the interface gain constants and subtraction circuit, summation and integration circuits, and 1-ADC. Third is the digital unit, which includes a complex programmable logic device (CPLD), peripheral interface controller (PIC), universal serial bus (USB) bridge and clock

generator. Finally, the power supply unit generates stable ±5 V for the comparators, feedback voltage, and the digital unit 3.3 V. The following sections will discuss each unit in more detail.

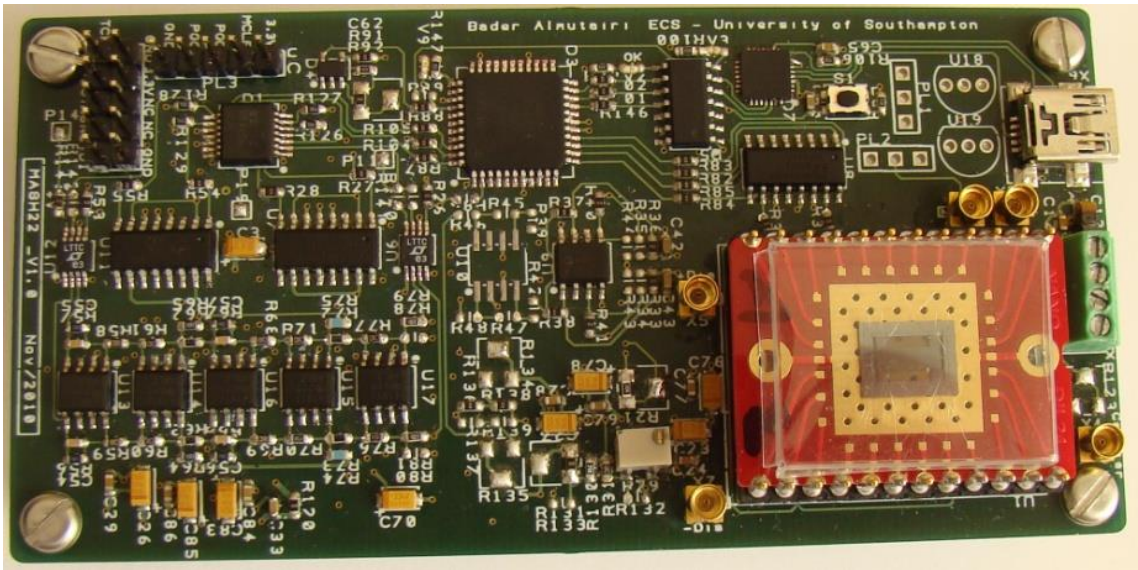


Figure 6-2: PCB circuit and capacitive MEMS accelerometer.

6.2 Sensing Element Characterization

The sensing element used in this research is a capacitive MEMS accelerometer; it was recently fabricated by the Southampton Nanofabrication Centre using a dicing free and dry release process [120] on silicon-on-insulator (SOI) wafer with device layer thickness of 50 µm. The accelerometer block diagram is depicted in Figure 6-3. The Qinetiq sensing element that was discussed in chapter 5, is not further investigated in this research.

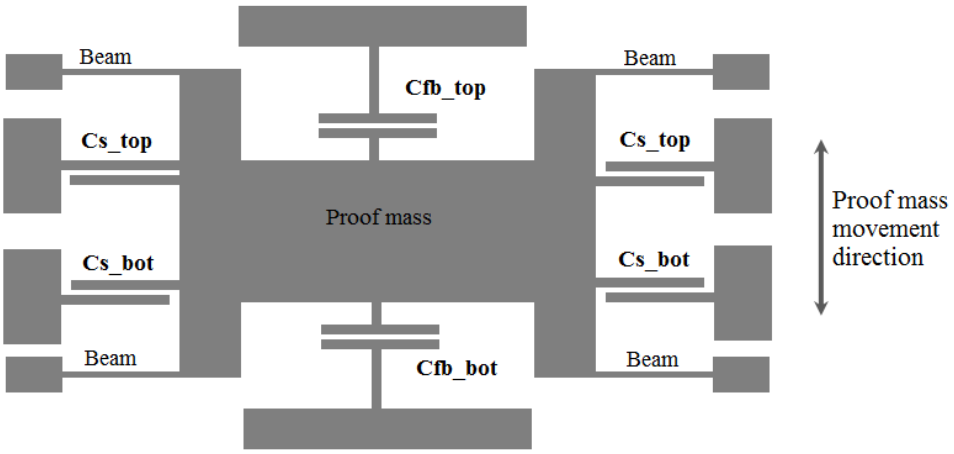


Figure 6-3: Capacitive MEMS accelerometer design.

The proof mass is supported with four elastic beams. The mass mechanical displacement is converted into a form of a capacitance change by the differential parallel capacitors C_{s_top} and C_{s_bot}. The proof mass is controlled by the differential capacitive actuators C_{fb_top} and C_{fb_bot}. The damping effect in the sensor arises mainly from the air squeeze film damping, between the electrodes of the capacitors. The scanning electron microscope (SEM) image of the MEMS accelerometer is shown in Figure 6-4 (left). The sensor was attached to the PCB package using crystal bond and wire-bonded to the corresponding pads to be electronically tested as shown in Figure 6-4 (right).

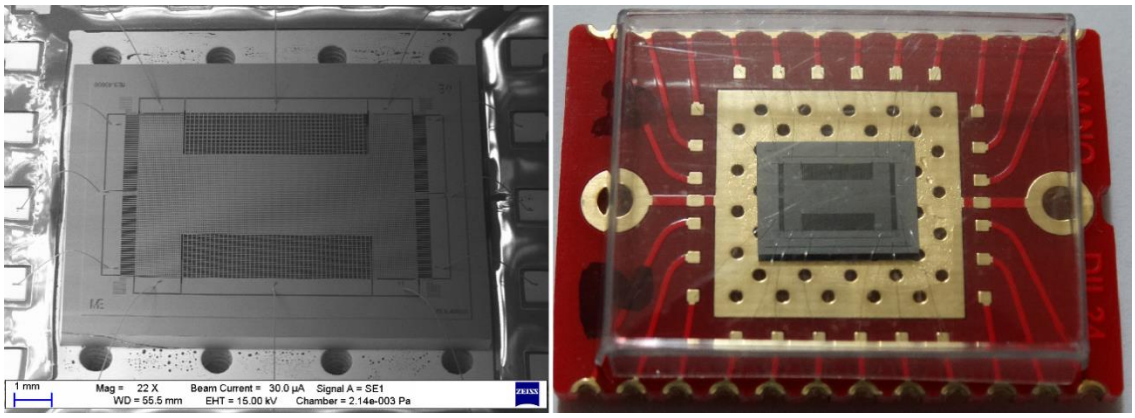


Figure 6-4: (Left) SEM image of the capacitive MEMS accelerometer fixed on a PCB carrier with a crystal-bond and wire-bonded to the PCB pads and (right) photo of the sensor with the transparent cap.

The design of the MASH architecture for MEMS requires correct characterization of the sensing element. As discussed in section 4.2, the digital filter D_2 needs the lumped model parameters of the sensor; i.e., proof mass 'm', damping coefficient 'b' and spring constant 'k'. It also needs the overlap area 'A' and the nominal gap 'd_0' between the sense electrodes in order to model the pickoff gain 'K_{po}' and it needs the overlap area of the feedback actuators to model the feedback gain 'K_{fb}'. These parameters are all theoretically derived from the fabrication mask layout, however, the MEMS accelerometer encounters a considerable degree of imperfection in the microfabrication process, and due to this issue, the lumped model parameters vary from the theoretical design, which leads to noticeable changes in the sensor dynamics. This effect has a serious impact on the MASH22 architecture performance; hence, extra attention was given to correct the theoretical parameters to more closely approximate the real ones.

6.2.1 Proof Mass Weight Calculation

The SOI wafer used in the fabrication process has a silicon device layer of 50µm thickness with ±1µm tolerance, hence, the proof mass weight calculation has to consider this tolerance. The proof mass was designed with etch holes, and due to the over-etching in the device layer etch process, the proof mass lost some amount of its original value. The calculation of the mass weight considered a 5% loss of the mass theoretical area due to over-etching and a 49 µm device layer thickness. Therefore, the mass weight was found to be:

$$m=1.622\times10^{-6}\ kg \tag{6.1}$$

6.2.2 Damping Measurement

The damping coefficient b in inertial MEMS sensors is mainly viscous air damping, which is proportional to the velocity of the proof mass, the measurement process estimates the damping ratio ζ instead of the damping coefficient b of the system. If the natural frequency ω_n (rad/s) and the mass weight m are known, the damping coefficient *b* of the sensor is given by:

$$b = 2\zeta\omega_n m \tag{6.2}$$

The resonant frequency ω_r of the damped accelerometer is given by:

$$\omega_r = \omega_n\sqrt{1-\zeta^2} \tag{6.3}$$

The damping ratio ζ of the sensor can be measured by impulse response in the time domain [121] or half power in the frequency domain [122].

- **Impulse response**

  The damping ratio of the MEMS accelerometer can be calculated in the time domain with the system impulse response. It uses the measurement of the proof mass consecutive peak displacements $x_1, x_2,\ldots,x_n$ as shown in Figure 6-5, where each peak has a drop in amplitude compared to the previous peak. The plot of the natural logarithm of each peak $ln(x_j)$ with its corresponding order *j* gives a straight line, the decaying slope of the line is denoted by δ and is designated the logarithmic decrement. The straight line can be given by [121]:

$$z_j = ay_j + b \tag{6.4}$$

where $z_j = \ln x_j$, $a = -\delta$, $y_j = j - 1$, and $b = \ln x_1$.

The damping ratio ζ can be given by [121]:

$$\zeta = \frac{|\delta|}{\sqrt{4\pi^2 + \delta^2}} \tag{6.5}$$

Since the measurement of the peaks is accompanied by error, all measurements may not fit on a straight line. Therefore, a least squares fitting can be applied to equation 6.4 to minimize this error, as follows:

$$\begin{bmatrix} \sum_{j=1}^{n} y_j^2 & \sum_{j=1}^{n} y_j \\ \sum_{j=1}^{n} y_j & n \end{bmatrix} \begin{bmatrix} a \\ b \end{bmatrix} = \begin{bmatrix} \sum_{j=1}^{n} (\ln x_j) y_j \\ \sum_{j=1}^{n} (\ln x_j) \end{bmatrix} \tag{6.6}$$

By solving equation 6.6 for *a* and *b*, the logarithmic decrement (δ=a) can be substituted in equation 6.5 to obtain the damping ratio ζ.

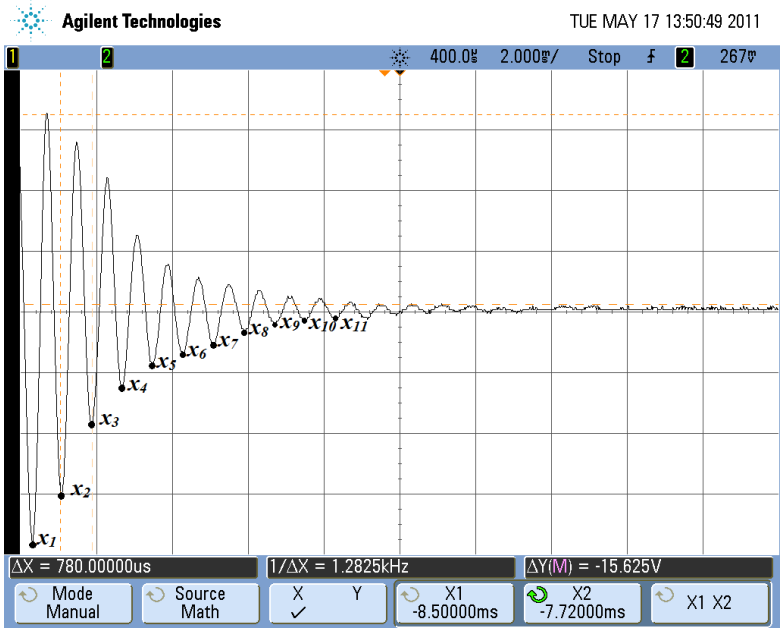


Figure 6-5: Accelerometer impulse response showing 1.282 kHz resonance frequency and 11 peak displacements.

j	x_j (v)	$Ln\ x_j$	y_j	$(Ln\ x_j)\ y_j$
1	19.5	2.9704	0	0
2	15.6	2.7473	1	2.7473
3	9.8	2.2824	2	4.5648
4	6.9	1.9315	3	5.7946
5	5	1.6094	4	6.4378
6	4.1	1.4110	5	7.0549
7	3.3	1.1939	6	7.1635
8	2.3	0.8329	7	5.8304
9	1.7	0.5306	8	4.2450
10	1.4	0.3365	9	3.0283
11	1.3	0.2624	10	2.6236

Table 6-1: Impulse response peaks values.

The peak measurements of the impulse response shown in Figure 6-5 are listed in the Table 6-1. By inserting the values from the table into equation 6.6, the solution becomes:

a= -0.2823 and b= 2.8758

Substituting δ = a in equation 6.5 results in:

Damping ratio ζ= 0.0449

- **Half power**

  Another approach to measure the damping is the half-power (Bandwidth method), which is based on studying the frequency response of the accelerometer at the resonance frequency f_r (Hz). The frequency response of the accelerometer is obtained in Figure 6-6 by applying an electrostatic force on the proof mass and observing the output of the pickoff circuit, which is proportional to the mass displacement. The resonance peak is shaped by the amount of damping found in the system. The bandwidth is determined by the region between the two frequencies f_1 and f_2 with the same magnitude. It is convenient to set the bandwidth with the half power of the resonance peak; i.e., $1/\sqrt{2}$ of the amplitude. The damping ratio ζ can be approximated by [122]:

$$\zeta = \frac{1}{2}\frac{f_2 - f_1}{f_r} \qquad 6.7$$

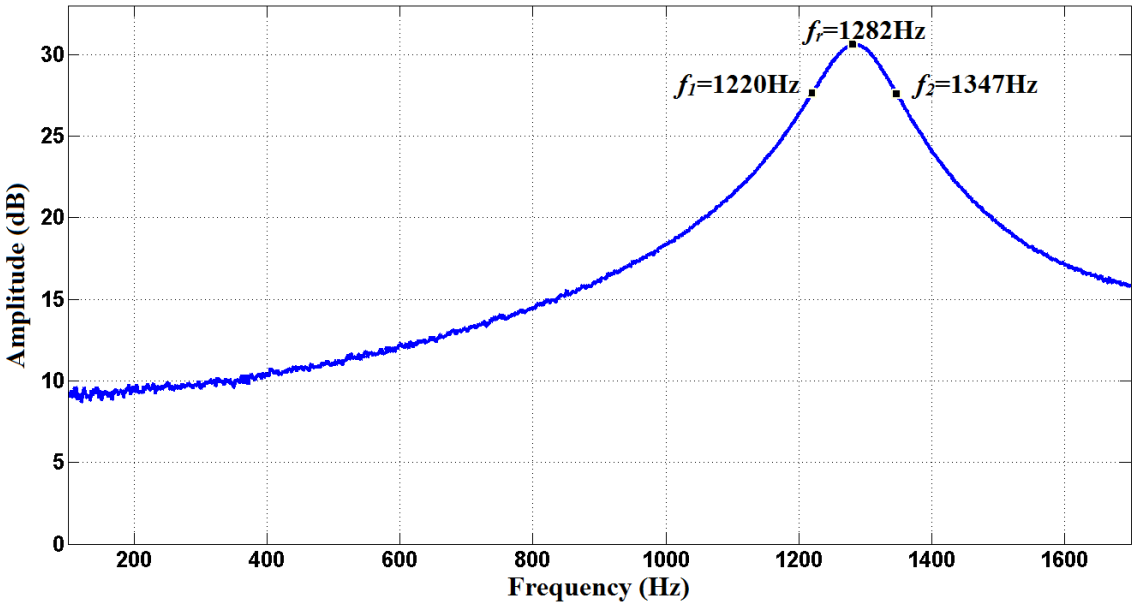


Figure 6-6: Electrostatic force frequency response of the accelerometer.

Figure 6-6 shows the frequency response of the accelerometer which indicates the resonance frequency f_r at 1.282 kHz, f_1 at 1.220 kHz and f_2 at 1.347 kHz. Using equation 6.7, the damping ratio ζ of the accelerometer can be estimated as follows:

$$\zeta = \frac{1}{2}\frac{1347 - 1220}{1282} = 0.0495$$

As can be seen, both methods give similar damping measurements; therefore, an average value ζ = 0.0472 can be taken from both results to calculate the damping coefficient b using equations 6.2 and 6.3 as follows:

$$\omega_n = \frac{1282}{\sqrt{1 - 0.0472^2}} = 8.0649\ k\ rad/sec$$

$$b = 2 \times 0.0472 \times 8.0649k \times 1.62\mu = 0.0012 \quad \frac{N}{m/s}$$

6.2.3 Spring Constant Measurement

During the experimental analysis, the spring constant is an important parameter that dramatically changes the sensor dynamics. If the natural frequency ω_n and the proof mass

m of the sensing element are known, the spring constant can be calculated using equation 2.4, as follows:

$$k = \omega_n^2 \times m = 8.0649k^2 \times 1.62\mu = 105.4 \ {}^{N}/_{m} \qquad 6.8$$

When the sensor is operated in closed loop mode, the feedback signal applied to one of the actuators, together with the carrier signal applied to the proof mass, add a negative electrostatic spring constant to the physical one, which shifts the resonance frequency of the sensor to a lower value [40].

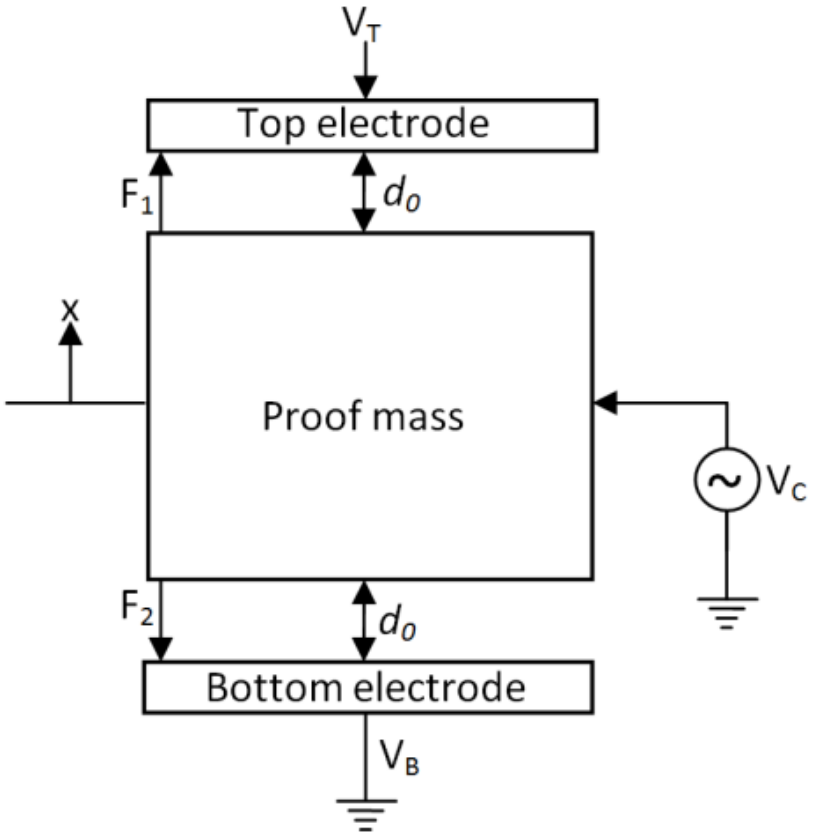


Figure 6-7: Sensing element model with feedback and carrier signals.

Figure 6-7 shows a block diagram of the sensor during a digital closed loop operation. For a nominal gap d0 and a displacement x, the electrostatic spring can be calculated by taking the derivative of the net electrostatic force with respect to the displacement, as follows [40]:

$$K_{elec} = \frac{d}{dx}(F1 - F2) \qquad 6.9$$

The net electrostatic force acting on the proof mass is given by:

$$F1 - F2 = \frac{1}{2}\varepsilon_0 A\left[\frac{(V_T + V_C)^2}{(d_0 + x)^2} - \frac{(V_C - V_B)^2}{(d_0 - x)^2}\right] \quad 6.10$$

At any given time during a digital closed loop operation, one actuator (V_B) is grounded and the other (V_T) is biased, while the carrier signal (V_C) is connected to the proof mass. If the V_C is a high frequency signal in a form of a sine wave, then V_C^2 is given by:

$$V_C^2 = V^2 sin^2(\omega_c t) = \frac{V^2}{2} \underbrace{- \frac{V^2}{2}\cos(2\omega_c t)}_{2\omega_c \gg \omega_n} = \frac{V^2}{2} \quad 6.11$$

The high frequency component in equation 6.11 can be neglected, since it is far beyond the sensor dynamic operation. Therefore, the net electrostatic force acting on the proof mass becomes:

$$F1 - F2 = \frac{1}{2}\varepsilon_0 A\left[\frac{{V_T}^2 + \frac{V^2}{2}}{(d_0 + x)^2} - \frac{\frac{V^2}{2}}{(d_0 - x)^2}\right] \quad 6.12$$

After some mathematical manipulations, and assuming small mass displacement (i.e., $x^4 < x^2 << d_0$), the net electrostatic force is given by:

$$F1 - F2 = \frac{1}{2}\varepsilon_0 A\left[\frac{{V_T}^2 d_0 - 2{V_T}^2 x - 2V^2 x}{d_0^3}\right] \quad 6.13$$

Taking the derivative of equation 6.13 with respect to the displacement x, the electrostatic spring (K_{elec}) is given by:

$$K_{elec} = \frac{d}{dx}(F1 - F2) = \frac{1}{2}\varepsilon_0 A\left[\frac{-2{V_T}^2 - 2V^2}{d_0^3}\right] \quad 6.14$$

Substituting V_T = 12 V, V_B = 0, and V = 6.5 V in equation 6.14 yields an electrostatic spring K_{elec} = -12 N/m. Table 6-2 shows that the resonant frequency of the sensor is shifted by about 80 Hz due to the electrostatic spring softening.

	Mass ‘m’ kg	Damping coeff. ‘b’ N.s/m	Spring constant ‘k’ N/m	Res. freq. ‘F_r’ Hz
Without K_{elec} effect	1.622 µ	0.012	105	1282
	If V_T = 12 V, V_B = 0, and V_C = 6.5 V $\xrightarrow{yields}$ K_{elec} = -12 N/m			
With K_{elec} effect	1.622 µ	0.012	93	1206

Table 6-2: Resonant frequency shift due to electrostatic spring softening.

An experimental test was performed on the sensor with the same voltage values for V_T, V_B and V_C to study the effect of the electrostatic spring K_{elec} during the closed loop operation. The result is shown in Figure 6-8, as the blue line shows a resonant frequency of 1.282 kHz when applying small voltage values (V_T = 3 V, V_B = 1 V and V_C =1 V), which do not have major effect on the resonance frequency. The red line shows that the resonant frequency was shifted to 1.202 kHz due to the electrostatic spring softening when higher voltage values are applied. The result closely agrees with the theoretical expectation shown in Table 6-2.

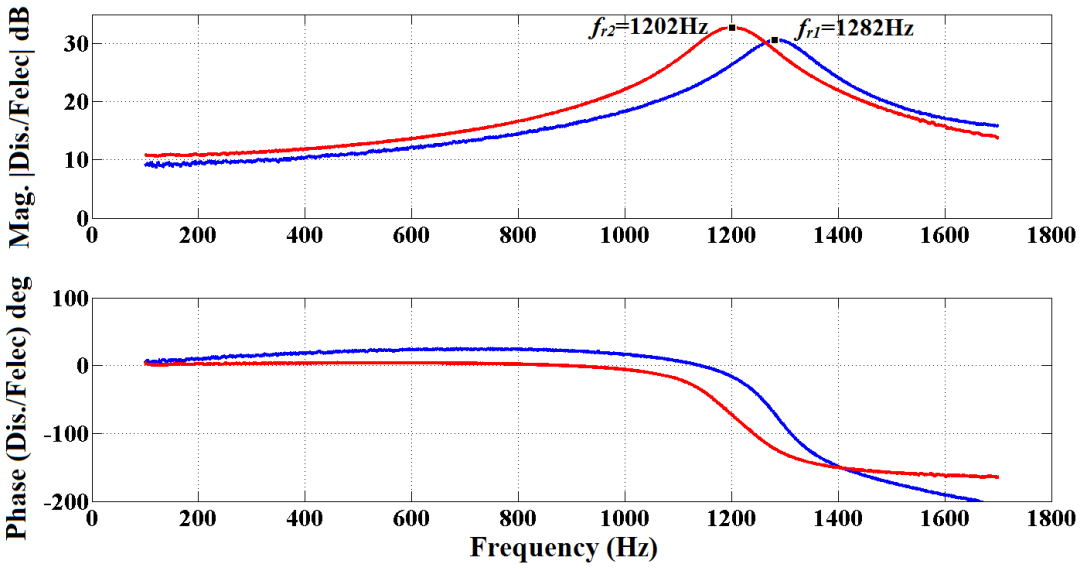


Figure 6-8: Bode plot of the sensor showing the resonant frequency shift due to the electrostatic spring softening.

6.3 The Pickoff Circuit

The main function of the pickoff circuit is to convert the change in capacitance due to mass displacement into a voltage that can be interfaced with the rest of the controlling circuit. Several methods can be used to implement the readout circuit; these can differ by circuit topology i.e. Switched capacitance (SC) [123] [124] or continues time (CT) [125], or modulation/demodulation techniques: Amplitude modulation (AM) [125] [126], where the change in capacitance ΔC is converted to a change in signal amplitude (i.e., capacitance-to-voltage conversion (CVC)) [127], or Frequency modulation, where the change in capacitance is converted to change in signal frequency (i.e., capacitance to frequency conversion (CFC) [68] [128]). The change in capacitance of a differential capacitive inertial MEMS sensor could be measured by a single-ended interface charge amplifier, or a differential interface charge amplifier, as shown in Figure 6-9. Generation of an accurate output V_x in the single ended circuit would require that both excitation signals $V_{carrier+}$ and $V_{carrier-}$ must be matched. Due to the unmatchable electric signals and the operating temperature, this method would result in inaccuracy in the output signal. The differential configuration solves these two issues by connecting two parallel charge amplifiers on each electrode. In this way, the outputs V_{x+} and V_{x-} work independently. Both outputs will be subtracted from each other and the final output will be free from the common error that existed in both outputs [15].

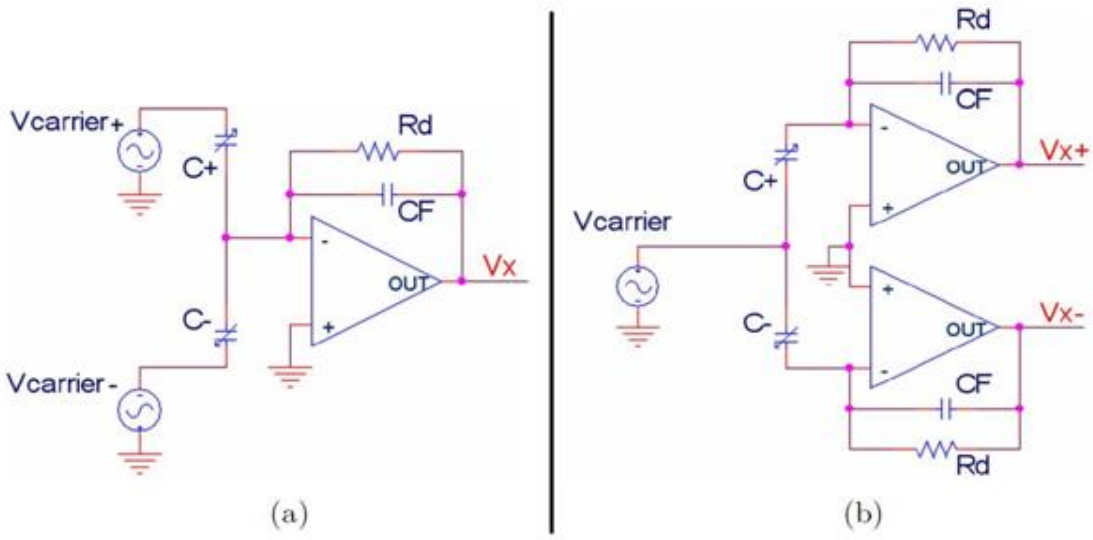


Figure 6-9: Differential capacitance charge amplifiers, (a) single-ended interface, and (b) differential interface.

The implementation of the pickoff circuit in this project is based on the differential capacitance to voltage converter for the CT circuits, with the use of the amplitude modulation technique. The basic circuit of the CVC is shown in Figure 6-10. The CVC is completely symmetrical and consists of: a) two frequency-independent half ac-bridges, which act as AM modulators, b) two AM demodulators, and c) an instrumentation amplifier that rejects common mode signals and provides the boost gain if necessary.

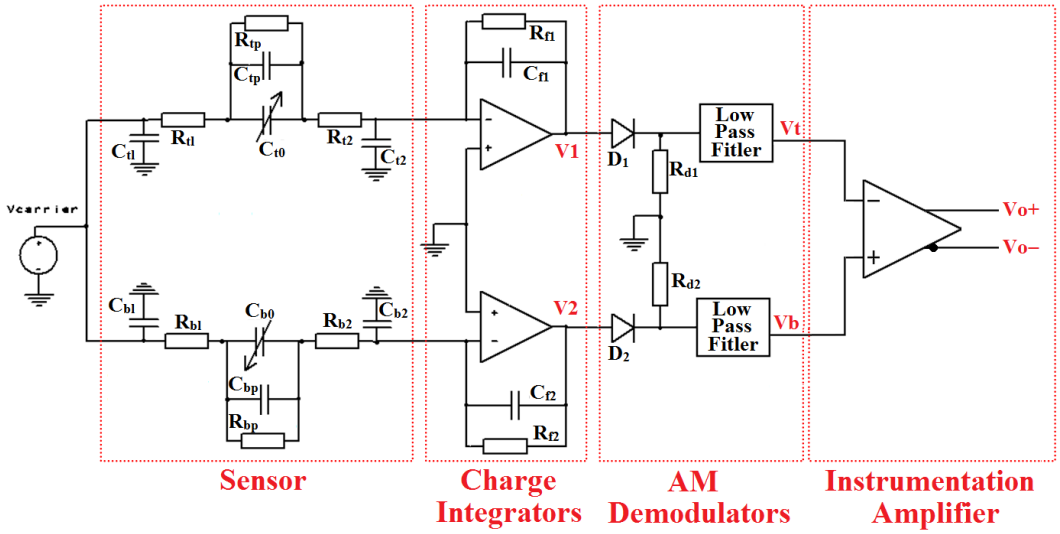


Figure 6-10: Pickoff circuit showing the sensor with the sense and parasitic capacitors, the charge amplifier, the AM demodulator and the instrumentation amplifier.

Since both top and bottom parts of the circuit in Figure 6-10 are symmetric in term of implementation and functionality, this discussion will focus on the top part as a single line circuit.

The sensing element with other parasitic components can be modelled as shown in Figure 6-10 [125]. It is connected by a voltage carrier source V_{carrier} to one of the sense electrode capacitances; the other electrode is connected to the virtual ground input of the charge integrator. Two parasitic capacitances C_{t1} and C_{t2} are found in parallel with the excitation voltage source and the input of the charge integrator; they do not have any influence on the measurement, as they do not change the current that goes through the sense capacitor C_{t0}. The main effective parasitic capacitance is C_{tp}, which is in parallel with the C_{t0}. The carrier signal is in a form of a sine wave and is given by:

$$V_{\text{carrier}} = A_{\text{carrier}} \sin(2\pi f_{\text{carrier}} t) \qquad 6.15$$

where A_{carrier} and f_{carrier} are the amplitude and the frequency of the carrier signal, respectively.

The series parasitic resistance R_{t1} models the wiring resistance and the equivalent series resistance of the sense capacitor C_{t0} and its value is typically less than 1 Ω. The parallel parasitic resistor R_{tp} models the insulation resistance of sense capacitor C_{t0} and its value is in the Mega Ω range, typically more than 200 MΩ [125] . The series parasitic resistance R_{t1} and R_{tp} together with $C_{tp} + C_{t0}$, introduce a high pass filter with certain cut-off frequency f_{c1} , and a low pass filter with certain cut-off frequency f_{c2}. Thus, the excitation voltage frequency has to be chosen in between these two filters' cut-off frequencies to eliminate the effect of these two parasitic components.

In the charge integrator, the feedback resistor R_{f1} provides the necessary DC bias current and forms, together with the feedback capacitor C_{f1}, a high pass filter with certain cut-off frequency f_{c3}. Another cut-off frequency f_{c4} is set by the Gain Bandwidth Product (GBP) of the Op-Amp, which acts as a low pass filter. Once again, the excitation voltage has to have a frequency higher than the charge integrator's cut-off frequency f_{c3} and lower than f_{c4}. The transfer function of $V_{carrier}$ to V_1 can be derived as follows:

$$\frac{\mathrm{V1}}{V_{\text{carrier}}} = -\frac{R_{f1}}{R_{tp}+R_{t1}} \times \frac{1 + s\,R_{tp}\left(C_{t0}+C_{tp}\right)}{\left(1+sR_{f1}C_{f1}\right)\left(1+s\,\frac{R_{tp}R_{t1}}{R_{tp}+R_{t1}}\left(C_{t0}+C_{tp}\right)\right)} \qquad 6.16$$

The Gain Bandwidth Product (GBP) of the Op-Amp and the three cut-off frequencies specified by equation 6.16 define a band pass region for the carrier signal $V_{carrier}$, as follows:

$$\mathrm{f_{carrier}} \gg \mathrm{f_{c1}} = \frac{\mathrm{R_{tp}}}{2\pi \mathrm{R_{tp}}(C_{t0}+C_{tp})}$$

$$\mathrm{f_{carrier}} \ll \mathrm{f_{c2}} = \frac{\mathrm{R_{tp}}+\mathrm{R_{t1}}}{2\pi \mathrm{R_{tp}R_{t1}}(C_{t0}+C_{tp})}$$

$$\mathrm{f_{carrier}} \gg \mathrm{f_{c3}} = \frac{1}{2\pi \mathrm{R_{f1}C_{f1}}}$$

$\mathrm{f_{carrier}} \ll \mathrm{f_{c4}} =$ Op-Amp GBP.

By choosing $f_{carrier}$ within a band pass region set by the above limits, equation 6.16 becomes frequency independent and can be simplified as follows:

$$\frac{V1}{V_{carrier}} = -\frac{C_{t0} + C_{tp}}{C_{f1}} \qquad 6.17$$

Equation 6.17 shows that the response of the charge integrator to the input carrier signal, within the band pass region, is amplified by $\frac{C_{t0}+C_{tp}}{C_{f1}}$ with 180° phase.

When C_{t0} changes due to acceleration, V1 changes accordingly; hence, amplitude modulation is achieved. If acceleration is applied to the sensor in the form of a sinusoid signal with frequency of f_{Acc}, the change in the sense capacitance C_{t0} can be represented as follows:

$$C_{t0} = C_{t0} + \Delta C_{t0} \sin(2\pi f_{Acc} t) \qquad 6.18$$

Substituting 6.17 in 6.18 yields an AM signal at the output of the Op-amp:

$$V1 = -\frac{C_{t0} + \Delta C_{t0} \sin(2\pi f_{Acc} t) + C_{tp}}{C_{f1}} V_{carrier} \qquad 6.19$$

AM demodulation is needed to recover the acceleration signal from the AM signal. There are two main approaches for the AM demodulation: phase-detection demodulation (synchronous demodulation) [39] [129] or peak detection demodulation (envelope detection demodulation) [37]. Synchronous demodulation makes use of analogue multipliers to employ a phase lock-in of the carrier signal, in which the AM signal (V1) is multiplied by a reference signal of the same frequency and the same phase of the carrier signal and then low-pass filtered. The result is a copy of the acceleration signal, but its scale factor depends on the phase difference between the carrier and the reference signal. This ensures that most of the noise is filtered out except for noise that is in phase with the carrier signal. This noise would include low frequency noise sources such as 1/f noise, actuating signal cross-talk and electromagnetic interference (EMI) [74]. Synchronous detectors are complex, requiring analogue multipliers and other components to ensure phase matching between the carrier signal and the reference signal.

Envelope detection demodulation can be constructed using a rectifier, which rectifies the signal and leaves only the positive part to flow past a low pass filter to remove the high

frequency carrier signal and recover the acceleration signal. The rectifier can be a simple diode, as depicted in Figure 6-11, or precision rectifier circuit, as shown in Figure 6-12 [130]. Unfortunately, diode rectifiers are not linear and can cause signal distortion, especially for low-level input signals. Rectification of signals smaller than a diode voltage drop cannot be performed with a simple diode, so a precision rectifier using op-amps is used in order to have a circuit that behaves like an ideal diode and has the ability to control the output gain by simple resistors.

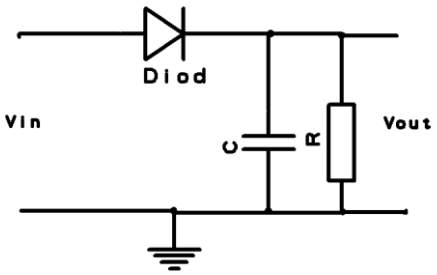


Figure 6-11: Diode rectifier circuit.

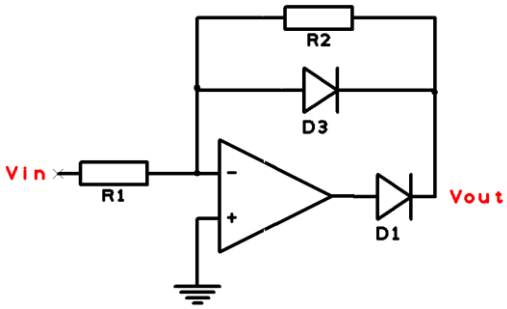


Figure 6-12: Precision rectifier circuit.

The first prototype of the PCB circuit of this project makes use of the diode envelope demodulator because of its simple implementation with a low pass filter with a cut-off frequency f_{cd} equal to half of the ΣΔM sampling frequency $f_s/2$ = 67.5 kHz. The output of the demodulator (V_t) can be represented by the following equation:

$$V_t = -\frac{C_{t0} + \Delta C_{t0}\sin(2\pi f_{Acc}t) + C_{tp}}{C_{f1}} A_{\text{carrier}} - V_{diode} \qquad 6.20$$

where V_{diode} is the forward voltage of the diode.

Similarly, the output of the modulator at the second end of the differential line (V_b) can be represented as follows:

$$V_b = -\frac{C_{b0} - \Delta C_{b0}\sin(2\pi f_{Acc}t) + C_{bp}}{C_{f2}} A_{\text{carrier}} - V_{diode} \qquad 6.21$$

The nominal capacitance for both sense electrodes can be assumed equal; i.e., $C_{t0} = C_{b0} = C_0$. The final part of the pickoff circuit is the instrumentation amplifier, which amplifies the difference between V_t and V_b with a gain factor of K_{inst}, as follows:

$$V_{out} = K_{inst} * (V_t - V_b) \qquad 6.22$$

$$V_{out} = K_{inst}\left(-\frac{C_0 + \Delta C_x \sin(2\pi f_{Acc}t) + C_{tp}}{C_f} A_{\text{carrier}} - V_{diode} + \frac{C_0 - \Delta C_x \sin(2\pi f_{Acc}t) + C_{bp}}{C_f} A_{\text{carrier}} + V_{diode}\right) \quad 6.23$$

The parasitic capacitance $C_{tp}$and C_{bp} can be assumed equal; hence, their effects can be removed by the differential behaviour of the circuit and the output can be simplified as:

$$V_{out} = -K_{inst} A_{\text{carrier}} \frac{2\Delta C_0 \sin(2\pi f_{Acc}t)}{C_f} \quad 6.24$$

Substituting 2.5 in 6.24 yields:

$$V_{out} = -K_{inst} A_{\text{carrier}} \frac{2\varepsilon_0 A x}{d_0^2} \frac{\sin(2\pi f_{Acc}t)}{C_f} \quad 6.25$$

$$V_{out} = -2K_{inst} A_{\text{carrier}} \frac{x C_0}{d_0} \frac{\sin(2\pi f_{Acc}t)}{C_f} \quad 6.26$$

Finally, the pickoff gain (K_{po}) can be given by:

$$K_{po} = \frac{V_{out}}{x} = -2K_{inst} A_{\text{carrier}} \frac{C_0}{C_f d_0} \qquad [\frac{v}{m}] \quad 6.27$$

Figure 6-13 shows the schematic diagram of the pickoff circuit, which was designed and simulated using OrCAD – Pspice software. The sense capacitance and the pickoff circuit are implemented with a differential line scheme (top and bottom lines), both lines are symmetrical in term of functionality and components. The simulation analysis of the pickoff circuit included a model of the capacitive MEMS accelerometer as second order function H(s). The input acceleration is a sinusoidal signal of 6 m/s^2 at 256 Hz, accordingly, the accelerometer responds with a proof mass displacement of 90 nm amplitude, which is indicated Figure 6-14 (a). The change in capacitance due to the mass displacement (i.e. Δx to ΔC) is modelled using the YX variable admittance block. The main effective parasitic capacitors (C_{tp} and C_{bp}) were measured and included in the simulation, which are in parallel to the sense capacitors. The charge integrator was designed to measure the change of capacitance and output AM signal, as shown in Figure 6-14 (b). The diode rectifier outputs only a voltage signal above the forward bias voltage as shown in Figure 6-14 (c). The low pass filter removes the high frequency carrier signal and leaves the low frequency proof mass movement as shown in Figure 6-14 (d). Finally, the instrumentation amplifier combines the differential movement of the proof mass and rejects the common mode signals as shown in Figure 6-14 (e).

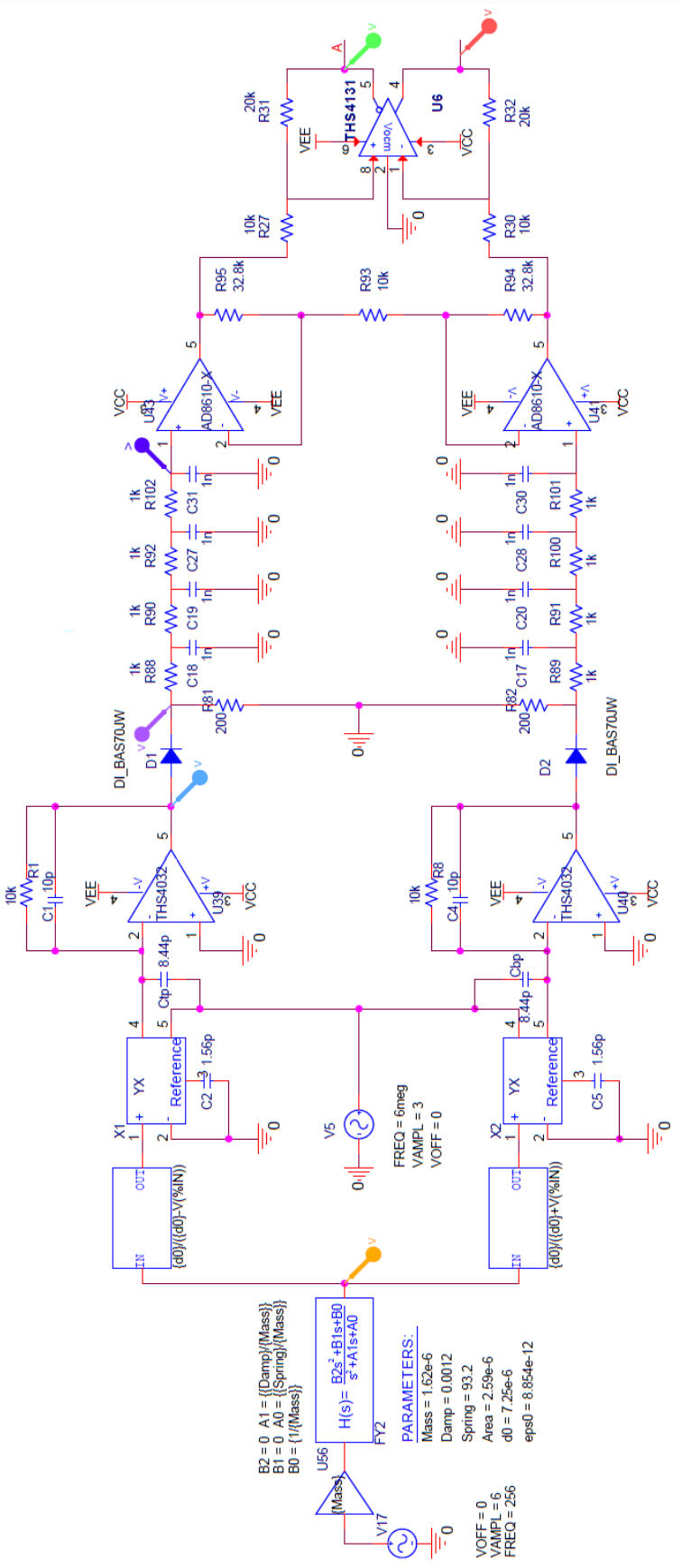


Figure 6-13: Pickoff circuit OrCAD/PSpice schematic diagram.

