UNIVERSITY OF
Southampton

University of Southampton Research Repository
ePrints Soton

Copyright © and Moral Rights for this thesis are retained by the author and/or other copyright owners. A copy can be downloaded for personal non-commercial research or study, without prior permission or charge. This thesis cannot be reproduced or quoted extensively from without first obtaining permission in writing from the copyright holder/s. The content must not be changed in any way or sold commercially in any format or medium without the formal permission of the copyright holders.

When referring to this work, full bibliographic details including the author, title, awarding institution and date of the thesis must be given e.g.

AUTHOR (year of submission) "Full thesis title", University of Southampton, name of the University School or Department, PhD Thesis, pagination

http://eprints.soton.ac.uk

UNIVERSITY OF SOUTHAMPTON

Automated Synthesis of Mixed-Technology MEMS Systems with Electronic Control

by

Chenxu Zhao

A thesis submitted in partial fulfillment for the
degree of Doctor of Philosophy

in the
Faculty of Engineering, Science and Mathematics
School of Electronics and Computer Science

December 2010

UNIVERSITY OF SOUTHAMPTON

ABSTRACT

FACULTY OF ENGINEERING, SCIENCE AND MATHEMATICS
SCHOOL OF ELECTRONICS AND COMPUTER SCIENCE

Doctor of Philosophy

AUTOMATED SYNTHESIS OF MIXED-TECHNOLOGY MEMS SYSTEMS WITH ELECTRONIC CONTROL

by Chenxu Zhao

Micro-Electro-Mechanical Systems (MEMS) design requires an integration of elements from two or more disparate physical domains: mechanical (translational, rotational, hydraulic), electrical, magnetic, thermal, etc. Different parts of a MEMS system are traditionally designed separately, using different methodologies and different tools applied to different energy domains. Although major Hardware Description Languages (HDLs) such as VHDL, Verilog and SystemC have been supplemented with analogue and mixed-signal (AMS) extensions which are essential in analogue and mixed-technology design, development of corresponding analogue and mixed-technology synthesis methodologies is still lagging behind. Therefore, there is an increasing need for automated synthesis techniques that can reduce the development cycle and facilitate the generation of optimal configurations. This research investigates and develops techniques for automated high-level performance optimisation and synthesis of mixed-technology MEMS systems.

Results of this research have been published in 9 papers at peer reviewed international conferences and one two-part journal paper. Specific contributions of this research can be summarised as follows. Firstly, a dedicated distributed model of a mixed-technology MEMS case study of an accelerometer operating in a Sigma-Delta force-feedback control scheme is developed. The distributed behaviour is essential in the MEMS accelerometer design because it has been observed that sense finger resonance, usually not included in conventional models, affects the performance of the electromechanical Sigma-Delta feedback control. As shown in the simulation results, the Sigma-Delta loop failure, when the sense fingers bend seriously or oscillate, is captured by the proposed model but cannot be correctly modelled using conventional approach.

Secondly, a novel, holistic approach is proposed for automated optimal layout synthesis of MEMS systems embedded in electronic control circuitry from user-defined high-level performance specifications and design constraints. The synthesis technique has been implemented in SystemC-A and named SystemC-AGNES. The method efficiently, and in an automated manner, generates suitable layouts of mechanical sensing element and configurations of the Sigma-Delta control loop by combining primitive components stored in a library and optimising them according to user specifications. Synthesis results show that the proposed technique explores the configuration space effectively, and it develops new structures which have not been investigated before. This contribution has been published as a two part paper in the Sensors & Transducers Journal.

Finally, to enhance the modelling efficiency and capability of SystemC-A, for mixed-technology systems with crucial distributed behaviour, language extension has been proposed to efficiently support general partial differential equations(PDEs) modelling.

Contents

List of Figures

List of Tables

List of Listings

Nomenclature

$MEMS$	Micro-Electro-Mechanical Systems
HDL	Hardware Description Language
AMS	Analogue and Mixed-Signal
PDE	Partial Differential Equation
$ODAE$	Ordinary Differential and Algebraic Equation
IC	Integrated Circuit
FEA	Finite Element Analysis
CAD	Computer-Aided Design
EDA	Electronic Design Automation
FDA	Finite Difference Approximation
RF	Radio Frequency
FET	Field Effect Transistors
$MESFET$	Metal Semiconductor Field Effect Transistor
VHF	Very High Frequency
DLP	Digital Light Projection
DMD	Digital Micromirror Devices
LCD	Liquid Crystal Display
$SPICE$	Simulation Program with Integrated Circuit Emphasis
DAC	Digital to Analogue Converter
ADC	Analogue to Digital Converter
STF	Signal Transfer Function
NTF	Noise Transfer Function
$PRSA$	Parallel Recombinative simulated annealing
CIF	Caltech Intermediate Format
$MOGA$	Multi-Objective Genetic Algorithm

Academic Thesis: Declaration Of Authorship

I, ...Chenxu Zhao... [please print name]

declare that this thesis and the work presented in it are my own and has been generated by me as the result of my own original research.

[title of thesis] ...Automated Synthesis of Mixed-Technology MEMS Systems with Electronic...

....Control...

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. Either none of this work has been published before submission, or parts of this work have been published as: [please list references below]:

Signed: ....

Date: ....

Acknowledgements

I would like to express my gratitude to all those who helped me during my research. Firstly, my enormous thanks must go to my supervisor, Dr. Tom Kazmierski, who has continued to offer me brilliant academic supervision and guidance through the years. Also, I would like to express my sincere gratitude to Dr. Peter Wilson who gave me extremely valuable feedback on my work in transfer viva.

I would like to say thank you to my parents, who have always been there for me, especially when I experienced difficulties and became frustrated with the work at hand. A special acknowledgement should be shown to my amazing fiancée Jingli Liu. I could not have achieved so much without your love.

I am also very grateful to Mr. Brian Nealon, who has been very patient and generous, offering endless support with my paperwork.

Finally, I want to mention my fantastic colleagues and friends: Chuan Bai, Dafeng Zhou, Ruiqi Chen, Leran Wang, He Yu, Yao Lu, Junwen Wang, and Kai Ying; your companionship and support has been everlasting ,and for that you will remain my best friends forever.

To my parents.

Chapter 1

Introduction

1.1 Background and motivation

The complexity of MEMS systems originates primarily from the complicated coupling relationships between different energy domains, i.e. mechanical, electrical, magnetic, thermal, etc. Despite the strongly coupled nature of mixed-technology MEMS systems, different parts of such systems are traditionally designed separately using different methodologies and different tools that are applied to different energy domains.

Traditionally, two engineering teams collaborate to create a MEMS-based IC: one team uses a Finite Element Analysis (FEA) based CAD such as CoventorWare [1] to create the MEMS mechanical model, and the other team, meanwhile, uses an EDA tool from electronics CAD vendors such as Cadence to create the associated ICs. This hybrid modelling approach is very common in MEMS design [2]. Although this approach provides accurate behaviour simulation of MEMS devices with their associated electronics, it requires multiple tools and it is difficult to provide IC designers with an automated synthesis and performance optimisation system. This difficulty is primarily caused by disparities between the different tools and the inconvenience of generating new MEMS macromodels, when the the layouts of MEMS devices change, for incorporation into the IC simulations performed at the IC design stage. This makes existing MEMS design methodologies very inefficient and leads to extensive and therefore time-consuming design iterations.

Analogue and Mixed-Signal (AMS) Hardware Description Languages (HDLs) such as VHDL-AMS which was standardised by the IEEE in 1999 [3] and later equipped with another IEEE standard for multiple energy domain packages [4, 5] and SystemC-A [6], are able to integrate components from different energy domains into a single model. Several AMS HDLs based MEMS models have already been reported in the literature, such as a yaw rate sensor [7] and a vibration sensor array [8]. However, automated design methodologies for the whole integrated system supporting mixed physical domains are still lagging. This is mainly due to the fact that state-of-the-art tools supporting AMS HDLs such as the commonly used SystemVision from Mentor Graphics [9] are not designed to support simulation-based synthesis and optimisation that allow users to develop and implement complex numerical algorithms. Wang proposed a methodology to realise a genetic optimisation algorithm (GA) in a VHDL-AMS testbench [10], but the software tools used took about 16 hours to complete a simple task.

Usually, the design of a MEMS system requires a significant amount of specialist human resources and time in the iterative trial-and-error design process to determine the crucial trade-offs in meeting performance specifications. As a result, there is an increasing need for automated synthesis techniques that would shorten the development cycle and facilitate the generation of optimal configurations for a given set of performance and constraint guidelines. Some methodologies have already been proposed for the automated synthesis of mechanical parts in MEMS systems [11, 12, 13, 14, 15]. In those approaches, the automated design of MEMS is accomplished either by simulation-based optimisation or formulating the design requirements as a numerical nonlinear constrained optimisation problem, and solved with powerful optimisation techniques. However, these methodologies are constrained to the layout synthesis of a mechanical MEMS device without considering its associated electronics [2].

This research aims to develop a high-level topology synthesis methodology for mixed-technology MEMS systems based on HDLs. Although major HDLs with AMS extensions such as VHDL-AMS and SystemC-A are very powerful and flexible mixed physical domain modelling tools, they still face a challenge when modelling MEMS-related applications. This is because current HDLs can only describe an analogue system with ordinary differential and algebraic equations (ODAEs). Support for partial differential equations (PDEs) are intentionally omitted in the development of major AMS HDLs due to the complexity of underlying numerical techniques [16]. This limits accurate modelling of MEMS devices that have distributed physical behaviour that play vital roles in the system performance because of the devices' small sizes. Thus, implementation of PDEs in major AMS HDLs has become increasingly attractive [17, 18, 19].

1.2 Research aims and contributions

The primary aim of this research is to investigate and develop techniques for automated high-level synthesis and performance optimisation of mixed-technology MEMS systems to match the rapid development of MEMS technology. The contributions for this research are:

- **An accurate distributed model of a MEMS accelerometer with sense finger dynamics.** This contribution presents an approach to modelling distributed physics effects of MEMS devices with VHDL-AMS and SystemC-A to accurately predict the performance of critical mechanical components. A surface micromachined capacitive MEMS accelerometer with a Sigma-Delta control scheme is used as a case study to demonstrate the methodology. With such an accelerometer, it is well known that the sense finger resonance in the mechanical sensing element affects the performance of the electromechanical Sigma-Delta feedback loop [20]. However, the conventional approach normally applied in simulations of such systems, where a second-order ODE is used to model the mechanical sensing element, cannot capture the effect of sense finger dynamics. The distributed model is achieved by the spatial discretisation of PDEs using a Finite Difference Approximation (FDA) approach that leaves the time derivatives intact to be handled by the VHDL-AMS or SystemC-A analogue solver. The number of discretisation points is a critical parameter which determines the accuracy of the behaviour of the distributed system. A series of simulation experiments was carried out to determine the minimum required number to correctly reflect the Sigma-Delta loop failure when the fingers bend seriously or oscillate. The analysis provides modelling guidelines to facilitate correct trade-offs in calculating the sense finger length when designing practical MEMS accelerometers based on an electromechanical Sigma-Delta control loop. Two papers describing this contribution were published at international conferences: BMAS 2007 (Behavioral Modeling and Simulation Conference) and FDL 2009 (Forum on Specification & Design Languages).

- **Automated synthesis of MEMS systems with associated electronic control system.** This contribution presents a holistic methodology for automated optimal synthesis of MEMS systems embedded in electronic control circuitry from user-defined high-level performance specifications and design constraints. The proposed approach is based on a simulation-based optimisation where the genetic-based synthesis of both mechanical layouts and associated electronic control configurations is coupled with calculations of optimal design parameters. The proposed genetic-based synthesis technique has been implemented in SystemC-A, and is named SystemC-AGNES. A practical case study of an automated design of a capacitive MEMS accelerometer with Sigma-Delta control demonstrates the operation of the SystemC-AGNES platform. The results of the synthesis show that the proposed approach can effectively explore the design space and obtain the optimal solution according to predefined performance specifications. Three conference papers related to this contribution have been published at international conferences: BMAS 2008, ISCAS 2010 (International Symposium on Circuits and Systems), and ICIA 2010 (International Conference on Information and Automation). A two-part journal paper which outlines the proposed synthesis approach has been published in the Sensor & Transducer Journal.

- **PDE extension for SystemC-A.** The current version of SystemC-A can only support the calculation of ordinary derivatives with respect to time, and faces difficulties when applied to the modelling of complex systems with distributed physical effects. This contribution proposes a syntax extension for SystemC-A to enhance the ability to support the modelling of PDEs. The efficiency of this new approach has been investigated by the modelling and simulation of two case studies. A paper describing the contribution has been accepted by the DATE'2011 (Design, Automation and Test in Europe) conference.

1.3 Thesis structure

This thesis is composed of six chapters. Chapter 2 provides a review of related literature. It covers state-of-the-art MEMS modelling and synthesis techniques. Chapter 3 presents an initial manual design of a surface micromachined MEMS accelerometer with electrostatic Sigma-Delta control scheme. Additionally, an accurate distributed model of mechanical sensing element is proposed and implemented both in VHDL-AMS and SystemC-A. This model includes the sense finger dynamics effect. This ensures that the system makes correct behaviour predictions. Chapter 4 presents a genetic-based synthesis environment in SystemC-A named SystemC-AGNES for MEMS sensors design. Not only the mechanical layout of the sensing element, but also the configuration of associated electronic control are synthesised and optimised synchronously to find the optimal design. Chapter 5 presents a syntax extension to SystemC-A to provide support for PDEs modelling. Finally, Chapter 6 concludes the research contributions and provides directions for future research.

Chapter 2

Literature review

Section 2.1 of this chapter demonstrates the broad range of design innovation and applications of MEMS devices. Section 2.2 briefly reviews relevant simulation and modelling tools. Section 2.3 discusses the literature related to the operation principle, various sensing mechanisms, and operation modes of MEMS accelerometers. The MEMS accelerometer is one of the most sophisticated types of MEMS sensors, providing high production volumes. A surface micromachined capacitive MEMS accelerometer is used as the case study in this research. The latest synthesis approaches for MEMS are reviewed in Section 2.4. Finally, Section 2.5 concludes this chapter.

2.1 Introduction to MEMS

The term Micro-Electro-Mechanical Systems (MEMS) refers to the microfabrication technology which integrates mechanical and electrical components [21]. The field of MEMS entered a period of rapid and dynamic growth in the early 1990s, and currently MEMS systems are used in a wide range of applications due to their significant advantages, such as low cost, small size and low power consumption [22]. Examples of MEMS devices include MEMS inertia sensors [23, 22], Radio Frequency (RF) MEMS [24, 25, 26, 27], Optical MEMS [28], and bioMEMS [22].

1. MEMS inertia sensors

MEMS inertia sensors, consisting of accelerometers and gyroscopes, are widely used in consumer applications, mainly by the automotive industry, for example: in air bag release systems, alarm systems, active suspension or anti-lock brake systems. Modern high precision inertial navigation and guidance systems are also based upon MEMS sensors embedded in mixed-technology control loops [23]. Because MEMS inertia sensors can be inserted into tight spaces, they can be used in novel applications because of their small size. Applications include smart writing instruments, virtual-reality headgears, computer mouses(gyro mouses), electronic game controllers, etc [22].

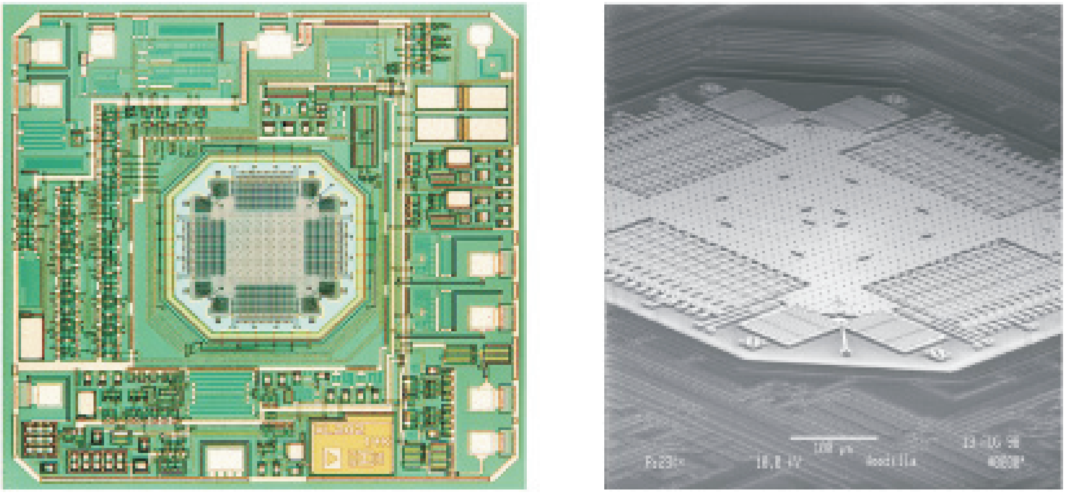


FIGURE 2.1: ADXL202: A fully integrated surface-micromachined dual axis accelerometer from Analog Devices[23].

A notable example of a MEMS inertia sensor is the ADXL series accelerometer developed by Analog Devices for the automotive market [29, 23]. This accelerometer consists of a suspended mechanical sensing element and signal-processing electronics integrated on the same substrate (Figure 2.1 [23]). The mechanical sensing element, which is based on capacitive sensing, is a suspended proof mass attached by many movable sense fingers. Each of the sense fingers is surrounded by two fixed fingers to form a differential capacitance pair. If acceleration is applied to the chip, the proof mass will move under an inertial force against the chip frame. The sense fingers move with the proof mass leading to the change of differential capacitance, which is read using on-chip signal-processing electronics.

A micromachined gyroscope is essentially an acceleration sensor that measures the angular velocity of an object by vibrating a proof-mass attached to the object and measuring its Coriolis acceleration [23]. Figure 2.2 shows a monolithic, surface-micromachined, vibratory gyroscope that is sensitive to rotations about the axis to the plane of the chip [30]. The gyroscope was fabricated by Sandia National Labs in an integrated surface micromachined MEMS process with a 2.25mm thick mechanical polysilicon layer and 2mm minimum gate length CMOS transistors [30]. To improve sensor bandwidth, linearity, and sensitivity to process and temperature variations, the sensing element of the gyroscope contains a Sigma-Delta force feedback control scheme.

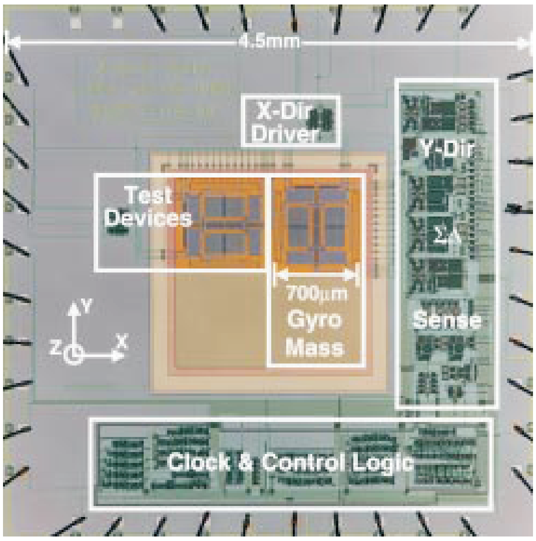


FIGURE 2.2: Photograph of the gyroscope die(Copied with permission)[30]

2. RF MEMS

RF MEMS encompass innovative components for RF wireless communication applications. RF MEMS components, including RF switches and relays, resonators, varactors (tunable capacitors), microintegrated inductors and filters, offer significant benefits compared to conventional RF components in terms of power consumption and cost [27].

One of the most popular RF MEMS devices, which is the essential component for RF reconfigurability, is the RF MEMS switch [31]. Since the first membrane-based MEMS switch was reported as early as 1979 [32], there has been a great deal of literature on the development of RF MEMS switches as a basic building block for more complex applications [33, 34]. In a typical RF integrated circuit, semiconductor switches such as FET and PIN diode switches are widely used. However, when the signal frequency becomes greater than 1 GHz, these typical semiconductor switches generally have many disadvantages, such as great insertion loss, poor electrical insulation, and high power consumption [31]. Compared with those traditional switches, RF MEMS switches exhibit promising characteristics [35, 31, 36]. For example, a commercial MESFET provides about 0.9 dB insertion loss which by itself consumes about 19% of generated RF power, while a MEMS switch could provide 0.2 dB insertion loss which would reduce the power loss to 4.5% [36].

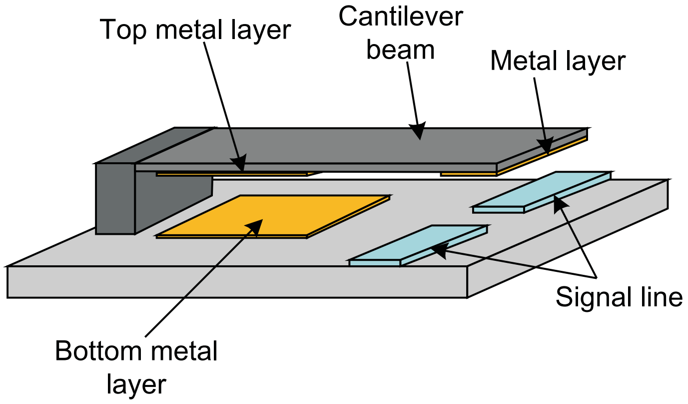


Figure 2.3: A typical cantilever RF MEMS switch structure

A typical cantilever beam RF MEMS switch structure is shown in Figure 2.3. The MEMS cantilever is fixed on one end, and is covered with a metal-layer to open or connect the microwave signal line on the free end. In addition, there is another metal-layer in the middle of cantilever beam that is suspended over a bottom metal contact to form a capacitor. When a bias voltage is applied between the contacts, the resulting electrostatic force makes the cantilever beam bend down towards the bottom contact. When the applied voltage reaches a certain threshold, the metal layer connects the signal line. If the magnitude of the voltage is reduced, the cantilever releases the metal layer and disconnects the signal line.

Another attractive example of an RF MEMS component is the micromechanical resonator [37]. It is emerging as a potential candidate for a variety of wireless communication applications because of its advantages. These advantages include: its tiny size, virtually zero DC power consumption, and the use of IC-compatible fabrication technologies to enable on-chip integration of MEMS resonators with transistor electronics. For example, up-to-date, clamped-clamped [38] and free-free [39] flexural-mode beams MEMS resonators with high quality factor (on the order of 10,000) have been popular in VHF range communication applications [37].

3. Optical MEMS

Optics is one of the earliest and most active areas in which MEMS technology has been applied [40]. This is because the efficient merging of optical, MEMs and microelectronic systems offers a significant potential for microoptoelectromechanical systems (MOEMs) in display and communications applications [28, 41, 42].

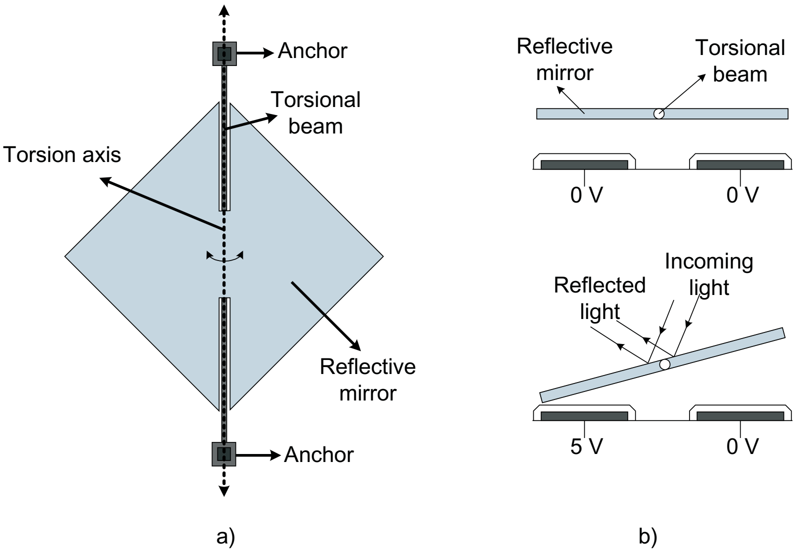


FIGURE 2.4: Structure of a single digital micromirror from Texas Instruments [22]. a) Top view; b)Cross-sectional view

The most notable example is the Digital Light Projection (DLP) display, which is a powerful technology for digital multimedia presentation in movie theater systems [22].

It is based on MEMS Digital Micromirror Devices(DMD), invented in 1987 by Larry Hornbeck of Texas Instruments [28]. The DMD is comprised of a rectangular array of up to two million individually addressable microscopic mirrors with an approximate area of $10 * 10um^2$. Figure 2.4 shows a schematic of a micromirror where the mirror is supported by two torsional beams and can rotate with respect to the torsion axis. The electrodes under the mirror are used to control its position by electrostatic attraction force. Consequently, the mirror can reflect light towards the screen and illuminate one pixel when placed at the correct angle. The DLP projection display offers advantages over the traditional Liquid Crystal Display (LCD) technology in terms of pixel fill factor, brightness, black level, and stability of color balance [22].

4. BioMEMS

Because of miniaturisation and rich functional integration, BioMEMS are also becoming popular for medical applications such as microfabricated neuron probes in neurobiological studies, drug injection needles, and physiological sensors [22, 43, 44].

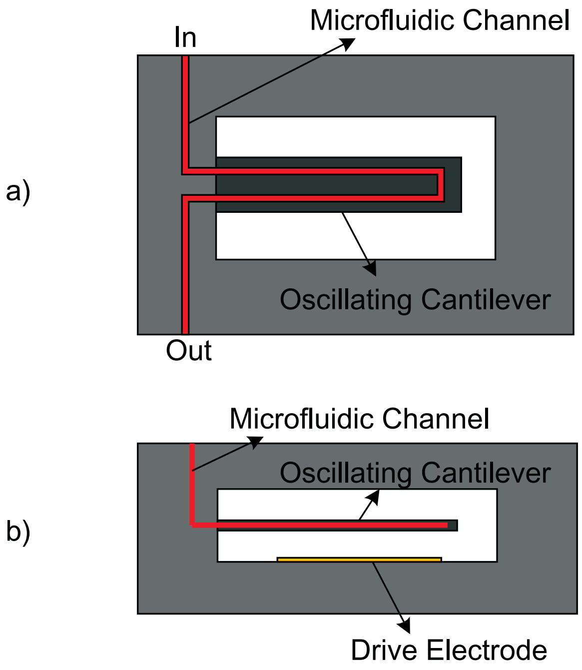


Figure 2.5: Vacuum-packaged suspended microchannel resonant mass sensor for detecting biomolecular materials in fluid Streams[43].a)top view; b)side view

An excellent example of a BioMEMS application is a microchannel resonant mass sensor which is intended to detect biomolecules in a microfluidic format [43]. It consists of a microchannel fabricated on a suspended cantilevered beam. The inside wall of the channel is treated to bond to the biomolecular substance of interest. An electrostatic drive causes the cantilever beam to oscillate at its resonant frequency. As biomolecular material accumulates in the microchannel, its mass increases, thus lowering the resonant frequency. A schematic illustration of this device appears in Figure 2.5.

2.2 MEMS Simulation and Design Tools

The simulation of MEMS systems is used to virtually build the device and predicts their behaviour before fabrication [45]. It shortens the development cycle considerably and reduces the cost of developing a commercial device. This is because various parameters in the virtual model can be changed more quickly than actually fabricating a prototype and redesigning [45]. However, simulation of MEMS systems is a challenging task because of the presence and interactions of multi-physical domains. Any MEMS design and modelling tool can be classified into two categories according to their design methodology:

- **FEA-based modelling:** This approach refers to using highly efficient and accurate numerical solvers, such as the Finite Element Analysis (FEA) method, for dealing with the equations of physics governing system behaviour. It is able to analyse complex geometries by subdividing them into a finite number of elements, and it is quite suitable to deal with complex differential equations with boundary conditions; hence, it is a commonly-used methodology for simulating various engineering applications. Many commercial MEMS CAD tools that use this technique are available, including CoventorWare [46] [47], ANSYS [48], SOLIDIS [49], etc. These tools provide more realistic simulation results than system-level modelling tools, but FEA-based tools are much more computationally demanding and not suitable for complete simulation of the MEMS systems with attendant electronics.

- **System-level (behavioral) modelling**: This is an attractive approach to predicting the main behaviour of MEMS systems in a reasonable amount of time. This approach uses system-level (behavioral) models to simplify complex physics and explore interaction among different domains [45]. System-level modelling tools involve Saber [50, 51], SPICE, Simulink and Hardware Description Languages (HDLs) with AMS extensions such as VHDL-AMS [52, 53, 54, 55], Verilog-AMS [56]. The multidomain problem is avoided in the block diagram-based system representation tools such as Simulink since they are typically physically dimensionless [45]. The HDLs, such as VHDL-AMS, are standard languages with the ability to support multiphysical domain modelling. Therefore, system-level modelling is quite suitable for designing MEMS mechanical components as well as associated electronics.

2.2.1 Finite Element Analysis (FEA) based MEMS design tools

2.2.1.1 CoventorWare

The CoventorWare suite of software tools, which is the most popular MEMS design toolset developed by Coventor, Inc., serves 70 percent of the global market [57]. It is a fully integrated MEMS design environment that is comprised of four major modules: **ARCHITECT, DESIGNER, ANALYZER** and **INTEGRATOR**. These modules can be jointly used to provide a complete MEMS design flow as illustrated in Figure 2.6.

ARCHITECT is a schematic-based system-level modelling environment that contains a comprehensive MEMS component library. **DESIGNER** is a physical design tool that generates three-dimensional (3-D) solid models of MEMS devices. **ANALYZER** does the 3-D physical simulation with best-in-class field solvers. It is the core of the ConventorWare. **INTEGRATOR** is used to extract system-level reduced-order model which can be directly inserted into system-level simulators such as Saber and Simulink. In the system-level simulators, the extracted models are then connected with the external electronics to perform the simulation of the entire MEMS system.

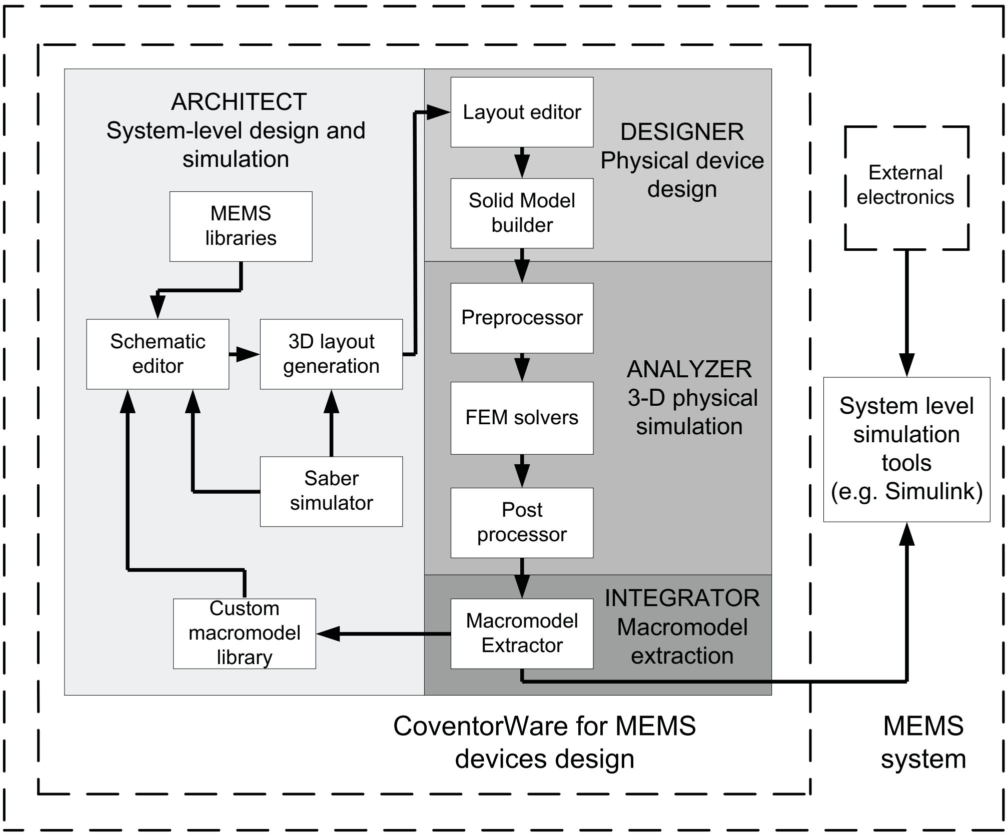


FIGURE 2.6: ConventorWare suite of software tools [57]

A number of papers have been published on the development of MEMS devices based on CoventorWare software. In the inertia MEMS sensors field, M. Webwer [1] analysed the effects of the high angular rates and high operating accelerations of a MEMS gyroscope, which is modeled in ARCHITECT. A.R. Sankar [58] used CoventorWare tools to analyse the temperature drift in a MEMS piezoresistive accelerometer. G. Gattiker [59] proposed an innovative design idea for a semi-invasive blood sampling, analysis and drug delivery bioMEMS device based on CoventorWare. CoventorWare is also used for RF MEMS devices design such as MEMS resonators [60], capacitors [61] and switches [62, 63].

2.2.1.2 Other FEA-based tools and design limitations

Although the FEA-based tools are quite suitable in designing MEMS mechanical components such as the mechanical sensing elements, simulation of the complete mixed-technology systems (e.g. inertial sensors with a Sigma-Delta control system) is restricted. Some FEA-based tools (such as ConventorWare, Ansys, FEMLAB, etc.) are capable of including circuits in their physics-based simulations. However, these capabilities are not yet at a level sufficient for modelling complex mixed-technology systems. This is especially true if the systems include digital and nonlinear analogue circuits [17].

Recent FEA-based tools are able to extract lumped behavioral (Reduced-Order) models which can be coupled to some system-level design tools for concurrently simulating mechanical components and associated ICs. These system-level design tools involve Saber [50, 51], SUGAR [64, 65, 66, 67, 68, 69], SPICE [70], and Simulink [71]. Although this hybrid approach allows design engineers to realise the co-design of micromechanical components and their surrounding IC components, it requires multiple design tools; this is inconvenient for generating macromodels of MEMS devices for incorporation into the IC simulations. Since this technique is also not suitable for use in the iterative optimisation design loop, it is difficult to provide IC designers with an automated synthesis and performance optimisation system.

2.2.2 System level modelling tools and HDLs

2.2.2.1 Simulink

Simulink, which is one of the most popular system-level modelling tools, is a toolbox within Matlab from Mathworks [71]. Simulink has a graphical interface in which users can simply build systems by connecting the chosen blocks from Simulink's library. The block library contains time continuous and discrete linear and nonlinear functions such as integrator, gain, s-domain transfer functions, mathematical functions and so on. Furthermore, Simulink supports a user-defined library which includes user-defined blocks [45].

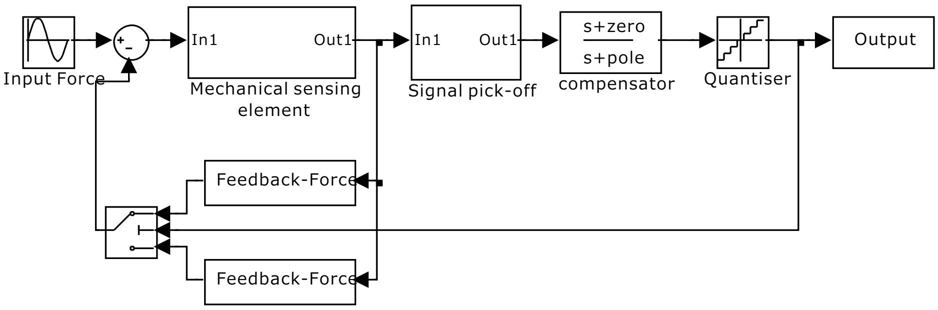


FIGURE 2.7: Simulink model of an accelerometer with Sigma-Delta force-feedback control. The model contains a mechanical sensing element, electronic Sigma-Delta control blocks and their interface [72]

Hierarchal modelling can be realised in Simulink by defining parameterised subsystems. For example, the Simulink model of an accelerometer with Sigma-Delta eletrostatic force-feedback control is shown in Figure 2.7 [72]. The model includes the mechanical sensing element, electronic signal pick-off blocks, and Sigma-Delta control blocks (A detailed description of the operation of such a digital accelerometer is explained in section 2.3). The block of the sensing element is a subsystem in the overall sensor system model, and it is shown in Figure 2.8.

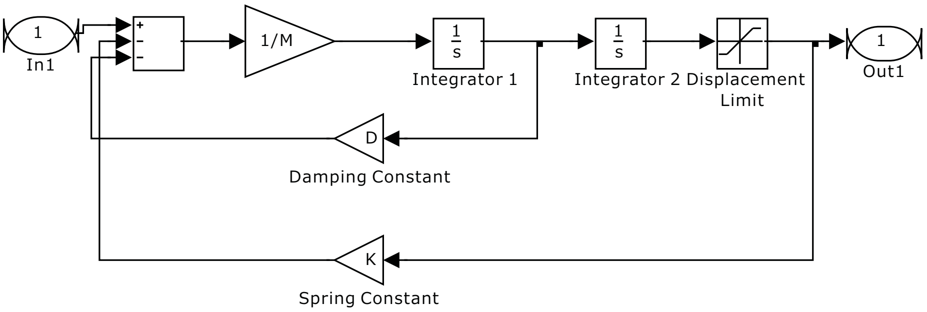


FIGURE 2.8: Simulink model of the sensing element of the digital accelerometer (mass-damping-spring system). Input is the external force and the output is the displacement of the inertial mass [73]

The mechanical sensing element model is treated as a suspended inertial mass with its motion damped by a dasher (mass-damping-spring system) [73]. The external force serves as the input of the sensing element. The restoring force from the spring is represented by multiplying the output displacement by the spring constant (Gain block with value K), while the damping force is obtained by multiplying the velocity of mass by the damping constant (Gain block with value D). The spring force and the damping force are subtracted from the input force to form the net force on the inertial mass. The net force is converted to the acceleration of the inertial mass after gain block (with value 1/M), displacement of the inertial mass is then obtained when the acceleration is integrated twice [73].

Simulink's main advantage is that the multidomain problem is avoided, since Simulink is physically dimensionless [45]. Thus, the MEMS sensor model, which includes the mechanical part and electronic control system, as well as their interface, can be easily simulated in a single environment [74]. Furthermore, the optimisation of many design parameters such as the mass, spring constant, and SNR can be realised by combining the Simulink model with other Matlab toolboxes, such as the GA toolbox.

2.2.2.2 SPICE

Although SPICE is an electronic circuit simulator, other physical domain components such as mechanical components can also be simulated using SPICE by mapping their domain quantities into equivalent electrical ones and developing an equivalent circuit [75, 76].

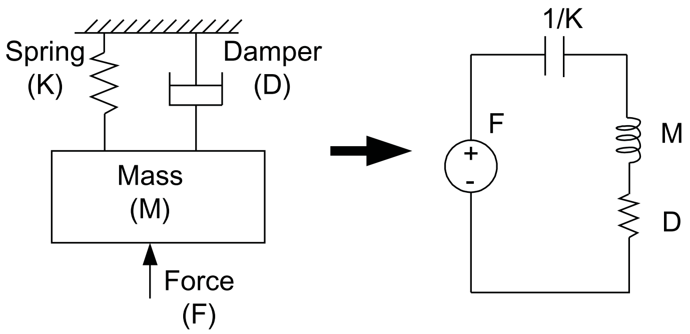


FIGURE 2.9: Equivalent circuit of the mass-damper-spring system[77]

For example, the mass-damper-spring subsystem, which is illustrated above, can be represented by the equivalent circuit shown in Figure 2.9 [77], where the inertial proof mass is represented by an electrical inductor (with inductance M); the spring is represented by a capacitor (capacitance 1/K); and the damper is represented by a resistor (resistance D). The force F is equivalent to the voltage; the velocity of the mass is equivalent to the current; while the displacement of the mass is analogous to the charge of the capacitor.

2.2.2.3 Hardware Description Languages (HDLs)

Nowadays, Hardware Description Languages (HDLs), such as VHDL, Verilog and SystemC, have been widely used to model and simulate digital electronic systems, and there is a trend to extend standard digital HDLs further by adding new language syntax elements to support mixed-signal and mixed-technology system modelling. The most popular HDLs with such AMS extension include VHDL-AMS [78] [79], Verilog-AMS [56] [80], SystemC-AMS and SystemC-A [81].

Among them, VHDL-AMS is the first to achieve the IEEE-approved standard, and it is extensively used in today's high-level system designs. SystemC-A, which was developed in ECS at University of Southampton in 2006, has been applied to complex simulation and modelling problems [6, 81]. More recently, in March 2010, Open SystemC Initiative (OSCI) released the AMS 1.0 standard for SystemC (SystemC-AMS) [82].

1. VHDL-AMS

VHDL-AMS, standardised as IEEE 1076.1-1999 [3], is a superset of the VHDL (IEEE standard 1076-1993 [83]). VHDL-AMS is one of the major AMS HDLs which supports modelling mixed digital and analogue components, as well as mixed electrical and non-electrical physical domains systems, at various abstraction levels [78].

VHDL-AMS can be used for the modelling and simulation of systems that contain discrete-event (digital) and continuous-time (analogue) signals [3]. Event-driven behaviour is modelled by concurrent processes that are sensitive to signal changes, while continuous-time models are implemented using ordinary differential and algebraic equations(ODAEs). Interactions between the discrete and the continuous parts of a model are supported in a flexible and efficient manner by VHDL-AMS [78].

VHDL-AMS provides new language elements (**Simultaneous Statement, Quantity, Terminal, Nature**) which facilitate the writing of analogue models that describe the behaviour of the system [78]. **Simultaneous statements** are a new class of statements in VHDL-AMS and are used for notating ODAEs. The values of any unknowns in the simultaneous statements are computed by an analogue solver. **Quantities**, which have time-continuous values with a finite number of discontinuities, represent the unknowns in ODAEs. Quantities can have several forms; they can be free quantities or interface quantities in the port list of a model to support signal flow modelling. Branch quantities represent the unknowns in the equations that describe conservative systems. There are two kinds of branch quantities: across quantities and through quantities. Across quantities represent effort-like effects, such as voltage and displacement; while through quantities represent flow-like effects, such as current and force. A branch quantity must be declared between two terminals. A **terminal** is a fixed node of a model which is declared to be of some physical nature such as electrical, thermal, mechanical, etc. **Nature** defines the types of across and through quantities incident to a terminal of the specified domain.

The ability of VHDL-AMS in modelling multiple energy domain systems is further enhanced by the IEEE VHDL 1076.1.1 standard [4, 5]. It defines a collection of VHDL 1076.1 packages that are compatible with IEEE 1076.1-1999 standard, along with recommendations for conforming use, in order to facilitate the interchange of simulation models of physical components and subsystems [5]. The packages include definitions of the most frequently used standard types, subtypes, natures, and constants for modelling in multiple energy domains [4, 5].

The IEEE 1076.1.1 packages can be divided into two classes: constant packages and energy domain packages. Constant packages define a set of basic physical constants (either fixed or user-defined), which allow models written using these packages to have a common basis for modelling physical systems. Energy domain packages define a set of types and natures that provide a common framework for modelling physical systems across a range of commonly used energy domains. The packages ensure that the interfaces are consistent, correct, and interoperable. [4, 5].

The VHDL-AMS, with IEEE 1076.1.1 standard, serves a broad class of applications. In the automotive industry, Fanucci et al. [84] presented a general architecture that was suitable for interfacing several kinds of sensors in automotive applications. In addition, a braking system was developed by Deligueta et al. [85]. At the bottom end, semiconductor device models for diodes and transistors have been developed in VHDL-AMS [86]. VHDL-AMS has also been used to design various MEMS systems, i.e. as MEMS sensors [87, 88, 89, 90, 91, 92], RF MEMS switches [93],RF MEMS Disk Resonator [94], MEMS harvesting systems [95], micromotors [96]. More examples of VHDL-AMS models can be found at the Southampton VHDL-AMS Validation Suite [97].

2. SystemC with AMS extensions

SystemC is a standardised HDL built based on C++ class libraries for the design and modelling of digital systems [98]. The first version of SystemC V0.9 was released and made available since the Open SystemC Initiative (OSCI) was announced at the Embedded Systems Conference in San Jose, California in 1999. After few revisions, the IEEE Standards Association approved the standard for SystemC language as IEEE 1666 standard on December 12, 2005 [98].

There have been many research results presented with the aim to extend SystemC to modelling AMS systems [81, 99, 100, 101]. An OSCI working group was established in 2003 [102] aiming to develop AMS extension to SystemC. In March 2010, OSCI released the AMS 1.0 standard for SystemC, named SystemC-AMS, which support modelling of embedded analog/mixed-signal applications at various levels of design abstraction.

SystemC-A, which was developed in ECS at the University of Southampton in 2006 [81], is a superset of SystemC developed to extend modelling capabilities of SystemC to the analogue and mixed-physical domain. In addition to standard digital modelling capabilities of SystemC, SystemC-A provides constructs to support user-defined ordinary differential and algebraic equations (ODAEs), analogue system variables, and analogue components to enable modelling of analogue and mixed-signal systems from very high levels of abstraction down to the circuit level. Support for digital-analogue interfaces is also provided for smooth integration of digital and analogue parts. The analogue simulator of SystemC-A uses efficient linear and nonlinear solvers to assure accurate and fast simulations of the analogue model. Most of the powerful features of VHDL-AMS are provided in SystemC-A in addition to a number of extra advantages such as high simulation speed and flexible data manipulation. SystemC-A has already been used to model mixed-signal systems, such as a switched-mode power supply [6], and mixed-physical domain systems, such as the automotive seating vibration isolation system [103]. The results of these applications prove that SystemC-A can be compared to well-established AMS HDLs such as VHDL-AMS [103].

2.3 MEMS Accelerometers

MEMS inertial sensors, which include accelerometers and gyroscopes, are a versatile group of sensors which can be applied widely in many areas. The MEMS accelerometer, which is presented in the upcoming section, is chosen as the case study in this project because it is one of the most important sensors in the MEMS field and has attracted significant interests since the first micromachined accelerometer was reported by Roylance et.al [104] in 1979.

The operation mode of the MEMS accelerometer can be either open loop or closed loop. Due to its inherent stability, simple electronic circuits interface and low cost, the open-loop accelerometer is attractive for a number of applications. Performance of the open-loop accelerometer relies entirely on the dynamics of the mechanical sensing element. Thus, the fabrication tolerances and nonlinear effects such as the spring softening effect limit the performance of the accelerometer. [45]

High-performance MEMS accelerometers exploit the advantages of the closed-loop control strategy to increase the dynamic range, linearity, and bandwidth of their sensors. In particular, Sigma-Delta modulators for closed-loop feedback control schemes, whose output is digital in the form of pulse-density-modulated bitstream, have become very popular in a number of MEMS applications [45, 105, 106].

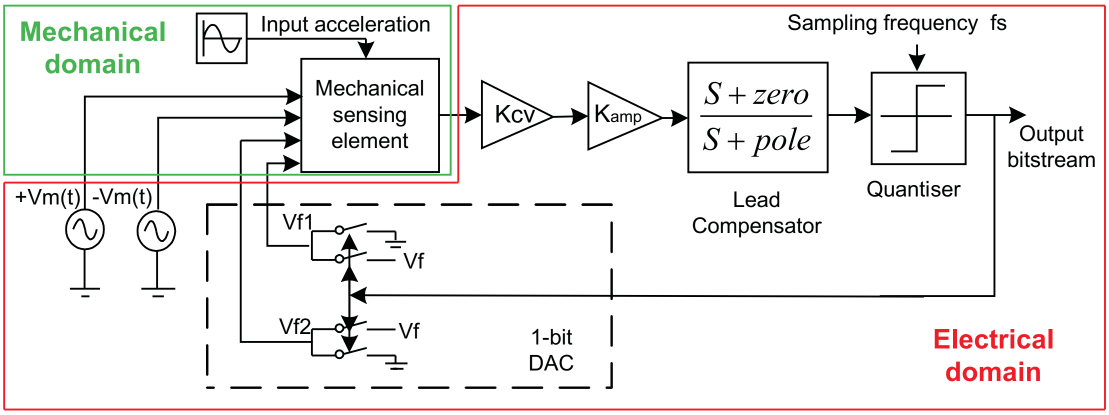


FIGURE 2.10: Second-order electromechanical Sigma-Delta accelerometer [106].

A conventional MEMS accelerometer with Sigma-Delta control scheme is shown in Figure 2.10 [106]. In this configuration, the mechanical sensing element is used as a loop filter to form the second-order electromechanical Sigma-Delta modulator. This is because the sensing element can be approximated by a second-order Mass-Damper-Spring transfer function which performs a similar function to that of two cascaded integrators in typical second-order electronic Sigma-Delta modulators. V_{f1} and V_{f2} are the feedback voltages obtained from the DAC, and $V_m(t)$ is a high frequency modulation carrier voltage. The gain K_{cv} represents the signal pick-off from mechanical domain to electrical domain, and K_{amp} is the gain of the voltage booster amplifier following the pick-off stage. The lead compensator ensures the stability of the control loop. It is an optional component depending on whether the sensing element is over-damped or under-damped. A one-bit quantiser is used for oversampling and generating a pulse-density modulated digital output signal.

2.3.1 MEMS accelerometer sensing mechanisms

Many sensing mechanisms for the MEMS accelerometer have been presented in the literature, and most of them first translate external acceleration into the displacement of the seismic mass and then convert the displacement to an electrical signal by changing certain physical properties. These techniques, based on sensing the displacement of the proof mass, are usually considered to be position sensing. The sensing mechanisms, such as piezoresistive [104] [107] [108] [109], piezoelectric [110] [111], capacitive [77], resonant [112] [113] and optical [114] [115] are all based on this position sensing technique. These mechanisms are categorised in Table 2.1 for comparison. Several common mechanisms used in MEMS sensors are discussed in this section. Among them, the capacitive sensing is of the primary interest in this research because it is one of the most commonly used sensing mechanisms in commercial accelerometers [45].

