Evolvable Smart-phone-based Point-of-Care Systems for In-Vitro Diagnostics

François Patou
PhD Thesis July 2016

DTU Nanotech
Department of Micro- and Nanotechnology

PhD. Thesis

Evolvable Smartphone-based Point-of-Care Systems for In-Vitro Diagnostics

François Patou

—

Technical University of Denmark
15/07/2016

—

Supervisors:
Winnie E. Svendsen
Jan Madsen
Maria Dimaki

DTU Nanotech
Department of Micro- and Nanotechnology

Preface

This thesis is submitted as part of work done to obtain the PhD degree at the Technical University of Denmark. The project is part of the EU Initial Training Network (ITN) EngCaBra, a European collaboration for the development of novel diagnostic and therapeutic tools for cancer and brain diseases, and funded as a Marie-Curie Actions fellowship.

The research was conducted in the Nano-Bio Integrated Systems group (NaBiS), led by Winnie E. Svendsen, at the Department of Micro- and Nanotechnology. The full project at DTU consisted of three PhD students, Azeem Zulfiqar, who was primarily responsible for the development and fabrication of the sensor hardware, Andrea Pfreundt, responsible for surface functionalization, protein work and microfluidic integration of the sensor, and myself.

All research, graphics and text is my original work to the best of my knowledge unless other authors and contributors are mentioned.

Abstract

Recent developments in the life-science *-omics* disciplines, together with advances in micro- and nanoscale technologies offer unprecedented opportunities to tackle some of the major healthcare challenges of our time. Lab-on-Chip technologies coupled with smart-devices in particular, constitute key enablers for the decentralization of many in-vitro medical diagnostics applications to the point-of-care, supporting the advent of a preventive and personalized medicine.

Although the technical feasibility and the potential of Lab-on-Chip/smart-device systems is repeatedly demonstrated, direct-to-consumer applications remain scarce. This thesis addresses this limitation. After identifying *system evolvability* as a key enabler to the adoption and long-lasting success of next-generation point-of-care systems by favoring the integration of new technologies, streamlining the reengineering efforts for system upgrades and limiting the risk of premature system obsolescence. Among possible strategies, platform-based design represents a particularly suitable entry point to the development of evolvable systems. One necessary condition, is for *change-absorbing* and *change-enabling* mechanisms to be incorporated in the platform architecture at initial design-time. Important considerations arise as to *where* in Lab-on-Chip/smart-device platforms can these mechanisms be integrated, and *how* to implement them.

Our investigation revolves around the silicon-nanowire biological field effect transistor, a promising biosensing technology for the detection of biological analytes at ultra low concentrations. We discuss extensively the sensitivity and instrumentation requirements set by the technology before we present the design and implementation of an evolvable smartphone-based platform capable of interfacing lab-on-chips embedding such sensors. We elaborate on the implementation of various architectural patterns throughout the platform and present how these facilitated the evolution of the system towards one accommodating for electrochemical sensing. Model-based development was undertaken throughout the engineering process. A formal SysML system model fed our evolvability assessment process. We introduce, in particular, a model-based methodology enabling the evaluation of *modular scalability*: the ability of a system to scale the current value of one of its specification by successively reengineering targeted system modules.

The research work presented in this thesis provides a roadmap for the development of evolvable point-of-care systems, including those targeting direct-to-consumer applications. It extends from the early identification of anticipated change, to the assessment of the ability of a system to accommodate for these changes. Our research should thus interest industrials eager not only to disrupt, but also to last in a shifting socio-technical paradigm.

Resume

Sammen med fremskridtene indenfor mikro- og nanoteknologier, giver den seneste udvikling indenfor biovidenskabens -*omics* discipliner en ikke før set mulighed for at takle nogle af de allerstørste udfordringer i vor tid indenfor sundhedspleje. Lab-on-a-chip teknologier, særligt i kombination med smarte enheder, udgør en vigtig forudsætning for at kunne decentralisere mange medicinske in-vitro diagnostiske applikationer til point-of-care, til støtte for fremkomsten af forebyggende og personliggjort medicin.

Selvom det er teknisk muligt, og selvom potentialet for lab-on-a-chip/smart-enheds -systemer, er påvist gentagne gange, forbliver direkte-til-forbrugeren applikationer en sjældenhed. Denne afhandling adresserer denne begrænsning. Vi bygger på den antagelse, at et systems potentiale for udvikling kan fremme indførelsen og den langvarige succes af næste-generations point-of-care systemer, ved at favorisere integrationen af nye teknologier, strømline den ingeniørmæssige arbejdsindsats i forbindelse med systemopgraderinger, og derved begrænse risikoen for at systemet bliver for tidligt forældet. Blandt de mulige strategier, udgør især platform-baserede designs et særlig egnet udgangspunkt for udviklingen af evolutions-egnede systemer. Dog kræver det at ændrings-absorberende og ændrings-muliggørende mekanismer inkorporeres i platformarkitekturen allerede under den indledende designfase. Vigtige overvejelser opstår om *hvor* i Lab-on-Chip/smart-enheds-platforme disse mekanismer integreres, samt *hvordan* disse implementeres.