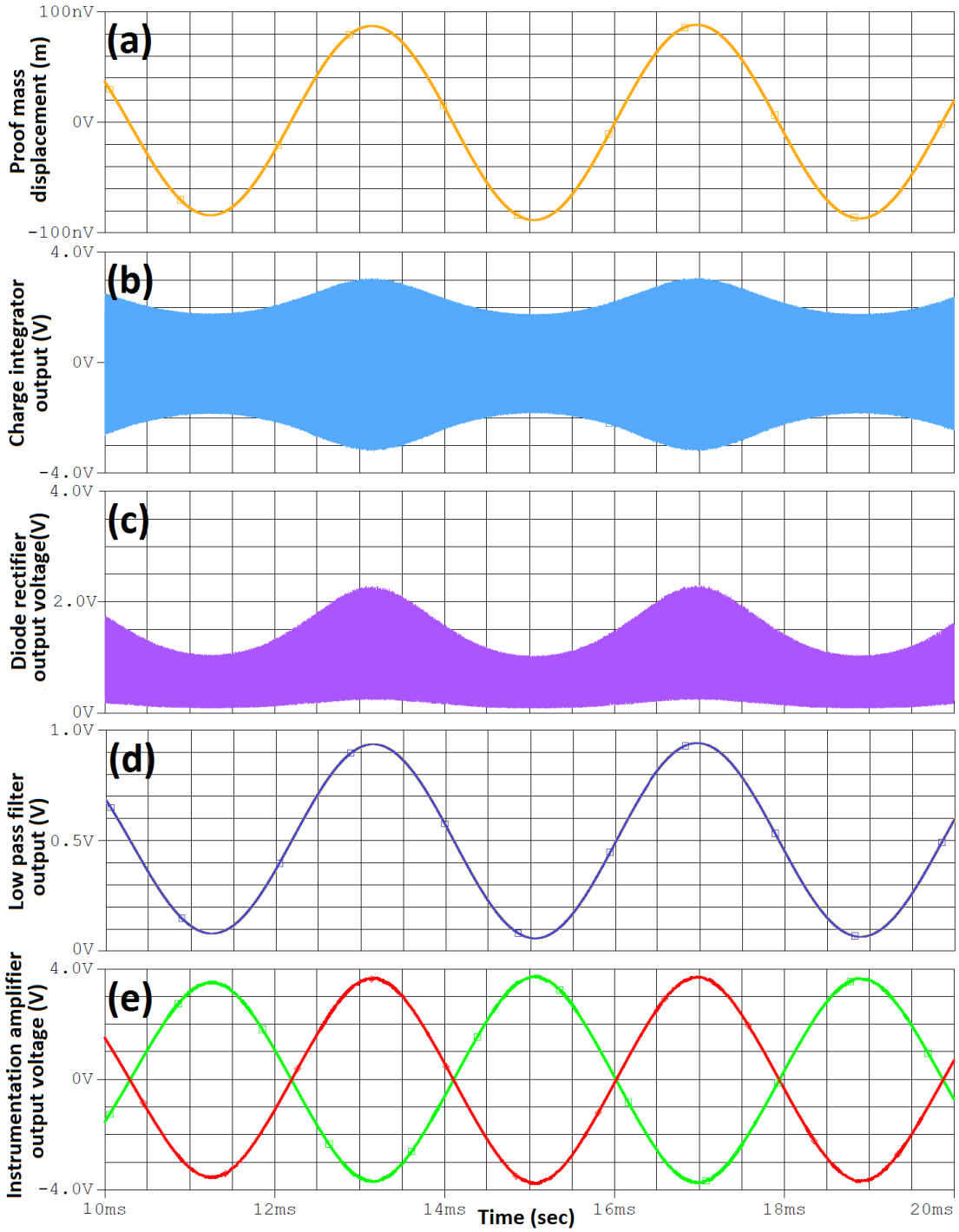


Figure 6-14: Pickoff circuit simulated measurement using OrCAD/PSpice showing (a) the proof mass displacement due to acceleration, (b) the output voltage of the charge integrator AM signal with 6 MHz frequency, (c) the output voltage of the rectifier diode, (d) the AM demodulated signal at the output of the low pass filter, and (e) the instrumentation amplifier differential output.

6.3.1 Pickoff Gain Calculation Using Optical Measurement

The digital filter D_2 design requires an accurate matching of the pickoff gain K_{po}, which is simply the ratio between the output voltage of the pickoff circuit to the mass displacement. One way to determine the K_{po} value is to use the optical measurement machine (MSA400), as shown in Figure 6-15. The idea is to measure the mass displacement 'x' due to the applied electrostatic force and observe the output voltage of the pickoff circuit.

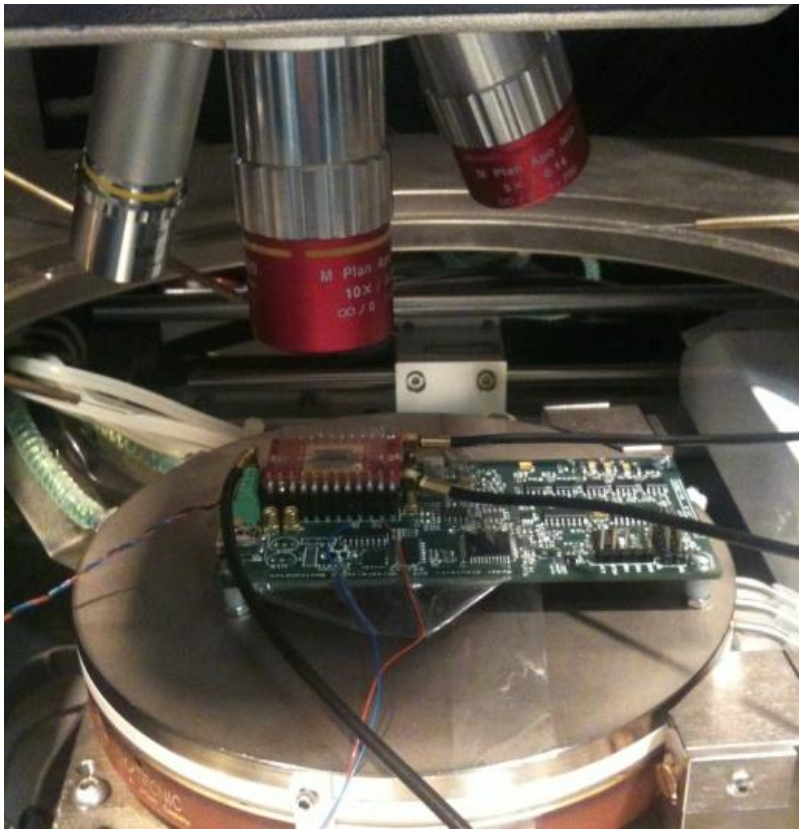


Figure 6-15: System under optical measurement test.

An experiment was performed to measure the pickoff gain K_{po} and the spring constant 'k'. It was performed in two steps, as follows:

1- Using the shaker table, the output of the pickoff circuit was measured for 1 g acceleration. The result shows a pickoff circuit output of 16.3 V/g, as shown in Figure 6-16.
2- Using the optical measurement system, the electrostatic force was tuned such that the pickoff circuit put out a voltage signal equivalent to the 1 g acceleration response obtained in step-1 (i.e. 16.3 V/F_{elec}). The result is shown in Figure 6-17.

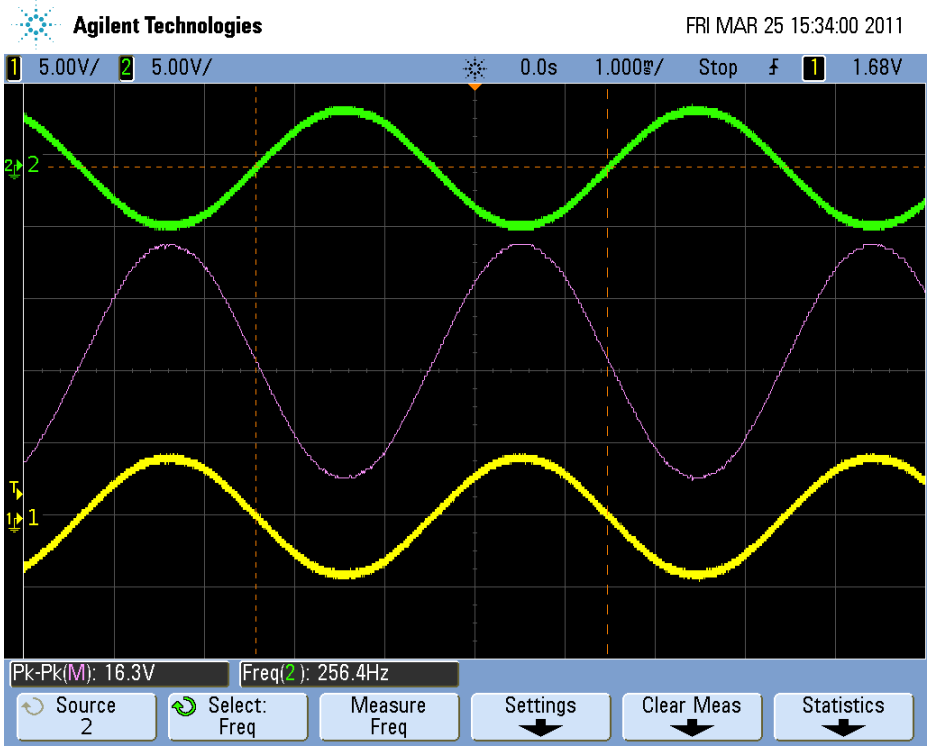


Figure 6-16: Output of the differential line pickoff circuit when 1g acceleration is applied to the sensor: green and yellow are the differential outputs, and purple is the difference of those two signals.

The optical measurement in Figure 6-17 shows the mass displacement due to the applied electrostatic force. The output of the pickoff circuit due to the applied electrostatic force was equal to that obtained due to the acceleration, as shown in Figure 6-16. Thus, the applied electrostatic force equals the inertial force of 1g acceleration, and the mass displacement was 155 nm. Therefore, the pickoff gain K_{po} is equal to:

$$1\ g\ \ (acceleration) \xrightarrow{yields} 155\ \text{nm}\ \ (\text{displacement}) \xrightarrow{\text{yields}} 16.3\ \text{V (pickoff output)}$$

$$K_{po} = \frac{16.3}{155\text{n}} = 105.161 \times 10^{6} \quad [\text{V/m}] \qquad 6.28$$

The spring constant k can be verified as follows [15]:

$$k = m \times \frac{a}{x} = 1.622\mu \times \frac{9.81}{155n} = 102.6\ \text{N/m} \qquad 6.29$$

The spring constant result closely agrees with the result obtained in equation 6.8.

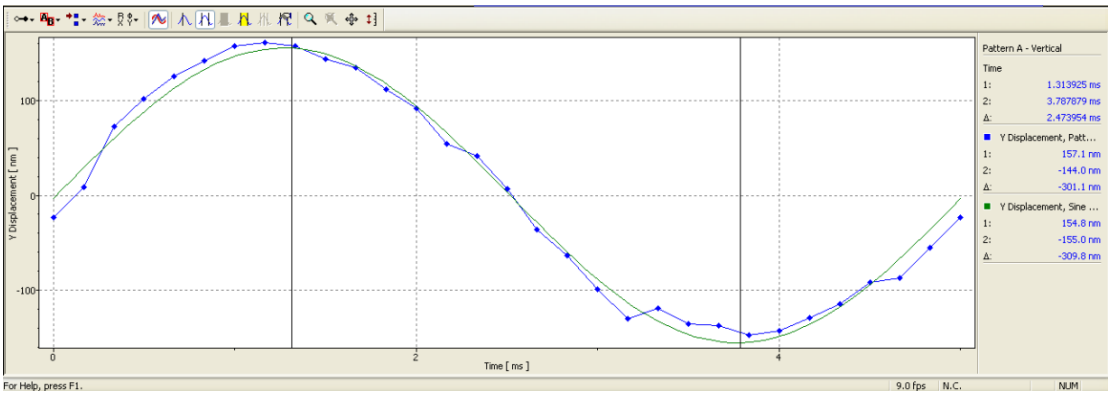


Figure 6-17: MSA400 optical measurement shows the proof mass displacement due to the applied electrostatic force (equivalent to 1 g acceleration).

6.3.2 The Effect of Feedback Signal Cross Talk on the Pickoff Circuit Output

One of the major problems in the design of the closed loop accelerometer is the cross-talk between the feedback and sense capacitors. The feedback signal, which occurs in the form of pulses, is coupled into the sense capacitors. The coupling is due to the parasitic capacitance $C_{fb\_tp1,\,2}$ and $C_{fb\_bp1,\,2}$, as shown in Figure 6-18. If the top feedback electrode is energized, and the bottom is grounded, the top parasitic capacitors $C_{fb\_tp1,\,2}$ are closer to the energized electrode. Hence, a larger coupling signal is fed to the top sense capacitor, while the $C_{fb\_bp1,\,2}$ is relatively far from the top electrode, and hence, a smaller coupling is fed to the bottom sense capacitor. Because of the difference in the coupled signal amplitude, the differential configuration of the pickoff circuit will not solve this issue.

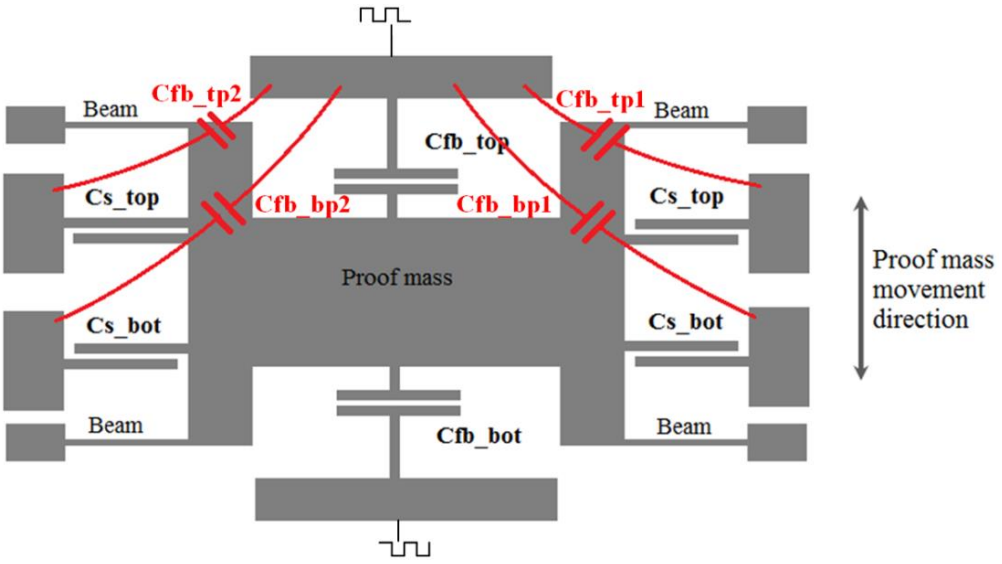


Figure 6-18: Feedback parasitic capacitor model.

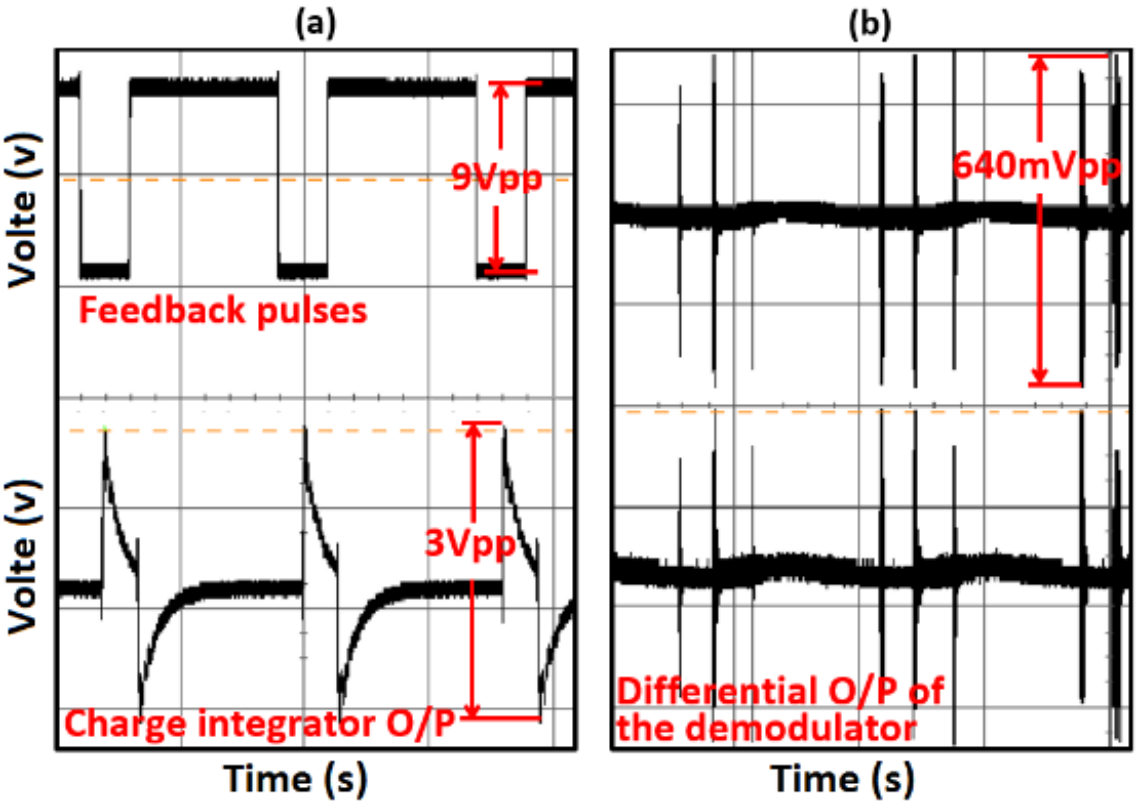


Figure 6-19: (a) Upper trace is the feedback digital signal and lower trace is the output signal from charge integrator. (b) is the differential output of the pickoff circuit, which shows the feedback voltage cross coupling effect.

During the digital closed loop operation, the feedback signal reaches a fundamental frequency equal to the limit cycle frequency, which was experimentally found to be 10 kHz. An experiment was carried out to study the effect of feedback parasitic capacitors. The proof mass was grounded, and both feedback electrodes were alternatively energized with pulses of 0 to 9 V amplitude and 10 kHz frequency. Figure 6-19(a) shows the applied digital feedback signal in the upper trace, and the cross talk signal in the lower trace, observed at the output of the charge integrator. The output signal consisted of a low frequency component equal to the applied feedback signal frequency, and a high frequency component due to the rapid transition of the feedback signal; i.e., from 0→9 V. The differential output of the demodulator shown in Figure 6-19(b) is contaminated by the coupled signals; this strongly affects the performance of the closed loop sigma-delta modulator.

The cross talk signal can be eliminated in several ways, including:

1- Time multiplexing scheme between the feedback and the sense electrodes [74], where the feedback signals are applied in a certain time, called the feedback phase, before they both are grounded in the quiet phase. In theory, grounding the feedback electrodes during the quiet phase gives the parasitic capacitance enough time to discharge and offers a relatively error free pickoff output before the next sampling clock occurs. In practice, this method does not give the expected performance as it introduces extra switching noise and requires the quiet phase to be relatively long, which degrades the closed loop performance.

2- Implementing a high pass or a band bass filter before the demodulator in the pickoff circuit. This will filter out the coupled signals caused by the digital feedback signals.

Because the system is already built with a pickoff circuit that consists of a charge integrator, diode and low pass filter, the second method can be effectively implemented. The solution is to shift the cut-off frequency of the charge integrator to a higher frequency, where any signal of a frequency less than this cut-off will be heavily attenuated. The low pass filtering during the demodulation process also ensures that any high frequency component is filtered out. Thus, only the modulated acceleration signal (i.e., proof mass movement), will pass through the high pass and the low pass filter unaffected.

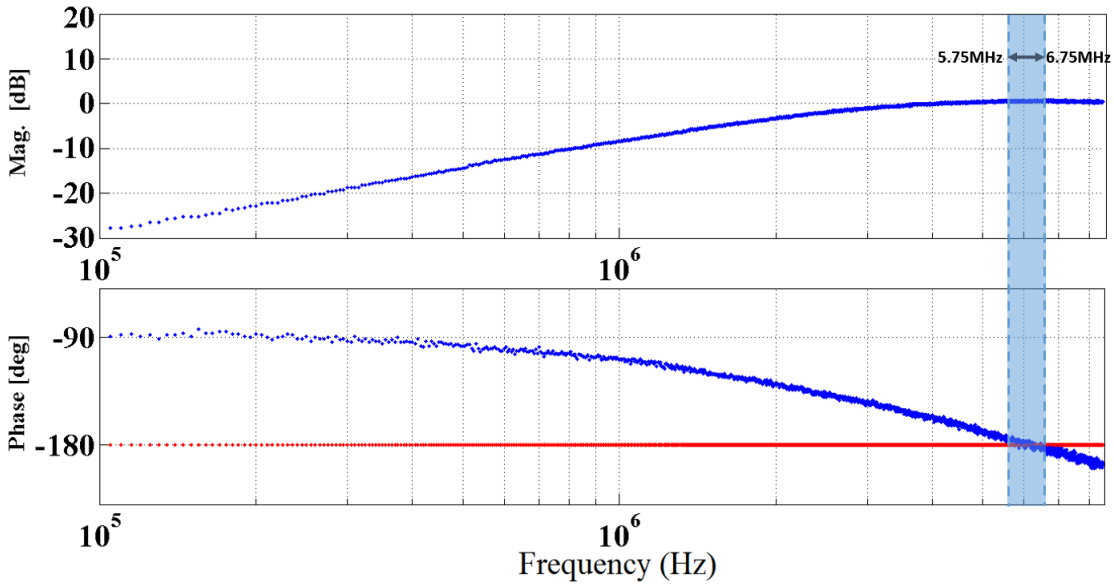


Figure 6-20: Frequency response of the charge integrator. The carrier signal was chosen at 6 MHz, which is within the indicated region (5.75 - 6.75 MHz).

The charge integrator was designed with high cut-off frequency of 3 MHz by choosing a small feedback resistance (R_F=10 kΩ) with the appropriate feedback capacitance. The Bode plot of the charge integrator is shown Figure 6-20. The carrier signal was chosen in the narrow region between 5.73 to 6.73 MHz where the response of the charge integrator gives 180° phase shift.

The low pass filter of the pickoff circuit was designed with a cut-off frequency at 62.5 kHz. The design was verified by emulating the feedback signal and grounding the proof mass. The results are shown in Figure 6-21(a); the output of the charge integrator signal is the lower signal which shows that, only the high frequency component due to the rapid transition of the feedback signal is passed through the charge integrator. Figure 6-21(b) shows the differential output of the demodulator for both sense electrodes, where the coupled signals are heavily attenuated.

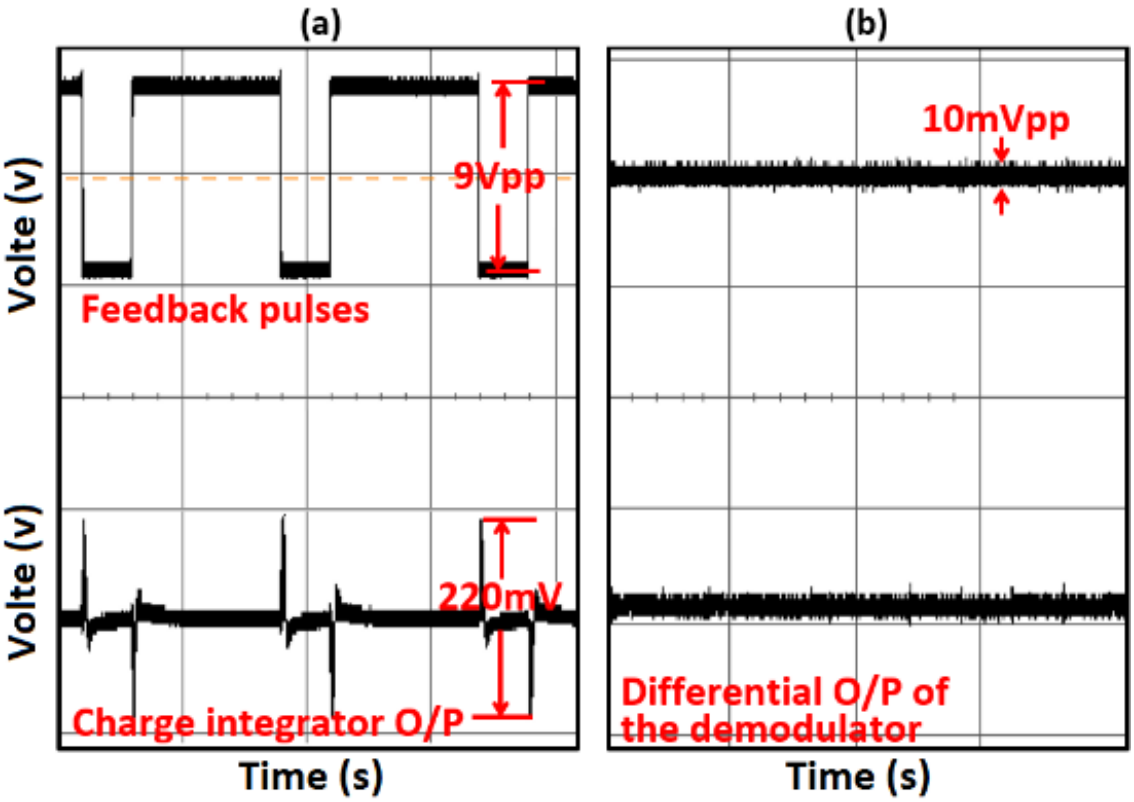


Figure 6-21: Left-upper trace is the feedback digital signal and left-lower trace is the output signal from charge integrator due to the coupling, but it is noticeably reduced due to the high pass effect. The right figure is the differential output of the pickoff circuit, which shows no visible feedback voltage cross coupling effect after the low pass filter.

6.4 Lead Compensator Circuit

The design of the second order EM-ΣΔM has to consider the phase lag along the control signal path; i.e., the output signal from the capacitive transducer back to the capacitive actuator. It is a crucial task to ensure feedback loop stability, by means of applying the proper actuation signal at the right time to return the displaced proof mass to its nominal position. Thus, the lead compensator circuit is an important design unit in the closed loop under-damped accelerometers. The propagation delay of the signal caused by the electronic circuit also has to be addressed in the lead compensator design.

There is no systematic approach to design the lead compensator for electromechanical sigma-delta modulators. One rule of thumb is to design a lead compensator with a relation to the sampling frequency f_s of the sigma-delta modulator; by setting the lead compensator pole at $f_s/5$ and the zero at $5f_s$; [39] however, this approach may not lead to the optimum design. Another approach to obtain the centre frequency and the phase lead is by including all possible delays in the signal path in the closed loop Simulink model, and then running an iterative simulation with a frequency sweep – each frequency with different phase values – until a maximum SNR value is achieved. The genetic algorithm (GA) approach can also be used to achieve a similar result [98].

Once the centre frequency ω_{com} (rad/sec) and the phase lead ϕ (rad) are chosen, the lead compensator's pole and zero are given by [88]:

$$\alpha = \frac{1 + sin(\phi)}{1 - sin(\phi)} \tag{6.30}$$

$$zero = \frac{\omega_{comp}}{\sqrt{\phi}} \tag{6.31}$$

$$pole = \alpha * zero \tag{6.32}$$

The electronic realization of the lead compensator is achieved by calculating the equivalent resistor and capacitor values of the active op-amp electronic circuit, as follows:

$$C_f = 1nF \qquad C_i = \frac{pole}{zero} \times C_f \tag{6.33}$$

$$R_i = \frac{1}{zero \times C_i} \qquad R_f = \frac{1}{pole \times C_f} \qquad 6.34$$

Figure 6-22 shows the differential design of the electronic lead compensator. This circuit has a negative polarity output that has to be considered in the closed loop design.

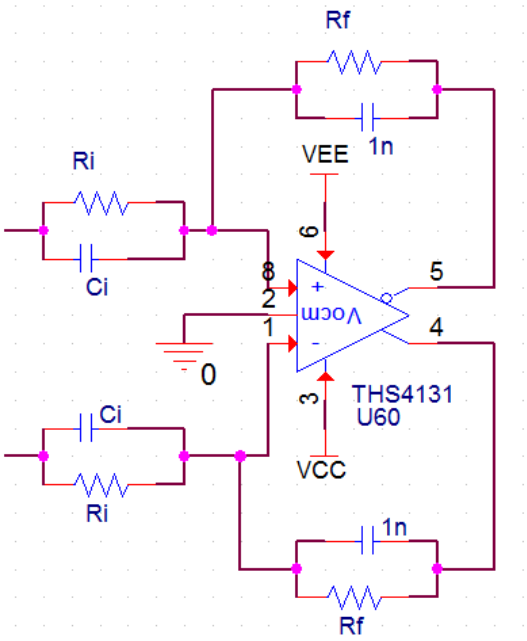


Figure 6-22: Electronic circuit for the lead compensator.

6.5 Sampling and Quantization (ADC)

The function of the ADC blocks, as shown in Figure 6-1, is to sample and quantize the analogue signal to a 1-bit digital output. The digital output is in the form of non-return-to-zero (NRZ) line coding; i.e., the logic 1 and 0 pulses are represented with a positive and negative voltage (e.g. ±5 V). The output of the ADC is designed to be differential (bit+ and bit-) at any given time. The real implementation of the ADC consists of a differential input comparator, D Flip-Flop, and analogue switch, as shown in Figure 6-23.

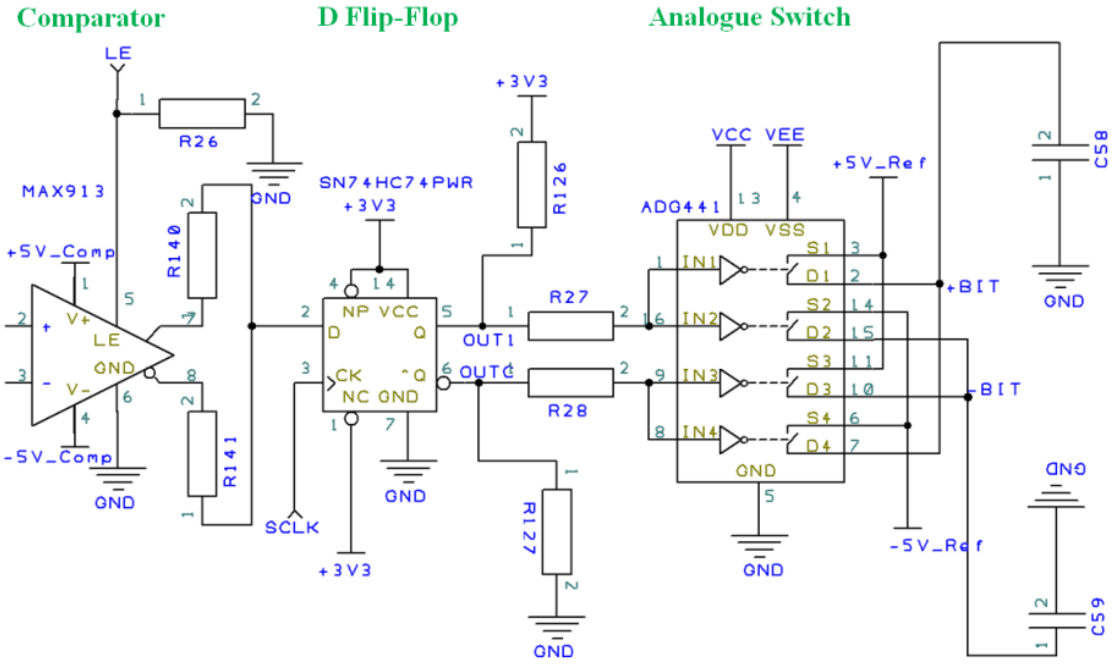


Figure 6-23: ADC electronic circuit.

The differential input comparator operates by a dual supply ±5 V. It continuously quantizes the difference between its input signals. The overload differential input for the comparator is ±10 V, which sets the maximum limit for the pickoff gain. The D Flip-Flop samples and holds the output of the comparator for a period of $1/f_s$, where f_s is the sampling frequency. Finally, the single output of the D Flip-Flop is fed to the analogue switch, which in turn outputs a differential line signal of ±5 V, which will be used as feedback control signal.

The quantization error is the difference between the output and the input of the ADC unit (Out and Dis. signals, respectively). As discussed in section 4.2, the difference signal has to be properly scaled, with the scaling gains KR, KS and K2, in order not to overload the second loop. The hardware implementation of this unit holds the same differential signal line scheme. The subtraction is achieved using a summer circuit that has inputs of a crossed signal polarity of the differential lines; i.e., the displacement signal (Dis-) is added to the ADC output (Out+), as shown in Figure 6-1.

6.6 Electrostatic Feedback Force

The implementation of the DAC in the electromechanical sigma-delta modulators is achieved using analogue switch. A negative feedback force is achieved by means of electrostatic force, by applying the required voltage pulse on one actuator (e.g., the top electrode), while the other actuator (e.g., the bottom electrode) is grounded, and vice versa, based on the sign of the bit-stream. The analogue switch is controlled by the CPLD, as illustrated by the block diagram in Figure 6-24. The CPLD unit will be discussed in section 6.9.1.

The equivalent electrostatic force generated by the digital pulses is given by [15]:

$$K_{fb} = \frac{\frac{1}{2}\,\varepsilon_0\, Area\, V_{fb}^2}{d_0^2} \qquad 6.35$$

where $\varepsilon_0$is the air permittivity, Area is the actuator overlap area, V_{fb} is the feedback signal voltage and d_0 is the nominal gap between the actuator and the proof mass.

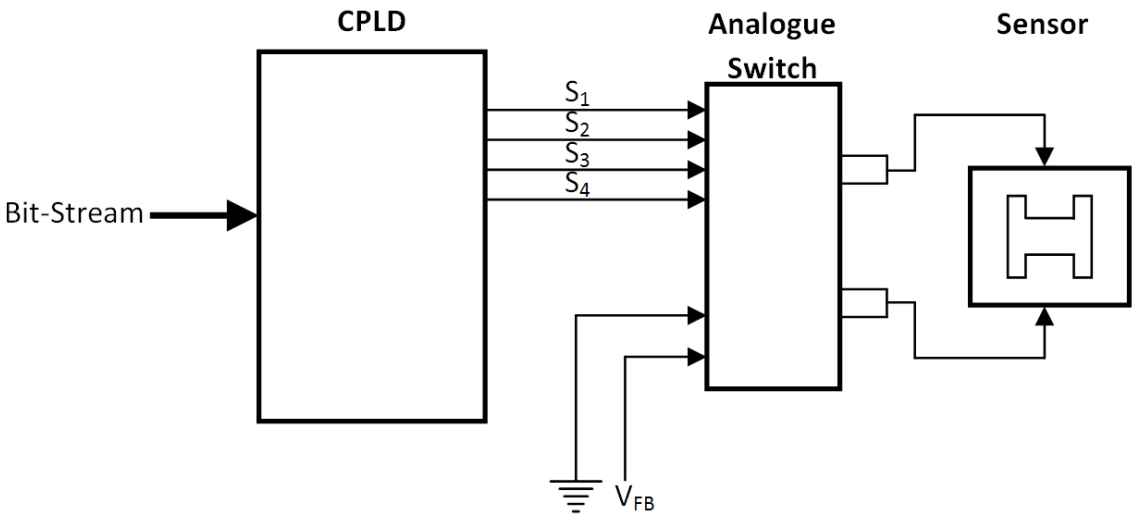


Figure 6-24: Feedback force DAC block diagram.

6.7 Second Order Electromechanical Sigma-Delta Modulator (First Loop)

In this section, simulation and real measurement results are presented for the SD2 (first loop). The simulation was performed in terms of higher level modelling using Simulink, as discussed in section 4.7.1, and electronic circuit modelling using PSpice, as shown in Figure 6-25.

The Spice model is comprised of the sensor model as a second order transfer function, which relates the input acceleration to the proof mass displacement. This displacement induces a capacitance change, which is detected by the pickoff circuit and converted to a proportional voltage signal. The lead compensator circuit is used to stabilize the control loop. The quantizer converts the analogue voltage to a digital signal in a form of a pulse density modulated bitstream. The electrostatic feedback force is applied to the proof mass to maintain it at its nominal position.

The models (Simulink and PSpice) were examined by applying a sine wave acceleration signal of 1.3g amplitude and 256Hz frequency. The real accelerometer circuit was also tested using a shaker table with the same signal amplitude and frequency. The result of the Simulink model is shown in Figure 6-26, the PSpice model in Figure 6-27 and the real hardware measurement is shown in Figure 6-28. The results are presented in the time and frequency domains in left and right sides, respectively. The time domain measurement show pulse modulation signals. For the real hardware in Figure 6-28 (top trace) shows pulse modulated output signal of the first loop and the recovered input signal after low pass filtering (bottom trace). The frequency domain measurement for all three systems shows similar noise shaping; both hardware measurement and Simulink result achieve a noise floor level of -90 dB (equivalent to 190 µg/√Hz), and the quantization noise is shaped and pushed to higher frequency. The hardware measurements closely agree with the simulated results.

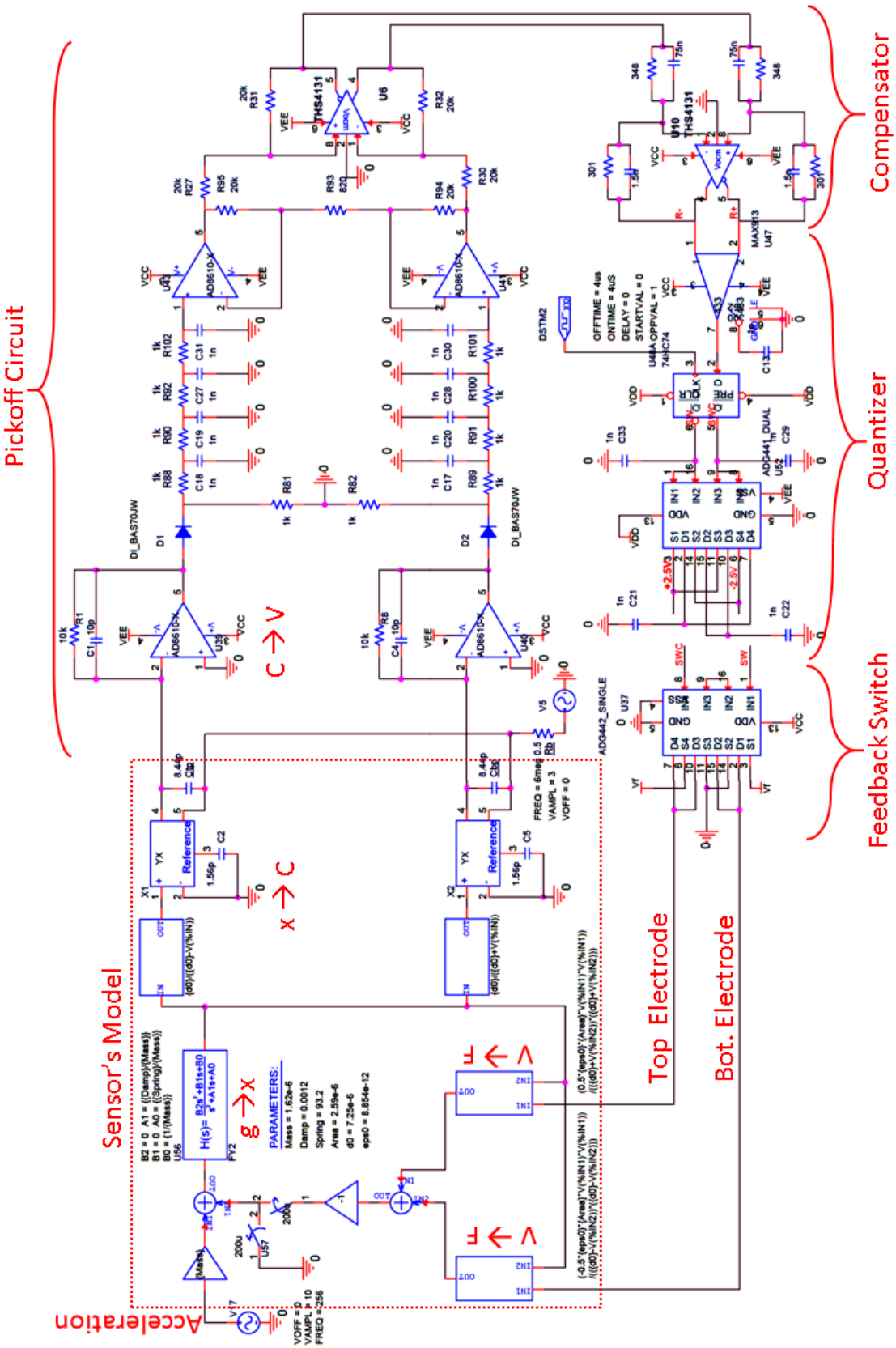


Figure 6-25: Spice model of the electromechanical SD2.

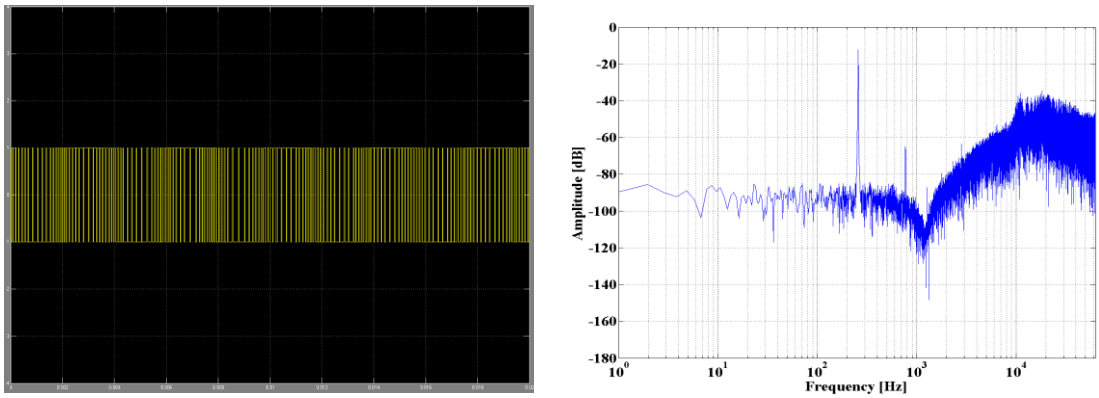


Figure 6-26: Simulink results of the SD2 (first loop), left is time domain and right is frequency domain results. FFT settings: sample rate = 131 kHz, number of samples = 128k, with Hanning window.