Sensing mechanism	***Measured signal***	***Features***	***Temperature drift***	***Sensitivity***	***Bandwidth***
Piezoresistive[109]	Resistance	Temperature dependent	$0.2\%/^{\circ}C$	2mV/G	1KHz
Capacitive[77]	Capacitance	Simple, low temperature drift	$150ppm/^{\circ}C$	38mV/G	10kHz
Resonant[113]	Frequency	High sensitivity	$45ppm/^{\circ}C$	1V/G	5kHz
Optical[114]	Light	Hand assembly	$0.05\%/^{\circ}C$	100mV/G	1kHz
Piezoelectric[111]	Voltage	Relative high sensitivity, complex fabrication	$0.03\%/^{\circ}C$	320mV/G	1Hz-200kHz

Table 2.1: MEMS accelerometer sensing mechanisms

2.3.1.1 Piezoresistive sensing

The first micromachined accelerometer, which was proposed by Roylance in 1979 [104], is based on the piezoresistive sensing mechanism. The mechanical sensing element is based on the bulk micromachined fabrication technique and contains a proof mass that is attached to the supporting frame through a cantilever beam as the suspension system. Piezoresistive material (piezoresistor) is placed on the upper surface of the cantilever beam to measure the out-of-plane acceleration of the proof mass. When an external acceleration is applied to the accelerometer, the proof mass moves and the cantilever bends, causing the strain experienced by the piezoresistor that leads to a change in its electrical resistance (piezoresistive effect). A relationship between acceleration and voltage can be derived by implementing Wheatstone bridge circuits to capture the resistance change of the piezoresistor.

The MEMS piezoresistive accelerometers are widely used due to the simplicity of their sensor structure, the fabrication process and the read-out circuits design. However, the main drawback of this sensing mechanism is that the output signal is strongly temperature dependent because the thermal noise is inherently generated by the piezoresistive material and the output signal is relatively small. It results in low resolution [107] [108].

2.3.1.2 Resonant sensing

The resonant accelerometer usually contains a proof mass attached by a mechanical resonator. When applying acceleration, the movement of the proof mass changes the strain of the resonator, thus leading to a change of its resonant frequency which will be measured. Many resonant sensors have been proposed in the literature. For example, Roessig et al. presented a surface micromachined resonant accelerometer [112]. The accelerometer consists of two double-ended tuning fork (DETF) resonators which are attached to a proof mass by a pivot beam. When the system is operating, the proof mass hinges about the beam and applies forces to the two DETFs. One of the resonators is subjected to a tensile force which raises its resonant frequency; while the other is subjected to a compressive force which decreases the resonant frequency. The difference of the resonant frequencies of the resonators is the output of the accelerometer [112]. The nominal resonant frequency of the DETF resonators reaches 68KHz, which leads the system to experience good sensitivity in terms of the change of frequency per acceleration. This is the major advantage of the resonant sensing technique. Furthermore, the frequency output of the system can be converted into digital form by applying a frequency counter [45].

2.3.1.3 Capacitive sensing

Among a variety of sensing mechanisms, capacitive sensing, which uses a capacitor to sense the deflection of the proof mass, is the dominant type in MEMS inertial sensors. Based on fabrication techniques, micromachined accelerometers can be classified into two main categories: bulk micromachined accelerometers and surface micromachined accelerometers. Early capacitive accelerometers were typically based on bulk micromachining fabrication with several wafers of the capacitive structure assembled by bonding techniques [45]. Figure 2.11 [23] shows a typical example of the bulk micromachined accelerometer. The middle wafer, which consists of the proof mass and suspension system, forms the capacitors with the top and bottom cap wafers(electrodes). The deflection of the proof mass changes the spacing between the electrodes of capacitors, leading to a differential change in capacitance, which can be measured easily. The bulk micromachined capacitive sensors have higher sensitivity and lower noise floor than the surface micromachined devices because they have much a larger mass and a larger sensing capacitance.

In recent years, surface micromachined MEMS accelerometers have gained much popularity because surface micromaching fabrication technique allows integration of sensing element with associated electronics on the same chip. Furthermore, the size of this class

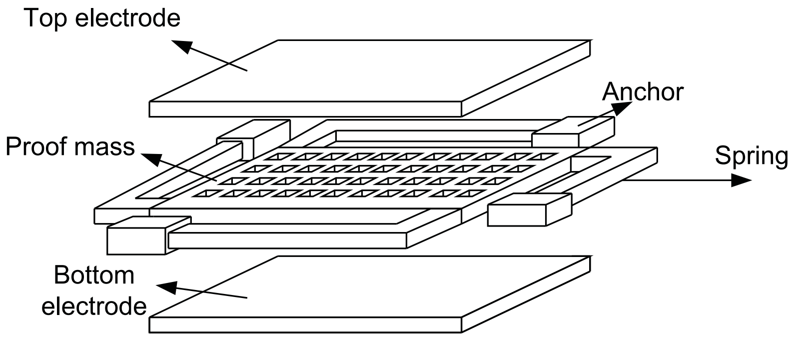


FIGURE 2.11: A bulk micromachined capacitive accelerometer [23]

of accelerometers is usually smaller than those bulk micromachined devices. The capacitive accelerometers fabricated by polysilicon surface micromaching technology have been successfully used in automotive applications.

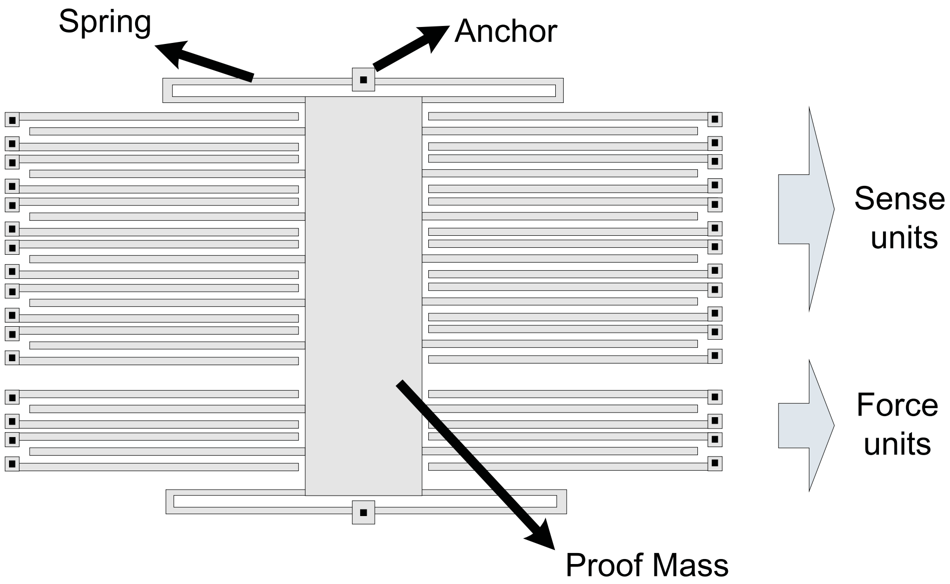


FIGURE 2.12: A schematic of surface micromachined capacitive accelerometer [116]

Figure 2.12 shows a typical design for a surface micromachined capacitive sensing element structure introduced by Sherman [116]. This structure is widely applied in ADXL series accelerometers made by Analog Devices. The sensing element consists of a proof mass suspended above a substrate by springs. The proof mass is equipped by a number of sense and force comb finger units. Each comb finger unit contains a movable sense or force finger (connected to the proof mass) that is placed between two fixed fingers. In closed-loop operation, a feedback voltage is applied to one of the fixed fingers in the force comb finger unit such that the resulting electrostatic force pulls the moving proof mass back to its original position. If the proof mass is equipped with sense fingers with number N_s, N_s differential capacitance bridges are formed by the sense comb finger units. When the mass deflects due to the external acceleration, the differential change in capacitance is expressed by the following equation (this assumes that the sense finger is a rigid body moving with the proof mass without bending):

$$C_{s1} - C_{s2} = N_s \varepsilon_0 A \left(\frac{1}{G - x} - \frac{1}{G + x}\right) \tag{2.1}$$

where C_{s1} and C_{s2} are differential capacitances, A is the area of the capacitance plates, ε_0 is the permittivity of free space, G is the initial space between sense finger and fixed fingers in a sense comb finger unit, and x is the relative displacement of the proof mass with respect to substrate.

There are many advantages of the capacitive sensing mechanism such as the good steady-state response, high sensitivity, low noise performance, low power dissipation, low temperature sensitivity and compatibility with VLSI technology scaling. The main drawback of the capacitive MEMS accelerometers is that they are susceptible to Electromagnetic Interference(EMI), but this issue can be resolved by using good packaging and shielding [45].

2.3.2 Interface circuit for capacitive sensing mechanism

The change in capacitance of the differential capacitive MEMS accelerometers is measured by the signal pick-off circuit, which is usually a charge amplifier. To demonstrate the operation, a single-ended charge amplifier is shown in Figure 2.13 [117]. In practical, differential charge amplifier is widely used to reject the undesired common mode interference such as switch charge injection and variations in the magnitude of the excitation voltage [118].

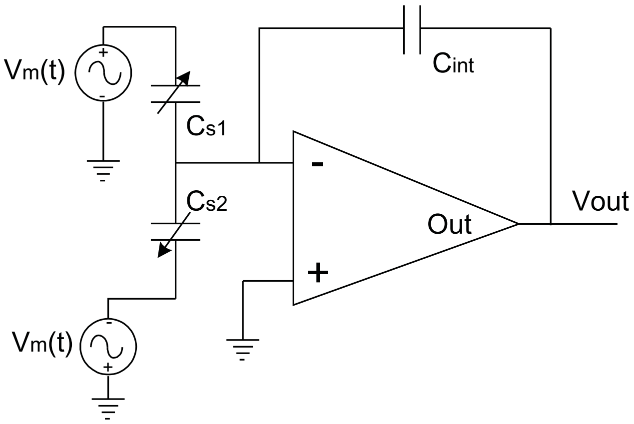


FIGURE 2.13: Pick-off circuit by applying a charge amplifier [117]

The variables C_{s1} and C_{s2} represent the sensing capacitors that have the same initial capacitance C_0. The high frequency excitation carrier voltage signal ($V_m(t)$) and antiphase signal are applied on the fixed finger electrodes of the sense comb finger units. The center electrode is connected to the negative input terminal of an operational amplifier. The modulated output voltage of the charge amplifier is given by:

$$V_{out} = -2V_m(t)\frac{C_{s1} - C_{s2}}{C_{int}} \tag{2.2}$$

Typically, C_{int} is set to $2C_0$, where C_0 is the initial capacitance of the variable capacitors. Thus, the output voltage can be calculated as:

$$V_{out}(t) = -\frac{Gx}{G^2 - x^2}V_m(t) \tag{2.3}$$

For small displacements, we can assume that $G^2 \gg x^2$. Hence, the output voltage becomes proportional to the deflection of the proof mass.

$$V_{out}(t) = -\frac{x}{G}V_m(t) \tag{2.4}$$

For the surface micromachined accelerometer with N_s sense fingers, the final modulated output voltage of the charge amplifier can be approximated by:

$$V_{out}(t) = -N_s \frac{x}{G} V_m(t) \tag{2.5}$$

The charge amplifier is followed by a demodulator, which recovers the original signal from the modulated voltage. Therefore, ideally, the interface circuit can be represented by an ideal gain block that relates the displacement of the proof mass to an electrical signal in the system-level model.

2.3.3 Operation principle of the mechanical sensing element

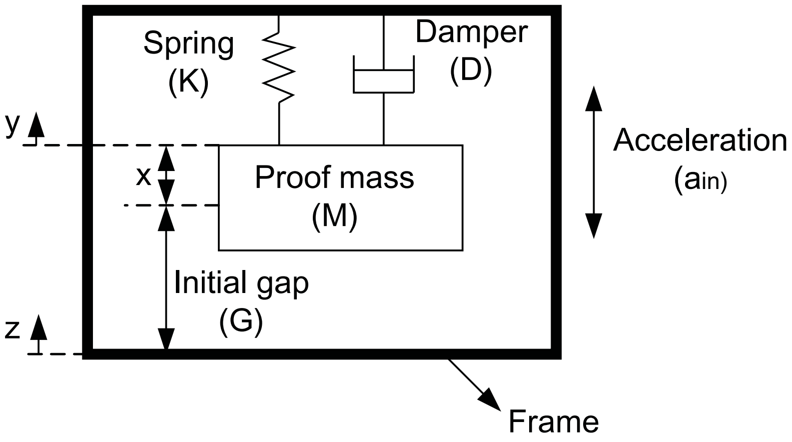


FIGURE 2.14: Mechanical sensing element model of an MEMS acceleroemter.

The measurement of acceleration always relies on classical Newton's mechanics. The mechanical sensing element model of a MEMS accelerometer is illustrated in Figure 2.14. As shown in the figure, a proof mass (M) is connected to the frame by a suspension spring (K). A damping coefficient (D), which arises from various factors such as squeeze film damping, is defined as a dashpot. The mechanical sensing element model in Figure 2.14 ideally can be described in mathematical form based on Newton's second law [45]:

$$M \frac{d^2 y}{dt^2} = D \frac{dx}{dt} + Kx \tag{2.6}$$

where M is the mass of the mechanical sensing element, D is the damping coefficient, K is the spring constant of the suspension system, y is the displacement of the proof mass, and x is the relative displacement of proof mass with respect to the reference frame, which is equal to the subtraction of frame displacement (z) and the proof mass displacement (y), $x = z - y$. Thus, Equation 2.6 can be converted into the following form:

$$Ma_{in} = M\frac{d^2z}{dt^2} = M\frac{d^2x}{dt^2} + D\frac{dx}{dt} + Kx \tag{2.7}$$

where a_{in} is the exterior input acceleration.

To analyse the dynamic performance of the accelerometer, Equation 2.7 can be represented in the form of a second order transfer function by applying Laplace transform [45].

$$H(s) = \frac{x(s)}{a_{in}(s)} = \frac{1}{s^2 + \frac{D}{M}s + \frac{K}{M}} = \frac{1}{s^2 + \frac{\omega_0}{Q}s + \omega_0^2} \tag{2.8}$$

where s is the Laplace operator, ω_0 is the natural resonant frequency of the mechanical sensing element:

$$\omega_0 = \sqrt{\frac{K}{M}} \tag{2.9}$$

and Q is the quality factor, which is given by:

$$Q = \frac{\sqrt{KM}}{D} = \frac{M\omega_0}{D} \tag{2.10}$$

2.3.4 Design parameters of the mechanical sensing element

1. Static sensitivity

The static sensitivity of the mechanical sensing element illustrates how the system is sensitive to the excitation acceleration. In the surface micromachined capacitive mechanical sensing element, static sensitivity (S) can be defined as the differential change in capacitance over the input acceleration (a_{in}), which is given by [45]:

$$S = \frac{C_{s1} - C_{s2}}{a_{in}} = \frac{N_s \varepsilon_0 A}{a_{in}} \left(\frac{1}{G-x} - \frac{1}{G+x} \right) \quad (F/g) \tag{2.11}$$

where x is the relative displacement of the proof mass when the system is excited by input acceleration (a_{in}). In the steady state condition, where the input acceleration (a_{in}) is a constant, the internal stress on the suspension spring is a constant that is equal to the force on the proof mass [45]. Thus, the displacement of the proof mass (x) is given by:

$$x = \frac{M a_{in}}{K} = \frac{a_{in}}{\omega_0^2} \tag{2.12}$$

2. Resonant frequency

The physical design parameters of the mechanical sensing element (spring constant K, the damping coefficient D and the mass of the proof mass M) must be carefully designed depending on the requirements of the accelerometer. As shown in Equation 2.9, the natural resonant frequency, which determines the upper boundary of the bandwidth of the open-loop accelerometer [117], can be increased by reducing the mass of the proof mass and increasing the spring constant. However, an important design trade-off should be taken into consideration as the static sensitivity is reduced while the resonant frequency is increasing (Equation 2.11 and 2.12). This design trade-off can be overcome by applying a force feedback control loop to the mechanical sensing element [117].

3. Quality factor

The dynamic response of the mechanical sensing element can be categorised into three types according to the quality factor (Q): under-damped($Q > 0.5$), critical-damped($Q = 0.5$) and over-damped($Q < 0.5$). Figure 2.15 shows the time domain analysis of a mechanical sensing elements with different damping coefficients (D), i.e. different quality factors. As shown in the figure, under-damped sensing element is fast to respond but the step response exhibits significant overshot and ringing. The output of the over-damped sensing element achieves the steady value very slowly but without any overshot. The critical-damped sensing element offers the fastest response without overshot [45].

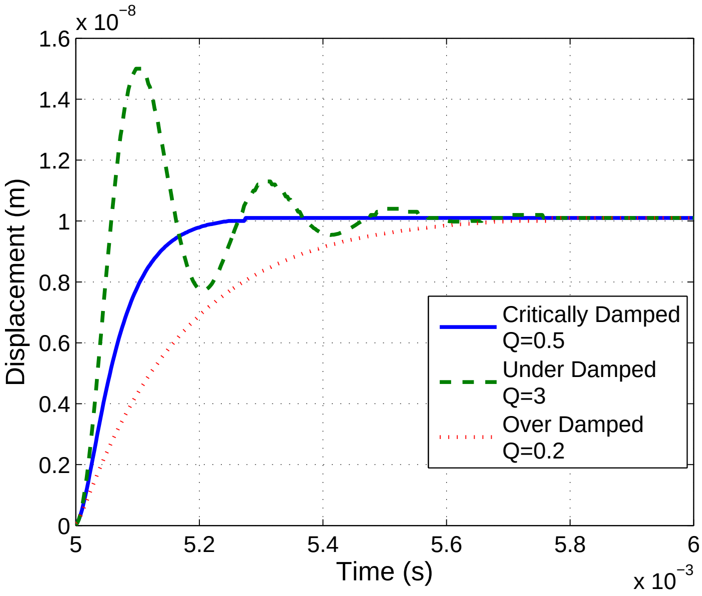


FIGURE 2.15: Step response of a mechanical sensing element with different quality factor

4. Mechanical Noise

Because of the small size of the mechanical sensing element, the measurement signal power has a low value which can be degraded easily by Brownian noise. The noise equivalent acceleration (a_N) is given by [119]:

$$a_N = \sqrt{\frac{4K_B T\omega_0}{MQ}} \tag{2.13}$$

where K_B is the Boltzmann constant and T is the temperature in Kelvin. ω_0 is the resonant frequency of the sensing element, M is the mass, and Q is the quantity factor. As shown in the equation, this noise can be reduced by increasing the mass and the quality factor of the mechanical sensing element. Thus, noise can be reduced by mechanical structure optimisation and packaging [45].

2.3.5 Sigma-Delta modulation technique

As illustrated in Figure 2.10, the topology of a closed-loop digital MEMS accelerometer is inspired by Sigma-Delta modulators. Thus, this section provides a brief review of Sigma-Delta modulators.

Analogue-to-digital converters(ADCs) can be divided into two categories: Nyquist-Rate converters and oversampling converters. Compared with the Nyquist-rate ADCs, oversampling ADCs, such as Sigma-Delta modulator, can achieve higher resolution and release critical requirements on the IC fabrication process by sacrificing the signal bandwidth [120]. Oversampling and noise shaping are the two main techniques employed in the Sigma-Delta modulators to achieve their advantages. The oversampling technique makes the noise spread over a wider frequency range; while the noise shape dynamically decreases the noise in the signal band; therefore, higher resolution is available [120, 121, 122, 123].

2.3.5.1 Oversampling and noise shaping

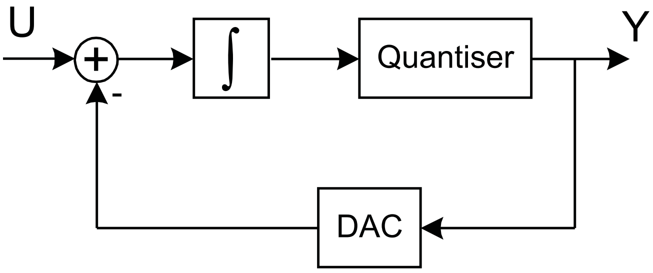


Figure 2.16: First order Sigma-Delta modulator [122]

To illustrate the operation of the Sigma-Delta modulators, the structure of a simple first-order Sigma-Delta modulator, which is a feedback loop consisting of one quantiser, one digital to analogue converter (DAC), and a loop filter (an integrator in the first order structure), is shown in Figure 2.16.

The quantiser is the main component of the modulator that introduces an error (regarded as quantisation noise) during the quantisation process and affects the performance of the system. To reduce the non-linear distortion from the quantiser, a single-bit quantiser is usually preferred to multi-bit ones. Thus, only single-bit quantiser is considered in this section. Quantisation noise can be treated as white noise whose root-mean-square(RMS) value e_{RMS} can be given by the following well-known equation [122]:

$$e_{RMS} = \frac{\Delta}{\sqrt{12}} \tag{2.14}$$

where Δ is the quantization step, i.e. the interval between two successive quantization levels [122].

As one of the key techniques in Sigma-Delta modulators, oversampling can reduce noise level while keeping the input signal's power in the signal band. This because the quantization noise is approximated as a white noise whose power is always spread over half of the sampling bandwidth uniformly, and the power of the noise signal is a constant. If the Sigma-Delta modulator is sampled at frequency f_s, we can recall Equation 2.14 to derive the power spectral density (PSD) of the quantisation noise ($S_e(f)$):

$$S_e(f) = (\frac{2e_{RMS}^2}{f_s}) = \frac{\Delta^2}{\sqrt{6f_s}} \tag{2.15}$$

As shown in the equation above, the increment of the sampling frequency spread the noise to a wider frequency range and reduces the noise power density. The power of the noise in the signal band (P_e) can be calculated by integrating $S_e(f)$ over bandwidth of interest (f_0) [122]:

$$P_e = (\frac{e_{RMS}^2}{3OSR^3}) \tag{2.16}$$

where OSR is the oversampling ratio and is calculated as $OSR = f_s/2f_0$. f_0 is the maximum signal frequency, i.e. the signal bandwidth. $2f_0$ is regarded as the Nyquist frequency. f_s is the oversampling frequency. Thus, OSR defines how much faster the signal is sampled in a Sigma-Delta modulator than in a Nyquist-rate converter.

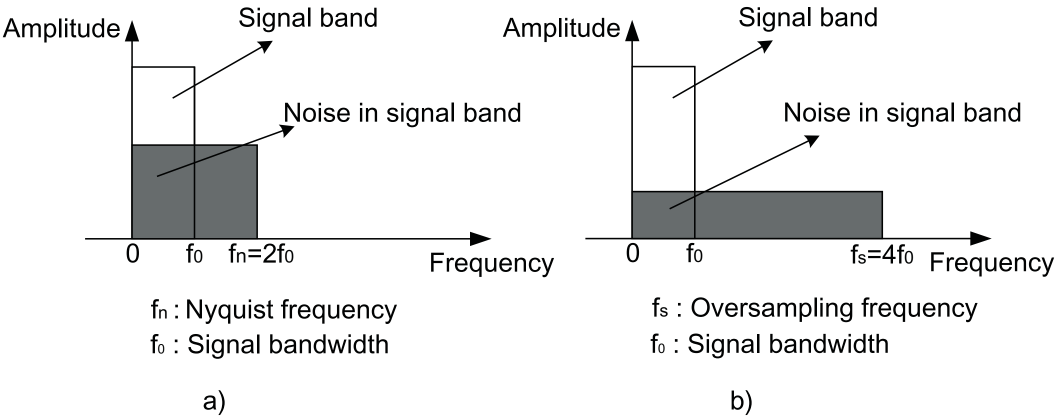


FIGURE 2.17: Noise spectrum of conventional Nyquist converter(a) and oversampling converter(b) [122]

Compared with the typical Nyquist converters, Sigma-Delta modulators use a sampling frequency that is much higher than the Nyquist frequency. As shown in Figure 2.17, the quantisation noise is spread over a wider spectrum. Therefore, this results in a greater reduction of the noise in the signal bandwidth.

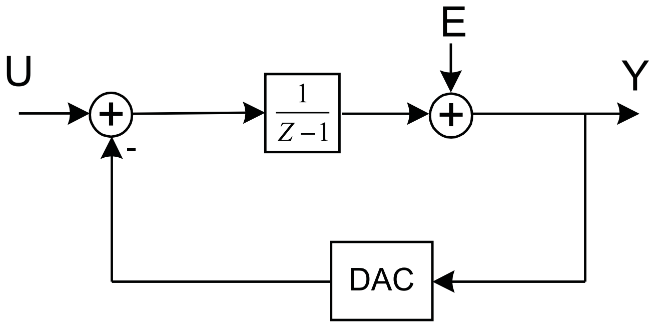


FIGURE 2.18: Linearised z-domain model of the first order Sigma-Delta modulator [122]

To illustrate the noise shaping technique of the Sigma-Delta modulators, a linearised z-domain model of the first-order Sigma-Delta modulator is presented in Figure 2.18. As shown in the figure, the quantiser can be treated as an adder with an additive quantisation noise source E which is independent of the circuit input U. According to the linearised model, the signal transfer function (STF) and the noise transfer function (NTF) are given by [122]:

$$STF(z) = \frac{Y(z)}{U(z)} = \frac{1/(z-1)}{1+1/(z-1)} = z^{-1} \tag{2.17}$$

$$NTF(z) = \frac{Y(z)}{E(z)} = \frac{1}{1+1/(z-1)} = 1 - z^{-1} \tag{2.18}$$

The STF of the Sigma-Delta modulator is just a delay. This means the input signal in the bandwidth of interest is well preserved. On the other hand, the NTF of the Sigma-Delta modulator is a high-pass filter function. If z is replaced by $e^{j2\pi f/f_s}$, PSD of the output noise is given by [122]:

$$S_q(f) = (2sin(\pi f/f_s))^2 S_e(f) \tag{2.19}$$

As shown in the equation above, the quantisation noise in the signal bandwidth is strongly attenuated and pushed into the higher frequency band. The in-band noise power can be obtained by integrating $S_q(f)$ between 0 to f_0. Assume $OSR \gg 1$, the in-band noise power (P_e) is given by [122]:

$$P_e = \frac{\pi^2 e^2_{RMS}}{3(OSR)^3} \tag{2.20}$$

It is clear that adding more integrators to form a high-order loop filter in the feed-forward signal path of the Sigma-Delta modulator will result in better noise shaping. For example, a second-order Sigma-Delta modulator can be implemented by adding another integrator and feedback path to the first-order Sigma-Delta modulator as shown in Figure 2.19. The linearised model of the this modulator is shown in Figure 2.20.

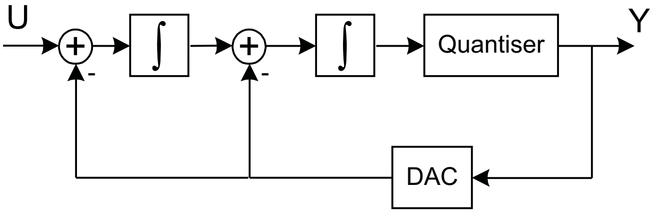


FIGURE 2.19: Second-order Sigma-Delta modulator [122]

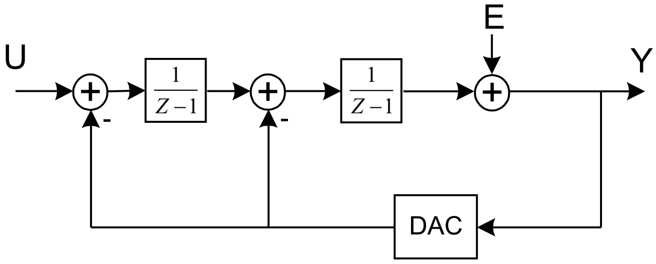


FIGURE 2.20: Linearised z-domain model of the second-order Sigma-Delta modulator [122]

From linearised model, the STF and NTF of the second-order Sigma-Delta modulator are given by:

$$STF(z) = \frac{Y(z)}{U(z)} = \frac{1/(z-1)}{1+1/(z-1)} = z^{-2} \tag{2.21}$$

$$NTF(z) = \frac{Y(z)}{E(z)} = \frac{1}{1+1/(z-1)} = (1-z^{-1})^2 \tag{2.22}$$

As shown in the transfer functions, the input signal is delayed more on the propagation to the output which means the input signal is still well preserved; however, noise is differentiated more times and sharper noise shaping function is achieved. The in-band quantisation noise power (P_{e2}) for the second-order Sigma-Delta modulator is given by [122]:

$$P_{e2} = \frac{\pi^4 e_{RMS}^2}{5(OSR)^5} \tag{2.23}$$

In principle, higher-order NTFs can be derived by adding more integrators and feedback paths to the loop [122]. Figure 2.21 shows the noise shaping of the Sigma-Delta modulators with different order. As clearly shown in this figure, the shape of the noise becomes sharper and the quantization noise is pushed to a much higher frequency band when the order of modulator is increased.

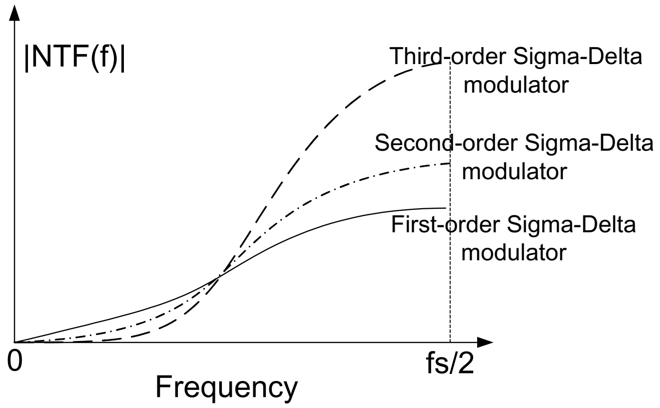


FIGURE 2.21: Noise shaping with different order of Sigma-Delta modulator [121]

2.3.5.2 High-order Sigma-Delta modulators

As discussed in the last section, one obvious way to improve the performance of the Sigma-Delta modulator is to increase the loop order. There are two different architectures for implementing high-order Sigma-Delta modulators: single-stage modulators and multi-stage modulators.

Many topologies are available for implementing a single-stage higher-order Sigma-Delta modulator. The interpolative architecture, invented by Chao in 1990, is one of the most commonly-used structures [124]. This architecture contains a series of integrators with distributed feedback and feed-forward signal paths as depicted in Figure 2.22 [125]. The major drawback of the single-stage high-order Sigma-Delta modulator is that increasing the loop order to more than third order results in instability of the system [122]. This is because of the nonlinear limitations of the quantiser [122]. To establish loop stability, extensive simulation is usually required to carefully determine the modulator coefficients [123].

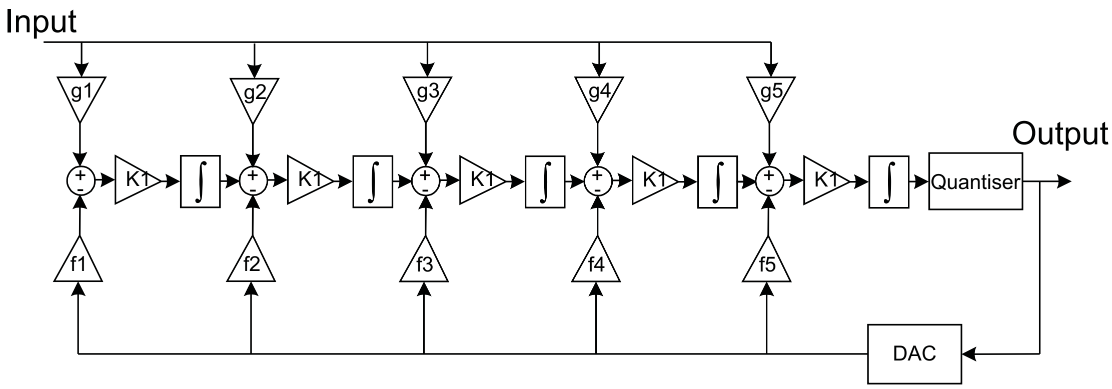


FIGURE 2.22: A single-stage fifth-order Sigma-Delta modulator [125]

To overcome the stability problem, several lower order (first or second order) single-stage Sigma-Delta modulators can be cascaded to form a multi-stage higher-order Sigma-Delta modulator (MASH structure) [122, 126]. A general multi-stage MASH structure is shown in Figure 2.23 [122]. The z-transform outputs of the two stages are:

$$Y_1(z) = STF_1U(z) + NTF_1E_1(z) \tag{2.24}$$

$$Y_2(z) = STF_2E_1(z) + NTF_2E_2(z) \tag{2.25}$$

The basic concept of this architecture is to cancel the first stage quantisation noise E_1 at the output using digital filters D_1 and D_2. According to the above two equations, the relationship of the digital filters are given by:

$$NTF_2D_1 = STF_2D_2 \tag{2.26}$$

Usually, the digital filters are designed to make: $D_1 = STF_2$ and $D_2 = NTF_1$. Thus, the overall output is given by:

$$Y(z) = STF_1STF_2U(z) + NTF_1NTF_2E_2(z) \tag{2.27}$$

As shown in the equation, only the quantisation noise of the last stage E_2, which is shaped by overall order of the modulator, appears in the modulator output. The

Sigma-Delta modulators in MASH structure display excellent stability properties as compared with single-stage modulators; however, the MASH modulators require precise filter matching among digital filters and the analogue components of the modulators. A mismatch results in a substantial degradation of the overall performance of the modulator.

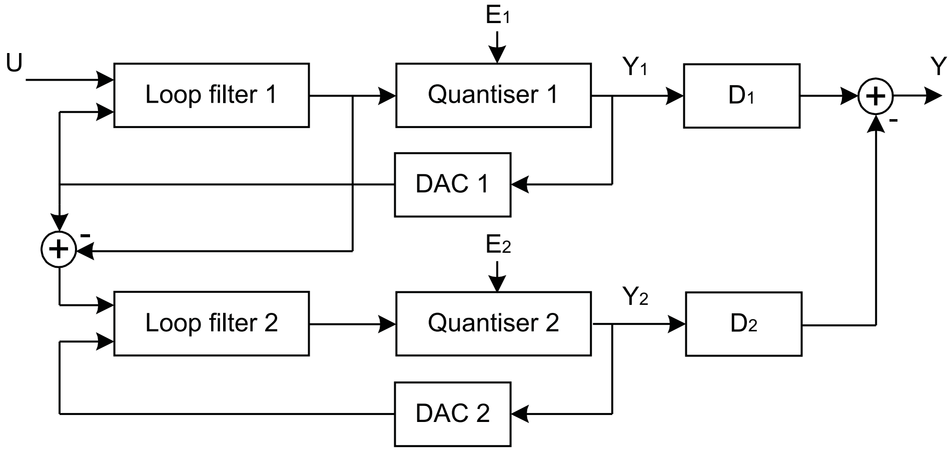


FIGURE 2.23: A multi-stage Sigma-Delta modulator [122]

2.3.6 Overview of high-order electromechanical Sigma-Delta modulators

In the conventional second-order electromechanical Sigma-Delta modulator as shown in Figure 2.10, the dynamics of the mechanical sensing element limit the noise shaping properties. Compared with typical electronic second order Sigma-Delta modulators, the gain of mechanical integrators is quite low resulting in a lower signal-to-noise ratio (SNR) in second-order electromechanical Sigma-Delta modulators. This is considered insufficient in high-performance applications. For example, for most automotive and other low-cost applications which usually require the resolution of the accelerometer about $10mG$, second-order electromechanical Sigma-Delta modulator still can achieve this performance requirement; however, it is difficult for second-order modulator to obtain a resolution less than $5\mu G$ for inertial navigation applications [23].

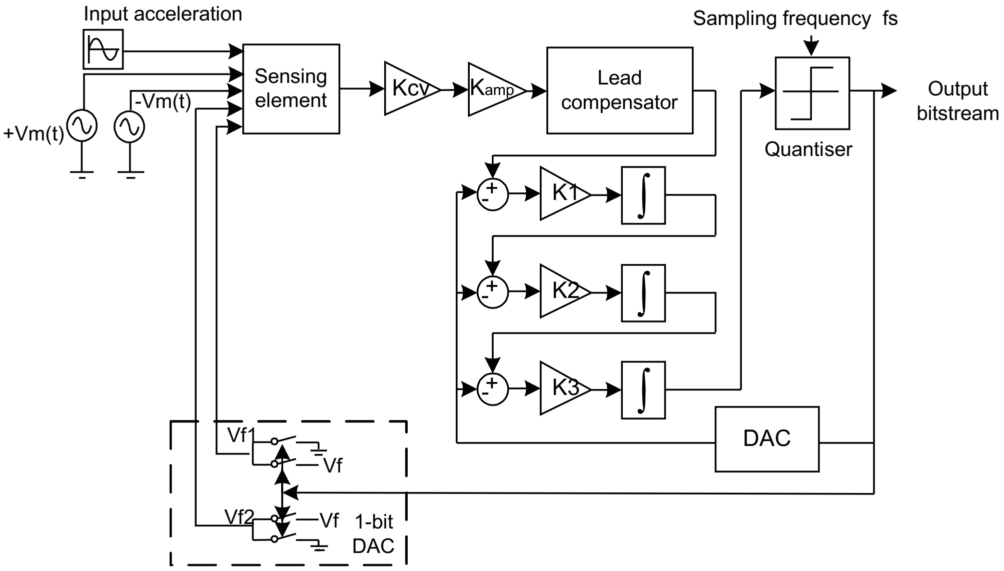


FIGURE 2.24: A single-stage fifth order electromechanical Sigma-Delta modulator [118]

In order to improve the performance of MEMS accelerometer, higher order electromechanical Sigma-Delta modulator designs are increasingly becoming attractive [118, 127, 105, 128]. Dong *et al.* [118] used a mechanical sensing element and additional cascaded integrators with distributed feedback, based on a third order distributed electronic loop filter, to form a fifth order electromechanical Sigma-Delta modulator as shown in Figure 2.24. The experiment demonstrated great improvement of the SNR compared with that of a second order structure. Petkov and Boser [105] fabricated a fourth order Sigma-Delta interface for micromachined inertial sensors based on a chain of integrators with feed-forward summation. More available structures, such as a sixth order multiple-feedback (MF) electromechanical Sigma-Delta topology, are demonstrated by Dong *et al.* [128]. These topologies are all based on the idea of inserting an additional electronic loop filter between the interface front-end and the quantiser. The additional filter, which provides high gain only in the signal band and rejects the out-of-band electronic noise, increases the order of the Sigma-Delta modulator [105] and dramatically decreases the noise floor in signal band. Kraft *et al.* [129] presented a novel multistage noise shaping (MASH) structure in which the electromechanical Sigma-Delta modulator is cascaded with a purely electronic Sigma-Delta modulator. Such an architecture typically has large fabrication tolerances because accurate cancelation of the quantisation noise in this structure relies on the values of mechanical sensor parameters [105].

2.4 MEMS synthesis methodologies

Although MEMS systems are forming the basis for a rapidly growing industry and fields of research, many MEMS designers still rely on back-of-the-envelope calculations. This is due to a lack of efficient computer-aided design (CAD) tools that can assist with the initial stages of design exploration [130]. A significant amount of specialist human resources and time is consumed in the iterative trial-and-error design process [14]. Therefore, there is an increasing need for automated synthesis techniques that would shorten the development cycle and facilitate the generation of optimal configurations for a given set of performance and constraint guidelines. This section discusses some recent MEMS synthesis methodologies, several of which are listed in Table 2.2.

	MEMS Synthesis Methodologies	***Year***	***Features***
1	Equation-based layout synthesis of MEMS	1999	Less accurate than simulation-based approach as lumped parameter model equations are used. It is highly knowledge intensive.
2	Simulation-based layout synthesis of MEMS	2002	Easy to use. Simulation with FEA accuracy by NODAS. Long computation time is the major problem.
3	Hierarchical evolutionary synthesis of MEMS (BG/GP approach)	2004	Combining genetic programming and bond graphs to synthesise behavioural models. Long computation time.
4	Hierarchical MEMS synthesis and optimisation	2005	Two levels of optimisation: global genetic algorithms and local gradient-based refinement
5	Case-based reasoning (MEMS-CBR)	2006	Reuse past successful design cases to generate better solutions. Case library is difficult to develop.

Table 2.2: MEMS Synthesis Methodologies

2.4.1 Equation-based layout synthesis of MEMS

A rapid layout synthesis of a lateral surface micromachined accelerometer from high-level functional specifications and design constraints is described by Tamal *et al.* [11, 131, 132, 12]. The goal of synthesis is to select the optimal design that minimises an objective function such as the device area. The design flow is shown in the Figure 2.25.

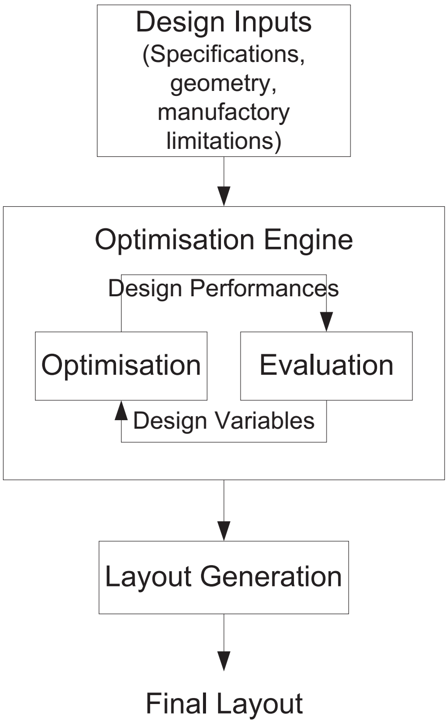


FIGURE 2.25: Equation-based MEMS synthesis flow [11]

This is an equation-based approach which is highly knowledge intensive. It requires lumped parameter model equations to characterise the behaviour of the device being synthesised. The design problem is then formulated into a nonlinear constrained optimisation problem with these equations and constraints on the device's behaviour. In this approach, the optimisation is carried out by using a gridded numerical optimisation algorithm in which the search for the optimal design is guided by an object function in an evaluation module. The optimisation-based design process iterates on the values of the design variables till the evaluation module indicates that the design specifications are met. Finally, for visualization of the synthesized results, a parameterised layout generator similar to the Consolidated Micromechanical Element Library (CaMEL) software was used. It provides a popular Caltech Intermediate Format(CIF) layout format output when given accelerometer layout parameters.

2.4.2 Simulation-based optimal layout synthesis methodology

A simulation-based optimal layout synthesis methodology for CMOS MEMS accelerometer is presented by Gupta *et al.* [13]. The synthesis flow is shown in Figure 2.26. In this approach, Parallel Recombinative simulated annealing (PRSA), which uses a number of parallel annealing tasks instead of a single annealing task to search for the global optimum, is used as the optimisation algorithm. The simulation of the CMOS accelerometer is implemented in NODAS, which can perform detailed simulation with FEA accuracy. The data processing capabilities in Cadence's OCEAN environment have been used for encapsulation of the NODAS model as OCEAN supports processing of all types of simulation data from NODAS through the use of evaluation scripts. Finally, a parameterised layout generator is used to generate the layout of the accelerometer.

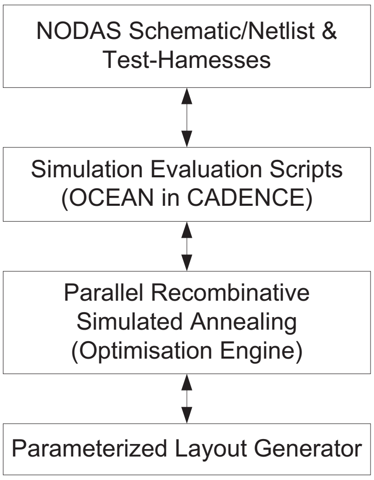


FIGURE 2.26: Simulation-based MEMS synthesis flow [13]

Compared with the equation-based synthesis approach, the simulation-based approach is much easier to use because designers do not need to re-derive behavioural equations when there are changes in the device's topology. Furthermore, this simulation-based evaluation is more accurate as it uses FEA-based simulation tool. However, long computation time is generally the major problem of such a simulation-based approach. In this approach, the annealing algorithm needs a few thousand evaluations of candidate solutions before converging on the optimal solution [13]. However, computation time is a minimum of tens of seconds for each evaluation.

2.4.3 Hierarchical evolutionary synthesis of MEMS

Fan *et al.* presented a hierarchical evolutionary approach to MEMS synthesis [14]. The synthesis flow is shown in the Figure 2.27. In this approach, the design of MEMS is divided into two levels: system-level behavioral macromodel design and physical layout synthesis.

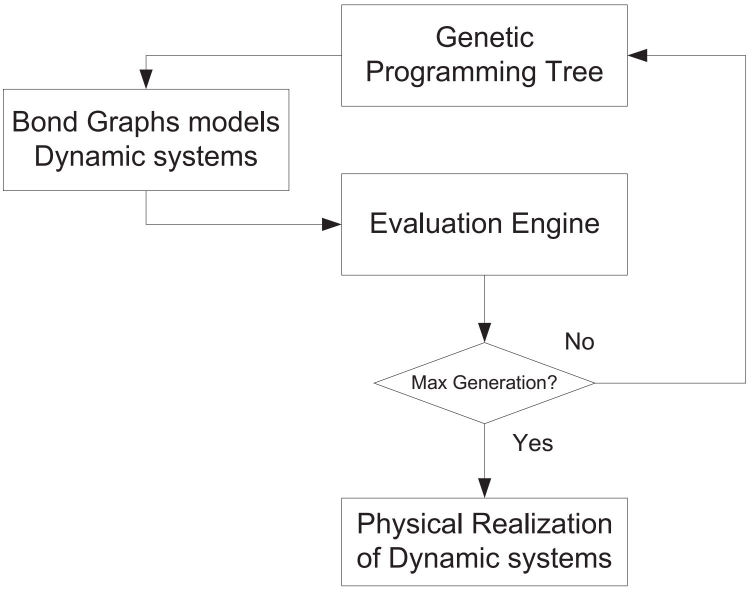


FIGURE 2.27: Hierarchical evolutionary MEMS synthesis flow [14]

At the system level, a BG/GP approach, combining bond graphs and genetic programming (GP), is used to generate and search for design candidates of system-level macromodels that meet the predefined behavioral specifications. A bond graph [133] is a graphical description of a physical dynamic system. It is an energy-based graphical technique for modelling and analysing dynamic systems, especially hybrid multi-domain systems [134]. The BG/GP approach implemented a bond graph class in C++, and then changed the bond graph topologically using a genetic programming, yielding new design alternatives [14]. However, it took about 20 hours for the GP program to obtain satisfactory results.

At the physical layout synthesis level, the selection of geometric parameters for MEMS devices is formulated as a constrained optimisation problem and addressed using a multi-objective constrained genetic algorithm (GA) approach.

2.4.4 Hierarchical MEMS synthesis and optimisation

A hierarchical synthesis and optimisation technique has been developed for MEMS design automation by Zhang *et al.* [130, 135, 136, 137]. The MEMS synthesis flow is shown in Figure 2.28.

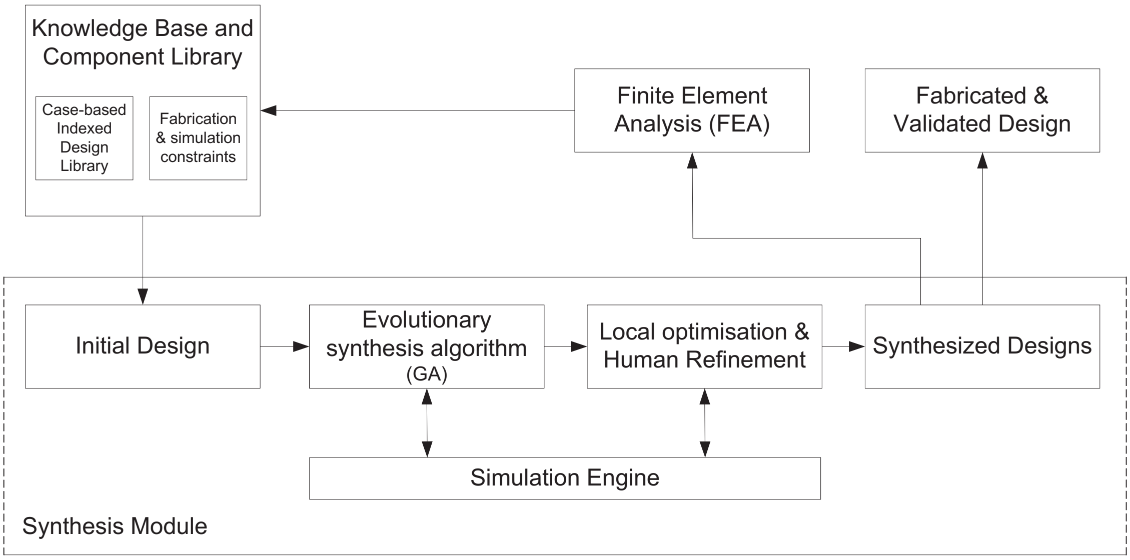


FIGURE 2.28: Hierarchical MEMS synthesis and optimization flow [130]

When a designer specifies the design objectives, constraints and stopping criteria, an initial valid design or a set of designs is loaded into the design synthesis module from the MEMS design component library. The design synthesis module uses the multi-objective genetic algorithm (MOGA) optimisation algorithm to mutate the initial design, creating the population for the next generation. All of the designs in a generation are evaluated by the SUGAR MEMS simulator to determine their performance attributes. The MOGA optimisation process stops when the stopping criteria are met. A conventional gradient-descent optimisation algorithm has been implemented to further refine the best designs resulting from MOGA synthesis. Finally, the synthesised designs are evaluated using Finite Element Analysis (FEA) tools.

2.4.5 Case-based reasoning for the design of MEMS systems (MEMS-CBR)

Cobb *et al.* [15] introduced a case-based reasoning (CBR) technique to design MEMS resonant structures. Case-based reasoning tools utilise human knowledge from past successful design cases to guide human designers and computer-aided design (CAD) programs towards better design concepts to deal with the complexities of a new design problem. Figure 2.29 illustrates the design flow for MEMS-CBR.