Vores undersøgelse er centreret omkring silicium-nanowire field effect transistorer, en lovende biosensing teknologi for detektionen af biologiske analytter i ultra lave koncentrationer. Vi diskuterer udførligt følsomheds- og instrumentale krav som teknologien stiller, før vi præsenterer design og implementationen af en smartphone-baseret platform, som kan fungere som grænseflade til lab-on-a-chips med integration af disse sensorer. Vi redegøre videre for implementationen af forskellige arkitektoniske mønstre gennemløbende for hele platformen, og præsenterer hvordan disse faciliterede udviklingen af systemet mod et, som kunne rumme elektrokemisk detektion. Model-baseret udvikling blev anvendt gennem hele den ingeniørmæssige udviklingsproces. En formel SysML system model fodrede vores vurderingsproces af systemets udviklingspotentiale. Vi introducerer, især, en modelbaseret metode, som muliggør evalueringen af den *modulare skalerbarhed*: et systems evne til at skalere de nuværende specifikations-værdier, ved successiv omstrukturering af målrettede systemmoduler.

Det videnskabelige arbejde præsenteret i denne afhandling fremlægger en køreplan for udviklingen af evolutions-egnede systemer, inklusive dem der er møntet direkte på forbruger-applikationer. Det strækker sig fra den tidlige identifikation af forventede ændringer, til vurderingen af systemets evne til at rumme disse ændringer. Vores forskning skulle derfor interessere industrikunder som er ivrige efter, ikke bare at forstyrre, men også at forblive i et omskifteligt socio-teknisk paradigme.

Acknowledgments

I would first of all like to thank Assoc. Prof. Winnie E. Svendsen, group leader at the Department of Micro and Nanotechnology, at DTU, for giving me the chance to pursue my PhD project within the Nano-Bio-Integrated Systems group (NaBiS). This has been a considerable opportunity for me to go back to research and development after spending some time in the industry as a field application engineer. This certainly has been a life changing experience, opening up so many new perspectives on what my aspirations of a fulfilling career really are. I am deeply grateful for her continuous trust and support during the past three years.

I would like to particularly thank Prof. Jan Madsen, Deputy Director at DTU Compute, Head of Section of Embedded Systems Engineering and co-supervisor of my PhD studies. Jan has provided me with valuable guidance, and has supported my vision and efforts throughout this project. He also provided me with considerable insight into what the future of complex systems at the intersection of computing and biology might look like. Jan has thus greatly contributed to shaping some of the new interests I have developed throughout the past few years and motivated me to aim high.

I would like to thank everyone in the NaBiS group for their valuable inputs and for allowing me to become familiar with their fields of research. A particular thank you goes to Maria Dimaki, Senior researcher at NaBiS and co-supervisor on this project, for reviewing my work and for her day-to-day support.

This project is funded as part of the EU Marie Curie Initial Training Networks (ITN) Biomedical engineering for cancer and brain disease diagnosis and therapy development (EngCaBra). Project no. PITN-GA-2010-264417.

I want to thank the European Union for making this project possible and for similar networks that enable researchers and students like myself to experience a very international and interdisciplinary environment. I am grateful for such a great opportunity and hope that initiatives like this will continue to inspire future researchers. In this spirit I want to mention my fellow EngCaBra fellows, who I am so glad to have gotten to know and to work with: Azeem Zulfiqar, Andrea Pfreundt, Franziska Uhlenbrock, Joanna Obacz, Rene Splinter, Wei Gong, Daniela Pucciarelli, Ben van Lier, Amir Shadmani, Valeria Fioravanti, Sona Kontsekova, Reza Ebrahimifard, Joao Pimentel and Filippo Cipriani.

I would like to address a big thank you to my family: my sisters Caroline and Nathalie and parents Annie and Xavier for their long-distance support. A final special thank you goes to my girlfriend Lubica and friends here in Copenhagen. Life as an expat is certainly challenging, but they certainly made these last three years really worth it.

Contents

Abstract **2**

Acknowledgments **4**

1 Background **9**
1.1 Landscape of our modern healthcare systems . . . 9
1.1.1 In-Vitro Medical Diagnostics . . . 10
1.1.1.1 Personalized medicine and the *-omics* Era . . . 10
1.1.1.2 Towards direct-to-consumer IVMD applications at the Point-of-Care 12
1.2 Point-of-Care systems and enabling technologies . . . 15
1.2.1 Lab-on-Chip . . . 15
1.2.2 Lab-on-Chip-based system design and integration challenges . . . 16
1.2.3 Smart-devices and Lab-on-Chip-based IVMD testing . . . 17
1.2.4 Design challenges for direct-to-consumer smartphone-based IVMD . . . 19
1.3 Summary . . . 20

2 Introduction **22**

3 An engineering systems framework for evolvable Point-of-Care In-Vitro Diagnostics **23**
3.1 Systems engineering and engineering systems . . . 24
3.1.1 Engineering systems . . . 24
3.1.2 Systems engineering . . . 25
3.1.3 Conceptual design of engineering systems . . . 26
3.1.3.1 Systems architecting . . . 26
3.1.4 The *-ilities* of engineering systems . . . 27
3.1.4.1 Change-related *-ilities* . . . 28
3.1.4.2 System evolvability . . . 29
3.1.5 Design frameworks and methodologies . . . 32
3.2 Model-based systems engineering for evolvable systems . . . 33
3.2.1 Model-based systems engineering . . . 33
3.2.1.1 SysML modelling of cyber-physical systems . . . 34
3.2.2 Evolvable platform-based design . . . 35
3.3 Summary . . . 37