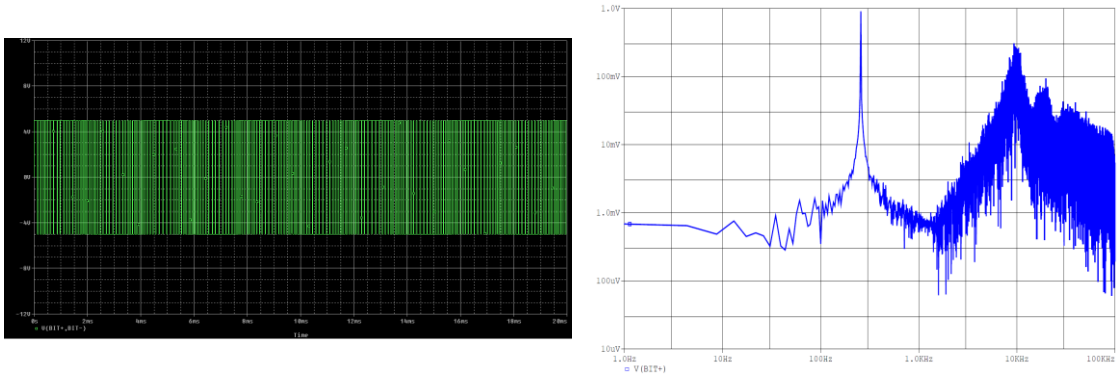


Figure 6-27: PSpice results of the SD2 (first loop), the left side shows the time domain and right side shows the frequency domain results.

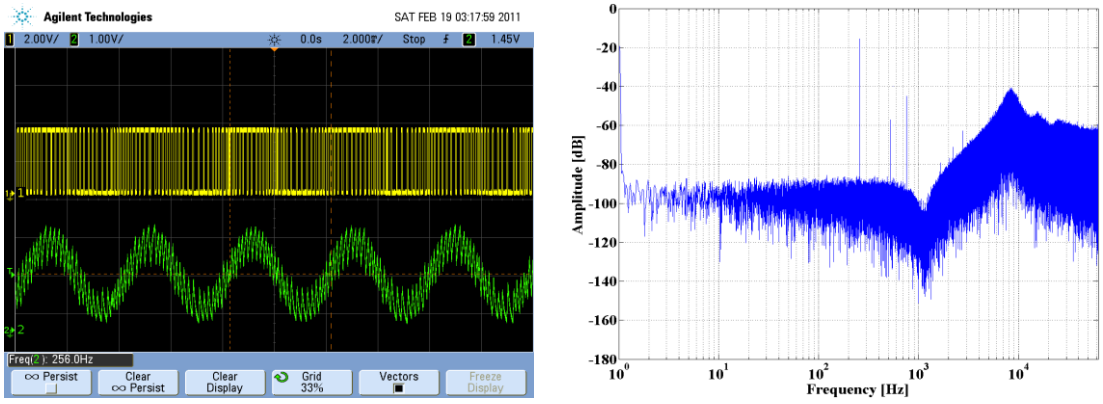


Figure 6-28: Hardware measurements of the SD2 (first loop), the left side shows the time domain and the right side shows the frequency domain results. FFT settings: sample rate = 125 kHz, number of samples = 512k, with Hanning window.

6.8 Second Order Electronic Sigma-Delta Modulator (Second Loop)

The main function of the second order electronic sigma-delta modulator (2nd loop) is to convert the analogue quantization error signal of the first loop into a digital bitstream output. The conversion is performed with the same sampling frequency as the first loop; i.e., 125 kHz. The second loop can also be attached to the first loop to form a fourth order single loop sigma-delta modulator (SD4).

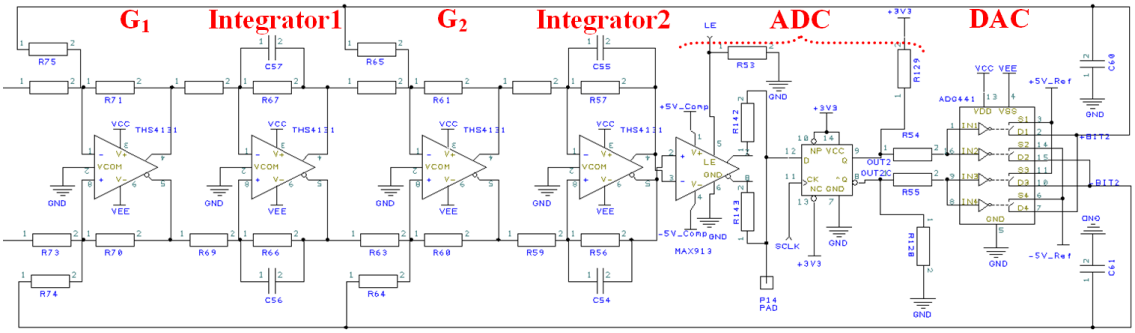


Figure 6-29: Second order electronic sigma-delta modulator circuit design.

The design of the second loop maintains the differential line structure of the first loop, as shown in Figure 6-29. The circuit was tested using a sine wave input signal of 0.8 V amplitude and 256 Hz frequency; the result is shown in Figure 6-30. The noise floor reaches -110 dB (equivalent to 22 µV/√Hz), which is -20 dB compared with the noise floor of the electromechanical 2nd order SDM. This difference is due to the high pickoff circuit electronic noise and the high DC gain of the electromechanical modulator that is mainly affected by the large spring constant (k = 93 N/m).

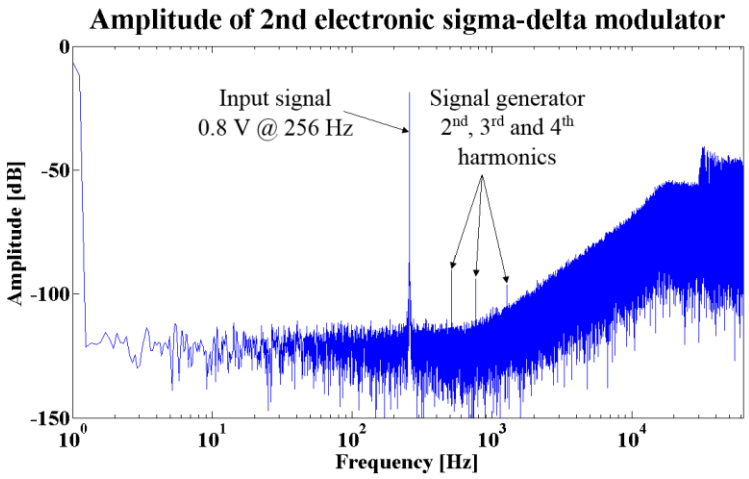


Figure 6-30: Hardware spectrum measurement of the 2nd order electronic sigma-delta modulator. FFT settings: sample rate = 125 kHz, number of samples = 512k, with Hanning window.

6.9 Digital Circuit

The digital circuit performs several tasks: it controls the switching of the feedback electrostatic force applied to the top and bottom electrodes, and it provides the ADCs with the sampling clock. It also receives bitstreams from ADCs and sends them to a personal computer (PC) for signal processing. The digital circuit consist of a CPLD, PIC, USB bridge, and on-board clock generator.

6.9.1 CPLD Unit

The CPLD IC is used to perform parallel tasks that vary between logic control and data transmission. It receives a 1 MHz clock signal generated by the on-board oscillator, and scales it down to provide the PIC and the D Flip-Flops with 125 KHz sampling clock. It also generates the switching signals to the feedback switch, in order to perform the feedback and quiet phases that are discussed in section 6.3.2, as shown in Figure 6-31.

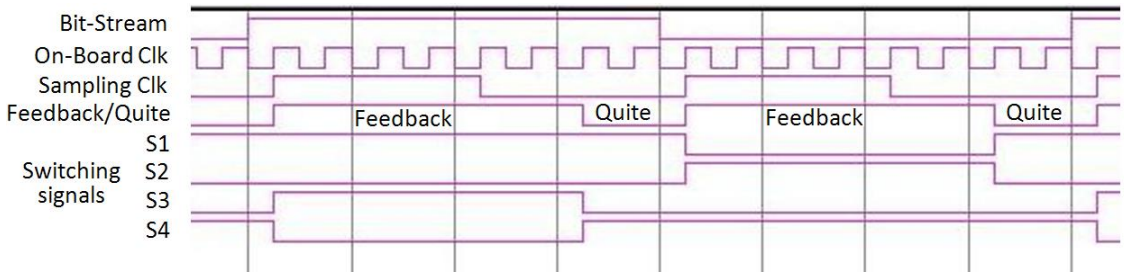


Figure 6-31: CPLD switching control signal time diagram.

The data handling performed by the CPLD starts with receipt of the two bitstreams 1 and 2 from the first and second loops, respectively, and sends them to the PIC along with the sampling clock. The CPLD also receives a controlling command from the PIC to switch between SD2 and SD4. The VHDL code (Very high speed integrated circuit Hardware Description Language) of the CPLD is listed in appendix B.1.

6.9.2 PIC and USB Bridge

The main function of the PIC is to receive the two bitstreams from the CPLD on every raising edge of the sampling clock, and then to multiplex them into one 8-bit frame and send the frame to a PC via the USB Bridge. It is very important to send the two bits, bit-1 from the first loop and bit-2 from the second loop, in the same time stamp, because the digital filtering in the PC strongly depends on the time matching between the two bitstreams. The PIC can also implement an overload detection mechanism for the higher order sigma-delta modulator. This can be done by observing the bit stream pulse duration: if the pulse duration exceeds a certain time, it will send the SD2/SD4 signal to the CPLD in order to switch to the SD2 architecture. The PIC can also receive a command from the PC for manual switching between SD2 and SD4 modulator. (Please refer to appendix C.1 for the full code)

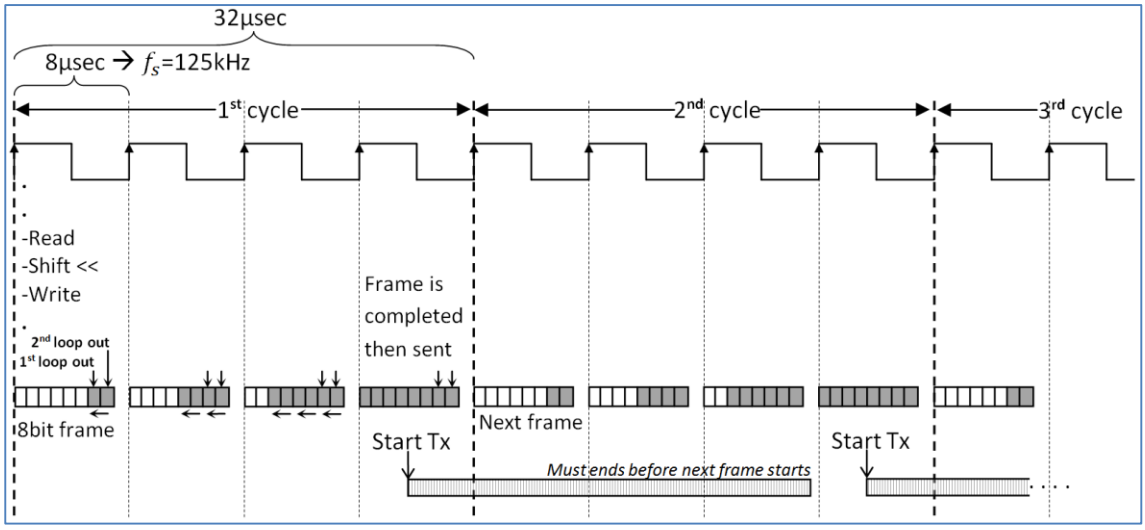


Figure 6-32: PIC timing diagram.

The sampling clock is used as an interrupt signal: at every raising edge of the sampling clock, an interrupt routine is called, which reads the two bits that are sent by the CPLD, and packs them into an 8-bit frame. At the fourth interrupt signal, the 8-bit frame is

completed and ready to be sent to the PC via the USB bridge. The PIC has to fulfil the data packing and transmission requirements without losing any data. The PIC requires about 30 instructions to read, shift, and pack the two bits before the next interrupt occurs. The transmission of the 8-bit frame also has to be fast enough to be completed before the next 8-bit frame is constructed. Figure 6-32 shows the timing constraints to aid in the choice of the proper PIC for this application. It indicates that, for a sampling frequency of 125 kHz, the processing time of 30 instructions must finish in less than 8 μsec, which requires a PIC running at minimum of 16 MHz. It also indicates that, in order for the data transmission to finish in less than 32 μsec, the PIC needs to have a universal asynchronous receiver/transmitter (UART) module capable of running at a minimum of 312.5 kbps. If the sampling frequency is doubled (i.e. OSR = 128), a PIC running at 32 MHz with UART module transmitting at minimum of 625 kbps is required.

6.10 MATLAB Coding and Digital Filter

To complete the MASH architecture, the two bit-streams have to be digitally filtered in the PC using MATLAB. The MATLAB code extracts the two bitstreams and applies the digital filters. The flow chart in Figure 6-33 summaries the code implementation, (please refer to appendix A.4 for the full code). The digital filter D_1 is a simple delay, which has the order of the following stage, i.e. $D_1=Z^{-2}$. The second filter D_2 is designed according to equation 4.12.

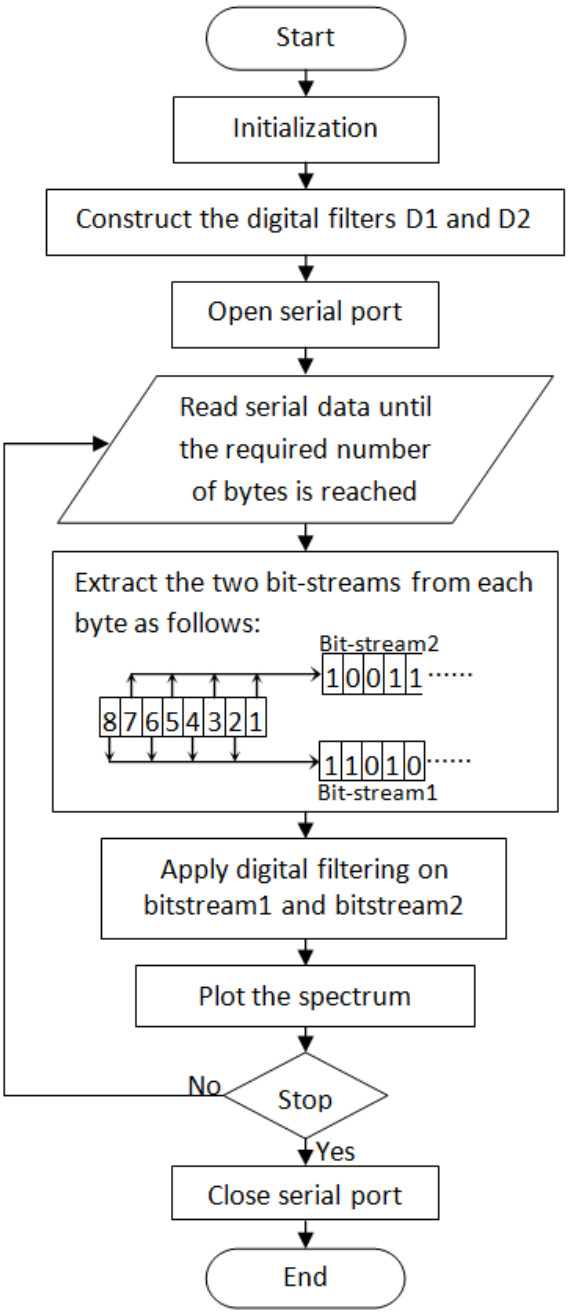


Figure 6-33: The MATLAB code flow chart.

6.11 MASH22 Experimental Results

The MASH22 accelerometer is implemented using the system parameters shown in Table 6-3. The two bitstreams from both loops were received by MATLAB and digitally filtered using D_1 and D_2. The dynamic measurement of the MASH22 was performed using a shaker table as shown in Figure 6-34.

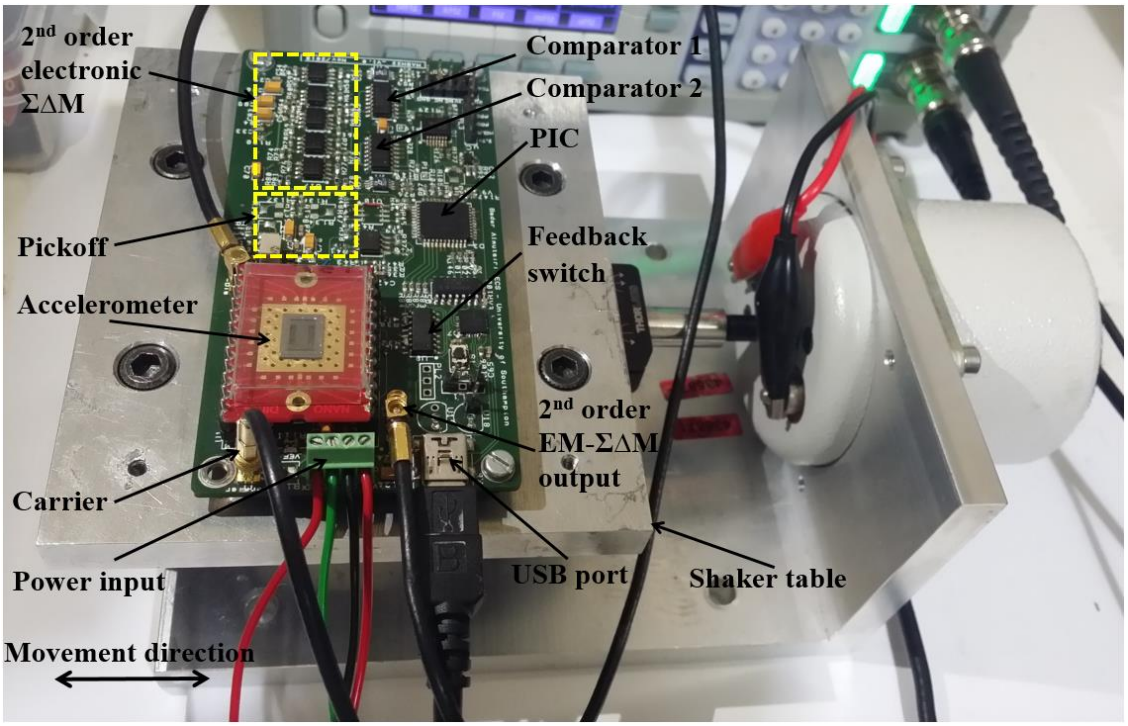


Figure 6-34: MASH22 system on the shaker table.

Figure 6-36 (a, b) show the noise shaping of the measurements with a noise floor of -90 dB (equivalent to 190 µg/√Hz) for the first stage SD2 (2^{nd} order ΣΔ accelerometer), and a noise floor around -110 dB (equivalent to 19 µg/√Hz) for the MASH22 structure. It can be seen that the measured noise floor of the first stage closely agrees with the simulated one shown in Figure 6-35 (a). However, the measured noise floor of the MASH22 structure differs from the simulated one in Figure 6-35 (b) by about 20 dB. Moreover, it shows a flat spectrum within the bandwidth of interest (1 kHz) due to other dominant noise sources, such as the pickoff circuit electronic noise and the environmental cross-coupling.

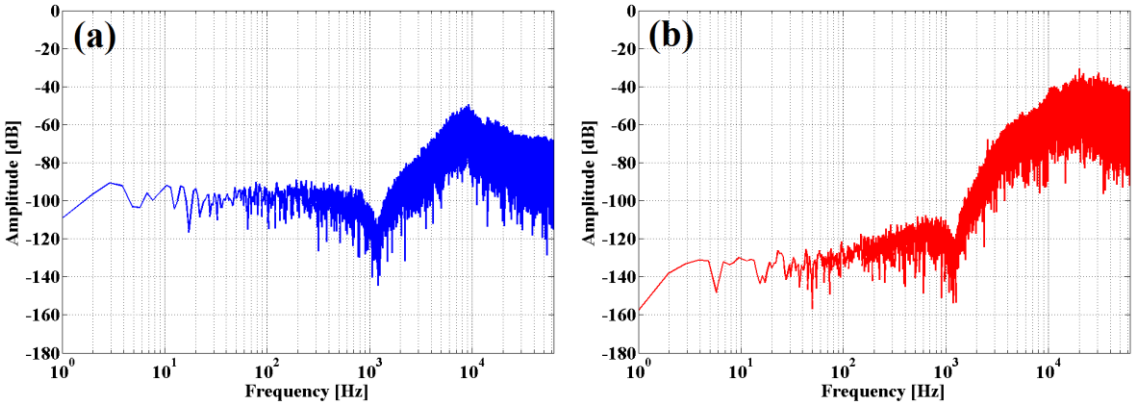


Figure 6-35: Simulation noise shaping analysis, (a) 2nd order sigma-delta accelerometer spectrum indicating a noise floor of -90 dB (190 µg/√Hz), (b) MASH22 accelerometer spectrum indicating a noise floor of -130 dB (1.9 µg/√Hz). FFT settings: sample rate = 131 kHz, number of samples = 512k, with Hanning window.

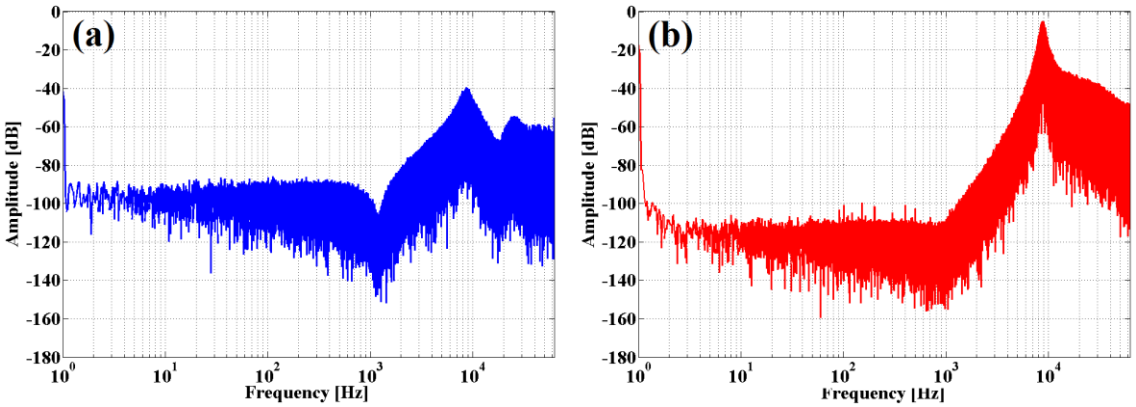


Figure 6-36: Experimental noise shaping results, (a) 2nd order sigma-delta accelerometer spectrum indicating a noise floor of -90 dB (190 µg/√Hz), (b) MASH22 accelerometer spectrum indicating a noise floor of -110 dB (19 µg/√Hz). FFT settings: sample rate = 125 kHz, number of samples = 512k, with Hanning window.

Parameter	**Value**	**Parameter**	**Value**
Mass 'm'	1.62×10^{-6} kg	**Pickoff gain 'K_{po}'**	2.1×10^{6}
Damper 'b'	1.2×10^{-3} N.s/m	**Boost gain 'K_{bst}'**	50
Spring 'k'	93.2 N/m	**Sampling freq. 'f_s'**	125×10^{3} Hz
Nominal gap'd_0'	7.25 µm	**Feedback voltage 'V_{fb}'**	12 V
Sense cap. 'C_s'	1.56 pF	**KR, KS, K2**	0.4, 1, 0.7
Feedback cap. 'C_f '	3.40 pF	**G1, G2**	1, 2

Table 6-3: MEMS accelerometer and system parameters.

6.12 Comparative Results of MASH22 and SD4

In this section, comparative experimental results are presented for two fourth order ΣΔMs: 1) a multi-stage (MASH22) architecture, and 2) a single-loop (SD4) architecture. Both architectures were investigated by system level modelling (in Simulink) in chapter 5. The comparison was performed with respect to stability, noise shaping and the parameter sensitivity due to fabrication tolerances. As discussed earlier, the circuit is configurable to accommodate both MASH22 and SD4 architectures, which allows them to be tested in the same noise environment.

6.12.1 Noise Shaping and Noise Floor Level

Dynamic measurements were taken for both systems using a shaker table set to a sinusoidal acceleration of 0.6 g at 550 Hz. The measured spectra of the MASH22 and SD4, using the same accelerometer, are shown in Figure 6-37 and Figure 6-38, respectively. Both systems achieved a noise floor level of -110 dB (equivalent to 19 µg/√Hz) within a bandwidth of 1 kHz.

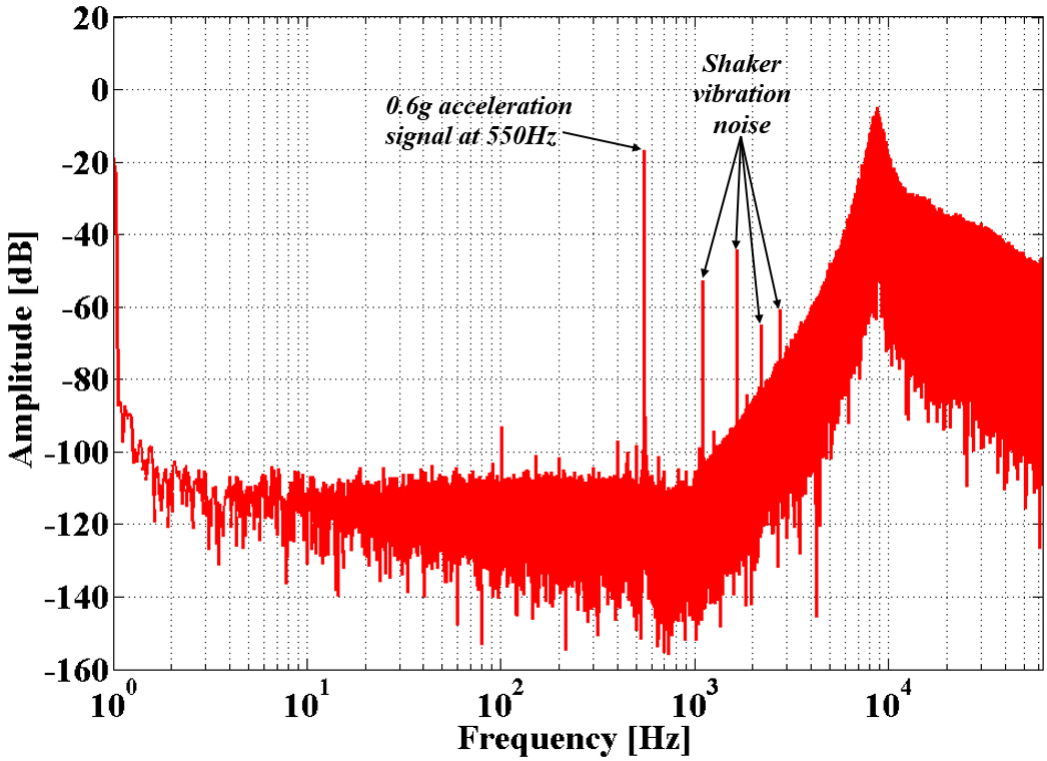


Figure 6-37: Noise shaping of the MASH22 with 0.6 g acceleration signal at 550 Hz, showing a noise floor level of -110 dB (19 µg/√Hz). FFT settings: sample rate = 131 kHz, number of samples = 512k, with Hanning window.

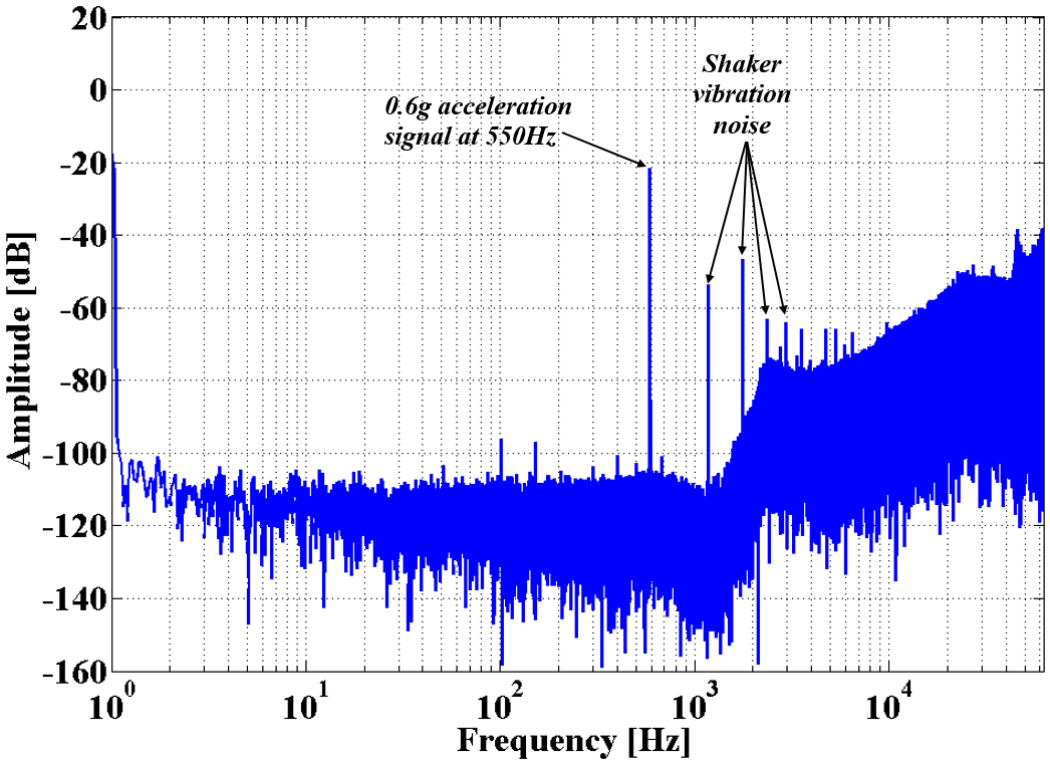


Figure 6-38: Noise shaping of the SD4 with 0.6 g acceleration at 550 Hz, showing a noise floor level of -110 dB (19 µg/√Hz). FFT settings: sample rate = 131 kHz, number of samples = 512k, with Hanning window.

6.12.2 Stability and Overload Input Level

The simulation analysis carried out in chapter 5 showed an overload acceleration input level (OLA) of 0.9 and 0.4 of the full scale input for MASH22 and SD4, respectively. Recalling equation 4.17, the overload input acceleration is given by:

$$OLA = OLF \times K_{fb}/(m \times 9.81) \qquad 6.36$$

Thus, the OLAs of MASH22 and SD4 are expected to be 1.6 g and 0.6 g, respectively. Both systems were excited with 1.5 g sinusoidal acceleration; the results are shown in Figure 6-39 and Figure 6-40 for MASH22 and SD4, respectively. The MASH22 confirms its ability to remain stable with the same noise floor of -110 dB (equivalent to 19 µg/√Hz). On the other hand, the SD4 system performance is dramatically degraded; the SD4 was not able to handle this acceleration level. The MASH22 has a higher dynamic range due to its high overload input level compared with the SD4.

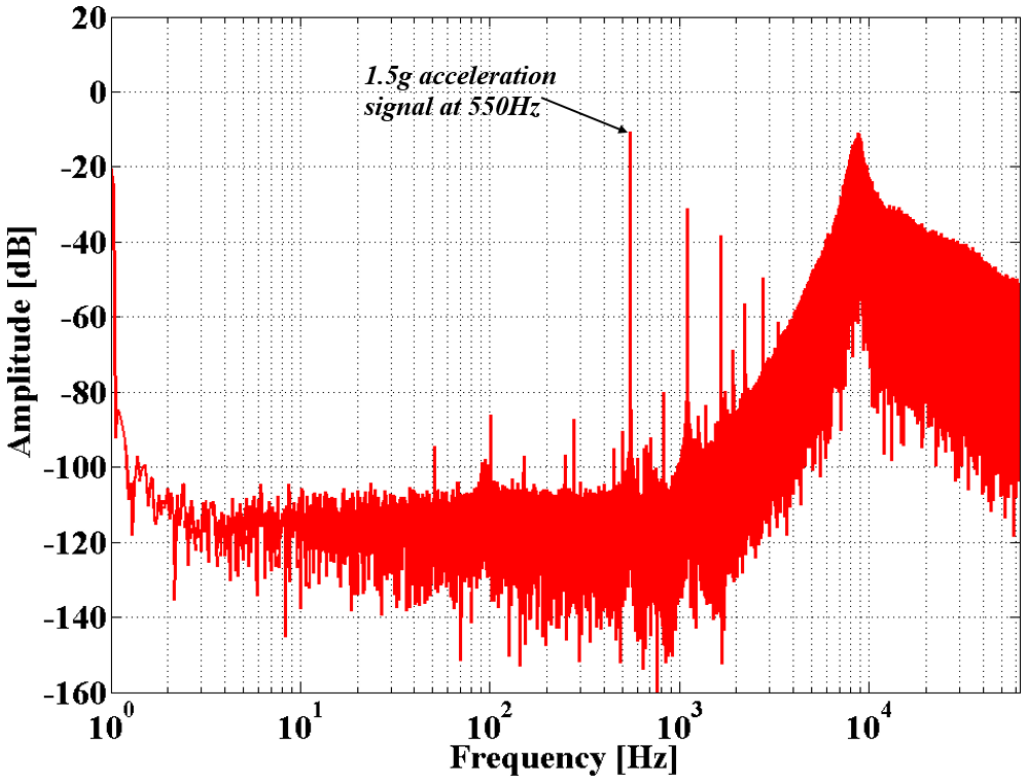


Figure 6-39: Noise shaping of the MASH22 with 1.5 g acceleration at 550 Hz. FFT settings: sample rate = 131 kHz, number of samples = 512k, with Hanning window.

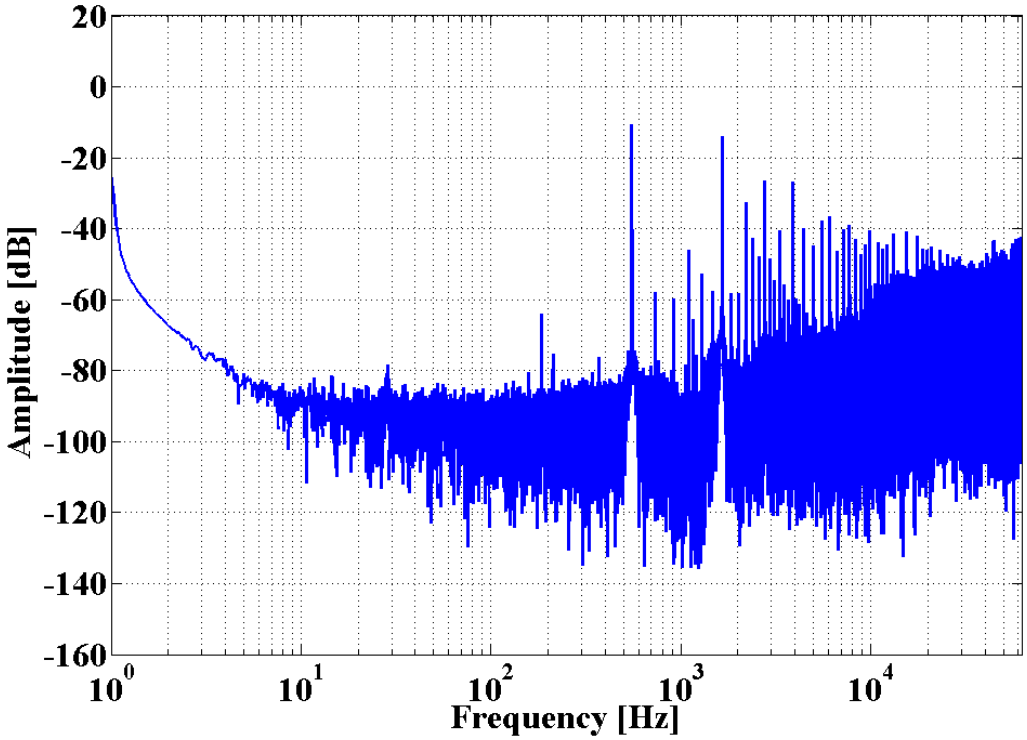


Figure 6-40: Noise shaping of the SD4 with a 1.5 g acceleration at 550 Hz, showing an unstable system. FFT settings: sample rate = 131 kHz, number of samples = 512k, with Hanning window.

6.12.3 Parameter Sensitivity

A parameter sensitivity analysis was performed for both architectures. For this analysis, with no acceleration applied, the sensing element is replaced with another sensing element that has a parameter variation of about 12% compared to the original one. The parameters of the new sensing element are measured using the procedures discussed in section 6.2. As expected the MASH22 exhibited performance degradation of about 10 dB, due to a leakage of the quantization noise to the input signal as shown in Figure 6-41. The quantization noise leakage from the first stage is a result of the mismatch between the digital filter D_2 and the new sensing element. The SD4 architecture confirms its immunity to the sensing element parameter variation as shown in Figure 6-42.

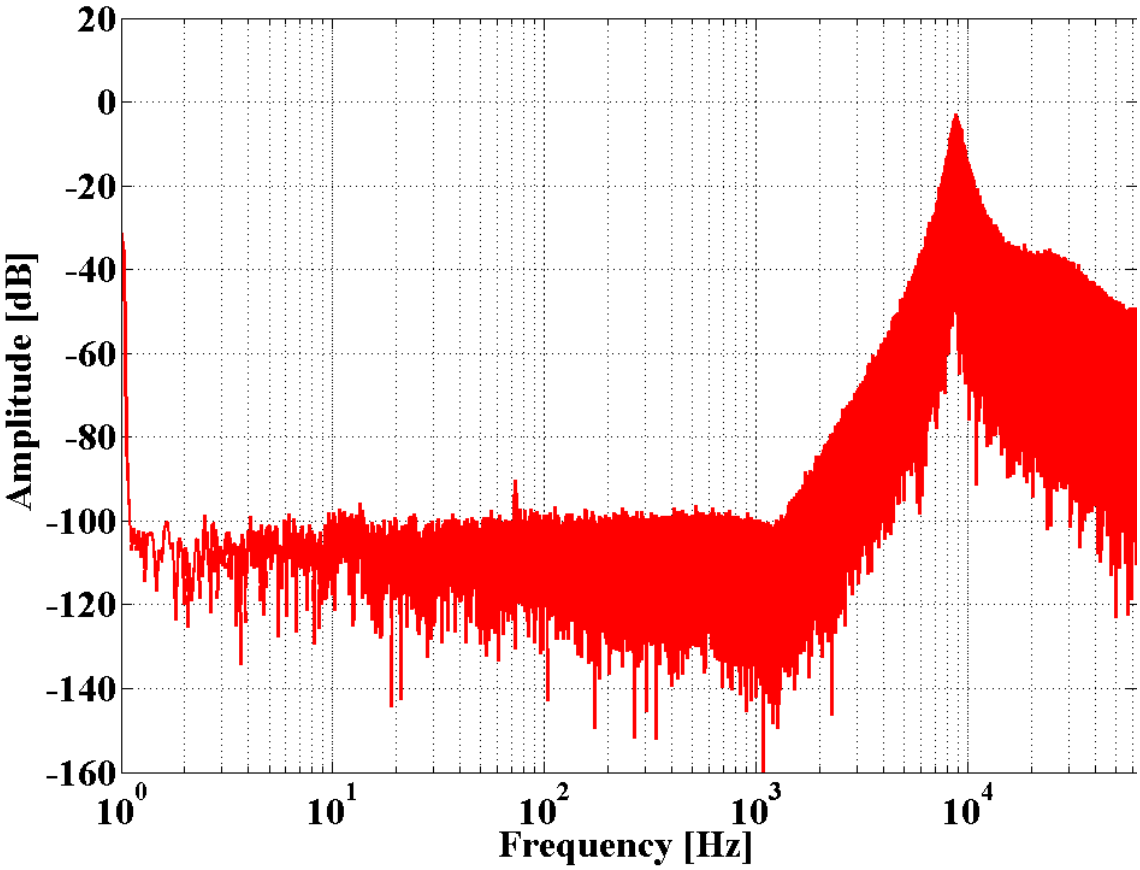


Figure 6-41: MASH22 noise floor increased to -100 dB (60 µg/√Hz) due to the change of the sensor's parameters. FFT settings: sample rate = 131 kHz, number of samples = 512k, with Hanning window.

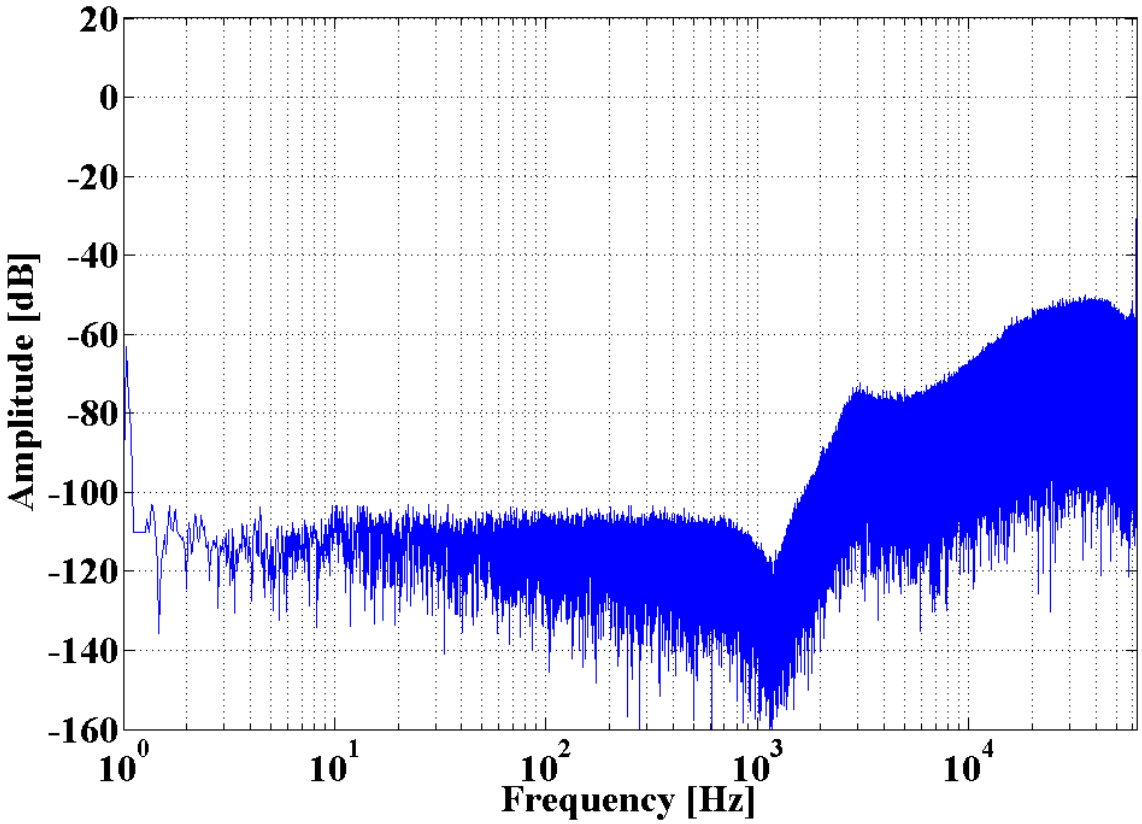


Figure 6-42: SD4 maintains the same noise floor of -110 dB (19 µg/√Hz) and confirms its immunity to the sensing element parameter variation. FFT settings: sample rate = 131 kHz, number of samples = 512k, with Hanning window.

6.12.4 Comparative Study Conclusion

An experimental comparative study was carried out between two 4th order ΣΔMs for a micromachined, capacitive accelerometer: 1) a multi-stage (MASH22) architecture; and 2) a single loop (SD4) architecture. Both architectures achieved a noise floor level of -110 dB (equivalent to 19 µg/√Hz) with a bandwidth of 1 kHz. The SD4 architecture confirmed its immunity to the sensing element parameter variation while the MASH22 performance was degraded due to the leakage of the quantization noise to the input signal. The MASH22 shows excellent stability properties and a high overload input threshold, high dynamic range and high noise shaping performance. Clearly, the MASH structure improves the performance of the modulator by 20 dB compared with the SD2 result, which has a noise floor of -90 dB (equivalent to 190 µg/√Hz). The experimental results proof that the EM-MASH concept is practical. Although, the MASH22 showed an improvement with its superior stability and high overload input level, it is sensitive for the sensor and electronic circuit parameter variation.