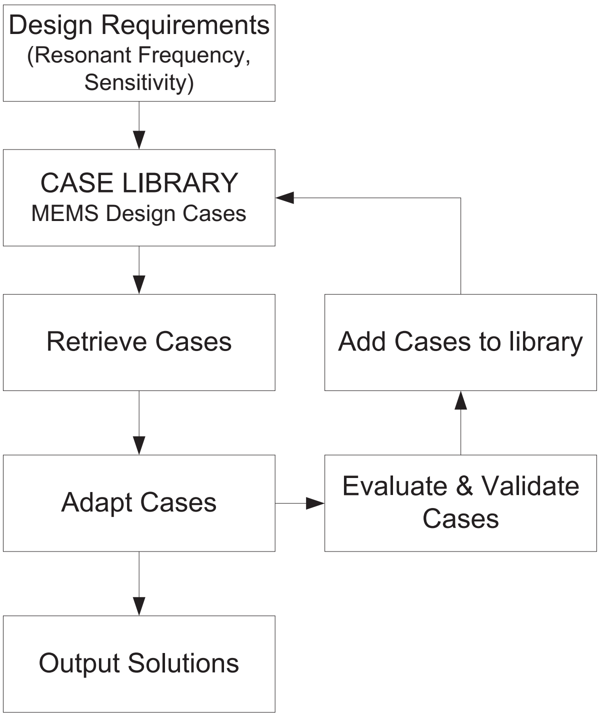


FIGURE 2.29: Case-based reasoning design flow for MEMS systems (MEMS-CBR) [15]

The most relevant cases are retrieved from the case library using efficient and accurate retrieval algorithms according to the input specifications. The case library contains MEMS components, building blocks, and entire devices. Once cases are retrieved, they are adapted to fit the current design problem, using parametric optimisation or more exploratory techniques, such as genetic algorithms. Cases are initially validated and evaluated with a MEMS simulation tool. SUGAR is used as the simulation tool in this research. If new designs have been synthesised from the system, they are validated further with fabrication and testing before being added to the case library for future use. However, before the development of a MEMS-CBR system, acquisition of MEMS design cases is difficult.

2.5 Concluding remarks

In this chapter, a review of the related literature has been presented. Firstly, a broad range of applications of MEMS systems were reviewed. In this research, a surface micromachined capacitive MEMS accelerometer with Sigma-Delta control scheme, which is one of the most sophisticated types of MEMS inertia sensors, was selected as a case study. Through the surveyed literature on the simulation and modelling tools for MEMS systems, AMS HDLs, such as VHDL-AMS and SystemC-A, were chosen as the modelling tools for the case study since they are very powerful and flexible mixed physical domain modelling tools which are able to integrate mechanical MEMS devices and associated electronics into a single model.

The AMS HDLs still face a challenge when modelling MEMS systems with distributed behaviour because current AMS HDLs can only describe an analogue system by ODEs. This limits accurate modelling of MEMS devices with distributed physical behaviours which play vital roles in the system performance. For example, it is well known that performance of a MEMS capacitive acceleroemter with a Sigma-Delta control is affected by the sense finger resonance in the mechanical sensing element [20]. However, the conventional approach normally applied in simulations of such systems, where a lumped mass-damper-spring system is used to model the mechanical sensing element, cannot capture the effect of the sense finger dynamics. In Chapter 3, we present an approach to modelling distributed sense finger dynamics of the mechanical sensing element with

VHDL-AMS and SystemC-A. This enables us to accurately predict the performance of the system.

Several modelling and performance optimisations techniques for MEMS systems have been reviewed. However, these approaches are constrained to the layout synthesis of the MEMS mechanical element. In high-performance MEMS systems, an electronic control system is usually applied. How to deal with the automated optimal MEMS sensing element and electronic control loop co-design is the major target of this research. In Chapter 4, we present a novel, holistic methodology for automated optimal synthesis of MEMS systems embedded in electronic control circuitry from user-defined high-level performance specifications and design constraints.

In Chapter 5, we propose a new syntax extension for SystemC-A to support general PDEs modelling. This syntax extension further enhances the modelling efficiency and capability of SystemC-A for mixed-technology systems with crucial distributed behaviour.

Chapter 3

Modelling of MEMS accelerometers with Sigma-Delta control in VHDL-AMS and SystemC-A

This chapter presents an approach to modelling distributed physical effects of MEMS devices with VHDL-AMS and SystemC-A to enable accurate performance prediction of critical mechanical components. As a case study, a surface micromachined capacitive MEMS accelerometer with Sigma-Delta control scheme is used to demonstrate the methodology. In such an accelerometer, it is well-known that the sense finger resonance in the mechanical sensing element affects the performance of the electromechanical Sigma-Delta feedback loop; however, correct behaviour cannot be predicted by the conventional lumped mechanical sensing element model, where a second-order ordinary differential equation (ODE) is commonly used. In this chapter, a distributed approach, where the sense fingers are modelled as cantilever beams whose motion can be described by Partial Differential Equations (PDEs), has been applied to capture the effects of the sense finger dynamics in the MEMS accelerometer with Sigma-Delta control.

This chapter is organised as follows. In order to compare with our proposed distributed approach, section 3.1 presents the conventional methodology to design and model MEMS accelerometer with Sigma-Delta control in VHDL-AMS and SystemC-A. This section also provides the theories to calculate the lumped parameters of the mass-damper-spring system (i.e. mass, spring constant and damping coefficient) according to the layout of mechanical sensing element. Section 3.2 proposes an improved distributed mechanical model and provides detailed analysis of how sense finger dynamics affect the operation of the accelerometer. Section 3.3 provides a comparison between VHDL-AMS and SystemC-A according to the simulation results of the MEMS accelerometer. Finally, Section 3.4 draws conclusions from this work.

3.1 Conventional model of a MEMS capacitive accelerometer with Sigma-Delta control

As mentioned in section 2.3, high performance MEMS sensors usually take advantage of a Sigma-Delta force feedback control strategy to improve linearity, dynamic range, and bandwidth, and provide direct digital output in the form of pulse density modulated bitstream, which can interface with a digital signal processor. This approach has been applied to MEMS accelerometers and gyroscopes [105,106].

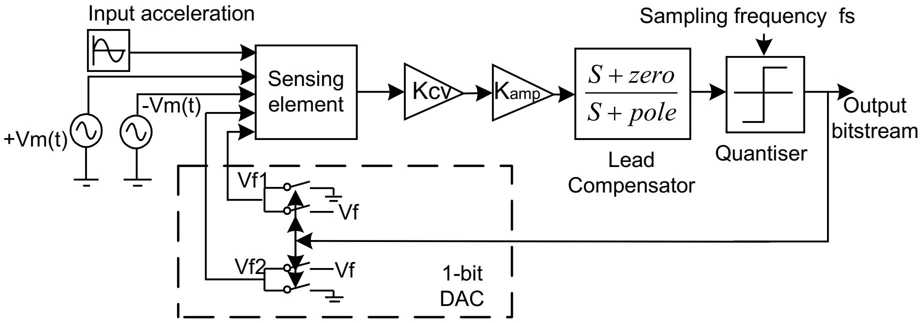


FIGURE 3.1: Second-order electromechanical Sigma-Delta modulator[106]

The diagram of a second-order electromechanical Sigma-Delta modulator is shown in Figure 3.1. The mechanical sensing element is followed by the signal pick-off circuit which is represented by a gain block K_{cv}. K_{amp} is the gain of the voltage booster amplifier following the pick-off stage. $V_{f1}(t)$ and $V_{f2}(t)$ are the feedback voltages obtained from the DAC to generate electrostatic feedback force in the mechanical sensing element, and $V_m(t)$ is a high frequency modulation voltage. A lead compensator is required to stabilize the system. A one-bit quantiser is used to oversample and convert the analogue voltage to a pulse density modulated digital signal. f_s is the oversampling frequency. If the signal bandwidth of the system is f_0, the oversampling ratio (OSR) of the system is given by:

$$OSR = \frac{f_s}{2f_0} \tag{3.1}$$

As shown in the Figure 3.1, the mechanical sensing element of the MEMS Sigma-Delta modulator is used as a loop filter. This is because the sensing element is conventionally approximated by a second-order mass-damper-spring system which performs a similar function to that of two cascaded integrators in a typical second-order electronic Sigma-Delta modulator. Thus, in such a configuration, dynamics of the mechanical sensing element limit the performance of the system. The mechanical sensing element in the closed-loop system is usually modelled by following equation:

$$Ma_{in} + F_{feedback} = M\frac{d^2x}{dt^2} + D\frac{dx}{dt} + Kx \tag{3.2}$$

where x is the relative displacement of the proof mass with respect to the substrate, a_{in} is the input acceleration, and $F_{feedback}$ is the feedback force. M, K, D are lumped parameters which represent proof mass (kg), spring constant (N/M) and damping coefficient (N·s/m) respectively.

The schematic of the mechanical sensing element used in this research is shown in Figure 3.2. This topology is similar to that of ADXL series accelerometers from Analog Devices where the proof mass is suspended by four cantilever beam springs and equipped

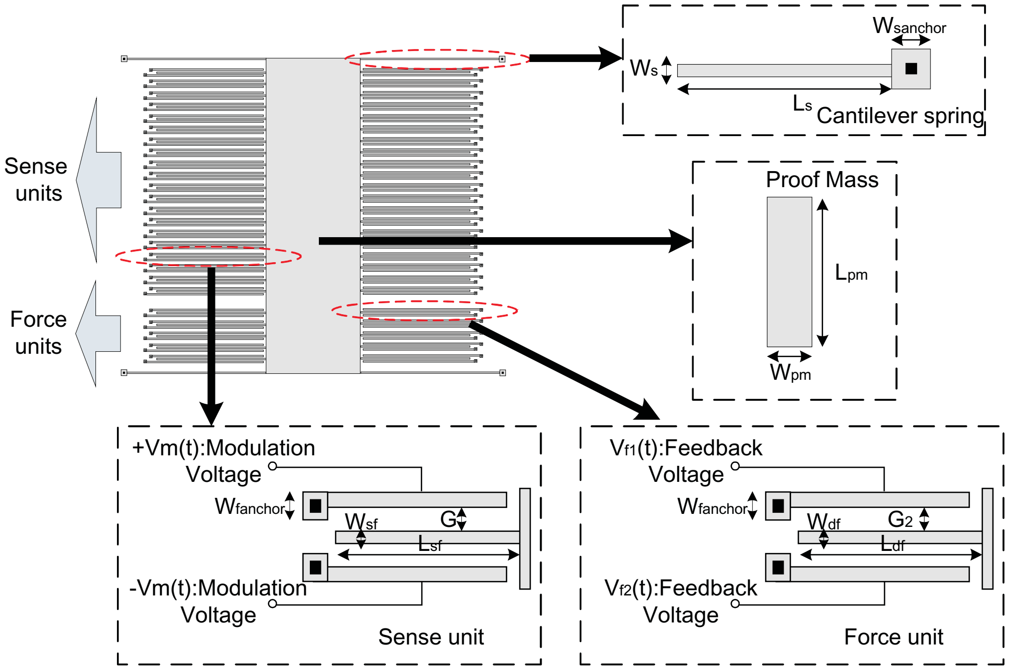


FIGURE 3.2: Layout of the mechanical sensing element

with movable comb fingers which are placed between the fixed fingers as the common centre electrode to form capacitance bridges. Such a constructed mechanical sensing element can detect a differential change in capacitance caused by the displacement of movable fingers, and convert it to voltage by associated interface circuits. Among these capacitance bridges, most capacitance groups act as sensing capacitance (sense units) and a few other capacitance groups (force units) are used to generate electrostatic feedback force. In the closed-loop operation, feedback voltages ($V_{f1}(t)$ and $V_{f2}(t)$) are applied to the fixed fingers in each force unit such that the resulting electrostatic force pulls the moving proof mass back to its original position. Assuming the displacement of the force finger is much smaller than the initial gap (G_2) between the force finger and the fixed fingers in a force unit, the expression for the feedback electrostatic force is given by:

$$F_{feedback} = \frac{N_f \varepsilon_0 L_{ff} T}{2G_2^2}(V_{f1}^2 - V_{f2}^2) \tag{3.3}$$

where ε_0 is dielectric constant, N_f is the number of the force fingers, L_{ff} and T are the length and the thickness of the force fingers respectively.

Symbol	***Design Variables of sensing element***	***Value***
W_{pm}	Width of proof mass	$120\mu m$
L_{pm}	Length of proof mass	$450\mu m$
T	Thickness of springs, comb fingers, and proof mass	$2.0\mu m$
L_s	Length of cantilever spring	$176\mu m$
W_s	Width of cantilever spring	$2.0\mu m$
$W_{sanchor}$	Width of cantilever spring anchor	$10.0\mu m$
L_{sf}	Length of sensing fingers	$150\mu m$
W_{sf}	Width of sensing fingers	$2.0\mu m$
L_{ff}	Length of force fingers	$150\mu m$
W_{ff}	Width of force fingers	$2.0\mu m$
G	Initial gap between sense finger and fixed fingers in a sense unit	$1.3\mu m$
G_2	Initial gap between force finger and fixed fingers in a force unit	$1.3\mu m$
N_s	Number of sensing comb fingers	54
N_f	Number of driving comb fingers	4
$W_{fanchor}$	Width of finger anchor	$5.0\mu m$

Table 3.1: Dimension of the mechanical sensing element

Now, we will provide theories to calculate the lumped parameters, i.e. mass (M), damping coefficient (D) and spring constant (K) in Equation 3.2, according to layout of the mechanical sensing element. In this design, all MEMS components in the sensing element, i.e. proof mass, springs and comb fingers, are made in a mechanical polysilicon layer with $2\mu m$ thickness (T). The proof mass consists of 54 sense units and 4 force units. For simplicity, both sense fingers and force fingers have the same length ($150\mu m$). The initial gap between the movable fingers and fixed fingers is $1.3\mu m$. The specific structural parameters of the mechanical sensing element are listed in Table 3.1.

Proof mass (M):

The proof mass (M) can be calculated by assuming it is a single polysilicon with density $\rho = 2330kg/m^3$:

$$\begin{aligned} M &= \rho(V_{mass} + V_{fingers}) \\ &= \rho(W_{pm}L_{pm} + (N_s + N_f)L_{sf}W_{sf})T \\ &= 2330 \times (120 \times 10^{-6} \times 450 \times 10^{-6} + (54+4) \times 150 \times 10^{-6} \times 2 \times 10^{-6}) \times 2 \times 10^{-6} \\ &= 3.32 \times 10^{-10} Kg \end{aligned} \tag{3.4}$$

where V_{mass} and $V_{fingers}$ are the volumes of the proof mass and movable fingers respectively. Other parameters, such as W_{pm}, L_{pm} and T are structural parameters of the proof mass and comb fingers listed in Table 2.1.

Spring constant (K):

The suspension system of the mechanical sensing element consists of four cantilever springs as shown in Figure 3.2. The expression for the spring constant of each cantilever is given by [77]:

$$\begin{aligned} K_{cantilever} &= \frac{12EI_s}{L_s^3} = \frac{EW_s^3T}{L_s^3} \\ &= \frac{190 \times 10^9 \times (2 \times 10^{-6})^3 \times 2 \times 10^{-6}}{(176 \times 10^{-6})^3} \\ &= 0.56 N/M \end{aligned} \tag{3.5}$$

where E=170 × $10^9 N/m^2$ is the Young's modulus for polysilicon. I_s is the moment of inertia of the cantilever which is equal to $\frac{W_s^3T}{L_s^3}$. W_s, L_s and T represent the width, length, and thickness of the cantilever spring respectively. Their values are listed in Table 3.1. Because the proof mass is supported by four cantilevers of equal dimensions, each spring shares 1/4 of the total force load. Thus, the total mechanical spring constant is $4K_{cantilever}$.

$$K_{mechanical} = 4 \times K_{cantilever} = 4 \times 0.56 = 2.24 N/M \tag{3.6}$$

The calculated spring constant above does not take into account the electrostatic spring softening effect. In a sense unit shown in Figure 3.3, when a high frequency square modulation voltage $V_m(t)$ is applied to the fixed fingers, electrostatic forces are generated on the sense finger that lead to a change of the actual spring constant from its mechanical value. This phenomenon is regarded as electrostatic spring softening and is also included in our mechanical sensing element model. The net force on the sense finger (F_e) is given by [77]:

$$F_e = F_{e1} - F_{e2} = \frac{\varepsilon_0 A V_m^2}{2}\left[\frac{1}{(G-x)^2} - \frac{1}{(G+x)^2}\right] \tag{3.7}$$

where F_{e1} and F_{e2} are electrostatic forces, G is the initial gap between the sense finger and fixed fingers, x is the displacement of the sense finger, A is the area of the sense finger sidewall ($A = L_{sf}T$), ε_0 is dielectric constant and V_m is the amplitude of the modulation voltage (1V in this design).

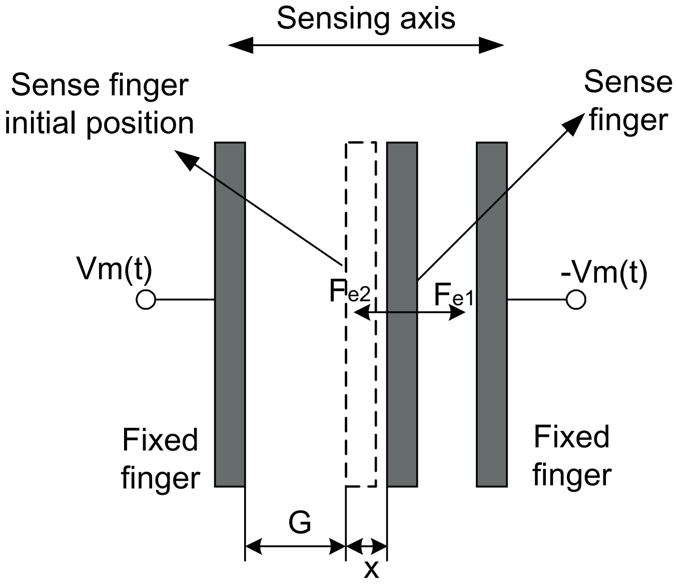


FIGURE 3.3: Electrostatic spring softening effect

The electrostatic spring constant can be calculated by differentiating Equation 3.7. Assuming $x << G$ and considering there are N_s sense units, the electrostatic spring constant is given by [77]:

$$\begin{aligned} K_e &= N_s(\frac{d(F_e)}{dx}) = -N_s(\frac{2\varepsilon_0 L_{sf} T V_m^2}{G^3}) \\ &= -54 \times \frac{2 \times 8.85 \times 10^{-12} \times 150 \times 10^{-6} \times 2 \times 10^{-6} \times 1^2}{(1.3 \times 10^{-6})^3} \\ &= -0.13 N/M \end{aligned} \tag{3.8}$$

Consequently, the effective spring constant is equal to:

$$K = K_{mechancial} + K_e = 2.24 - 0.13 = 2.11 N/M \tag{3.9}$$

Damping coefficient (D):

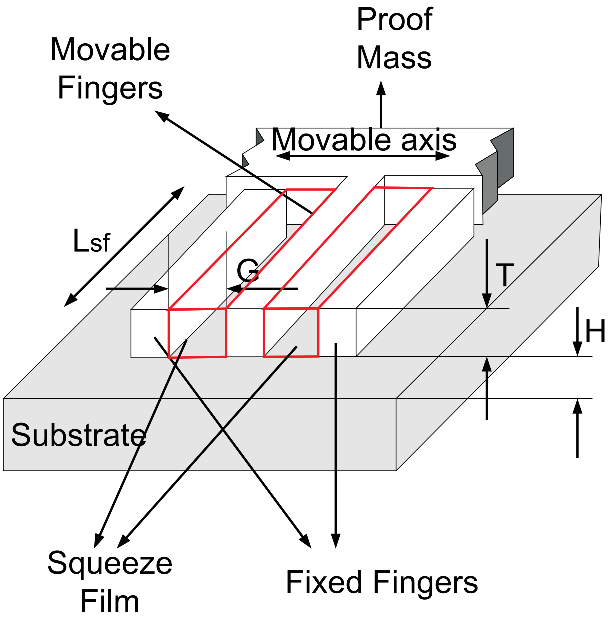


FIGURE 3.4: Schematic view of the arrangement for the calculation of the squeeze-film damping. For a displacement of the movable finger, the air in the gaps between the movable and the fixed fingers is compressed or expanded. L_{sf}, T and G are the length, thickness and width of the squeeze film

For a displacement of the movable structures, the gas in the gaps between the movable and the fixed structures is compressed or expanded and begins to stream. For this capacitive mechanical sensing element, squeeze-film damping between the comb-fingers, which is illustrated in Figure 3.4, usually dominates all other forms of damping. Squeeze-film damping can be modelled by assuming the Hagen-Poiseuille flow between comb-fingers [138]. Neglecting the fringing fields, the damping coefficient is given by:

$$\begin{aligned} D &= 14.4(N_f + N_s)\mu L_{sf}(\frac{T}{G})^3 \\ &= 14.4 \times (54+4) \times 1.85 \times 10^{-5} \times 150 \times 10^{-6}(\frac{2 \times 10^{-6}}{1.3 \times 10^{-6}})^3 \\ &= 8.44 \times 10^{-6} N \cdot s/m \end{aligned} \tag{3.10}$$

where L_{sf} and T represent the length and thickness of the movable fingers, G is the initial gap between fixed fingers and movable fingers, and μ is the viscosity coefficient of the air.

Performance parameters of the mechanical sensing element:

From the calculated lumped parameters, we can derive some performance parameters as illustrated in Section 2.3.4.

1. Resonant frequency f_0:

Recalling Equation 2.9, the resonant frequency(f_0) can be calculated from the mass(M) and the effective spring constant(K):

$$f_0 = \frac{1}{2\pi}\sqrt{\frac{K}{M}} = \frac{1}{2\pi}\sqrt{\frac{2.11}{3.32 \times 10^{-10}}} = 12.6KHz \tag{3.11}$$

There is only one resonant mode(12.6kHz) in the lumped mass-damper-spring system. In reality, the mechanical sensing element is a distributed element with many resonant modes. Our proposed distributed sensing element model, which will be discussed in Section 3.2, captures the higher resonant modes of the sensing element and provides more accurate simulation results than the conventional lumped model.

2. Quality factor Q:

Recalling Equation 2.10, the quality factor (Q) of the mechanical sensing element is derived from the lumped parameters, i.e. proof mass (M), spring constant (K) and damping coefficient (D):

$$Q = \frac{\sqrt{KM}}{D} = \frac{M\omega_0}{D} = \frac{\sqrt{2.11 \times 3.32 \times 10^{-10}}}{8.44 \times 10^{-6}} = 3.13 \tag{3.12}$$

As illustrated in Section 2.3.4, the dynamic response of the mechanical sensing element can be divided into three types according to the value of quality factor (Q): if $Q < 0.5$, the sensing element is over-damped; if $Q = 0.5$, it is critically damped; otherwise, it is under-damped. Thus, the mechanical sensing element designed here is under-damped ($Q > 0.5$).

Parameters	***Symbol***	***Value***
Input acceleration amplitude	a_{in}	$1g(1g = 9.8m/s^2)$
Input frequency	f_{in}	$1000Hz$
Signal bandwidth	f_0	$2048Hz$
Oversampling ratio	OSR	256
Oversampling frequency	f_s	$1.048576MHz$
Signal pick-off gain	K_{cv}	41×10^6
Boost gain	K_{amp}	37
Modulation voltage amplitude	V_m	$1V$
Feedback voltage	V_f	$0.8V$
Compensator zero frequency	$zero$	$5kHz$
Compensator pole frequency	$pole$	$250kHz$

Table 3.2: Design parameters of the second-order electromechanical Sigma-Delta modulator

3.1.1 VHDL-AMS implementation of the MEMS accelerometer with Sigma-Delta control

In this section, we will illustrate the VHDL-AMS implementation of the second-order electromechanical Sigma-Delta modulator where, conventionally, the mechanical sensing element is modeled as a second-order mass-spring-damper system. The VHDL-AMS models are implemented with the design parameters summarised in Table 3.2.

Listing 3.1 presents the VHDL-AMS code of the testbench architecture. In this Listing, the model contains seven components (based on the system diagram shown in Figure 3.1): acceleration source, mechanical sensing element, signal pick-off gain, voltage boost gain, lead compensator, quantiser and one-bit DAC. The acceleration source is used to generate input stimulus. A lead compensator is required to stabilize the system because the mechanical sensing element is under-damped.

```
library IEEE;
use IEEE.FUNDAMENTAL_CONSTANTS.all;
use IEEE.ELECTRICAL_SYSTEMS.all;
use IEEE.MECHANICAL_SYSTEMS.all;
use IEEE.MATH_REAL.all;
use IEEE.STD_LOGIC_1164.all;

entity test_ACCELEROMETER is
end entity test_ACCELEROMETER;

architecture testbench of test_ACCELEROMETER is
    quantity a: ACCELERATION;
    quantity d: DISPLACEMENT;
    quantity V1,V2,vp,vb,como: VOLTAGE;
    signal output: std_logic;

begin
  Acceleration:  entity a_source(SINE)
                 generic map (MAG=>1.0*PHYS_GRAVITY, FREQ=>1000.0)
                 port map (op=>a);

  Sensing:       entity sensing_element
                 generic map (Wpm=>120.0e-6, Lpm=>450.0e-6, T=>2.0e-6,
                              Ls=>176.0e-6,  Ws=>2.0e-6,  Lsf=>150.0e-6,
                              Wsf=>2.0e-6, Lff=>150.0e-6, Wff=>2.0e-6,
                              G=>1.3e-6, G2=>1.3e-6, Ns=>54.0,
                              Nf=>4.0,Vm=1.0)
                 port map (ain=>a,Vf1=>V1,Vf2=>V2,pos=>d);

  Pick-off_gain: entity gain
                 generic map (K=>41.0e6)
                 port map (ip=>d, op=>vp);

  Boost_gain:    entity gain
                 generic map (K=>37)
                 port map (ip=>vp, op=>vb);

  Compensation:  entity compensator
                 generic map (Zero=>5000.0, Pole=>250000.0)
                 port map(ip=>vb, op=>como);
```

```

Q :             entity quantizer
                generic map (Fs=>2048.0*256.0*2.0)
                port map(ip=>como, op=>output);

DAC:            entity WORK.DAC(bhv)
                port map (ip=>output,op1=>V1,op2=>V2);
end architecture testbench;
```

Listing 3.1: VHDL-AMS testbench of second-order electromechanical Sigma-Delta modulator

The VHDL-AMS implementation of the mechanical sensing element model is shown in Listing 3.2 (Implementations of other components can be found at the Southampton VHDL-AMS Validation Suite website [97]). The generic parameters, listed in the entity declaration, are the dimension parameters of the mechanical sensing element (Table 3.1) and amplitude of the modulation voltage (Table 3.2). The interface ports of the model are declared by four interface quantities, i.e. **ain**, **Vf1**, **Vf1**, and **Pos**. The input quantity **ain** is the input acceleration generated by acceleration source, while the input quantities **Vf1** and **Vf2** represent the feedback voltages obtained from the one-bit DAC. The output of the model is the relative displacement of the proof mass (**Pos**). The architecture of the mechanical sensing element model contains the equations for the calculation of lumped parameters and the lumped second-order ODE (Equation 3.2) to model the behaviour of the system. It is worth noting that each quantity in the model is defined by its physical name, such as displacement, acceleration, damping, etc, by using the IEEE 1076.1.1 multiple energy domain standard packages. These names are connected with their corresponding physical natures.

```
library IEEE;
use IEEE.MECHANICAL_SYSTEMS.all;
use IEEE.FUNDAMENTAL_CONSTANTS.all;
use IEEE.MATERIAL_CONSTANTS.all;
use IEEE.MATH_REAL.all;

entity sensing_element is
  generic( --Dimension of mechanical sensing element--
         Wpm:real:=120.0e-6; Lpm:real:=450.0e-6; T:real:=2.0e-6;
```

```
         Ls:real:=176.0e-6;  Ws:real:=2.0e-6;  Lsf:real:=150.0e-6;
         Wsf:real:=2.0e-6; Lff:real:=150.0e-6;  Wff:real:=2.0e-6;
         G:real:=1.3e-6; G2:real:=1.3e-6; Ns:real:=54.0;
         Nf:real:=4.0; Vm:voltage:=1.0);
  port( quantity ain : in ACCELERATION;  --Input acceleration
        quantity Vf1 : in VOLTAGE;   --Feedback voltage to top fixed
                                     --fingers in force units
        quantity Vf2 : in VOLTAGE;   --Feedback voltage to bottom fixed
                                     --fingers in force units
        quantity pos : out DISPLACEMENT);--Displacement of proof mass
end entity sensing_element;

architecture behav of sensing_element is
        quantity M:MASS;
        quantity K:STIFFNESS;
        quantity Kmechanical:STIFFNESS;
        quantity Ke:STIFFNESS;
        quantity D:DAMPING;
        quantity Ff:FORCE;
begin
     --Mass of sensing element--
      M==PHYS_RHO_POLY*(Wpm*Lpm*T+(Ns+Nf)*Lsf*Wsf*T);

     --Mechanical spring--
      Kmechanical==4.0*PHYS_E_POLY*Ws*Ws*Ws*T/(Ls*Ls*Ls);

     --Electrosttic spring--
      Ke==-1.0*Ns*(2.0*PHYS_EPS0*Lsf*T*Vm*Vm)/(G*G*G);

     --Effective spring constant--
      K==Kmechanical+Ke;

     --Damping coefficient--
      D==14.4*(Ns+Nf)*1.85e-5*T*Lsf*Lsf*Lsf/(G*G*G);

     --Feedback force--
      Ff==0.5*Nf*PHYS_EPS0*Lff*T*(Vf1*Vf1-Vf2*Vf2)/(G2*G2);

     --Behaviour of mechanical sensing element--
      M*pos'DOT'DOT+D*pos'DOT+K*pos==M*ain+Ff;

end architecture behav;
```

Listing 3.2: Conventional VHDL-AMS model of the mechanical sensing element

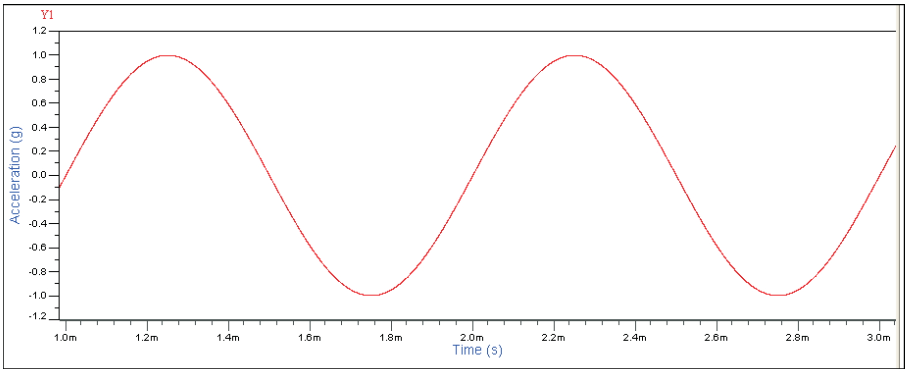
Simulations were carried out using the SystemVision VHDL-AMS simulator from Mentor. The system input was a 1kHz sine wave acceleration with 1g amplitude as shown in Figure 3.5(a). Figure 3.5(b) shows the output bitstream of the electromechanical Sigma-Delta modulator. As illustrated in the figure, the pulse density is inversely proportional to the input signal which means that the Sigma-Delta control works. Because the input force is balanced by the feedback, the proof mass almost holds its initial position with minor displacement (about 0.2nm) as shown in Figure 3.5(c).

3.1.2 SystemC-A implementation of the MEMS accelerometer with Sigma-Delta control

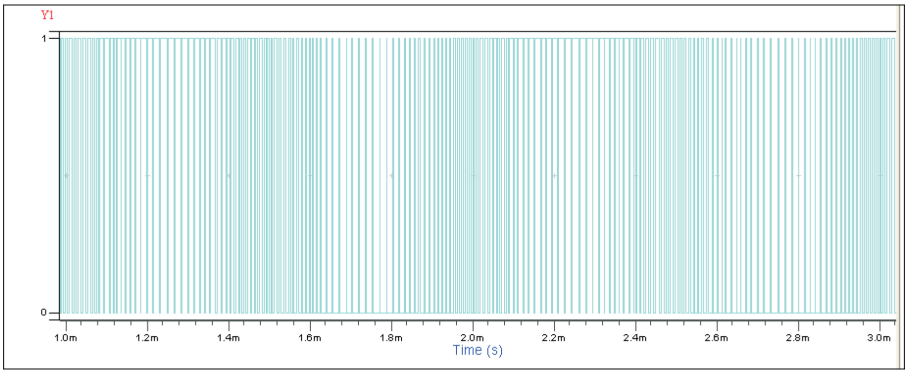
The second-order electromechanical Sigma-Delta modulator has also been modelled in SystemC-A for comparison. The main components of the system (i.e. mechanical sensing element, compensator, DAC, etc.) are modelled as individual modules together with a testbench. The testbench is shown in Listing 3.3 where all the components are connected together by signals. The SystemC-A models are implemented with the same design parameters as those in VHDL-AMS models (Table 3.2).

```
void testbench::system(){
 //Connecting signals
   sc_signal<double> ain,Vf1,Vf2,d,Vp,Vb,como,bitout;

 //components netlist
  AcceleratonS_sin *Ain     =new AccelerationS_sin("Ain", &ain,1000.0,1.0*9.8);
   Sensing_Element *Sensing =new Sensing_Element("Sensing",&ain,&Vf1,&Vf2,&d);
   Pick_off_gain   *Pick    =new Pick_off_gain("Pick",&d,&Vp);
   Boost_gain      *Boost   =new Boost_gain("Boost",&Vp,&Vb);
   Compensator     *Com     =new compensator("Com",&Vb,&como);
   Quantiser       *Q       =new comparator("Q",&como,&bitout);
   1_bit_DAC       *DAC     =new comparator("DAC",&bitout,&Vf1,&Vf2);

  sc_start(0.04,SC_SEC); //Simulation time 0.04Sec.
  }
```

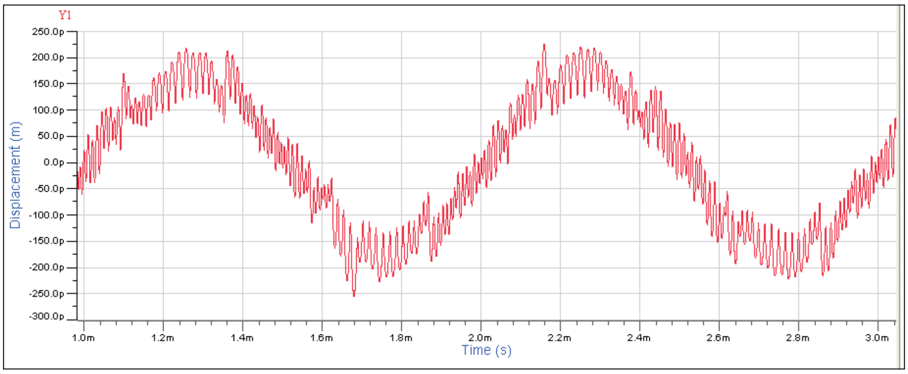
Listing 3.3: SystemC-A testbench of the second-order electromechanical Sigma-Delta modulator



(a) Input acceleration: a sine wave acceleration with 1kHz frequency and 1g ($9.8m/s^2$) amplitude



(b) Output bitstream of electromechanical Sigma-Delta modulator



(c) Displacement of the proof mass

FIGURE 3.5: Simulation results of the conventional VHDL-AMS model of the second-order electromechanical Sigma-Delta modulator in response to a sinusoidal acceleration with 1g amplitude and 1kHz frequency

Listing 3.4 shows the SystemC-A implementation of the mechanical sensing element. In this Listing, the mechanical sensing element is modeled as a SystemC-A component (*Sensing_Element*) which is derived from an abstract base class (*sc_a_component*). The components constructor defines the components I/O ports, quantities and dimensional parameters. The associated *Build* method of the *Sensing_Element* component is used to model the ODAEs of the system. The associated function with *Build*() is *Equation*(), which is used to describe a first-order ODE. In order to model the second-order ODE (mass-spring-damper system) in SystemC-A, the equation should first be reduced to two first-order ODEs as shown in the Listing 3.4 (lines 43-47). In SystemC-A, function $X()$ (lines20-21) returns the value of a quantity, and $Xdot()$ (lines 22-23) performs the differentiator operation on a quantity.

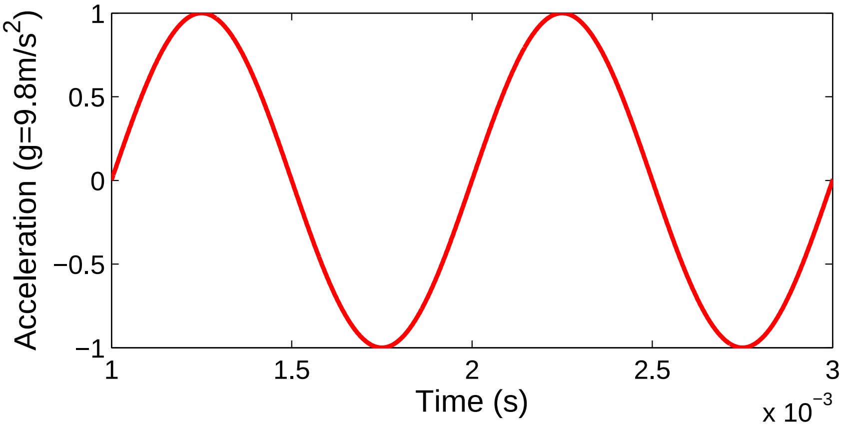
```
1 Sensing_Element::Sensing_Element(char nameC[5],sc_signal<double>*ain,
2        sc_signal<double>*Vf1,sc_signal<double>*Vf2,sc_signal<double>*pos):
3    component(nameC,0, 0, 0){
4        pos_sig=pos;
5        ain_sig=Input;
6        Vf1_sig =Vf1; Vf2_sig =Vf2;
7        y1 = new Quantity("y1");  //y1: displacement of proof mass
8        y2 = new Quantity("y2");  //y2=y1'
9      //Design parameters of mechanical sensing elemet//
10        Wpm=120.0e-6; Lpm=450.0e-6; T=2.0e-6; Ls=176.0e-6;
11        Ws=2.0e-6; Lsf=150.0e-6; Wsf=2.0e-6;  Lff=150.0e-6;
12        Wff=2.0e-6; G=1.3e-6; G2=1.3e-6; Ns=54.0; Nf=4.0; Vm=1.0;
13    }
14
15 void Sensing_Element::Build(void){
16
17      pos_sig->write(Y1n);           //Output: Displacement of Proof Mass
18      ain=ain_sig->read();           //Input: Acceleration
19      Vf1=Vf1_sig->read(); Vf2=Vf2_sig->read();    //Input: Feedback voltages
20      Y1n=X(y1);        //X():read value of a Quantity
21      Y2n=X(y2);
22      Y1dotn=Xdot(y1);  //Xdot():performs differentiator operation on Quantity
23      Y2dotn=Xdot(y2);
24
25    //Mass of sensing element//
26       M=2330.0*(Wpm*Lpm*T+(Ns+Nf)*Lsf*Wsf*T);
27
28    //Mechanical spring//
29       Kmechanical=4.0*12.0*Ws*Ws*Ws*T/(Ls*Ls*Ls);
30
31    //Electrosttic spring//
32       Ke=-1.0*Ns*(2.0*8.85e-12*Lsf*T*Vm*Vm)/(G*G*G);
33
34    //Effective spring constant//
35       K=Kmechanical+Ke;
```

```

    //Damping coefficient//
       D=14.4*(Ns+Nf)*1.85e-5*T*Lsf*Lsf*Lsf/(G*G*G);

    //Feedback force//
       Ff=0.5*Nf*8.85e-12*Lff*T*(Vf1*Vf1-Vf2*Vf2)/(G2*G2);

   //-------2nd-order ODE is divided into two 1st-order ODEs------//
   //------y1'=y2;                                     -----------//
   //------y2'=f/M-(D/M)*y2-(K/M)*y1;  -----------//
        Equation(y1,-Y1dotn + Y2n);
        Equation(y2,-Y2dotn +(M*ain+Ff)/M-(D/M)*Y2n-(K/M)*Y1n);
}
```

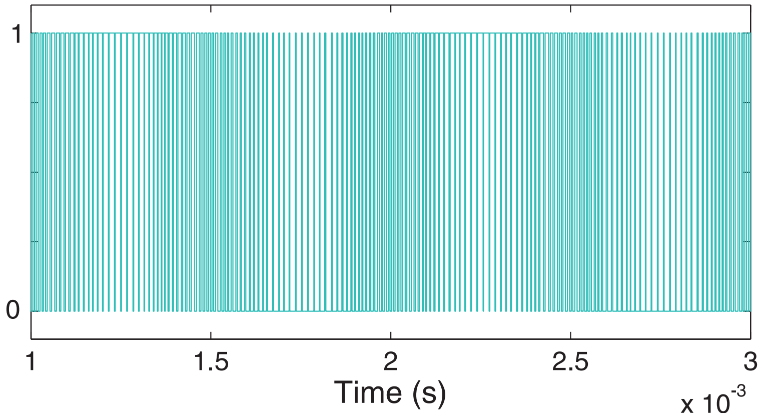
Listing 3.4: SystemC-A implementation of the conventional mechanical sensing element model

The SystemC-A model of the second-order electromechanical Sigma-Delta modulator is simulated using the same stimulus as used in the VHDL-AMS model simulation, i.e. a sinusoidal acceleration with 1kHz frequency and 1g amplitude. As shown in Figure 3.6, simulation results are all consistent with those of the VHDL-AMS model.

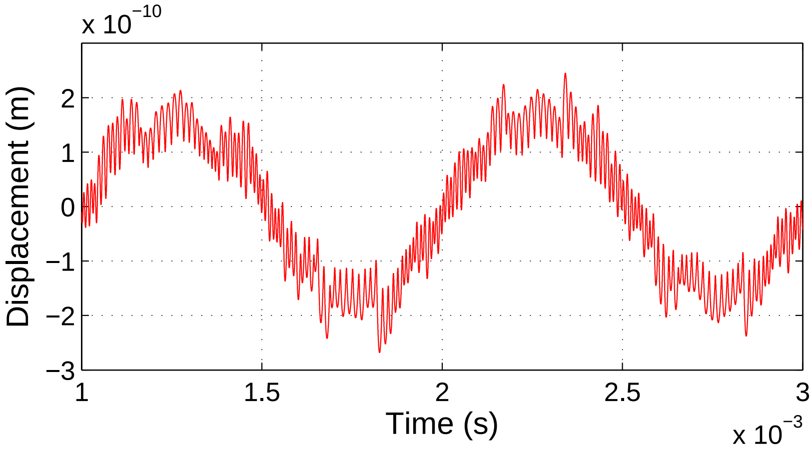
The signal-to-noise ratio (SNR) is one of the most important parameters in evaluating the performance of the electromechanical Sigma-Delta modulator. It can be derived from analysing the power spectral density (PSD), which is calculated from the Fast Fourier Transform (FFT) of the output bitstream. However, SystemVision, the VHDL-AMS simulator used in this research, does not support text I/O operations which means it is difficult to export output results for post-simulation data processing. In contrast, SystemC-A is a flexible modelling language where the implementation of postprocessing of simulation results is quite easy. Figure 3.7 shows the PSD of the output bitstreams of the second-order electromechanical Sigma-Delta modulator. As this figure shows, the Sigma-Delta control loop works correctly, and the noise in the signal band is dynamically decreased by the oversampling and noise shaping techniques. A peak at a frequency about 70kHz indicates the maximum unity-gain frequency needed for a stable closed-loop operation.



(a) Input acceleration



(b) Output bitstream of electromechanical Sigma-Delta modulator



(c) Displacement of the proof mass

Figure 3.6: Time-domain simulation results of the SystemC-A model of a second-order electromechanical Sigma-Delta modulator in response to a sinusoidal acceleration with 1g amplitude and 1KHz frequency

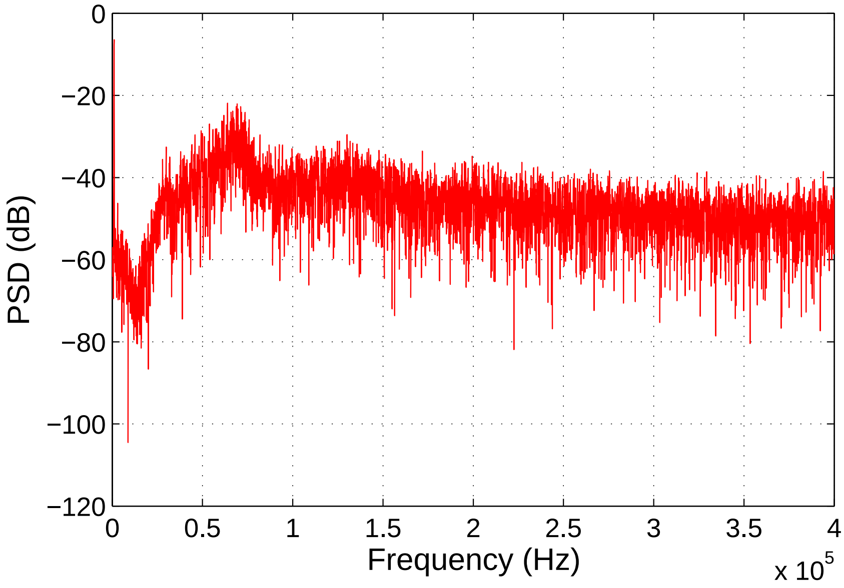


FIGURE 3.7: Power spectral density of the output bitstream derived from the conventional second-order electromechanical Sigma-Delta modulator model in response to a sinusoidal acceleration with 1g amplitude and 1kHz (sense finger length $L_{sf} = 150\mu m$). PSD is obtained by 65536 point FFT of the output bitstream using hanning window

3.2 Accurate mechanical sensing element model with sense finger dynamics

3.2.1 Influence of sense finger dynamics

The proof mass of a capacitive accelerometer is equipped with sense fingers placed between fixed capacitive plates to form capacitive bridges (Figure 3.2). The drawback of this configuration is that the bending of sense fingers relative to the proof mass can significantly affect the performance of the electromechanical Sigma-Delta control loop [20]. Sense fingers, which are excited by feedback, might bend seriously and oscillate at their resonant frequencies sometimes leading to a failure of the Sigma-Delta control loop.

However, the effects caused by the sense finger dynamics, cannot be captured by the conventional mechanical sensing element model discussed in Section 3.1. The conventional model only contains the dynamic of the lumped proof mass, which is modelled by the mass-damper-spring system. This means the sense fingers are treated as rigid bodies moving together with the lumped mass without bending. Recalling Equation 3.11, the resonant frequency of the conventional lumped mechanical model is approximately by:

$$f_0 = \frac{1}{2\pi}\sqrt{\frac{K}{M}} \tag{3.13}$$

where K is the effective spring constant, and M is the mass of the sensing element.

In reality, the sensing element is a distributed element with many resonant modes. The lowest resonant mode is the same as that in the conventional model (Equation 3.13) and corresponds to the sense fingers moving with the proof mass with minor bending. The higher resonant modes are related to the sense finger resonances, at which the sense fingers bend significantly while the lumped mass only has a small deflection. The sense finger resonant frequencies can be approximated by those of a cantilevered beam with a rectangular cross-section. The first two resonant frequencies of a sense finger are given by [20]:

$$f_{ri} = \frac{1}{2\pi}\alpha_i^2 \frac{W_{sf}}{L_{sf}^2}\sqrt{\frac{E}{12\rho}} \quad i = 1, 2 \tag{3.14}$$

$$\alpha_1 = 1.875 \quad \alpha_2 = 4.694 \tag{3.15}$$

where i is the mode index number, E is Youngs modulus of polysilicon, and ρ is the material density. W_{sf} and L_{sf} are the width and length of the sense fingers respectively. The sense finger dimensions in this design are: $L_{sf} = 150\mu m$, $W_{sf} = 2\mu m$. Therefore, the first three resonant frequencies, derived from Equation 3.13-3.15 are: $f_0 = 12.7kHz$, $f_{r1} = 122.7kHz$ and $f_{r2} = 769kHz$.

It is reported that there is a limit on the first resonant frequency of sense finger in order not to degrade the Sigma-Delta control performance [20]. If the first sense finger resonant frequency is near to the unity-gain frequency, the Sigma-Delta loop might oscillate at the finger resonant frequency and the control loop will break down [20]. As shown in Figure 3.7, the unity-frequency of the system is around 70kHz which is much less than the finger resonance(122.7kHz). Thus, the finger dynamics do not affect this implementation too much. However, they would become significant if the length of the sense fingers is increased, i.e. the resonant frequencies of sense fingers are decreased (Equation 3.14).

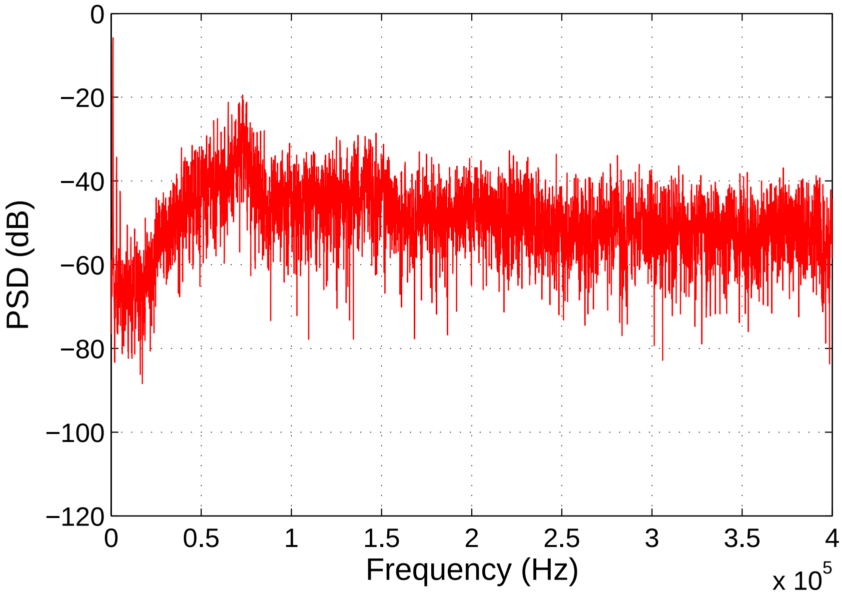


FIGURE 3.8: Power spectral density of the output bitstream derived from the conventional electromechanical Sigma-Delta modulator model in response to a sinusoidal acceleration with 1g amplitude and 1kHz frequency (sense finger length $L_{sf} = 190\mu m$). PSD is obtained by 65536 point FFT of the output bitstream using hanning window

As derived from Equation 3.14, if the sense finger length is above $190\mu m$, the first resonant frequency approximately equals the unity-gain frequency of the system. In this case, the sense finger may resonate, which results in a failure of the Sigma-Delta control. This effect has already been illustrated in a electromechanical Sigma-Delta force-feedback gyroscope [20]. However, as shown in the VHDL-AMS and the SystemC-A implementations of the conventional mechanical sensing element, the change of sense finger length only leads to a minor change of the lumped parameters, i.e. mass, effective spring and damping coefficient. The PSD of the output bitstream of the conventional second-order electromechanical Sigma-Delta modulator model, when the finger length has been increased to $190\mu m$, is shown by the trace in Figure 3.8. Simulation results indicate that the conventional model fails to reflect the correct behaviour because it shows a correct operation where in fact the Sigma-Delta control breaks down.