4 Silicon-nanowire biological field-effect transistors for ultrasensitive Point-of-Care systems **39**
4.1 SiNW-bioFET sensing technology . . . 39
4.1.1 MOS-FET architecture . . . 39
4.1.2 MOS-FET behavior . . . 40
4.1.2.1 MOS-FET depletion regime . . . 40
4.1.2.2 MOS-FET subthreshold regime . . . 41
4.1.2.3 MOS-FET linear regime . . . 41
4.1.3 The Silicon Nanowire biological Field Effect Transistor (SiNW-bioFET) . . 42
4.1.3.1 SiNW-bioFETs sensitivity . . . 44
4.1.3.2 SiNW-bioFETs integrability . . . 48
4.2 Lock-in synchronous detection . . . 49

4.2.1 Fundamental principles of lock-in amplification . . . 50
4.2.1.1 Input signal and synchronous references . . . 50
4.2.1.2 Phase-Sensitive Detection . . . 51
4.2.1.3 Digital lock-in amplification . . . 53
4.3 Model-based sensitivity analysis of lock-in amplified SiNW-bioFET signals . . . 53
4.3.1 Analytical and behavioral SiNW-bioFET/instrumentation models . . . 54
4.3.1.1 SiNW-bioFET model . . . 55
4.3.1.2 Signal conditioning . . . 60
4.3.1.3 Bandwidth consideration . . . 62
4.3.1.4 Signal conditioning noise analysis . . . 65
4.3.2 Phase-sensitive detection noise analysis . . . 73
4.3.2.1 Low-Pass FIR Filtering . . . 74
4.3.3 Parametric sensitivity analysis . . . 80
4.4 Summary . . . 85

5 Evolvable Smartphone-based Platform for SiNW-bioFET-based Point-of-Care applications 86
5.1 Conceptual design of evolvable platforms . . . 86
5.1.1 Function space definition and change-enabling mechanisms . . . 86
5.1.2 Architecture variants and mapping . . . 90
5.2 Platform implementation . . . 91
5.2.1 Mobile software design . . . 91
5.2.1.1 Object-oriented platform architecting . . . 91
5.2.1.2 Software composability mechanisms . . . 92
5.2.1.3 Specification of SiNW-bioFET acquisitions . . . 96
5.2.1.4 LOC program execution handling . . . 97
5.2.2 Hardware design . . . 99
5.2.2.1 Digital design . . . 99
5.2.2.2 Analog design . . . 101
5.2.3 Embedded software design . . . 103
5.2.3.1 Overview . . . 103
5.2.3.2 Waveform generation and synchronous detection . . . 104
5.2.3.3 Digital signal processing for continuous sensor monitoring . . . 111
5.2.3.4 Wireless communication protocol . . . 114
5.2.3.5 Illustrative application . . . 117
5.3 Platform evolution . . . 119
5.3.1 Electrochemical analysis functional composition . . . 121
5.3.2 Electrochemical analysis validation . . . 121
5.3.3 Application: electrochemical detection of dopamine . . . 123
5.3.4 Summary . . . 127

6 Evaluating evolvability 128
6.1 Model-based evaluation of modular scalability . . . 128
6.2 Scalability analysis for the continuous acquisition bandwidth specification . . . 133
6.2.1 Model-based estimation of continuous acquisition bandwidth . . . 135
6.2.2 System scalability over generations . . . 137
6.3 Discussion . . . 139
6.4 Summary . . . 140

7 Discussion 141
7.1 On the integrability of biosensors . . . 141
7.2 Model-based platform design, hybrid-systems and evolvability . . . 142
7.2.1 Towards a unifying design framework for IVMD systems . . . 142
7.2.2 Platform architecting for evolvability . . . 142
7.2.3 Modular scalability . . . 143
7.3 Is it worth it? . . . 143

8 Conclusion 146

List of References **148**

List of Abbreviations **162**

Appendices **165**

A Publication 1 **165**

B Publication 2 **175**

C Publication 3 - [In Press] **181**

D Publication 4 - [Submitted - IEEE Transactions on Biomedical Engineering] **187**

E Publication 5 - [Submitted - Sensing and BioSensing Research] **199**

F Signal conditioning analytical model **219**

G SiNW-bioFET Simscape model **223**

H Parametric study of SiNW-bioFET Limit Of Detection source files **226**

I POCLowLevelTask source file **231**

J POCEncoder source file **233**

K BLE Firmware **236**

“Adapt or perish, now as ever, is nature’s inexorable imperative”

— H. G. Wells

Chapter 1

Background

1.1 Landscape of our modern healthcare systems

Healthcare represents today one of the most important societal challenges of the coming decades. The global demographic explosion, the aging of the Western population, and the economic imbalances between Western and developing countries constitute some of the factors at the origin of the significant dissymmetry in the nature of the healthcare-related issues afflicting different parts of the World.

On one side, the increasing healthcare demand of the Western World's aging population stretches the restricted budgets of our institutions and challenges the limited availability of our care personnel and facilities. Chronic diseases hold a particular place in that scenery. By accounting to up to 84% of the annual 3 trillion dollars of healthcare expenditures in the United States [1], conditions like cancer, and cardiovascular or neurodegenerative diseases are not only debilitating but also represent a significant economic burden, threatening health accessibility and quality of care. Dementia for instance, which includes conditions such as Alzheimer's disease, afflicts more than 46 million people worldwide and will affect more than 131 million individuals by 2050. It will, by 2018, burden the global economy by up to one trillion dollars [2]. Another example is Heart Failure (HF), one of the most prevalent chronic conditions in the Western world. 20% of Americans older than age 40 will develop heart failure in their lifetime. With more than 650.000 new cases diagnosed each year in the US, HF has a mortality superior to 50% at 5 years after initial diagnosis. HF consultations at the doctor's office alone cost 1.8B dollars in the US in 2013, and more than 30B including hospitalization costs [3].