6.13 Summary

The implementation of the MASH22 accelerometer was discussed in this chapter. The circuit was designed to accommodate both MASH22 and SD4 modulators. It was also designed with a differential line scheme, which offers common mode rejection along the signal path. Since the successful implementation of the MASH architecture depends on the accurate matching between the digital filters and the system parameters, extra attention was paid to the accurate parameterization of the sensing element and pickoff circuit. The sensing element showed a difference between the theoretical parameters and the effective real ones; for instance, the spring constant dramatically changed when the sensor was operated in closed loop mode, where the electrostatic force applied to the proof mass introduced a negative electrostatic spring that added a considerable softening to the mechanical spring, hence, altering the dynamic response of the sensor. The effective value of the spring constant, with defined feedback and carrier signal voltages, was revised by considering a real measurement of the resonance frequency, as well as by deriving a matching equation to estimate the spring constant softening of the sensing element under these conditions. The pickoff circuit was discussed and analysed taking into consideration the parasitic capacitance that accompanies the sense capacitance, and the pickoff gain was determined experimentally.

The feedback signal cross-talk to the pickoff circuit was a major concern in the real implementation of the SD2 (first loop). This issue was solved by carefully utilizing the pickoff circuit in such a way that the charge integrator passes only high frequency signals (i.e., the carrier signal) and the low pass filter passes only the modulated proof mass movement signal.

The on-board digital unit services both MASH and SD4 architectures by supplying the sampling frequency, control signals and bitstream PC transmission. The digital filtering is carried out in real time in MATLAB, and the noise shaping of both architectures is obtained. The MASH real performance showed a close agreement to the theoretical analysis, and validated the concept of using MASH in MEMS accelerometers. The MASH showed a 20 dB enhancement compared with the SD2. Moreover, it shows a flat spectrum within the bandwidth of interest due to other dominant noise sources, such as the pick of circuit electronic noise.

An experimental comparative study was carried out between the MASH22 and a comparable SD4. The MASH22 and SD4 achieved similar performance in terms of noise floor. The MASH22 showed excellent stability properties and higher overload input threshold compared to SD4. However, it showed degradation of the performance due to its sensitivity to the parameter variations. On the other hand, the SD4 architecture maintained its performance and showed immunity to sensing element variations. However, a calibration method will be discussed in the next chapter to overcome the MASH quantization leakage problem due to the parameter variation.

Chapter 7: **Design and Implementation of a MASH20 Electromechanical Sigma-Delta Modulator for Capacitive MEMS Sensors Using Dual Quantization Method**

7.1 Introduction

In the preceding chapters, the performance of the EM-MASH22 was investigated using theoretical and experimental approaches. To achieve a performance higher than the fourth order MASH22, several ΣΔM stages must be cascaded. The complexity of the required digital filters increases as the number of stages and/or the order of the individual stages increases. This makes digital filter matching difficult to achieve.

The dual quantization technique represents another approach to the design of a ΣΔM [31-33]. Single-loop electromechanical ΣΔMs that use this technique have been reported in the literature [81-86]. However, a MASH version of this technique has not been explored for use in electromechanical ΣΔMs despite its potential advantages compared to the single loop approach.

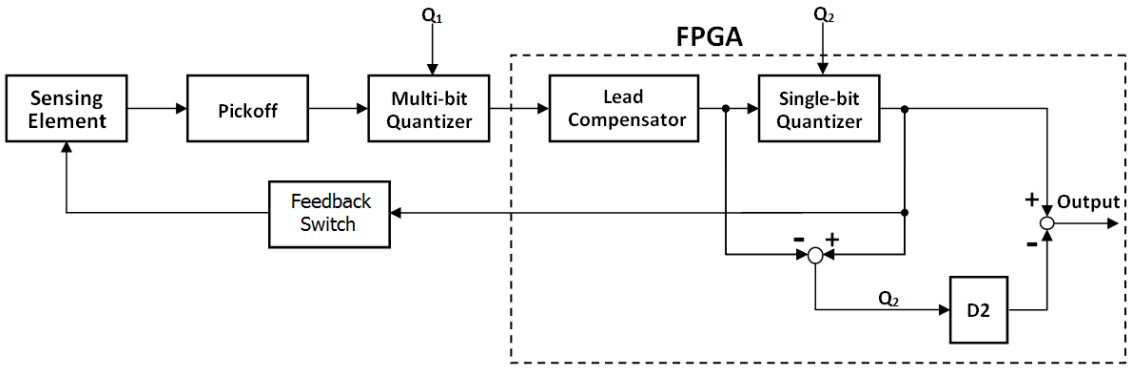


Figure 7-1: Block diagram of the electromechanical MASH20 ΣΔM.

This chapter presents a novel EM-MASH that employs the dual quantization technique and adopts an electromechanical 2-0 multi-stage noise-shaping structure (EM-MASH20). As shown in Figure 7-1, the modulator consists of a second order EM-ΣΔM accelerometer interfaced to a multi-bit quantizer, which is then followed by a digital compensator and a single-bit quantizer controlling an electrostatic feedback force to close the loop. The

structure of the MASH20 has the advantage of reduced noise induced by the multi-bit quantizer, and instead of using multibit feedback that leads to the analogue closed loop problems discussed in section 2.4.2.1, the structure maintains the advantages of the single-bit digital closed loop. Both of the quantization noise sources (Q1 and Q2) generated by the two quantizers are shaped by the sensing-element loop filter. However, as the single-bit quantization noise is in digital form and is considerably greater than the multi-bit quantization noise, it can be directly cancelled by a digital filter without the need for a second-stage ΣΔM as in the MASH22. This reduces the complexity of the digital filter compared to those required for the MASH22; thus, digital filter matching is easier to achieve. This eases the use of the digital signal processing capabilities to compensate for the filter mismatch. One method of overcoming the leakage problem is to use optimization algorithms such as the genetic algorithm (GA) or the simulated annealing algorithm (SA), as will be discussed in section 7.7.

The system modelling and noise analysis will be discussed, followed by the simulation results. The design and fabrication process of a high-g capacitive MEMS accelerometer will be addressed. Next, the hardware implementation and the experimental results of the MASH20 will be presented. Finally, a digital calibration of the filter mismatching will be presented.

7.2 System Modelling and Noise Analysis

The system level view of the MASH20 control system is illustrated by the Simulink model in Figure 7-2. The MASH20 consists of a capacitive MEMS accelerometer (M) embedded in a digitally controlled force-feedback loop that forms an SD2.

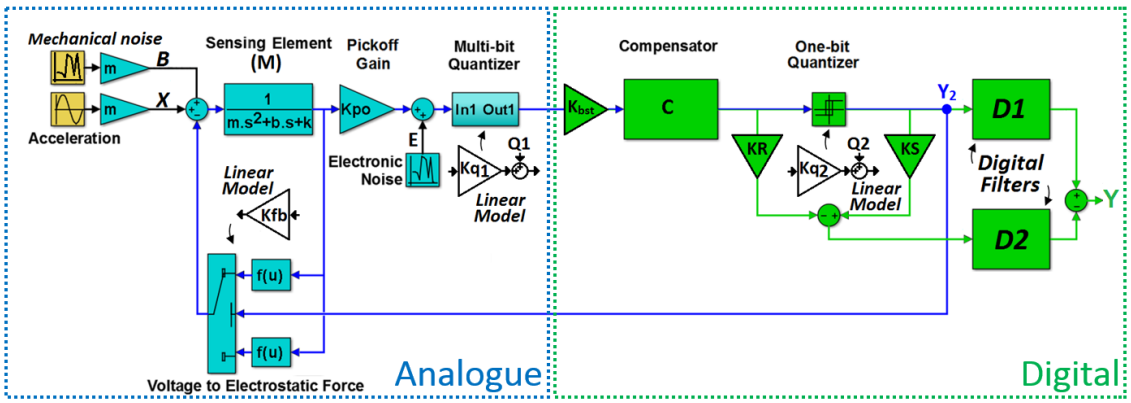


Figure 7-2 Simulink model of the EM-MASH20 with the linear model components.

The differential sense capacitance change due to the acceleration input is converted by the pickoff circuit into a proportional voltage, which is then digitized by a multi-bit quantizer. A digital compensator (C) is used to maintain loop stability. This design retains the intrinsic linearity of single-bit feedback by using a single-bit quantizer to generate feedback pulses in order to maintain the proof mass in its nominal position. To achieve a performance higher than an SD2, the single-bit quantization noise must be cancelled by the digital filters.

The multi-bit quantization noise (Q1) is considerably smaller than the single-bit quantization noise (Q2) (to be discussed in more detail later in this section), making Q2 the dominant noise of the system. Because Q2 is in a digital form, it can be cancelled directly by digital filters without the need for a second-stage ΣΔM, as in a MASH22.

The system was linearized to design the digital filters (D_1 and D_2) and to inspect the effects of the different noise sources. As discussed in section 4.2, assuming the input signal is sufficiently busy , the two quantizers were modelled as gain constants (K_{q1} and K_{q2}) and additive white noise signals (Q1, Q2). The pickoff and the feedback circuits can also be modelled as simple gain constants (K_{po}, K_{bst}, and K_{fb}) for small mass deflection. The gain constants (K_R, K_S) were used to scale Q2 for cancellation by digital filters.

The output signal Y_2 of the SD2 can be derived from the linear model using the superposition principle, taking the following form:

$$Y_2 = S\mathrm{TF}_{SD2}\,(X + B) + Q1NTF_{SD2}\,Q1 + \ Q2NTF_{SD2}\,Q2 + ENTF_{SD2}\,E \qquad 7.1$$

The overall system performance is now determined by four noise sources: Mechanical noise force (B) due to the Brownian motion of the proof mass, electronic noise (E) due to the thermal and the flicker noise generated by the pickoff circuit, and the quantization noises Q1 and Q2.

If X is the input force due to the acceleration, the signal transfer function (STF_{SD2}), the multi-bit quantization noise transfer function ($\mathrm{Q1NTF}_{SD2}$), the single-bit quantization noise transfer function ($\mathrm{Q2NTF}_{SD2}$), and the pickoff circuit electronic noise transfer function (ENTF_{SD2}) can be derived as follows:

$$STF_{SD2} = \frac{Y_2}{X} = \frac{MK_{po}K_{bst}CK_{q1}K_{q2}}{1 + MK_{po}K_{bst}CK_{fb}K_{q1}K_{q2}} \quad 7.2$$

$$Q1NTF_{SD2} = \frac{Y_2}{Q_1} = \frac{K_{bst}CK_{q2}}{1 + MK_{po}K_{bst}CK_{fb}K_{q1}K_{q2}} \quad 7.3$$

$$Q2NTF_{SD2} = \frac{Y_2}{Q_2} = \frac{1}{1 + MK_{po}K_{bst}CK_{fb}K_{q1}K_{q2}} \quad 7.4$$

$$ENTF_{SD2} = \frac{Y_2}{E} = \frac{K_{bst}CK_{q1}K_{q2}}{1 + MK_{po}K_{bst}CK_{fb}K_{q1}K_{q2}} \quad 7.5$$

To calculate the dominant quantization noise in the system, the spectral density of the modulation noise for the multi-bit quantizer $N1_{SD2}(f)$ can be found by multiplying the quantization noise spectral density Q1(f) with the modulus of the $Q1NTF_{SD2}$ as follows [56]:

$$N1_{SD2}(f) = Q1(f) \ |Q1NTF_{SD2}| \quad 7.6$$

For a busy input signal, the quantization noise spectral density Q1(f) is of the following form [56]:

$$Q1(f) = Q1_{rms}\sqrt{2T_s} \quad 7.7$$

where T_s is the sampling time, and $Q1_{rms}$ is the RMS value of the quantization noise in the following form:

$$Q1_{rms} = \frac{\Delta}{\sqrt{12}} \quad 7.8$$

where Δ is the level spacing of the quantizer. Assuming a uniform quantizer, the level spacing Δ is:

$$\Delta = \frac{V_{fs}}{2^m - 1} \quad 7.9$$

where V_{fs} is the full scale input voltage, and m is the number of quantizer bits.

By rearranging the above equations, the spectral density of the multi-bit modulation noise $N1_{SD2}(f)$ at the output Y_2 takes the following form:

$$N1_{SD2}(f) = \sqrt{\frac{T_s}{6}}\left(\frac{V_{fs}}{2^m - 1}\right) |Q1NTF_{SD2}| \qquad 7.10$$

Similarly, the spectral density of the modulation noise for the single-bit quantizer N2(f) can be calculated as:

$$N2_{SD2}(f) = \sqrt{\frac{T_s}{6}}(V_{fs}) |Q2NTF_{SD2}| \qquad 7.11$$

The electronic noise generated by the pickoff circuit is mainly thermal white noise. The flicker noise is largely suppressed as the pickoff circuit employs a high frequency carrier signal (6 MHz) and a high pass filter, as will be discussed in section 7.5.1. The electronic noise spectral density was measured and found to be E = 15 μV/√Hz as shown in Figure 7-19. Therefore, the electronic noise spectral density at the output of the modulator is given by:

$$E_{SD2}(f) = E\ |ENTF_{SD2}| \qquad 7.12$$

The mechanical noise spectral density is given by:

$$B_{SD2}(f) = B\ |STF_{SD2}| \qquad 7.13$$

The mechanical noise force (B) is given by:

$$B = \sqrt{4K_B Tb} \qquad 7.14$$

where K_B is the Boltzmann constant, T is the Kelvin temperature, b is the damping coefficient.

Using the above equations with the system parameters listed in Table 7-1 and assuming a quantizer full scale input voltage of 5 V, and a 12-bit quantizer, the spectral density of the noise sources at the SD2 output are shown in Figure 7-3.

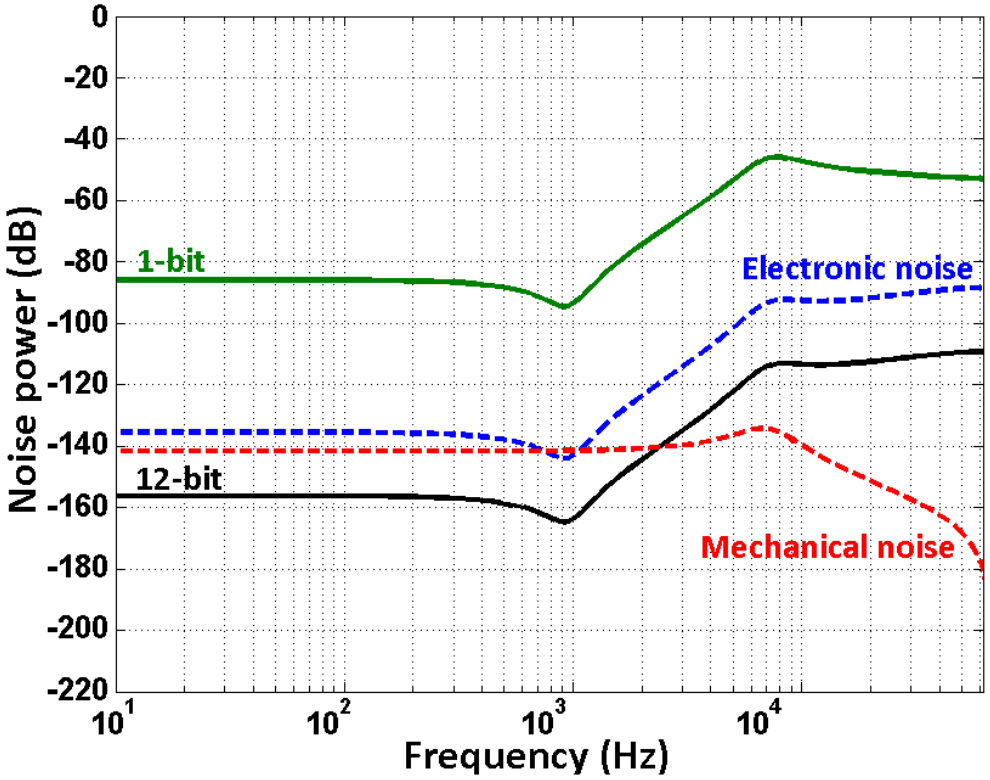


Figure 7-3: Spectral density of the SD2 at output Y_2 showing: the modulation noise of the 12-bit quantizer, the 1-bit quantizer, electronic noise and mechanical noise.

It can be seen that the quantization noise of Q2 in the digital domain is the performance limiter of the SD2. Thus, it will be the focus of the MASH20 design to cancel it using the digital filters D_1 and D_2.

The output signal Y of the MASH20 can be derived from the linear model by using the superposition principle as follows:

$$Y = \mathrm{STF}_{MASH20}\,(X + B) + Q1NTF_{MASH20}\,Q1 + Q2NTF_{MASH20}\,Q2 + ENTF_{MASH20}\,E \qquad 7.15$$

where STF_{MASH20}, $\mathrm{Q1NTF}_{MASH20}$, $\mathrm{Q2NTF}_{MASH20}$, and ETF_{MASH20} are given by:

$$STF_{MASH20} = \frac{Y}{X} = STF_{SD2}\left[D_1 - D_2\left(K_S - \frac{K_R}{K_{q2}}\right)\right] \qquad 7.16$$

$$Q1NTF_{MASH20} = \frac{Y}{Q1} = Q1NTF_{SD2}\left[D_1 - D_2\left(K_S - \frac{K_R}{K_{q2}}\right)\right] \qquad 7.17$$

$$Q2NTF_{MASH20} = \frac{Y}{Q2} = \left[D_1\ Q2NTF_{SD2} - D_2\ \left(K_S - \frac{K_R}{K_{q2}}\right) Q2NTF_{SD2} - D_2\ \left(\frac{K_R}{K_{q2}}\right)\right] \quad 7.18$$

$$ENTF_{MASH20} = \frac{Y}{E} = ENTF_{SD2}\left[D_1 - D_2\ \left(K_S - \frac{K_R}{K_{q2}}\right)\right] \quad 7.19$$

In order to cancel the quantization noise of Q2, $Q2NTF_{MASH20}$ must equal zero. To achieve this, after some mathematical manipulation on (7.18), D_2 takes the following form:

$$D_2 = \frac{D_1\ Q2NTF_{SD2}}{\left(K_S - \frac{K_R}{K_{q2}}\right) Q2NTF_{SD2} + \left(\frac{K_R}{K_{q2}}\right)} \quad 7.20$$

Note that if D_1, K_S, K_R, and K_{q2} equal one (i.e., they are unity gain constants), D_2 and $Q1NTF_{MASH20}$ can be simplified to:

$$D_2 = Q2NTF_{SD2} \quad 7.21$$

$$Q1NTF_{MASH20} = Q1NTF_{SD2} \quad 7.22$$

This simplifies the implementation of the digital filter D_2.

If Q2 is completely cancelled, Q1, E and B noises will appear at the modulator output Y. The spectral density of the modulation noise for the multi-bit quantizer $N1_{MASH20}(f)$ at the output of the MASH20 takes the following form:

$$N1_{MASH20}(f) = \sqrt{\frac{T_s}{6}}\left(\frac{V_{fs}}{2^m - 1}\right) |Q1NTF_{MASH20}| \quad 7.23$$

Using the above equation and assuming a quantizer full scale input of 5V, the spectral densities of $N1_{MASH20}(f)$ for various multi-bit quantizeres (ranging from 4 to 16-bit) along with the electronic and mechanical noise are shown in Figure 7-4. The figure shows that, the MASH20 performance can be evaluated as determined by the resolution of the multi-bit quantizer, which agrees with (7.17) and (7.22). It can also be observed that a high performance EM-ΣΔM can be realized with the dual quantization technique using a multi-bit quantizer of eight bits or more, such that the electronic and the mechanical noise

are the performance limiter of the system. In this study, a 12-bit quantizer was chosen with a sampling rate of 131 kHz.

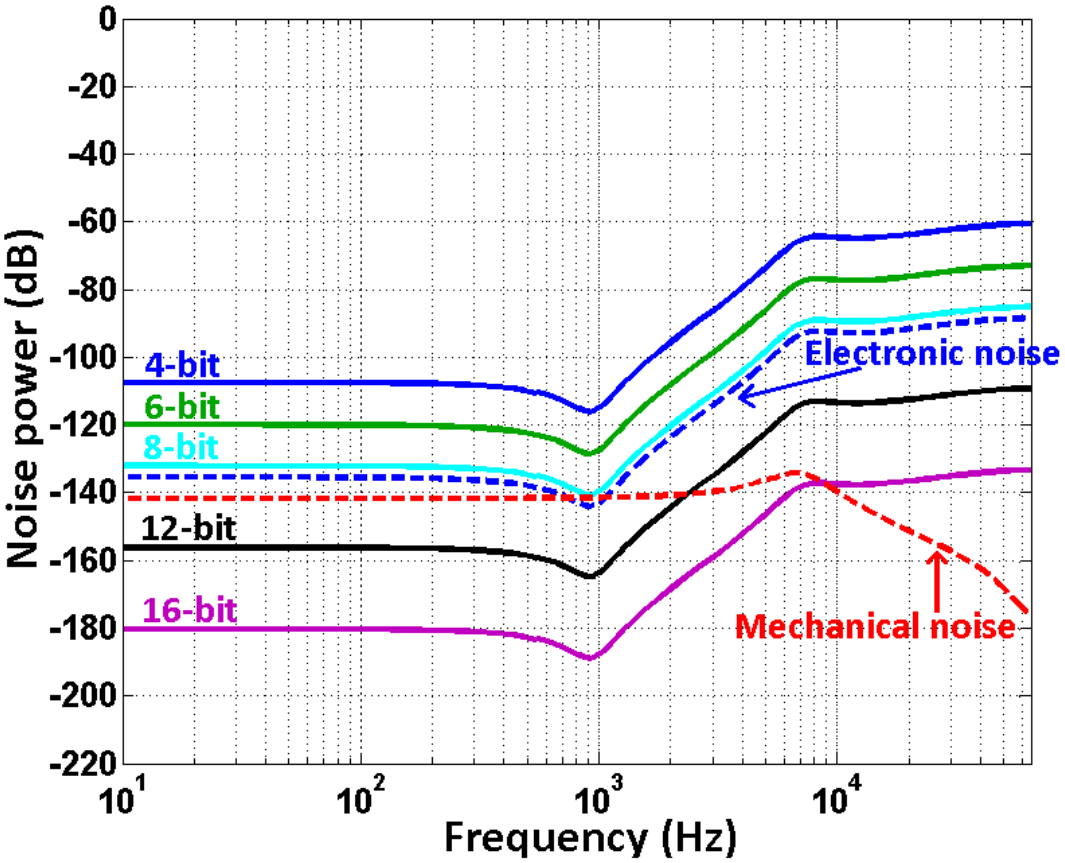


Figure 7-4: Spectral density of the multi-bit modulation noise (ranging from 4- to 16-bit), the electronic and mechanical noise sources observed at the output Y.

7.3 System Modelling and Simulation Results

The lead compensator and the digital filter D_2 are presented as infinite impulse response (IIR) filters within the Simulink model in Figure 7-2. In order to implement the EM-MASH20 using an FPGA device, the lead compensator C and the digital filter D_2 need to be represented in signal flow diagram using the direct form realization method [131] which uses minimum number of delay units and multipliers, as shown in Figure 7-5. The floating point data type is set by default in Simulink model, however, in order to create a VHDL code, the model has to be converted to fixed point data. The fixed point tool box and the HDL (hardware description language) coder offered by MATLAB [132] were utilized to generate a code suitable for FPGA implementation.

The ModelSim® HDL simulator is a tool offered by Mentor Graphics [133] and can be integrated within the Simulink model. After obtaining the VHDL code of the digital components, the Simulink model in Figure 7-6 was built with the VHDL code embedded in the ModelSim HDL simulator. This step was helpful to verify the performance of the VHDL code with the nonlinear parts of the system.

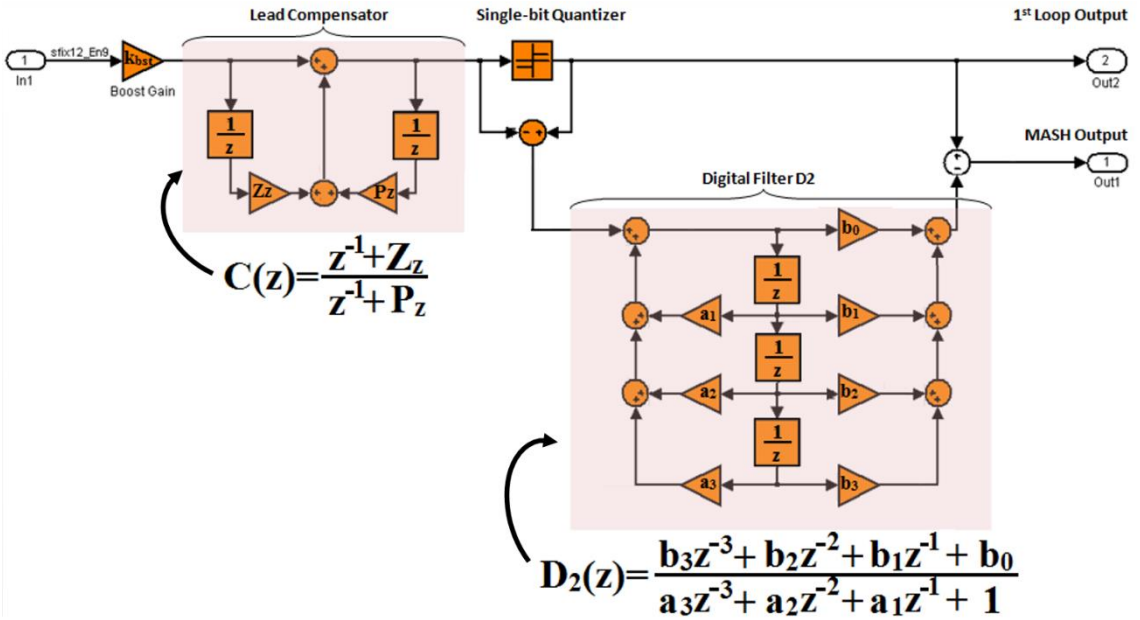


Figure 7-5: Digital realization of the lead compensator and the digital filter D2 with the direct form method.

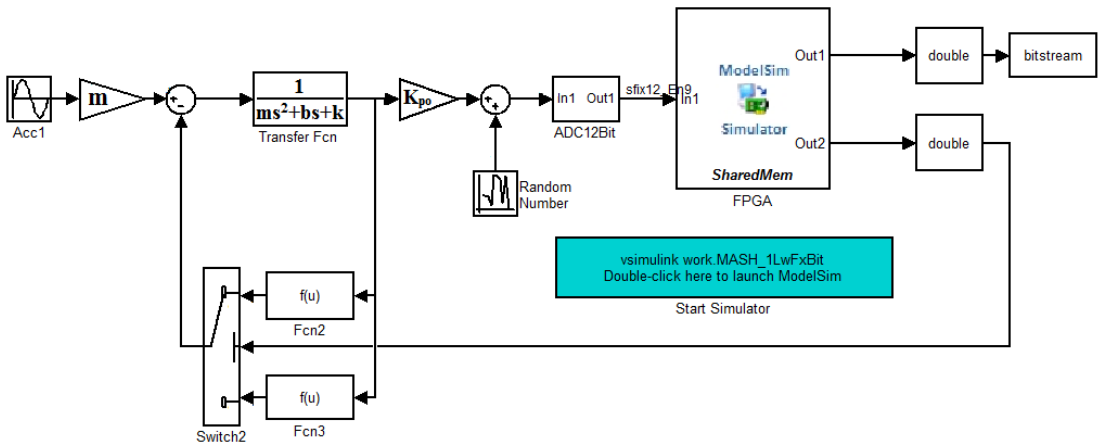


Figure 7-6: ModelSim HDL simulator integrated within the Simulink environment to verify the HDL code.

Figure 7-7 shows the output signals of the MASH20 structure generated by the ModelSim HDL simulator (top) and the Simulink model (middle). The difference between the two

signals (bottom) is zero. This result confirms that the VHDL code is an exact replica of the MASH20 digital part in Simulink, and can be embedded within an FPGA device.

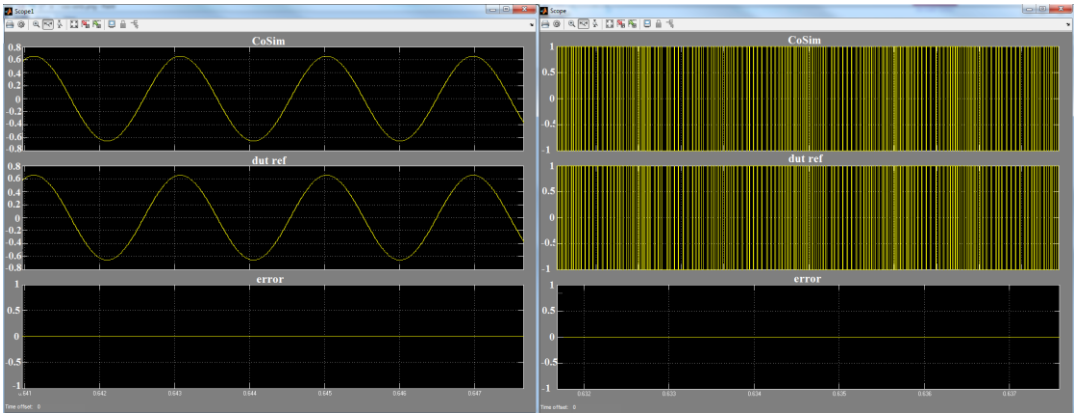


Figure 7-7: (Left) the output of the MASH structure and (Right) the output of the 1st loop. The Simulink output (top) and the ModelSim output (middle) are compared and show a zero error (bottom).

7.3.1 Noise Shaping

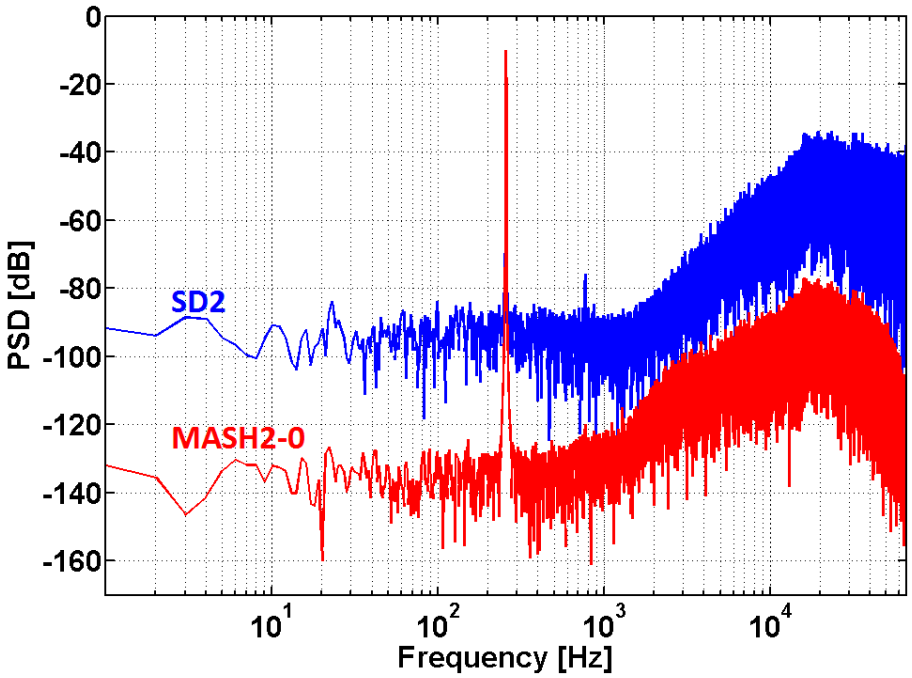


Figure 7-8: Noise shaping of the SD2 (blue) and the MASH20 (red). The noise floor of the SD2 is about -90 dB (equivalent to 1.5 mg/√Hz), while the MASH20 is about -130 dB. FFT settings: sample rate = 131 kHz, number of samples = 512k, with Hanning window.

The system was simulated with a realistic values of the mentioned noise sources using Simulink. The noise shaping of the SD2 at output Y_2 is shown in Figure 7-8 (blue line). The quantization noise Q2, which dominates over other noise sources in the system, is shaped by the SD2 with a noise floor of around -90 dB (equivalent to 1.5 mg/√Hz) within a bandwidth of 1 kHz. With regard to the noise shaping of the MASH20 at the output Y the noise floor is lowered by 40 dB and achieved -130 dB (equivalent to 15 µg/√Hz) within a bandwidth of 1 kHz. As expected, the digital filter cancelled out Q2. However, as shown in Figure 7-4, the theoretical limit of the 12-bit quantization noise Q2 is -155 dB, which indicates that the electronic noise E is now the dominant noise of the MASH20.

7.3.2 Overload Acceleration Input and Dynamic Range

As discussed in section 4.4, the overload acceleration input (OLA) is an important criterion in an EM-ΣΔM which affects the stability, the dynamic range and the SNR. Using the same sensing element parameters and a feedback voltage of 12 V, a simulation analysis was carried out to compare the performance of the MASH20 with a SD4 and an MASH22 in terms of the OLA, the SNR and the dynamic range. The result is shown in Figure 7-9; it can be seen that MASH20 shares the advantage of MASH22 of having high OLA of 21 g and OLF of 0.85 compared with SD4 which only achieved an OLA of 12 g and an OLF of 0.45. However, MASH20 achieved higher SNR of 115 dB and higher dynamic range of 118 dB than MASH22 and SD4.

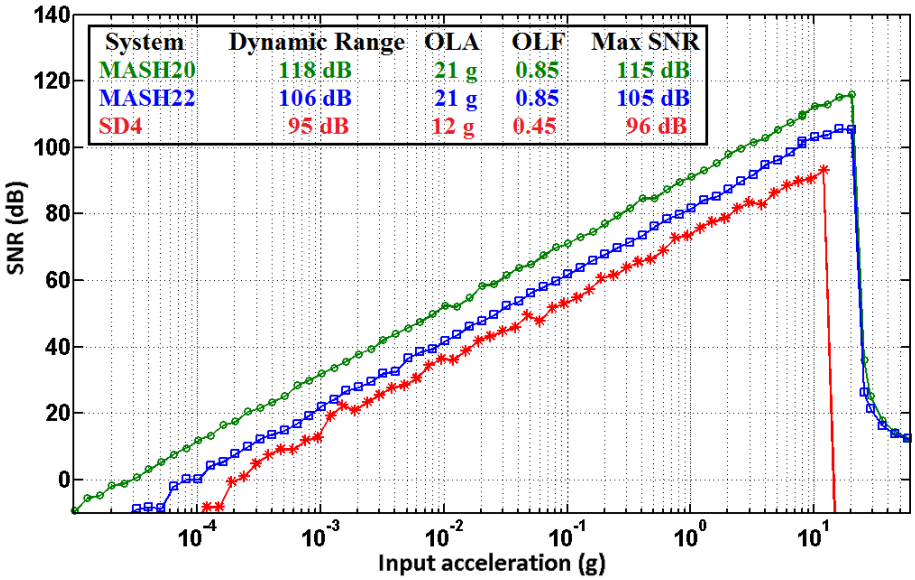


Figure 7-9: Input acceleration (g) vs. SNR of the MASH20, MASH22 and SD4. MASH20 shows higher SNR and dynamic range compared with MASH22 and SD4.

7.3.3 Parameter Sensitivity

Figure 7-10 (top) shows the Monte Carlo simulation for MASH22 (left) and MASH20 (right). The system parameters for each architecture were varied by 20% of their nominal values using a continuous uniform distribution function for 300 iterations. The Monte Carlo simulation of MASH20 involved variations of 5 parameters (m, b, k, K_{po} and K_{fb}). On the other hand, the Monte Carlo simulation of MASH22 involved variations of 14 system parameters; on top of the aforementioned MASH20 parameters, MASH22 takes the compensator pole and zero, the scaling gain constants (KR, KS and K2) and the second stage parameters (the two integrators and their associated gain constants) into account. In general, both MASH22 and MASH20 are sensitive to parameter variation; the digital filter failed to completely filter out the quantization noise of the first loop, and in the worst case, the overall performance (output SNR) reaches SD2 level performance. However, MASH20 performance falls from higher SNR levels than the MASH22. For example, it can be seen that at 18 g input acceleration, the output performance of MASH20 (Figure 7-10, top-right) falls from 112 dB down to 67 dB (44% degrade) and the performance of MASH22 (Figure 7-10, top-left) falls from 100 dB down to 63 dB (37% degrade). Nevertheless, MASH20 benefits from the reduced number of system parameters dependency and ease of digital filter implementation.

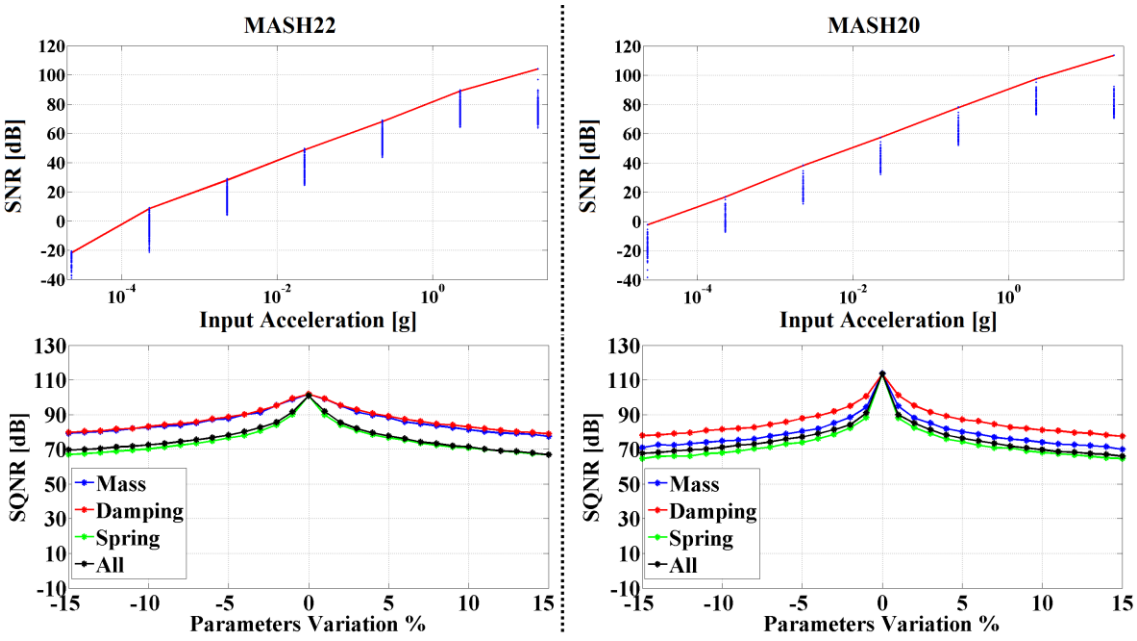


Figure 7-10. MASH20 sensing element parameter sensitivity analysis, spring constant variation has the greatest effect on the MASH20 performance.

Further investigation was carried out to compare MASH20 with MASH22 performance in terms of tolerance to variations in the sensing element parameters only. Figure 7-10 (bottom) shows the performance of MASH22 (left) and MASH20 (right) when the range of the sensing element parameters (proof mass [m], damping coefficient [b] and spring constant [k]) were varied by ± 15% of their nominal values. As expected, both systems are sensitive to sensing element variation, where the performance degraded accordingly, with the spring constant of the sensor having the greatest effect. Again, MASH20 performance falls from higher SNR levels than the MASH22 However, both systems achieved robust stability despite the large variation.

The digital filter design process implies a comprehensive knowledge of the $Q2NTF_{SD2}$, which is a function of the sensing element parameters and other analogue parameters, all of which are subject to manufacturing tolerance and imperfections. This causes a leakage of the single-bit quantization noise to the output and degrades the performance of the MASH20. However, since the MASH20 does not require a second stage ΣΔM, the complexity of the digital filter D_2 is reduced, as indicated in in equation 7.21. This will potentially allow the use of the digital signal processing capabilities to compensate for the filter mismatch. One method of overcoming the leakage problem is to use optimization algorithms, as will be discussed in section 7.7.

7.3.4 System Linearity

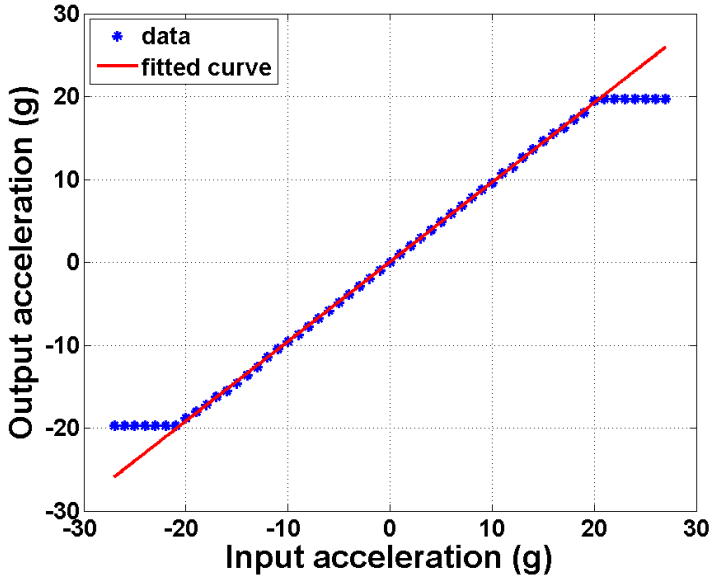


Figure 7-11: Linearity response of the MASH20 for an input range of ±20g.

The linearity of the MASH20 was obtained by simulation over an acceleration input range of ±20 g, as shown in Figure 7-11. The maximum nonlinearity of the system was obtained in terms of a percentage of the maximum diversion of the simulated data from the straight line, which was found to be 1.1%.

7.4 Accelerometer Sensing Element Design

7.4.1 Accelerometer Theoretical Design

To take advantage of the high over load acceleration input of the MASH20 a ±20 g accelerometer was designed. As discussed in section 4.4, small mass deflection can be achieved through closed loop operation so that the spring force can be neglected compared to the electrostatic feedback force (K_{fb}). The maximum acceleration input OLA is estimated as follows:

$$\text{OLA} = \frac{K_{fb} * OLF}{mass * 9.81} \qquad 7.24$$

By using equation 7.24 and an OLF of 0.85 for the MASH20 as obtained in section 7.3.2, a maximum input acceleration of 20 g acceleration is achievable by applying a feedback voltage of 12 V.

The main motivation of the accelerometer design was to maximize the feedback capacitor area and minimize the nominal gap in order to achieve high electrostatic feedback force. A simplified layout of the accelerometer is shown in Figure 7-12. The sensing and feedback capacitors were designed to have a nominal gap of 4µm. The set of capacitors (C_{aTop} and C_{aBot}) on the top and bottom serve as differential capacitive actuators, while the left and right (C_{sTop} and C_{sBot}) are differential sense capacitors. To accommodate as many actuators as possible, the proof mass is stretched horizontally, and the straight spring beams are located at the corners of the sensor. This enabled a maximum feedback area of around 7.5 μm^2.

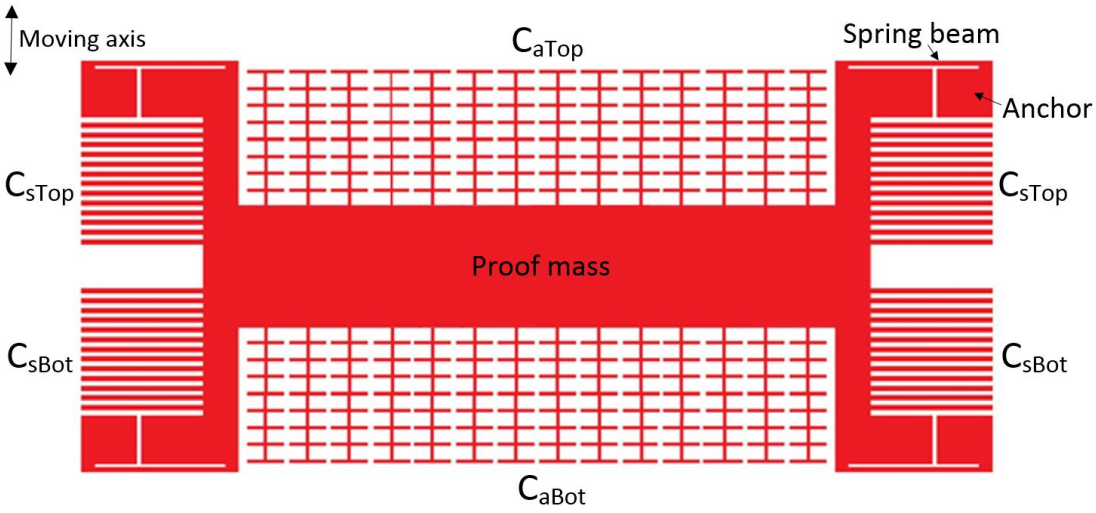


Figure 7-12: Accelerometer layout structure; the proof mass is anchored to the substrate with four spring beams. The sense capacitors are on either sides of the sensor, and the actuator capacitive are suited in top and bottom of the sensor.