3.2.2 Distributed model of the mechanical sensing element

This section presents an improved distributed mechanical sensing element model implemented in VHDL-AMS and SystemC-A. The model includes sense finger dynamics which provides accurate performance predictions of a MEMS capacitive accelerometer in a mixed-technology control loop.

The proposed distributed model is derived from the geometry of the sense electrode as illustrated in Figure 3.9. C_{s1} and C_{s2} are the total distributed differential capacitances between the beam and the electrodes. $V_m(t)$ is the high frequency square modulation voltage.

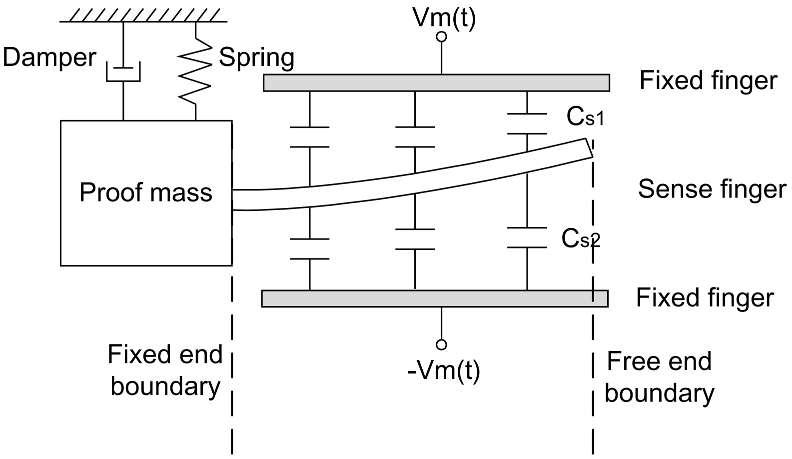


FIGURE 3.9: Distributed model for mechanical sensing element

For the purpose of accurate modelling, the motion of the sense finger can be described by Euler-Bernoulli equation [139]:

$$\rho S\frac{\partial^2 y(x,t)}{\partial t^2} + C_D I\frac{\partial^5 y(x,t)}{\partial x^4 \partial t} + EI\frac{\partial^4 y(x,t)}{\partial x^4} = F_e(x,t) \tag{3.16}$$

where $y(x,t)$ is a function of time and position, and represents the deflection of the sense finger. E, I, C_D, ρ, S are all physical properties of the beam. ρ is the material density, S is the cross sectional area ($W_{sf} * T$), W_{sf} and T are finger's width and thickness, E represents the Young's modulus, which defines a material's shearing strength, I is the second moment of area which could be calculated by $I = W_{sf}T^3/12$, EI is usually regarded as the flexural stiffness, C_D is the internal damping modulus, and $F_e(x,t)$ is the distributed electrostatic force along the finger.

The boundary conditions at the clamped end and the free end of the sense finger are shown in the following equations.

At the clamped end (x=0):

$$y(0,t) = z(t) \tag{3.17}$$

$$\theta = \frac{\partial y(0,t)}{\partial x} = 0 \tag{3.18}$$

and at the free end (x=L_{sf}):

$$BM = EI\frac{\partial^2 y(L_{sf},t)}{\partial x^2} = 0 \tag{3.19}$$

$$F_s = -EI\frac{\partial^3 y(L_{sf},t)}{\partial x^3} = 0 \tag{3.20}$$

where θ, BM and F_s denote the slope angle, the bending moment and the shear force respectively, and L_{sf} is the sense finger length.

The clamped end of a sense finger y(0,t) moves with the proof mass whose deflection could be modelled by the mass-spring-damper system:

$$M\frac{d^2x(t)}{dt^2} + D\frac{dx(t)}{dt} + Kx(t) = F_{feedback}(t) + Ma(t) \tag{3.21}$$

where x(t) is the proof mass deflection, M, D and K are the mass, damping coefficient and spring constant respectively; $F_{feedback}(t)$ is the electrostatic feedback while $a(t)$ represents an external input acceleration to the system.

For VHDL-AMS and SystemC-A implementation, the Finite Difference Approximation (FDA) approach is applied to convert the partial differential equation to a series of ODAEs to overcome the limitations of VHDL-AMS and SystemC-A where partial equations cannot directly be modelled. If the sense finger is divided into N segments, the deflection of the finger is discretised as:

$$y_n(t) = y(n\triangle x, t) \quad n = 1, 2, 3...N \tag{3.22}$$

where $y_n(t)$ is the deflection of segment n, and $\triangle x$ is the discretisation step size which equals to $\frac{L_{sf}}{N}$. The spatial derivatives, hence, can be approximated by finite differences:

For the first-order spatial derivatives:

$$\frac{\partial y_n(t)}{\partial x} = \frac{y_n(t) - y_{n-1}(t)}{\triangle x} \quad n = 1, 2...N \tag{3.23}$$

For the second-order spatial derivatives:

$$\frac{\partial^2 y_n(t)}{\partial x^2} = \frac{y_{n+1}(t) - 2y_n(t) + y_{n-1}}{\triangle x^2} \quad n = 1, 2...N \tag{3.24}$$

For the third-order spatial derivatives:

$$\frac{\partial^3 y_n(t)}{\partial x^3} = \frac{y_{n+2}(t) - 3y_{n+1} + 3y_n(t) - y_{n-1}}{\triangle x^3} \quad n = 1, 2...N \tag{3.25}$$

For the fourth-order spatial derivatives:

$$\frac{\partial^4 y_n(t)}{\partial x^4} = \frac{y_{n+2}(t) - 4y_{n+1} + 6y_n(t) - 4y_{n-1} + 6y_{n-2}}{\triangle x^4} \quad n = 1, 2...N \tag{3.26}$$

Without considering boundary conditions, Equation 3.16 is transformed into a series of second-order ODEs:

$$\begin{aligned}\rho S\frac{d^2y_n}{dt^2}+\frac{C_DI}{(\triangle x)^4}(\frac{dy_{n+2}}{dt}-4\frac{dy_{n+1}}{dt}+6\frac{dy_n}{dt}-4\frac{dy_{n-1}}{dt}+\frac{dy_{n-2}}{dt})\\+\frac{EI}{(\triangle x)^4}(y_{n+2}-4y_{n+1}+6y_n-4y_{n-1}+y_{n-2})=\frac{fe_n(t)}{\triangle x}\quad n=1,2...N\end{aligned}\tag{3.27}$$

Boundary conditions provide additional equations. The slope angle at the fixed end (n=1) is approximated as:

$$\theta=\frac{\partial y_1(t)}{\partial x}=\frac{y_1(t)-2y_0}{\triangle x}=0\tag{3.28}$$

and the bending moment BM and shear force F_s at the free end (n=N) as:

$$BM=-\frac{\partial^2 y_N(t)}{\partial x^2}=-\frac{y_{N+1}(t)-2y_N(t)+y_{N-1}(t)}{\triangle x^2}=0\tag{3.29}$$

$$F_s=-\frac{\partial^3 y_1(t)}{\partial x}=-\frac{y_{N+2}(t)-3y_{N+1}+3y_N-y_{N-1}}{\triangle x}=0\tag{3.30}$$

The governing PDE of the sense finger with boundary conditions is hence converted to the following ODAEs:

For segment 1:

$$y_1(t)=x(t)\tag{3.31}$$

For segment 2:

$$\begin{aligned}\rho S\frac{d^2y_2}{dt^2}+\frac{C_DI}{(\triangle x)^4}(\frac{dy_4}{dt}-4\frac{dy_3}{dt}+6\frac{dy_2}{dt}-3\frac{dy_1}{dt})\\+\frac{EI}{(\triangle x)^4}(y_4-4y_3+6y_2-3y_1)=\frac{fe_2(t)}{\triangle x}\end{aligned}\tag{3.32}$$

For segments n=3,4,5...N-2:

$$\begin{aligned}\rho S\frac{d^2y_n}{dt^2}+\frac{C_DI}{(\triangle x)^4}(\frac{dy_{n+2}}{dt}-4\frac{dy_{n+1}}{dt}+6\frac{dy_n}{dt}-4\frac{dy_{n-1}}{dt}+\frac{dy_{n-2}}{dt})\\+\frac{EI}{(\triangle x)^4}(y_{n+2}-4y_{n+1}+6y_n-4y_{n-1}+y_{n-2})=\frac{fe_n(t)}{\triangle x}\end{aligned}\tag{3.33}$$

For segment N-1:

$$\begin{aligned}\rho S\frac{d^2y_{N-1}}{dt^2}+\frac{C_DI}{(\triangle x)^4}(-2\frac{dy_N}{dt}+5\frac{dy_{N-1}}{dt}-4\frac{dy_{N-2}}{dt}+\frac{dy_{N-3}}{dt})\\+\frac{EI}{(\triangle x)^4}(-2y_N+5y_{N-1}-4y_{N-2}+y_{N-3})=\frac{fe_{N-1}(t)}{\triangle x}\end{aligned}\tag{3.34}$$

For segment N:

$$\begin{aligned}\rho S\frac{d^2y_n}{dt^2}+\frac{C_DI}{(\triangle x)^4}(\frac{dy_N}{dt}-2\frac{dy_{N-1}}{dt}+\frac{dy_{N-2}}{dt})\\+\frac{EI}{(\triangle x)^4}(y_N-2y_{N-1}+y_{N-2})=\frac{fe_N(t)}{\triangle x}\end{aligned}\tag{3.35}$$

Equation 3.31 represents the motion of the clamped end of the sense finger ($y_1(t)$) which moves with the lumped proof mass whose deflection x(t) is obtained from the solution of Equation 3.21.

3.2.3 VHDL-AMS implementation of the distributed mechanical sensing element model

The VHDL-AMS implementation of the proposed distributed mechanical sensing element model, which includes sense finger dynamics, is shown in the Listing 3.5. As the entity of this model is the same as that of conventional model (Listing 3.2), it is not given in detail here. The sense finger is discretised into ten segments and the differential coefficient dx represents the step size ($L_{sf}/10$). Quantities y1 to y10 (line 13) represent the displacements of discretised segments. From the PDE and the boundary conditions, ten ODEs are created to describe the distributed behaviour of the sensing element. The dimensional parameters of the mechanical sensing element used in this model are the same as those used in the conventional model (Listing 3.2).

```
1 library IEEE;
2 use IEEE.MECHANICAL_SYSTEMS.all;
3 use IEEE.FUNDAMENTAL_CONSTANTS.all;
4 use IEEE.MATERIAL_CONSTANTS.all;
5 use IEEE.MATH_REAL.all;
6
7  entity sensing_element is
```

```
        ...
end entity sensing_element;

architecture behav of sensing_element is
     --The displacement of each segment--
        quantity Y1,Y2,Y3,Y4,Y5,Y6,Y7,Y8,Y9,Y10:DISPLACEMENT;
     --The distributed electrostatic force--
          quantity FE2,FE2,FE3,FE4,FE5,FE6,FE7,FE8,FE9,FE10:FORCE;
          constant N:real:=10.0;
          constnat dx:real:=Lsf/N;
              ...
begin
     --Mass of sensing element--
      M==PHYS_RHO_POLY*(Wpm*Lpm*T+(Ns+Nf)*Lsf*Wsf*T);
     --Mechanical spring--
      Kmechanical==4.0*PHYS_E_POLY*Ws*Ws*Ws*T/(Ls*Ls*Ls);
     --Electrosttic spring--
      Ke==-1.0*Ns*(2.0*PHYS_EPS0*Lsf*T*Vm*Vm)/(G*G*G);
     --Effective spring constant--
      K==Kmechanical+Ke;
     --Damping coefficient--
      D==14.4*(Ns+Nf)*1.85e-5*T*Lsf*Lsf*Lsf/(G*G*G);
     --Feedback force--
      Ff==-0.5*Nf*PHYS_EPS0*Lff*T*(Vf1*Vf1-Vf2*Vf2)/(G2*G2);
     --Distributed electrostatic forces along the sense finger--
     FE2==0.5*PHYS_EPS0*T*dx*(Vm*Vm/((d0-y2)**2)-Vm*Vm/((d0+y2)**2));
     FE3==0.5*PHYS_EPS0*T*dx*(Vm*Vm/((d0-y3)**2)-Vm*Vm/((d0+y3)**2));
                ...
     FE9==0.5*PHYS_EPS0*T*dx*(Vm*Vm/((d0-y9)**2)-Vm*Vm/((d0+y9)**2));
     FE10==0.5*PHYS_EPS0*T*dx*(Vm*Vm/((d0-y10)**2)-Vm*Vm/((d0+y10)**2));
     --Proof mass motion equation--
     M*X'DOT'DOT+D*X'DOT+K*pX==M*ain+Ff;
    --The root segment displacement--
     y1==X;
    --The second segment motion equation--
PHYS_E_POLY*I*(Y4-4.0*Y3+6.0*Y2-3.0*Y1)/dx**4+PHYS_RHO_POLY*S*Y2'DOT'DOT+
CD*(Y4'DOT-4.0*Y3'DOT+6.0*Y2'DOT-3.0*Y1'DOT)/dx**4==FE2/dx;
     --The third segment motion equation--
PHYS_E_POLY*I*(Y5-4.0*Y4+6.0*Y3-4.0*Y2+Y1)/dx**4+PHYS_RHO_POLY*S*Y3'DOT'DOT
+CD*(Y5'DOT-4.0*Y4'DOT+6.0*Y3'DOT-4.0*Y2'DOT+Y1'DOT)/dx**4==FE3/dx;
       ...
    --The ninth segment motion equation--
PHYS_E_POLY*I*(-2.0*Y10+5.0*Y9-4.0*Y8+Y7)/dx**4+PHYS_RHO_POLY*S*Y9'DOT'DOT+
CD*(-2.0*Y10'DOT+5.0*Y9'DOT-4.0*Y8'DOT+Y7'DOT)/dx**4==FE9/dx;
```

```
    --The free end segment motion equation--
PHYS_E_POLY*I*(Y10-2.0*Y9+Y8)/dx**4+PHYS_RHO_POLY*S*Y10'DOT'DOT+CD*(Y10'DOT
-2.0*Y9'DOT+Y8'DOT)/dx**4==FE10/dx;
    --Average displacement of sense finger--
    pos==(Y1+Y2+Y3+Y4+Y5+Y6+Y7+Y8+Y9+Y0)/10.0;
end architecture behav;
```

Listing 3.5: VHDL-AMS code of distributed mechanical sensing element

In order to analyse the influence of the sense finger dynamics on the Sigma-Delta control system's performance, the conventional sensing element model illustrated in Listing 3.2 is replaced by the proposed distributed one, and time-domain simulation of the entire second-order electromechanical Sigma-Delta modulator system has been carried out with the same stimulus as in the conventional model design. Figure 3.10, which shows the input acceleration and output bitstream of the system, indicates that the Sigma-Delta control loop still works correctly, i.e the output pulse density is inversely proportional to the input signal. Figure 3.11, which plots the deflections of the proof mass and free end of the sense finger, shows that the sense finger moves with the proof mass with minor bending.

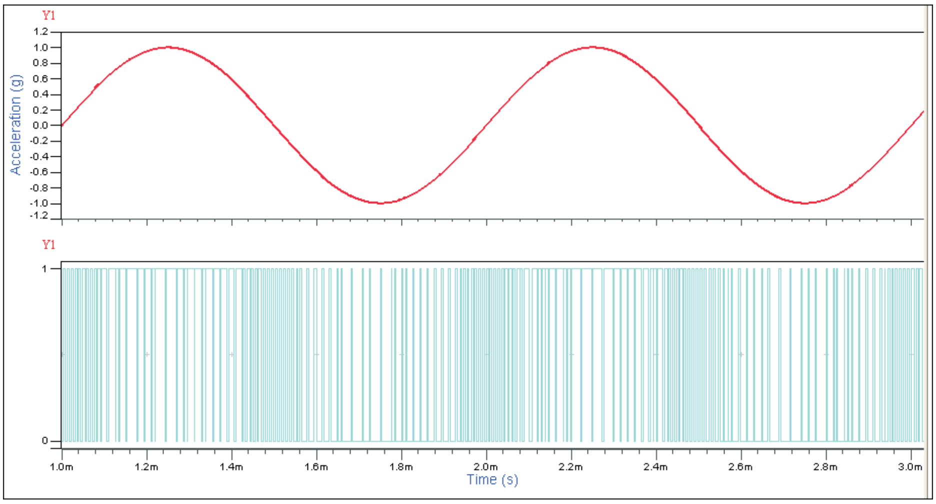


Figure 3.10: Simulation results of the improved VHDL-AMS model of the second-order electromechanical Sigma-Delta modulator ($L_{sf} = 150\mu m$) in response to a sinusoidal acceleration with 1g amplitude and 1kHz frequency. Top trace: input acceleration; Bottom trace: output bitstream

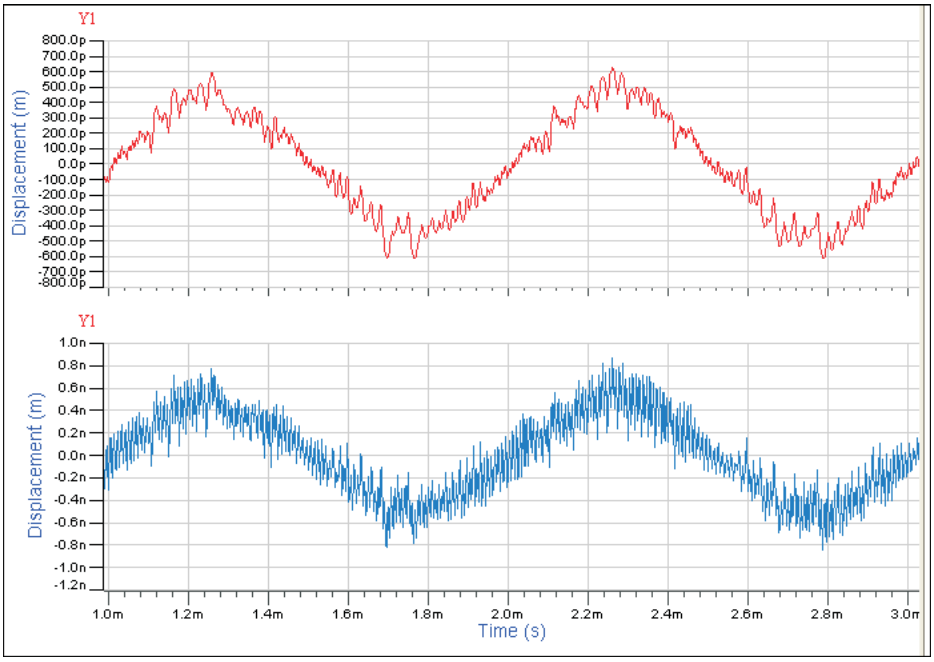


FIGURE 3.11: Simulation results of the improved VHDL-AMS model of the second-order electromechanical Sigma-Delta modulator ($L_{sf} = 150\mu m$) in response to a sinusoidal acceleration with 1g amplitude and 1kHz frequency. Top trace: displacements of the proof mass; Bottom trace: displacements of the free end of the sense finger

As discussed in Section 3.2.1, in order to assure correct control loop operation, the finger length should be as short as possible to keep the resonant frequencies of the sense finger away from the unity-gain frequency of the control loop. However, shorter fingers cause smaller capacitances and therefore lower sensitivity as demonstrated in Equation 2.11. The analysis below provides modelling guidelines to facilitate correct trade-offs in the calculation of the sense finger lengths when designing practical MEMS accelerometers based on electromechanical Sigma-Delta control.

A series of simulations were conducted using the same experimental environment as above but with a different sense finger length. The simulations indicate that correct behaviour of the Sigma-Delta control in the studied design is assured when the finger length does not exceed $190\mu m$. Figure 3.12 and Figure 3.13 display the simulation results when the finger length was increased to $190\mu m$. As shown in Figure 3.12, the sense finger bends significantly and resonates at its first resonant frequency which results in a breakdown of the Sigma-Delta control. Displacement of the proof mass is also increased to 1.6nm (only 0.6nm when the finger length is $150\mu m$); this is because the failure of the

Sigma-Delta control leads to ineffective feedback. As shown in Figure 3.13, the system generates nearly a fixed-density output bitstream, which does not reflect the input signal at all. It is worth reiterating that the conventional model fails to reflect this fact and appears to show a correct operation where in fact the control breaks down.

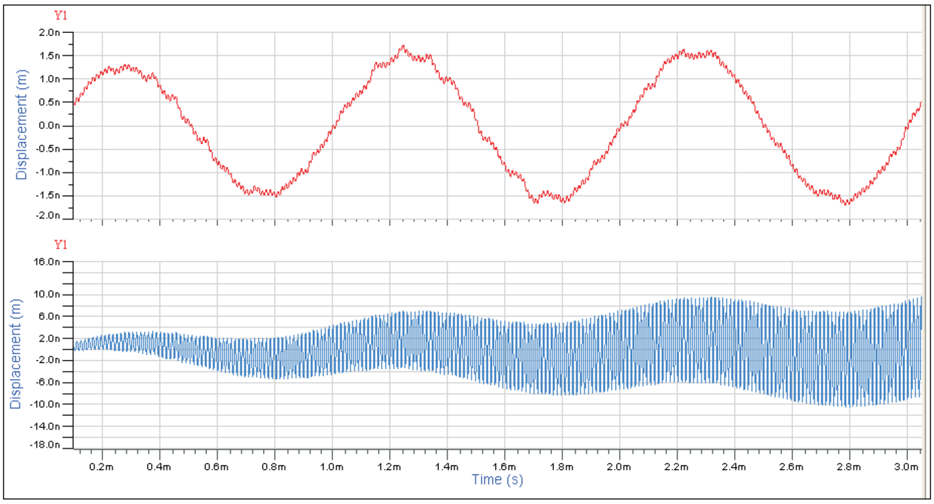


Figure 3.12: Simulation results of the improved VHDL-AMS model of the second-order electromechanical Sigma-Delta modulator ($L_{sf} = 190\mu m$) in response to a sinusoidal acceleration with 1g amplitude and 1kHz frequency. Top trace: displacements of the proof mass; Bottom trace: displacements of the free end of the sense finger

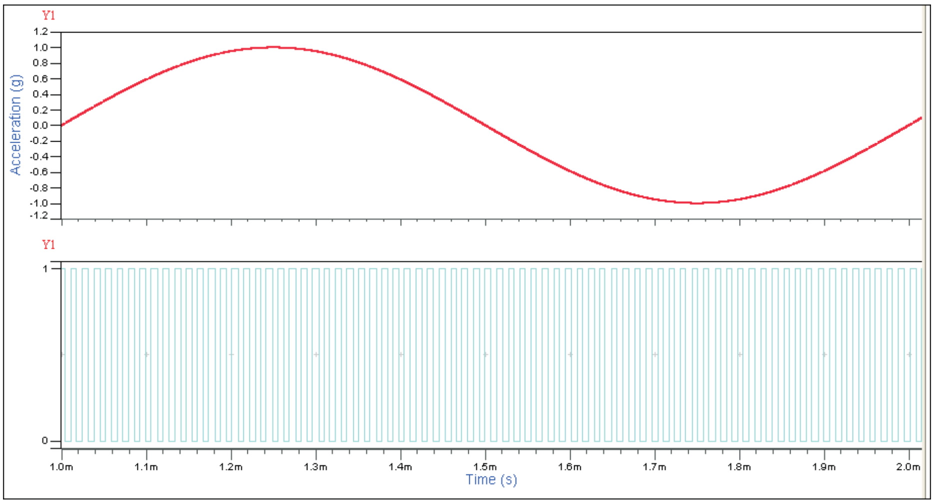


Figure 3.13: Simulation results of the improved VHDL-AMS model of the second-order electromechanical Sigma-Delta modulator ($L_{sf} = 190\mu m$) in response to a sinusoidal acceleration with 1g amplitude and 1kHz frequency. Top trace: input acceleration; Bottom trace: output bitstream

3.2.4 SystemC-A implementation of the distributed mechanical sensing element

Listing 3.6 shows the SystemC-A code of the proposed distributed mechanical sensing element model. The sense finger is also divided into ten sections, whose deflections are defined by quantities y1-y10 (Lines 7-9). To implement the distributed model in SystemC-A, each second-order ODE (Equation 3.31-3.35) should be converted to two first-order ODEs (Lines 39-70).

```
1  Sensing_Element::Sensing_Element(char nameC[5],sc_signal<double>*ain,
2         sc_signal<double>*Vf1,sc_signal<double>*Vf2,sc_signal<double>*pos):
3     component(nameC,0, 0, 0){
4         pos_sig=pos;         ain_sig=Input;
6         Vf1_sig =Vf1;        Vf2_sig =Vf2;
7         y1 = new Quantity("y1");
8               ...
9         y10 = new Quantity("y10");
10        z1 = new Quantity("z1");
11             ...
12        z10 = new Quantity("z10");
13  /*---Design parameters of mechanical sensing elemet---*/
14       Wpm=110.0e-6; Lpm=400.0e-6; T=3.0e-6; Ls=150.0e-6;
15             ...
16        }
17 void Sensing_Element::Build(void){
18       pos_sig->write((Y1n+Y2n+Y3n+Y4n+Y5n+Y6n+Y7n+Y8n+Y9n+Y10n)/10);
19       ain=ain_sig->read();
20       Vf1=Vf1_sig->read(); Vf2=Vf2_sig->read();
21       Y1n=X(y1); Z1n=X(z1);
22             ...
23       Y10n=X(y10); Z10n=X(z10);
24       Y1dtn=Xdot(y1); Z1dtn=Xdot(z1);
25             ...
26       Y10dtn=Xdot(y10); Z10dtn=Xdot(z10);
27     //Mass of sensing element//
28        M=2330.0*(Wpm*Lpm*T+(Ns+Nf)*Lsf*Wsf*T);
29     //Mechanical spring//
30        Kmechanical=4.0*12.0*Ws*Ws*Ws*T/(Ls*Ls*Ls);
31     //Electrosttic spring//
32        Ke=-1.0*Ns*(2.0*8.85e-12*Lsf*T*Vm*Vm)/(G*G*G);
33     //Effective spring constant//
34        K=Kmechanical+Ke;
```

```
   //Damping coefficient//
      D=14.4*(Ns+Nf)*1.85e-5*T*Lsf*Lsf*Lsf/(G*G*G);
   //Feedback force//
      Ff=0.5*Nf*8.85e-12*Lff*T*(Vf1*Vf1-Vf2*Vf2)/(G2*G2);
//------Each 2nd order ODE is divided into two 1st order ODEs------//
//------let y1'=z1; y2'=z2 ... y10'=z10 ---------------------------//
  //--The root segment displacement--//
     Equation(y1,-Y1dtn + Z1n);
     Equation(z1,-Z1dtn +(M*ain+Ff)/M-(D/M)*Y2n-(K/M)*Y1n);
  //--The second segment motion equation--//
     Equation(y2,-Y2dtn + Z2n);
     Equation(z2,-Z2dtn -(CD/(ro*S*dx*dx*dx*dx))*(Z4n-4*Z3n+6*Z2n-3*Z1n)
              -(E*I/(ro*S*dx*dx*dx*dx))*(Y4n-4*Y3n+6*Y2n-3*Y1n)+
              ((ep*Lsf*dx)/(2*dx*ro*S))*(Vm*Vm/((G-Y2n)*(G-Y2n))
              -Vm*Vm/((G+Y3n)*(G+Y3n))));
  //--The third segment motion equation--//
     Equation(y3,-Y3dtn + Z3n);
     Equation(z3,-Z3dtn -(CD/(ro*S*dx*dx*dx*dx))*(Z5n-4*Z4n+6*Z2n-4*Z1n
              +Z1n)-(E*I/(ro*S*dx*dx*dx*dx))*(Y5n-4*Y4n+6*Y3n-4*Y2n+Y1n)
              +((ep*Lsf*dx)/(2*dx*ro*S))*(Vm*Vm/((G-Y3n)*(G-Y3n))
              -Vm*Vm/((G+Y5n)*(G+Y5n))));

                 ...

   //--The ninth segment motion equation--//
     Equation(y9,-Y9dtn + Z9n);
     Equation(z9,-Z9dtn -(CD/(ro*S*dx*dx*dx*dx))*(-2*Z10n+5*Z9n-4*Z8n+Z7n)
              -(E*I/(ro*S*dx*dx*dx*dx))*(-2*Y10n+5*Y9n-4*Y8n+Y7n)
              +((ep*Lsf*dx)/(2*dx*ro*S))*(Vm*Vm/((G-Y7n)*(G-Y7n))
              -Vm*Vm/((G+Y9n)*(G+Y9n))));
   //--The free end segment motion equation--//
     Equation(y10,-Y9dtn + Z10n);
     Equation(z10,-YZ10dtn -(CD/(ro*S*dx*dx*dx))*(Z10n-2*Z9n+Z8n)
              -(E*I/(ro*S*dx*dx*dx*dx))*(Y10n-2*Y9n+Y8n)
              +((ep*Lsd*dx)/(2*dx*ro*S))*(Vm*Vm/((G-Y10n)*(G-Y10n))
              -Vm*Vm/((G+Y9n)*(G+Y9n))));
 }
```

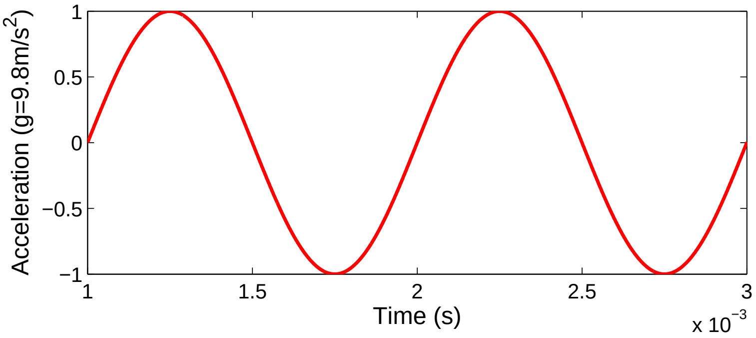
Listing 3.6: SystemC-A implementation of distributed mechanical sensing element

Figure 3.14 and Figure 3.15 show the time-domain simulation results of the improved SystemC-A model of the second-order electromechanical Sigma-Delta modulator with different sense finger lengths, i.e. $150\mu m$ and $190\mu m$ respectively. The simulation results are all consistent with those of VHDL-AMS model.

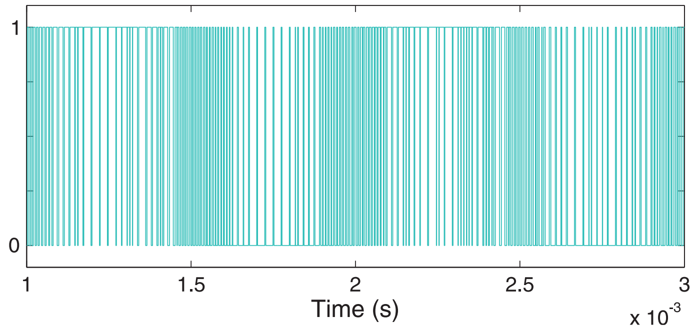
Figure 3.16 shows the PSD of the electromechanical Sigma-Delta modulator output bitstream when the finger length is $150\mu m$. The Sigma-Delta control loop works correctly, while the noise in the signal band is dynamically decreased by oversampling and noise shaping techniques. The peak at about 70kHz indicates the unity-gain frequency, while the peak at about 123kHz represents the first resonant frequency of the sense finger. The finger dynamics do not affect this implementation too much because the unity gain frequency is much below the finger resonance. However, if the sense finger length increases, the finger resonance frequencies move towards the unity frequency and affect the performance of the Sigma-Delta control. Figure 3.17 shows the SNR of the second-order electromechanical Sigma-Delta modulator with varying sense finger length. In this design, failures of the Sigma-Delta control loop are captured when the sense finger length exceeds $190\mu m$. The PSD of the output bitstream of the system with $190\mu m$ finger length is shown by the trace in Figure 3.18. The oversampling and noise shaping techniques of the Sigma-Delta modulator fail to drive the noise to the higher band. Presented results provide further evidence that the classical lumped model is inadequate in the design of MEMS accelerometers with a Sigma-Delta force-feedback control scheme, because it does not capture failures of the control loop caused by the mechanical motions of the sense finger.

3.2.5 Minimum number of discrete sections

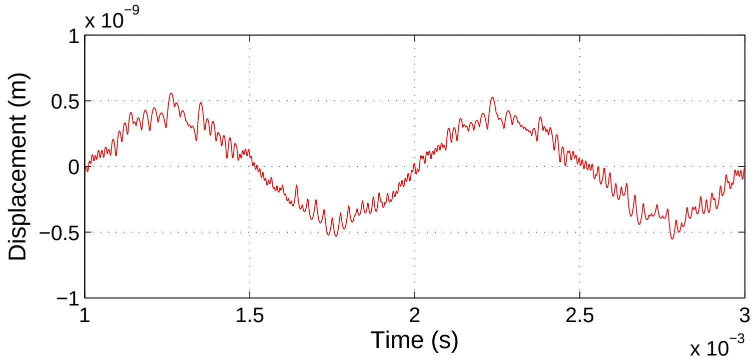
The number of discrete sections is a critical parameter which determines the accuracy of the behaviour of the distributed system. A series of simulation experiments, using the SystemVision simulator from Mentor Graphics, were carried out to establish the minimum number of discrete sections with which the distributed mechanical sensing element model is accurate enough to reflect the correct behaviour of the system. Typical



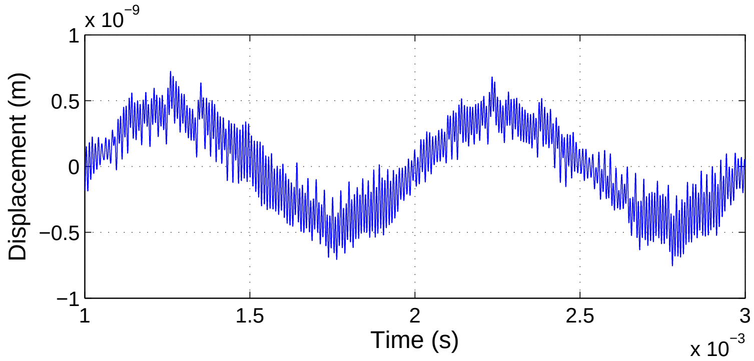
(a) Input acceleration



(b) Output bitstream

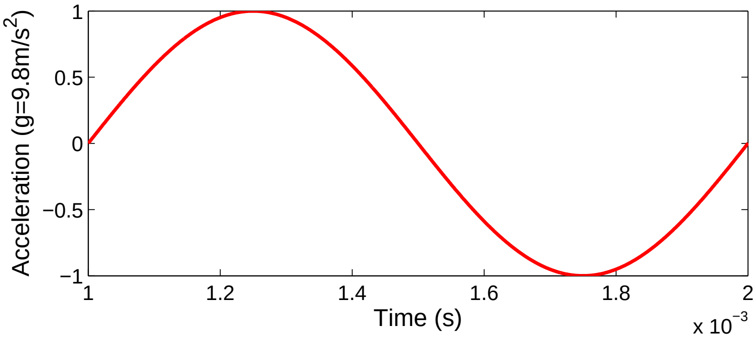


(c) Displacements of the proof mass

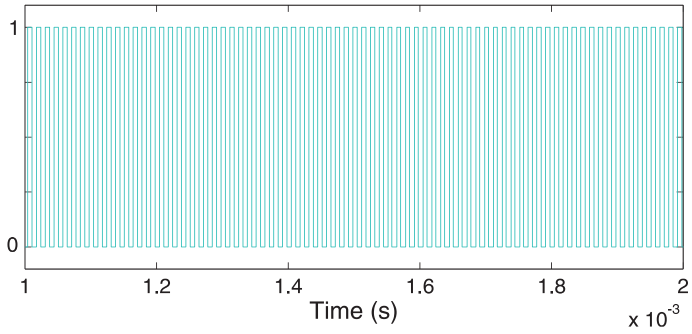


(d) Displacements of the free end of the sense finger

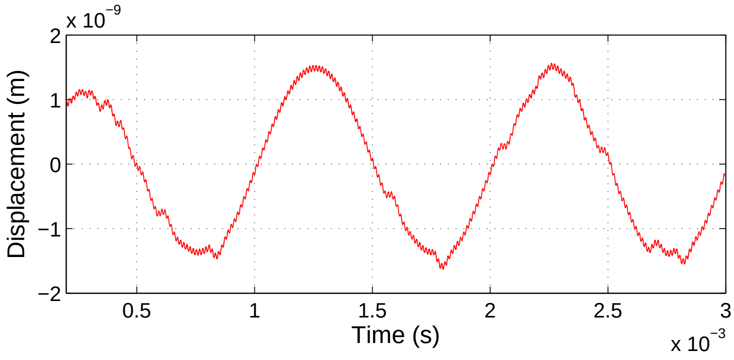
Figure 3.14: Simulation results of the improved SystemC-A model of the second-order electromechanical Sigma-Delta modulator in response to a sinusoidal acceleration with 1g amplitude and 1kHz frequency (Sense finger length= $150\mu m$)



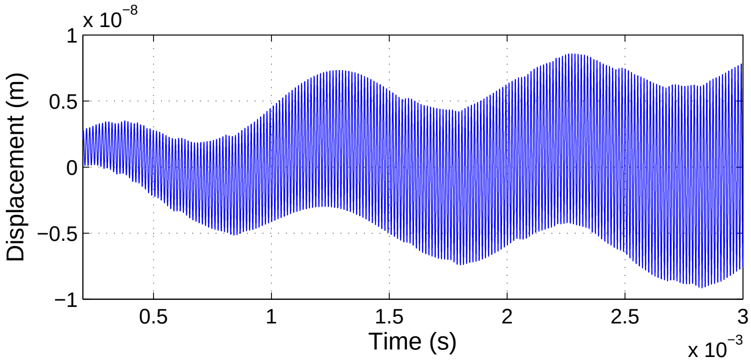
(a) Input acceleration



(b) Output bitstream



(c) Displacements of the proof mass



(d) Displacements of the free end of the sense finger

FIGURE 3.15: Simulation results of the improved SystemC-A model of the second-order electromechanical Sigma-Delta modulator in response to a sinusoidal acceleration with 1g amplitude and 1kHz frequency(Sense finger length= $190\mu m$)

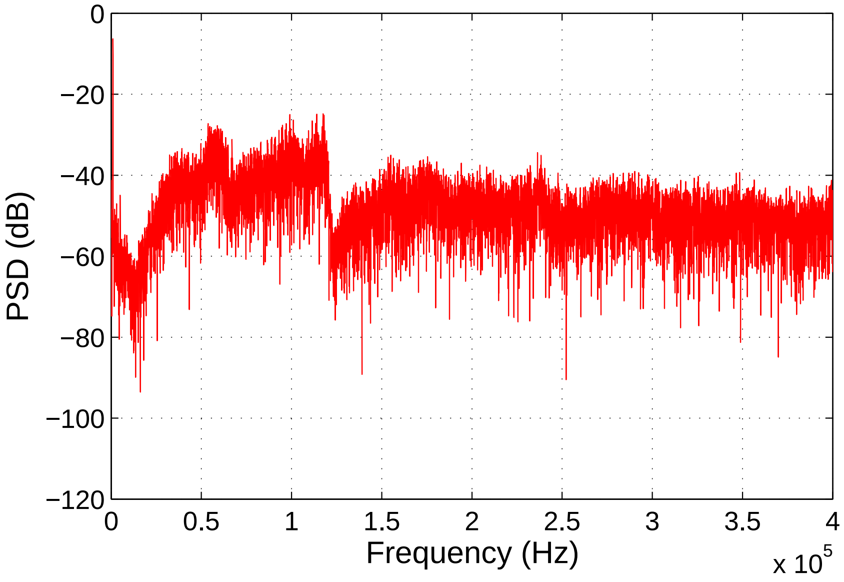


Figure 3.16: Power spectral density of the output bitstream derived from the improved electromechanical Sigma-Delta modulator model in response to a sinusoidal acceleration with 1g amplitude and 1kHz(sense finger length $L_{sf} = 150\mu m$). PSD is obtained by 65536 point FFT of the output bitstream using hanning window

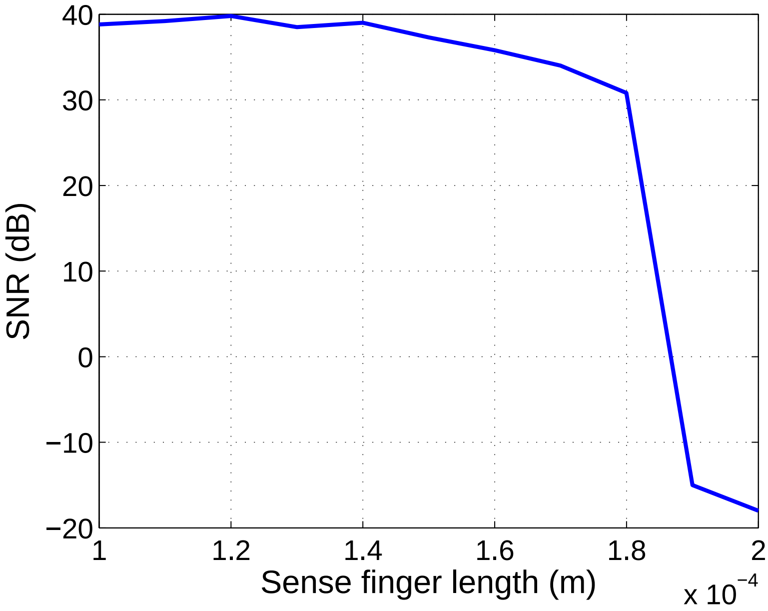


Figure 3.17: SNR of the second-order electromechanical Sigma-Delta modulator VS sense finger length. The failure of the Sigma-Delta control is captured when finger length over $190\mu m$

jolt and step input acceleration signals are used in these simulations. Such excitations are common in typical MEMS accelerometer applications in automobile safety systems.

Failures of the Sigma-Delta control loop can be detected by observing the duty ratio of

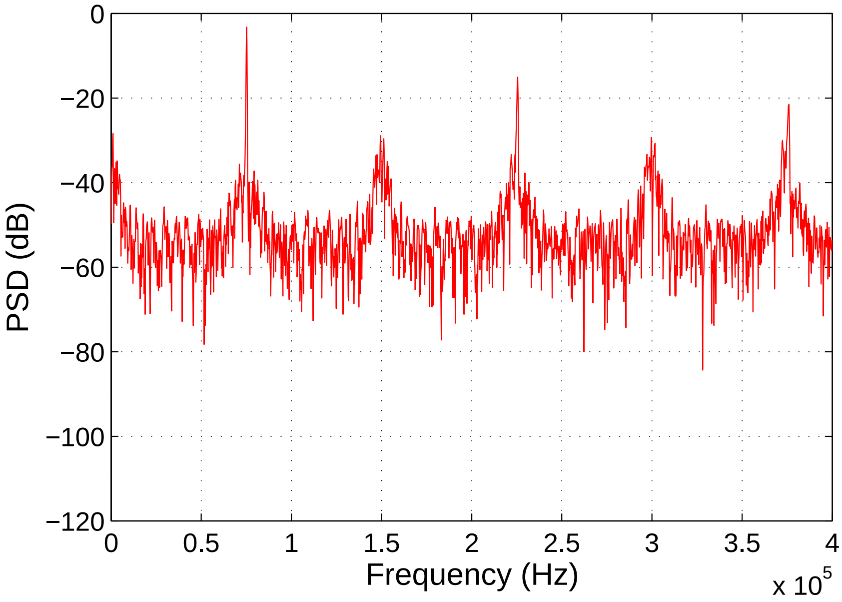


FIGURE 3.18: Power spectral density of the output bitstream derived from the improved electromechanical Sigma-Delta modulator model in response to a sinusoidal acceleration with 1g amplitude and 1kHz(sense finger length $L_{sf} = 190\mu m$). PSD is obtained by 65536 point FFT of the output bitstream using hanning window

Length	Number of Sections			
	4	5	10	20
50um	87.15%	89.06%	88.39%	88.94%
100um	87.80%	89.03%	89.91%	88.34%
150um	90.10%	89.42%	88.85%	88.44%
160um	87.59%	84.64%	83.91%	82.77%
180um	85.19%	75.33%	73.71%	70.18%
190um	82.32%	***55.10%***	***54.60%***	***53.57 %***
195um	69.33%	***53.32%***	***54.16%***	***52.45%***

TABLE 3.3: Output bitstream duty duty ratio (1g jolt input acceleration)

the output bitstream which describes the proportion of 1s in a period of time. Namely, the duty ratio around 50%, i.e. a fixed-density output bitstream, represents the failure of the Sigma-Delta control because the input signal cannot be captured. The time periods selected in the jolt and step inputs experiments are 0.7ms to 0.8ms (around the peak of jolt input) and 1ms to 1.1ms respectively. The duty ratios obtained for the jolt and step inputs for the conventional model are 89.79% and 80.35% respectively almost regardless of finger length. However, as shown in Tables 3.3 and 3.4, in reality the duty ratios calculated from the proposed accurate model reduce to nearly 50% (values in bold font) when the sense finger length is above $190\mu m$. If the length is under this value,

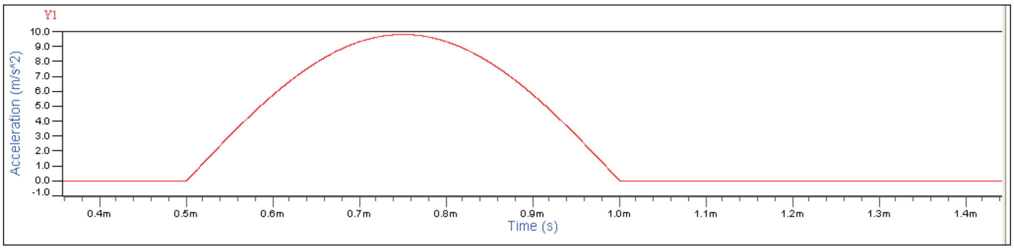
Length	Number of Sections			
	4	5	10	20
50um	74.30%	79.78%	80.05%	79.84%
100um	75.34%	80.45%	81.60%	81.48%
150um	75.19%	77.98%	78.24%	80.06%
160um	73.24%	68.49%	69.20%	66.67%
180um	72.63%	60.21%	59.80%	59.00%
190um	67.42%	***53.67%***	***53.28%***	***53.62%***
195um	60.10%	***51.64%***	***51.76%***	***51.62%***

Table 3.4: Output bitstream duty ratio (1g step input acceleration)

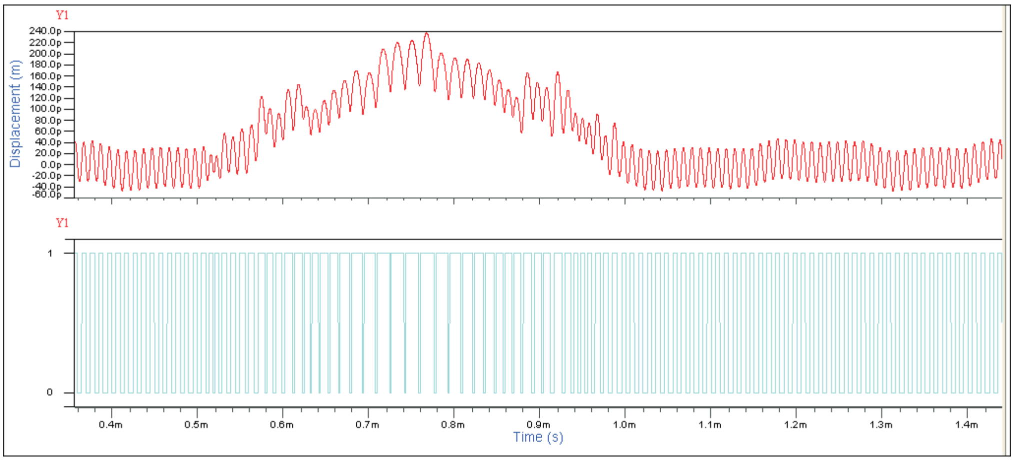
sense fingers move with the proof mass and experience minor bending. Serious bending occurs when the length is over 190μm leading to a failure of control loop with a nearly fixed-density output bitstream.

The number of discrete sections of sense finger determines the accuracy of the proposed distributed model. As shown in Table 3.3 and Table 3.4, if each sense finger is divided into 5 sections or more, correct behaviour is captured accurately, while 4 sections are not adequate as the control loop still appears to work for longer fingers (duty ratios are 82.32% and 67.42% for the jolt and step stimulus when finger length increases to 190μm).