The healthcare landscape in the developing world, on the other hand is still daunted by the high incidence of infectious diseases. HIV/AIDS, tuberculosis or malaria remain predominant causes of death, especially in children [4], mainly as a result of insufficient resources and infrastructures [5, 6]. A special mention goes to cancer, which is diagnosed in about 14 millions new individuals every year. Against a common preconception, cancer mainly affects low- and middle-income countries with more than 57% of the Worldwide newly diagnosed cases and more than 70% of cancer-related deaths [1].

The global healthcare landscape of our modern societies is yet rapidly changing. Its three pillars: diagnostics, therapeutics and clinical management are all benefiting from the accelerating pace of innovation that drives digital and medical technologies. These developments offer the potential to profoundly improve the quality and cost-effectiveness of healthcare worldwide. The scope of this thesis will predominantly cover the opportunities offered by advances made in the first of the three pillars of healthcare, and in particular in the field of In-Vitro Medical Diagnostics (IVMD).

1.1.1 In-Vitro Medical Diagnostics

IVMD describes the comprehensive set of "examinations of *in-vitro* specimens, including blood and tissue donations, derived from the human body, solely or principally for the purpose of providing information: concerning a physiological or pathological state, or concerning a congenital abnormality, or to determine the safety and compatibility with potential recipients, or to monitor therapeutic measures" [7]. In other words, IVMD enables the assessment of individual patients susceptibility for disease. It assists the the diagnosis and prognosis of disease, and increasingly allows the monitoring of treatment response via clinically relevant surrogate biological markers.

Roughly 10 billion IVMD tests are performed each year in the United States alone [8]. Although representing only 2% of the total healthcare expenditures, IVMD tests support more than 70% of all clinical management decision-making [9]. The increasing importance of IVMD in clinical management is mirrored its soaring market growth, with a global value expected to reach $70 billion by 2017 [10].

IVMD is still today mainly characterized by a centralized, technical, high-throughput biological sample analysis process. Laboratory medicine, as it is sometimes referred to, is yet undergoing a major paradigm shift. Technological innovations of the past few decades and the emergence of several key societal trends have resulted in *broader and superior analytical and predictive IVMD tools and methods* and in the *partial decentralization of IVMD testing* [11]. These two phenomena have become particularly obvious in recent years and deserve further explanations.

1.1.1.1 Personalized medicine and the *-omics* Era

The on-going IVMD paradigm shift partly originates in one of the paramount scientific achievements of the past century: In 2001, the group of more than 200 researchers led by Francis Collins announced the completion of the first human full genome sequencing [12]. It took more than 10 years and $3 billion for The International Human Genome Sequencing Consortium to read the 3 billion DNA base-pairs that compose the human genome. Since then, significant advances in sequencing technologies have brought the cost-per-base-pair of sequenced DNA to drop exponentially, iconically faster than the decay of the cost of computation as predicted by utterly famous Moore's law [13]. Full genome sequencing today amounts to less than $1000 and is routinely completed within a few hours.

Advances in DNA sequencing, albeit the most significantly remarkable, only partially illustrate the substantial progress made in the so called *-omics* disciplines in recent years. Life science technologies for DNA, but also RNA, proteins or metabolites analyses have enabled the advent of the *genomics, transcriptomics, proteomics, metabolomics, epigenomics* or *microbiomics* disciplines, opening up a broad new spectrum of IVMD applications for what is commonly known as *personalized medicine*. Personalized medicine, also referred to as *precision medicine* [14] or *individualized medicine* [15] defines a medicine targeting the unique biological profile of each individual patient or person. The potential offered by personalized medicine relies on the vast amount and variety of biological *-omics* data that, once integrated and processed, may provide valuable diagnostic and predictive health information.

The field of genomics alone is animated by fast developments in sequencing, but also in the bioinformatics methods and tools that allow the recovery of interpretable information. The power of genomics for predicting innate disease susceptibility also increases with the number of individuals getting their full genome sequenced. This can be illustrated by the number of rare Mendelian diseases that are now rooted to identified genomic variants: from 4 in 2010, this number grew to 68 in 2012 and is thought to lead to the identification of the genetic basis for all 7000 Mendelian diseases in the coming years [15, 16]. The power

of genomics is yet not limited to the identification of congenital Mendelian diseases and to risk-stratification for polygenic conditions. Genome sequencing is increasingly applied to the human gut microbiome, the DNA content of which is several orders of magnitude larger than that of the entirety of our own body cells. The field of microbiomics is steeply gaining interest among the research community as the bacterial lines present in our gut have already been demonstrated to play a role in numerous conditions, going from autoimmune to neurological [17, 18]. Whole-genome sequencing also finds applications in infectious diseases by allowing the identification of the viral or bacterial strain responsible for an outbreak and guiding the prescription of the most relevant antibiotic in the latter case [19]. Cancer research is certainly one of the fields where genome sequencing sources the greatest hope: cancer genomics are revolutionising our understanding of the mechanisms underlying cancer onset and evolution. It even questions the classification, i.e. the taxonomy of cancer which is today still based on the organ of origin but which should instead be founded on the importance of the oncogenic profile of an individual [15]. Cancer genomic research has demonstrated the heterogeneity of tumor DNA and the relevance of both cell-free, plasma circulating tumor DNA as a biomarker for cancer screening [20], and that of the full *-omic* profiling of tumor cells, including those circulating in blood [21].