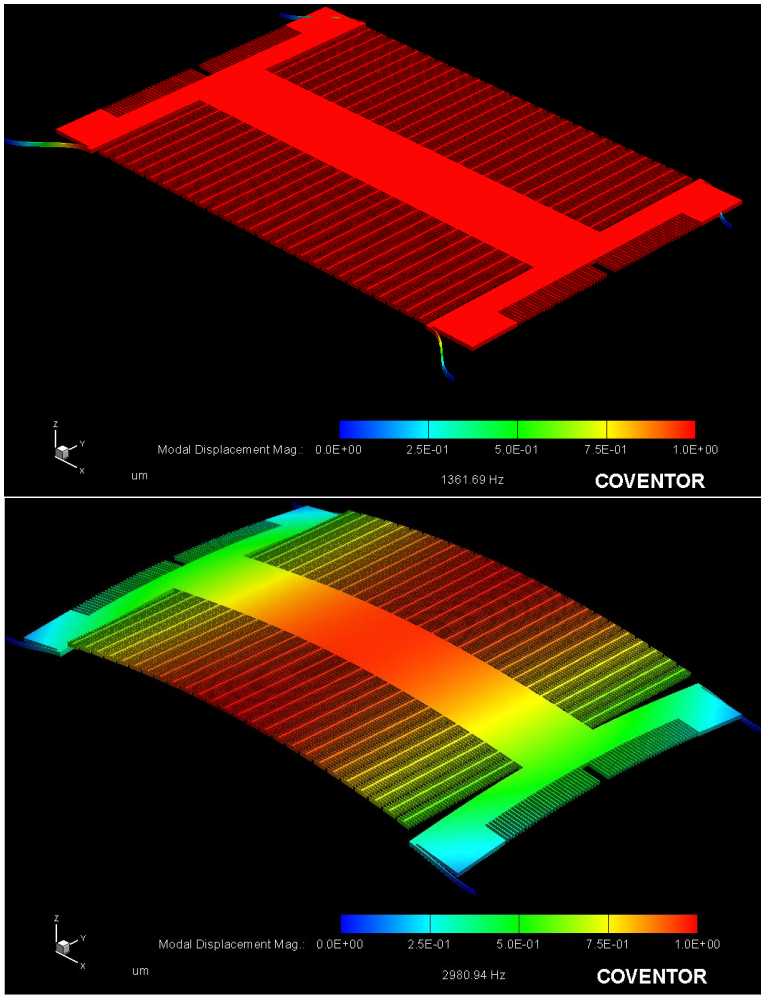


Figure 7-13: FEM analysis (using CoventorWare) of the capacitive MEMS accelerometer, (top) the first in-plane mode is at 1.361 kHz, and (bottom) the out-of-plane mode is at 2.98 kHz.

The finite element method (FEM) analysis of the sensor is shown in Figure 7-13. The first mode of the sensing element is in-plane movement at a frequency of 1.36 kHz, as shown in Figure 7-13(top). In addition, the sensor was designed so that the out-of-plane movement is located at a higher frequency, roughly 2.9 kHz, as shown in Figure 7-13 (bottom).

7.4.2 Microfabrication

A summary of the accelerometer fabrication steps is given in Figure 7-14. The SOI microfabrication process of the capacitive accelerometer was adopted from the dicing free dry release recipe described in [134]. The SOI wafer has a device layer of 50 µm, a buried oxide (BOX) layer of 2 µm, and a handle layer of 525 µm. The fabrication process was carried out using two masks: a device layer mask to define the sensor structure, and a handle layer mask to define the deep trenches, which were used to remove the handle layer block under the proof mass. The structure of the sensor was designed with a minimum feature size of 6 µm and 4 µm wide trenches. Due to a fault encountered the deep reactive ion etching (DRIE) machine, at the time of the fabrication stage, it was not possible to complete the fabrication in the Southampton Nanofabrication Centre. Therefore, the fabrication was performed by Mir Enterprises Limited – UK.

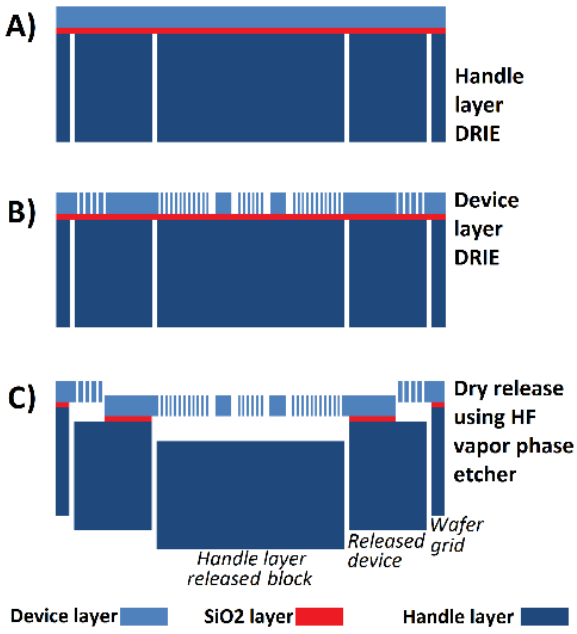


Figure 7-14: SOI microfabrication steps of the high-g accelerometer sensor. Step-A is the handle layer DRIE, step-B is the device layer DRIE, and step-C is the dry release using the HF vapour phase etcher.

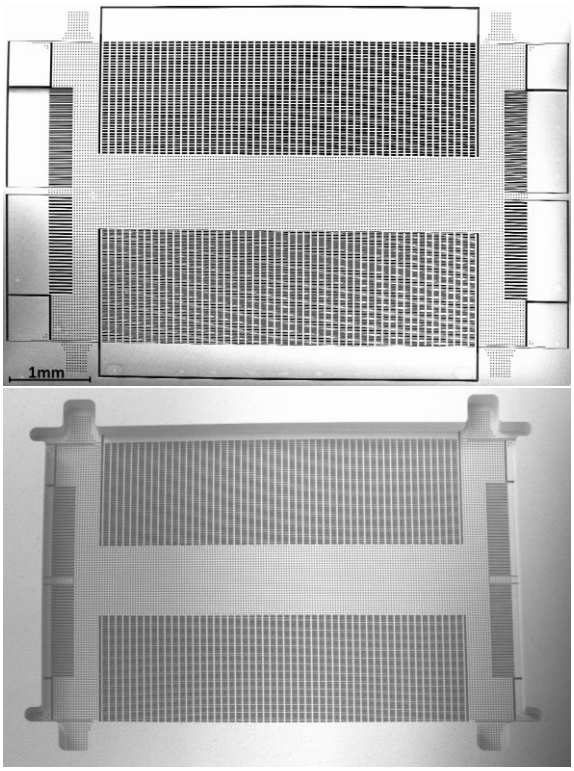


Figure 7-15: SEM images of the fabricated accelerometer, (top) top view, and (bottom) bottom view of the sensor.

The fabrication began (Figure 7-14, step A) by defining the trench pattern on the handle layer, which involved a lithography procedure using photoresist AZ9260. The handle layer trenches were then processed using deep reactive ion etching (DRIE) down to the buried oxide layer. The photoresist mask on the handle layer was then stripped using photoresist solvent and treated with an oxygen plasma asher to guarantee that no photoresist remained. The second step of the fabrication (Figure 7-14, step B) involved lithography of the device layer to pattern the wafer with the sensor's structural features. The process was carried out using AZ9260 photoresist, which was then followed by the DRIE process to etch the device layer down to the BOX.

The HF vapour phase etch technique was then used to etch the BOX layer (Figure 7-14, step C). The release process was performed in three phases, beginning with the release of the proof mass, followed by the release of the handle layer block beneath the sensor, and finally, the release of the whole sensor off a remaining wafer grid. As the HF vapour phase etcher required a wafer to be processed top side facing down, the released handle layer block was supported by the sensor rather than the anchored proof mass, and the sensor block was supported by the wafer grid.

The top and bottom sides of the MEMS sensor are shown in the SEM images of Figure 7-15. The design has four rest areas for the handle layer block so that, when fully released, the handle layer block is secured in terms of movement, and will not apply any pressure to the delicate anchored proof mass.

7.4.3 Accelerometer Characterization

The open loop frequency response of the accelerometer is shown in Figure 7-17. The red line represents the acceleration input response of the sensor. The sensor was collocated to a reference sensor and excited using a shaker table. As can be seen, the resonance frequency is at 1.315 kHz, which closely agrees with the FEM simulation. The black line represents the electrostatic force response, which was obtained by applying an excitation voltage on the top and bottom electrodes and observing the output. As discussed in section 6.2.3. This type of measurement exhibits the spring softening phenomena, as a result, the resonance frequency shifts down to 1.23 kHz. As in MASH22, the design of the MASH20 accelerometer requires correct characterization of the sensing element. The accelerometer was characterized using the procedure discussed in section 6.2.

The parameters of the accelerometer MEMS sensor and the electronic units are listed in Table 7-1.

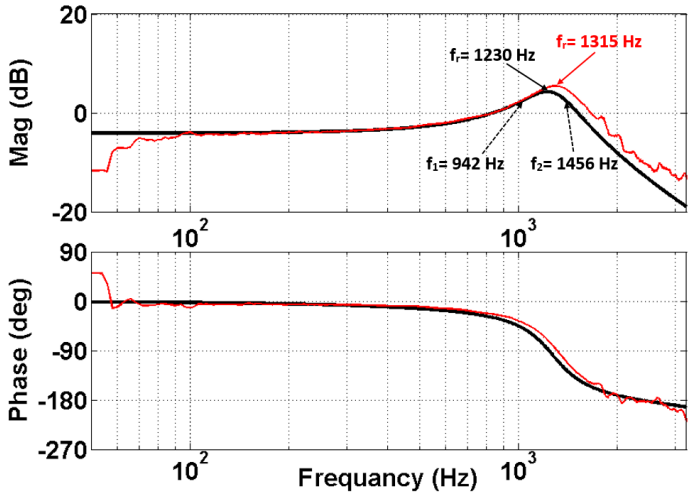


Figure 7-16: Frequency response of the high-g accelerometer. The red line represents the acceleration input with a resonance frequency at 1315Hz, and the black line represents electrostatic force with a resonance frequency shifted down to 1230Hz because of electrostatic spring softening, f1=942 Hz and f2=1456Hz.

Symbol	Description	Value
m	Proof mass [kg]	1.59×10^{-6}
b	Damping coefficient [N.s/m]	5.25×10^{-3}
k	Spring constant [N/m]	99
d_0	Nominal gap between electrodes [m]	4.5×10^{-6}
C_s	Sense capacitor [F]	2.5×10^{-12}
C_f	Feedback capacitor [F]	20.2×10^{-12}
Area	Feedback capacitor overlap area [m^2]	7.5×10^{-6}
K_{po}	Pickoff gain [V/m]	5×10^{6}
K_{bst}	Boost gain	6
Zz	Compensator zero (Hz)	6×10^{3}
Pz	Compensator pole (Hz)	45×10^{3}
Fs	Sampling frequency (Hz)	131×10^{3}
V_{fb}	Feedback voltage (V)	12
K_R, K_S	Scaling factors	1,1

Table 7-1: Accelerometer and MASH20 system parameters.

7.5 Hardware Implementation

7.5.1 Electromechanical MASH20 Electronic Circuit

As shown in the block diagram in Figure 7-17, the system is divided into analogue and digital circuit blocks. The second order EM-ΣΔM is designed using discrete analogue components with a fully differential signal path to eliminate common electronic noise. The analogue circuit accommodates the sensing element, which is followed by a pickoff circuit that measures the change in capacitance due to the acceleration and outputs a proportional voltage signal. The output signal of the pickoff is then converted into a digital signal using a 12-bit ADC. In order to close the loop, the capacitive actuators (top and bottom) are excited by the feedback pulse signals. The digital circuit consists of an FPGA unit. This unit reads the 12-bit data from the ADC at sampling rate of 131 kHz,

performs the MASH20 digital filtering, and then outputs the feedback switching signals and MASH20 acceleration output. The system is designed to operate with USB and Ethernet ports to transmit real-time data. The electronic circuit is shown in Figure 7-18.

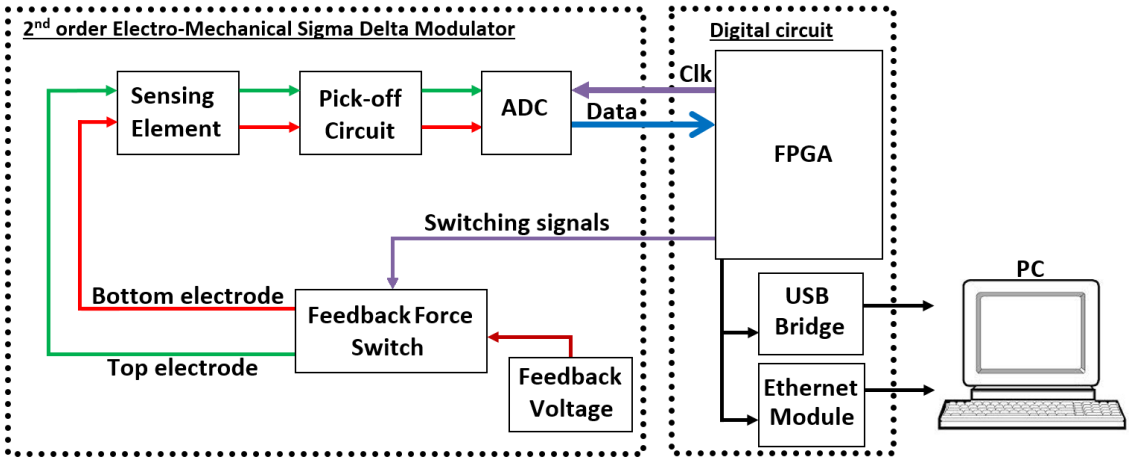


Figure 7-17: MASH20 electronic circuit block diagram, showing the analogue and digital parts of the system.

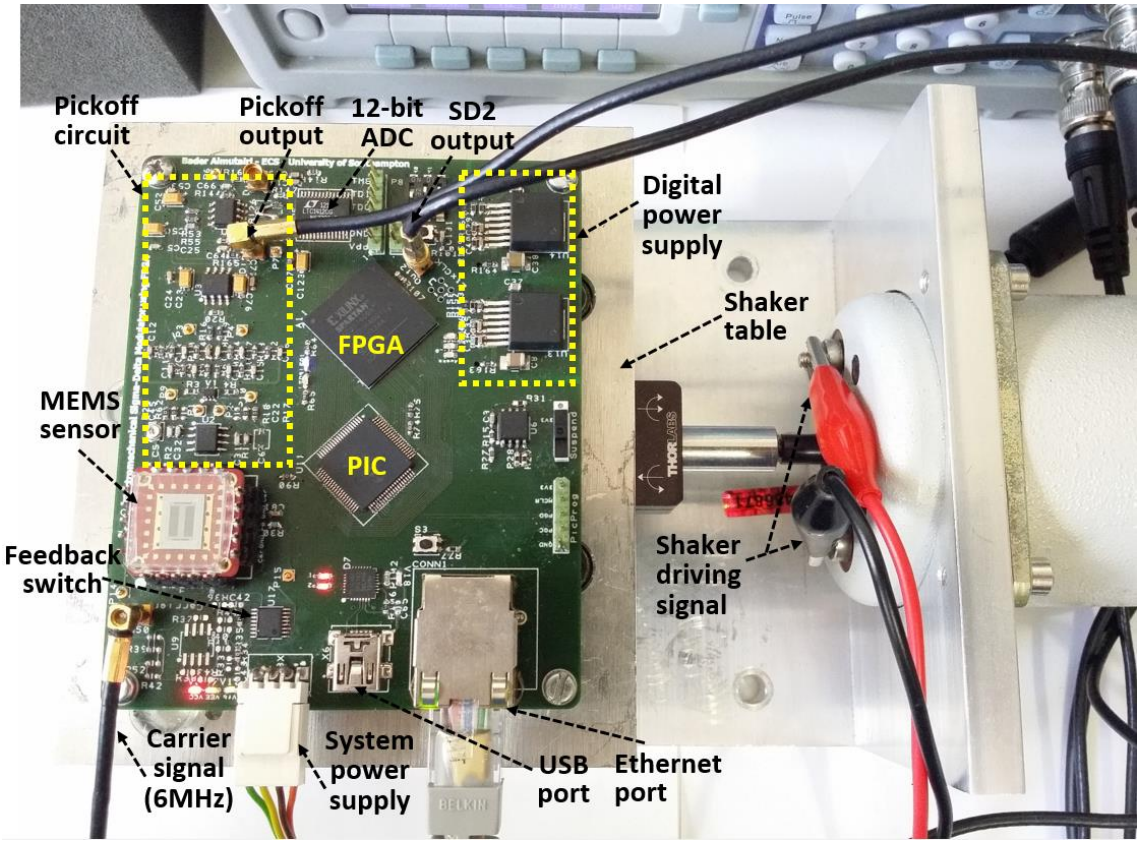


Figure 7-18: MASH20 electronic circuit showing the different parts of the system along with the MEMS accelerometer sensor that was tested using a shaker table.

Figure 7-18 shows the accelerometer MEMS sensor closely located to the pickoff circuit. The pickoff circuit of the MASH20 is similar to the one implemented in MASH22 that is discussed in section 6.3. The power spectral density of the electronic noise at the output of the pickoff circuit was measured using R&S FSV4 signal analyser [135], as shown in Figure 7-19. It shows that the electronic noise within the 1 kHz bandwidth is around 15 µV/√Hz, which was used in the simulation analysis of the MASH20 to close the gap between the theoretical and experimental analysis.

The LTC1412 12-bit ADC [136] was chosen in the MASH20. It has differential analogue inputs to minimize the common mode noise and parallel digital output. The digital output data is represented in tow's complement form with (12,9) fixed point format (i.e., 3 bits signed decimal and 9 bits fraction length), which was modelled in Simulink to consider the 12-bit quantization error.

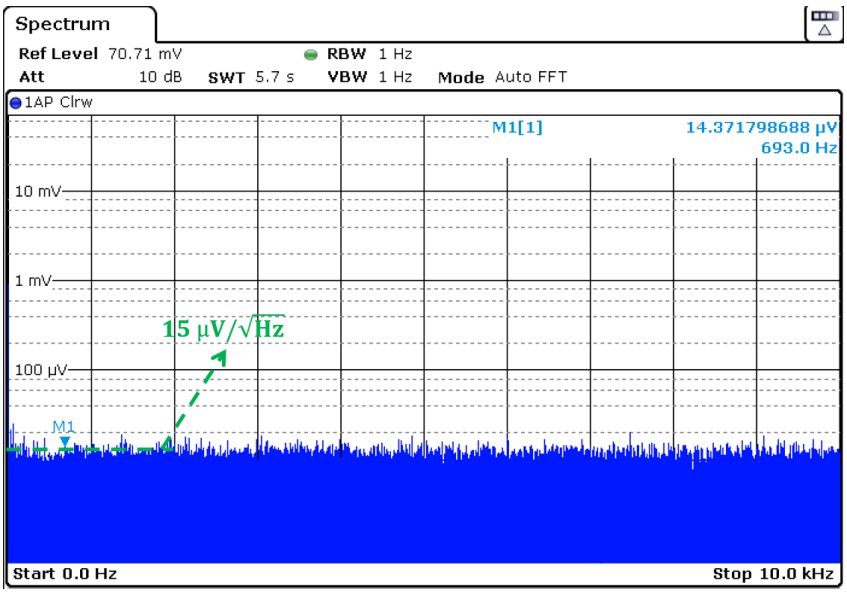


Figure 7-19: Pickoff circuit measured electronic noise, the spectrum shows 15 µv/√Hz in the frequency band up to 10 kHz.

The core of the system is the FPGA, which consumed the most time and effort during the system design, programming and implementation. After a lengthy study, the FPGA XC3S1400A-5FTG256C from Xilinx [137] was chosen as it includes adequate resources (e.g. 32 multiplier units) to accommodate the MASH20 structure. The FPGA has 256 pin of the ball gate array (BGA) package, thus, dense signal routing was involved underneath it. At power up, the FPGA reads the VHDL code from an SPI flash ROM. The code includes the ADC driver which reads the 12-bit data every clock cycle. The FPGA is

connected to an external crystal clock of 25 MHz, which was reduced to 131 kHz for MASH20 system sampling. The digital data is processed by the MASH20 architecture within the FPGA, and then passed to the PIC device.

The PIC 18F87J60 from Microchip [138] was chosen to manage the data and transmutation overhead from the FPGA. It has built in Ethernet module, which was utilized to communicate to the PC using via the user datagram protocol (UDP) (refer to Appendix C.2 for full C code). At every sampling clock, the PIC receives the MASH20 data from the FPGA and transmits it to the PC for MATLAB processing and spectrum plotting.

The digital power supply unit was carefully designed so that it can provide a stable 1.2 V and 3.3 V for the FPGA, PIC, serial peripheral interface (SPI) flash read only memory (ROM), and the USB-bridge.

In order to close the loop, the FPGA sends the switching signal to the feedback analogue switch ADG1636. The switch is then outputs a feedback voltage on one electrode (e.g. top electrode) and the other (e.g. bottom electrode) will be grounded in alternating fashion.

The schematic diagram of the PCB is presented in Appendix D.2 . The VHDL code was developed using the Xilinx ISE Design Suite software package, and is presented in Appendix B.2.

7.6 Experimental Results

Dynamic measurements were taken for MASH20 and SD2 using a shaker table set to a sinusoidal acceleration of 0.5 g at 400 Hz as shown in Figure 7-20, both of which are in good agreement with the simulated output power spectra density shown in Figure 7-8. The noise floor of the SD2 output (in blue) was roughly -90 dB (equivalent to 1.5 mg/√Hz), and the MASH20 output (in red) achieved a noise floor of approximately -130 dB (equivalent to 15 µg/√Hz). The MASH20 enhanced the performance of the system by roughly 40 dB.

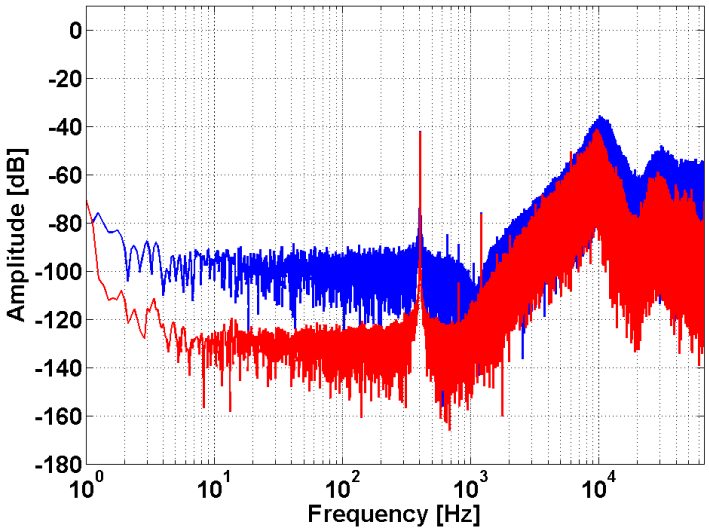


Figure 7-20: Measured noise shaping of the SD2 (blue), with a noise floor of roughly -90 dB (1.5 mg/√Hz), and the MASH20 (red), with a noise floor of roughly -130 dB (15 µg/√Hz). FFT settings: sample rate = 131 kHz, number of samples = 512k, with Hanning window.

The MASH2-0 performance was experimentally compared with a fourth order single loop SD4 and a MASH2-2, which were both implemented on the same PCB board by programming the FPGA accordingly. The results are shown in Figure 7-21. The SD4 (in blue) and MASH2-2 (in green) show fourth order noise shaping with a noise floor around -110 dB (equivalent to 150 µg/√Hz). Clearly, the MASH2-0 (in red) achieved a lower noise floor of -130 dB (equivalent to 15 µg/√Hz).

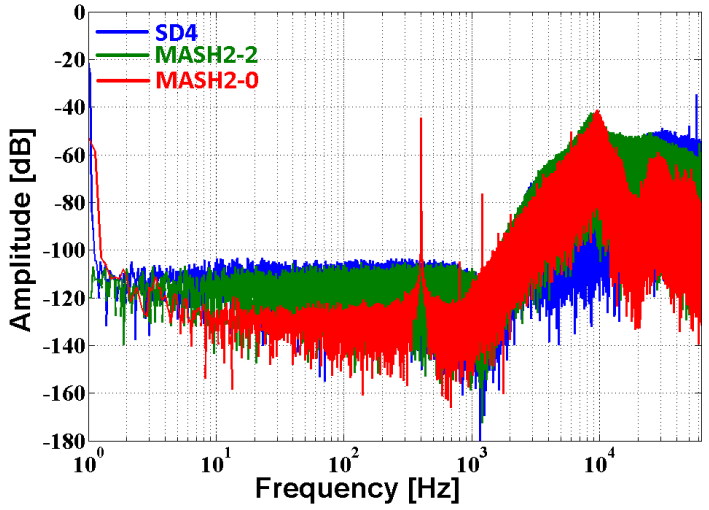


Figure 7-21: Measured noise shaping of the SD4 (blue), with a noise floor of -110 dB (150 µg/√Hz), the MASH2-2 (green), with a noise floor of -110 dB (150 µg/√Hz), and the MASH2-0 (red), with a noise floor of -130 dB (15 µg/√Hz). FFT settings: sample rate = 131 kHz, number of samples = 512k, with Hanning window.

The output of the MASH20 accelerometer was observed for a period of three hours. The gathered data were processed to calculate bias instability using the Allan variance method [139]. The measurement showed the system bias instability to be 20 µg over the three-hour period, as shown in Figure 7-23.

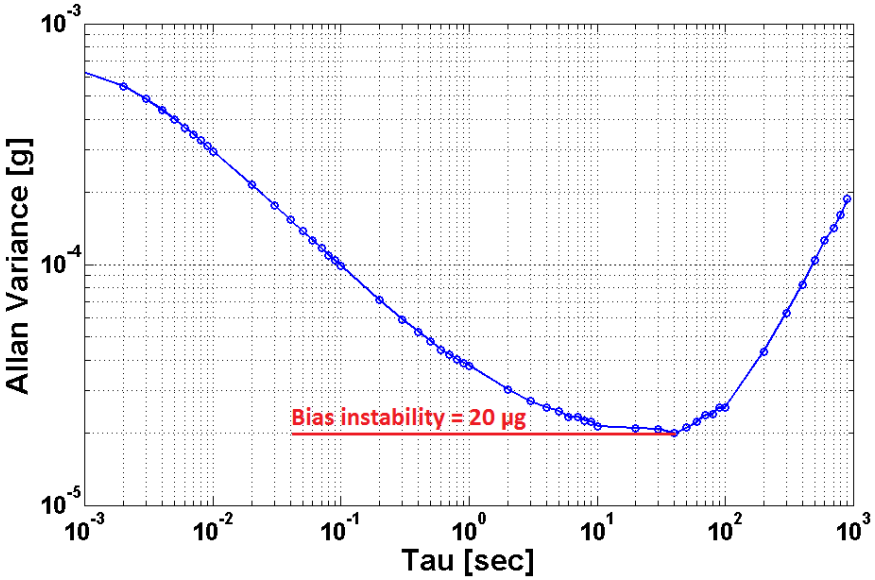


Figure 7-22: Allan variance stability analysis of the MASH20 accelerometer shows bias instability of 20 µg for a three hour period.

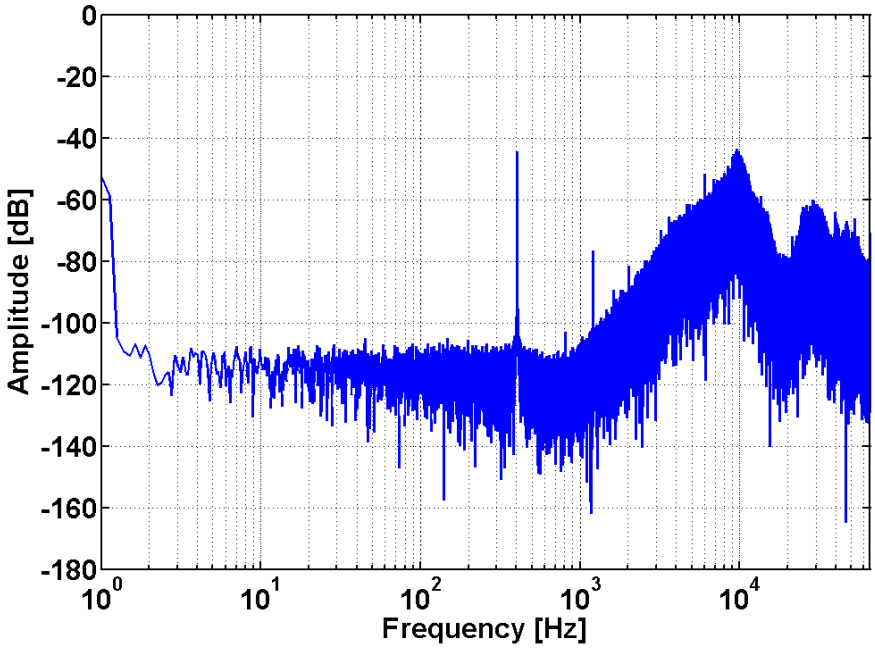


Figure 7-23: Measured noise shaping of the MASH20 after the accelerometer was replaced with another of 10% parameter variation. The noise floor increased to roughly -110 dB (150 µg/√Hz). FFT settings: sample rate = 131 kHz, number of samples = 512k, with Hanning window.

The MASH20 modulator was experimentally tested with another accelerometer from the same fabrication batch that had a parameter variation of roughly 10% compared to the original accelerometer. As expected, the modulator exhibited performance degradation, achieving a noise floor of roughly -110 dB (equivalent to 150 μg/√Hz), as shown in Figure 7-23. However, it attained 20 dB lower noise floor compared with the SD2 shown in Figure 7-20.

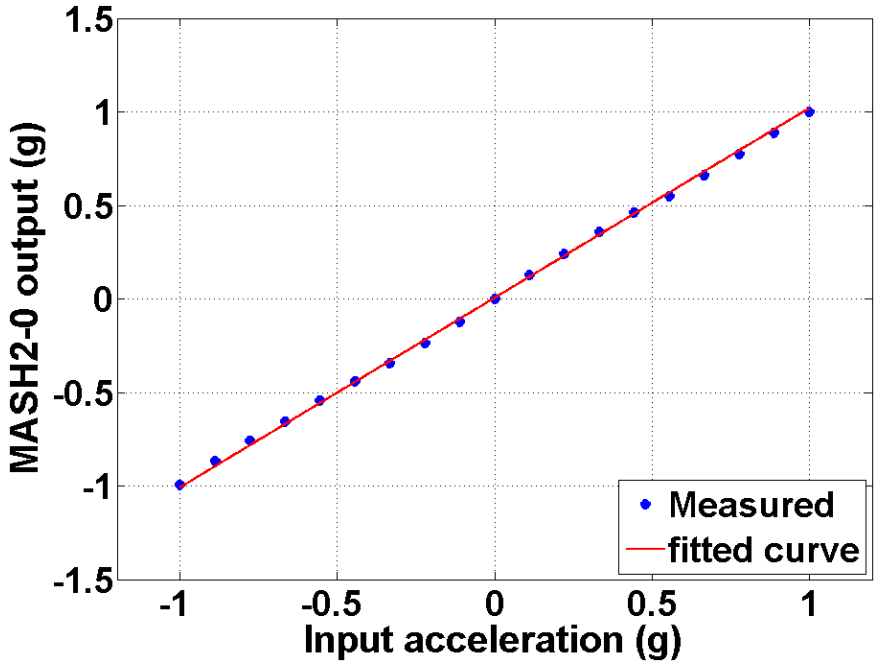


Figure 7-24: Measured linearity response over static acceleration range ±1g

The linearity of the MASH20 was experimentally measured for static acceleration, as shown in Figure 7-24. The system has a linear response over static acceleration range ±1 g. The maximum nonlinearity is calculated based on the worst diversion of the measured result with the straight line, and is found to be 1.65%.

7.7 Digital Filter Calibration of the Analogue Parameter Mismatch Using an Optimization Algorithm

Theoretical and experimental studies in this research show that one drawback of the EM-MASH is its sensitivity to variations in system parameters. This section presents one way to solve the mismatching problem that arises due to component manufacturing tolerance and imperfection using an optimization technique in the digital domain. It is not the purpose of this section to determine the best algorithm for the MASH mismatching problem, but rather to prove that optimization algorithms can be used to solve this problem. The genetic algorithm (GA) and simulated annealing (SA) are two commonly used optimization techniques and have been well reported in the literature. They will be addressed in this section as examples. A brief summary of the details of each algorithm will be provided along with their implementation using MATLAB functions.

The GA is a global search technique that mimics the process of natural selection [140] [141]. It uses a population of individual solutions and selects some individuals based on a fitness function. These selected individuals are then used within a crossover procedure to produce new offspring. Over successive generations, the fittest individuals survive and the population is directed towards the area of the optimal solution.

The SA algorithm is inspired from the physical process of heating a metal beyond melting point and then gradually lowering the temperature to end up with a solid state and minimum structural defects [142] [143]. The SA algorithm compares the current solution with another trial solution. Both solutions are then compared based on the objective function. If the adjacent solution gives a better result, this solution automatically replaces the current one, and the algorithm starts again. If the adjacent solution has a result worse than the current one, the algorithm will accept it based on the current temperature and the acceptance function, and then the algorithm starts again. This step makes the algorithm to escape a local optima and searches for nearby solutions. When the SA is in early stages, the solution with worse results will most likely be accepted, and, as the SA progresses, the algorithm gets closer to the final solution.

The optimization algorithm (GA or SA) can be performed for each individual sensor as a calibration step in order to achieve minimum noise floor. When changing the sensor with another one from the same fabrication batch, a mismatch of its parameters with the digital

filter D_2 is introduced, which can reach 20% depending on the fabrication tolerance. The sensor's parameters are the proof mass 'm', damping coefficient 'b', spring constant 'k', nominal gab 'd_0', nominal sense capacitor 'C_0' and feedback overlap area 'Area'. Accordingly, the pickoff gain 'K_{po}' and the feedback gain 'K_{fb}' are affected. This mismatch causes a leakage of the quantization noise and raises the noise floor. However, the digital part parameters, that is, the boost gain 'K_{bst}', the compensator, the scaling gain constants K_R and K_S, and the digital filter D_1 remain unchanged.

The digital filter D_2 of the MASH20 structure is simple and needs to match fewer analogue parameters compared with the MASH22. When D_1, K_R and K_S are implemented as unity gain constants, the digital filter D_2 can be further simplified and represented in terms of the sensor's parameters (i.e., m, b and k) along with other system parameters as follows:

$$D_2 = \frac{[\mathrm{bT} + \mathrm{k} + \mathrm{mT}^2]\mathrm{Z}^3 + [2\mathrm{k} - 2\mathrm{mT}^2]\mathrm{Z}^2}{[\mathrm{bT} + \mathrm{k} + \mathrm{mT}^2 + \mathrm{G}]\mathrm{Z}^3 + [2\mathrm{k} - 2\mathrm{mT}^2 + (2\mathrm{G} - \mathrm{GZ}_z)]\mathrm{Z}^2} \cdots \\ \cdots \frac{+[-\mathrm{bT} + \mathrm{k} + \mathrm{mT}^2]\mathrm{Z}}{+[-\mathrm{bT} + \mathrm{k} + \mathrm{mT}^2 + (\mathrm{G} - 2\mathrm{GZ}_z)]\mathrm{Z} - [\mathrm{GZz}]} \quad 7.25$$

where G is given by:

$$G = \mathrm{K_{po}K_{bst}K_{q1}K_{q2}K_{fb}} \quad 7.26$$

and T is the numerical integration step size and is equal to 2×Fs [131].

This reduces the implementation complexity and the computational time, as the optimization algorithm will work on five variables only (m, b, k, K_{po} and K_{fb}).

An offline optimization was used to solve the mismatch problem. The procedure starts by taking offline data from the SD2 (A) and the quantization error (B) as indicated in Figure 7-25. The GA or SA is then used to optimize the system parameters and adjust the digital filter D_2.

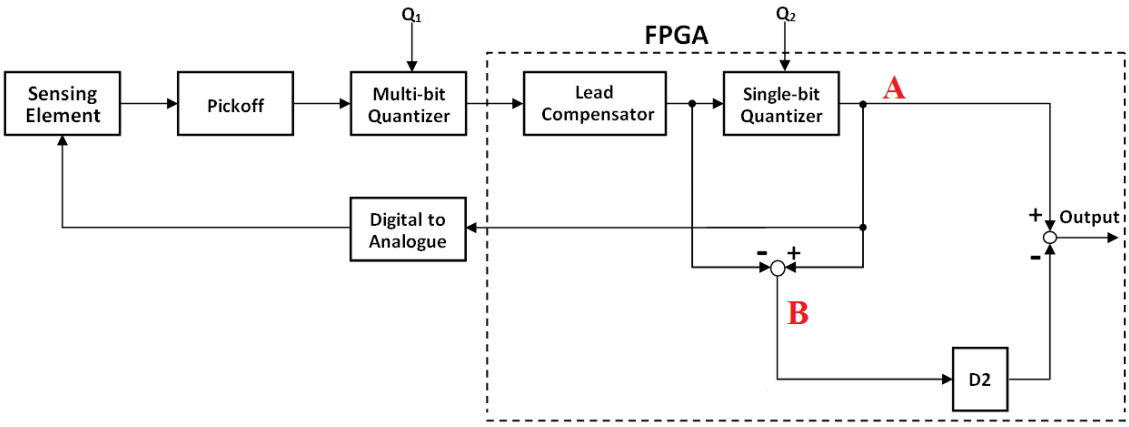


Figure 7-25. MASH20 block diagram showing the FPGA unit with the sub-unit implemented in the digital domain. The offline data of the first stage (A) and the quantization error (B) are sent by Ethernet link to MATLAB for GA optimization.

The optimization process will try to identify values for the analogue parameters by going through a large number of iterations with a predefined varying percentage of the original parameter. The GA and SA solvers offered by MATLAB [144] [145] are used to find the minimum of a function: the fitness function for the GA or the objective function for the SA. The goal is to achieve the lowest noise floor for the MASH20. The GA and SA settings are shown in Table 7-2.

GA	
Settings	Value
Population size	100
Number of generations	20
Number of parameters	5
Crossover fraction	0.8
Elite count	2
Lower and upper bounds	± 20 %

SA	
Settings	Value
Initial points of search matrix size	100 x 5
Initial temperature	100
Number of parameters	5
Re-anneal interval	100
Termination tolerance (Tolfun)	1×10^{-7}
Lower and upper bounds	± 20 %

Table 7-2: GA and SA MATLAB code settings

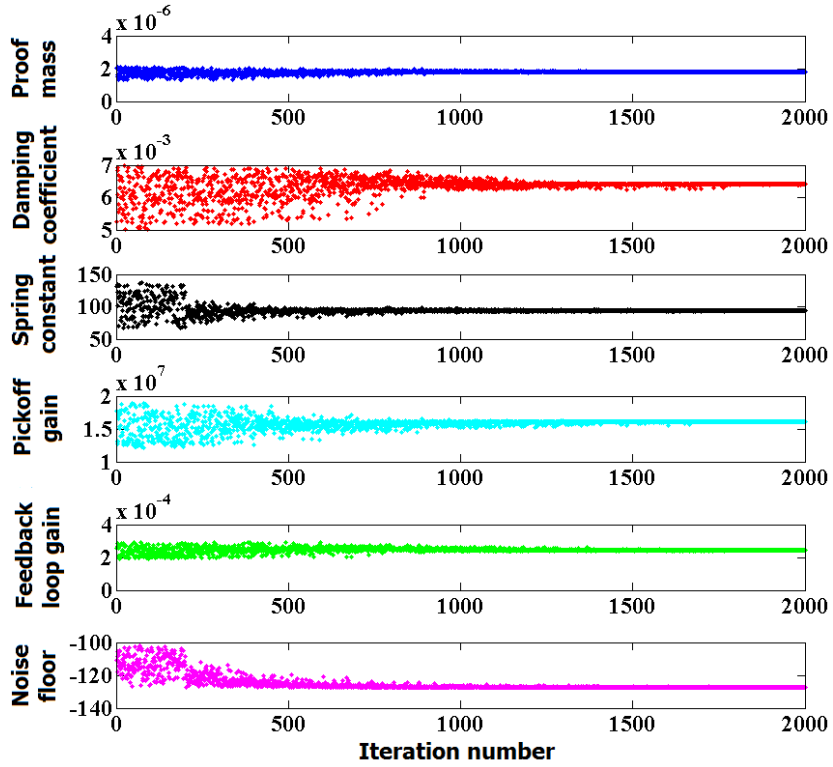


Figure 7-26. Genetic algorithm (GA) convergence plots of the proof mass (blue), damping coefficient (red) and spring constant (black), pickoff gain (cyan) and feedback loop gain (green) parameter optimization. Output score is the noise floor (purple).

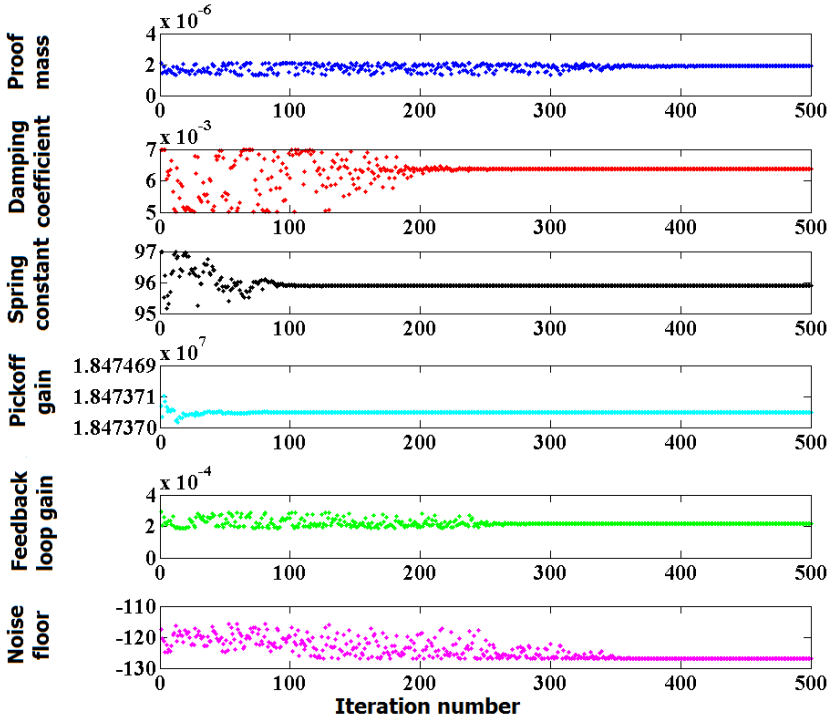


Figure 7-27. Simulated annealing (SA) convergence plots of the proof mass (blue), damping coefficient (red) and spring constant (black), pickoff gain (cyan) and feedback loop gain (green) parameter optimization. Output score is the noise floor (purple).