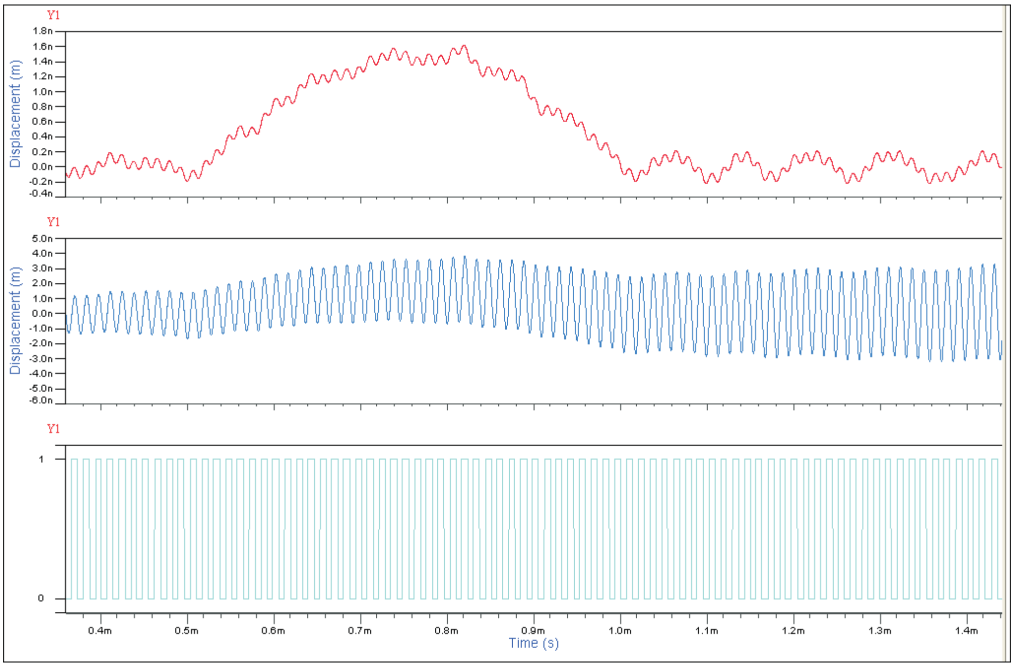
Simulation results for jolt and step input signals are shown in Figures 3.19 and 3.20 respectively demonstrating again that the break down of the control loop is correctly captured by the proposed model (Sense finger is divided into 5 sections). The maximum deflection of the free end of the sense finger is around 4nm in jolt and step input signal experiments with the finger length is 190μm. The failure of the Sigma-Delta control loop also causes large deflection of proof mass (about 2nm), as the control is ineffective and does not provide adequate electrostatic feedback to pull the proof mass back into the original position. Figures 3.19 and 3.20 also indicate that the conventional model fails to reflect the true behaviour with only 0.2nm displacement of the proof mass.



(a) Input acceleration

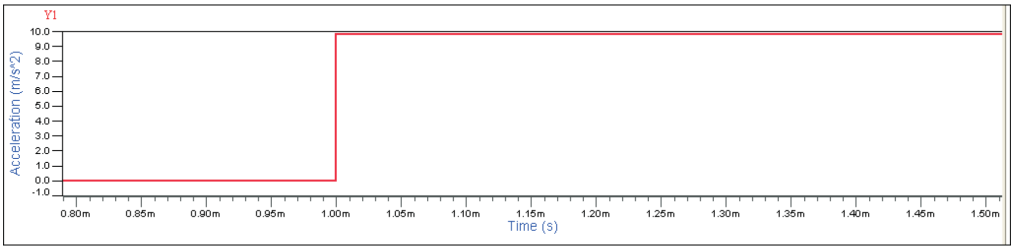


(b) Simulation results of the conventional model. Top trace: displacements of proof mass; Bottom trace: output bitstream

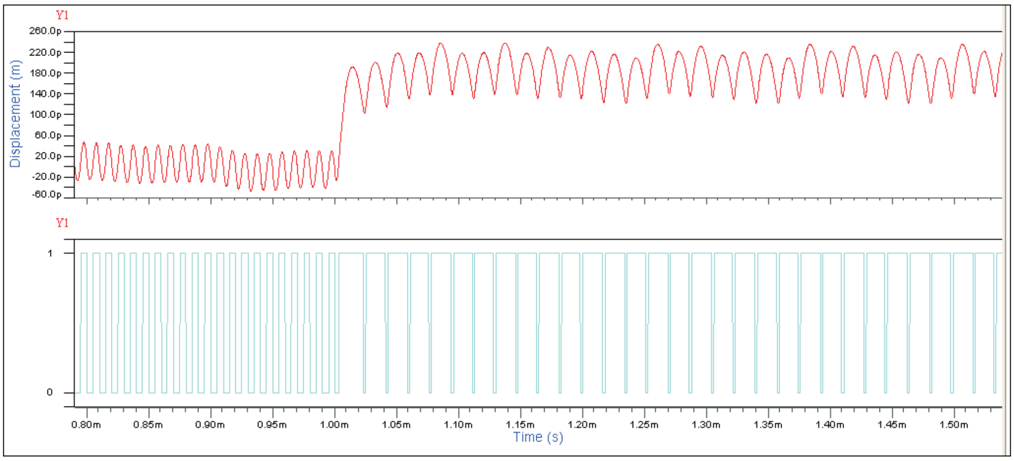


(c) Simulation results of the improved model. Top trace: displacements of proof mass; Middle trace: free end of sense finger; Bottom trace: output bitstream

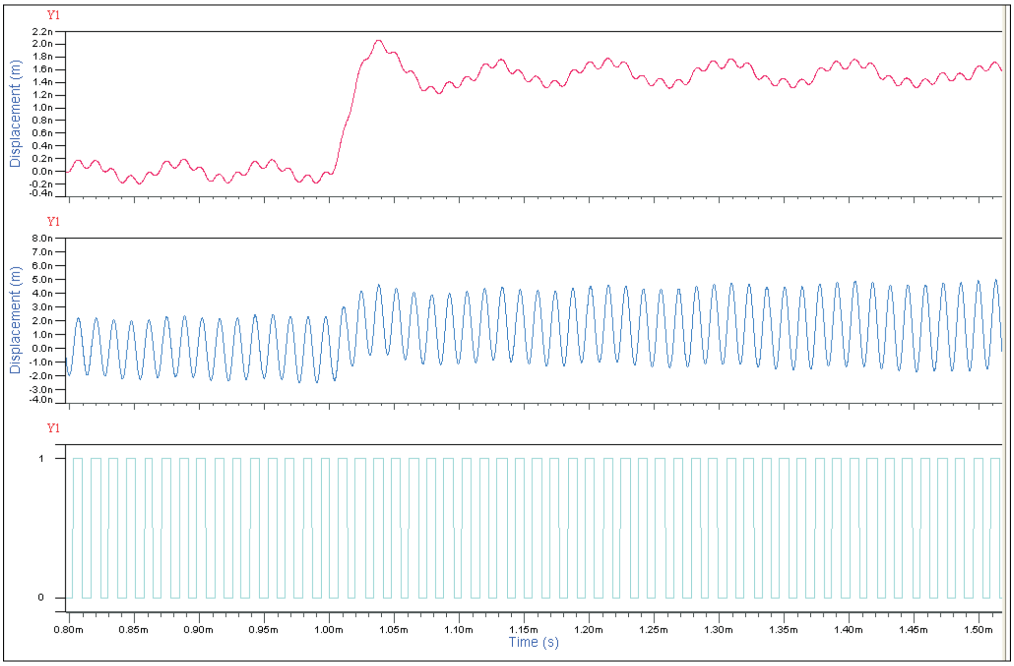
FIGURE 3.19: Simulation results of the electromechanical Sigma-Delta modulator with 1g jolt input acceleration. a) Input acceleration; b) Simulation results of the conventional model; c) Simulation results of the improved model



(a) Input acceleration



(b) Simulation results of the conventional model. Top trace: displacements of proof mass; Bottom trace: output bitstream



(c) Simulation results of the improved model. Top trace: displacements of proof mass; Middle trace: free end of sense finger; Bottom trace: output bitstream

FIGURE 3.20: Simulation results of the electromechanical Sigma-Delta modulator with 1g step input acceleration. a) Input acceleration; b) Simulation results of the conventional model; c) Simulation results of the improved model

3.3 Comparison between VHDL-AMS and SystemC-A

The second-order electromechanical Sigma-Delta modulator models, with both conventional lumped and proposed distributed mechanical sensing element, were implemented in VHDL-AMS and SystemC-A in this chapter. The SystemC-A simulator used in this research is based on an efficient experimental analogue solver developed by H. Al-Junaid [81]. The VHDL-AMS models were simulated using a commercial simulator, SystemVision from Mentor Graphics [9]. Both simulators produced highly comparable results as illustrated in Figures 3.10-3.15. However, SystemC-A provides additional advantages of high simulation speed and flexible data manipulation.

In C++ based modelling and simulation environments such as SystemC-A, users can easily export results into text files and view them or undertake more analysis. This is a great advantage over the SystemVision VHDL-AMS simulator which does not support text I/O operations. The lack of text I/O in SystemVision leads to difficulties in postprocessing simulation data.

The second-order electromechanical Sigma-Delta modulator models, with both conventional lumped and proposed distributed mechanical sensing element, has been used to compare the simulation speed of SystemC-A with that of SystemVision. Simulations were carried out using the same simulation time (40ms) and a fixed time step (50ns) in both simulators. The relevant statistics are shown in Table 3.5. The difference in the simulation speed represents a factor of almost two times in favour of SystemC-A.

	Conventional model	***Distributed model***
Time step	$50ns$	$50ns$
Simulation Time	$40ms$	$40ms$
CPU time(SystemC-A)	$8.2s$	$14.8s$
CPU time(SystemVision)	$16.9s$	$25.8s$

Table 3.5: Electromechanical Sigma-Delta modulator simulation statistics

3.4 Concluding remarks

In this chapter, a capacitive MEMS accelerometer with Sigma-Delta control is modelled in VHDL-AMS and SystemC-A. Firstly, the mechanical sensing element of such accelerometer is modelled by the conventional approach, where a second-order ODE is commonly used. However, the conventional model is not accurate enough to capture the sense finger resonances which may seriously affect the performance of the Sigma-Delta feedback control.

Secondly, a distributed mechanical sensing element model, which includes sense finger dynamics, is presented. Simulation results show that the distributed model correctly reflects the way in which finger dynamics affect the performance of the control loop. In contrast, the conventional model does not capture the well-known failure of the control loop when sense fingers bend significantly or resonate.

The distributed model of the sensing element is developed by spatial discretisation of the PDE to obtain a series of ODEs using the FDA approach. The number of discrete sections is a critical number which determines the accuracy of the model. A series of experiments are conducted to find the minimum number of sections in this model. The analysis provides modelling guidelines to facilitate correct trade-offs when designing of MEMS accelerometer in the Sigma-Delta control loop.

In the next chapter, a holistic synthesis approach to designing MEMS sensors with Sigma-Delta control system is presented. The proposed genetic-based synthesis technique is implemented in SystemC-A and named SystemC-AGNES. SystemC-A is chosen because of its high simulation speed and flexible manipulation of data.

Chapter 4

A holistic approach to automated synthesis and optimisation of mixed-technology digital MEMS sensors

This chapter presents a novel, holistic methodology for automated optimal layout synthesis of MEMS systems embedded in electronic control circuitry from user-defined high-level performance specifications and design constraints. The proposed approach is based on simulation-based optimisation where the genetic-based synthesis of both mechanical layouts and associated electronic control loops is coupled with calculations of optimal design parameters. The proposed genetic-based synthesis technique has been implemented in SystemC-A and named SystemC-AGNES. It integrates a MEMS primitive library, an electronic control loop primitive library, an efficient fast MEMS simulation engine implemented in SystemC-A and an evolutionary computation method (GA). The underlying MEMS models in the MEMS primitive library include distributed mechanical dynamics described by partial differential equations to enable accurate performance prediction of critical mechanical components.

SystemC-AGNES is applicable to a wide class of MEMS sensors with electronic control. We demonstrate its operation using a surface micromachined capacitive MEMS accelerometer in a Sigma-Delta control loop as a case study. The capacitive digital MEMS accelerometers are notoriously difficult to design using traditional methods because the mechanical element forms an integral part of the Sigma-Delta control loop. This feature makes a separation of the two technology domains in the design process very effortful.

This chapter is organised as follows. Section 4.1 focuses on the layout synthesis of mechanical components while the associated Sigma-Delta control is fixed to form a second-order structure. As discussed in Section 2.3.6, the performance of the electromechanical Sigma-Delta modulator can be further improved by inserting an electronic loop filter in the Sigma-Delta control scheme to form a higher-order topology. The full synthesis methodology, which combines both the mechanical layout synthesis and the associated high-order Sigma-Delta control configuration synthesis, is then outlined in Section 4.2.

The design of the associated Sigma-Delta control for MEMS sensors is inspired from that of electronic Sigma-Delta modulators; thus, the synthesis approach developed in Section 4.2 can also be used to synthesise general Sigma-Delta modulators for applications other than digital MEMS sensors. This is illustrated in Section 4.3. Finally, Section 4.4 draws conclusions from this work.

4.1 Layout synthesis of MEMS component with distributed mechanical dynamics in SystemC-AGNES

The proposed automated synthesis approach explores the design according to user defined specifications and optimises the structural parameters of the mechanical MEMS elements and the associated electronic control loop parameters. The automated optimal synthesis flow is outlined in Algorithm 4.1 and also shown in Figure 4.1.

Algorithm 4.1: Genetic-based synthesis algorithm

```
Input: Design constraints: $SNR$, $sensitivity$, $area\ of\ sensing\ element$,
       Stopping criterion: $No\_Generation$,
       Genetic Algorithm(GA) setting: $Population\_Size$, $P\_Xover$......
Output: Synthesised optimal layout of mechanical sensing element and associated
        Sigma-Delta control system: Optimal_Solution
begin
    Set $t \longleftarrow 1$; $t$ is the number of generation;
    for $i = 1 : +1 : N = Population\_Size$ do //Initialisation and encoding
        Population(i): Randomly generate layout of mechanical sensing element and
        associated Sigma-Delta control system by combing the primitive components in
        stored MEMS and Electronic control loop SystemC-A libraries according to user
        defined constraints to form the first generation of the Genetic-based algorithm.;
        Model(i): SystemC-A model generation according to the initial topology.;
    end
    //Genetic-based synthesis module;
    repeat
        for $i = 1 : +1 : N = Population\_Size$ do //Evaluation
            Simulation(Model(i));
            Evaluate(Model(i)): Evaluate the performance of the synthesised design;
        end
        Update(Optimal_Solution): store solution with current best performance;
        Selection(): Select solutions with better performance as parents for crossover
        operation, the number of selected design is defined by users;
        Crossover(): Randomly choose pairs of solutions from selected designs after
        selection operation as parents to generate offspring until new generation is
        generated;
        Mutation(): Each chromosome in new generation has probability to mutate;
        for $i = 1 : +1 : N = Population\_Size$ do
            Model(i): SystemC-A model generation according to the topology.;
        end
        t=t+1;
    until $t = No\_Generation$;
    return Optimal_Solution;
end
```

After specifying the design objectives and constraints, such as the die area of the sensing element and feedback voltage in the electronic control loop, available components in the MEMS primitive library and the electronic control loop primitive library are combined automatically to form a valid initial design set. This set of initial designs is loaded into the synthesis module as the first generation. The synthesis module uses a genetic algorithm to create new MEMS structures and optimises their parameters for

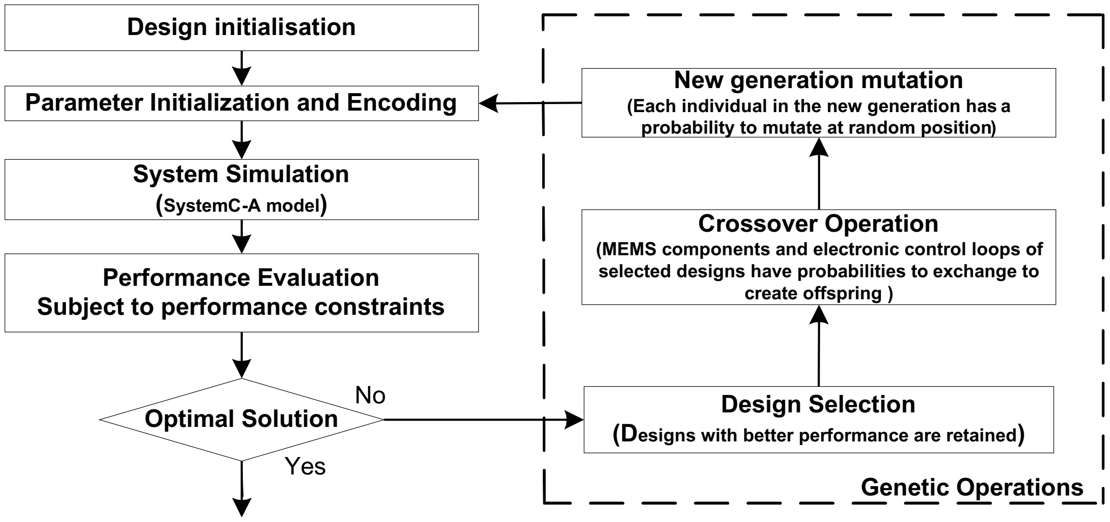


Figure 4.1: Automated synthesis flow in SystemC-AGNES

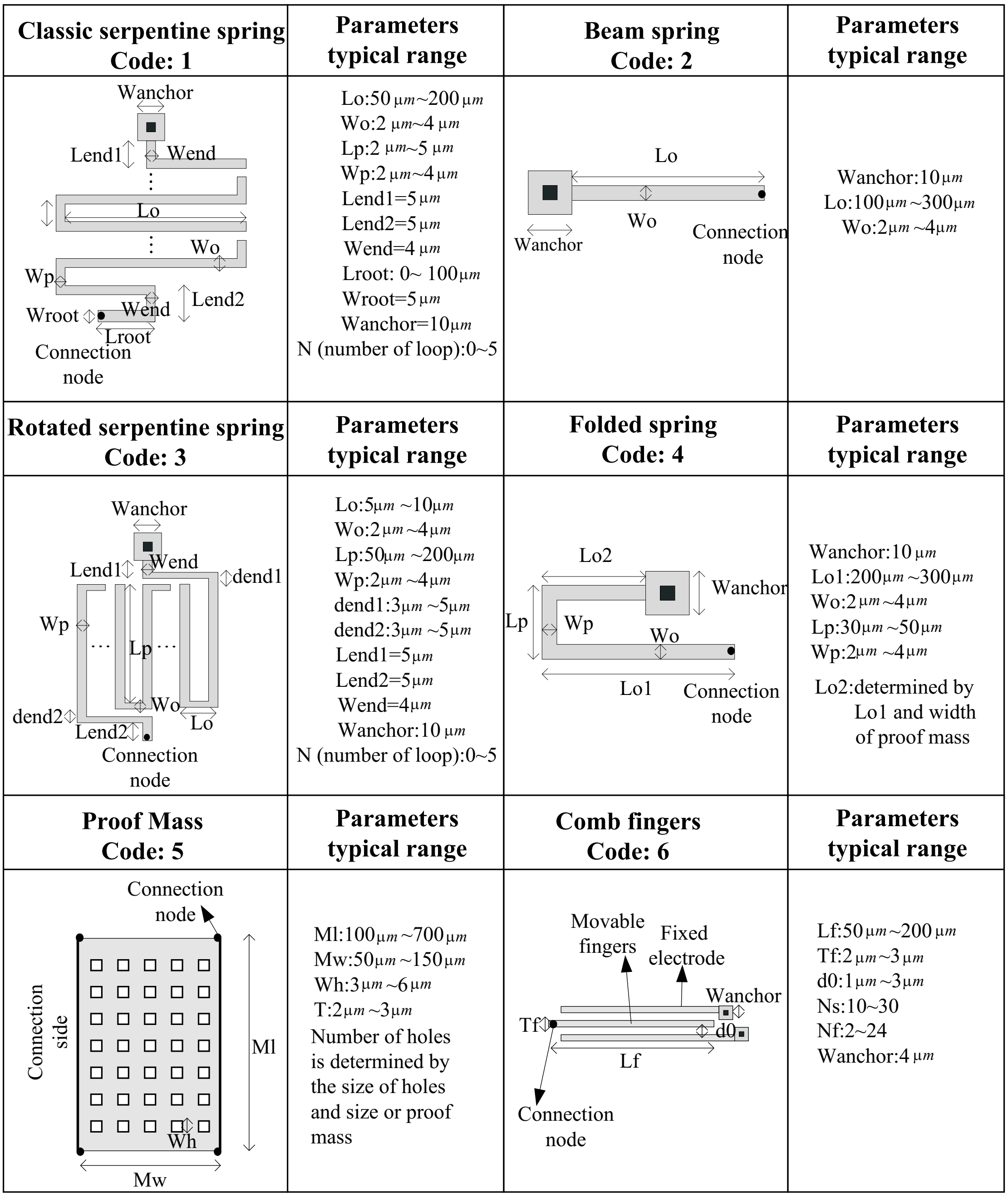
best performance. Our approach integrates mixed-technology models into a single simulation engine (SystemC-A) which could be easily invoked from various optimisation loops. Unlike traditional MEMS design tool sets, this approach avoids a generation of macromodels in order to realise co-design and co-simulation.

4.1.1 Synthesis initialisation

There are two libraries, the MEMS primitive library and the electronic control loop primitive library, each containing typical components that are widely used in practical MEMS designs. Every member in the libraries is a data structure record which includes its type code, geometrical parameters for MEMS primitives, system-level design parameters for electronic primitives and constraints. This section focuses on layout synthesis of the mechanical component while the configuration of the associated electronic control loop is fixed.

4.1.1.1 MEMS primitive library

The mechanical part of a surface micromachined capacitive MEMS accelerometer is composed of a proof mass, springs and comb fingers. In the capacitive structure, the proof mass is suspended by springs and is equipped with sense and force comb fingers which are placed between fixed fingers to form a capacitive bridge. The sense fingers move with the proof mass resulting in a differential imbalance in capacitance which is measured. The electrostatic force acting on the force fingers is used as the feedback signal to pull the proof mass in the desired direction. The available mechanical components in the MEMS primitive library (Figure 4.2) are discussed below.



Classic serpentine spring Code: 1	**Parameters typical range**	**Beam spring Code: 2**	**Parameters typical range**
	Lo:50 µm~200 µm Wo:2 µm~4 µm Lp:2 µm~5 µm Wp:2 µm~4 µm Lend1=5 µm Lend2=5 µm Wend=4 µm Lroot: 0~ 100µm Wroot=5 µm Wanchor=10µm N (number of loop):0~5		Wanchor:10µm Lo:100µm ~300µm Wo:2µm ~4µm
Rotated serpentine spring Code: 3	**Parameters typical range**	**Folded spring Code: 4**	**Parameters typical range**
	Lo:5µm ~10µm Wo:2µm ~4µm Lp:50µm ~200µm Wp:2µm ~4µm dend1:3µm ~5µm dend2:3µm ~5µm Lend1=5µm Lend2=5µm Wend=4µm Wanchor:10 µm N (number of loop):0~5		Wanchor:10 µm Lo1:200µm ~300µm Wo:2 µm ~4µm Lp:30µm ~50µm Wp:2µm ~4µm Lo2:determined by Lo1 and width of proof mass
Proof Mass Code: 5	**Parameters typical range**	**Comb fingers Code: 6**	**Parameters typical range**
	Ml:100µm ~700 µm Mw:50µm ~150µm Wh:3µm ~6 µm T:2µm ~3µm Number of holes is determined by the size of holes and size or proof mass		Lf:50µm ~200 µm Tf:2 µm ~3 µm d0:1 µm ~3µm Ns:10~30 Nf:2~24 Wanchor:4 µm

Figure 4.2: MEMS primitive library

1. Springs: Four typical springs are available in the MEMS primitive library for this case study: classic serpentine spring, rotated serpentine spring, folded spring and cantilever beam spring. These springs are widely used in various MEMS mechanical structures [140]. The layout and geometrical parameters with constraints are shown in Figure 4.2.

2. Proof mass: The proof mass contains square etch holes for release. The number of these holes is determined by the size of the proof mass and the size of holes. There are 4 connecting nodes and 2 connecting sides on the proof mass, 4 connecting nodes are used to connect springs and 2 connecting sides are used for comb sense and force fingers connection.

3. Comb fingers: The sensing element dynamics in the sense-direction is normally modelled to reflect only one resonant mode by a lumped mass, spring, and damper, which is represented by a simple second-order ordinary differential equation (ODE).

In reality, the sense comb fingers in a capacitive structure are distributed elements with many resonant modes. As their dynamics affect the performance of a Sigma-Delta control system, the motion of the sense finger should be distributed, for example using the following PDE [139]:

$$\rho S\frac{\partial^2 y(x,t)}{\partial t^2} + C_D I\frac{\partial^5 y(x,t)}{\partial x^4 \partial t} + EI\frac{\partial^4 y(x,t)}{\partial x^4} = F_e(x,t) \tag{4.1}$$

where $y(x,t)$, a function of time and position, represents the deflection of the beam. E, I, C_D, ρ, S are all physical properties of the beam. ρ is the material density, S is the cross sectional area, E represents the Young's modulus which defines a material's shearing strength, I is the second moment of area, EI is usually regarded as the flexural stiffness, C_D is the internal damping modulus, $F_e(x,t)$ is the distributed electrostatic force along the beam.

To implementation the PDE in SystemC-A, Finite Difference Approximation (FDA) approach is applied to convert the PDE to a series of ODAEs as illustrated in the last chapter.

4.1.1.2 Electronic control loop

High-performance MEMS sensors exploit the advantages of closed-loop control strategy to increase the dynamic range, linearity, and bandwidth of sensor. In particular, digital Sigma-Delta modulators for closed-loop feedback control schemes, whose output is digital in the form of pulse-density-modulated bitstream, have become very attractive. A conventional second-order electromechanical Sigma-Delta modulator is shown in Figure 4.3. In this configuration, the mechanical sensing element is used as a loop filter to form the second order electromechanical Sigma-Delta modulator. A detailed illustration of this system can be found in Section 3.1. However, the equivalent DC gain of the mechanical integrator in the second-order electromechanical Sigma-Delta modulator is relatively low, and this leads to a poor signal-to-noise ratio (SNR). To improve the SNR, the mechanical element can be cascaded with additional electronic integrators to form high-order topologies. The example of automated synthesis discussed in this section focuses on the synthesis of the MEMS mechanical layout, and the electromechanical Sigma-Delta modulator is fixed and of second-order. Full synthesis, which includes both MEMS layout and electronic control loop, is presented in Section 4.2.

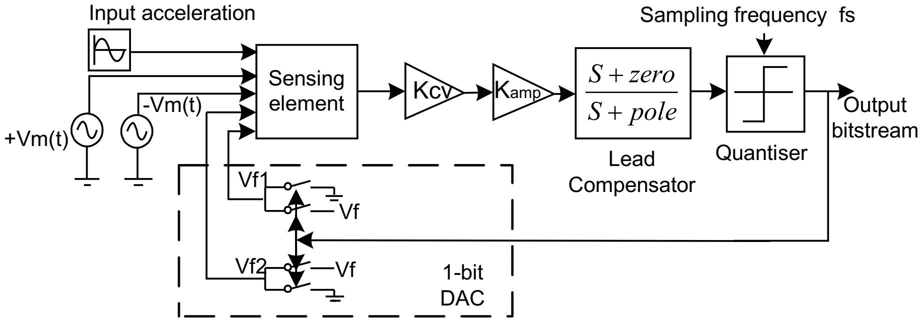


FIGURE 4.3: Second-order electromechanical Sigma-Delta modulator for MEMS sensors

4.1.1.3 Parameter initialisation and Encoding

The automated design process starts with a specification of the design objectives and constraints. Drawing from the MEMS primitive library and electronic control loop primitive library, a set of configurations is automatically selected (parents of first generation in GA) and loaded into the synthesis module. These feasible configurations not only contain MEMS mechanical layouts but also associated electronic control system topologies. Figure 4.4 and Table 4.1 show an example of a feasible configuration to illustrate the parameter initialisation and encoding phase. This MEMS accelerometer here contains 4 spring beams, 14 force fingers, 20 sense fingers and a proof mass with associated Sigma-Delta control loop. The component code of each component is shown in the Figure 4.2. Then the geometrical layout parameters of mechanical part and the associated system-level design parameters of electrical control systems are generated to describe the feasible layouts combining with the component code (Figure 4.4 and Table 4.1).

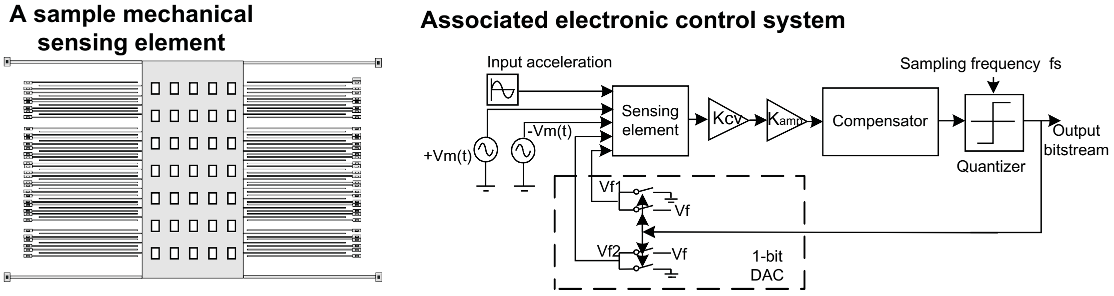


Figure 4.4: Example of parameter initialisation and encoding.

MEMS component library	***Code***	***Description***
Spring	2	Cantilever beam spring
Proof mass	5	Proof mass with etching holes
Comb drive	6	Sense and force fingers
Electronic Control loop library	***Code***	***Description***
Control system	1(fixed)	Sigma-Delta Control

Table 4.1: Representation of a population member in GA for the MEMS accelerometer example.

4.1.2 Genetic approach to synthesis

Genetic Algorithm (GA) has been selected for our case studies as it is a very popular and well-tested optimisation algorithm which has demonstrated good performance in a wide variety of complex global optimisation problems where modelling difficulties arise and there is no obvious way to find optimal solutions [141]. It has already been used for mechanical layout optimisation [142].

The optimisation problem is considered as a constrained optimisation as both of the design and performance parameters are bound by inequality constraints that must be met:

$$Minimize \quad or \quad maximize: \quad F(x) \tag{4.2}$$

Subject to:

$$x_n \in [V_{n\_low}, V_{n\_high}], n = 1, 2, 3... \tag{4.3}$$

where $F(x)$ is the fitness function to be optimised with design parameter vector x, x_n represents the nth design parameter, V_{n_low} and V_{n_high} are the lower and upper constraints of the nth design parameter.

Performance figures of the candidate designs are evaluated by the fitness function that rates the solutions according to their performance parameters. The fitness function is usually constructed in a weighted scalar error form:

$$F(x) = w\frac{R}{R'} \tag{4.4}$$

where w is the weight coefficient. R is a system performance parameter obtained from each simulation, and R' is the designer specified objective value.

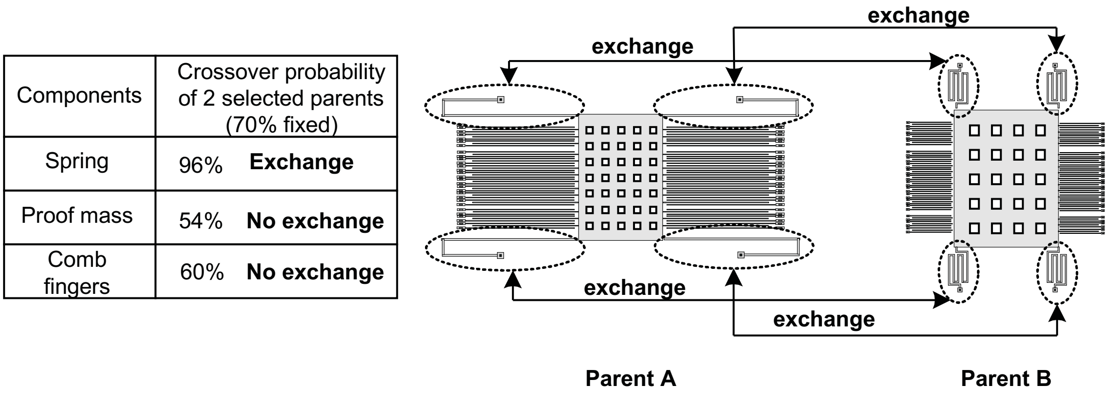
In the case study discussed below, a performance evaluation engine is added to the simulator to enable measurements of the power spectrum density (PSD) and signal-to-noise ratio (SNR), as the design objectives, through an FFT of the output bitstream.

Algorithm 4.2: Proposed crossover operation algorithm for mechanical sensing element

```
Input: Retained topologies after selection operation:$Topology[Parents\_Size]$,
       User defined generation size:$Generation\_size$,
       Parents selection probability:$P\_parent$,
       Gene crossover probability:$G\_xover$
Output: New generation: $Topology[Generation\_Size]$
begin
    $G\_size \longleftarrow Parents\_Size$; $G\_size$ is the size of current generation;
    repeat
        for $i = 1 : +1 : Parents\_Size$ do
            $x = rand()$; Generate random crossover probability for Topology[i];
            if $x > P\_parent$ then
                $Parent = Parent + 1$; $Parent$ is the number of the parents has already been selected;
                if $Parent\%2 = 0$ then Check whether two parents are selected
                    for $k = 1 : +1 : Component\_Size$ do
                        $y = rand()$; generate random crossover probability for gene[k];
                        if $y > C\_xover$ then
                            $xover(Topology[j].component[k], Topology[i].component[k])$,
                            Swap components of two selected parents;
                        end
                    end
                    $Topology[G\_size + 1] = Offspring1$;
                    $Topology[G\_size + 2] = Offspring2$;
                    $G\_size = G\_size + 2$; Current generation size update;
                else
                    $j = i$;
                end
            end
            if $G\_size = Generation\_size$ then Check current generation size, whether new generation obtained
                Break;
            end
        end
    until $G\_size = Generation\_size$;
end
```

The die area of and static sensitivity of the mechanical sensing element are also used as system performance objectives or constraints.

After the synthesis initialisation, the classical genetic operations of selection, crossover and mutation are applied to the current generation parents in order to create a new generation. In the selection operation, designs with better performance (higher fitness) are retained. After the selection, if the crossover operation is triggered, i.e. crossover probability is higher than a fixed threshold, new MEMS layouts are composed from primitives and associated control systems by exchanging elements of randomly selected parents, such as mechanical springs and electronic control blocks. The crossover algorithm for mechanical sensing element layout design is outlined in Algorithm 4.2. Details of an example of the crossover operation in mechanical sensing element synthesis are illustrated in Figure 4.5. As shown in the figure, in this example only the crossover probability of the spring component is higher than the trigger probability of 70%, so the spring components of parents A and B exchange leaving the other components unchanged in the creation of new offspring.

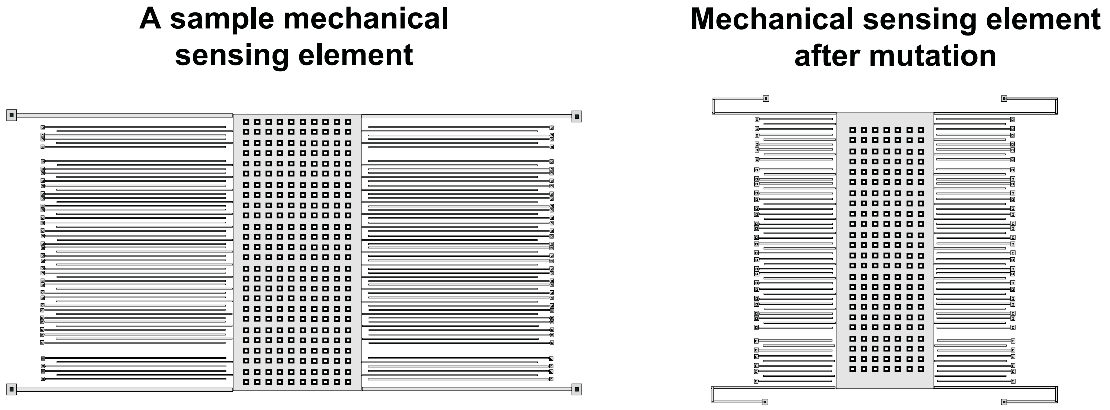


Components	Crossover probability of 2 selected parents (70% fixed)
Spring	96% **Exchange**
Proof mass	54% **No exchange**
Comb fingers	60% **No exchange**

Figure 4.5: An example of crossover operation in mechanical layout synthesis

For each individual in the new generation, the genes in their chromosomes have a fixed probability to mutate at random positions. The mutation operation is required to prevent falling genetic algorithm into local optimum. The mutation operation for the mechanical sensing element is illustrated in Algorithm 4.3 and also shown by the example in Figure 4.6 and Table.4.2.

The mutation operation contains two phases: component mutation and component parameter mutation. In the first phase, if the mutation probability for the components is higher than the fixed trigger (50% in this example) such as the beam spring and comb fingers, new components are automatically composed using the MEMS primitive library and each parameter of the mutated components gets a random value within its


A sample mechanical sensing element

Mechanical sensing element after mutation

FIGURE 4.6: An example of mutation operation in mechanical layout synthesis

MEMS component	***Mutation probability (trigger 50%)***	***Component parameters***	***Parameters mutation probability (trigger 60%)***
Beam Spring	56% (spring mutation)		Parameters of mutated spring get random value within range
Proof mass with etching holes	30% (No mutation)	L_m:Length	55% No parameter mutation
		W_m:Width	70% Parameter mutation
		W_h:Size of holes	23% No parameter mutation
		N_h:Number of holes	92% Parameter mutation
		T:Thickness	10% No parameter mutation
Comb fingers	73% (finger mutation)		Parameters of the mutated sense and force fingers get random value within range

TABLE 4.2: An example of mutation operation in mechanical layout synthesis

specified range. If there is no mutation in the first phase, the mutation probability of each component parameter will be compared with the trigger (60% in the example) to decide whether this parameter should mutate. As shown in Figure 4.6, after mutation the beam spring mutated to a folded spring and the comb fingers mutated to themselves but with different parameters such as a shorter length and a higher number of force fingers. For the proof mass, only the number of holes and width were changed at the second mutation phase. In this research, several mutation trigger points are tested to decide the suitable value for this design. A low mutation trigger may result in loss of good solutions; while a very large mutation trigger may lead the GA to obtain a local optimum [141]. Finally, mutation triggers for both component mutation and parameter mutation are set to 85% in this research.

This evolution process finishes when the generation number exceeds the specified maximum number. The optimal solution within a given generation is that with the highest fitness.

Algorithm 4.3: Proposed mutation operation algorithm for mechanical sensing element

```
Input: Generated new generation:$Topology[Population\_Size]$,
       Component mutation probability:$P\_component$,
       Parameter mutation probability:$P\_parameter$
Output: Topologies after mutation: $Topology[Population\_Size]$
begin
    for $i = 1 : +1 : Population\_Size$ do
        for $j = 1 : +1 : Component\_Size$ do
            $P(Topology[i].component[j]) = rand()$;Generate random topology mutation probability for component Topology[i].component[j];
            if $P(Topology[i].component[j]) > P\_component$ then
                Component is replaced by a new one which is randomly chosen from primitive library;
                Each parameter of the new component gets random value within range;
            else
                for $K = 1 : +1 : Component\_parameter\_Size$ do
                    $P(Topology[i].component[j].parameter[k]) = rand()$;Generate random parameter mutation probability for parameter Topology[i].component[j].parameter[k];
                    if $P(Topology[i].component[j].parameter[k]) > P\_parameter$ then
                        parameter mutates to new value within range;
                    end
                end
            end
        end
    end
end
```

4.1.3 Synthesis verification to provide appropriate performance metrics for the synthesised MEMS geometries

The practical operation of the proposed synthesis flow for the accelerometer embedded in a conventional Sigma-Delta control loop is demonstrated by three experiments listed in Table.4.3. In the first experiment, the system is optimised for maximum SNR with performance constraints, and in the second and third experiments - for maximum static sensitivity and minimum area respectively.

	Design objective	**Performance constraints**	**Synthesised layout**	**SNR** (dB)	**Static sensitivity** (fF/g)	**Area** (m^2)
1	Maximum SNR	SNR>30dB Area<2e-7m^2 $S > 1fF/G$	Fig.4.8.(a)	39.8	1.8	1.82e-7
2	Maximum static sensitivity	SNR>30dB $S > 2fF/G$	Fig.4.8.(b)	32.9	4.77	3.78e-7
3	Minimum area of mechanical sensing element	SNR>30dB Area<1.5e-7m^2	Fig.4.8.(c)	31.5	0.27	1.07e-7

Table 4.3: Synthesis experiments.

The synthesis process was carried out using the following design parameters (defined in Section 3.1):

1) Oversampling ratio: OSR=128

2) Bandwidth: $f_0 = 512Hz$

3) Oversampling frequency: $f_s = 2^{17}Hz \approx 131KHz$

4) Input acceleration: Sinusoidal acceleration with 100Hz frequency and 1g amplitude ($a_{in} = 9.8m/s^2$, $f_{in} = 100Hz$)

The fitness functions for these three experiments are listed below:

Experiment 1(maximum SNR):

$$Fitness = w\frac{SNR}{SNR'} \tag{4.5}$$

where SNR' is the designer specified objective value (30dB in Experiment 1). SNR is obtained from a performance evaluation engine which is embedded in synthesis flow to enable measurements of the power spectrum density (PSD) and SNR through FFT of the output bitstream after each simulation. w is the weight coefficient which is set to

1 if all user-defined performance constraints are met, otherwise w is set to 0.0001. For example, in Experiment 1, if a synthesised design can achieve 30dB SNR with area and static sensitivity sensing element less than $2.5e-7m^2$ and $1.0fF/g$, w equals 1 that means the algorithm finds a feasible solution satisfying specified performance.

Experiment 2 (maximum static sensitivity of sensing element):

$$Fitness = w\frac{S}{S'} \tag{4.6}$$

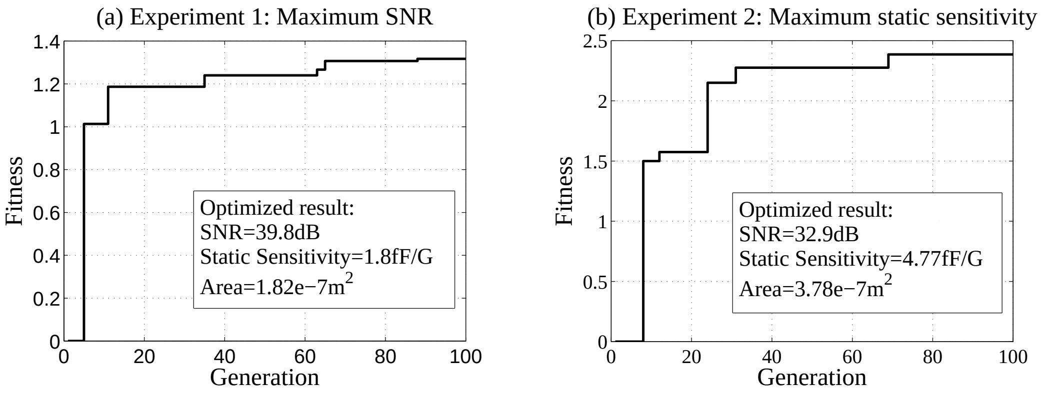
where S is the static sensitivity of the synthesised sensing element and S' is the user-defined objective value $(2fF/g)$ in this experiment. The weight coefficient w has the same value as specified in Experiment 1.

Experiment 3 (Minimum area of mechanical sensing element):

$$Fitness = w\frac{Area}{Area'} \tag{4.7}$$

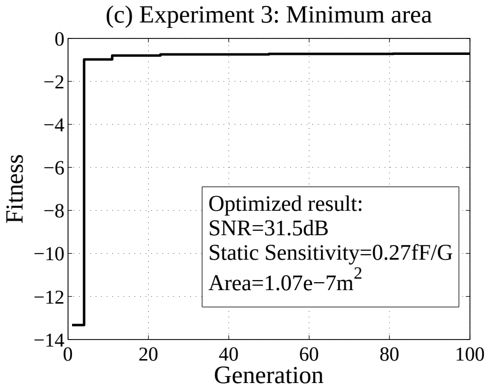
where $Area$ is the die area of the synthesised mechanical sensing element and $Area'$ is the predefined objective value $(1.5e-7m^2)$. In order to maximise the fitness parameter, w is set to -1 if performance constraints are met or -10 otherwise.

Design of MEMS accelerometer in a Sigma-Delta force-feedback control loop contains many crucial trade-offs. For example, in this case study, static sensitivity is dependent on the length and the number of sense fingers. However, the performance of Sigma-Delta modulation may be severely affected by the length of sense fingers to the extent that a complete failure of the Sigma-Delta control may occur when the fingers are too long. The maximum number of fingers is also limited by the length of proof mass. The presented genetic-based synthesis approach deals with these trade-offs effectively for a given choice of the design objectives.



(a) Experiment 1: Maximum SNR

(b) Experiment 2: Maximum Static sensitivity



(c) Experiment 3: Minimum area

FIGURE 4.7: Fitness improvement between generations

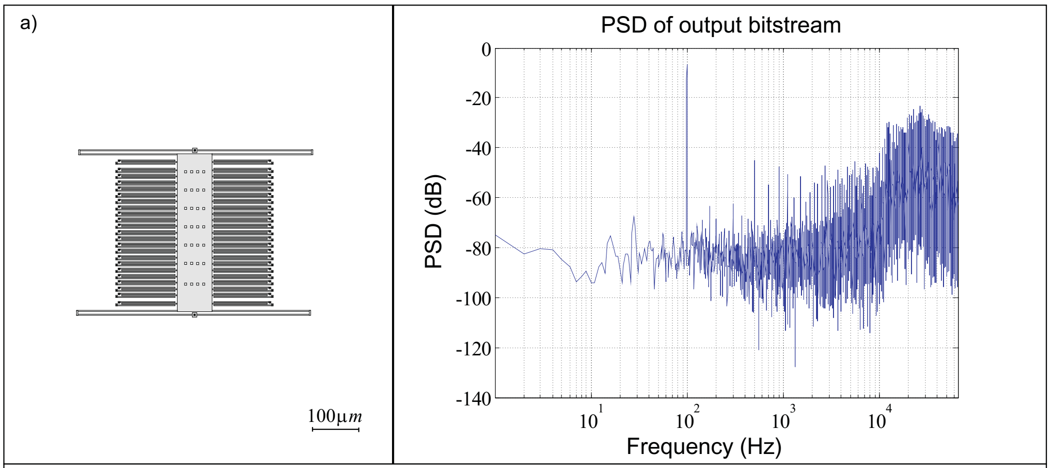
The fitness improvement during the synthesis flow is shown in Figure 4.7. The synthesised mechanical layouts and parameters of its associated electronic control system are shown in Figure 4.8 and Table 4.4. As can be seen from the synthesised results for each experiment, the genetic synthesis algorithm composed different layout structures and produced different performance parameters. As expected, the structure optimised for maximum sensitivity has more and longer sense fingers. Area optimised accelerometer in experiment 3 shows a great area improvement over other experiments. The control loop is fixed in this case study to form a conventional second-order electromechanical Sigma-Delta modulator. However, the noise floor in higher order electromechanical Sigma-Delta modulator can be reduced drastically leading to great improvement of the SNR comparing with second-order Sigma-Delta accelerometer. It is discussed in the next section where the higher-order control system is automated optimal synthesised with layout synthesis of mechanical sensing element simultaneously.

MEMS components	***Experiment 1***	***Experiment 2***	***Experiment 3***
Proof mass	Ml = $341\mu m$ Mw = $73\mu m$ T = $2.9\mu m$ Wh = $4.9\mu m$ Nh = 28	Ml = $695\mu m$ Mw = $136\mu m$ T = $2.85\mu m$ Wh = $4.2\mu m$ Nh = 496	Ml = $205\mu m$ Mw = $125\mu m$ T = $2.5\mu m$ Wh = $5.7\mu m$ Nh = 40
Comb fingers	Lf = $122\mu m$ Tf = $2.2\mu m$ d0 = $1.0\mu m$ Ns = 42 Nf = 4 Wanchor=$4\mu m$	Lf = $183\mu m$ Tf = $2.1\mu m$ d0 = $1.5\mu m$ Ns = 50 Nf = 10 Wanchor=$4\mu m$	Lf = $84.6\mu m$ Tf = $2\mu m$ d0 = $1.36\mu m$ Ns = 24 Nf = 8 Wanchor=$4\mu m$
Spring	(Folded spring) Lo1 = $218\mu m$ Lo2=$255\mu m$ Wo = $2\mu m$ Lp = $11.5\mu m$ Wp = $2.1\mu m$	(Classic serpentine spring) N = 2 Lo = $182\mu m$ Wo = $2.0\mu m$ Lp =$4.5\mu m$ Wp = $2.6\mu m$ Lroot=45μ W=$5\mu m$	(Beam spring) Lo =$200\mu m$ Wo = $2.0\mu m$
Sigma-Delta Control system	Vf = 0.6V Vm = 1.2V K_{amp}=23.8 zero=1KHz pole=120KHz	Vf = 0.94V Vm = 1.0V K_{amp}=4.8 zero = 2KHz pole= 130KHz	Vf = 0.72V Vm = 1.5V K_{amp}=9 zero = 1KHz pole= 100KHz

Table 4.4: Summary of synthesised results for Experiments 1, 2 and 3

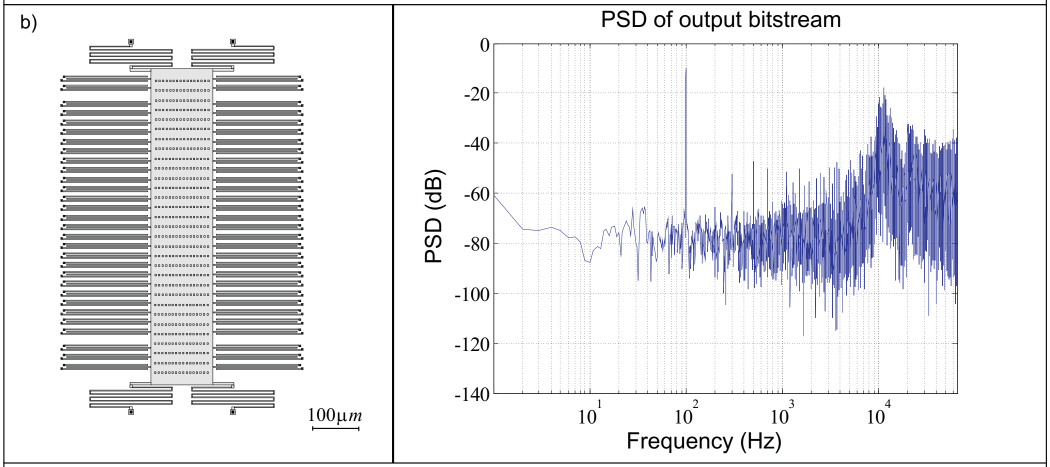
4.2 Synthesis of a high-order MEMS accelerometer with associated control loop

Conventionally, the mechanical sensing element of a MEMS sensor is used as a loop filter to form a second-order single-loop electromechanical Sigma-Delta modulator. This is because the sensing element can be approximated by a second-order mass-damper-spring transfer function which performs the similar function to that of two cascaded integrators in typical second-order electronic Sigma-Delta modulators. In such a configuration, the dynamics of the mechanical sensing element limit the noise shaping properties. Compared with typical electronic second-order Sigma-Delta modulators, the DC gain of mechanical integrators is quite low which results in a lower SNR in second-order electromechanical Sigma-Delta modulators [118]. This is considered insufficient in high performance applications such as inertial navigation systems.