Figure 1.1.1: Evolution of the cost of whole-genome sequencing. Reproduced from [13]

Single-cell cancer genomics thus stands as one of the illustrative applications where the convergence of the *-omics* analytical tools (i.e. both for sequencing and interpretation) is needed. The informativeness of DNA is, even though tremendous, limited. It can and often must be complemented by *transcriptomics*: the evaluation of the transcription profile from DNA to RNA (e.g. using RNA sequencing technolology); *proteomics*: the investigation of the translation of the transcriptome to proteins (i.e. most often using biosensing and microarray technologies); and *epigenetics*: mainly the investigation of the DNA methylation and histone protein modifications and their modulation of gene expression. Single-cell *-omics* profiling now even evolves toward the incorporation of both linear *-omics* data together with spatial information on the distribution of that information within the cell [22].

These techniques, although still mainly tools dedicated to biological and medical research may yet find applications addressing personalized care much more directly. One of them is pharmacogenomics. The availability of the full genome of an individual makes it possible in some cases to tailor pharmacological treatment (i.e. either the nature of the

compound or the dose) to the specific gene variants this individual bears. One of the sought benefits of pharmacogenomics is to drastically increase the odds of response to treatment and reducing the risks of non-efficacy and adverse effects. About 100 FDA-approved drugs are already associated with recommendations for specific genotypes, but day to day clinical management is still failing to incorporate these recommendations into practice [15]. Pharmacogenomics illustrates the potential of personalized medicine to address the shortcomings of population-based medicine and the one-size-fits-all pharmacological model: "right patient, right drug, right time" (and right dose) [23]. It also represents the most singular hope to overcome the challenges associated with the increase of drug development costs and the poor fraction of investigated compounds making it though the development pipeline (i.e. the high attrition rate) [24].

Although genomics seems to depict a relatively *static* profile of the cell and of the individual e.g. when it is used for disease risk assessment, it can also translate the *dynamism* of the processes impacting health when used for microbiomics or in the context of circulating tumor cell sequencing or for pathogen identification. Transcriptomics, metabolomics, or proteomics are themselves drawing a uniquely dynamic portrait of cellular processes. These can assist in the detection and/or quantification of the myriad of processes which, from the DNA blueprint, may lead to disease onset or response to therapeutics. Although pre-natal or newborn whole-genome sequencing is generalizing and represents the first necessary step of a young individual in personalized health, discrete or even continuous multi-*omics* profiling will be fundamental in helping to assess and predict the results of the complex interactions between our innate biological features, our behaviour and environment. Personalized medicine applications extend "from prewomb to tomb" [15, 25, 26], an indication of the central role *-omics* technologies should play in future standard healthcare practices.

Many questions yet remain as to the scope, timing and impact of the outcome of *-omics* research and its technologies. Diagnostics and therapy effectiveness assessment, particularly relevant within the scope of this report, constitute two of the fields that are most likely to benefit from the growing understanding of disease mechanisms made possible by multi-*omics* investigations. To this day, IVMD applications are still widely limited by the small number of validated biomarkers, known to act as surrogate (i.e. proxy) for disease. While more than 150.000 scientific publications discuss the predictive values of thousands of potential biomarkers, only about a hundred of these markers are currently validated for use in clinical practice [27]. Numerous recent studies have yet underlined new *-omics* candidate markers for diagnosing, stratifying and monitoring the evolution of conditions for which IVMD had so far no relevance [28].

The *-omics* revolution discussed above thus promises broader and superior analytical and predictive IVMD tools and methods for personalized medicine than ever before. Clinical chemistry, hematology, immunology, microbiology and other classical analytic disciplines of laboratory medicine are increasingly supplemented by a torrent of data (and more and more information) that can support better health-related decision-making. IVMD applications are yet not only broader and more performant, they are also increasingly *decentralized*, a trend on which we elaborate in the ensuing section.

1.1.1.2 Towards direct-to-consumer IVMD applications at the Point-of-Care

As we mentioned earlier, IVMD is still mainly achieved through central-laboratory testing, in highly centralized and highly technical facilities, not without reminding the early days of computing (figure 1.1.2). Despite the astounding level of automation exhibited by clinical laboratories, a substantial amount of trained personnel is still required to oper-

ate the cumbersome and costly equipment that allows the multitude of high-throughput IVMD tests currently leveraged in routine medical decision-making. Time-to-result, result accuracy, reproducibility, connectivity and low costs have been the key drivers of central laboratory processes for more than two decades [11].