Figure 7-26 shows the GA convergence plot and Figure 7-27 shows the SA convergence plot of the MASH20 parameters: proof mass, damping coefficient, spring constant, pickoff gain and feedback loop gain. In both convergences, the aforementioned parameters were set with a lower and upper limit of ± 20% of the measured ones. As shown in Figure 7-26, in the early generations, the GA starts with a broad set of solutions for all parameters. The GA then generates a new set of solutions based on the current generation that produced the minimum noise floor value. The process is repeated from one generation to another until all parameters converge on optimum values that give minimum noise floor value. The convergence plot shows that after around 1800 generations (15 min), optimum parameters were achieved with a minimum noise floor of -127 dB. On the other hand, as shown in Figure 7-27, the SA algorithm has similar behaviour; however, it achieved the -127 dB noise floor after around 380 iterations (3 min), which is about five times faster than the GA algorithm. Although these methods prove their ability to solve the digital filter mismatch problem for the MASH20, they can be applied to any MASH structure.

Symbol	Measured	Genetic Algorithm		Simulated Annealing	
		1st run	2nd run	1st run	2nd run
m	1.59×10^{-6}	1.814×10^{-6}	1.816×10^{-6}	1.787×10^{-6}	1.887×10^{-6}
b	5.70×10^{-3}	6.41×10^{-3}	6.11×10^{-3}	6.92×10^{-3}	6.37×10^{-3}
k	94	94.09	91.36	93.14	95.89
K_{po}	15.6×10^{6}	16.12×10^{6}	16.49×10^{6}	17.7×10^{6}	18.4×10^{6}
K_{fb}	235×10^{-6}	2438×10^{-6}	2265×10^{-6}	2143×10^{-6}	2162×10^{-6}
Time (min)	-	15	15	3	3

Table 7-3: MASH20 digital filter D_2 parameters comparison obtained by experimental measurement, GA and SA algorithms.

Table 7-3 shows the parameters of MASH20 digital filter D_2 obtained by experimental measurement, GA and SA algorithms. The results from the GA and SA are obtained twice. It is clear that there is no unique solution, as both GA and SA algorithms provided two different solutions that give the minimum noise floor. Both solutions are within 4-10 % of the measured values. The MASH structure together with the mentioned GA or SA calibration methods can be used to closely characterize the sensing element, the

pickoff gain and the feedback loop gain, as an alternative to the characterization method presented in section 6.2. The output result of the GA and SA algorithms are obtained with less time, cost and effort when compared with characterization procedures that require costly laboratory equipment and a great deal of time, such as the MSA400 for optical measurement. However, these are preliminary results. More experimental work and research will be addressed in future work with the FPGA implementation to reach a decisive conclusion regarding the feasibility of these algorithms for characterizing the sensor. Nevertheless, it has been proven that the optimization algorithms are able to solve the mismatch problem and achieve the lowest noise floor.

7.8 Summary

This chapter presented the theoretical and experimental analysis of a novel EM-MASH20 applied to an accelerometer implemented using the dual quantization technique. Most of the system's components were implemented in the digital domain using an FPGA, which reduces electronic crosstalk noise and temperature effects, as well as providing extra flexibility in terms of modifying the modulator's parameters. This approach also provides excellent matching between the digital filter and the modulator's digital parameters. The power consumption of the MASH20 was around 125 mA at 9 volts, which is about half the power consumed by the MASH22, which was 300 mA at 9 volts. However, the design and implementation of the MASH20 encountered a level of complexity. The signal routing and placement of the FPGA required good PCB layout knowledge and was time consuming. Moreover, the FPGA chip requires ball grid array assembly tools; therefore, extra cost is added during PCB manufacturing. Furthermore, the MASH20 design requires good knowledge of the VHDL/Verilog coding.

A capacitive MEMS accelerometer was designed on this study to achieve high input acceleration of 20 g when it is operated in digital closed loop. The simulation analysis of the MASH20 accelerometer showed that the performance of the SD2 is enhanced by around 40 dB in terms of the noise floor.

The experimental analysis was carried out with a sampling frequency of 131 kHz, the noise floor of the power spectral density of the SD2 output was around -90 dB (equivalent to 1.5 mg/√Hz), while the MASH20 showed a high performance output that reached around -130 dB (equivalent to 15 µg/√Hz) noise floor within a bandwidth of 1 kHz. The

sensor achieved a maximum nonlinearity of 1.65% and a bias instability over a period of three hours of 20 µg. The experimental measurements of the study closely matched the simulated measurements and prove that the MASH20 structure is applicable for high performance accelerometer.

The MASH20 accelerometer shares the same advantages of the MASH22 in term of robust stability, high overload input acceleration and linearity compared with single loop EM-ΔΣM. Furthermore, MASH20 reduces the complexity of MASH22 and adds flexibility to design the digital filter. Nevertheless, as in MASH22, the MASH20 is sensitive to parameter variations which can potentially be addressed by an adaptive control algorithm. Calibration method was investigated to solve this problem by utilizing the digital domain capabilities. The method is based on the genetic algorithm (GA) which was instigated and verified using MATLAB.

Chapter 8: **Conclusion and Future Work**

8.1 Conclusion

In this research, EM-MASH were considered structures for interfacing with capacitive MEMS inertial sensors, which deploy closed-loop control on inertial sensors and emit a measurement in the form of a digital signal. The potential advantages of an EM-MASH over the single-loop, high-order ΣΔMs applied to inertial MEMS sensors are its inherent stability and high overload input level. These properties are attributed to the use of lower-order ΣΔMs in the individual stages of MASH. Furthermore, MASH presents a high dynamic range and high noise-shaping performance because of its overall high-order ΣΔM architecture.

So far, the EM-MASH has been inadequately explored. The current work serves as a solid basis for the application of the EM-MASH, for which various MASH architectures (MASH21, MASH22, MASH211, MASH221 and MASH222) were theoretically studied. The findings were validated with simulations.

The EM-MASH also overcomes the disadvantages that affect single-loop ΣΔM. The former exhibits better stability, dynamic range and high overload input. During the simulation, for example, MASH22 generates an SNR of 112 dB, a dynamic range of 110 dB and an overload input of 0.917. Further study on MASH22 was carried out to confirm whether increasing the MASH order enhances performance. The simulation shows an SNR value of 133 dB under electronic noise, a dynamic range of 130 dB and an overload input of 0.917.

An experimental comparative study was performed for MASH22 architecture and a fourth order single-loop (SD4) architecture. Both architectures achieve a noise floor level of –110 dB (equivalent to 19 μg/√Hz) at a bandwidth of 1 kHz. The SD4 architecture proves resistant to variations in sensing element parameters, whereas MASH22 exhibits a degraded performance because of the leakage of quantization noise to the input signal. For its lower-order stages, however, MASH22 shows excellent stability and a high overload input threshold. This architecture also takes advantage of the high dynamic range and high noise-shaping performance of higher-order ΣΔM.

The research presents a novel EM-MASH that employs the dual quantization technique and adopts an electromechanical 2-0 multi-stage noise-shaping structure (EM-MASH20). MASH20 is designed to generate a performance higher than that of a fourth order MASH22 structure with a simpler and configurable structure. Moreover, MASH20 does not require an electronic second-stage ΣΔM, thus reducing digital filter complexity to a level lower than that achieved with MASH22.

Most of the MASH20 components were implemented in the digital domain by using an FPGA, which reduces electronic cross-talk noise and temperature effects, as well as provides extra flexibility in terms of modifying the modulator's parameters. This approach also enables excellent matching between the digital filter and the modulator's digital parameters. Nevertheless, as in MASH22, the MASH20 structure is sensitive to parameter tolerances of micro-machined sensing elements. However, the reduced complexity of the MASH20 filter eases the use of digital signal processing capabilities in compensating for filter mismatching. One method for resolving the filter leakage problem is to use the optimization algorithms discussed in Section 7.7.

To take advantage of the high over load acceleration input of the MASH20 a ±20 g accelerometer was designed. The simulation analysis of the MASH20 accelerometer shows that the performance of the second order EM-ΣΔM (SD2) improves by around 40 dB in terms of noise floor. An experimental analysis was carried out with a sampling frequency of 131 kHz. The noise floor of the power spectral density of the SD2 output is around -90 dB (equivalent to 1.5 mg/√Hz), whereas that of MASH20 reaches around -130 dB within a bandwidth of 1 kHz. The sensor achieves a maximum nonlinearity of 1.65% and a bias instability of 20 μg over a period of three hours.

Table 8-1 shows a performance comparison between the MASH20, MASH22 and SD4 results, along with those of other high-order single-loop EM-ΣΔM systems, as achieved in this study. For all control loop structures, the noise floor of the system did not exceed the sensor's mechanical noise floor. It was clear in this research that, with a sampling frequency of 131 kHz and using the same sensing element and noise environment (achieved by using same PCB board), the MASH structure a provided higher dynamic range than the fourth-order single loop. Furthermore, the MASH20 achieved a noise floor of 15 μg/√Hz, while the MASH22 and SD4 achieved the same noise floor of around 150 μg/√Hz.

As mentioned in section 3.1, the performance of the sensor depends on three different areas: sensor mechanical design, the front-end interface and the control structure. Different performance results were reported for the same control structure, that is, SD4 in work 3, 4, 5 and 6. This is because they employed different sensing element designs, feedback voltage and test environments. In work 7, a high-sensitivity accelerometer with an out-of-plane structure was used. It has a mechanical noise floor of 0.4 μg/√Hz, a closed-loop noise floor of 1.7 μg/√Hz and a dynamic range of 120 dB. The system utilised an overload recovery mechanism, which switches to a second order EM-ΣΔM architecture in the event of a high-g shock. With regards to the control structure, if this system employs the MASH20 structure instead of the fifth-order single-loop ΣΔM, the dynamic range is expected to increase and the stability of the system will be improved.

Work	Feedback voltage [V]	Full scale range [g]	Noise floor [μg/√Hz]	Bias instability	Dynamic range	Mechanical noise floor
1. MASH20 - this work	12	21	15	20 μg	118 dB	8.1 μg/√Hz
2. MASH22 - this work	12	21	150	-	106 dB	8.1 μg/√Hz
3. SD4 - this work	12	12	150	-	95 dB	8.1 μg/√Hz
4. SD4 [23]	12	40.3	11.3	18.1 μg	131 dB	4.6 μg/√Hz
5. SD4 [22]	5	-	150	-	-	-
6. SD4 [87]	3	-	4	-	95 dB	1 μg/√Hz
7. SD5 [82]	9	11	1.7	0.15 mg	120 dB	0.4 μg/√Hz

Table 8-1: Performance comparison for MASH22, MASH20 and other reported EM-ΣΔMs.

8.2 Future Work

This section addresses the possible areas for improvement of EM-MASH. These areas were identified on the basis of the current findings.

8.1.1 Practical Implementation of the On-board Genetic Algorithm

The theoretical and experimental studies in this research show that the only drawback of EM-MASH is its sensitivity to variations in system parameters. Section 7.7 presents a digital calibration technique that uses the optimization algorithms, in which the digital domain of the system is used to overcome parameter sensitivity. The section discussion confirms the ability of the GA and the SA to identify system parameters and restore the performance of EM-MASH to its optimum level. Given that MASH20 has a reduced digital filter and few parameters to solve, the unused resources within the FPGA can be used to perform on-board digital calibration. For example, a customisable FPGA implementation of a general-purpose GA engine was reported in [146]. It was successfully synthesised on Xilinx Virtex II Pro FPGA (XC2VP30-7FF896) at a clock speed of 50 MHz. The implementation featured the use of a 13% logic slice count and a 1% memory block count. The same implementation can be adopted and tailored to the optimisation of MASH20 parameters.

8.1.2 Adaptive Control Algorithm

Adaptive algorithms are underlain by a well-known and solidly developed concept [147], which has found application in system identification, noise cancellation, channel equalisation and signal predication. Noise cancellation that uses least mean square (LMS) adaptive control was successfully applied in the electronic MASH2-0 [82-84] to compensate for quantization error leakage. The idea is to inject a test signal right before the location of the 1-bit quantizer and attempt to cancel the signal using an LMS adaptive filter (Figure 8-1). If the signal is cancelled, the quantization error is also cancelled because they both follow the same path.

LMS works in three steps:

1- Digital filtering is conducted on ‘E’, which contains the test signal and the error signal.

2- Error leakage is estimated by calculating the correlation between the test signal and the MASH20 output. If they are completely uncorrelated, then the result is zero, indicating that the test signal is completely removed from the MASH20 output.
3- LMS coefficients are updated on the basis of the correlation results, after which Step 1 is re-initiated.

LMS continually updates its coefficients on the basis of correlation results; thus, the test signal and the quantization error gradually minimise with time [106]. Consequently, the performance of MASH eventually improves. Any mismatch that occurs between digital filter D_2 and its analogue parameters results in the leakage of the test signal. Accordingly, LMS compensates for the mismatch. This approach is applicable in the time domain with the use of a finite impulse response (FIR) filter. The test signal must be carefully chosen, so that it is white and uncorrelated with the final output and the input signal. The effectiveness of the approach is supported by the simplicity of hardware implementation, which can be carried out using shift registers and adders. Such implementation substantially reduces the complexity of a design and can be used in an application specific integrated circuit (ASIC).

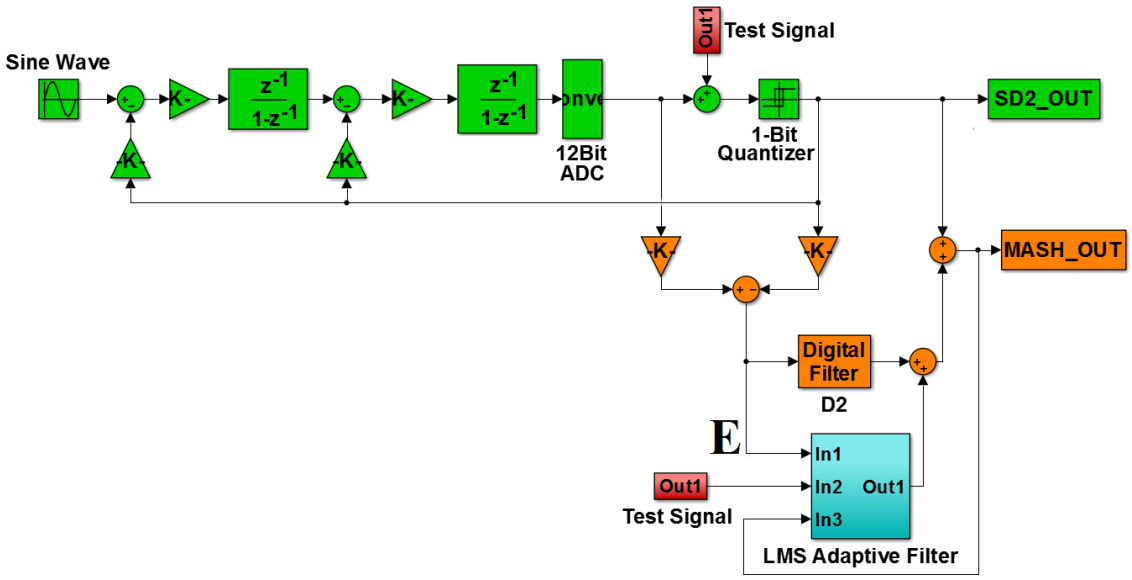


Figure 8-1: Simulink model of an electronic MASH20-ΣΔM with an LMS adaptive filter.

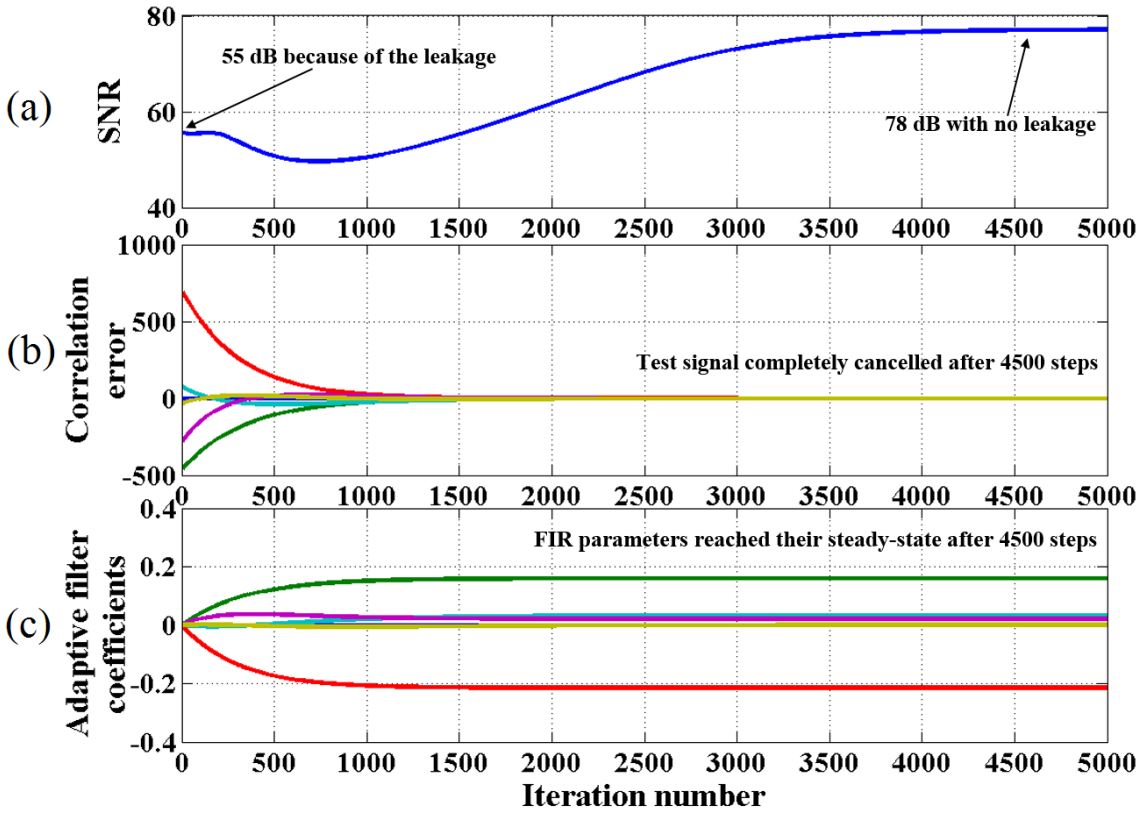


Figure 8-2: Adaptive control filter output; (a) SNR of the MASH20 output; (b) correlation result for the five FIR coefficients; and (c) LMS FIR coefficients.

To demonstrate the LMS adaptive algorithm, the MASH20 Simulink model in Figure 8-1 was simulated with a parameter variation of 10%. Figure 8-2 shows (a) the SNR value of MASH20, (b) the correlation result for the five FIR coefficients and (c) the five FIR coefficients. At first iteration, MASH20 shows an SNR of 55 dB, which is due to the mismatch in the digital filter parameters. The initial values of the five FIR coefficients were set to zero. Accordingly, high correlation values were expected. As the number of iterations increases, the correlation value decreases and the FIR coefficients approach their optimal values. After about 4500 iterations, the correlation values are at their minimum (close to zero). Accordingly, little change in FIR coefficients occurs, and the output of MASH20 improves with an optimum SNR value of around 78 dB.

The demonstration above was performed for the electronic MASH20 with an FIR of five coefficients. However, this approach is not as straightforward when implemented with EM-MASH because of the large number of system parameters. Further study and verification should be carried out.

8.1.3 Pickoff Circuit Performance Enhancement

This research concludes that the pickoff circuit noise is the dominant noise in the MASH system. Therefore, other pickoff circuit techniques can be employed to enhance MASH performance. Continuous time (CT) pickoff circuits generally have a lower noise floor than switched-capacitor (SC) pickoff circuits [148]. However, SC pickoff circuits are desired for implementation in ASIC due their robustness and compatibility with other discrete time signal components. SC pickoff circuits with correlated double sampling (CDS) and chopper stabilisation techniques generate good performance; in some publications [67] [149-151], an electronic noise of tens of nV/√Hz was achieved. These circuits can eliminate 1/f noise, voltage offset, operational amplifier finite gain and kt/c switching noise [67], while the chopper stabilisation further reduces the offset [152]. A configurable ultra-low noise SC pickoff circuit with resolution of 50zF/√Hz has been manufactured by Si-ware [153]. Such IC can be used as interface for capacitive MEMS accelerometers (or gyroscopes) and employed within the MASH structure.

The advantages of the pickoff circuit implemented with ASIC are low electronic noise, low parasitic capacitance when wire-bonded near the MEMS sensor and low power requirement for operation. In fact, EM-MASH20 can be implemented with ASIC, in which a SC pickoff circuit can be interfaced to a flash ADC. The flash ADC can then be interfaced to an FIR digital filter, which consists of the LMS adaptive control discussed in the previous section. The interface and control circuit and the MEMS sensor can be integrated within a single package that offers good shielding and immunity against external noise sources, such as the electromagnetic interference (EMI).

Appendices

Appendix A MATLAB Code

A.1 SNR Estimation

```
%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%                                SNR ESTIMATION
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
w=2*pi*(0:bw);                  %define the bandwidth in rad/sec

[Mag0,Phase] = bode(Q2NTF,w);      %obtain the magnitude of the NTF

MagDb=[];
MagSqr=[];
Mag=[];
for index=1:bw+1,               %calculate the square of magnitude
  Mag(index)=Mag0(:,:,index);
  MagSqr(index)=Mag(index)^2;
end

q=2;                         %quantization level

esqrRMS=q^2/12;                 %refer to equation 4.18

Esqr=esqrRMS*2/Fs;              %refer to equation 4.19

NsqrOut=Esqr*MagSqr;            %refer to equation 4.20

nsqr0=trapz(NsqrOut);           %refer to equation 4.21

RMSinput=((x)/sqrt(2))^2;       %refer to equation 4.22

SQNR_ESTIMATION=10*log10(RMSinput/nsqr0) %refer to equation 4.23
```

A.2 Design and Simulation of MASH222

```
clear all
clc
t0=clock;
epsilon=8.854e-12;
x=0.9173;

%%%%%%%%%%%%% Parameter of the Sensor %%%%%%%%%%%%%
Vfeedback=10;
m=1.7452e-6;
b=3.5721e-4;
k=5.492;
AA=1.89e-6 ;
d0=6e-6;
d1=d0;
Kfb=0.5*epsilon*AA*(Vfeedback^2)/(d0^2);
```

```
Kpo=5e6; % Kpo=8e8;
Kbst=200;
% %%%%%%%%%%%%%%%%%%%%%
enoise=6e-9;                    % Elelctronic Noise
Lfactor=Vfeedback/d0;           % Feedback Linearizaton factor
Fn=sqrt(k/m)/2/pi;                 % The natrual frequency of the
sensor

%%%%%%%%% Quantization Gains %%%%%%
Kq1=1;
Kq2=0.5947;
Kq3=0.4478;
KR=1;
KS=0.5;
K2=1.7;
KR2=1;
KS2=1.6805;
K3=0.7686;
%%%%%%%%%%%%%%%%%%
bw=1024;             % Bandwidth
OSR=64;              % Oversampling ratio
Fs=OSR*2*bw;         % Oversampling frequency
Ts=1/Fs;             % Sampling Time
N=64*1024;           % Samples number
Finput=256;          % input Frequency
Ntransient=2048;
%%%%%%%%%%%%%%%%%%%%%
%%%%%%% Compensator Design %%%%%%%%%%

Fph=45400;          % Frequency in Hz
PhaseLag=56;        % Phase Lead in degrees
WL=Fph*2*pi;        % convert from Hz to rad/sec
PhiMargin=PhaseLag*(pi/180); % convert from deg to radian
alpha=(1+sin(PhiMargin))/(1-sin(PhiMargin));
zero=WL/sqrt(alpha);
pole=alpha*zero;
Comp=tf([1 zero],[1 pole]);

%%%%%%%%%%%% END %%%%%%%%%%%%%%%%%%%
%%%%%%%%%%
sysC=tf(1, [m b k]);
MzCz=c2d(sysC*Comp,Ts,'zoh');

A=tf([1],[1 -1],Ts);
[Anum,Aden] = tfdata(A);

B=tf([1],[1 -1],Ts);
[Bnum,Bden] = tfdata(B);

A3=tf([1],[1 -1],Ts);
[A3num,A3den] = tfdata(A3);

B3=tf([1],[1 -1],Ts);
[B3num,B3den] = tfdata(B3);

G1=1;
G2=2.5;
G3=1;
```

```
G4=2.5;


NTF1=minreal(1/(1+MzCz*Kpo*Kbst*Kq1*Kfb));
STF2=minreal(A*B*Kq2/(1+G1*A*B*Kq2+G2*B*Kq2));

NTF2=minreal(1/(1+G1*A*B*Kq2+G2*B*Kq2));
STF3=minreal(A3*B3*Kq3/(1+G3*A3*B3*Kq3+G4*B3*Kq3));
NTF3=minreal(1/(1+G3*A3*B3*Kq3+G4*B3*Kq3));


%%%%%%%%%%%%%%%%%%%%%%%%%%
%%% Desing the Digital Filters %%
D1=tf(1,[1 0 0 0 0],Ts);
[D1num,D1den] = tfdata(D1);

D2=minreal(D1*NTF1/(STF2*(NTF1*(KS*K2-KR*K2/Kq1)+KR*K2/Kq1)));
[D2num,D2den] = tfdata(D2);

D3=minreal(D2*NTF2/(STF3*(NTF2*(KS2*K3-KR2*K3/Kq2)+KR2*K3/Kq2)));
[D3num,D3den] = tfdata(D3);

%%%%%%%%%%%%% END %%%%%%%%
 %%%%%%% Calculate the Noise Transfer Functions %%%%%%%%

F1=minreal(MzCz*Kpo*Kbst*Kq1*D1/(1+MzCz*Kpo*Kbst*Kq1*Kfb));
F2=minreal(A*B*Kq2*D2/(1+G1*A*B*Kq2+G2*B*Kq2));
F3=minreal((KS*K2*Kq1*Kpo*Kbst-
KR*K2*Kpo*Kbst)*MzCz/(1+MzCz*Kpo*Kbst*Kq1*Kfb));
F1a=minreal(Kq1*D1/(1+MzCz*Kpo*Kbst*Kq1*Kfb));

STF=minreal(F1-(F2*F3));
Den=minreal(1+Kq2*G1*A*B+Kq2*G2*B);

Q3NTF=minreal(-D3*NTF3);
E1NTF=minreal(STF/(MzCz*Kpo));
E2NTF=minreal(-D2*Kq2*B/Den);
E3NTF=minreal(-D2*Kq2/Den);

%%%%%%%%%%%%%%%%%% END %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

P = bodeoptions;
P.grid ='on';
P.XLim = [1 Fs/2];
P.FreqUnits ='Hz';
P.PhaseUnits ='deg';
figure(4);
bodemag(Q3NTF,'r',E1NTF,'g',E2NTF,'b',E3NTF,'.k',STF,'m',P);
legend('Q3NTF','E1NTF','E2NTF','E3NTF','STF',4);


%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%
%
************************************************************************
******
% Open Simulink diagram first
```

```
%
*****************************************************************
******

sim('MASH222_S3wc',    (N+Ntransient)/Fs);    %    Starts    Simulink
simulation
%
*****************************************************************
******
%   Calculates SNR and PSD of the bit-stream and of the signal
%
*****************************************************************
******
Vref=1;
w=hann(N)';
% w=hann_pv(N);
f=Finput/Fs ;        % Normalized signal frequency
fB=N*(bw/Fs);        % Base-band frequency bins
yy1=zeros(1,N);
yy1=bitstream(2+Ntransient:1+N+Ntransient)';
ptot=zeros(1,N);
[snr,ptot,ps,pn]=calcSNR(yy1(1:N),f,fB,w,N,Vref);
figure(20);
semilogx(linspace(1,Fs/2,N/2), pn(1:N/2), 'b');
xlabel('Frequency [Hz]')
ylabel('Amplitude [dB]')
axis([0 Fs/2 -280 0]);
grid on;
hold on
figure(1);
semilogx(linspace(1,Fs/2,N/2), ptot(1:N/2), 'r');
xlabel('Frequency [Hz]')
ylabel('Amplitude [dB]')
axis([0 Fs/2 -300 0]);
text_handle = text(2,-75, sprintf('MASH_2_2_2 SNR = %4.1fdB E-Noise=
%g',snr,enoise));
grid on;
hold on

%%%%%%%%%%%%%%                    Display                    parameters
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
s0=sprintf('** Simulation Parameters **');disp(s0)
s1=sprintf('    Fs(Hz)=%1.0f',Fs);disp(s1)
s2=sprintf('    Ts(s)=%1.6e',Ts);disp(s2)
s3=sprintf('    Fin(Hz)=%1.4f',Finput);disp(s3)
s4=sprintf('    BW(Hz)=%1.0f',bw);disp(s4)
s5=sprintf('    OSR=%1.0f',OSR);disp(s5)
s6=sprintf('    Npoints=%1.0f',N);disp(s6)
s7=sprintf('    tsim(sec)=%1.3f',(N+Ntransient)/Fs);disp(s7)
s8=sprintf('    Nperiods=%1.3f',N*Finput/Fs);disp(s8)
s1=sprintf('    SNR1(dB)=%1.3f',snr);disp(s1)
s3=sprintf('    Natural freq.(Hz)=%1.3f',Fn);disp(s3)
s4=sprintf('                      Simulation       time       =%1.3f
min',etime(clock,t0)/60);disp(s4)
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%
```

A.3 MASH20 Digital Calibration Using GA

```
clc;
close all;
global gn;
gn=1;
%
**********************************************************************
******
% GA parameters
%
**********************************************************************
******
population_size = 150;        % GA population size
Gen_No = 30;                  % GA number of generations
Params_No=5;                  % Number of parameters
%
**********************************************************************
******
%
**********************************************************************
******
LB(1) = 1e4;          UB(1) = 100e6;     % Kpo
LB(2) = 1e-06;        UB(2) = 2e-06;     % mass of proof mass in kg
LB(3) = 0.0001;       UB(3) = 0.01;      % damping factor
LB(4) = 10;           UB(4) = 220;       % spring constant.
LB(5) = 1.7e-04;      UB(5) = 5e-04;     % Kfb
%
**********************************************************************
******
%
**********************************************************************
******
% GA initialization
%
**********************************************************************
******
init_pop = init_population(LB,UB,population_size);
options                =                gaoptimset('PopulationSize',
population_size,'Generations',...
    Gen_No,'InitialPopulation', init_pop);
%
**********************************************************************
******
% Run the GA
%
**********************************************************************
******
[x]                                                                  =
gamultiobj(@D2_Filter_Fitness,num_params,[],[],[],[],LB,UB,options
);


%
**********************************************************************
******
% Digital Filter Fitness function
```

```
%
*****************************************************************
******
function [score] = D2_Filter_Fitness(x)
global gn;
%
*****************************************************************
******
%
*****************************************************************
******
Kpo=x(1);                          % pickoff gain
m = x(2);                          % mass of proof mass in kg
b = x(3);                          % damping factor
k = x(4);                          % spring constant
Kfb = x(5);                        % feedback gain
Zz=0.75;                           % Compensator Zero in Z-domain
Pz=0;                              % Compensator pole in Z-domain
bw = 1024;                         % bandwidth
Kbst=0.5*(1-Pz)/(1-Zz);            % boost gain
Kq1=1;
Kq2=2;
Fs=25000000/190;
Ts=1/Fs;                            % sampling period
Finput=512;
%
*****************************************************************
******
% Digital Filter Calculation
%
*****************************************************************
******
T=2/Ts;
Z=tf([1 0],[1],Ts);
G=Kpo*1.6*Kbst*Kfb*Kq1*Kq2;
D2=((b*T+k+m*T^2)*Z^3+(k*2-m*T^2*2)*Z^2+(-
b*T+k+m*T^2)*Z)/((b*T+k+m*T^2+G)...
    *Z^3+(k*2-m*T^2*2+(2*G-G*Zz))*Z^2+(-b*T+k+m*T^2+(G-
2*G*Zz))*Z+(-G*Zz));
warning off all;
%
*****************************************************************
******
%Load offline data
%
*****************************************************************
******
load('MASH20NF_6MHz2Vpp_0P0_0Z75_0Kbst5_1Mb_20-Mar-
2014_22_33_26_S21_12V.mat')
f = sfi(0, 15, 12);
f.bin=error1;
OUT=double(f)';
P1=2*B1-1;
P2=filter(D2num{1,1},D2den{1,1},OUT);
P=P1-P2;
N=length(P1);
w=hann(N)';
f=Finput/Fs ;        % Normalized signal frequency
```

```
ptot=zeros(1,N);
fBL=N*(10/Fs);      % Lower limit Base-band frequency bins
fBH=N*(bw/Fs);
%
**************************************************************
******
% calculate the noise floor
%
**************************************************************
******
[~,ptot,~,~]=calcSNR(P,f,fBL,fBH,w,N);
score = mean(ptot(round(N*(10/Fs):round(N*(800/Fs)))));
%
**************************************************************
******
% plot the results
%
**************************************************************
******
figure(1)
subplot(3,1,1), plot(gn,m,'.b')
hold on;
subplot(3,1,2), plot(gn,b,'.r')
hold on;
subplot(3,1,3), plot(gn,k,'.k')
hold on;
figure(2)
subplot(3,1,1), plot(gn,Kpo,'.c')
hold on;
subplot(3,1,2), plot(gn,Kfb,'.g')
hold on;
subplot(3,1,3), plot(gn,score,'.m')
hold on;
gn = gn+1;
drawnow;
```

A.4 MASH22 Digital Filtering

Please refer to section 6.10 for flowchart illustration

```
tic;
clear all;
clc;
epsilon=8.845e-12;              % permitivity of free space
% name the file to be saved
SD='MASH';
S2='Sensor1.mat';
nD=4;
savedata=1; % save data ? 1 =Yes ,0 =No
SD2SD4Stop;
STOP=0;
%
**************************************************************
******
% Process the sensor values sensor1
```

```
%
**********************************************************************
******
m = 1.62e-06;                       % mass of proof mass in kg
b = 1.2e-3;                       % damping factor
k = 93.2;                       % spring constant
AA = 2.5e-06;                     % Overlabed area between plates
d0 = 7.24e-06;                      % nominal gap

%
**********************************************************************
******
% Process the system parameters
%
**********************************************************************
******
bw = 1024;                        % bandwidth
osr =64;                      % oversampling ratio
Fs = osr*2*bw;                      % sampling frequency
Fs=125000;
Ts=1/Fs;                          % sampling period

%
**********************************************************************
******
% Define the simulation setup
%
**********************************************************************
******
N = bw*1024*nD;                   % calculate the number of samples
nMb=N/1024/1024;
Ntransient=0;
simt = (N+Ntransient)/Fs;                   % calculate simulation
time

%
**********************************************************************
******
% Define the input
%
**********************************************************************
******
sf = 2;                           %input as fraction of bw
Finput = bw/sf;                   % input frequency
AmpScale = 0.7;


%
**********************************************************************
******
% Define the compensator
%
**********************************************************************
******
%%%%%%%%%%%%%%%%%%%%%%%%%%%%
Fph = 3.7e+04;                    % freq
PhaseLag = 77.8;                  % phase
WL=Fph*2*pi;
```

```
PhiMargin=PhaseLag*(pi/180);
alpha=(1+sin(PhiMargin))/(1-sin(PhiMargin));
zero=WL/sqrt(alpha);
pole=alpha*zero;
Kcomp=pole/zero;
Comp=Kcomp*tf([1 zero],[1 pole]);
[Compn,Compd] = tfdata(Comp);
Cfs=10e-12;
Cis=Kcomp*Cfs;
Ris=1/(Cis*zero);
Rfs=1/(Cfs*pole);
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
Ri=1e3;
Cf=5.6e-9; %8e-9;
Rf=340e3;
% Interface Gains
KR=0.37;
KS =1;                %KS
K2 =0.7;                %K2
% LowPassFilter
LPF=tf(1,[1e3*1e-9 1]);
LPF5th=LPF*LPF*LPF*LPF*LPF;
[LPFn,LPFd] = tfdata(LPF5th);
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%
%
*****************************************************************
******
%
*****************************************************************
******
% Digital Filters Design
%
*****************************************************************
******
Kpo=6e+07;
enoise = 6e-9;
Vfeedback =12;                %Feedback Voltage
Lfactor =Vfeedback/d0;
Kfb=0.5*epsilon*AA*(Vfeedback^2)/(d0^2);
Kq1=1;
Kq2=0.4;
Kq2=1;
G1=1;
G2=2;
sysC=tf(1, [m b k]);
MzCz=c2d(sysC*Comp,Ts,'matched');
Integ=tf(Rf/Ri,[Rf*Cf 1]);
A=c2d(Integ,Ts,'matched');
B=c2d(Integ,Ts,'matched');

D1=tf(1,[1 0 0],Ts);
[D1num,D1den] = tfdata(D1);

D2=minreal((1/(B*A)+(G1*Kq2)+(G2*Kq2/A))*D1/(Kq2*K2*(KS+MzCz*KR*Kp
o*Kfb)));
[D2num,D2den] = tfdata(D2);
```

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

bitstream1=zeros(1,N);
bitstream2=zeros(1,N);
% load B1B2_110dB
ptot=zeros(1,N);
Ntransient=0;
serial1=serial('COM6', 'BaudRate', 500000, 'InputBufferSize',
N*1.2);        %setup the COM port
fopen(serial1);
Frame=N/4; % number of reqired frames to fulfill N
j=0;
 figure(1);
while(~STOP && toc < 600),
    while(serial1.BytesAvailable < Frame && toc < 600)
    end
  clear buffer;
  buffer = fread(serial1, serial1.BytesAvailable);
  inputstream(1:Frame) = buffer(length(buffer)-
Frame+1:length(buffer)); % grab the reqired Frames
  Bindex=1;
  index=1;
    while(Bindex <= Frame)
        bitstream1(index) = bitget(inputstream(Bindex),7);
        bitstream1(index+1) = bitget(inputstream(Bindex),5);
        bitstream1(index+2) = bitget(inputstream(Bindex),3);
        bitstream1(index+3) = bitget(inputstream(Bindex),1);
        bitstream2(index) = bitget(inputstream(Bindex),8);
        bitstream2(index+1) = bitget(inputstream(Bindex),6);
        bitstream2(index+2) = bitget(inputstream(Bindex),4);
        bitstream2(index+3) = bitget(inputstream(Bindex),2);
        index=index+4;
        Bindex=Bindex+1;
    end
    if (savedata==1)
    timee=clock;
    dayy=date; % save the file with time stamp
    s1=sprintf('%s_%1.2gMB_%s_%d_%d_%d_%d',SD,nMb, dayy, timee(4),
timee(5), round(timee(6)));
    s1=strcat(s1,S2);
    save(s1, 'bitstream1','bitstream2');
    end
Vref=1;
w=hann(N)';
f=Finput/Fs ;        % Normalized signal frequency
fB=N*(bw/Fs);        % Base-band frequency bins
yy1=zeros(1,N);
yy1=2*bitstream1-1;
ptot=zeros(1,N);
fBL=N*(10/Fs);       % Lower limit Base-band frequency bins
fBH=fB;     % Higher limit Base-band frequency bins
[snr,ptot,ps,pn]=calcSNR(yy1(1:N),f,fBL,fBH,w,N);
figure(1);
semilogx(linspace(1,Fs/2,N/2), ptot(1:N/2), 'b');
title('Amplitude of a SD2 Sigma-Delta Accelerometer')
xlabel('Frequency [Hz]')
```

```
ylabel('Amplitude [dB]')
text_handle = text(2,-45, sprintf('SD2 SNR = %4.1fdB ',snr));
grid on
axis([0 Fs/2 -180 0]);
drawnow;

%%%%%%%%%%%%% RealTime Digital Filtering for MASH %%%%%%%%

P1=-filter(D1num{1,1},D1den{1,1},2*bitstream1-1);
P2=filter(D2num{1,1},D2den{1,1},2*bitstream2-1);
P=P1-P2;
%%%%%%%%%%%%%% plot the spectrum for MASH %%%%%%%%%%%%%%%%
[snr,ptot,ps,pn]=calcSNR(P(1:N),f,fBL,fBH,w,N);
figure(3);
semilogx(linspace(1,Fs/2,N/2), ptot(1:N/2), 'r');
axis([0 Fs/2 -180 0]);
grid on
title('Amplitude of a MASH Sigma-Delta Accelerometer')
xlabel('Frequency [Hz]')
ylabel('Amplitude [dB]')
text_handle = text(2,-45, sprintf('MASH_2_2 SNR = %4.1fdB ',snr));
axis([0 Fs/2 -180 0]);
drawnow;
end
fclose(serial1);
```

Appendix B VHDL Code

B.1 MASH22 CPLD Code

```
LIBRARY ieee;
USE ieee.std_logic_1164.all;
ENTITY CPLDSW IS
  PORT (
 OUT1          :  IN  STD_LOGIC;     -- Bitstream from the first
Modulator
OUT2           :  IN  STD_LOGIC;     -- Bitstream from the Second
Modulator
Gclk1          : IN STD_LOGIC;    -- Main Clock
LE         : OUT STD_LOGIC;  -- LE
SCLK       : OUT STD_LOGIC;  -- Sample clock to the FFs
SCLK2uC     : OUT STD_LOGIC;    -- Sample clock to the uC
OUT_1       : OUT STD_LOGIC;    -- First Modulator Bitstream OUT to
the uC
OUT_2       : OUT STD_LOGIC;    --Second Modulator Bitstream OUT to
the uC
LED         : OUT STD_LOGIC;    -- OK LED
S1                : OUT STD_LOGIC;  -- FVB to bottom plate
S2                 : OUT STD_LOGIC;     -- GND to bottom plate
S3                  : OUT STD_LOGIC;    -- GND to top plate
S4            : OUT STD_LOGIC   -- FVB to top plate
  );
END CPLDSW;
architecture RTL of CPLDSW is
SIGNAL CTR : INTEGER RANGE 0 TO 15 := 0;
SIGNAL FBQT: STD_LOGIC := '1';
SIGNAL iSCLK: STD_LOGIC := '1';
begin
CTRL  :  process  (Gclk1)  --  Divid  the  incoming  main  clock  and
generate the sampling clock
begin
 IF rising_edge(Gclk1) THEN
           CTR <= CTR  + 1;
        CASE CTR IS
          WHEN 3 => iSCLK <= '0';
              WHEN 5 => FBQT <= '0';
              WHEN 7 => iSCLK <= '1';
                            CTR <= 0;
                            FBQT <= '1';
          WHEN OTHERS => null;
        END CASE;
 END IF;
end process;
SCLK<=iSCLK;
LE<=iSCLK;
SCLK2uC<=iSCLK;
--  switching  logic  based  on  sign  of  the  bitstream  and  the
feedback/quite phase signal
S1 <= (FBQT and OUT1) or (not FBQT);
S2 <= (FBQT and (not OUT1));
S3 <= (FBQT and OUT1);
```

```
S4 <= (FBQT and (not OUT1)) or (not FBQT);
OUT_1 <= OUT1;
OUT_2 <= OUT2;
LED <= '0';
end RTL;
```

B.2 MASH20 FPGA Code

```
library IEEE;
use IEEE.STD_LOGIC_1164.ALL;
use IEEE.STD_LOGIC_UNSIGNED.ALL;
use IEEE.STD_LOGIC_ARITH.ALL;
entity HK is
    Port (
                CLK_25MHZ : in  STD_LOGIC;
                ADC_BUSY : in  STD_LOGIC;
                ADC_CONV : OUT  STD_LOGIC;
                ADC_CS : OUT  STD_LOGIC;
                S1 : OUT  STD_LOGIC;
                S2 : OUT  STD_LOGIC;
                DIN  :  IN  std_logic_vector(11  downto  0);  --
sfix12_En10
                MMCX_OUT : OUT  STD_LOGIC;
                DOUT : OUT std_logic_vector(15 downto 0);
                RB0_OUT_CLK : OUT  STD_LOGIC;
                C7Led : inout  STD_LOGIC;
           C5Led : inout  STD_LOGIC);
end HK;
architecture Behavioral of HK is
  -- Component Declarations
COMPONENT MASH20
  PORT( clk                                 :   IN    std_logic;
        reset                               :   IN    std_logic;
        clk_enable                          :   IN    std_logic;
        In1                              :   IN    std_logic_vector(11
DOWNTO 0);  -- sfix12_En9
        ce_out                              :   OUT   std_logic;
        Out1                                :   OUT   std_logic;
        Out2                             :   OUT   std_logic_vector(15
DOWNTO 0)  -- sfix16_En14
        );
  END COMPONENT;
   -- Component Configuration Statements
  FOR ALL : MASH20
    USE ENTITY work.MASH20(rtl);
SIGNAL CTR0 : INTEGER RANGE 0 TO 50000005 := 0;
SIGNAL CTR1 : INTEGER RANGE 0 TO 200 := 0;
SIGNAL CTR2 : INTEGER RANGE 0 TO 200 := 0;
SIGNAL ADCData  : std_logic_vector(11 DOWNTO 0);  -- sfix12_En10
SIGNAL SCLK : std_logic;
SIGNAL reset : std_logic:='0';
SIGNAL clk_enable : std_logic:='1';
SIGNAL ce_out : std_logic;
SIGNAL SD2OUT : std_logic;
```