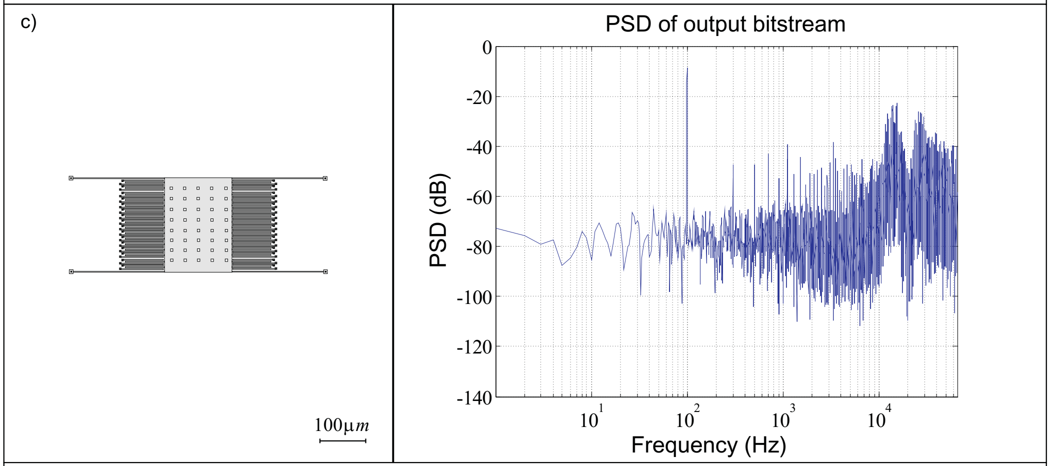
System Performance

SNR=39.8dB Sensitivity= 1.8e-015F/g area=1.82e-007m^2 Resonant frequency= 8.2KHz



System Performance

SNR=32.9dB Sensitivity= 4.77e-015F/g area=3.78e-007m^2 Resonant frequency= 4.6KHz



System Performance

SNR=31.5dB Sensitivity= 2.7e-016F/g area=1.07e-7m^2 Resonant frequency= 7.9KHz

FIGURE 4.8: Synthesised results. a): Experiment 1 (Maximum SNR); b): Experiment 2 (Maximum static sensitivity); c): Experiment 3 (Minimum area of mechanical sensing element)

In order to improve the SNR, higher-order electromechanical Sigma-Delta control schemes design becomes increasingly attractive. A general topology of a high-order electromechanical Sigma-Delta modulator is shown in Figure 4.9. The performance of the MEMS sensor embedded in a high-order Sigma-Delta control loop is greatly improved due to the additional purely electronic loop filters. Thus, the design of the higher order electromechanical Sigma-Delta modulator is focused on the loop filter structure.

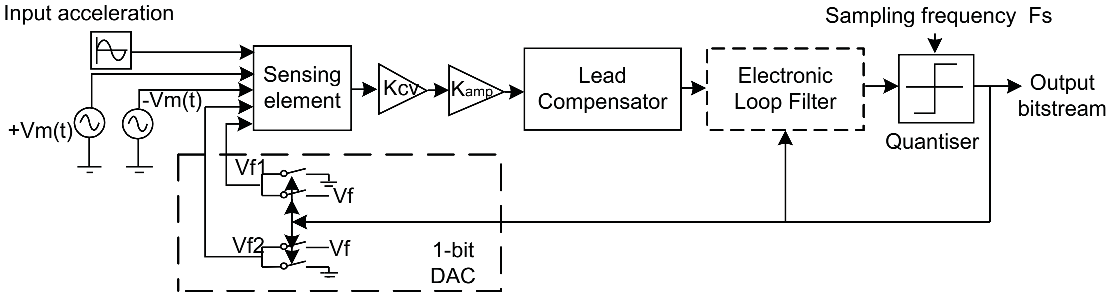


Figure 4.9: Configuration of a high-order electromechanical Sigma-Delta modulator

The high-order electronic loop filter can be developed by a series of integrators using different topologies such as multiple feedback topologies and a combination of distributed feedback and feedforward topologies. It is worth noting that not only the topology and order of the electronic integrator loop filter but also the mechanical sensing element determine the noise shaping in a high-order electromechanical Sigma-Delta modulator. This means that both the loop stability and the SNR depend on the sensor as well as loop filter parameters. Dong *et al.* [118] used a parameter sweep method to explore the optimal coefficients of the loop filters for several fixed topologies. The mechanical sensing element is also fixed. This limits the adaptability of the control system to different types of sensors.

This section presents a novel genetic-based methodology for automated optimal synthesis of high-order Sigma-Delta control topology for MEMS sensors. It develops further the concepts presented in Section 4.1 which focuses on the layout synthesis of the mechanical part. A case study is discussed where the proposed genetic-based synthesis approach implemented in SystemC-AGNES is applied to a high-order Sigma-Delta control system in an electromechanical MEMS accelerometer. The proposed approach efficiently generates suitable configurations of the Sigma-Delta control loop by combining primitive components stored in a library and optimises them according to the user specifications. This methodology can efficiently explore the configuration space and develop new structures with better performance. Compared with a manually designed fifth-order electromechanical Sigma-Delta modulator, the synthesised design gets 20dB improvement for SNR. The approach is combined with the layout synthesis of the mechanical sensing element described in last section to realise the automated optimal design of MEMS systems embedded in electronic control circuitry from user-defined high-level performance specifications and design constraints.

4.2.1 Synthesis initialisation

In the initialisation phase of the synthesis process, a set of configurations is automatically generated from data in the MEMS and electronic control loop primitive libraries to create the first generation of the GA. The MEMS primitive library is discussed in Section 4.1.1, and the primitives stored in the electronic control loop primitive library are explained below. Sample primitive components of the electronic control loop are shown in Figure 4.10. New loop filter topologies will be automatically generated from these primitives.

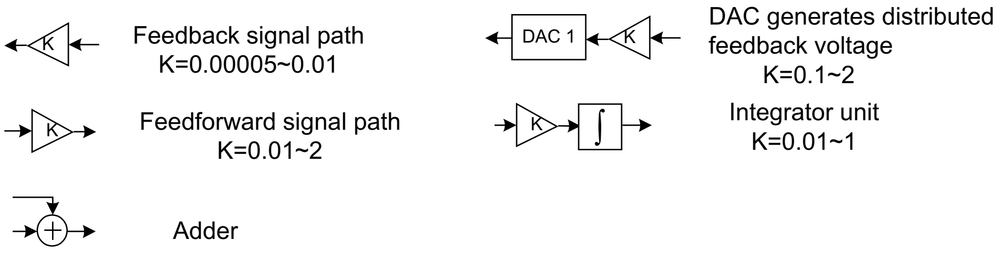


FIGURE 4.10: Primitive components in the electronic control loop library

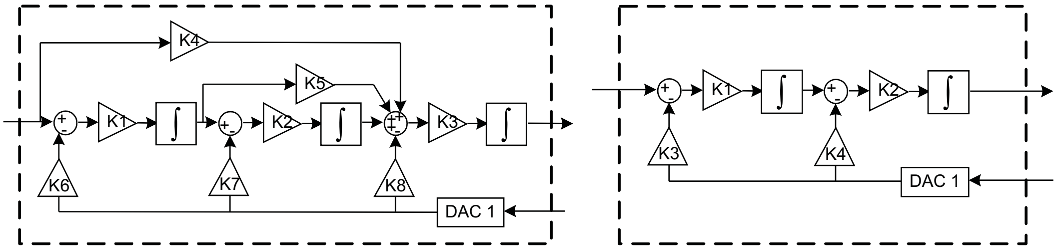
The electronic loop filter synthesis is based on a series of integrators (Integrator unit in the library). The maximum number of the integrators is defined by users and the minimum number of integrators is set to zero to form the conventional second-order electromechanical Sigma-Delta modulator since the sensor may be used in some applications whose requirement of performance is not crucial. The electromechanical Sigma-Delta modulator should ideally benefits the advantages of the mature topologies used in Sigma-Delta A/D converters with feedforward or feedback paths or the combination of them both. Thus, feedback and feedforward paths are added to the library. DAC1 is used to generate the distributed feedback voltage to integrator units from output bitstream.

Typically, the distributed feedback signal from DAC1 is used to determine the pole positions and loop stability. Feedback paths between integrators generate complex pair of zeros in order to further suppress the total noise in signal band.

Algorithm 4.4: Initial loop filter topologies generation

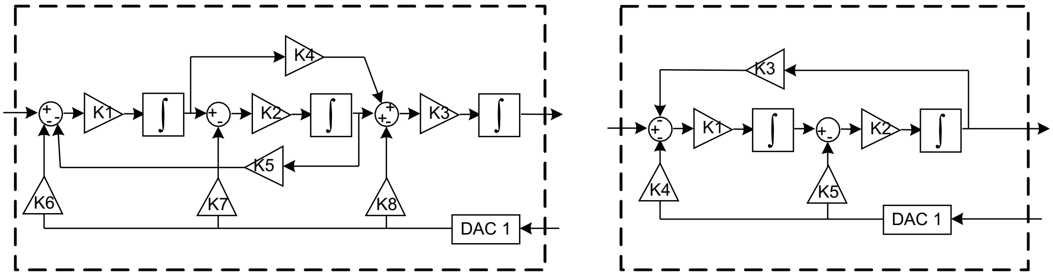
```
Input: SystemC-A electronic control loop primitive library
Output: Initial loop filter topologies:$topology[Population\_Size]$
begin
    for $i = 1 : +1 : Population\_Size$ do
        A chain of integrators are connected. The number of integrators is determined
        by the predefined order of loop filter;
        Randomly generate feedback and feedforward signal paths among input, output
        and integrators;
        Each signal path gets initial random value within range,topology[i] is generated;
    end
end
```

The automated generation of the loop filter topology in the initialisation phase is divided into several steps as outlined in Algorithm 4.4. Firstly, the system will generate a random number of integrators to determine the order of the loop filter (N). The maximum allowed order of the loop filter (Nmax) is defined by the user. The number of integrators can be zero such that a conventional second-order electromechanical Sigma-Delta modulator can be generated without an electronic loop filter. Each integrator is randomly connected with DAC1 and other integrators by feedforward and feedback signal paths to produce different topologies of the loop filter. Some feasible configurations of loop filters, which can be generated by combining primitives using Algorithm 4.4, are illustrated in Figure 4.11. These feasible configurations of loop filters are analysed through simulations.



3rd order loop filter with distributed feedback and feedforward paths topology

2nd order loop filter with distributed feedback topology



3rd order loop filter with combination of feedforward and feedback paths topology

2nd order loop filter with feedback paths topology

FIGURE 4.11: Examples of feasible configurations of the electronic loop filter generated by the Algorithm 4.4

Figure 4.12 and Table 4.5 show a sample feasible configuration of a fourth-order electromechanical Sigma-Delta modulator to illustrate the parameter initialisation and encoding phase. The corresponding parameter initialisation and encoding for the mechanical sensing element was discussed in Section 4.1. The loop filter in this sample MEMS accelerometer configuration is based on a second-order distributed feedback and feedforward topology, and it contains two integrator units with one feedforward path between them.

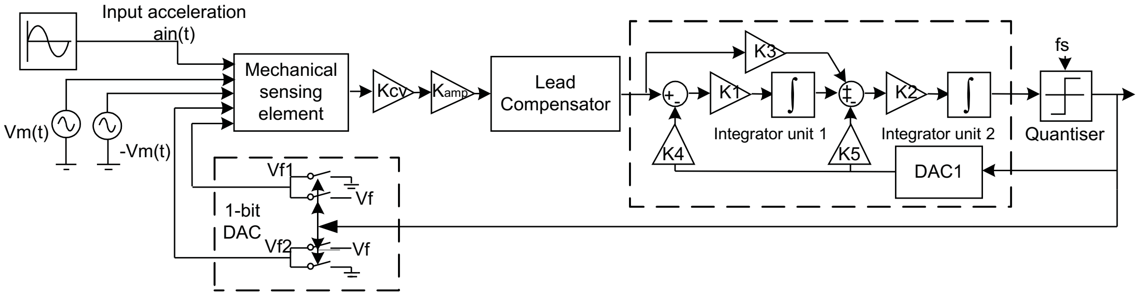


FIGURE 4.12: Example of parameter initialisation and encoding.

Loop Filter Topology	***Value***	***Encoding***	***Description***
Order of loop filter	2	1 2	Integrator 1 Integrator 2
Feedback path	0	0	No feedback path between integrators
Feedforward path	1	1	Feedforward path from input to to integrator 2
Distributed feedback from DAC1	2	1 2	Feedback to integrator 1 Feedback to integrator 2

Table 4.5: Representation of a population member of the MEMS Sigma-Delta electronic loop filter example in GA

4.2.2 Genetic synthesis of electronic control

In the genetic-based synthesis approach, exploration of the solution is also guided by the fitness functions which were illustrated in the last section. The SNR, die area and static sensitivity of the mechanical sensing element are used as system performance constraints or objectives to compare the synthesis results with the results reported in Section 4.1.

The topology synthesis of the loop filter can be divided into two steps: selection and new generation reproduction. In the selection phase, a proportion of designs in the current generation are retained through a fitness-based process (measured by fitness function) to breed the next generation. In the reproduction phase, the standard genetic operations of crossover and mutation are applied to the selected designs to generate the new generation. Firstly, in the crossover operation, the synthesis flow randomly chooses any two topologies as parents to generate offsprings. An example of the crossover operation is shown in Figure 4.13 and Table 4.6. In this example, crossover probabilities of compensator and loop filter components are higher than the user defined trigger probability 70% that means these two components of selected parents will exchange to create offspring. As long as the crossover operation is triggered, the system will automatically judge the mechanical sensing element whether it is an under-damped system or not after crossover. This operation is used to determine whether the lead compensator is required. The crossover operation will end when a new generation is obtained.

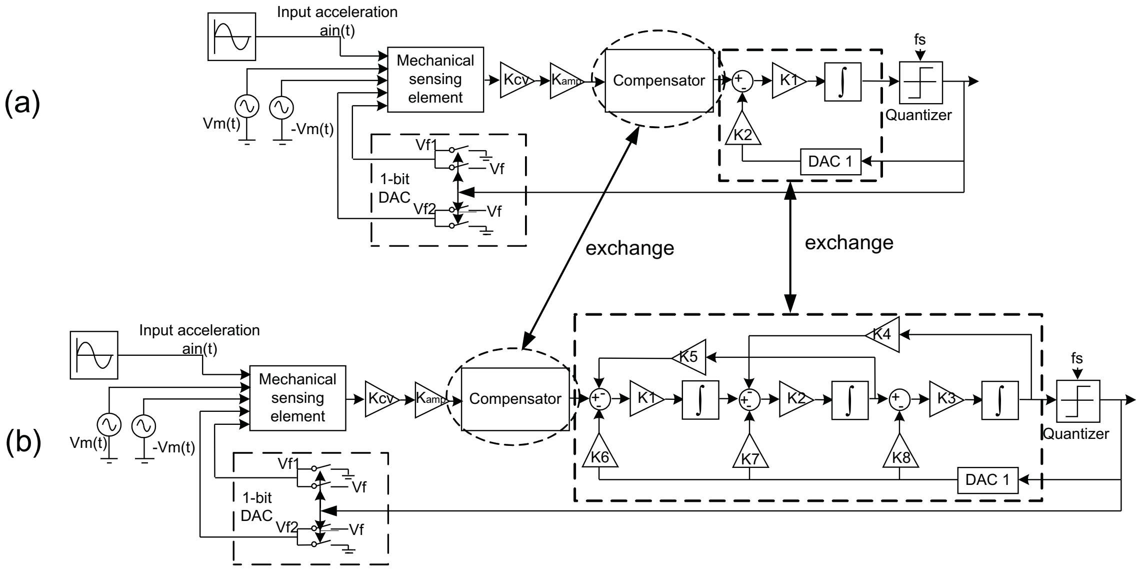


FIGURE 4.13: An example of crossover operation in Sigma-Delta control loop synthesis. a)Parent A:third-order electromechanical Sigma-Delta modulator; b)Parent B: fifth-order electromechanical Sigma-Delta modulator

Control system components	***Crossover probability (Trigger probability 70%)***	***Description***
Compensator	83%	Exchange compensators
Loop filter	74%	Exchange Loop filters
1-bit DAC	21%	No exchange of DACs
Gain block (K_{cv})	65%	No exchange of gain blocks K_{cv}
Gain block (K_{amp})	32%	No exchange of gain blocks K_{amp}

TABLE 4.6: An example of crossover operation in Sigma-Delta control loop synthesis

Subsequently, every individual in the new generation gets a fixed probability to mutate. The process of mutation operation is outlined in Algorithm 4.5. The mutation operation for the loop filter topology contains two phases: topology mutation and component parameter mutation. In the topology mutation phase, if the topology mutation probability of a selected design is higher than the user defined trigger (50% as an example), a new topology is generated from the electronic control loop primitive library, and each parameter of the mutated component gets a random initial value within the allowed value range as illustrated in Figure 4.10.

Figure 4.14 and Table 4.7 show an example of the topology mutation process. In this example, the configuration of the randomly selected third-order loop filter mutated to

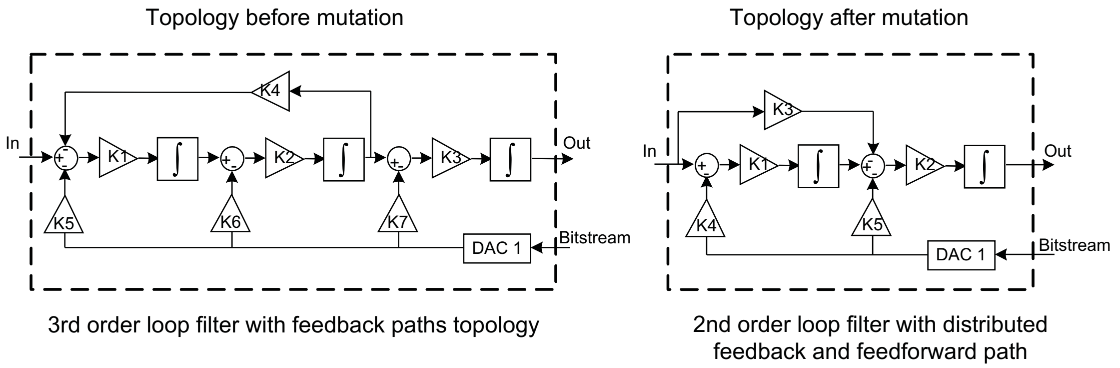
Algorithm 4.5: Proposed mutation operation algorithm for electronic loop filter

```
Input: Generated new generation:$Topology[Population\_Size]$,
       Topology mutation probability:$P\_topology$,
       Parameter mutation probability:$P\_parameter$
Output: Topologies after mutation: $Topology[Population\_Size]$
begin
    for $i = 1 : +1 : Population\_Size$ do
        $P(Topology[i]) = rand()$;Generate random topology mutation probability for Topology[i];
        if $P(Topology[i]) > P\_topology$ then
            Generate chain of integrators determined by the predefined order of Sigma-Delta modulator;
            Randomly determine the type of integrators;
            Generate feedforward and feedback paths among input, output, and integrators;
            Each signal path get random value within constraints;
        else
            for $j = 1 : +1 : Parameter\_Size$ do
                $P(Topology[i].Parameter[j]) = rand()$;Generate random parameter mutation probability;
                if $P(Topology[i].Parameter[j]) > P\_parameter$ then
                    parameter mutates to new value within constraints;
                end
            end
        end
    end
end
```

a topology with a second-order distributed feedback and a feedforward path. Each parameter of the new loop filter is randomly initialised within the constraints following the topology mutation operation. If there is no topology mutation for the selected design, the parameters of each component in the design has a chance to mutate while keeping the topology unchanged.

4.2.3 Synthesis experiments of MEMS accelerometer with high-order Sigma-Delta control loop

The proposed synthesis flow for MEMS accelerometer with high-order Sigma-Delta control is illustrated by four experiments as shown in Table 4.8. In the first two experiments, the systems are optimised for maximum SNR with different performance constraints, and



3rd order loop filter with feedback paths topology

2nd order loop filter with distributed feedback and feedforward path

FIGURE 4.14: An example of mutation operation in loop filter synthesis.

Control loop component	***Topology mutation probability (trigger 50%)***	***Topology generation process***	***Description***
Loop filter	71% Topology mutate	1.Order of loop filter = 2	Generate random order of loop filter within range (0 to N_{max}), In this example, it is second order
		2.Integrator units encoding Integrator 1 & Integrator 2	Encode Integrators,2 integrators in the new topology
		3.Distributed feedback path generation Integrator 1:61% > 50%(trigger) Integrator 2:89% > 50%(trigger)	If generation probability over trigger, integrator gets feedback from DAC1
		4.Feedback path generation generation probability: 30% < 50%(trigger)	If probability over trigger, feedback path is generated between Integrator 1 and 2
		5.Feedforward path generation From In to Out: 28% < 50%(trigger) From In to Integrator 2: 92% > 50%(trigger) From Integrator 1 to Out: 46% < 50%(trigger)	If probability over trigger, feedforward path is generated

TABLE 4.7: An example of mutation operation in loop filter synthesis.

in the third and forth experiments - for maximum static sensitivity and minimum area respectively. In order to compare the results with the second-order electromechanical Sigma-Delta modulators in Section 4.1, the same design parameters are applied in the synthesis process:

1) ***Oversampling ratio:*** OSR=128

2) ***Bandwidth:*** $f_0 = 512Hz$

	Design objective	**Performance constraints**	**Synthesised topology**	**SNR** (dB)	**Static sensitivity** (fF/g)	**Area** (m^2)
1	Maximum SNR	SNR>90dB Area<1.5e-7m^2	Fig.4.17	108	0.246	1.41e-7
2	Maximum SNR	SNR>90dB	Fig.4.18	114	0.76	2.7e-7
3	Maximum static sensitivity	SNR>75dB Area<3.0e-7m^2 $S > 2fF/g$	Fig.4.19	88.4	2.27	2.1e-7
4	Minimum area of mechanical sensing element	SNR>75dB Area<1.5e-7m^2	Fig.4.20	85	0.11	8.5e-8

TABLE 4.8: Synthesis experiments.

3) Oversampling frequency: $fs = 2^{17}Hz \approx 131KHz$

4) Input acceleration: Sinusoidal acceleration with 100Hz frequency and 1g amplitude ($a_{in} = 9.8m/s^2$, $f_{in} = 100Hz$)

5) Maximum order of electronic loop filter: $N_{max} = 3$

As the maximum order of electronic loop filter is set to 3 in the experiments, the maximum order of the electromechanical Sigma-Delta modulator is 5. Fitness improvement of the synthesis flow is shown in Figure 4.15. The topology of manually designed fifth-order electromechanical Sigma-Delta modulator is illustrated in Figure 4.16 and Table 4.9. The synthesised mechanical layouts and its associated Sigma-Delta control system are shown in Figure 4.17-4.20 and Table 4.9. The system output bitstream is measured by its PSD illustrated in Figure 4.16-4.20.

The objectives of Experiments 1 and 2 are to maximise the SNR but with different constraints as shown in Table 4.8. It is worth noting that the SNR in Experiment 2 is further improved from that in Experiment 1 because the area is not constrained. Both the synthesised results of experiment 1 and 2 show better performance than the manual designed Sigma-Delta accelerometer with same order control system. Compared

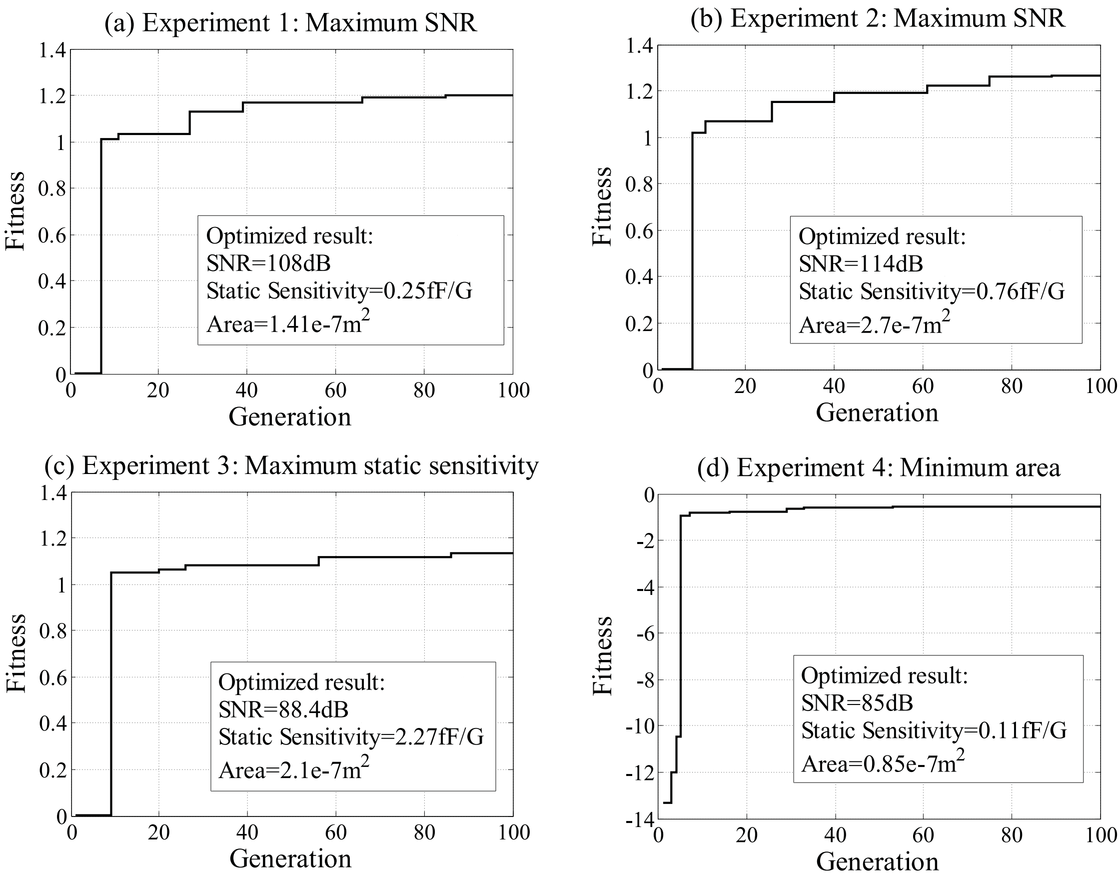


FIGURE 4.15: Fitness improvement between generations

with the manual design shown in Figure 4.16, the synthesis Experiment 2 improved the SNR figure by nearly 20dB. As expected, the accelerometer optimised for the area in Experiment 4 shows an almost threefold area improvement over the manual design but the SNR figure is degraded by about 10dB.

It can be seen from the results of the above synthesis experiments that the proposed synthesis approach efficiently explores the design space to generate suitable configurations of MEMS mechanical layout and its associated Sigma-Delta control loop by combining primitive components stored in the libraries and optimises them according to the user-defined specifications.

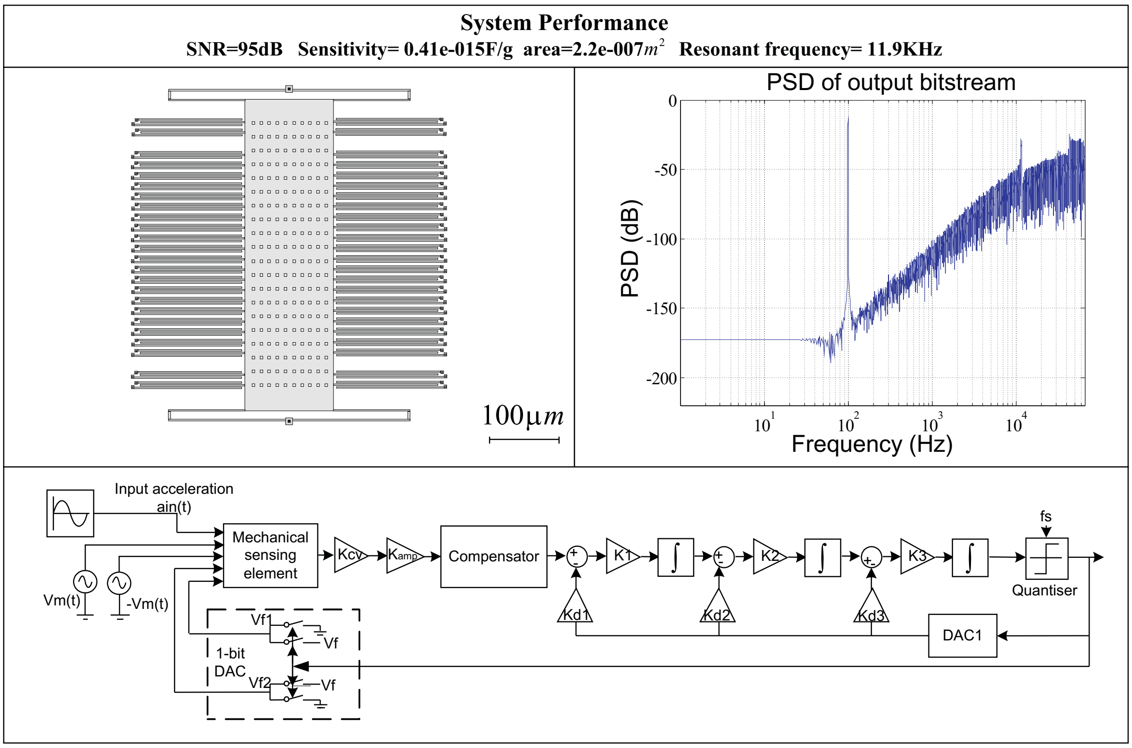


FIGURE 4.16: Manual Design (Fifth-order Sigma-Delta accelerometer)

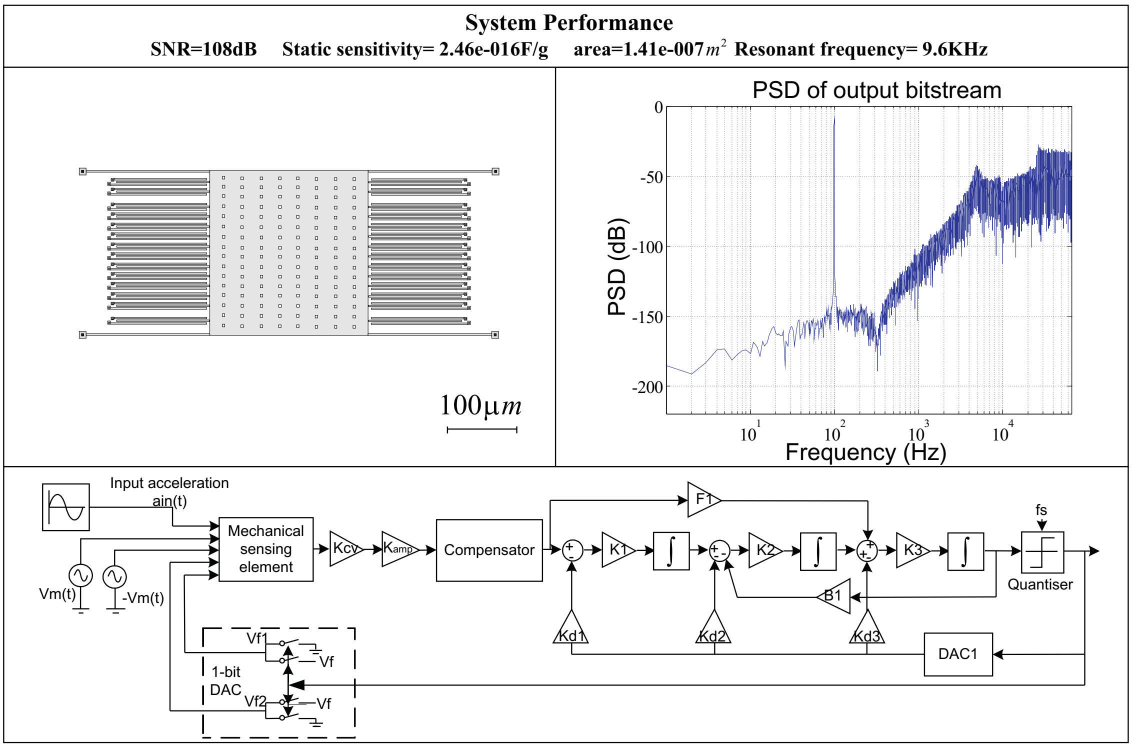


FIGURE 4.17: Synthesised result in Experiment 1 (Maximum SNR).

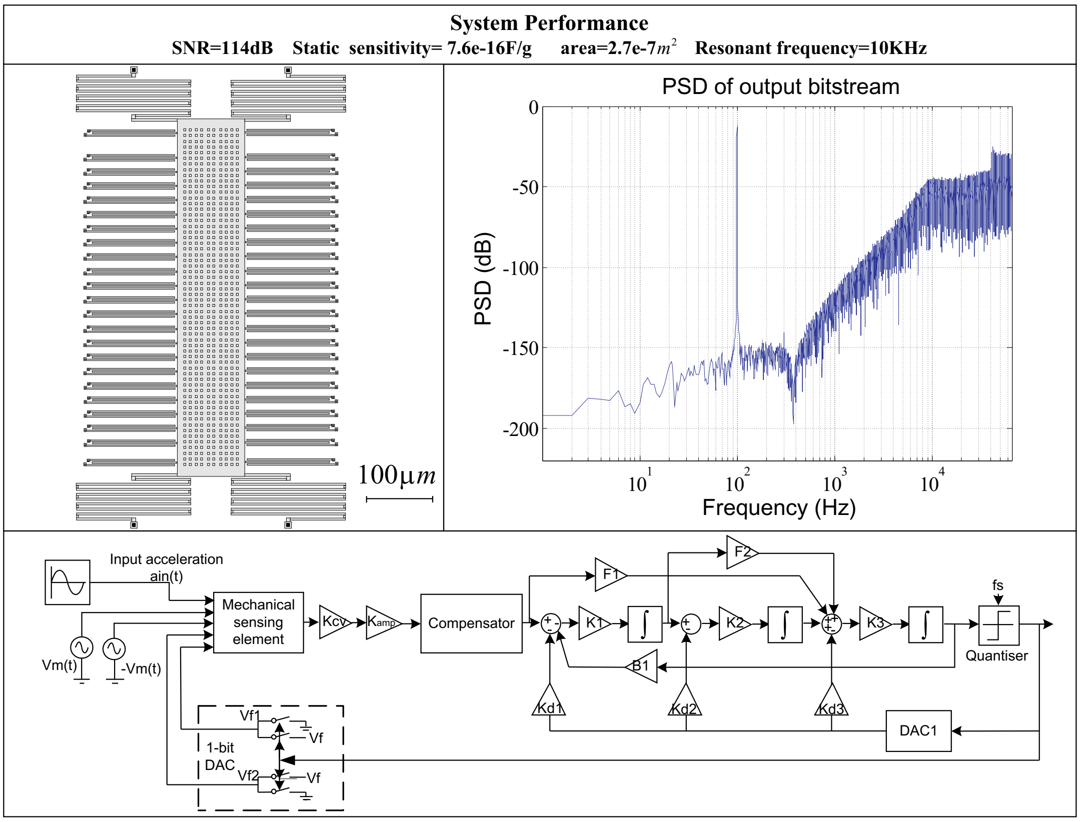


Figure 4.18: Synthesised result in Experiment 2 (Maximum SNR).

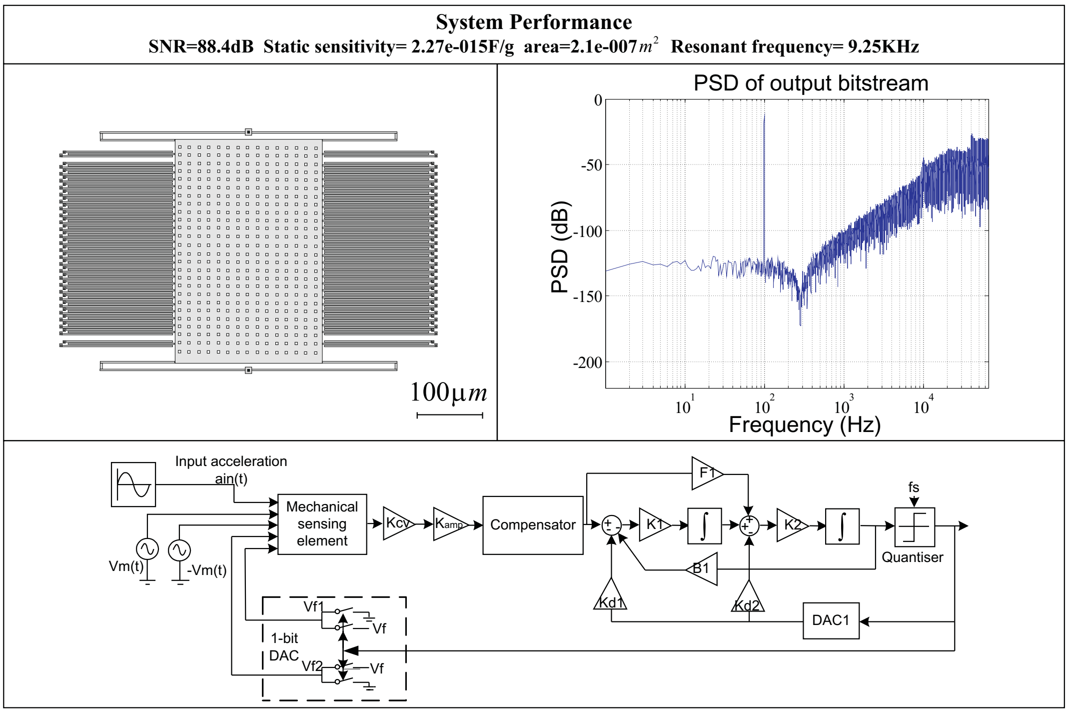


Figure 4.19: Synthesised result in Experiment 3 (Maximum static sensitivity of sensing element).

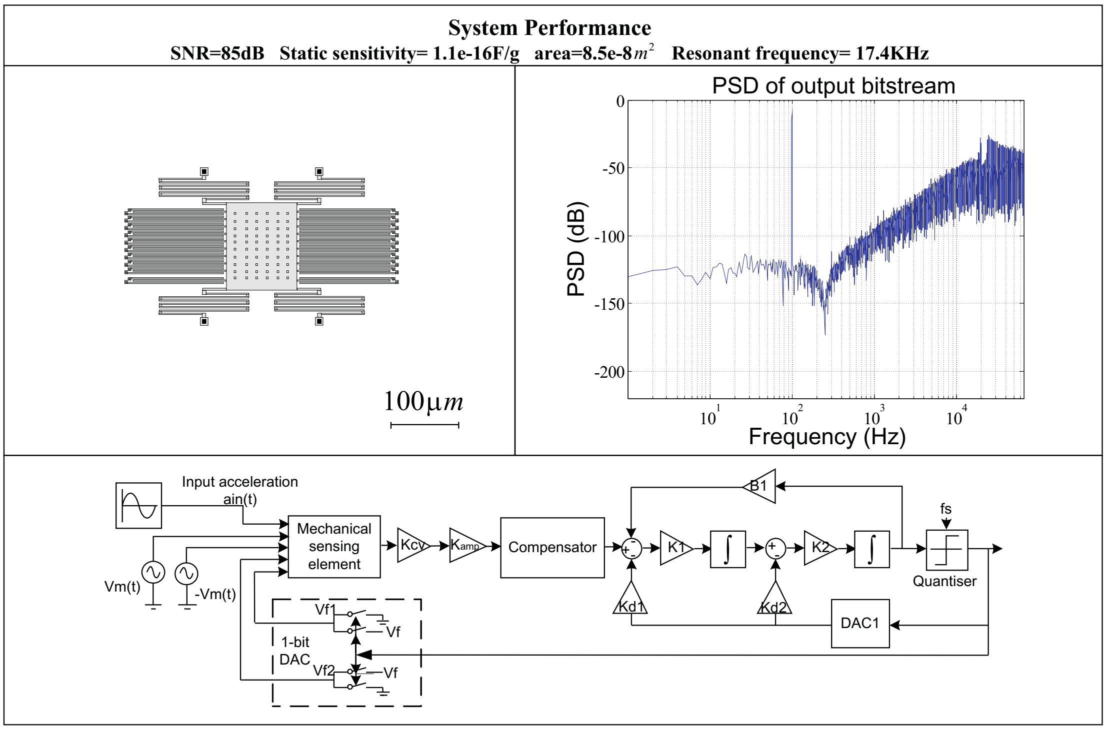


FIGURE 4.20: Synthesised result in Experiment 4 (Minimum area of sensing element).

4.3 Genetic-Based High-Level Synthesis of Sigma-Delta Modulator in SystemC-AGNES

High-level design of Sigma-Delta modulators remains mostly manual and it is critical to explore the feasible topologies because of the large number of connections between components in Sigma-Delta modulators (integrators, DAC, quantiser). Typically, a library of traditional topologies is available from which designers can select according to their experience, while structure design is accessible only to a small number of expert designers [143].

In order to decrease the complexity of the design procedure, several tools for automated design of Sigma-Delta modulators have been developed recently [143, 144, 145, 146, 147, 148]. Most of the methodologies are based-on the optimisation of the coefficients of signal paths for preset popular Sigma-Delta modulator topologies [144, 145, 146]. Ruiz-Amaya *et al.* [146] developed a toolbox in MATLAB/Simulink environment to optimise the coefficients of the selected Sigma-Delta modulator structures using an adaptive statical

MEMS components	***Manual design***	***Experiment (1)***	***Experiment (2)***	***Experiment (3)***	***Experiment (4)***
Proof mass	Ml=450μm Mw=130μm T=2.5μm Wh=4μm Nh=200	Ml = 236μm Mw = 237μm T = 2.7μm Wh = 3.4μm Nh = 128	Ml = 563μm Mw = 102μm T = 3.0μm Wh = 3.0μm Nh = 600	Ml=341μm Mw = 225μm T = 2.7μm Wh = 4.5μm Nh = 390	Ml=130μm Mw = 107μm T = 2.2μm Wh = 3.5μm Nh = 60
Comb fingers	Lf=150μm Tf=2μm d0=1.5μm Ns=40 Nf=8 Wanchor =4μm	Lf = 136μm Tf = 2.0μm d0 = 1.74μm Ns = 22 Nf = 6 Wanchor =4μm	Lf = 130μm Tf = 2μm d0 = 1.4μm Ns = 42 Nf = 4 Wanchor =4μm	Lf = 164μm Tf = 2.3μm d0 = 1.0μm Ns = 50 Nf = 4 Wanchor =4μm	Lf = 139μm Tf = 2.0μm d0 = 1.5μm Ns = 18 Nf = 2 Wanchor =4μm
Spring	Folded spring Lo1=180μm Lo2=115μm Wo=2μm Lp=16.0μm Wp=2.0μm	Beam spring Lo =180μm Wo = 2.5μm	Classic serpentine spring N = 3 Lo = 177μm Wo = 3.0μm Lp =3.9μm Wp = 3.2μm Lroot=75μm	Folded spring Lo1=227μm Lo2=114μm Wo=2.5μm Lp=9.3μm Wp=2.0μm	Classic serpentine spring N = 2 Lo=134μm Wo=2.2μm Lp=3.2μm Wp =3.5μm Lroot=35μm
Control system	(5th order) Vf=0.45V Vm=1.5V K=20 zero=1KHz pole=300KHz K_{d1}=0.8 K_{d2}=0.8 K_{d3}=0.8 K_1=0.078 K_2=0.38 K_3=0.458	(5th order) Vf = 0.51V Vm = 1.1V K=27.3 zero =10KHz pole=82KHz K_{d1}=1.5 K_{d2}=1.5 K_{d3}=1.5 K_1=0.46 K_2=0.076 K_3=0.53 F_1=0.06 B_1=0.006	(5th order) Vf = 0.35V Vm = 1.2V K=33.7 zero=10KHz pole=200KHz K_{d1}=1.1 K_{d2}=1.1 K_{d3}=1.1 K_1=0.24 K_2=0.16 K_3=0.61 F_1=0.1 F_2=0.08 B_1=0.007	(4nd order) Vf = 0.3V Vm = 1.0V K=9 zero=1KHz pole=103KHz K_{d1}=1.5 K_{d2}=1.5 K_1=0.76 K_2=0.61 F_1=0.01 B_1=0.0004	(4nd order) Vf = 0.35V Vm = 1.2V K=48 zero=1KHz pole=100KHz K_{d1}=1.46 K_{d2}=1.46 K_1=0.38 K_2=0.42 B_1=0.001

TABLE 4.9: Summary of synthesised results for Experiments 1, 2, 3 and 4

optimisation algorithm based on simulated annealing. A behaviour simulation-based synthesis tool (DAISY) is programmed in C language by Francken *et al.* [147] A set of selected topologies are stored in a library. The synthesis tool automatically tested all the topologies in the library and chose the one with the smallest power consumption according to design specifications (SNR and signal bandwidth). The major limitation of these techniques is that the design space for topology exploration is restricted. Thus, only local optimality is achieved for predefined design objectives.

To overcome the limitation, some methodologies are presented to realise the topology synthesis for Sigma-Delta modulators [143, 148]. Tang [143] proposes an MINLP-based synthesis flow. In this approach, a generic representation, which describes all possible topologies for a certain order single-bit single-loop Sigma-Delta modulator, is defined to derive the symbolic TF(Transfer Function). The MINLP description contains nonlinear equations that express the generic TF and a cost function describing signal-path complexity, sensitivity, and power consumption. Finally, the MINLP description is embedded into a design flow to obtain the optimal topology satisfying the design specifications. However, the TF is difficult to build because the complexity of the symbolic terms grows roughly with the order of modulator [143]. In [148], Yetik creates a tool in MATLAB to automatically generate the transfer functions of Sigma-Delta modulators which are used as inputs for the synthesis algorithm to find all the possible topologies to achieve the desired frequency response. However, the coefficients of the synthesised topology are not optimised in this approach.

This section presented a novel methodology based on SystemC-AGNES for automated and optimal topology synthesis of Sigma-Delta modulators according to the design constraints. A single-loop Sigma-Delta modulator is used as a case study to demonstrate the proposed synthesis technique. However, this approach is general, it can be extended to multi-loop Sigma-Delta modulators.

4.3.1 Design initialisation

The genetic-based optimal synthesis flow for Sigma-Delta modulators is similar to that for digital MEMS sensors as shown in Figure 4.1. A Sigma-Delta modulator primitive library is developed, and its components are shown in Figure 4.21. For simplicity, non-idealities of components are not considered in this research.

In the design initialisation phase, a set of topologies is automatically generated by assembling the primitives in the library and loading them into the synthesis module as the

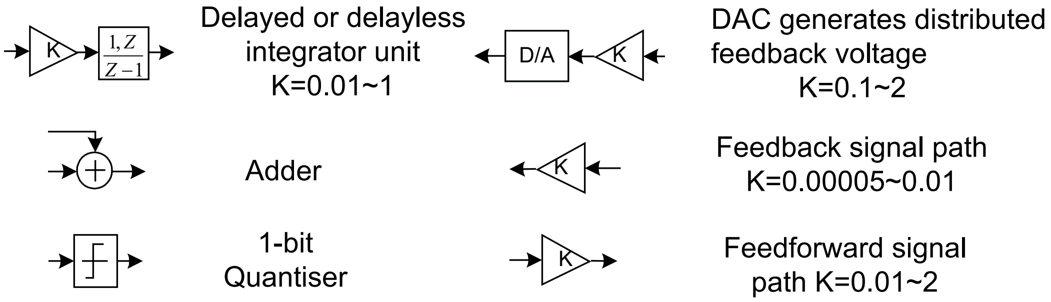


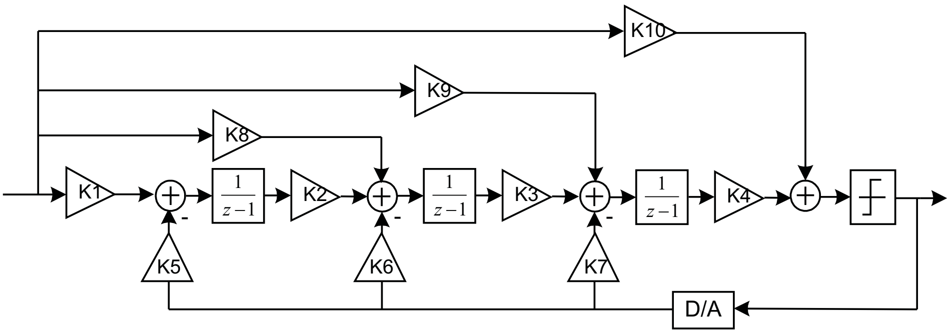
FIGURE 4.21: Sigma-Delta modulator primitive library

first generation in GA. Each topology in the initial set is generated in 3 steps. Firstly, the number of integrators are determined by the predefined order of Sigma-Delta modulators. Each integrator type (delayed or delayless type) in the modulator is randomly defined by the system. Secondly, components in the modulators can be randomly connected by feedforward and feedback paths. Finally, all the coefficients in the generated topology are assigned random initial values. Subsequently, the SystemC-A model is automatically generated according to this topology. Figure 4.22 shows some well-known third-order single-loop Sigma-Delta modulator topologies [122] that can be generated in the design initialisation phase.