(a) (b)

Figure 1.1.2: (a) MIT computer laboratory (1965) *Photograph from the MIT museum* (b) Queen Elizabeth hospital, Birmingham, (2012) *Source:*[29]

Recently though, this IVMD paradigm started to shift towards the partial decentralization of in-vitro testing, bringing IVMD technologies closer to diagnosticians (e.g. general practitioners) or to the patient themselves. The American company Theranos is a good illustration of the disruptive capabilities of innovative decentralized IVMD solutions. The company shook the conventional US business model behind IVMD testing by introducing a proprietary microfluidics and diagnostics technology, allowing it to offer on-demand, cheaper in-vitro routine blood tests in local Wallgreens drugstores [8]. Another example, this time illustrating the emergence of consumer-driven *-omics* applications, is that of 23andMe, the California-based company that offers direct-to-consumer genetic testing for the determination of ancestry, and the identification of common gene variants associated a few dozen genetic disorders [30]. Despite the regulatory misadventures of 23andMe and the vivid controversies targeting Theranos [31, 32], both these examples attest of the momentum that is now driving a segment of IVMD towards Near Patient Testing (NPT) or Point-of-Care (POC) testing as well as towards on-demand, direct-to-consumer IVMD applications.

POC testing is defined as "[biochemical] testing at or near the site of patient care whenever the medical care is needed" [33]. The sought benefits of POC testing are the rapid or immediate recovery of in-vitro information (i.e. within seconds or few minutes) to help assess health status and provide decision-making support at the point of need. This in turn may benefit the outcome of ensuing medical care [34]. The POC market is growing fast and is expected to reach a global market value of $27.5 billion by 2018 [35].

"Point-of-Care" may describe various locations and settings such as the diagnostician's office, home, the patient bedside, paramedical support vehicles etc. which comes with its share of confusion. Hänscheid insists on the frequent misuse of the "Point-of-Care" qualifier [36]. The author denounces the inadequacy of the POC terminology to describe tests requiring laboratory (even basic) infrastructures when the targeted patient population and addressable diseases require testing in the most elementary environment, sometimes without technical assistance or in an entirely decentralized setting (e.g. home-based testing) . Hänscheid refers to three main IVMD test settings: the clinic, the community (local-

clinic, pharmacy, etc.) and home (figure 1.1.3). If we adopt Hänscheid semantics, then we must acknowledge the value of IVMD testing decentralization with regards to the specific setting under consideration. For instance, Weber et al. demonstrated the value of bedside hematologic testing on post-operative cardiac surgery patients at risks of hemorrhage [37]. The study asserted the deep interest and vivid hope of the clinical community for hospital bedside IVMD testing in cardiovascular health management [38]. Several manufacturers already propose bedside technologies for screening or monitoring conditions such as myocardial infarction or heart failure. In a local community setting this time, in England, a recent study showed the possible health and cost-effectiveness benefits of local-facilities IVMD testing for sexually-transmissible Chlamydia infections [39].

Figure 1.1.3: From left to right: Roche Cobas® 8800 molecular analyzer [40] ; p-BNC Point-of-Care analyzer [41] ; Alere Determine™ HIV 1/2 Ab/Ab disposable test strip [42].

NPT and POC technologies and applications also spark great enthusiasm for tackling global health challenges specific to low- and middle- income countries [43, 44, 45]. In this setting, POC has repeatedly demonstrated its potential for improving infectious disease screening and diagnosis and to positively impact treatment outcomes. Peter et al. recently showed how a simple lateral-flow strip, i.e. disposable test could help diagnose tuberculosis in HIV-patients (randomized controlled clinical trial in 10 African countries) and consequently reduce the relative risk of mortality at 8-week by 17% [45]. POC testing has enabled the percentage of African population screened for Malaria to go from 5% in 2000 to 45% in 2010 [46]. The local-clinic IVMD POC testing of CD4+ T cells in Mozambique similarly resulted in twice as many patients being diagnosed with the retrovirus and subsequently initiate therapy within half the time, compared to standard clinical practices [47]. Other examples confirm the potential offered by the integration of POC testing in standard healthcare practices in such settings. In order for this potential to be fully exploited though, numerous regulatory, operational and economical issues yet still remain to be solved [43]. From a technology perspective this time, *instrumented* POC solutions have sometimes been judged inadequate to the context specific to low- and middle-income countries. Their costs, poor maintainability, and above all the requirements for infrastructures and trained personnel have mainly been pointed at as limiting factors to their generalization [48].

As we will see in the next section, the technological trends animating the field of POC IVMD are yet mainly evolving towards an increase in complexity and instrumentation. We will yet elaborate on the opportunities arising to handle and leverage that complexity for the translation of new or more performant IVMD applications both in the Western World and in low- and middle-income countries. To support Hänscheid's claims we will from this point on refer to POC technologies or solutions to describe IVMD tests that may

be performed at the patient bedside or in a home-based setting, without the assistance of technical personnel.

1.2 Point-of-Care systems and enabling technologies

The greatest part of the new *-omics* applications and POC testing technologies we have been referring to so far has been made possible by advances in nanotechnology, microfluidics, and biosensing. Combined, progress in these three extensive fields have conditioned the advent of Lab-on-Chip (LOC) technologies and systems. LOCs are designed to embed the functionalities of central chemical or biological analyses laboratories on chips with footprints often reaching sizes down to a few centimetres square. They allow for routines tests to be performed using sample volumes commonly down to a few nanoliters. LOC technologies have thus supported the advent of POC testing, potentially empowering diagnosticians and patients to carry out automated screening, diagnostics or monitoring procedures outside of the clinical setting.