```
SIGNAL MASHOut: std_logic_vector(15 DOWNTO 0);
begin
  u_MASH20 : MASH20
    PORT MAP( clk => SCLK,
              reset => reset,
              clk_enable => clk_enable,
                   In1 => ADCData,  -- sfix12_En10
                   ce_out => ce_out,
                   Out1 => SD2OUT,
              Out2 => MASHOut
                   );
ADC_CS<='0';

LED_Flash : process (CLK_25MHZ)
begin
 IF rising_edge(CLK_25MHZ) THEN
           CTR0 <= CTR0  + 1;
         CASE CTR0 IS
          WHEN 25000000 => C7Led <= '0'; C5Led <= '1';
             WHEN 50000000 => C7Led <= '1'; C5Led <= '0' ;  CTR0
<= 0;
          WHEN OTHERS => null;
        END CASE;
 END IF;
end process;

ADC_Read : process (CLK_25MHZ)
begin
 IF rising_edge(CLK_25MHZ) THEN
           CTR1 <= CTR1  + 1;
         CASE CTR1 IS
          WHEN 3 => ADC_CONV <= '1';
              WHEN 5 => ADCData<=DIN;
             WHEN 9 => ADC_CONV <= '0'; CTR1 <= 0;
          WHEN OTHERS => null;
        END CASE;
 END IF;
end process;

SCLK_GEN : process (CLK_25MHZ)
begin
 IF rising_edge(CLK_25MHZ) THEN
        CTR2 <= CTR2  + 1;
        CASE CTR2 IS
             WHEN 95=>
                    SCLK <='1';
                    RB0_OUT_CLK<='1';
                    MMCX_OUT<=SD2OUT;
                    S1<=  SD2OUT;
                    S2<=  not SD2OUT;
                    DOUT <= MASHOut;
            WHEN 190=> CTR2 <= 0;SCLK <='0'; RB0_OUT_CLK<='0';
          WHEN OTHERS => null;
        END CASE;
 END IF;
end process;
```

```
end Behavioral;


-- -------------------------------------------------------------
-- Module: Compensator
-- -------------------------------------------------------------
LIBRARY IEEE;
USE IEEE.std_logic_1164.ALL;
USE IEEE.numeric_std.ALL;
ENTITY Compensator IS
  PORT( clk                               :   IN    std_logic;
        reset                             :   IN    std_logic;
        enb                               :   IN    std_logic;
        In1                         :   IN    std_logic_vector(11
DOWNTO 0);  -- sfix12_En9
        Out1                        :   OUT   std_logic_vector(31
DOWNTO 0)  -- sfix32_En29
        );
END Compensator;
ARCHITECTURE rtl OF Compensator IS
  -- Signals
  SIGNAL In1_signed                       : signed(11 DOWNTO 0);
-- sfix12_En9
  SIGNAL Kbst_mul_temp                    : signed(43 DOWNTO 0);
-- sfix44_En35
  SIGNAL Kbst_out1                        : signed(31 DOWNTO 0);
-- sfix32_En24
  SIGNAL Unit_Delay_out1                  : signed(11 DOWNTO 0);
-- sfix12_En9
  SIGNAL ZxKbst_mul_temp                  : signed(43 DOWNTO 0);
-- sfix44_En35
  SIGNAL ZxKbst_out1                      : signed(31 DOWNTO 0);
-- sfix32_En24
  SIGNAL Sum_out1                         : signed(31 DOWNTO 0);
-- sfix32_En29
  SIGNAL Unit_Delay1_out1                 : signed(31 DOWNTO 0);
-- sfix32_En29
  SIGNAL P_mul_temp                       : signed(63 DOWNTO 0);
-- sfix64_En64
  SIGNAL P_out1                           : signed(31 DOWNTO 0);
-- sfix32_En33
  SIGNAL Sum9_sub_cast                    : signed(31 DOWNTO 0);
-- sfix32_En24
  SIGNAL Sum9_out1                        : signed(31 DOWNTO 0);
-- sfix32_En24
  SIGNAL Sum_add_cast                     : signed(32 DOWNTO 0);
-- sfix33_En24
  SIGNAL Sum_add_cast_1                   : signed(32 DOWNTO 0);
-- sfix33_En24
  SIGNAL Sum_add_temp                     : signed(32 DOWNTO 0);
-- sfix33_En24
  SIGNAL Sum_cast                         : signed(31 DOWNTO 0);
-- sfix32_En24

BEGIN
  -- Kbst=1-P/1-Z
```

```
   -- CompZ=Kbst*(z-Z)/(z-P)
  In1_signed <= signed(In1);
  Kbst_mul_temp <= to_signed(1404342989, 32) * In1_signed;

  Kbst_out1    <=    "01111111111111111111111111111111"    WHEN
(Kbst_mul_temp(43) = '0') AND (Kbst_mul_temp(42) /= '0') ELSE
      "10000000000000000000000000000000" WHEN (Kbst_mul_temp(43)
= '1') AND (Kbst_mul_temp(42) /= '1') ELSE
      Kbst_mul_temp(42 DOWNTO 11);

  Unit_Delay_process : PROCESS (clk, reset)
  BEGIN
    IF reset = '1' THEN
      Unit_Delay_out1 <= to_signed(0, 12);
    ELSIF clk'EVENT AND clk = '1' THEN
      IF enb = '1' THEN
        Unit_Delay_out1 <= In1_signed;
      END IF;
    END IF;
  END PROCESS Unit_Delay_process;

  ZxKbst_mul_temp <= to_signed(1339918479, 32) * Unit_Delay_out1;

  ZxKbst_out1    <=    "01111111111111111111111111111111"    WHEN
(ZxKbst_mul_temp(43) = '0') AND (ZxKbst_mul_temp(42) /= '0') ELSE
      "10000000000000000000000000000000"                    WHEN
(ZxKbst_mul_temp(43) = '1') AND (ZxKbst_mul_temp(42) /= '1') ELSE
      ZxKbst_mul_temp(42 DOWNTO 11);

  Unit_Delay1_process : PROCESS (clk, reset)
  BEGIN
    IF reset = '1' THEN
      Unit_Delay1_out1 <= to_signed(0, 32);
    ELSIF clk'EVENT AND clk = '1' THEN
      IF enb = '1' THEN
        Unit_Delay1_out1 <= Sum_out1;
      END IF;
    END IF;
  END PROCESS Unit_Delay1_process;

  P_mul_temp <= 1374389535 * Unit_Delay1_out1;

  P_out1     <=     "01111111111111111111111111111111"     WHEN
(P_mul_temp(63) = '0') AND (P_mul_temp(62) /= '0') ELSE
      "10000000000000000000000000000000" WHEN (P_mul_temp(63) =
'1') AND (P_mul_temp(62) /= '1') ELSE
      P_mul_temp(62 DOWNTO 31);

  Sum9_sub_cast <= resize(P_out1(31 DOWNTO 9), 32);
  Sum9_out1 <= Sum9_sub_cast - ZxKbst_out1;
  Sum_add_cast <= resize(Kbst_out1, 33);
  Sum_add_cast_1 <= resize(Sum9_out1, 33);
  Sum_add_temp <= Sum_add_cast + Sum_add_cast_1;

  Sum_cast     <=     "01111111111111111111111111111111"     WHEN
(Sum_add_temp(32) = '0') AND (Sum_add_temp(31) /= '0') ELSE
```

```
      "10000000000000000000000000000000" WHEN (Sum_add_temp(32) =
'1') AND (Sum_add_temp(31) /= '1') ELSE
      Sum_add_temp(31 DOWNTO 0);
    Sum_out1    <=    "01111111111111111111111111111111"    WHEN
(Sum_cast(31) = '0') AND (Sum_cast(30 DOWNTO 26) /= "00000") ELSE
      "10000000000000000000000000000000"  WHEN  (Sum_cast(31)  =
'1') AND (Sum_cast(30 DOWNTO 26) /= "11111") ELSE
      Sum_cast(26 DOWNTO 0) & '0' & '0' & '0' & '0' & '0';
  Out1 <= std_logic_vector(Sum_out1);

END rtl;


-- -------------------------------------------------------------
-- Module: MASH20
-- -------------------------------------------------
LIBRARY IEEE;
USE IEEE.std_logic_1164.ALL;
USE IEEE.numeric_std.ALL;
ENTITY MASH20 IS
  PORT( clk                               :   IN    std_logic;
        reset                             :   IN    std_logic;
        clk_enable                        :   IN    std_logic;
        In1                           :   IN    std_logic_vector(11
DOWNTO 0);  -- sfix12_En9
        ce_out                            :   OUT   std_logic;
        Out1                              :   OUT   std_logic;
        Out2                          :   OUT   std_logic_vector(15
DOWNTO 0)  -- sfix16_En13
        );
END MASH20;
ARCHITECTURE rtl OF MASH20 IS
  -- Component Declarations
  COMPONENT Compensator
    PORT( clk                             :   IN    std_logic;
          reset                           :   IN    std_logic;
          enb                             :   IN    std_logic;
          In1                         :   IN    std_logic_vector(11
DOWNTO 0);  -- sfix12_En9
          Out1                        :   OUT   std_logic_vector(31
DOWNTO 0)  -- sfix32_En29
          );
  END COMPONENT;
  -- Component Configuration Statements
  FOR ALL : Compensator
    USE ENTITY work.Compensator(rtl);
  -- Signals
  SIGNAL enb                              : std_logic;
  SIGNAL Compensator_out1                  : std_logic_vector(31
DOWNTO 0);  -- ufix32
  SIGNAL Compensator_out1_signed          : signed(31 DOWNTO 0);
-- sfix32_En29
  SIGNAL Sign_out1                        : signed(31 DOWNTO 0);
-- int32
  SIGNAL Compare_To_Zero_out1             : std_logic;
```

```
  SIGNAL Sum_sub_cast                     : signed(31 DOWNTO 0);
-- sfix32_En29
  SIGNAL Sum_sub_cast_1                   : signed(32 DOWNTO 0);
-- sfix33_En29
  SIGNAL Sum_sub_cast_2                   : signed(32 DOWNTO 0);
-- sfix33_En29
  SIGNAL Sum_sub_temp                     : signed(32 DOWNTO 0);
-- sfix33_En29
  SIGNAL Sum_out1                         : signed(31 DOWNTO 0);
-- sfix32_En29
  SIGNAL Unit_Delay2_out1                 : signed(23 DOWNTO 0);
-- sfix24_En16
  SIGNAL Gain7_mul_temp                   : signed(47 DOWNTO 0);
-- sfix48_En38
  SIGNAL Gain7_out1                       : signed(23 DOWNTO 0);
-- sfix24_En16
  SIGNAL Unit_Delay3_out1                 : signed(23 DOWNTO 0);
-- sfix24_En16
  SIGNAL Gain8_mul_temp                   : signed(47 DOWNTO 0);
-- sfix48_En38
  SIGNAL Gain8_out1                       : signed(23 DOWNTO 0);
-- sfix24_En16
  SIGNAL Sum1_out1                        : signed(23 DOWNTO 0);
-- sfix24_En16
  SIGNAL Unit_Delay4_out1                 : signed(23 DOWNTO 0);
-- sfix24_En16
  SIGNAL Gain9_mul_temp                   : signed(47 DOWNTO 0);
-- sfix48_En39
  SIGNAL Gain9_out1                       : signed(23 DOWNTO 0);
-- sfix24_En16
  SIGNAL Sum4_add_cast                    : signed(24 DOWNTO 0);
-- sfix25_En16
  SIGNAL Sum4_add_cast_1                  : signed(24 DOWNTO 0);
-- sfix25_En16
  SIGNAL Sum4_add_temp                    : signed(24 DOWNTO 0);
-- sfix25_En16
  SIGNAL Sum4_out1                        : signed(23 DOWNTO 0);
-- sfix24_En16
  SIGNAL Sum2_add_cast                    : signed(24 DOWNTO 0);
-- sfix25_En16
  SIGNAL Sum2_add_cast_1                  : signed(24 DOWNTO 0);
-- sfix25_En16
  SIGNAL Sum2_add_temp                    : signed(24 DOWNTO 0);
-- sfix25_En16
  SIGNAL Sum2_out1                        : signed(23 DOWNTO 0);
-- sfix24_En16
  SIGNAL Sum1_add_cast                    : signed(23 DOWNTO 0);
-- sfix24_En16
  SIGNAL Sum1_add_cast_1                  : signed(24 DOWNTO 0);
-- sfix25_En16
  SIGNAL Sum1_add_cast_2                  : signed(24 DOWNTO 0);
-- sfix25_En16
  SIGNAL Sum1_add_temp                    : signed(24 DOWNTO 0);
-- sfix25_En16
  SIGNAL Gain5_mul_temp                   : signed(47 DOWNTO 0);
-- sfix48_En38
```

```
  SIGNAL Gain5_out1                       : signed(23 DOWNTO 0);
-- sfix24_En16
  SIGNAL Gain3_cast                       : signed(47 DOWNTO 0);
-- sfix48_En38
  SIGNAL Gain3_out1                       : signed(23 DOWNTO 0);
-- sfix24_En16
  SIGNAL Gain4_mul_temp                   : signed(47 DOWNTO 0);
-- sfix48_En37
  SIGNAL Gain4_out1                       : signed(23 DOWNTO 0);
-- sfix24_En15
  SIGNAL Gain6_mul_temp                   : signed(47 DOWNTO 0);
-- sfix48_En43
  SIGNAL Gain6_out1                       : signed(23 DOWNTO 0);
-- sfix24_En20
  SIGNAL Sum6_add_cast                    : signed(24 DOWNTO 0);
-- sfix25_En16
  SIGNAL Sum6_add_cast_1                  : signed(23 DOWNTO 0);
-- sfix24_En16
  SIGNAL Sum6_add_cast_2                  : signed(24 DOWNTO 0);
-- sfix25_En16
  SIGNAL Sum6_add_temp                    : signed(24 DOWNTO 0);
-- sfix25_En16
  SIGNAL Sum6_out1                        : signed(23 DOWNTO 0);
-- sfix24_En16
  SIGNAL Sum5_add_cast                    : signed(24 DOWNTO 0);
-- sfix25_En15
  SIGNAL Sum5_add_cast_1                  : signed(23 DOWNTO 0);
-- sfix24_En15
  SIGNAL Sum5_add_cast_2                  : signed(24 DOWNTO 0);
-- sfix25_En15
  SIGNAL Sum5_add_temp                    : signed(24 DOWNTO 0);
-- sfix25_En15
  SIGNAL Sum5_out1                        : signed(23 DOWNTO 0);
-- sfix24_En15
  SIGNAL Sum3_add_cast                    : signed(23 DOWNTO 0);
-- sfix24_En15
  SIGNAL Sum3_add_cast_1                  : signed(24 DOWNTO 0);
-- sfix25_En15
  SIGNAL Sum3_add_cast_2                  : signed(24 DOWNTO 0);
-- sfix25_En15
  SIGNAL Sum3_add_temp                    : signed(24 DOWNTO 0);
-- sfix25_En15
  SIGNAL Sum3_cast                        : signed(23 DOWNTO 0);
-- sfix24_En15
  SIGNAL Sum3_out1                        : signed(23 DOWNTO 0);
-- sfix24_En16
  SIGNAL Sum8_sub_cast                    : signed(31 DOWNTO 0);
-- sfix32_En26
  SIGNAL Sum8_sub_cast_1                  : signed(32 DOWNTO 0);
-- sfix33_En26
  SIGNAL Sum8_sub_cast_2                  : signed(31 DOWNTO 0);
-- sfix32_En26
  SIGNAL Sum8_sub_cast_3                  : signed(32 DOWNTO 0);
-- sfix33_En26
  SIGNAL Sum8_sub_temp                    : signed(32 DOWNTO 0);
-- sfix33_En26
```

```
  SIGNAL Sum8_cast                        : signed(31 DOWNTO 0);
-- sfix32_En26
  SIGNAL Sum8_out1                        : signed(15 DOWNTO 0);
-- sfix16_En13

BEGIN
  u_Compensator : Compensator
    PORT MAP( clk => clk,
              reset => reset,
              enb => clk_enable,
              In1 => In1,  -- sfix12_En9
              Out1 => Compensator_out1  -- sfix32_En29
              );

  Compensator_out1_signed <= signed(Compensator_out1);


  Sign_out1 <= to_signed(1, 32) WHEN Compensator_out1_signed > 0
ELSE
      to_signed(-1, 32) WHEN Compensator_out1_signed < 0 ELSE
      to_signed(0, 32);


  Compare_To_Zero_out1 <= '1' WHEN Sign_out1 >= 0 ELSE
      '0';

  Sum_sub_cast    <=    "01111111111111111111111111111111"    WHEN
(Sign_out1(31)   =   '0')   AND   (Sign_out1(30   DOWNTO   2)   /=
"00000000000000000000000000000") ELSE
      "10000000000000000000000000000000" WHEN (Sign_out1(31) =
'1')       AND       (Sign_out1(30       DOWNTO       2)       /=
"11111111111111111111111111111") ELSE
      Sign_out1(2 DOWNTO 0) & '0' & '0' & '0' & '0' & '0' & '0' &
'0' & '0' & '0' & '0' & '0' & '0' & '0' & '0' & '0' & '0' & '0' &
'0' & '0' & '0' & '0' & '0' & '0' & '0' & '0' & '0' & '0' & '0' &
'0';
  Sum_sub_cast_1 <= resize(Sum_sub_cast, 33);
  Sum_sub_cast_2 <= resize(Compensator_out1_signed, 33);
  Sum_sub_temp <= Sum_sub_cast_1 - Sum_sub_cast_2;

  Sum_out1     <=     "01111111111111111111111111111111"     WHEN
(Sum_sub_temp(32) = '0') AND (Sum_sub_temp(31) /= '0') ELSE
      "10000000000000000000000000000000" WHEN (Sum_sub_temp(32) =
'1') AND (Sum_sub_temp(31) /= '1') ELSE
      Sum_sub_temp(31 DOWNTO 0);

  enb <= clk_enable;

  Gain7_mul_temp <= 5052854 * Unit_Delay2_out1;

  Gain7_out1       <=       "01111111111111111111111"       WHEN
(Gain7_mul_temp(47) = '0') AND (Gain7_mul_temp(46 DOWNTO 45) /=
"00") ELSE
      "10000000000000000000000" WHEN (Gain7_mul_temp(47) = '1')
AND (Gain7_mul_temp(46 DOWNTO 45) /= "11") ELSE
      Gain7_mul_temp(45 DOWNTO 22);
```

```
  Gain8_mul_temp <= (-4633609) * Unit_Delay3_out1;

  Gain8_out1       <=       "01111111111111111111111"       WHEN
(Gain8_mul_temp(47) = '0') AND (Gain8_mul_temp(46 DOWNTO 45) /=
"00") ELSE
      "10000000000000000000000" WHEN (Gain8_mul_temp(47) = '1')
AND (Gain8_mul_temp(46 DOWNTO 45) /= "11") ELSE
      Gain8_mul_temp(45 DOWNTO 22);

  Unit_Delay2_process : PROCESS (clk, reset)
  BEGIN
    IF reset = '1' THEN
      Unit_Delay2_out1 <= to_signed(0, 24);
    ELSIF clk'EVENT AND clk = '1' THEN
      IF enb = '1' THEN
        Unit_Delay2_out1 <= Sum1_out1;
      END IF;
    END IF;
  END PROCESS Unit_Delay2_process;


  Unit_Delay3_process : PROCESS (clk, reset)
  BEGIN
    IF reset = '1' THEN
      Unit_Delay3_out1 <= to_signed(0, 24);
    ELSIF clk'EVENT AND clk = '1' THEN
      IF enb = '1' THEN
        Unit_Delay3_out1 <= Unit_Delay2_out1;
      END IF;
    END IF;
  END PROCESS Unit_Delay3_process;


  Unit_Delay4_process : PROCESS (clk, reset)
  BEGIN
    IF reset = '1' THEN
      Unit_Delay4_out1 <= to_signed(0, 24);
    ELSIF clk'EVENT AND clk = '1' THEN
      IF enb = '1' THEN
        Unit_Delay4_out1 <= Unit_Delay3_out1;
      END IF;
    END IF;
  END PROCESS Unit_Delay4_process;


  Gain9_mul_temp <= 6895190 * Unit_Delay4_out1;

  Gain9_out1       <=       "01111111111111111111111"       WHEN
(Gain9_mul_temp(47) = '0') AND (Gain9_mul_temp(46) /= '0') ELSE
      "10000000000000000000000" WHEN (Gain9_mul_temp(47) = '1')
AND (Gain9_mul_temp(46) /= '1') ELSE
      Gain9_mul_temp(46 DOWNTO 23);

  Sum4_add_cast <= resize(Gain9_out1, 25);
  Sum4_add_cast_1 <= resize(Gain8_out1, 25);
```

```
  Sum4_add_temp <= Sum4_add_cast + Sum4_add_cast_1;

  Sum4_out1 <= "01111111111111111111111" WHEN (Sum4_add_temp(24)
= '0') AND (Sum4_add_temp(23) /= '0') ELSE
      "10000000000000000000000" WHEN (Sum4_add_temp(24) = '1')
AND (Sum4_add_temp(23) /= '1') ELSE
      Sum4_add_temp(23 DOWNTO 0);

  Sum2_add_cast <= resize(Sum4_out1, 25);
  Sum2_add_cast_1 <= resize(Gain7_out1, 25);
  Sum2_add_temp <= Sum2_add_cast + Sum2_add_cast_1;

  Sum2_out1 <= "01111111111111111111111" WHEN (Sum2_add_temp(24)
= '0') AND (Sum2_add_temp(23) /= '0') ELSE
      "10000000000000000000000" WHEN (Sum2_add_temp(24) = '1')
AND (Sum2_add_temp(23) /= '1') ELSE
      Sum2_add_temp(23 DOWNTO 0);

  Sum1_add_cast <= resize(Sum_out1(31 DOWNTO 13), 24);
  Sum1_add_cast_1 <= resize(Sum1_add_cast, 25);
  Sum1_add_cast_2 <= resize(Sum2_out1, 25);
  Sum1_add_temp <= Sum1_add_cast_1 + Sum1_add_cast_2;

  Sum1_out1 <= "01111111111111111111111" WHEN (Sum1_add_temp(24)
= '0') AND (Sum1_add_temp(23) /= '0') ELSE
      "10000000000000000000000" WHEN (Sum1_add_temp(24) = '1')
AND (Sum1_add_temp(23) /= '1') ELSE
      Sum1_add_temp(23 DOWNTO 0);

  Gain5_mul_temp <= 4486624 * Unit_Delay3_out1;

  Gain5_out1       <=       "01111111111111111111111"       WHEN
(Gain5_mul_temp(47) = '0') AND (Gain5_mul_temp(46 DOWNTO 45) /=
"00") ELSE
      "10000000000000000000000" WHEN (Gain5_mul_temp(47) = '1')
AND (Gain5_mul_temp(46 DOWNTO 45) /= "11") ELSE
      Gain5_mul_temp(45 DOWNTO 22);

  Gain3_cast <= resize(Sum1_out1 & '0' & '0' & '0' & '0' & '0' &
'0' & '0' & '0' & '0' & '0' & '0' & '0' & '0' & '0' & '0' & '0' &
'0' & '0' & '0' & '0' & '0', 48);

  Gain3_out1 <= "01111111111111111111111" WHEN (Gain3_cast(47) =
'0') AND (Gain3_cast(46 DOWNTO 45) /= "00") ELSE
      "10000000000000000000000" WHEN (Gain3_cast(47) = '1') AND
(Gain3_cast(46 DOWNTO 45) /= "11") ELSE
      Gain3_cast(45 DOWNTO 22);

  Gain4_mul_temp <= (-4251703) * Unit_Delay2_out1;

  Gain4_out1       <=       "01111111111111111111111"       WHEN
(Gain4_mul_temp(47) = '0') AND (Gain4_mul_temp(46 DOWNTO 45) /=
"00") ELSE
      "10000000000000000000000" WHEN (Gain4_mul_temp(47) = '1')
AND (Gain4_mul_temp(46 DOWNTO 45) /= "11") ELSE
      Gain4_mul_temp(45 DOWNTO 22);
```

```
  Gain6_mul_temp <= (-5316094) * Unit_Delay4_out1;

  Gain6_out1       <=       "011111111111111111111111"       WHEN
(Gain6_mul_temp(47) = '0') AND (Gain6_mul_temp(46) /= '0') ELSE
      "100000000000000000000000" WHEN (Gain6_mul_temp(47) = '1')
AND (Gain6_mul_temp(46) /= '1') ELSE
      Gain6_mul_temp(46 DOWNTO 23);

  Sum6_add_cast <= resize(Gain5_out1, 25);
  Sum6_add_cast_1 <= resize(Gain6_out1(23 DOWNTO 4), 24);
  Sum6_add_cast_2 <= resize(Sum6_add_cast_1, 25);
  Sum6_add_temp <= Sum6_add_cast + Sum6_add_cast_2;

  Sum6_out1 <= "011111111111111111111111" WHEN (Sum6_add_temp(24)
= '0') AND (Sum6_add_temp(23) /= '0') ELSE
      "100000000000000000000000" WHEN (Sum6_add_temp(24) = '1')
AND (Sum6_add_temp(23) /= '1') ELSE
      Sum6_add_temp(23 DOWNTO 0);

  Sum5_add_cast <= resize(Gain4_out1, 25);
  Sum5_add_cast_1 <= resize(Sum6_out1(23 DOWNTO 1), 24);
  Sum5_add_cast_2 <= resize(Sum5_add_cast_1, 25);
  Sum5_add_temp <= Sum5_add_cast + Sum5_add_cast_2;

  Sum5_out1 <= "011111111111111111111111" WHEN (Sum5_add_temp(24)
= '0') AND (Sum5_add_temp(23) /= '0') ELSE
      "100000000000000000000000" WHEN (Sum5_add_temp(24) = '1')
AND (Sum5_add_temp(23) /= '1') ELSE
      Sum5_add_temp(23 DOWNTO 0);

  Sum3_add_cast <= resize(Gain3_out1(23 DOWNTO 1), 24);
  Sum3_add_cast_1 <= resize(Sum3_add_cast, 25);
  Sum3_add_cast_2 <= resize(Sum5_out1, 25);
  Sum3_add_temp <= Sum3_add_cast_1 + Sum3_add_cast_2;

  Sum3_cast <= "011111111111111111111111" WHEN (Sum3_add_temp(24)
= '0') AND (Sum3_add_temp(23) /= '0') ELSE
      "100000000000000000000000" WHEN (Sum3_add_temp(24) = '1')
AND (Sum3_add_temp(23) /= '1') ELSE
      Sum3_add_temp(23 DOWNTO 0);

  Sum3_out1 <= "011111111111111111111111" WHEN (Sum3_cast(23) =
'0') AND (Sum3_cast(22) /= '0') ELSE
      "100000000000000000000000" WHEN (Sum3_cast(23) = '1') AND
(Sum3_cast(22) /= '1') ELSE
      Sum3_cast(22 DOWNTO 0) & '0';


  Sum8_sub_cast   <=   "01111111111111111111111111111111"   WHEN
(Sign_out1(31)   =   '0')   AND   (Sign_out1(30   DOWNTO   5)   /=
"00000000000000000000000000") ELSE
      "10000000000000000000000000000000" WHEN (Sign_out1(31) =
'1') AND (Sign_out1(30 DOWNTO 5) /= "11111111111111111111111111")
ELSE
```

```
      Sign_out1(5 DOWNTO 0) & '0' & '0' & '0' & '0' & '0' & '0' &
'0' & '0' & '0' & '0' & '0' & '0' & '0' & '0' & '0' & '0' & '0' &
'0' & '0' & '0' & '0' & '0' & '0' & '0' & '0';
  Sum8_sub_cast_1 <= resize(Sum8_sub_cast, 33);

  Sum8_sub_cast_2  <=  "01111111111111111111111111111111"  WHEN
(Sum3_out1(23) = '0') AND (Sum3_out1(22 DOWNTO 21) /= "00") ELSE
      "10000000000000000000000000000000"  WHEN  (Sum3_out1(23)  =
'1') AND (Sum3_out1(22 DOWNTO 21) /= "11") ELSE
      Sum3_out1(21 DOWNTO 0) & '0' & '0' & '0' & '0' & '0' & '0'
& '0' & '0' & '0' & '0';
  Sum8_sub_cast_3 <= resize(Sum8_sub_cast_2, 33);
  Sum8_sub_temp <= Sum8_sub_cast_1 - Sum8_sub_cast_3;

  Sum8_cast    <=    "01111111111111111111111111111111"    WHEN
(Sum8_sub_temp(32) = '0') AND (Sum8_sub_temp(31) /= '0') ELSE
      "10000000000000000000000000000000" WHEN (Sum8_sub_temp(32)
= '1') AND (Sum8_sub_temp(31) /= '1') ELSE
      Sum8_sub_temp(31 DOWNTO 0);

  Sum8_out1 <= "0111111111111111" WHEN (Sum8_cast(31) = '0') AND
(Sum8_cast(30 DOWNTO 28) /= "000") ELSE
      "1000000000000000"  WHEN  (Sum8_cast(31)  =  '1')  AND
(Sum8_cast(30 DOWNTO 28) /= "111") ELSE
      Sum8_cast(28 DOWNTO 13);

  Out2 <= std_logic_vector(Sum8_out1);

  ce_out <= clk_enable;
  Out1 <= Compare_To_Zero_out1;
END rtl;
```

Appendix C C Language Code

C.1 MASH22 PIC Code

```
#define        Clk                          PORTA,2
#define        Bit0                         PORTC,0
#define        Bit1                         PORTC,1

unsigned char BitStream,temp,BYTE,COUNT,ctr,SD2SD4;
//This interrupt routine is called  every rising edge of the CLK
//at RA2
   void interrupt()
   {
     if (UART1_Data_Ready() == 1) { // soft switch between
SD2/SD4
      SD2SD4 = UART1_Read();
      if  (SD2SD4 == 'A' ) {PORTC.B2=0;}
      if  (SD2SD4 == 'B' ) {PORTC.B2=1;}
     }
     ctr=ctr+1;
     temp=PORTC;                // Read BitStream from Port C
     temp= temp & 0x03;
     BitStream=temp | BitStream;
     if (ctr<4) {BitStream=BitStream<<2;}
     else{
          UART1_Write(BitStream);       // send data via UART
          ctr=0;
          BitStream=0x00;
         }
     INTCON.INTF=0;
  }

 void main() {
  INTCON = 0x00;            //  Disable all interrupts

  ANSELA  = 0;             // AN pins as digital
  ANSELC  = 0;

  C1ON_bit = 0;           // Disable comparators
  C2ON_bit = 0;

  TRISA2_bit = 1;         // set RA2 pin as input  (CLK)
  TRISC0_bit = 1;         // set RC0 pin as input  (Bitstream1)
  TRISC1_bit = 1;         // set RC1 pin as input  (Bitstream2)
  TRISC2_bit = 0;         // set RC2 pin as output  (LED)
  TRISC3_bit = 0;         // set RC3 pin as output  (LED)


  SCS0_bit=0;          // set CPU at 32Mhz
  SCS1_bit=0;

  IRCF0_bit=0;
  IRCF1_bit=1;
  IRCF2_bit=1;
  IRCF3_bit=1;
```

```
  SPLLEN_bit=1;

  PORTC.B2=0;                          // LED 1 ON
  PORTC.B3=0;                          // LED 2 ON

  ctr=0;
  COUNT=4;
  BYTE=4;

  UART1_Init(500000);
  Delay_ms(100);          // Delay for UART module to be
Initialized

  INTCON.GIE=1;           // Global interrupt enable
  INTCON.INTE=1;          // RA2/INT External Interrupt Enable
  OPTION_REG.INTEDG=1;    // Interrupt on rising edge of RA2/INT
  #pragma disablecontexsaving
while(1){}
}
```

C.2 MASH20 Ethernet Code

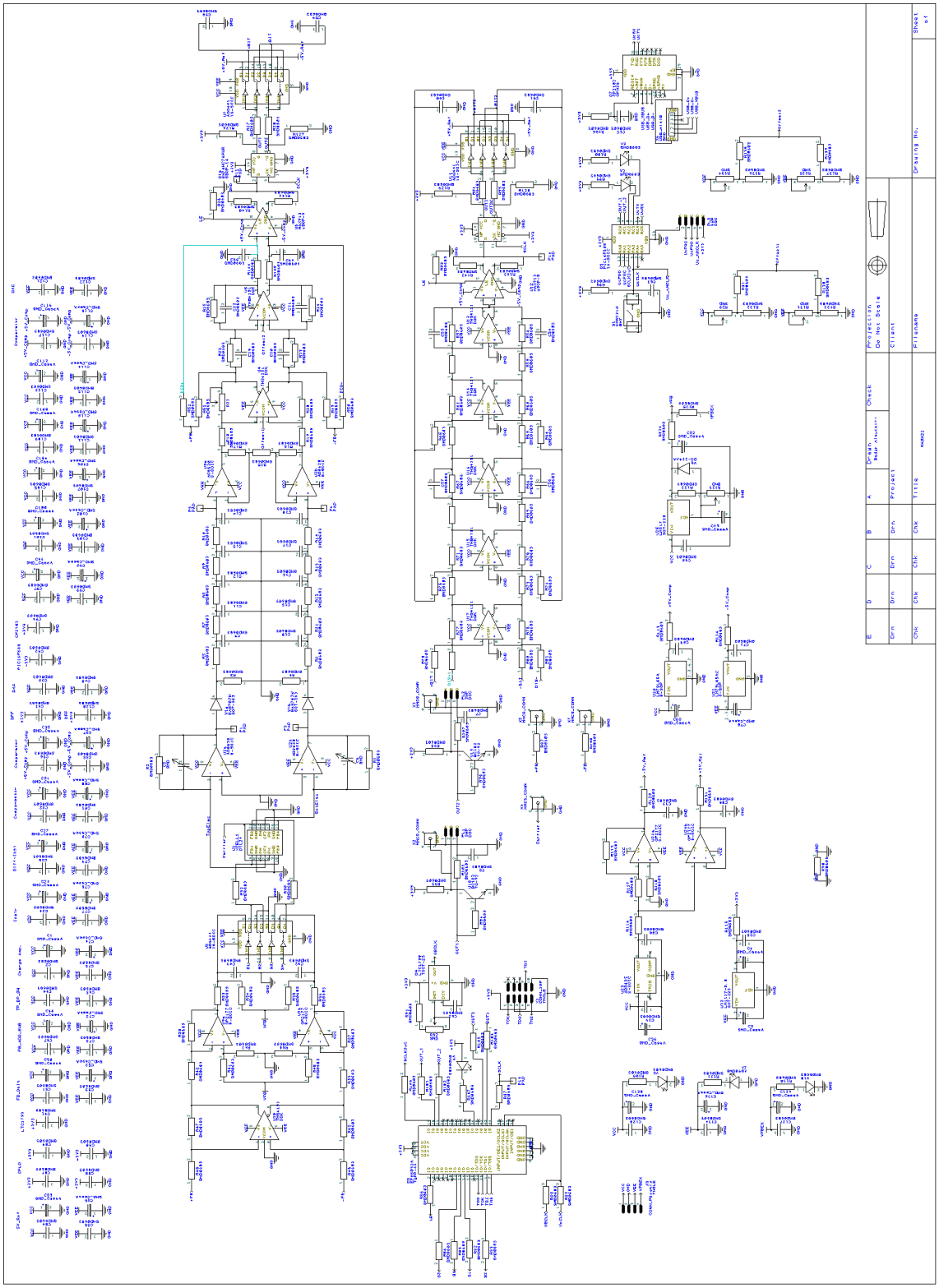
```
// SerialRAM connections
sbit Chip_Select at RG3_bit;
sbit Chip_Select_Direction at TRISG3_bit;
// End SerialRAM connections
#include "__NetEthInternal.h"
// mE ehternet NIC pinout
sfr sbit Net_Ethernet_Intern_Rst at LATC0_bit;  // for writing to
output pin always use latch
sfr sbit Net_Ethernet_Intern_CS  at LATC1_bit;  // for writing to
output pin always use latch
sfr sbit Net_Ethernet_Intern_Rst_Direction at TRISC0_bit;
sfr sbit Net_Ethernet_Intern_CS_Direction  at TRISC1_bit;
// end ethernet NIC definitions
unsigned char    myMacAddr[6] = {0x00, 0x14, 0xA5, 0x76, 0x19,
0x3f} ;  // my MAC address
unsigned char  myIpAddr[4] = {111, 11,  11, 10 } ;              //
my IP address
unsigned char  gwIpAddr[4] = {111, 11,  11,  1 } ;              //
gateway (router) IP address
unsigned char     ipMask[4]       = {255,  255,  255,    0  } ;
// network mask
unsigned char     dnsIpAddr[4]  =  {111,  11,    11,    254  } ;
// DNS server IP address
unsigned char  destMAC[6] = {0x00, 0x50, 0xB6, 0x54, 0x46, 0x2D}
;  // dest MAC address
unsigned char  IpAddr[4] = {111, 11,   11,  11 };              //
dest IP address
unsigned long int i,address;
unsigned int j,k;
char FFlag;
void Net_Ethernet_Intern_UserTCP(SOCKET_Intern_Dsc *socket) {
```

```
}
unsigned  int         Net_Ethernet_Intern_UserUDP(UDP_Intern_Dsc
*udpDsc) {

         // while(udpDsc->dataLength--)
         // {
          if (Net_Ethernet_Intern_getByte()=='a') FFlag=1;
          else FFlag=0;
         // }

  return(0);
}
void MCUInit() {
 // INTCON = 0x00;      // disable interrupt
 // RCON.IPEN = 0;

  ADCON1 |= 0x0F;      // Configure AN pins as digital
  CMCON  |= 7;         // Disable comparators

  PORTA = 0 ;
  TRISA = 0xfc ;          // set PORTA as input for ADC
                               // except RA0 and RA1 which will
be used as
                               // ethernet's LEDA and LEDB
  PORTB = 0 ;
  TRISB = 0; ;          // set PORTB as input

  PORTC = 0 ;
  TRISC = 0xff ;          // set PORTB as input

  PORTE = 0 ;
  TRISE = 0xff ;          // set PORTB as input

  PORTF = 0 ;
  TRISF = 0xff ;          // set PORTB as input

  PORTJ = 0 ;
  TRISJ = 0 ;          // set PORTB as input*/

  PORTD = 0 ;
  TRISD = 0xff ;             // set PORTD as input

  FFlag=0;

  Chip_Select_Direction = 0;             // Set CS# pin as Output
  Chip_Select = 1;                        // Deselect SerialRAM
  OSCTUNE = 0b01000000;       //41 MHz
  SPI1_Init();                           // Initialize SPI module
  Delay_ms(2000);
  Net_Ethernet_Intern_Init(myMacAddr,                  myIpAddr,
Net_Ethernet_Intern_FULLDUPLEX) ; // init ethernet board
  Net_Ethernet_Intern_confNetwork(ipMask, gwIpAddr, dnsIpAddr);
  Delay_ms(2000);
}
void main() {
  MCUInit();
```

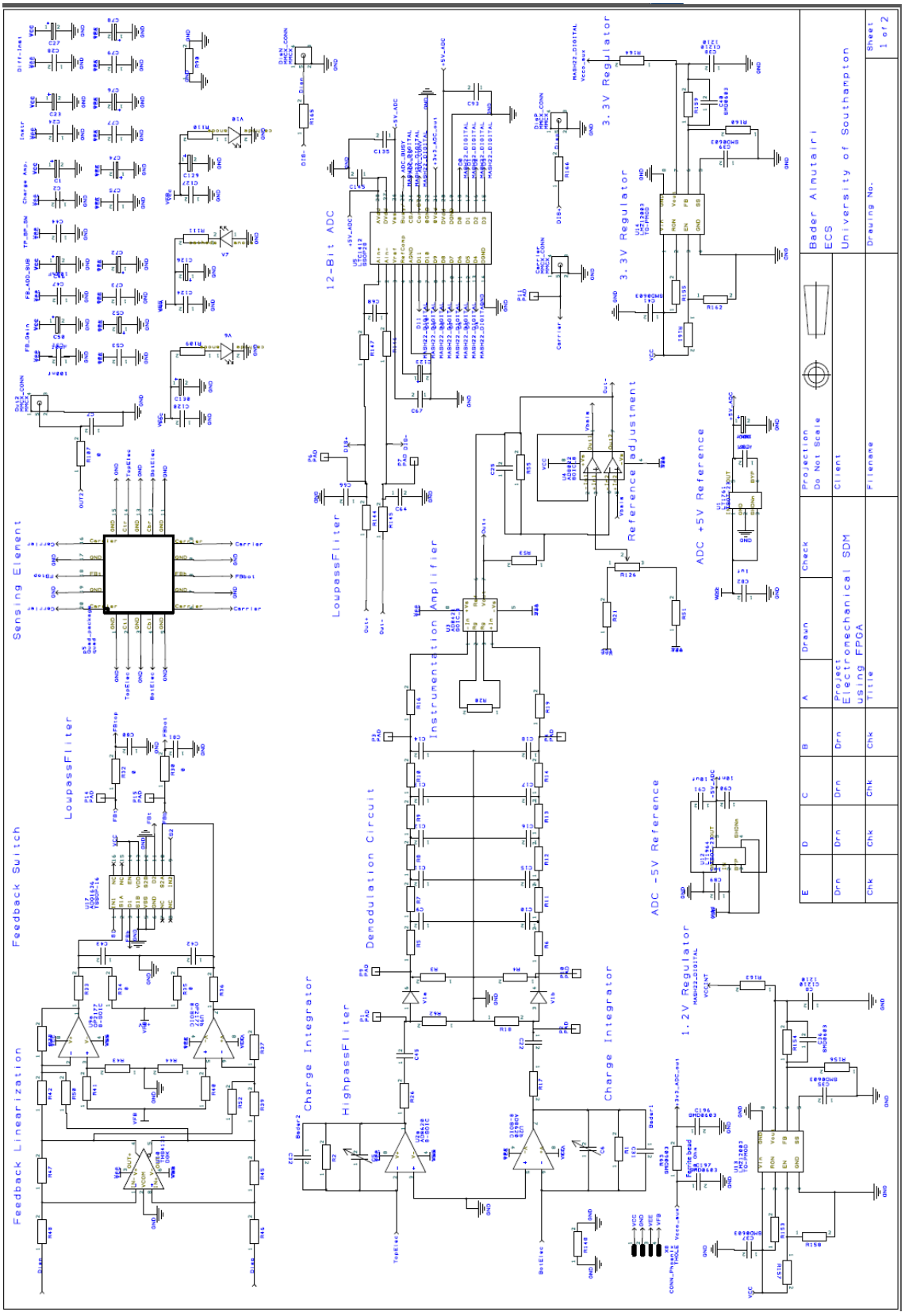
```
  while(1)
  {
          i=0;
          Chip_Select = 0;                              // Select
SerialRAM
          SPI1_Write(2);                    // Write instruction
          SPI1_Write(0);               // Sending 16 bits address
          SPI1_Write(0);               // Sending 16 bits address
          SPI1_Write(0);
        while (i<131072)
      {
          while (PORTB.B0==0){};
          SPI1_Write(PORTH);
          SPI1_Write(PORTE);                        // Writing one
byte of data
          i=i+2;
          while (PORTB.B0==1){};
      }
      Chip_Select = 1;                       // Deselect SerialRAM
    Net_Ethernet_Intern_doPacket();
    if (FFlag==1)
    {
      address=0;
      for(k=0; k<128; k++)
        {
          Net_Ethernet_Intern_payloadInitUDP();
          j=0;
          Chip_Select = 0;                              // Select
SerialRAM
          SPI1_Write(3);                        // Read instruction
          SPI1_Write(address >> 16);                // Sending 16
bits address
          SPI1_Write(address >> 8);                 // Sending 16
bits address
          SPI1_Write(address);
        while (j<1024)
          {
              Net_Ethernet_Intern_putByte(SPI1_Read(0));
              Net_Ethernet_Intern_putByte(SPI1_Read(0));
              j=j+2;
          }
          Chip_Select = 1;                              // Deselect
SerialRAM
          address=address+1024;
          Net_Ethernet_Intern_flushUDP(destMAC,  IpAddr,  10101,
10101, 1024);
       }