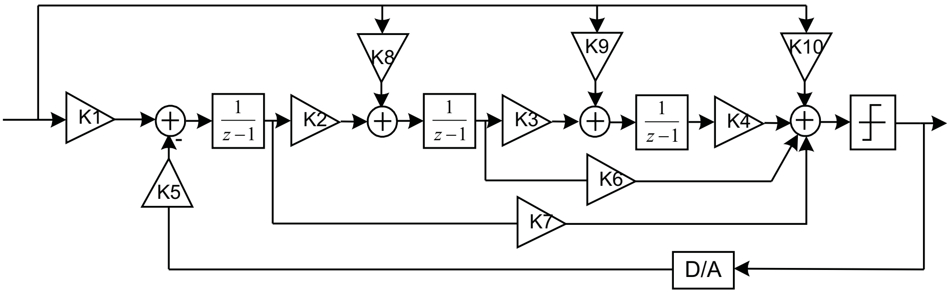
4.3.2 Genetic approach to synthesis

The performance of each design in the initial set is evaluated by the evaluation engine which measures the power spectrum density (PSD) and signal-to-noise ratio (SNR) through FFT analysis of the Sigma-Delta modulator output bitstream.

After evaluating the initial designs, selection and new generation reproduction processes are applied to the current generation parents to breed the new generation. This genetic-based synthesis process is similar to that used in electronic loop filter design. In the selection operation, a proportion of designs with better performance are retained. In the reproduction phase, the standard genetic operations of crossover and mutation are applied to the selected designs to generate the new generation. Firstly, if the crossover operation is triggered (crossover probability of two parents exceeds a fixed threshold),



(a) Integrators with distributed feedback and input coupling (CIFB structure)



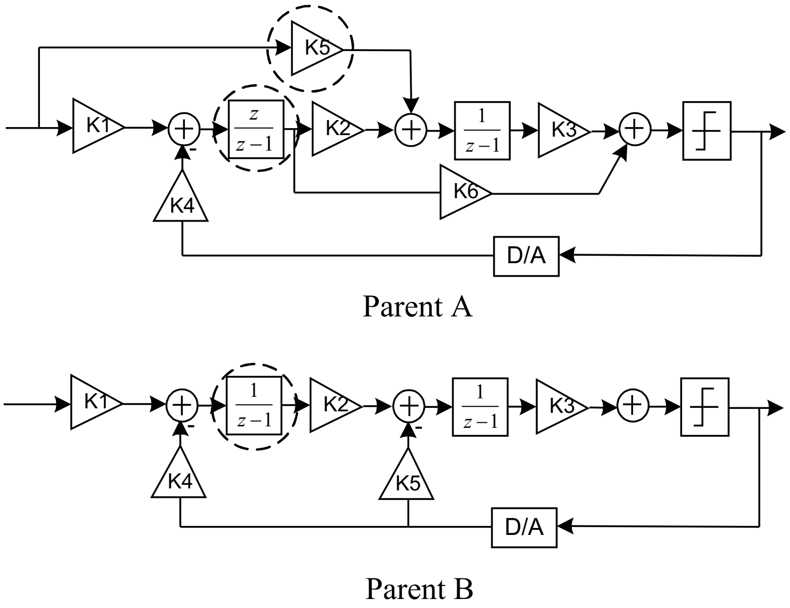
(b) Chain of integrators with feedforward signal paths (CIFF structure)

Figure 4.22: Examples of single-loop Sigma-Delta modulator topologies generated by combing primitives in the library

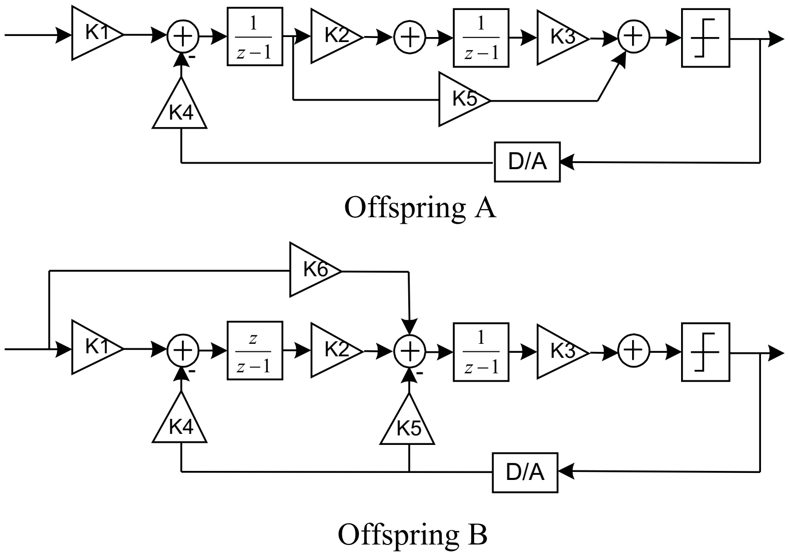
new offspring can be generated by exchanging components of selected parents such as the signal paths and integrators.

An example of crossover operation is illustrated in the Figure 4.23. As shown in the figure, the crossover probabilities of the first integrator and the feedforward signal path from the input to second integrator are higher than the trigger probability in this example. Thus, these two components of parents A and B are exchanged, leaving the other components unchanged to get new offspring.

The mutation operation contains two phases: topology mutation and component's coefficient mutation. In the first phase, if the topology mutation probability is higher than the fixed trigger, a new topology is automatically generated from the Sigma-Delta modulator primitive library, and each parameter in the generated topology obtains a random value within range as illustrated in the design initialisation phase. If there is no topology mutation for the selected design, the parameter of each component in the design, such as the signal path gain, has a chance to mutate while maintaining an unchanged topology.



(a) Selected parents for crossover operation



(b) Generated offspring after crossover operation

FIGURE 4.23: An example of crossover operation for Sigma-Delta modulator synthesis showing two parents A and B and the corresponding offspring

4.3.3 Synthesis experiments

In this section, automated synthesis of a third-order single-loop Sigma-Delta modulator is used as a case study to demonstrate the practical operation of the proposed approach. The synthesised results are compared with a traditional modulator [149, 150].

The synthesis of a third-order Sigma-Delta modulator is demonstrated by two experiments as shown in Table 4.10. In experiment 1, the topology is synthesised for maximum SNR, and in experiment 2 - for minimum complexity(minimum signal-path).

	Design objective	***Performance constraints***	***Objective Reference***
1	Maximum SNR	SNR≥110dB No.of signal path≤15	SNR=110dB
2	Minimum signal path	SNR≥110dB No.of signal path≤12	No.of signal path=12

Table 4.10: Synthesis experiments

The synthesis process was carried out using the following design parameters:

1) Oversampling ratio: OSR=128

2) Bandwidth: $f_0 = 20KHz$

3) Oversampling frequency: $f_s = 5.12MHz$

4) Input voltage: Sinusoidal voltage with 10KHz frequency and 1V amplitude ($V_{in} = 1V$, $f_{in} = 10KHz$)

5) Order of Sigma-Delta modulator: N=3

The fitness functions for these two experiments are given by:

Experiment 1:

$$Fitness = w\frac{SNR}{SNR'} \tag{4.8}$$

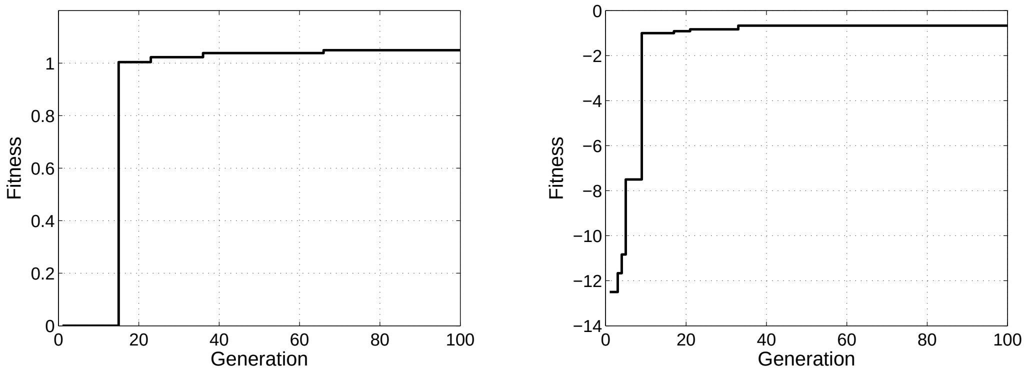
where SNR' is the objective reference value ($SNR' = 110dB$). w is set to 1 if all user defined performance constraints are met, otherwise w is set to 0.0001. For example, if a synthesised topology can achieve 110dB SNR with less than 15 signal paths, w will equal 1, meaning the algorithm has found a feasible solution.

Experiment 2:

$$Fitness = w\frac{NPath}{NPath'} \tag{4.9}$$

$NPath$ is the number of signal paths in the synthesised structure. In order to maximise the fitness parameter, w is set to -1 if the performance constraints are met or to -10 otherwise.

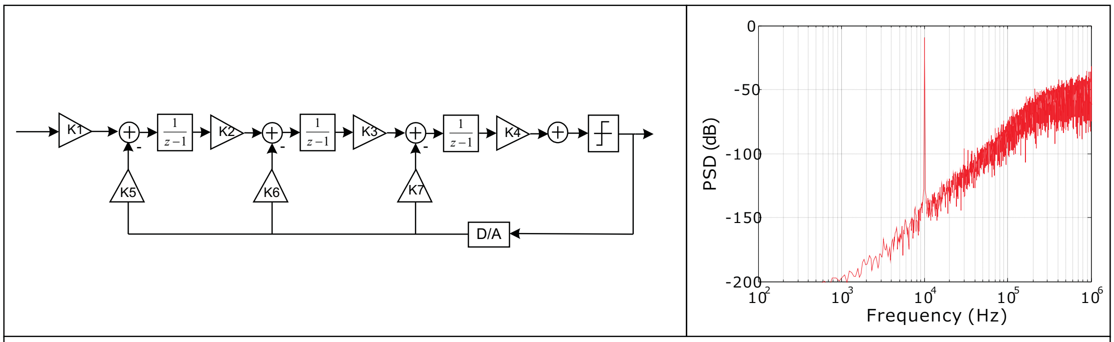
The fitness improvement during the synthesis flow is shown in Figure 4.24. It is clear that the synthesis approach finds a feasible solution and then further explores the design space to approach the optimal solution.



(a) Experiment 1:Maximum SNR (b) Experiment 2: Minimum Signal Paths

FIGURE 4.24: Fitness improvement between generations

The synthesised Sigma-Delta modulator and their associated PSD, which is derived from the FFT of output bitstream, are shown in Figure 4.25. The traditional third-order Sigma-Delta modulator [149, 150] is also plotted for comparison. It is obvious that the noise floor in synthesised modulator of experiment 1 can be reduced further leading to about a 12dB improvement of the SNR comparing with the traditional modulator. In experiment 2, the synthesis approach is used to explore design space to find the topology, which has minimum number of signal paths while SNR is maintained above 110dB. As shown in the synthesised result, a modulator with 8 signal paths achieves the design specifications. Although this topology contains 1 more signal path as compared with the traditional one, it achieves around 9dB improvement of SNR. Figure 4.26. plots the SNR curves of the synthesised and traditional modulators. As shown in the figure, the synthesised solutions achieve better dynamic range (the input amplitude achieves zero-crossing SNR). As illustrated in the experimental results, the proposed approach realised automated topology synthesis of Sigma-Delta modulator according to

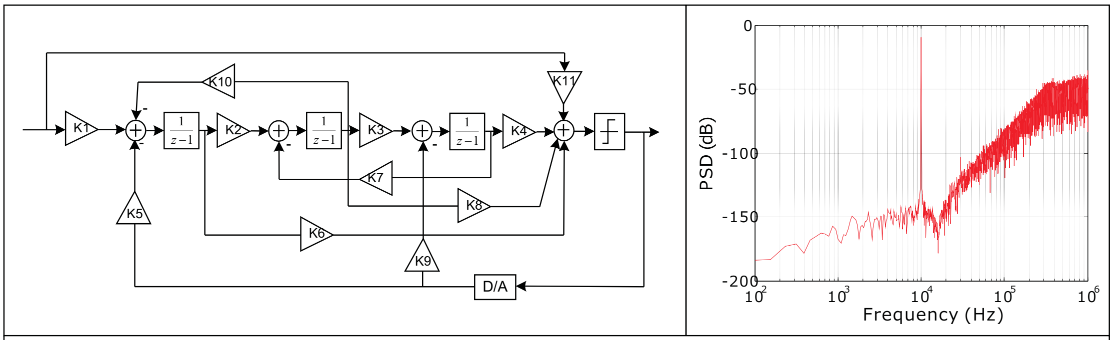


K1=1 K2=0.5 K3=0.3 K4=0.4 K5=1 K6=1 K7=1

System Performance

SNR=103.5dB DR=120dB Number of Signal paths=7

(a) Traditional 3rd order Sigma-Delta modulator

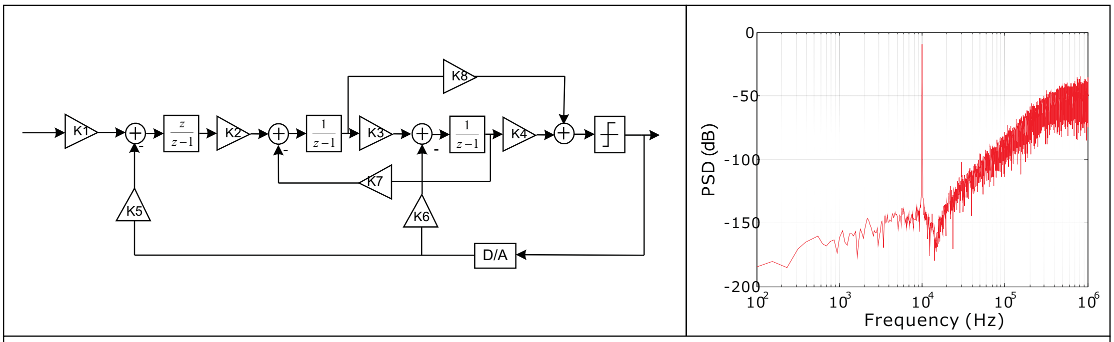


K1= 0.45 K2= 0.84 K3=0.32 K4=0.3 K5=0.45 K6=0.24 K7=1.06e-3 K8=0.42 K9=1.1 K10=5.6e-5 K11=1.8

System Performance

SNR=115.3dB DR=136dB Number of Signal paths=11

(b) Synthesised result of experiment 1 (Maximum SNR)



K1= 0.54 K2=0.714 K3=0.348 K4=0.352 K5=0.545 K6=1.173 K7=9.38e-4 K8=0.539

System Performance

SNR=112.1dB DR=126dB Number of Signal paths=8

(c) Synthesised result of experiment 2(Minimum signal paths)

Figure 4.25: Synthesised and traditional third-order Sigma-Delta modulator topologies

user-defined design specifications and constraints. The coefficients of the topology are also optimised simultaneously.

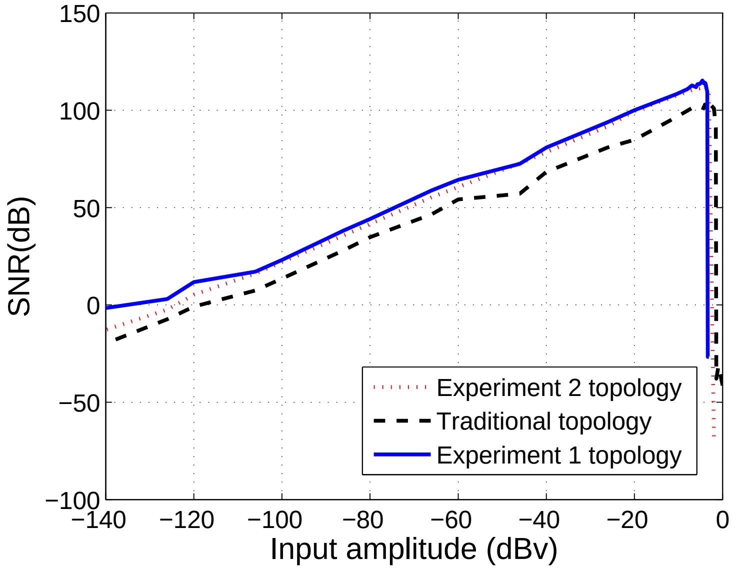


Figure 4.26: SNR curves for the synthesised and traditional Sigma-Delta structures

4.4 Concluding remarks

This chapter presents an effective simulation-based synthesis flow for automated synthesis of MEMS sensors with associated high-order electronic Sigma-Delta control systems. Design of such MEMS systems is notoriously difficult using traditional methods as the mechanical element forms an integral part of the electromechanical Sigma-Delta control system. The performance of the system is not only determined by the electronic control system configuration, but also by the dynamics of the mechanical sensing element. The proposed holistic synthesis approach, implemented in SystemC and named SystemC-AGNES, automates both the layout synthesis of the sensor's mechanical part and the configuration synthesis of the electronic control loop by simultaneously searching for the optimal solution according to user-defined constraints. It especially worth noting that the noise floor in synthesised higher order electromechanical Sigma-Delta modulators can be reduced drastically, by about 20dB and 74dB, compared with the high-order manual design and second-order design respectively.

Chapter 5

An extension to SystemC-A to support mixed-technology systems with distributed components

Although major AMS HDLs, such as SystemC-A and VHDL-AMS, are very powerful and flexible mixed physical domain modeling tools, they face a challenge in modelling mixed-technology microsystem applications such as energy harvesting systems and MEMS sensors. This is because current HDLs only support ODAEs modelling. This limits accurate modelling of systems with distributed effects (mechanical [20], electromagnetic(EM) [151], thermal [152, 153], etc.) which cannot be neglected and may even play vital roles. For example, electromechanical Sigma-Delta MEMS sensor designs, e.g. accelerometers and gyroscopes, which are based on the incorporation of mechanical sensing elements into Sigma-Delta modulator control loops, have attracted great research interest [118]. The mechanical sensing element, which is usually modelled by the lumped mass-spring-damper model (a second order ODE), is also a part of the loop filter in these systems. However, the lumped model only can capture the first resonant mode which is not accurate enough as higher order mechanical resonant modes may significantly affect the performance and stability of the Sigma-Delta control loop [20]. Consequently, it is necessary to improve the accuracy of the mechanical model and use partial rather than ordinary differential equations.

Some attempts have already been made to implement PDEs within the existing limits of major AMS HDLs [154, 155, 152]. Among them, a proposal for syntax extension to VHDL-AMS (named VHDL-AMSP) has been presented [154]. Pending the development of a new standard, a preprocessor has been developed to convert VHDL-AMSP into the existing VHDL-AMS 1076.1 standard automatically which can be simulated using currently available simulators. In this chapter, we propose the first full implementation of the PDE extension to SystemC-A where no preprocessor is required.

This chapter is organised as follows. Section 5.1 outlines the SystemC-A syntax extension and implementation. Two typical case studies, a distributed transmission line and a MEMS cantilever, are used to illustrate modelling capabilities offered by the proposed extended syntax of SystemC-A in sections 5.2 and 5.3. Finally, Section 5.4 draws conclusions from this work.

5.1 SystemC-A syntax extension and implementation

This section describes the new syntax of SystemC-A with which users can define PDEs. The abstract base class for PDEs is derived from the existing SystemC-A abstract base class *sc_a_component*. Both PDEs and their boundary conditions are generated from the new abstract base class *sc_a_PDE_base* in the proposed extension. This new abstract base class also inherits the virtual *build* method which is invoked by the SystemC-A analogue kernel at each time step to build the system matrix from contributions of all the components. A sample component class hierarchy with PDE extension is shown in Figure 5.1. The mechanical component in this example includes user-defined PDEs and associated boundary conditions which are derived from the PDE base class.

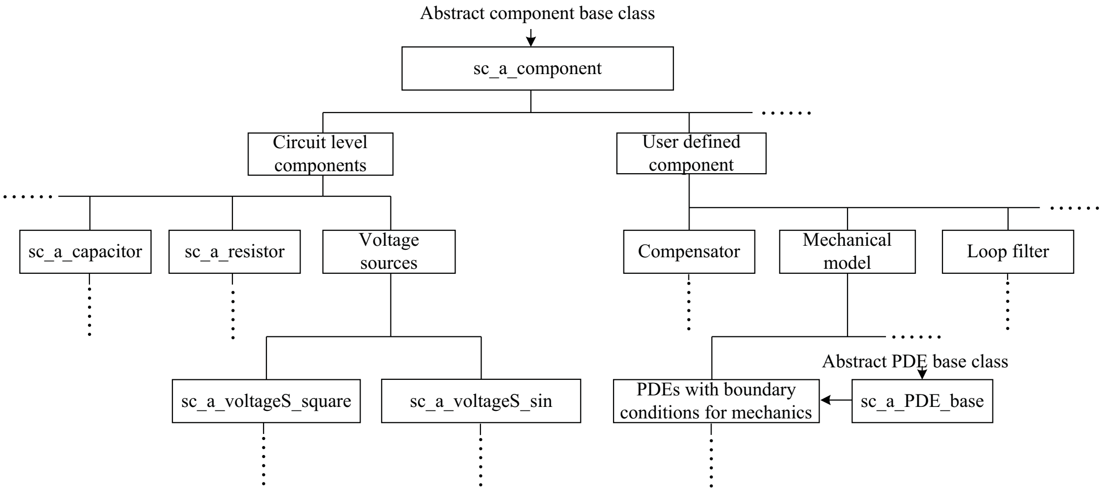


FIGURE 5.1: SystemC-A component hierarchy with PDE extensions.

A finite difference approximation approach is used to discretise the PDEs with respect to spatial variables and leave the time derivatives unchanged (as discussed in section 3.2). Consequently, PDEs are converted to a series of ODEs which can be handled by the existing SystemC-A analogue solver. The modelling flow in SystemC-A with PDE extensions is shown in the Figure 5.2.

The following example of a simple one dimensional PDE demonstrates the new syntax:

$$\frac{\partial Q(x,t)}{\partial x} + A\frac{\partial Q(x,t)}{\partial t} = B \tag{5.1}$$

Let the boundary condition be:

$$\frac{\partial^{M+N} Q(x,t)}{\partial x^M \partial t^N} = C; \tag{5.2}$$

where $Q(x,t)$ is the partial quantity of interest, A is the parameter, B is the excitation, C is the right hand side value of the boundary condition equation, x is a spatial variable and t is time.

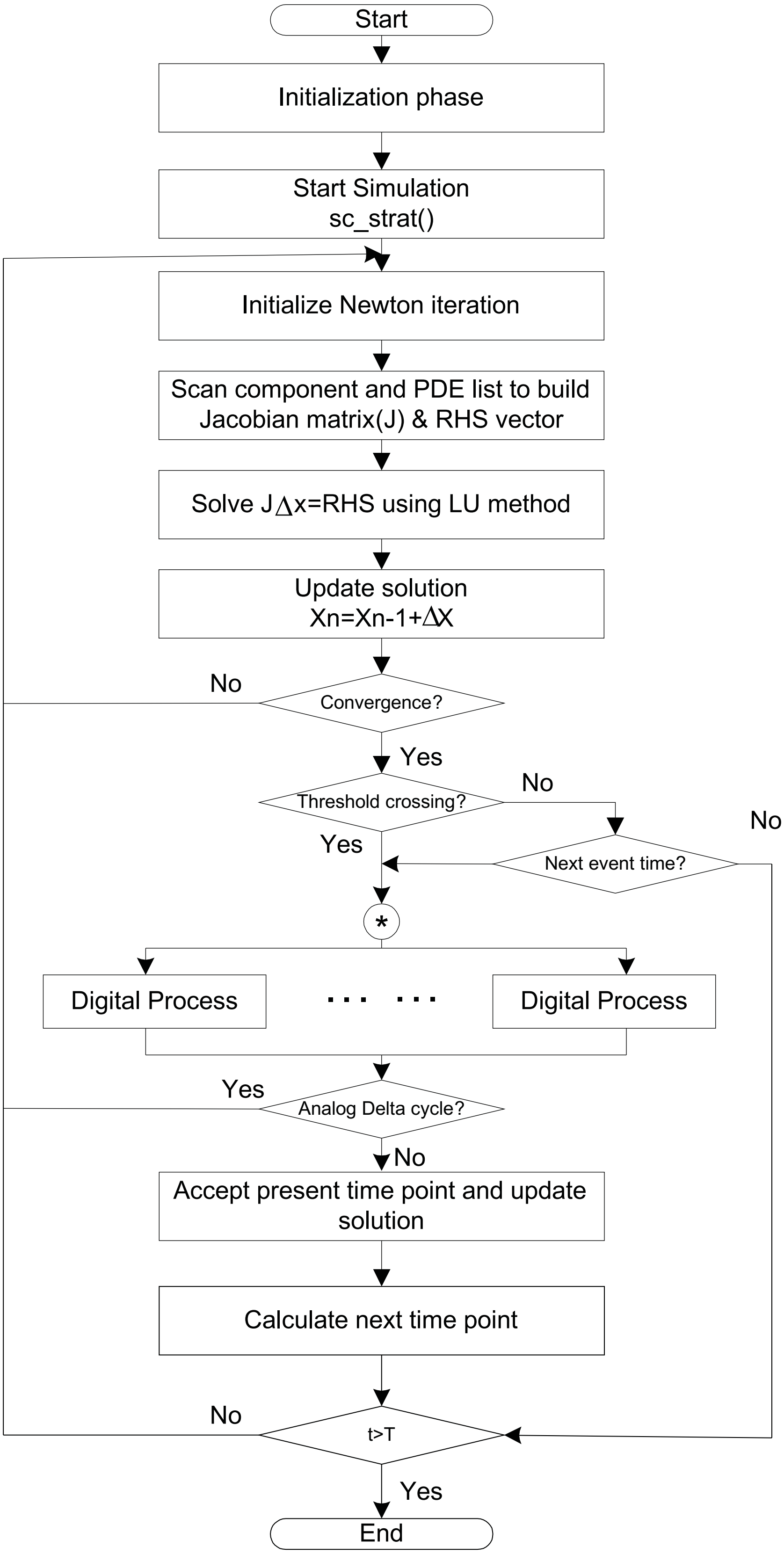


FIGURE 5.2: Simulation cycle with PDE modelling in extended SystemC-A.

The extended SystemC-A code for this example is:

```
sc_a_PDE_example::sc_a_PDE_example{
 sc_a_PDE_base("PDE_example"){
 PDE_Coordinate_Declaration("Q","x",R,N_node,dx);
 }
}

void sc_a_PDE_example::Build{
 Pdxdt_Boundary(M,N,"Q", x, C);
 for(x=1;x<=N_node;x++)
{
  PDE(x,-Pdx(1,"Q",x)-A*Pdt(1,"Q",x)+B);
}
}
```

Listing 5.1: Extended SystemC-A code for a example

5.1.1 Spatial Coordinate and Partial Quantity

Currently, in SystemC-A, three types of analogue system variables (node, flow, quantity), which are derived from an abstract base class called sc_a_system_variable, are defined. In the proposed PDEs extension, a new type of system variable (Partial Quantity), which is also derived from the abstract base class, is defined as illustrated in Figure 5.3.

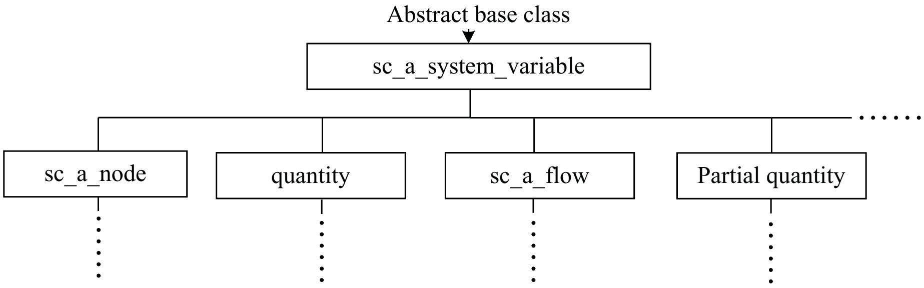


Figure 5.3: Analogue system variables

The method *PDE_coordinate_Declaration*() is used for partial quantity definition and spatial coordinate declaration. Multiple coordinate declarations will form a hypercube in the multi-dimensional space. As shown in the example code, the spatial coordinate "x" with the range R is divided into N_node segments, and the partial quantity "Q" is discretised and defined inside the function in array format and discretisation step size dx(R/N_node) is returned.

The method $PDE\_Quantity()$ is used to read a value of a particular partial quantity. For example, $PDE\_Quantity("Q", x)$ returns the value of Partial Quantity Q at node x. This function's counterpart in SystemC-A is X() which reads the value of a quantity. Similar to the differentiator ($Xdot()$) and integrator ($INTEG()$) operators which can be performed on ordinary quantities, the new methods ($PDE\_Quantity\_dot()$ and $PDE\_Quantity\_INTEG()$) allow performing these two operators on partial quantities.

5.1.2 Partial Derivatives

If "Q" is a partial quantity, the function Pdx(N,"Q",x) represents the derivative of "Q" with respect to spatial coordinative at position x. N is an integral number which represents the derivative order. For example, Pdx(4,"Q",x) represents the 4th order partial derivative$\frac{\partial^4 Q}{\partial x^4}$. A partial quantity can also have a derivative with respect to time, using the attribute dt, so item Pdxdt(3,2,"Q",x) represents $\frac{\partial Q^5}{\partial^3 x \partial^2 t}$.

5.1.3 Boundary Conditions

Boundary condition can be declared by method $Pdxdt\_Boundary()$. As shown in the example code above, M and N determine the order of the derivative with respect to coordinative x and time t, x is the specified position where the conditions should apply and C is the right hand side value of the boundary condition equation. As an example, $Pdxdt\_Boundary(1, 0, "Q", 100.0, 0.0)$ represents the first order derivative of Q at the user specified spatial boundary (x=100.0) is equal to 0.0.

5.1.4 PDE Formulation Method

Function PDE() realizes the automatic equation formulation of the PDEs to be modelled. This function is required to be implemented in a "for" loop and the number of loops determined by the number of segments(N_node). After providing the RHS vector in the 2nd term of the PDE() function, Jacobian matrix will be automatically generated using a secant finite difference approximation which is defined in terms of system RHS ($f_i(x_j)$) and a scalar Δx:

$$J_{ij} = \frac{\partial f_i}{\partial x_j} = \frac{f_i(x_j + \Delta x_j) - f_i(x_j)}{\Delta x_j} \tag{5.3}$$

Finally, the function matrix is solved in the embedded SystemC-A analog solver.

5.2 Case study 1: Distributed lossy transmission line

In the first case study, a distributed lossy transmission line is used. Many methods for analysing the transient response of the transmission line have been developed due to the increasing demand of processing and transmitting more information at faster rates, which results in a more significant role played by the transmission line in high speed circuits and systems. One of the important approximation techniques for transmission line modelling is the Finite Difference, Time-Domain method or FDTD [156]. This method discretises the telegrapher's equations both in time and space and the resulting difference equations are solved using the leap-frog scheme. Another popular method, lumped approximation method, uses a number of lumped RLCG elements to approximate the distributed lines and then performs the analysis using conventional circuit simulators like SPICE [157]. This approximation method is also implemented in VHDL-AMS [158].

The transmission line structure considered in this section is a single microstrip (see Figure 5.4) which is one of the most common types of communication in modern high speed board layout. The single microstrip is a single piece of copper placed on top of a dielectric material mounted on a ground plane. As shown in Figure 5.4, W, T and L_{tline} are width, thickness and length of the microstrip respectively, while H is the thickness of the substrate dielectric.

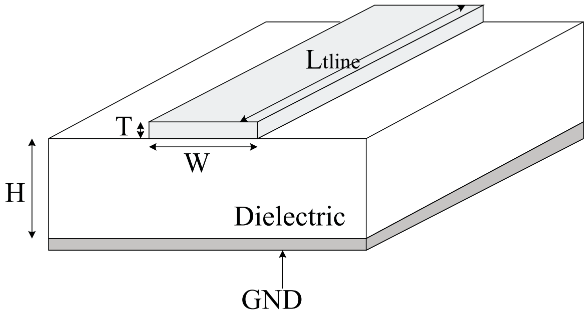


Figure 5.4: Lossy microstrip transmission line on a dielectric above ground

The following equations are the well-known governing equations of the lossy microstrip. The equations describe the voltage and current on the microstrip with distance and time.

$$-\frac{\partial^2 V(x,t)}{\partial x^2} = LC\frac{\partial^2 V(x,t)}{\partial t^2} + (RC+GL)\frac{\partial V(x,t)}{\partial t} + GRV(x,t) \tag{5.4}$$

$$-\frac{\partial^2 I(x,t)}{\partial x^2} = LC\frac{\partial^2 I(x,t)}{\partial t^2} + (RC+GL)\frac{\partial I(x,t)}{\partial t} + GRI(x,t) \tag{5.5}$$

In the equations above, x varies from 0 to L_{tline} (Length of the line). V(x,t) and I(x,t) are potential and current at position x of the microstrip at a certain time respectively. Parameters R, L, G and C represent resistance, inductance, conductance and capacitance per unit length and are related to the dimension and characteristics of the microstrip. For simplicity, the medium is assumed to be linear and homogeneous. Therefore, all the parameters of the microstrip are assumed to be constant.

Figure 5.5 shows a microstrip connected to a circuit. The microstrip is an integrated circuit interconnector. The signal propagation and losses along the microstrip are described by the PDEs in Equation 5.4 and 5.5. The interaction terminals between the

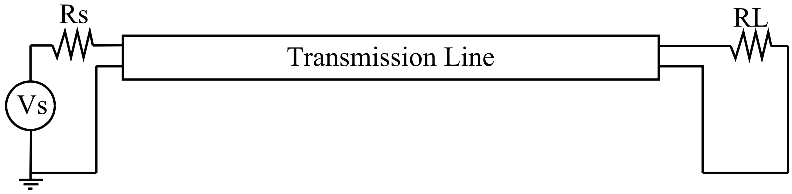


FIGURE 5.5: Transmission line with a circuit

microstrip and the circuit provide the following boundary conditions:

$$V_{in}(t) = V_s - R_s I_{in}(t) \tag{5.6}$$

$$V_{out}(t) = R_L I_{out}(t) \tag{5.7}$$

where V_{in} and V_{out} are the input and output voltages of the transmission line respectively. I_{in} and I_{out} are the input and output currents. V_s is the source voltage, R_s is the source resistance and R_L is the load resistance.

5.2.1 SystemC-A implementation of distributed microstrip transmission line

The SystemC-A model of the distributed microstrip transmission line present below provides an example of how the PDEs discussed above are implemented.

```
sc_a_Transmission_line::sc_a_Transmission_line (
 char nameC[5],sc_signal<double>*Vinput){
 PDEbase("Tline"){
      Vs_sig=Vinput;
   PDE_Coordinate_Declration("V","x",0.1,20,dx);
   PDE_Coordinate_Declration("I","x",0.1,20,dx);

 //Transmission Line parameters
   Length=0.1;      //Length of microstrip;
        .
        .
        .
```

```
 }
}
void sc_a_Transmission_line::Build{
      Vs=Vs_sig->read();
  //---Boundary condition--------//
      Pdxdt_Boundary(0,0,"V",0,Vs-PDE_quantity("I",0)*Rs);
      Pdxdt_Boundary(0,0,"V",Length,PDE_quantity("I",N_node)*RL);
      Pdxdt_Boundary(0,0,"I",0,(Vs-PDE_quantity("V",0))/Rs);
      Pdxdt_Boundary(0,0,"I",Length,PDE_quantity("V",N_node)/RL);
  //----PDEs--------------------//
   for(x=1;x<N_node+1;x++)
    {
       PDE(x,-Pdx(2,"V",x)+L*C*Pdt(2,"V",x)+
             (R*C+G*L)*Pdt(1,"V",x)+G*R*Pdx(0,"V",x));
       PDE(x,-Pdx(2,"I",x)+L*C*Pdt(2,"I",x)+
             (R*C+G*L)*Pdt(1,"I",x)+G*R*Pdx(0,"I",x));
    }
}
```

Listing 5.2: SystemC-A constructor of Transmission Line

5.2.2 Simulation Results

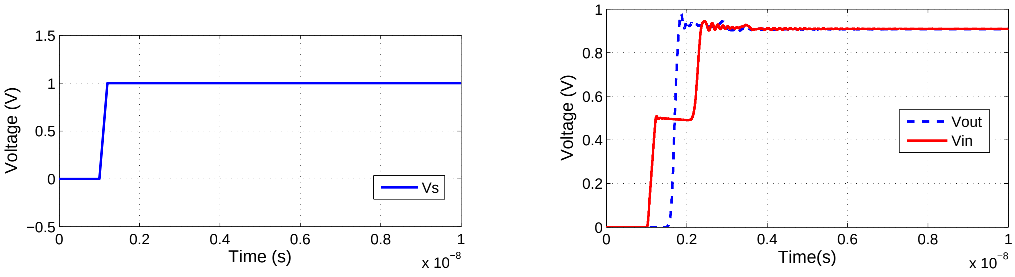
In this case study, the dimensions of the microstrip are: $W = 1.5mm$, $t = 0.035mm$, $H = 0.8mm$ and $L_{tline} = 0.1m$. The dielectric constant ε_r is 4.2. The per unit length parameters of the microstrip are $R = 18\Omega/m$, $L = 297nH/m$, $C = 115pF/m$, and $G = 0.02S/m$. These parameters are obtained by a field solver [159] according to the dimensions of the microstrip. The calculated delay of the microstrip (T_d) is about 0.53ns, characteristic impedance of microstrip (Z_0) is about 50Ω, and the effective dielectric constant ε_{eff} is about 3.0.

If the microstrip is connected to a circuit as shown in Figure 5.5, the voltage wave front will be reflected at the end of the transmission line. The reflection coefficient (ρ) is given by [160]:

$$\rho = \frac{R_L - Z_0}{R_L + Z_0} \tag{5.8}$$

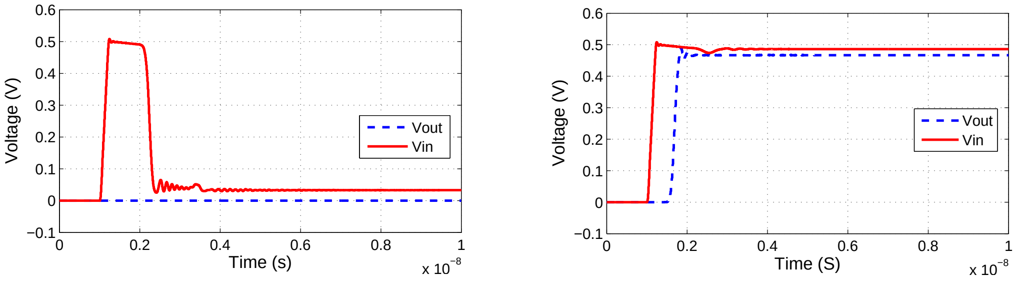
where R_L is the load resistance and Z_0 is the characteristic impedance.

Firstly, a step voltage source with 1V step voltage and 0.2ns rise time is used in the simulations to illustrate the behaviour of the microstrip model. Three special cases are chosen: open circuit, short circuit and matched circuit ($R_s = R_L = Z_0$). The simulation results are shown in Figure 5.6.



(a) Input Voltage

(b) Open circuit at the load



(c) Short circuit at the load

(d) Matched circuit

FIGURE 5.6: Simulation result of the transmission line model

Consider open circuit at the load, at time $1.55ns$, the voltage wave reaches load end and doubled wave travels back to the source end as the reflection coefficient is $\rho = 1$ ($R_L \gg Z_0$). At time $2.1ns$, the doubled voltage wave reaches the source as shown in Figure 5.6(b). In the case of the short circuit, the reflection coefficient $\rho = -1$ ($R_L \ll Z_0$), the reflected voltage reaches the source and leads to the drop of the voltage (Figure 5.6(c)). In the case of the matched circuit, no reflection occurs as $R_L = Z_0$ (Figure 5.6(d)). In all three cases, the loss of the microstrip is captured due to the R,G terms in the Equation 5.4.

Secondly, a 3GHz 1V sine wave stimulus is applied to the matched circuit and the simulation result is shown in Figure 5.7. The loss in the line is about 8 percent of the input signal. The wavelength of the signal is given by:

$$\lambda = \frac{V}{f} = \frac{C_0}{\sqrt{\varepsilon_{eff}} f} \tag{5.9}$$

where V is the wave's propagation speed(also known as phase velocity) at which the wave is propagating along the microstrip, C_0 is the speed of light in vacuum, ε_{eff} is the effective dielectric constant in the medium and f is the input frequency. The wavelength is about 0.058 meter according to Equation 5.9. The voltages along the line at a certain point in time($t = 3\mu s$) is shown in Figure 5.8. A lossy sine wave with a wavelength of about 0.06m is correctly captured.

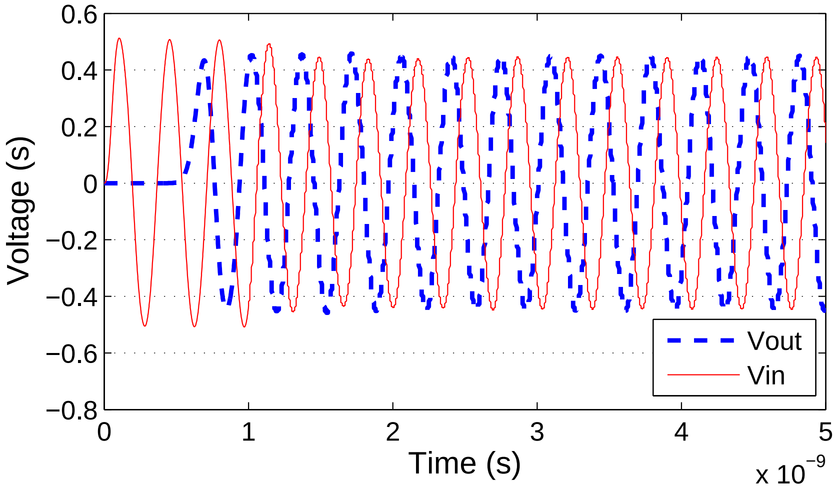


FIGURE 5.7: Simulation result of the transmission line with 3GHz 1V sine wave source voltage

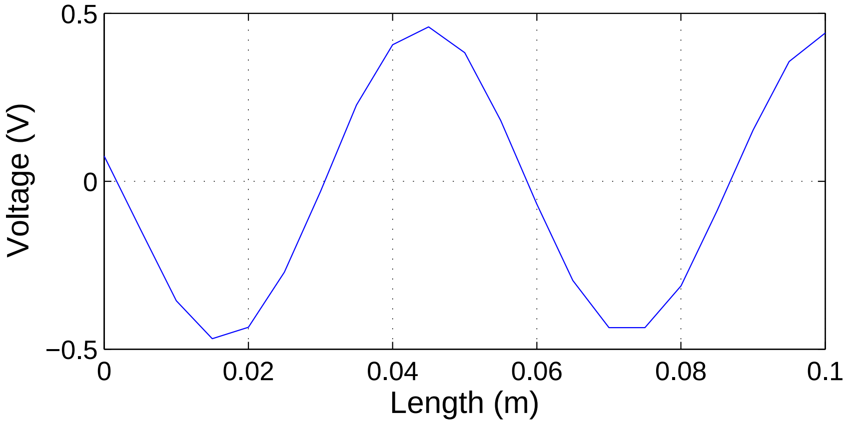


FIGURE 5.8: Voltage along the transmission line at time t=$3\mu s$

5.3 Case study 2: MEMS cantilever beam

Cantilever beams are the most ubiquitous structures in MEMS systems. The governing PDE of the motion of the cantilever beam is described as [161]:

$$\rho S_b \frac{\partial^2 y(x,t)}{\partial t^2} + C_D I \frac{\partial^5 y(x,t)}{\partial x^4 \partial t} + EI \frac{\partial^4 y(x,t)}{\partial x^4} = F(x,t) \tag{5.10}$$

where $y(x,t)$ is a function of time and position that represents the deflection of the beam. E, I, ρ, S_b are the physical properties of the beam: ρ is the material density, S_b is the cross sectional area $(W_b * T_b)$, W_b and T_b are the width and thickness of the beam, E represents the Young's modulus which defines a material's shearing strength, I is the second moment of area which could be calculated by $I = W_b T_b^3/12$, EI is usually regarded as the flexural stiffness, C_D is the internal damping modulus, $F(x,t)$ is the distributed force along the beam. In this case study, a force $(F_{in}(t))$ is applied to the free end of the cantilever beam (Figure 5.9).

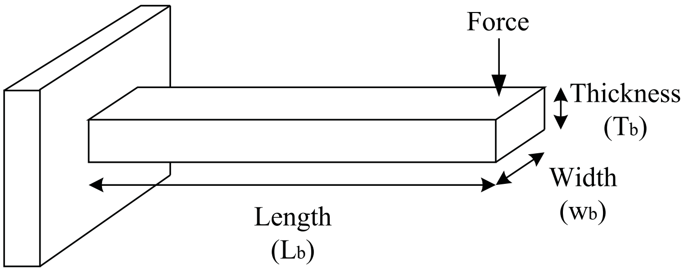


Figure 5.9: Structure of cantilever beam

The boundary conditions of the structure can be described by the following equations:

$$y(0,t) = 0; \tag{5.11}$$

$$\frac{\partial y(0,t)}{\partial x} = 0 \tag{5.12}$$

$$\frac{\partial^2 y(L,t)}{\partial x^2} = 0 \tag{5.13}$$

$$\frac{\partial^3 y(L,t)}{\partial x^3} = 0 \tag{5.14}$$

5.3.1 SystemC-A implementation of cantilever beam

The SystemC-A model of the cantilever beam is shown below:

```
sc_a_Cantilever::sc_a_Cantilever (
 char nameC[5],sc_signal<double>*Fsource){
  PDEbase("Cantilever"){
      Fin_sig=Fsource;
      PDE_Coordinate_Declaration("Y","x",150e-6,10,dx);
 //Cantilever beam parameters
   Wb=1e-6;    //Width of beam;
   Tb=1e-6;        //Thickness of beam;
      . . .
  }
 void sc_a_Cantilever::Build{
   //---Input Force------//
    Fin=Fin_sig->read();
   //---Boundary conditions--------//
    Pdxdt_Boundary(0,0,0,0);
    Pdxdt_Boundary(1,0,0,0);
    Pdxdt_Boundary(2,0,150e-6,0);
    Pdxdt_Boundary(3,0,150e-6,0);
 //-----Partial Differential Equations------------//
 //PDE function

   for(x=1;x<N_node+1;x++)
   {
    If(x==N_node)
    F=Fin;
    else F=0;
    PDE(x,-1*ro*A*Pdxdt(0,2,"y",x)-CD*I*Pdxdt(4,1,"y",x)
          -E*I*Pdxdt(4,0,"y",x)+F);
   }
 }
```

LISTING 5.3: SystemC-A constructor of Cantilever beam

5.3.2 Simulation Results

The dimensions of the cantilever beam in this case study are: $L_b = 150\mu m$, width $W_b = 1\mu m$, and thickness $T_b = 1\mu m$. The first two resonant frequencies of the cantilever beam could be calculated based on the equation below:

$$\omega = \alpha_{1,2}^2 \frac{T_b}{L^2}\sqrt{\frac{E}{12\rho}} \tag{5.15}$$

$$\alpha_1 = 1.875, \alpha_2 = 4.694$$

The first and the second resonant frequencies are 54KHz and 338KHz according to the equation above.

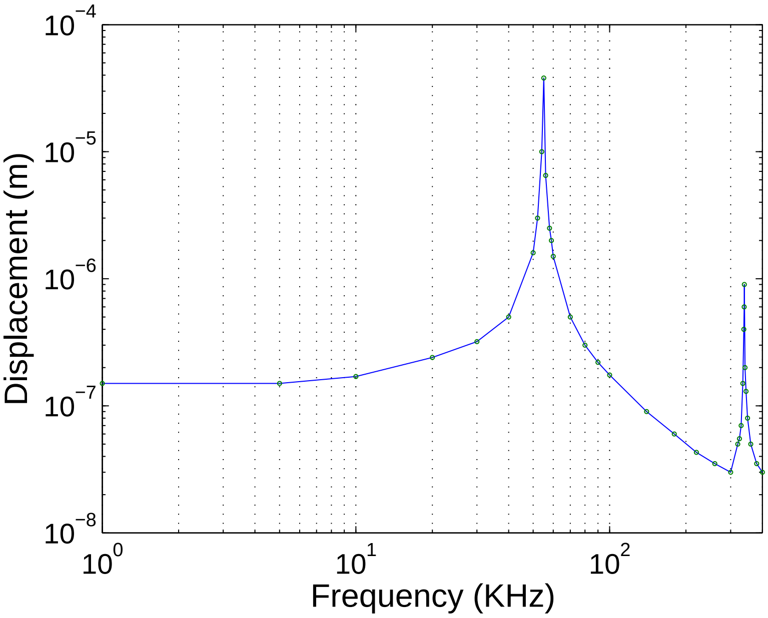
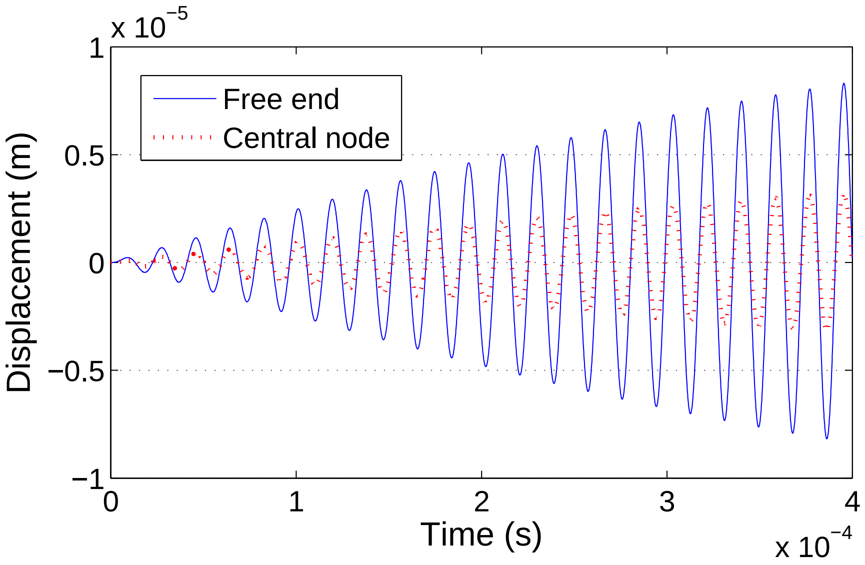
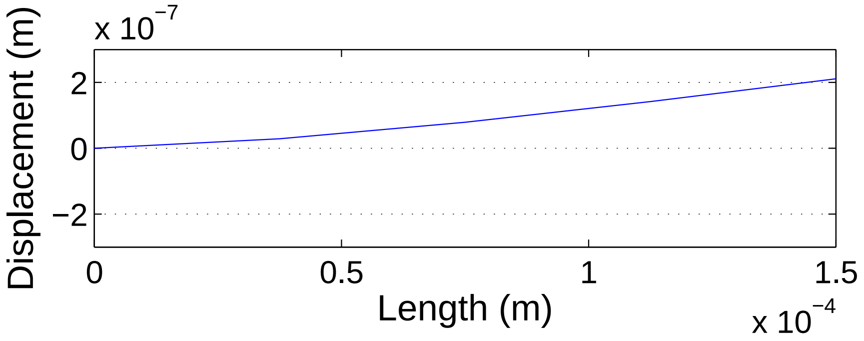


FIGURE 5.10: Displacement of the free end of beam with varying excitation frequencies (SystemC-A)

The cantilever beam model, which is excited by a sinusoidal force, is simulated to verify the behaviour of the distributed model. Figure 5.10 shows the magnitude of displacement of the free end of the beam derived from a series of transient simulations of the SystemC-A model with varying excitation frequencies. The displacements are small at low frequencies and become large at frequencies near to the resonance frequencies. The simulation results are consistent with the calculations. Figure 5.11 and Figure 5.12 show the transient simulation results of the cantilever beam model excited by sinusoidal input forces with 54KHz and 338KHz frequencies. The shapes of the cantilever beam at a certain time point ($t = 0.6\mu s$) are also plotted in these figures. The simulation results indicate that the cantilever beam resonance modes are correctly captured by the extended SystemC-A model.