1.2.1 Lab-on-Chip

Figure 1.2.1: From left to right, bottom to top: Paper-based microfluidics [49] ; Alere Determine^{TM} HIV 1/2 Ab/Ab disposable test strip lateral-flow immunoassay [42] ; Continuous flow microfluidics [50, 51] ; mLSI [52], digital- [53], and droplet-microfluidics [54]

LOC devices ubiquitously rely on microfluidic technology to carry out on-chip processing of the biological sample (e.g. blood, saliva, etc.). Some of the most common microfluidic functions include the separation, pre-concentration, and isolation of the analytes of interest [55, 56, 57]. The operation of specific biosensing schemes then enables the

extraction of the relevant diagnostic or predictive information. LOC devices are offering increasingly more diverse, integrated and better performing IVMD analytical features (figure 1.2.1). From the early and now common lateral-flow immunoassays especially valuable in low-income countries [48], a significant number of LOC families have evolved towards complex, instrumented devices, relying on actuated flow motions (e.g. using pumps and valves) and advanced biosensing schemes (e.g. optical [58], electrochemical [59], etc.). They have demonstrated significant functional capabilities ranging from cell sorting [60], rare cell isolation [61], down to single-cell [62], or even single-molecule [63] analyses.

Recently, advances in microfluidic Very Large Scale Integration (mVLSI) [64], droplet- [65] and digital microfluidics [66] have opened up new horizons for LOC design, operation and applicability. For instance, based on high component densities [67] and microfluidic logic [68, 69, 70] mVLSI technology has driven the emergence of software-programmable LOCs. Analogously to their electronic microprocessor counterparts, these LOCs can be programmed to execute complex application-specific microfluidic functions. High-throughput and subnanoliter processing volumes make them particularly interesting for single cell genomic and proteomic analyses or for chemical and biological synthesis [64].

Digital-microfluidics (DMF) constitutes one of the other key technologies driving the design of complex, instrumented LOCs. DMF has evolved drastically since the formulation of its fundamental principles in the early 2000s. It allows the discrete manipulation of liquid droplets on a 2D electrode array by using high electric potentials [71]. The digital nature of the DMF control processes enables the reconfiguration of the DMF scheme depending on the use-case at hand (e.g. program different reagents and sample mixing protocols, droplet splitting, dilutions, washing, etc.) without necessitating any hardware redesign [66]. DMF has already demonstrated its potential in a wide variety of applications including detection by immunoassay, Polymerase Chain Reaction (PCR), proteomics, sample preparation before next-generation sequencing, etc. More recent examples include the integrated sample preparation and multiplexed evaluation of single-cell RNA transcripts [72] and single-molecules extraction and analysis [73].

1.2.2 Lab-on-Chip-based system design and integration challenges

Instrumented LOCs have extended the scope and performances of possible on-chip chemical and biological testing at the expense of increased system integration challenges and often of stringent instrumentation and control requirements. Although the integration of microfluidic and biosensing functions owed Lab-on-Chips their denomination, many instrumented LOC devices require external equipment, annihilating the anticipated size gains offered by the integration level of the LOC alone: Lab-on-Chips would still too often be better designated as chips-in-a-lab [74].

An illustration of these issues and an interesting approach to solving them was recently presented by McDevitt and colleagues [41]. The group developed a portable analyzer *platform* allowing the operation of a panel of immunoassays tests. A disposable cartridge containing a number of compartmented antibodies under blister (enabling multiplexed analyses) is first inserted into the analyzer. The analyzer then releases and motivates the cartridge-embedded reagents and biological sample through the microfluidic design integrated on the cartridge. The latter contains porous agarose beads, which once functionalized, enable the capture of the antibody-specific analytes from the sample. After an appropriate incubation time, an optical detection module allows signal recovery before analyte quantitation can be performed on the system's embedded computer. The nature of the reagents, the flow actuation profile and the data analysis of the system can be adapted in only a few steps for accommodating new use-cases (on the condition that the functional and performance requirements of these use-cases can be satisfied by the platform specifications). McDevitt and colleagues developed their concept after they had

claimed in an earlier paper the need for *modular platforms* in view of the fast pace developments characterizing the *-omics* era [74]. They particularly pinpointed what they qualified as the lack of *scalability* and *universality* of most LOC-based systems. Their pioneering work in addressing *system-level* design requirements for the engineering of POC/NPT systems was among the first to acknowledge the contextual dynamism and uncertainty that accompanies the IVMD paradigm shift.

McDevitt's portable analyzer seems to better qualify for NPT, i.e. local-community-based testing, rather than POC applications if we refer to Hänscheid's clarifications [36]. Erickson et al. more recently elaborated on the issues pertaining to the costs and complexity of LOC/instrument integration for POC, direct-to-consumer applications [75]. Erickson underlines how the need to design, validate, and maintain or upgrade a specific instrument for operating LOCs constitutes a roadblock to the generalization of POC IVMD systems in the consumer market. Erickson argues that smart mobile technologies may be transformative in addressing that challenge, a point of view we present and defend in the ensuing section.

Figure 1.2.2: Illustration of the integration challenges for complex LOC devices. Reproduced from [76]

1.2.3 Smart-devices and Lab-on-Chip-based IVMD testing

With more than 7 billion users worldwide, of which 70% reside in low- and middle-income countries, mobile phones represent the most widespread and most rapidly adopted new technology in the history of mankind [34, 77]. Smartphones in particular, represent today 61% of all mobile-phones in the United States. They have also remarkably made their way to the developing World, where they are projected to equip about 1 billion users by 2017 [78]. Smartphones' ubiquity is just as impressive as the computational power they harvest. Most of today's smartphones significantly outperform Deep Blue, the supercomputer that famously defeated Garry Kasparov at Chess in 1997[1]. Perhaps even more remarkable, is the way smartphones have been revolutionizing most of the key aspects of day to day life in Western countries: from entertainment to education or finance. Of particular interest for us, smartphones offer tremendous opportunities for facilitating the decentralization of healthcare and clinical research [79].