       FFlag=0;
      }

  }
}
```

Appendix D Circuit Schematic

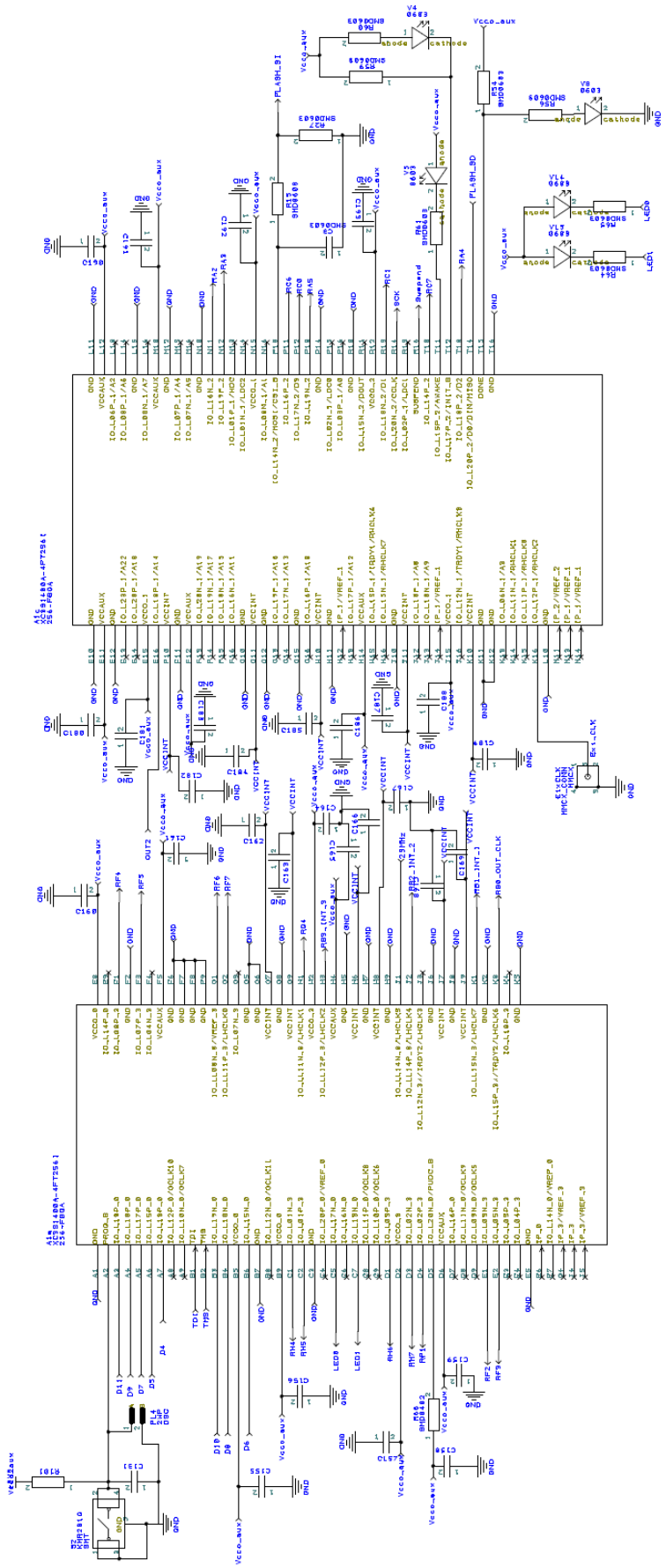
D.1 MASH22 PCB Schematic

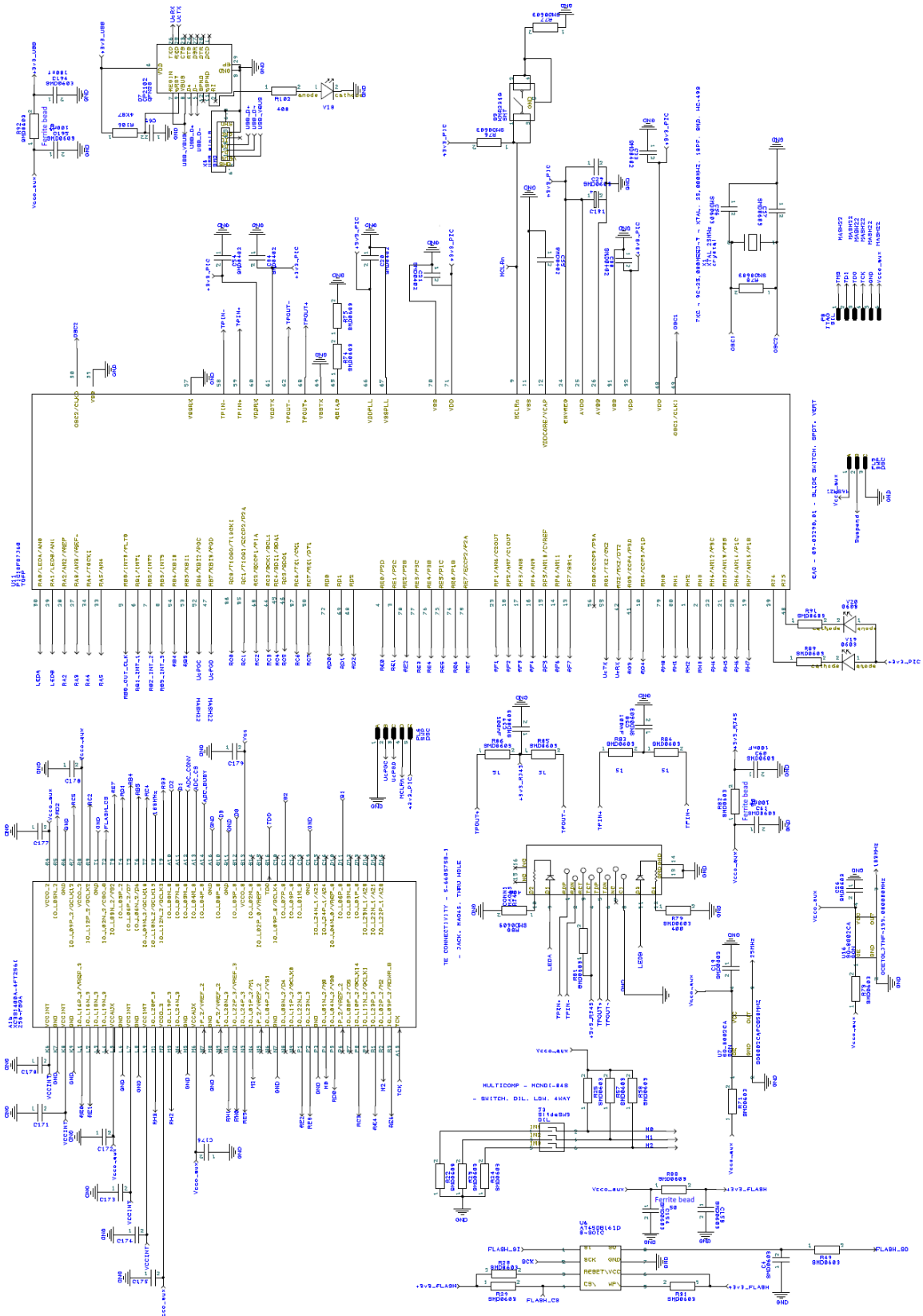


D.2 MASH20 Analogue Circuit



D.3 MASH20 Digital Circuit





List of References

[1] Nathanson, H.C., Newell, W.E., Wickstrom, R.A, Davis, J.R., "The resonant gate transistor," *IEEE Transactions on Electron Devices,* vol. 14, no. 3, pp. 117-133, March 1967.

[2] Perlmutter, M., Robin, L.,, "High-performance, low cost inertial MEMS: A market in motion!," *IEEE/ION Position Location and Navigation Symposium (PLANS), 2012,* pp. 225-229, 23-26 April 2012.

[3] Bogue, R., "Recent developments in MEMS sensors: a review of applications, markets and technologies," *Sensor Review,* vol. 33, no. 4, pp. 300-304, 2013.

[4] Hollocher, D. et al., "A very low cost, 3-axis, MEMS accelerometer for consumer applications," *IEEE Sensors 2009,* pp. 953-957, 25-28 Oct. 2009.

[5] Hasan, A. M., Samsudin, K., Ramli, A., Azmir, R., Ismaeel, S., "A review of navigation systems (integration and algorithms)," *Australian Journal of Basic and Applied Sciences,* vol. 3, no. 2, pp. 943-959, 2009.

[6] Brown, A.K., "GPS/INS uses low-cost MEMS IMU," *IEEE Aerospace and Electronic Systems Magazine,* vol. 20, no. 9, pp. 3-10, Sept. 2005.

[7] Stauffer, J. -M, Dietrich, O., Dutoit, B., "RS9000, a novel MEMS accelerometer family for mil/aerospace and safety critical applications," *Position Location and Navigation Symposium (PLANS), 2010 IEEE/ION,* pp. 1-5, 4-6 May 2010.

[8] Tanenhaus, M., Carhoun, D., Geis, T., Wan, E., Holland, A., "Miniature IMU/INS with optimally fused low drift MEMS gyro and accelerometers for applications in

GPS-denied environments," *IEEE/ION Position Location and Navigation Symposium (PLANS), 2012*, pp. 259-264, 23-26 April 2012.

[9] Elman, N., Patta, Y., Duc, H.L.H., Daniel, K., Masi, B., Scott, A., Cima, M., "Micro- and nano-electro mechanical (MEMS and NEMS)-based technologies for implanted biomedical devices," *IEEE International Electron Devices Meeting (IEDM)*, pp. 1-4, Dec. 15-17 Dec. 2008.

[10] Trifunovic, M., Vadiraj, AM., van Driel, W.D., "MEMS accelerometers and their bio-applications," *13th International Conference on Thermal, Mechanical and Multi-Physics Simulation and Experiments in Microelectronics and Microsystems (EuroSimE), 2012*, pp. 1/7 - 7/7, 16-18 April 2012.

[11] Lee, S., Nam, G-J., Chae, J., Kim, H., Drake, A.J., "Two-dimensional position detection system with MEMS accelerometers, readout circuitry, and microprocessor for padless mouse applications," *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, vol. 13, no. 10, pp. 1167-1178, Oct. 2005.

[12] Lapinski, M., Feldmeier, M., Paradiso, J.A., "Wearable wireless sensing for sports and ubiquitous interactivity," *IEEE Sensors 2011*, pp. 1425-1428, 28-31 Oct. 2011.

[13] Hu, Y., Yan, Y., Liang, J., Wang, L., "A miniature, low-cost MEMS AHRS with application to posture control of robotic fish," *IEEE International Instrumentation and Measurement Technology Conference (I2MTC), 2013* , pp. 1392 - 1395, 6-9 May 2013.

[14] Yazdi, N., Ayazi, F., Najafi, K., "Micromachined inertial sensors," *Proceedings of the IEEE*, vol. 86, no. 8, pp. 1640-1659, Aug. Aug. 1998.

[15] Beeby, S., Ensell, G., Kraft, M., White, N., MEMS mechanical sensors, London: Artech House, Inc, 2004.

[16] Choudhary, V., Iniewski, K., MEMS fundamental technology and applications, Boca Raton, FL: CRC Press - Taylor & Francis Group, 2013.

[17] Soen, J., Voda, A., Cyril, C., "Controller design for a closed-loop micromachined accelerometer," *Control Engineering Practice,* vol. 15, no. 1, p. 57–68, Jan. 2007.

[18] Borovic, B., Liu, A. Q., Popa, D., Cai, H., Lewis, F. L., "Open-loop versus closed-loop control of MEMS devices: choices and issues," *Journal of Micromechanics and Microengineering,* vol. 15, no. 10, pp. 1917-1924, October Aug. 2005.

[19] Gaura, E., Newman, R., Kraft, M., Smart MEMS and sensor systems, London: Imperial College Press, 2006.

[20] Henrion, W., DiSanza, L., Lp, M., Terry, S., Jarman, H., "Wide dynamic range direct digital accelerometer," *IEEE Solid-State Sensor and Actuator Workshop, 1990. 4th Technical Digest.,* pp. 153-157, 4-7 Jun 4-7 June 1990.

[21] Dong, Y., Kraft, M., Gollasch, C., Redman-White, W., "A high-performance accelerometer with a fifith-order sigma-delta modulator," *Journal of Micromechanics and Microengineering,* vol. 15, no. 7, pp. 1-8, June 2005.

[22] Petkov, V., Boser, B., "A fourth-order ΣΔ interface for micromachined inertial sensors," *IEEE Journal of Solid-State Circuits,* vol. 40, no. 8, pp. 1602-1609, Aug. 2005.

[23] Sönmez, U., Külah, H., Akın, T., "A fourth order unconstrained ΣΔ capacitive accelerometer," *16th International Solid-State Sensors, Actuators and Microsystems Conference (TRANSDUCERS),* pp. 707-710, 5-9 June 2011.

[24] Bourdopoulos, G., Pnevmatikakis, A., Anastassopoulos, V., Deliyannis, T., Delta Sigma modulator: Modeling, design, and applications, London: Imperial college press, 2003.

[25] Petkov, V., Boser, B., "High-order electromechanical SD modulatoin in micromachined inertial sensors," *IEEE transaction on circuits and systems,* vol. 53, no. 5, pp. 1016-1022, May 2006.

[26] Almutairi, B., Kraft, M., "Comparative study of multi stage noise shaping and single loop sigma–delta modulators for MEMS accelerometers," *Procedia Engineering, Eurosensors XXIV, Linz - Austria,* vol. 5, pp. 512-515, 5-8 Sept 2010.

[27] Almutairi, B., Kraft, M., "Multi stage noise shaping sigma-delta modulator (MASH) for capacitive MEMS accelerometers," *Procedia Engineering, Eurosensors XXV,* vol. 25, p. 1313 – 1316, 4-7 Spet. 2011.

[28] Almutairi, B., Kraft, M., "Experimental study of single loop sigma-delta and multi stage noise shaping (MASH) modulators for MEMS accelerometer," *IEEE Sensors Conference,* pp. 520-523, 28-31 October Oct. 2011.

[29] Almutairi, B., Kraft, M., "Multi stage noise shaping sigma–delta modulator (MASH) for capacitive MEMS accelerometers," *Sensors and Actuators A: Physical,* vol. 186, p. 169– 177, 2012.

[30] Kraft, M., Wilcock, R., Almutairi, B., "Innovative control systems for MEMS inertial sensors," *IEEE International Frequency Control Symposium (FCS), 2012*, pp. 1-6, 21-24 May 2012.

[31] Leslie, T.C., Singh, B., "An improved sigma-delta modulator architecture," *IEEE International Symposium on Circuits and Systems*, vol. 1, pp. 372-375, 1-3 May 1990.

[32] Sakimura, N., Yamaguchi, M., Yotsuyanagi, M., "Multibit delta-sigma architectures with two-level feedback loop using a dual-quantization architecture," *IEICE Trans. Fundamentals*, Vols. E84-A, no. 2, pp. 497-505, 2001.

[33] Rio, R., Medeiro, F., Perez-Verdue, B., de la Rosa, J., Rodriguez-Vazquez, A., CMOS cascade sigma-delta modulators for sensors and telecom: Error analysis and practical design, Dordrecht - The Netherlands: Springer, 2006.

[34] Korvink, J. G., Paul, O., MEMS: A practical guide to design, analysis, and application, New York: Willian Andrew, 2006.

[35] Kaajakari, V., Practical MEMS, Las Vegas, NY: Small Gear Publishing, 2009.

[36] Gad-el-Hak, M., MEMS: Introduction and fundamentals, New York: CRC Press, 2006.

[37] Senturia, S., Microsystem design, New York: Kluwer academic publishers, 2002.

[38] Lobontiu, N., Garcia, E., Mechanics of microelectromechanical systems, Boston: Kluwer academic Publishers, 2005.

[39] Kraft, M., “Closed loop digital accelerometer employing oversampling conversion,” Ph.D. dissertation, School of Engineering - Coventry University, Coventry - UK, 1997.

[40] Kempe, V., Inertial MEMS: principles and practice, Cambridge: Cambridge University Press, 2011.

[41] Maluf, N., Williams, K., An introduction to microelectromechanical systems engineering, Norwood, MA: Artech House, Inc, 2004.

[42] Bao, M., Yang, H., “Squeeze film air damping in MEMS,” *Sensors and Actuators A: Physical,* vol. 136, no. 1, pp. 3-27, May 2007.

[43] Sumali, H., “Squeeze-film damping in the free molecular regime: Model validation and measurement on a MEMS,” *Journal of Micromechanics and Microengineering,* vol. 17, no. 11, p. 2231–2240, Nov. 2007.

[44] Lu, C., Lemkin, M., Boser, B. E., “A monolithic surface micromachined accelerometer with digital output,” *IEEE Journal of Solid-State Circuits,* vol. 30, no. 12, pp. 1367-1373, Dec. 1995.

[45] vab Kampen, R. P., Vellekoop, M. J., Sarro, P. M., Wolffenbuttel, R. F., “Application of electrostatic feedback to critical damping of an integrated silicon capacitive accelerometer,” *Sensors and Actuators A: Physical,* vol. 43, no. 1-3, pp. 100-106, May 1994.

[46] Acar, C., Shkel, A., MEMS vibratory gyroscopes: Structural approaches to improve robustness, New York: Springer, 2009.

[47] Barker, N. S., Rebeiz, G. M., "Distributed MEMS true-time delay phase shifters and wide-band switches," *IEEE Transaction on Microwave Theory and Techniques,* vol. 46, no. 11, pp. 1881-1890, Nov. 1998.

[48] Fargas-Marques, A., Casals-Terre, J., Shkel, A.M., "Resonant pull-in condition in parallel-plate electrostatic actuators," *Journal of Microelectromechanical Systems,* vol. 16, no. 5, pp. 1044-1053, Oct. 2007.

[49] Kraft, Michael, Closed loop digital accelerometer employing oversampling conversion, PhD thesis: School of Engineering, Coventry University - UK, July 1997.

[50] Rosa, M. A., Bruyker, D., Volkel, A. R., Peeters, E., Dunec, J., "A novel external electrode configuration for the electrostatic actuation of MEMS based devices," *Journal of Micromechanical and Microengineering,* vol. 14, pp. 446-451, Jan. 2004.

[51] Hung, E. S., Senturia, S. D., "Extending the travel range of analog-tuned electrostatic actuators," *Journal of Microelectomechanical System,* vol. 8, no. 4, pp. 497-505, Dec. 1999.

[52] Seeger, J. I., Arbor, A., Crary, S. B., "Stabilization of electrostatically actuated mechanical devices," *International Conference on Solid State Sensors and Actuators, 1997. TRANSDUCERS '97 Chicago,* vol. 14, pp. 1133 - 1136, 16-19 Jun 1997.

[53] Seeger, J. I., Crary, S. B., "Stabilization of electrostatically actuated mechanical devices," *International Conference on Solid State Sensors and Actuators, TRANSDUCERS '97 Chicago,* vol. 2, pp. 1133-1136, 16-19 Jun 1997.

[54] Castañer, L., Pons, J., Nadal-Guardia, R., Rodríguez, A., "Analysis of the extended operation range of electrostatic actuators by current-pulse drive," *Sensors and Actuators A: Physical,* vol. 90, no. 3, pp. 181-190, May 2001.

[55] Schreier, R., Temes, G. C., Understanding delta-sigma data converters, New Jersey: John Wiley & Sons, Inc, 2005.

[56] Norsworthy, S., Schreier, R., Temes, G., Delta-sigma data converters, theory, design and simulation, New Jersey: John Wiley & Sons, Inc, 1997.

[57] Johns, D., Martin, K., Analog integrated circuit design, Toronto, Canada: John Wiley and Sons, Inc, 1997.

[58] Marks, R.J., Introduction to shannon sampling and interpolation theory, New York: Springer-Verlag, 1991.

[59] Unser, M., "Sampling-50 years after shannon," *Proceedings of the IEEE,* vol. 88, no. 4, pp. 569-587, 2000.

[60] Kozak, M., Izzet, K., Oversampled delta-sigma modulators, Dordrecht - Netherlands: Kluwer Academic Publishers, 2003.

[61] Yazdi, N., Najafi, K., "An all-silicon single-wafer micro-g accelerometer with a combined surface and bulk micromachining process," *Journal of Microelectromechanical Systems,* vol. 9, no. 4, pp. 544-550, Dec. 2000.

[62] Chae, J., Kulah, H., Najafi, K., "An in-plane high-sensitivity, low-noise micro-g silicon accelerometer with CMOS readout circuitry," *Journal of Microelectromechanical Systems,* vol. 13, no. 4, pp. 628-635, Aug. 2004.

[63] Abdolvand, R., Amini, B. V., Ayazi, F., "Sub-micro-gravity in-plane accelerometers with reduced capacitive gaps and extra seismic mass," *Journal of Microelectromechanical System,* vol. 16, no. 5, pp. 1036-1043, Oct. 2007.

[64] Smith, T., Nys, O., Chevroulet, M., DeCoulon, Y., Degrauwe, M., "A 15b electromechanical sigma-delta converter for acceleration measurements," *IEEE International Solid-State Circuits Conference (ISSCC'94),* p. 160–161, 16-18 Feb 1994.

[65] Rudolf, F., Gonseth, S., Brisson, R., Krebs, P., "New generation of high performance/high reliability MEMS accelerometers for harsh environment," *IEEE/ION Position, Location and Navigation Symposium - PLANS 2014,* pp. 7-11, 5-8 May 2014.

[66] Jiang, X., Wang, F., Kraft, M., Boser, B. E., "An integrated surface micromachined capacitive lateral accelerometer with 2 uG/rt-Hz resolution," *Digest of Solid State Sensor and Actuator Workshop, Hilton Head Island of America, United States,* pp. 202-205, June 2002.

[67] Yazdi, N., Kulah, H., Najafi, K., "Precision readout circuits for capacitive microaccelerometers," *Proceedings of IEEE Sensors,* vol. 1, pp. 28-31, 24-27 Oct. 2004.

[68] Chiang, C-T., Wang, C-S., Huang, Y-C., "A monolithic CMOS autocompensated sensor transducer for capacitive measuring systems," *IEEE Transactions on Instrumentation and Measurement,* vol. 57, no. 11, pp. 2472-2486, Nov 2008.

[69] Weijie Y., Howe, R.T., Gray, P.R., "Surface micromachined, digitally force-balanced accelerometer with integrated CMOS detection circuitry," *Solid-State*

Sensor and Actuator Workshop, 1992. 5th Technical Digest., IEEE, pp. 126-131, 22-25 Jun 22-25 Jun 1992.

[70] Lu, C., Lemkin, M., Boser, B.E., "A monolithic surface micromachined accelerometer with digital output," *Journal of Solid-State Circuit,* vol. 30, no. 12, pp. 160-161, 15-17 Feb. 1995.

[71] Kraft, M., Lewis, C., Hesketh, T., "Closed-loop silicon accelerometers," *Circuits, Devices and Systems, IEE Proceedings,* vol. 145, no. 5, pp. 325-331, Oct. 1998.

[72] Jiang, X., Seeger, J., Kraft, M., Boser, B., "A monolithic surface micromachined Z-axis gyroscope with digital output," *VLSI Circuits, 2000. Digest of Technical Papers. 2000 Symposium,* pp. 16-19, 15-17 June 2000.

[73] Kulah, H., Chae, J., Yazdi, N., Najafi, K., "Noise analysis and characterization of a sigma-delta capacitive microaccelerometer," *IEEE Journal of Solid-State Circuits,* vol. 41, no. 2, pp. 352-361, Feb. 2006.

[74] Dong, Y., "Control system for capacitive micromachined inertial sensor with high order sigma delta modulators," Ph.D. dissertation, ECS - University of Southampton, Southampton - UK, 2006.

[75] Lang, C., Tielert, R., "A low noise accelerometer with digital PID-type controller and multibit force feedback," *Proceedings of the 25th European Solid-State Circuits Conference, ESSCIRC '99.* , pp. 250-253, 21-23 Sept. 1999.

[76] Wu, J., Carley, L. R., "A simulation study of electromechanical delta-sigma modulators," *The 2001 IEEE International Symposium on Circuits and Systems. ISCAS 2001,* vol. 2, pp. 453 - 456, May 2001.

[77] de la Rosa, J.M., "Sigma-delta modulators: Tutorial overview, design guide, and state-of-the-art survey," *IEEE Transactions on Circuits and Systems I: Regular Papers,* vol. 58, no. 1, pp. 1-21, Jan. 2011.

[78] Baird, R. T., Fiez, T. S.,, "Linearity enhancement of multibit ΔΣ A/D and D/A converters using data weighted averaging," *IEEE Transactions on Circuits and Systems II: Analog and Digital Signal Processing,* vol. 42, no. 15, pp. 753-762, Dec 1995.

[79] Arias, J.; Kiss, P.; Boccuzzi, V.; Quintanilla, L., Enriquez, L., Vicente, J., Bisbal, D., Pablo, J. S., Barbolla, J. , "Nonlinearity correction for multibit ΔΣ DACs," *Transactions on Circuits and Systems I: Regular Papers, IEEE,* vol. 52, no. 6, pp. 1033-1041, June 2005.

[80] Dong, Y., Kraft, M., Redman-White, W., "Force feedback linearization for higher-order electromechanical sigma-delta modulators," *Journal of Micromechanics and Microengineering,* vol. 16, pp. S54-S60, 2006.

[81] Pastre, M., Kayal, M., Schmid, H., Huber, A., Zwahlen, P., Nguyen, A.-M., Dong, Y., "A 300Hz 19b DR capacitive accelerometer based on a versatile front end in a 5th-order ΔΣ loop," *Proceedings of ESSCIRC,* pp. 288-291, 2009.

[82] Dong, Y., Zwahlen, P., Nguyen, A., Rudolf, F., Stauffer, J., "High performance inertial navigation grade sigma-delta MEMS accelerometer," *Position Location and Navigation Symposium (PLANS), 2010 IEEE/ION,* pp. 32-36, 4-6 May 2010.

[83] Dong, Y., Zwahlen, P., Nguyen, A.M., Frosio, R., Rudolf, F., "Ultra-high precision MEMS accelerometer," *16th International Solid-State Sensors, Actuators and Microsystems Conference (TRANSDUCERS),* pp. 695-698, 5-9 June 2011.

[84] Northemann, T., Maurer, M., Buhmann, A., He, L., Manoli, Y., "Excess loop delay compensated electro-mechanical bandpass sigma-delta modulator for gyroscopes," *Proceedings of the Eurosensors XXIII conference,* vol. 1, no. 1, pp. 1183-1186, 2009.

[85] Northemann, T., Maurer, M., Rombach, S., Buhmann, A., Manoli, Y., "Drive and sense interface for gyroscopes based on bandpass sigma-delta modulators," *Proceedings of 2010 IEEE International Symposium on Circuits and Systems (ISCAS),* pp. 3264-3267, May 30 2010-June 2 2010.

[86] Northemann, T., Maurera, M., Rombacha, S., Buhmanna, A., Manolia, Y., "A digital interface for gyroscopes controlling the primary and secondary mode using bandpass sigma–delta modulation," *Sensors and Actuators A: Physical,* vol. 162, no. 2, pp. 388-393, Aug. 2010.

[87] Amini, B. V., Abdolvand, R., Ayazi, F., "A 4.5-mW closed-loop ΣΔ micro-gravity CMOS SOI accelerometer," *IEEE Journal of Solid-State Circuits,* vol. 41, no. 12, pp. 2983-2991, Dec. 2006.

[88] Dorf, R., Bishop, R., Modern control systems, Upper Saddle River, NJ: Prentice Hall, 2008.

[89] Raman, J., Rombouts, P., Weyten, L., "An unconstrained architecture for systematic design of higher order ΣΔ force-feedback loops," *IEEE Transactions on Circuits and Systems I: Regular Papers,* vol. 55, no. 6, pp. 1601-1614, July 2008.

[90] Dong, Y., Kraft, M., Redman-White, W., "Higher order noise-shaping filters for high-performance micromachined accelerometers," *IEEE Transactions on Instrumentation and Measurement,* vol. 56, no. 5, pp. 1666-1674, Oct. 2007.

[91] Chao, K., Nadeem, S., Lee, W., Sodini, C., "A higher order topology for interpolative modulators for oversampling A/D converters," *IEEE Transactions on Circuits and Systems*, vol. 37, no. 3, pp. 309-318, March 1990.

[92] Raman, J., Rombouts, P., Weyten, L., "An unconstrained architecture for high-order ΣΔ force-feedback inertial sensors," *IEEE International Symposium on Circuits and Systems (ISCAS)*, pp. 3063-3066, 27-30 May 2007.

[93] Raman, J., Cretu, E., Rombouts, P., Weyten, L., "A closed-loop digitally controlled MEMS gyroscope with unconstrained sigma-delta force-feedback," *IEEE Sensors Journal*, vol. 9, no. 3, pp. 297 - 305, March 2009.

[94] Schreier, R., "The Delta-Sigma Toolbox forMATLAB Oregon State University," 14 Jan 2000. [Online]. Available: http://www.mathworks.com/matlabcentral/fileexchange/19-delta-sigma-toolbox. [Accessed 07 Aug 2011].

[95] El-Shennawy, A.K., Aboushady, H., El-Sayed, A., "Design method of a ΣΔ-based closed loop gyroscope," *Design and Test Workshop (IDT), 2009 4th International*, pp. 1-4, 15-17 Nov. 2009.

[96] Aboushady, H., Louerat, M., "Systematic approach for discrete-time to continuous-time transformation of ΣΔ modulators," *IEEE International Symposium on Circuits and Systems (ISCAS)*, vol. 4, pp. IV-229- IV-232, 2002.

[97] Lota, J., Al-Janabi, M., Kale, I., "Nonlinear-stability analysis of higher order Δ-Σ modulators for and sinusoidal inputs," *IEEE Transactions on Instrumentation and Measurement*, vol. 57, no. 3, pp. 530-542, Jan 2008.

[98] Wilcock, R.; Kraft, M., "Genetic Algorithm for the Design of Electro-Mechanical Sigma Delta Modulator MEMS Sensors," *Sensors 2011*, vol. 11, no. 10, p. 9217–9232, Sept. 2011.

[99] Deb, K., Multi-objective optimization using evolutionary algorithms, Hoboken, New Jersey: John Wiley & Sons Ltd., 2001.

[100] Hayashi, T., Inabe, Y., Uchimura, K., Kimura, T., "A multistage delta-sigma modulator without double integration loop," *IEEE International Solid-State Circuits Conference. Digest of Technical Papers*, vol. XXIX, pp. 182 - 183, 19-21 Feb. 1986.

[101] Brooks, T.L., Robertson, D.H., Kelly, D.F., Del Muro, A, Harston, S.W., "A cascaded sigma-delta pipeline A/D converter with 1.25 MHz signal bandwidth and 89 dB SNR," *IEEE Journal of Solid-State Circuits*, vol. 32, no. 12, pp. 1896-1906, Dec 1997.

[102] Breems, L.J., Rutten, R., Wetzker, G., "A cascaded continuous-time ΣΔ modulator with 67-dB dynamic range in 10-MHz bandwidth," *IEEE Journal of Solid-State Circuits*, vol. 39, no. 12, pp. 2152-2160, Dec. 2004.

[103] Breems, L.J., Rutten, R., Veldhoven, V., Robert H. M., Van Der Weide, G., "A 56 mW continuous-time quadrature cascaded ΣΔ modulator with 77 dB DR in a near zero-IF 20 MHz band," *IEEE Journal of Solid-State Circuits*, vol. 42, no. 12, pp. 2696-2705, Dec. 2007.

[104] Uchimura, K., Hayashi, T., Kimura, T., Atsushi, I., "Oversampling A-to-D and D-to-A converters with multistage noise shaping modulators," *IEEE Transactions on Acoustics, Speech and Signal Processing*, vol. 36, no. 12, pp. 1899-1905, Dec. 1988.

[105] Wang, X., Moon, U., Liu, M., Temes, G.C., “Digital correlation technique for the estimation and correction of DAC errors in multibit mash ΔΣ ADCs,” *IEEE International Symposium on Circuits and Systems (ISCAS),* vol. 4, pp. IV-691,IV-694, 2002.

[106] Cauwenberghs, G.; Temes, G.C.,, “Adaptive digital correction of analog errors in MASH ADCs. I. Off-line and blind on-line calibration,” *IEEE Transactions on Circuits and Systems II: Analog and Digital Signal Processing,,* vol. 47, no. 7, pp. 621-628, Jul 2000.

[107] Kiss, P., Silva, J., Wiesbauer, A., Sun, T., Un-Ku M., Stonick, J.T., Temes, G.C., “Adaptive digital correction of analog errors in MASH ADCs. II. Correction using test-signal injection,” *IEEE Transactions on Circuits and Systems II: Analog and Digital Signal Processing,,* vol. 47, no. 7, pp. 629-638, Jul 2000.

[108] Shu, Y-S., Kamiishi, J., Tomioka, K., Hamashita, K., Song, B-S., “LMS-based noise leakage calibration of cascaded continuous-time \delta \sigma modulators,” *IEEE Journal of Solid-State Circuits,* vol. 45, no. 2, pp. 368-379, Feb. 2010.

[109] Fujimori, I., Longo, L., Hairapetian, A., Seiyama, K., Kosic, S., Cao, J., Chan, S., “A 90 dB SNR, 2.5 MHz output rate ADC using cascaded multibit ΣΔ modulation at 8x oversampling ratio,” *IEEE International Solid-State Circuits Conference, Digest of Technical Papers. ISSCC. 2000* , pp. 338-339, 9-9 Feb. 2000.

[110] vanderZwan, E. J., Dijkmans, E.C., “A 0.2-mWCMOS ΣΔ modulator for speech coding with 80 dB dynamic range,” *IEEE Journal of Solid-State Circuits,* vol. 31, no. 12, pp. 1873-1880, Dec. 1996.

[111] Breems, L. J., Rutten, R., Wetzker, G., "A cascaded continuous-time ΣΔ modulator with 67-dB dynamic range in 10-MHz bandwidth," *IEEE Journal of Solid-State Circuits,* vol. 39, no. 12, pp. 2152-2160, Dec. 2004.

[112] "A hybrid delta-sigma modulator with adaptive calibration," *International Symposium on Circuits and Systems, ISCAS '03,* vol. 1, pp. I-1025,I-1028, 25-28 May 2003.

[113] Kawahito, S., Maier, C., Schneider, M., Zimmermann, M., Baltes, H., "A 2-D CMOS microfluxgate sensor system for digital detection of weak magnetic fields," *IEEE Journal of Solid-State Circuits,* vol. 34, no. 12, pp. 1843-1851, Dec. 1999.

[114] Maghari, N., Kwon, S., Temes, G.C., Moon, U., "Mixed-order sturdy MASH Δ-Σ modulator," *IEEE International Symposium on Circuits and Systems (ISCAS),* pp. 257-260, 27-30 May 2007.

[115] Maghari, N., Un-Ku, M., "Multi-loop efficient sturdy MASH delta-sigma modulators," *IEEE International Symposium on Circuits and Systems (ISCAS),* pp. 1216-1219, 18-21 May 2008.

[116] Maghari, N., Kwon, S., Un-Ku, M., "74dB SNDR multi-loop sturdy-MASH delta-sigma modulator using 35dB opamp gain," *IEEE Journal of Solid-State Circuits,* vol. 44, no. 8, pp. 2212-2221, Aug. 2009.

[117] Kraft, M., Redman-White, W., Mokhtari, M., "Closed loop micromachined sensors with higher order SD-modulators," *Proceedings 4th Conference on Modeling and Simulation of Microsystems,* pp. 104-107, March 2001.

[118] Mokhtari, M., Advanced SDM control systems for MEMS capacitive sensing accelerometers, Ph.D. dissertation, ECS - Univerisity of Southampton: Southampton - UK, 2005.

[119] Liu, X., Liu, Y., Chen, W., Wu, Q., "A fourth-order sigma-delta interface circuit for closed-loop micromachined accelerometer," *Proceedings of the Eurosensors XXIII conference,* vol. 1, no. 1, pp. 1187-1190, September Sept 2009.

[120] Sari, I., Zeimpekis, I., Kraft, M., "A full wafer dicing free dry release process for MEMS devices," *Procedia Engineering,* vol. 5, pp. 850-853, 2010.

[121] Meirovitch, L., Fundamentals of vibrations, New York, USA: McGraw-Hill, 2001.

[122] Paz, M., Leigh, W., Structural dynamics: theory and computation, Massachusetts, USA: Kluwer Academic Publishers, 2004.

[123] Wongkomet, N., Boser, B.E.,, "Correlated double sampling in capacitive position sensing circuits for micromachined applications," *The 1998 IEEE Asia-Pacific Conference on Circuits and Systems (IEEE APCCAS),* pp. 723-726, 24-27 Nov 1998.

[124] Yazdi, N., Mason, A., Najafi, K., Wise, K., "A generic interface chip for capacitive sensors in low-power multi-parameter microsystems," *Sensors and Actuators A: Physical,* vol. 84, no. 3, pp. 351-361, 2000.

[125] Lotters, J., Olthuis, W., Veltink, P., Bergveld, P., "A sensitive differential capacitance to voltage converter for sensor applications," *IEEE Transactions on Instrumentation and Measurement,* vol. 48, no. 1, pp. 89-96, Feb. 1999.

[126] Shiah, J.; Rashtian, H.; Mirabbasi, S., "A low-noise high-sensitivity readout circuit for MEMS capacitive sensors," *Proceedings of 2010 IEEE International Symposium on Circuits and Systems (ISCAS)*, pp. 3280-3283, May 30 2010-June 2 2010.

[127] Jawed, S.A., Cattin, D., Gottardi, M., Massari, N., Baschirotto, A., Simoni, A., "A 828μW 1.8V 80dB dynamic-range readout interface for a MEMS capacitive microphone," *34th European Solid-State Circuits Conference, 2008. ESSCIRC*, pp. 442-445, 15-19 Sept. 2008.

[128] Lee, H., Woo, J.-K., Kim, S., "CMOS differential-capacitance-to-frequency converter utilising repetitive charge integration and charge conservation," *Electronics Letters*, vol. 46, no. 8, pp. 567-569, 15 April 2010.

[129] Tavakoli, M., Sarpeshkar, R., "An offset-canceling low-noise lock-in architecture for capacitive sensing," *IEEE Journal of Solid-State Circuits*, vol. 38, no. 2, pp. 244-253, Feb. 2003.

[130] Horowitz, P., Hill, W., The art of electronics, Cambridge: Cambridge University Press, 1989.

[131] Oppenheim, A., Schafer, R. W., Discrete Time Signal Processing, Upper Saddle River, New Jersey: Prentice Hall Signal Processing, 2010.

[132] The MathWorks, Inc., "HDL Coder," [Online]. Available: http://www.mathworks.co.uk/products/hdl-coder/. [Accessed 10 June 2014].

[133] Mentor Graphics, "ModelSim: ASIC and FPGA design," [Online]. Available: http://www.mentor.com/products/fv/modelsim/. [Accessed 2 Feb. 2014].

[134] Sari, I., Zeimpekis, I., Kraft, M., “A dicing free SOI process for MEMS devices,” *Microelectronic Engineering Journal*, vol. 95, p. 121–129, July 2012.

[135] ROHDE&SCHWARZ, “R&S®FSV Signal and Spectrum Analyzer - Overview - Rohde & Schwarz,” Rohde & Schwarz, [Online]. Available: http://www.rohde-schwarz.com/en/product/fsv-productstartpage_63493-10098.html. [Accessed June 2014].

[136] Linear Technology, “LTC1412 - 12-Bit, 3Msps, Sampling A/D Converter,” [Online]. Available: http://www.linear.com/product/LTC1412. [Accessed 14 Nov 2013].

[137] Xilinx Inc., [Online]. Available: http://www.xilinx.com/. [Accessed 03 March 2014].

[138] Microchip Technology, “PIC18F87J60 PIC18 Microcontroller,” [Online]. Available: http://www.microchip.com/wwwproducts/Devices.aspx?product=PIC18F87J60. [Accessed 15 Feb 2014].

[139] El-Sheimy, N., Hou, H., Niu, X., “Analysis and modeling of inertial sensors using allan variance,” *IEEE Transactions on Instrumentation and Measurement*, vol. 57, no. 1, pp. 140-149, Jan. 2008.

[140] Haupt, R. L., Haupt, S. E., Practical genetic algorithms, New York: John Wiley and Sons, Inc., 2004.

[141] Mitchell, M., An introduction to genetic algorithms, Cambridge, MA - USA: MIT Press, 1998.

[142] Corana, A., Marchesi, M., Marini, C., Ridella, S., "Minimizing multimodal functions of continuous variables with the "simulated annealing" algorithm," *ACM Transactions on Mathematical Software,* vol. 13, no. 3, pp. 262-280, Sept. 1987.

[143] Sexton, R. S., Dorsey, R. E., Johnson, J. D., "Optimization of neural networks: a comparative analysis of the genetic algorithm and simulated annealing," *European Journal of Operational Research,* vol. 114, no. 3, pp. 589-601, May 1999.

[144] The MathWorks, Inc, "Genetic Algorithm," [Online]. Available: http://www.mathworks.co.uk/discovery/genetic-algorithm.html. [Accessed 10 June 2014].

[145] The MathWorks, Inc, [Online]. Available: http://uk.mathworks.com/discovery/simulated-annealing.html. [Accessed 10 Jan 2015].

[146] Fernando, P.R., Katkoori, S., Keymeulen, D., Zebulum, R., Stoica, A., "Customizable FPGA IP core implementation of a general-purpose genetic algorithm engine," *IEEE Transactions on Evolutionary Computation,* vol. 14, no. 1, pp. 133,149, Feb. 2010.

[147] Fargabg-Boroujeny, B., Adaptive filters: Theory and applications, Chichester, UK: John Wiley & Sons, Ltd, 2013.

[148] Wu, J., Fedder, G.K., Carley, L.R., "A low-noise low-offset capacitive sensing amplifier for a 50-μg/√Hz monolithic CMOS MEMS accelerometer," *IEEE Journal of Solid-State Circuits,* vol. 39, no. 5, pp. 722 - 730, May 2004.

[149] Ye, Z., Yang, H., Yin, T., Huang, G., Liu, F., "High-performance closed-loop interface circuit for high-q capacitive microaccelerometers," *IEEE Sensors Journal,* vol. 13, no. 5, pp. 1425-1433, May 2013.

[150] Petkov, V.P., Balachandran, G.K., Beintner, J., "Fully differential charge-balanced accelerometer for electronic stability control," *IEEE Journal of Solid-State Circuits,* vol. 49, no. 1, pp. 262-270, Jan. 2014.

[151] Shiah, J., Mirabbasi, S., "A 5-V 290- μW low-noise chopper-stabilized capacitive-sensor readout circuit in 0.8- μm CMOS using a correlated-level-shifting technique," *IEEE Transactions on Circuits and Systems II: Express Briefs,* vol. 61, no. 4, pp. 254-258, April 2014.

[152] Yazdi, N., Najafi, K., "An interface IC for a capacitive silicon μg accelerometer," *IEEE International Solid-State Circuits Conference, Digest of Technical Papers. ISSCC,* pp. 132-133, 16 Feb. 1999.

[153] Si-Ware systems, "Configurable high performance interface ASIC for MEMS inertial sensors (SWS1110)," Si-Ware Egypt, [Online]. Available: http://www.si-ware.com/sws1110/. [Accessed 5 Sept 2014].