(a) Displacement of the free end and central node(54kHz sinusoidal input force at free end)

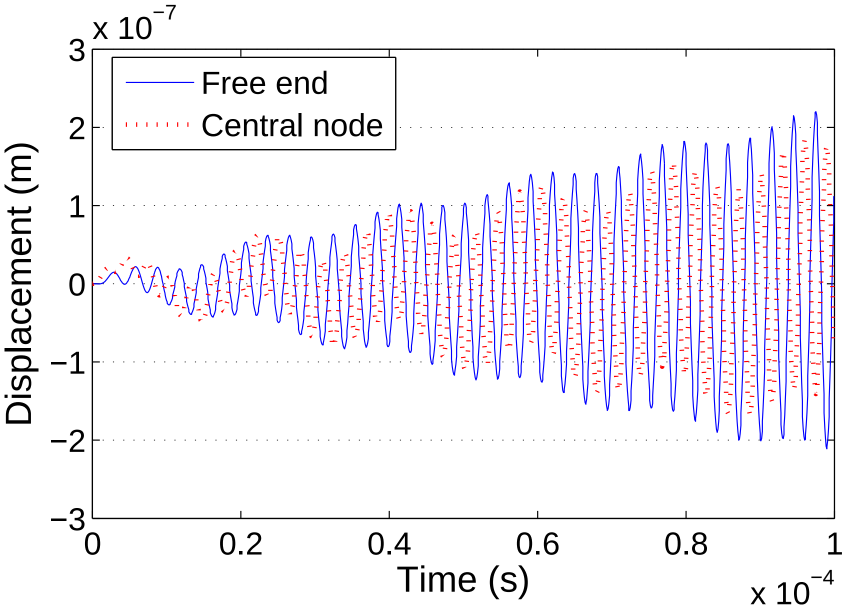


(b) Shape of cantilever beam at time 6e-5s(54kHz sinusoidal input force at free end)

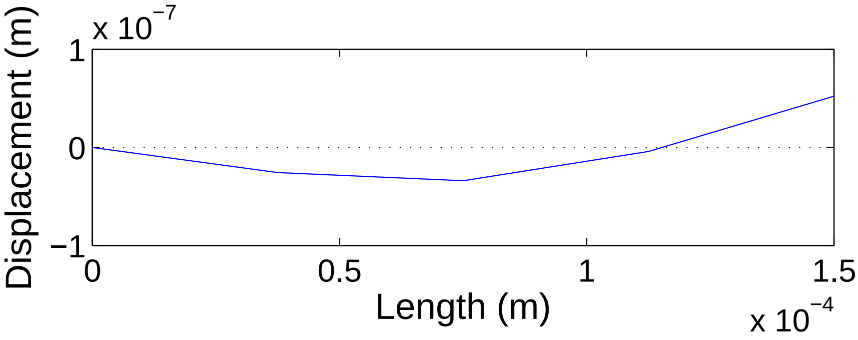
Figure 5.11: Transient simulation result of the cantilever model excited by a 54kHz sinusoidal input force.

5.4 Concluding remarks

This chapter proposes a syntax extension to SystemC-A to provide support for PDE modelling. This is the first full implementation of PDE support in SystemC-A where no preprocessor is required for conversion of user defined PDEs to a series of ODAEs. The proposed PDE extension has particular advantages in modelling of mixed physical-domain systems, especially systems with mechanical parts which exhibit distributed behaviour. The distributed effects present in such systems usually cannot be neglected,



(a) Displacement of the free end and central node(338kHz sinusoidal input force at free end)



(b) Shape of cantilever beam at time 6e-5s(338kHz sinusoidal input force at free end)

Figure 5.12: Transient simulation result of the cantilever model excited by a 338kHz sinusoidal input force.

may even play vital roles and be essential to predicting correctly the systems performance. The efficiency of the new syntax has been verified by its applications to a lossy microstrip and a MEMS cantilever. The distributed behaviour of these two case studies are correctly captured as indicated in the simulation results.

Chapter 6

Conclusions and future research

MEMS systems are currently used in a wide range of applications due to their low cost, small form factor and low power consumption. However, the design of a MEMS product is still a complex procedure which originates primarily from the interrelationships among different energy domains in a MEMS system. Although almost all MEMS devices are tightly integrated with electronics, MEMS devices and their associated ICs have traditionally been designed separated using different methodologies and different tools. The handoff between MEMS and IC designers is ad hoc and manual [2]. This conventional hybrid MEMS design approach is not well suited for meeting the cost and time-to-market demands of consumer markets. Major HDLs with AMS extension, i.e. VHDL-AMS, SystemC-A, etc, are able to deal with this problem because these HDLs support multi-energy domains modelling. Thus, MEMS design can be integrated into a single environment. This thesis presents a novel, holistic synthesis flow applied to automated layout synthesis of mechanical components of MEMS and configuration synthesis of associated electronic control system based on AMS HDLs. The next section summarises the contributions as well as proposes future work.

6.1 Thesis contributions

In Chapter 3, a MEMS case study of an surface-micromachined capacitive accelerometer operating in a Sigma-Delta force-feedback control scheme was modelled in VHDL-AMS and SystemC-A. Firstly, the mechanical sensing element of such accelerometer was modelled using the conventional approach where a second-order ordinary differential equation (ODE) is commonly used. It is well known that sense fingers in the mechanical sensing element might bend significantly or resonate, thus, leading to a failure of the electromechanical Sigma-Delta feedback control. However, as shown in the simulation results in Section 3.2, the conventional mechanical model is not accurate enough to capture the sense finger resonances. To correctly reflect the behaviour of the system, in Section 3.3, a distributed mechanical sensing element model, which includes sense finger dynamics, was proposed. The distributed model was developed by spatial descretisation of the governing partial differential equation(PDE) to obtain a series of ordinary differential equations(ODEs) using Finite Difference Approximation(FDA) approach. Simulation results showed that the distributed model correctly reflected the way in which finger dynamics affected the performance of the control loop. A comparison between VHDL-AMS and SystemC-A was provided in Section 3.4. Finally, SystemC-A was selected to implement the proposed synthesis algorithm for MEMS system because it is extremely well suited for complex modeling, implementation of post-processing of simulation results and optimisation algorithms.

In Chapter 4, a novel, holistic approach was proposed for automated optimal layout synthesis of MEMS systems embedded in electronic control circuitry from user-defined high-level performance specifications and design constraints. The synthesis technique has been implemented in SystemC-A and named SystemC-AGNES. A practical case study of an automated design of a capacitive MEMS accelerometer with high-order Sigma-Delta control demonstrated the operation of the SystemC-AGNES platform. Design of such MEMS systems is notoriously difficult using traditional methods as the mechanical element forms an integral part of the electromechanical Sigma-Delta control system. The performance of the system is not only determined by the electronic control system configuration, but also by the dynamics of the mechanical sensing element. The proposed synthesis method efficiently, and in an automated manner, generated suitable layouts of mechanical sensing elements and configurations of the Sigma-Delta control loop by combining primitive components stored in libraries, i.e. MEMS primitive library and electronic control primitive library, and simultaneously searching for the optimal solution according to user-defined constraints. It worth noting that the models in the MEMS primitive library include distributed mechanical dynamics described by PDEs that enables the performance of critical mechanical components to be accurately predicted. The synthesis results showed that the proposed technique explored the configuration space effectively and developed new Sigma-Delta structures which have not been previously investigated. The noise floors in the MEMS accelerometers synthesised by SystemC-AGNES were further reduced leading to an improvement of the SNR compared with a manually designed standard electromechanical Sigma-Delta MEMS accelerometer [118].

Current AMS HDLs, such as SystemC-A, only support ODAEs modelling. This limits the accurate modelling of mixed-technology systems with parts which frequently exhibit distributed behaviour. The distributed effects present in such systems usually cannot be neglected, may even play vital roles and be essential to correctly predicting the system's performance. Although, in Chapter 3, we proposed an approach to convert the PDEs to a set of ODEs which can be handled by VHDL-AMS and SystemC-A analogue solver, it was a tough task because it was done manually. In Chapter 5, a syntax extension to SystemC-A to provide support for PDE modelling was proposed. This is the first full implementation of PDE support in SystemC-A where no preprocessor is required for conversion of PDEs to a series of ODAEs. The efficiency of the new syntax was verified by two typical case studies: a lossy microstrip and a MEMS cantilever.

6.2 General vision for future work

The holistic technique for automated optimal synthesis of MEMS systems proposed in Chapter 4 has several areas which may be subject to further development. The synthesis approach in this research is based on a single-objective genetic algorithm. It can be further improved by applying multiple-objective genetic algorithm to obtain a global optimal solution [162].

So far, the Sigma-Delta control system for MEMS sensors considered in this research focuses on the single-stage structure. To date, some publications provide methodologies to incorporate the MEMS sensing element with a multi-stage noise shaping (MASH) Sigma-Delta modulator as closed-loop structure [129]. The multi-stage higher order Sigma-Delta modulator is constructed by cascading several low-order (first-order or second-order) single-stage Sigma-Delta modulators. The MASH architecture provides superior performance and overcomes some disadvantages encountered in the single-loop Sigma-Delta structure in terms of stability, dynamic range, and overload input level. Our holistic synthesis approach can be extended to support automated synthesis of both single-stage and multi-stage(MASH) electromechanical Sigma-Delta modulators in future work.

The MEMS and electronic control loop primitive libraries employed in Chapter 4 have a limited number of components. However, the synthesis technique presented is applicable to a wide variety of MEMS systems with electronic controls. Continuing work may focus on expanding the model library to make the design flow suitable for a larger scale of mixed-technology system designs such as MEMS gyroscopes [30] and MEMS energy harvesting systems [163].

Appendix A

Published Papers

This appendix lists all papers published during the course of this research.

1. Chenxu Zhao, Tom J. Kazmierski, "A Holistic Approach to Automated Synthesis of Mixed-technology Digital MEMS Sensors Part 1: Layout Synthesis of MEMS Component with Distributed Mechanical Dynamics", Sensors & Transducers Journal. Volume 117. pp. 1-15. ISSN 1726-5479, 2010
2. Chenxu Zhao, Tom J. Kazmierski, "A Holistic Approach to Automated Synthesis of Mixed-technology Digital MEMS Sensors Part 2: Synthesis of a MEMS System with Associated Control Loop", Sensors & Transducers Journal. Volume 117. pp. 16-28. ISSN 1726-5479, 2010
3. Chenxu Zhao, Tom J. Kazmierski, "An extension to SystemC-A for mixed-technology systems with distributed components", accepted by DATE'2011 (Design, Automation and Test in Europe) conference.
4. Chenxu Zhao, Tom J. Kazmierski, "Genetic based high-level synthesis of Sigma-Delta modulator in SystemC-A", Forum on Specification & Design Languages Conference (FDL 2010), Sep. 2010

5. Chenxu Zhao, Tom J. Kazmierski, "SystemC-A modelling of mixed-technology systems with distributed behaviour", Forum on Specification & Design Languages Conference (FDL 2010), Sep. 2010

6. Chenxu Zhao, Tom J. Kazmierski, "Genetic-based automated synthesis and optimization of MEMS accelerometers with Sigma-Delta control", IEEE International Symposium on Circuits and Systems (ISCAS 2010), May 2010

7. Chenxu Zhao, Tom J. Kazmierski, "An automated design flow for MEMS accelerometers with Sigma-Delta control". International Conference on Information and Automation (ICIA 2010), Jun. 2010

8. Chenxu Zhao, Tom J. Kazmierski," Analysis of sense finger dynamics for accurate Sigma-Delta MEMS accelerometer modelling in VHDL-AMS". In: 2009 Forum on Specification & Design Languages Conference(FDL 2009), Sep. 2009

9. Chenxu Zhao, Tom J. Kazmierski," Automated performance optimization and layout synthesis of MEMS accelerometer with Sigma-Delta force-feedback control loop", In: 2008 IEEE International Behavioral Modeling and Simulation Conference, Sep. 2008

10. Leran Wang, Chenxu Zhao, Tom J. Kazmierski, "An extension to VHDL-AMS for AMS systems with partial differential equations". In: Embedded Systems Specification and Design Languages (Selected Contributions from FDL'07), pp. 123-136, Springer. ISBN 978-1-4020-8296-2

11. Chenxu Zhao, Tom J. Kazmierski, "An efficient and accurate MEMS accelerometer model with sense finger dynamics for applications in mixed-technology control loops". In: IEEE Behavioral Modeling and Simulation Conference (BMAS 2007), September 20-21, 2007, San Jose, California, USA. pp. 143-147.

References

[1] M. Weber, M. Bellrichard, and C. Kennedy. High Angular Rate and High G Effects in the MEMS Gyro. *Sensors Magazine, (http://www.coventor.com/mems/techinfo/research.html)*, 2004.

[2] S. Breit and J. V. Kuijk. Bringing MEMS into the IC design flow. *EE Times, Design Article*, 2010.

[3] IEEE standard VHDL analog and mixed-signal extensions. *IEEE Std 1076.1*, 2004.

[4] IEEE Standard VHDL Analog and Mixed-Signal Extensions-Packages for Multiple Energy Domain Support. *IEEE Std 1076.1.1-2004, 17 June 2005, ISBN 0-7381-4646-3 SS95310.*

[5] P.R. Wilson, A.D. Brown, and H.A. Mantooth. Standard VHDL 1076.1.1 packages for multiple energy domain support. *In Behavioral Modeling and Simulation, 2003. BMAS 2003. Proceedings of the 2003 International Workshop on,*, pages 70–75, 2003.

[6] H. Al-Junaid and T. Kazmierski. Analogue and mixed-signal extension to SystemC. *IEE proc. Circuit Devices Systems*, 152(6):682–690, Dec. 2005.

[7] A. Franke J. Hauer T. Mahne, K. Kehr and B Schmidt. Creating virtual prototypes of complex micro-electro-mechanical transducers using reduced-order modelling methods and VHDL-AMS. *In Forum on Specification and Design Languages, Proceedings*, pages 209–221, 2005.

[8] J.E. Mehner G. Herrmann D. Muller M. Schlegel, F. Bennini and W. Dotzel. Analyzing and simulation of MEMS in VHDL-AMS based on reduced-order FE models. *Sensors Journal, IEEE*, 5(5):1019–1026, 2005.

[9] Datasheet. SystemVision for Mechatronic System Modeling. *Mentor Graphics Corporation*, 2006.

[10] L. Wang and T.J. Kazmierski. VHDL-AMS based genetic optimization of a fuzzy logic controller for automotive active suspension systems. *Proc. Int. Conf. Behacioral Modeling and Simulation Workshop,BMAS*, pages 124–127, Sept 2005.

[11] T. Mukherjee, Y. Zhou, and G. K.Fedder. Automated Optimal Synthesis of Microaccelerometer. *Twelfth IEEE International Conference on Micro Electro Mechanical Systems,*, pages 326–331, 1999.

[12] T. Mukherjee and G.K. Fedder. Design methodology for mixed-domain systems-on-a-chip. *VLSI '98. System Level Design. Proceedings. IEEE Computer Society Workshop on*, pages 96–101, 1998.

[13] V. Gupta. Approaches to Synthesis of a CMOS Accelerometer. *M.S. Thesis, Carnegie Mellon University, Pittsburgh, PA.*, Aug. 2002.

[14] Z. Fan, E. D.Goodman, J. Wang, and R. Rosenberg. Hierarchical Evolutionary Synthesis of MEMS. *Evolutionary Computation, 2004. CEC2004. Congress on*, 2:2320 – 2327, 2004.

[15] C.L. Cobb and A.M. Agogino. Case-Based Reasoning for the design of Mircoelectro-mechanical systems. *Proceedings of IDETC/CIE 2006 ASME 2006 International Design Engineering Technical Conferences & Computers and Information in Engineering Conference*, 2006.

[16] C.J. Shi and A. Vachoux. VHDL-AMS design objectives and rationale. *Current Issues in Electronic Modeling,Kluwer Academic Publishers*, (2):1–30, 1995.

[17] P.V. Nikitin and C.R. Shi. VHDL-AMS based modeling and simulation of mixed-technology microsystems: a tutorial. *Integration - The VLSI Journal*, (40):261–273, 2007.

[18] E. Normark P.V. Nikitin and C.J.R. Shi. Distributed electrothermal modeling in VHDL-AMS. *International Workshop on Behavioral Modeling and Simulation, (BMAS)*, pages 128–133, 2003.

[19] P. Fritzson L. Saldamli and B. Bachmann. Modelica for partial differential equations. *In 2nd International Modelica Conference, proceedings*, pages 307–314, 2002.

[20] J.I. Seeger, J. Xuesong, M. Kraft, and B.E. Boser. Sense finger dynamics in a Sigma-Delta force feedback gyroscope. *Tech. Digest of Solid State Sensor and Actuator Workshop, Hilton Head Island, USA.*, pages 296–299, Jun. 2000.

[21] K. Sharma, I. Macwan, L. Zhang, L. Hmurcik, and X. Xiong. Design optimization of MEMS comb accelerometer. *American Society for Engineering Education(ASEE) Zone 1 Conference*, Mar. 2008.

[22] C. Liu. *Foundations of MEMS.* Prentice Hall, September 5, 2005.

[23] M. Kraft. Micromachined inertial sensors: The state of the art and a look into the future. *IMC Measurement and Control*, 33(6):164 C 168, 2000.

[24] S. Lucyszyn, S. Pranonsatit, J. Y. Choi, R. W. Moseley, E. M. Yeatman, and A. S. Holmes. Novel RF MEMS Switches. *Proceedings of Asia-Pacific Microwave Conference 2007*, page 55 C 58, 2007.

[25] Joo-Young Choi, Jinyu Ruan, F. Coccetti, and S. Lucyszyn. Three-Dimensional RF MEMS Switch for Power Applications. *IEEE Transactions on Industrial Electronics,*, 56:1031 – 1039, 2009.

[26] S. Lucyszyn. RF MEMS - an introduction. *IEEE Workshop on MEMS for RF and Optoelectronics Applications,*, pages 3–26, May 2002.

[27] G. M. Rebeiz. *RF MEMS: theory, design, and technology.* John Wiley and Sons, 2003.

[28] D. Dudley, W. Duncan, and J. Slaughter. Emerging Digital Micromirror Device (DMD) Applications. *SPIE Proceedings,*, 4985, 2003.

[29] Rob OReilly, Alex Khenkin, and Kieran Harney. Sonic Nirvana: Using MEMS Accelerometers as Acoustic Pickups in Musical Instruments. *Analog Dialogue, A Forum for the exchange of circuits, systems, and software for real-world signal processing*, 43, 2009.

[30] J. Xuesong, J.I. Seeger, M. Kraft, and B.E. Boser. A Monolithic Surface Micromachined Z-Axis Gyroscope with Digital Output. *Proc. Symposium on VLSI Circuits*, pages 16–19, June 2000.

[31] G. M. Rebeiz. RF MEMS switches: status of the technology. *International Conference on Solid State Sensors, Actuators and Microsystems*, pages 1726–1729, June 2003.

[32] K.E. Petersen. Micromechanical membrane switches on silicon. *IBM Journal of Research and Development*, 23(4):376 – 385, 1979.

[33] H. U. Rahman, K. Y. Chan, and R. Ramer. Cantilever beam designs for RF MEMS switches. *Journal of Micromechanics and Microengineering*, 2010.

[34] E. R. Brown. RF-MEMS Switches for Reconfigurable Integrated Circuits. *IEEE Trans. on Microwave Theory and Techniques*, 46(11), 1998.

[35] G. M. Rebeiz and J B. Muldavin. RF MEMS switches and switch circuits. *IEEE Microwave Magazine*, 2:59–71, 2001.

[36] P.D. Grant, M.W. Denhoff, and R.R. Mansour. A Comparison between RF MEMS Switches and Semiconductor Switches. *MEMS, NANO and Smart Systems, 2004. ICMENS 2004*, pages 515–521, 2004.

[37] C. T.-C. Nguyen. Transceiver Front-End Architectures Using Vibrating Micromechanical Signal Processors. *Dig. of Papers, Topical Mtg on Silicon Monolithic ICs in RF Systems*, pages 23–32, 2001.

[38] F. D. Bannon, J. R. Clark, and C. T.C. Nguyen. High frequency micromechanical filters. *IEEE J. Solid-State Circuits*, 35(4):512–526, 2000.

[39] K. Wang, A.C. Wong, and C. T.C. Nguyen. VHF free-free beam high-Q micromechanical resonators. *IEEE/ASME J. Microelectromech. Syst.*, 9(3):347–360, 2000.

[40] M.E. Motamedi. Micro-Opto-Electro-Mechanical Systems. *Optical Engineering*, 33:3505–3517, 1994.

[41] M. Yue. A 2-D Microcantilever Array for Multiplexed Biomolecular Analysis. *Jouranl of Microelectromechanical systems*, 13:290–299, 2004.

[42] R.R.A. Syms and A. Lohmann. MOEMS Tuning Element for a Littrow External Caviry Laser. *Jouranl of Microelectromechanical systems*, 12:921–928, 2003.

[43] T.P. Burg, A.R. Mirza, N. Milovic, C.H. Tsau, G.A. Popescu, J.S. Foster, and S.R. Manalis. Vacuum-Packaged Suspended Microchannel Resonant Mass Sensor for Biomolecular Detection. *Microelectromechanical Systems, Journal of*, 15(6):1466 – 1476, 2006.

[44] T. Velten, H. H. Ruf, D. Barrow, N. Aspragathos, P. Lazarou, E. Jung, C. K. Malek, M. Richter, and J. Kruckow. Packaging of Bio-MEMS: Strategies, Technologies, and Applications. *IEEE Transcation on advanced packaging*, 28(4):533–546, 2005.

[45] S. Beeby, M.Kraft, G.Ensell, and N.White. *MEMS mechanical sensors.* Artech House, Inc. ISBN 1-58053-536-4, Boston,London, 2004.

[46] H.A. Rouabah, C.O. Gollasch, and M. Kraft. Design Optimisation of an Electrostatic MEMS Actuator with Low Spring Constant for an "Atom Chip". *Proceedings of 2005 NSTI Nanotechnology Conference and Trade Show*, 3:489–492, 2005.

[47] K. Govardhan and Z.C. Alex. MEMS Based Humidity Sensor. *Proc. of ISSS 2005 Int. Conf. Smart Materials Structures and Systems*, pages 28–30, July 2005.

[48] V. Teslyuk, Y. Kushnir, R. Zaharyuk, and M. Pereyma. A computer aided anslysis of a capacitive accelerometer parameters. *Int. Conf. CAD Systems in Microelectronics, CADSM 2007*, pages 548–550, Feb 2007.

[49] J.M. Funk, J.G. Korvink, J. Buhler, M. Bachtold, and H. Baltes. SOLIDIS : A tool for microactuator simulation in 3-D. *Journal of microelectromechanical systems*, 6(1):70–82, 1997.

[50] I.Getreu. Behavioral modelling of Analog Blocks using SABER simulator. *Proc. Microwave Circuits and Systems*, pages 997–980, 1989.

[51] G. Lorenz and R. Neul. Network-type modeling of micromachined sensor systems. *Proc. Modeling and Simulation of Microsystems '98*, pages 233–238, Apr. 1998.

[52] A. Holovatyy, M. Lobur, and S. Bobalo. Modeling and analysis of tehcnical characteristics of micromechanical Z-axis gimbaled gyroscope using VHDL-AMS. *Int. Conf. Perspective Technologies and Methods in MEMS Design, MEMSTECH 2008*, pages 72–74, May 2008.

[53] M. Shafique, A. Menon, K. Virk, and J. Madsen. System-level modeling and simulaiton of MEMS-based sensors. *9th International Multitopic Conference, IEEE INMIC 2005*, pages 1–6, Dec 2005.

[54] P.R. Wilson, J.N. Ross, A.D. Brown, and A. Rushton. Multiple domain behavioral modeling using VHDL-AMS. *Circuits and Systems, 2004. ISCAS '04*, 5(23-26):V–644–V–647, Dec 2005.

[55] S. Guessab and J. Oudinot. Modeling of a piezoelectric device with shocks management using vhdl-ams. *Proc. Int. Conf. Behavioral Modeling and Simulation Conference, BMAS*, pages 19– 24, 2004.

[56] F. Pecheux, C. Lallement, and A. Vachoux. VHDL-AMS and Verilog-AMS as alternative hardware description languages for efficient modeling of multidiscipline systems. *Computer-Aided Design of Integrated Circuits and Systems, IEEE Transactions on*, 24(2):204C225, Feb. 2005.

[57] CoventorWare for MEMS CAD Design, Multiphysics Modeling and Simulation. *Coventor, Inc. (www.coventor.com/coventorware.html)*, 2010.

[58] A.R. Sankar, V. Saini, S. Das, and S. Kal;. Temperature drift analysis in silicon micromachined piezoresistive accelerometer. *Proceedings of the IEEE India Annual Conference, INDICON 2004*, pages 561 – 563, 2005.

[59] G. Gattiker. An Actuated Microneedle for Automated Blood Sampling. *(http://www.coventor.com/mems/techinfo/research.html).*

[60] M. D. Silva. Using CoventorWare for BAW and FBAR Design. *Coventor, Inc.(http://www.coventor.com/mems/techinfo/research.html).*

[61] A. Bendali, R. Labedan, F. Domingue, and V. Nerguizian. Holes Effects on RF MEMS Parallel Membranes Capacitors. *Electrical and Computer Engineering, 2006. CCECE '06. Canadian Conference on*, pages 2140 – 2143, 2006.

[62] J. Hilbert and A. Morris. Rf Mems Switch Platforms Expedite MEMS Integration and Commercialization. *COMMS 2002.*

[63] Jung-Tang Huang and Shao-Yi Li. Using Residual Stresses to Develop CMOS-compatible RF MEMS Switches. *Electronic Materials and Packaging, 2006. EMAP 2006. International Conference on*, pages 1–6, 2006.

[64] J.V. Clark, N. Zhou, D. Bindel, L. Schenato, W. Wu, J. Demmel, and K.S.J. Pister. 3D MEMS Simulation Modeling Using Modified Nodal Analysis. *Proc. Microscale Syst.: Mech. and Meas.*, pages 68–75, Jun. 2000.

[65] J.V. Clark, N. Zhou, D. Bindel, S. Bhave, Z. Bai, J. Demmel, and K.S.J. Pister. Sugar:Advancements in a 3D multi-domain simulation package for MEMS. *Proc. Microscale Syst.: Mech. and Meas.*, Jun. 2001.

[66] J.V. Clark and K.S.J. Pister. Modeling,Simulation, and Verification of an Advanced Micromirror Using SUGAR. *J. of Microelectromechanical Syst,*, 16(6):1524–1536, Dec. 2007.

[67] J.V. Clark, N. Zhou, and K.S.J. Pister. MEMS simulation using sugar v0.5. *Proc.Transducer's Solid-State Sensor and Actuator Workshop,*, pages 191–196, Jun. 1998.

[68] J.V. Clark, D. Bindel, W. Kao, E. Zhu, A. Kuo, N. Zhou, J. Nie, J. Demmel, Z. Bai, S. Govindjee, K.S.J. Pister, M. Gu, and A. Agogino. Addressing the needs of complex MEMS design. *Proc.IEEE 15th annu. Int. Conf.Micro ElectroMech. Syst.*, pages 204–209, Jan. 2002.

[69] Z. Bai, D. Bindel, J.V. Clark, J. Demmel, K.S.J. Pister, and N. Zhou. New numerical techniques and tools in sugar for 3D MEMS simulation. *Proc.Tech.4th Int. Conf. modeling and simul. Microsyst,*, pages 31–34, Mar. 2001.

[70] G.K. Fedder and R.T. Howe. Multimode Digital Control of a Suspended Polysilicon Microstructure. *Journal of Microelectromechanical systems*, 5(4):287–297, 1996.

[71] Y.-J.J. Yang and P.-C. Yen. An Efficient Macromodeling Methodology for Lateral Air Damping Effects. *Journal of Microelectromechanical systems*, 14(4):812 – 828, Aug. 2005.

[72] D.Xiao, Wu X, and S. Li. System Design of Second-Order Sigma-Delta Micromachined Accelerometer. *Proceedings of the 1st IEEE International Conference on Nano/Micro Engineered and Molecular Systems*, pages 299–302, 2006.

[73] T.L. Grigorie. The Matlab/Simulink modeling and numerical simulation of an analogue capacitive micro-accelerometer. Part 1: Open loop. *Perspective Technologies*

and Methods in MEMS Design, MEMSTECH 2008. International Conference on, pages 105 – 114, 2008.

[74] T.L. Grigorie. The Matlab/Simulink modeling and numerical simulation of an analogue capacitive micro-accelerometer. Part 2: Closed loop. *Perspective Technologies and Methods in MEMS Design, MEMSTECH 2008. International Conference on*, pages 115 – 121, 2008.

[75] W. Gwarek. Analysis of arbitrarily shaped two-dimensional microwave circuits by finite-difference time-domain method. *IEEE Transactions on Microwave Theory and Technique*, 36:738–744, 1988.

[76] D. C. Hamill. Lumped equivalent circuits of magnetic components: The gyrator-capacitor approach. *IEEE Trans. Power Electron.*, 8(2):97–103, Apr. 1993.

[77] S.D. Senturia. Microsystem Design. *Kluwer Academic Publishers 2001. ISBN-0-7923-7246-8*, 2001.

[78] E. Christen and K. Bakalar. VHDL-AMSCa hardware description language for analog and mixed-signal applications. *Circuits and Systems II: Analog and Digital Signal Processing, IEEE Transactions on,*, 46(10):1263C1272, 1999.

[79] P.R. Wilson, J.N. Ross, A.D. Brown, and A. Rushton. Multiple domain behavioral modeling using VHDL-AMS. *In Circuits and Systems, 2004. ISCAS04. Proceedings of the 2004 International Symposium on*, 5:644C647, 2004.

[80] I. Miller and T. Cassagnes. Verilog-A and Verilog-AMS provides a new dimension in modeling and simulation. *In Devices, Circuits and Systems, 2000. Proceedings of the 2000 Third IEEE International Caracas Conference on,*, page C49/1CC49/6, Mar. 2000.

[81] H.J. Al-Junaid. SystemC-A: Analogue and Mixed-Signal Language For High Level System Design. *Thesis, ECS, University of Southampton*, 2006.

[82] OSCI. SystemC AMS extensions 1.0 language standard(http://www.systemc-ams.org/). 3 2010.

[83] IEEE standard VHDL. *IEEE Std 1076.1*, 1993.

[84] L. Fanucci, A. Giambastiani, F. Iozzi, C. Marino, and A. Rocchi. Platform based design for automotive sensor conditioning. *Design, Automation and Test in Europe (DATE05)*, 3:186C191, 2003.

[85] J. Deligueta, C. Richards, and E. Moser. Coupling of CAE tools for mixed simulation of multi-domain systems in automotive applications. *In Proceedings, IFAC Conference on Mechatronic Systems*, page 809C816, 2000.

[86] V.R. Kasulasrinivas and H.W. Carter. Modeling and simulating semiconductor devices using VHDL-AMS. *Behavioral Modeling and Simulation (BMAS)*, pages 22 – 27, 2000.

[87] M. Shafique, A. Menon, K. Virk, and J. Madsen. System level modeling and simulation of MEMS-basd sensors. *9th International Multitopic Conference, IEEE INMIC*, pages 1–6, 2005.

[88] M. Shafique, A. Menon, K. Virk, and J. Madsen. System-Level Modeling and Simulation of MEMS-based Sensors. *9th International Multitopic Conference, IEEE INMIC 2005,*, pages 1–6, Dec. 2005.

[89] A. Holovatyy, M. Lobur, and V. Teslyuk. VHDL-AMS model of mechanical elements of MEMS tuning fork gyroscope for the schematic level of computer-aided design. *Perspective Technologies and Methods in MEMS Design, 2008. MEMSTECH 2008. International Conference on*, pages 138 – 140, 2008.

[90] T. Sviridova, Y. Kushnir, and D. Korpyljov. VHDL-AMS models in MEMS simulations. *CAD Systems in Microelectronics, 2007. CADSM '07*, pages 566 – 566, 2007.

[91] A. Holovatyy, M. Lobur, and S. Bobalo. Modeling and analysis of technical characteristics of micromechanical z-axis gimbaled gyroscope using VHDL-AMS. *Perspective Technologies and Methods in MEMS Design, 2008. MEMSTECH 2008. International Conference on*, pages 72–74, 2008.

[92] R. Zaharyuk, V. Teslyuk, I. Farmaga, and H.A.Y. AlShawabkeh. VHDL-AMS model for capacitive interdigital accelerometer. *Perspective Technologies and Methods in MEMS Design. MEMSTECH*, pages 134 – 137, 2008.

[93] M. Kraemer, D. Dragomirescu, V. Puyal, and R. Plana. A VHDL-AMS model of RF MEMS capacitive shunt switches. *Semiconductor Conference, CAS 2009*, pages 307 – 310, 2009.

[94] M.M. Sakr, M.K. El-Shafie, and H.F. Ragai. Analysis and Modeling of RF-MEMS Disk Resonator. *MEMS, NANO and Smart Systems, The 2006 International Conference on*, pages 19 – 22, 2006.

[95] H. Boussetta, M. Marzencki, S. Basrour, and A. Soudani. Efficient Physical Modeling of MEMS Energy Harvesting Devices With VHDL-AMS. *Sensors Journal, IEEE*, 10(9):1427 – 1437, 2008.

[96] A. Endemano, J.Y. Fourniols, H. Camon, A. Marchese, S. Muratet, F. Bony, and M.P.Y. Desmulliez. VHDL-AMS modelling, simulation and testing of electrostatic micromotors. *Design, Test, Integration and Packaging of MEMS/MOEMS 2003. Symposium on*, pages 56 – 63, 2003.

[97] Southampton VHDL-AMS Validation. *(http://www.syssim.ecs.soton.ac.uk/)*.

[98] Open SystemC Initiative OSCI. SystemC Language Reference Manual. 2003.

[99] T. Bonnerud, B. Hernes, and T. Ytterdal. A Mixed-Signal Functional Level Simulation Framework Based on SystemC. *in Proceedings IEEE Custom Integrated Circuits Conference, San Diego California USA*, pages 6–9, May 2001.

[100] M.O'Nils, J.Lundgren, and B. Oelmann. A systemC extension for bebavioural level Quantificationof Noise Coupling in Mixed-Signal Systems. *in Proceedings IEEE International Symposium on Circuits and Systems, Bangkok, Thailand*, pages 25–28, May 2003.

[101] K.Einwich, Ch.Clauss, G. Noessing, P.Schwarz, and H.Zojer. Systemc extensions for mixed-signal system design,. *in Proceedings Forum on Specification and Design Languages, Lyon France*, pages 3–7, Sep. 2001.

[102] A. Vachoux, Ch. Grimm, and K. Einwich. SystemC-AMS Requirements, Design Objectives and Rationale. *in Proceedings Design, Automation and Test in Europe Conference and Exhibition, Messe Munich, Germany*, pages 3–7, Mar. 2003.

[103] T. Kazmierski H. Al-Junaid and L. Wang. SystemC-A Modeling of an Automative Seating Vibration Isolation System. *Forum on Specification and Design Languages (FDL 2006)*, 2006.

[104] L. M. Roylance. A batch-fabricated silicon accelerometer. *IEEE Trans. Electron Devices*, 26:1911–7, 1979.

[105] V. P. Petkov and B. E. Boser. A Fourth-Order SD Interface for Micromachined Inertial Sensors. *IEEE Journal of Solid-State Circuits*, 40(8), Aug. 2005.

[106] M. Kraft, C.P.Lewis, and T.G.Hesketh. Closed-loop silicon accelerometers. *IEE Proc.-Circuits Devices Syst.*, 145(5):325–331, October 1998.

[107] J. P. Lynch, A. Partridge, K. H. Law, T. W. Kenny, A. S. Kiremidjian, and E. Carryer. Design of Piezoresistive MEMS-Based Accelerometer for Integration with Wireless Sensing Unit for Structural Monitoring. *Journal of Aerospace Engineering*, JULY 2003.

[108] S. Huang, X. Li, Z. Song, Y. Wang, H. Yang, L. Che, and J. Jiao. A high-performance micromachined piezoresistive accelerometer with axially stressed tiny beams. *Journal of Micromechanics and Microengineering*, (15):993 C1000y, 2005.

[109] H. Seidel, U. Fritsch, R. Gottinger, J. Schalk, J. Walter, and K. Ambaum. A Piezoresistive Silicon Accelerometer With Monolithically Integrated CMOS-circuitry. *Proc. Eurosensors IX and Transducers*, page 597C600, 1995.

[110] D.L. DeVoe and A.P. Pisano. A fully surface-micromachined piezoelectric accelerometer. *International Conference on Solid-State Sensors and Actuators Chicago*, pages 16–19, June 1997.

[111] Y. Nemirovsky, A. Nemirovsky, P. Muralt, and N. Setter. Design of a Novel Thin-Film Piezoelectric Accelerometer. *Sensors and Actuators*, A56:239C249, 1996.

[112] T.A. Roessig, R.T. Howe, A.P. Pisano, and J.H. Smith. Surface-Micromachined Resonant Accelerometer. *International Conference on Solid-State Sensors and Actuators.*, pages 859–862, 1997.

[113] D. W. Burnsa, R. D. Horninga, W. R. Herba, J. D. Zooka, and H. Guckelb. Sealed-Cavity Resonant Microbeam Accelerometer. *Sensors and Actuators*, A53:249C255, 1996.

[114] Datasheet. FOA-100 Optical Accelerometer, (http://www.vibrosystm.com). *VibroSystm*, 2002.

[115] J.A. Plaza, A. Llobera, J.Berganzo, J.Garcia, C.Dominguez, and J.Esteve. Stress free quad beam optical silicon accelerometer. *IEEE Proceedings of Sensors*, 2:1064–1068, 2002.

[116] S. J. Sherman, W. K. Tsang, T. A. Core, R. S. Payne, D. E. Quinn, K. H.-L. Chau, J. A. Farash, and S. K. Baum. A low cost monolithic accelerometer; product/technology update. *IEDM Technical Digest*, pages 501–504, Dec. 1992.

[117] M. Kraft. Closed loop digital accelerometer employing oversamping conversion. *dissertation, Coventry University, School of Engineering, UK,*, 1997.

[118] Y. Dong, M. Kraft, C. Gollasch, and W. Redman-White. A high-performance accelerometer with a fifth-order Sigma-Delta modulator. *Journal of Micromechanics and Microengineering*, 2:S22–S29, 2005.

[119] Yazdi N, F. Ayazi, and K. Najafi. Micromachined Inertial Sensors. *Proceedings of the IEEE*, 86(8), Aug. 1998.

[120] S.R. Norsworthy, R. Schreier, and G.C. Temes. *Delta-Sigma Data Converters: Theory, Design, and Simulation.* John Wiley and Sons. Inc, Hoboken, New Jersey, USA, 1997.

[121] D.A. Jones and K. Martin. Analog integrated circuit design. *3rd.ed., wiley,NY,*, 1993.

[122] R. Schreier and G.C.Temes. Understanding Delta-Sigma Data Converters. *Published by John Wiley and Sons. Inc.*, page ISBN: 0471465852, 2005.

[123] K. Mucahit and I. Kale. *Oversampled Delta-Sigma Modulators.* Dordrecht Netherlands: Kluwer Academic Publishers,, 2003.

[124] C.H. Chao, S.Nadeem, W.L.Lee, and C.G.Sodini. A Higher Order Topology for Interpolative Modulators for Oversampling A/D Converters. *IEEE Trans. on Circuits and Systems*, 37(3), Mar. 1990.

[125] P. Ferguson. An 18-bit 20 kHz dual sigma-delta A/D converter. *ISSCC Dig. Tech. Pap.*, pages 68–69, Feb. 1991.

[126] Y. Matsuya. A 16-Bit Oversampling A/D Conversion Technology Using Triple-Integration Noise Shaping. *IEEE Journal of Solid-State Circuits*, SC-22(6):921–929, Dec. 1987.

[127] Y. Dong, M. Kraft, and W. Redman-White. Micromachined vibratory gyroscopes controlled by a high order band-pass sigma delta modulator. *IEEE Sensors Journal*, 7:50–69, 2007.

[128] Y. Dong, M. Kraft, and W. Redman-White. Higher order noise-shaping filters for high-performance micromachined accelerometers. *IEEE Trans. instrumentation and measurement*, 56(5):1666–1674, Oct 2007.

[129] M. Kraft, W. Redman-White, and M.E. Mokhtari. Closed loop micromachined sensors with higher order SD-Modulators. *Proc. 4rd Conf. on Modeling and Simulation of Microsystems*, pages 104–107, 2001.

[130] Y. Zhang, R. Kamalian, and A.M. Agogino. Hierarchical MEMS Synthesis and Optimization. *SPIE Conference on Smart Structures and Materials,*, (5763-12), 2005.

[131] V. Gupta and T. Mukherjee. Layout synthesis of CMOS MEMS Accelerometer. *Thesis, Department of ECE,Carnegie Mellon University, Pittsburgh, PA-15213*, 1999.

[132] V. Gupta and T. Mukherjee. Automated Optimal Synthesis of Microresonators. *Solid State Sensors and Actuators, 1997. TRANSDUCERS '97 Chicago., International Conference on*, 2:1109–1112, 1997.

[133] R.C. Rosenberg. Reflections on Engineering Systems and Bond Graphs. *Trans. ASME J. Dynamic Systems, Measurements and Control*, 115:242–251, 1993.

[134] N. Vargas-Hernandez, J.shah, and Z.Lacroix. Develeopemt of a Computer-Aided Conceptual Design Tool for Complex Electromechanical Systems. *Computational Synthesis: From Basic Building Blocks to Hign Level Functionality, Papers from the 2003 AAAI Symposium Technical Report*, pages 255–261, 2003.

[135] N. Zhou, A. M. Agogino, and K.S. Pister. Automated Design Synthesis for Micro-Electro-Mechanical Systems (MEMS). *Proceedings of the ASME Design Automation Conference,*, 2002.

[136] R. Kamalian, N. Zhou, and A.M. Agogino. A Comparison of MEMS Synthesis Techniques. *Proceedings of the 1st Pacific Rim Workshop on Transducers and Micro/Nano Technologies*, pages 239–242, 2002.

[137] R.H. Kamalian, H. Takagi, and A.M. Agogino. Optimized Design of MEMS by Evolutionary Multi-objective Optimization with Interactive Evolutionary Computation. *Proceedings of the GECCO 2004 (Genetic and Evolutionary Computation Conference)*, 2:1030–1041, 2004.

[138] W. Kuehnel. Modelling of the mechanical behaviour of a differential capacitor acceleration sensor. *Sensors and Actuators*, A48:101–108, 1995.

[139] Y. Sawada and A. Ohsumi. Active vibration control for exible cantilever beam subject to seismic disturbance. *IEEE International Conference on Systems Engineering*, pages 436 –439, 1992.

[140] G. Barillaro, A. Molfese, A. Nannini, and F. Pieri. Analysis, simulation and relative performances of two kinds of serpentine springs. *J. Micromech. Microeng.*, pages 736 – 746, 2005.

[141] J. Carnahan and R. Sinha. Nature's algorithm[genrtic algorithms]. *IEEE Journal of Potential*, 20(2):21–24, 2001.

[142] Y. Zhang, R. Kamalian, A.M.Agogino, and H.S. Carlo. Hierarchical MEMS synthesis and optimization. *SPIE Conference on Smart Structures and Materials*, 5763(12), 2005.

[143] H. Tang and A. Doboli. High-level synthesis of Delta-Sigma modulator topologies optimized for complexity,sensitivity, and power consumption. *In Trans. on Computer Aided Design of Integrated Circuit and Systems*, 25(3):597–607, 2006.

[144] S. Kadivar. A new algorithm for the design of stable higher order single loop sigma-delta analog-to-digital converters. *in Proc. Int. Cond. Computer-Aided Design*, pages 554–561, 1995.

[145] T.Kuo, K. Chen, and J. Chen. Automated coefficients design for high-order sigma-delta modulators. *IEEE Trans. Circuits Sys. Analog Digital Signal Process*, 46(1):6–15, 1999.

[146] F. Medeiro F. Fernandez R. Rio B. Perez-Verdu J. Ruiz-Amaya, J. Rosa and A. Rodriguez-Vazquez. An optimizationbased tool for the high-level synthesis of discrete-time and continuous-time Sigma-Delta modulators in the matlab/simulink environment. *In Proc. IEEE Int. Symp. Circuit Syst.*, pages 97–100, 2004.

[147] K.Francken and G.Gielen. DAISY: A simulation-based high-level synthesis tool for delta-sigma modulators. *In Proc. Int.Conf.Computer Aided Design(ICCAD)*, pages 188–192, 2000.

[148] O.Yetik, O. Saglamdemir, S. Talay, and G. Dundar. A coefficient optimization and architecture selection tool for Sigma-Delta modulators in MATLAB. *In Proc.DATE '07*, pages 87 – 92, 2007.

[149] L. Yao, M.S.Steyaert, and W. Sansen. A 1-V 140uW 88-dB audio sigma-delta modulator in 90-nm CMOS. *IEEE Journal of Solid-State Circuits*, 39(11), 2004.

[150] A. Marques, V. Peluso, M. Steyaert, and W. Sansen. Optimal parameters for Delta-Sigma modulator topologies. *IEEE Trans. Circuit Syst. Analog Digital.Signal Process.*, 45(9):1232–1241, 1998.

[151] D.A. White and M. Stowell. Full-wave simulation of electromagnetic coupling effects in RF and mixed-signal ICs using a time-domain finite-element method. *IEEE Trans. Microwave Theory Tech.*, 52(2):1404–1413, 2004.

[152] E. Normark P.V. Nikitin and C.J.R. Shi. Distributed electrothermal modeling in VHDL-AMS. *International Workshop on Behavioral Modeling and Simulation, 2003. BMAS*, pages 128–133, 2003.

[153] X. Huang and H.A. Mantooth. Event-driven electrothermal modelng of mixed-signal circuits. *International Workshop on Behavioral Modeling and Simulation, (BMAS)*, pages 10–15, 2000.

[154] L. Wang, C.Zhao, and T.J.Kazmierski. An Extension to VHDL-AMS for AMS Systems with Partial Differential Equations. *Forum on Specification & Design Languages Conference (FDL)*, 2007.

[155] C.R. Shi P.V. Nikitin and B. Wan. Modeling partial differential equations in VHDL-AMS. *In Systems-on- Chip Conference, 2003. Proceedings. IEEE International*, pages 345–348, 2003.

[156] R. P. Clayton. Incorporation of Terminal Constraints in the FDTD Analysis of Transmission Lines. *IEEE Trans. on Electromagnetic Compatibility,*, 36(2), 1994.

[157] T. Dhaene and D. De Zutter. Selection of lumped element models for coupled lossy transmission lines. *IEEE Trans. Computer-Aided Design,*, 11:805–815, 1992.

[158] M.Burford and T.Kazmierski. A VHDL-AMS Based Time Domain Skin Depth Model for Edge Coupled Lossy Transmission Stripline. *Forum on Specification and Design Languages,*, pages 197–208, 2005.

[159] Polar Instrument Ltd. Si8000 Controlled Impedance Field Solver-user guide. *(www.polarinstruments.com)*, 2003.

[160] S. V. Marshall, R. E. DuBroff, and G. G. Skitek. Electromagnetic Concepts and Applications. *4th ed., Prentice-Hall, Englewood Cliffs, NJ,*, 1996.

[161] S. M. Han, H. Benaroya, and T. Wei. Dynamics of transversely vibrating beams using four engineering theories. *Journal of Sound and Vibration*, 226(5):935–988, 1999.

[162] A. Konak, D.W. Coit, and A. E. Smith. Multi-objective optimisation using genetic algorithm: A tutorial. *Reliability Engineering and System Safety 91*, pages 992–1007, 2006.

[163] M. J. Tudor S. P. Beeby and N. M. White. Energy harvesting vibration sources for microsystems applications. *Measurement Science and Technology*, 17(12):R175–R195, 2006.