Smartphones and tablets are already providing widely adopted mobile-health/mHealth software solutions, ranging from image-analysis-based dermatologic evaluation [80] to applications helping patients to better adhere to their pharmacological treatment [81]. More

[1]The A9 System-On-Chip of an iPhone 6S can theoretically carry out up to 115.2 Giga Floating-point Operations Per Second (GFLOPS) whereas IBM's Deep Blue topped at 11.38 GFLOPS

recently, numerous physical activity tracking applications have emerged from the upcoming availability of smart-sensors, wearables, skin patches etc. allowing the monitoring of basic health-metrics such as heart rate, blood pressure, or body-mass index. These physiological signals are known to strongly correlate with disease severity for conditions such as hypertension or cardiac arrhythmias. They represent in those cases valuable surrogate biomarkers and validated clinical endpoints that can be evaluated to assess treatment efficacy or prognosis potentially *together* with in-vitro biomarkers. More recently even, medical grade imaging capabilities have been developed around smartphone platforms. A review of recent wearable and imaging technologies for smartphone-based applications and a comprehensive portrait of the possibilities and challenges ahead of mHealth is available in [79].

Of greater relevance for us, the ubiquity, functionality panel and computational power of smart devices make them strong candidates to cope with some of the instrumentation and control issues arising with the latest generations of LOC devices. Numerous successful examples of coupled smart-device/LOC systems have been described in the literature. Recent reviews of demonstrated smart-device/LOC proof-of-concept systems are available [34, 82].

Figure 1.2.3: Example of camera-based LOC/smartphone system. Reproduced from [83]

A great number of these systems rely on smartphone-integrated peripherals, most often their camera [84] (figure 1.2.3). Colorimetric detection-based systems have been utilized for applications ranging from the monitoring of pH level in sweat [85], chlorine concentrations in water [86] or C-reactive protein levels in blood [87]. Cell counting was also demonstrated using that same principle [88]. Smartphone-camera fluorescence-based systems recently enabled the quantitation of Prostate Specific Antigen [89], Giarda Lamblia cysts [90], E. Coli and salmonella [91] while also showing promise for flow cytometry [92]. Other optical detection methods such as chemiluminescence or electrochemiluminescence have allowed the quantitation of metabolites [93, 94], small reactive molecules [95] or hormones [96] while a smartphone-integrated optical microscopic diffraction analysis [83] recently helped in screening pre-cancerous and cancerous cells. Ozcan and colleagues

finally recently showed how smartphones could be used for reading Enzyme-Linked Immunosorbent Assays (ELISA) microplates [97].

System designs relying mainly on external instrumentation peripherals are also being widely investigated. Sia and colleagues [98] recently presented evidence of a highly functional, multiplexed, immunoassay-based dongle, successfully screening for HIV and syphilis from a finger prick in less than 15 minutes in a low-income country field setting (figure 1.2.4). They achieved sensitivity and specificities rivaling conventional central laboratory tests. Non-optical detection methods have also been documented: electrochemical impedance-based biosensing was recently demonstrated on a smartphone for the quantitation of bacteria concentration in water [99]. Finally a versatile electrochemistry-based detection system, relaying measurement data to the cloud for analysis via any first-generation mobile phone established the feasibility of chip-to-cloud systems in a field environment [78].

Although some of these systems do not involve control functions and make use of the sole smartphone camera for sensing readout purposes, we can anticipate that the increasing level of automation illustrated by software-programmable LOCs will be determinant in the definition of the functional role of smart-devices in next generation Smartphone/LOC systems. Recently, Li et al. [100] proposed an Android-based design allowing the control of a pneumatic unit embedding solenoid valves and pressure controllers. Suitable application-specific LOCs can be interfaced to this pneumatic unit and execute a predefined flow actuation scheme. The appropriate sequence of valve actuation and pressure sensor readout necessary to operate a given LOC is defined via a programming interface at the mobile Android-software layer.

Figure 1.2.4: Example of accessory-based LOC/smartphone system. Reproduced from [98]

1.2.4 Design challenges for direct-to-consumer smartphone-based IVMD

These later developments have thus partly confirmed the transformative potential of smart-mobile technologies for POC IVMD testing, at least as enablers for the development of integrated control and instrumentation solutions for next generation LOC devices. LOC/smartphone systems have undoubtedly nourished the enthusiasm for mHealth over the past few years and have certainly contributed to the massive increase in funding gen-

erated in the field [79]. The mHealth market value grows at an annual rate of roughly 55%, going from $1.5 billion globally in 2012 to an expected $21.5 billion by 2018 [101]. IVMD mHealth applications relying on recent advances in *-omics* technologies have recently made their apparition in the consumer market (e.g. QuantuMDx [102], Biomeme [103]).

Despite the promises offered by next generation state-of-the art LOC/smartphone systems, a second major roadblock could yet still hinder their success in the consumer market: their low frequency of use [75]. Unlike glucose monitoring, many IVMD tests need only be performed sporadically depending on the time-scale and clinical significance of the time-dependent fluctuations exhibited by the biomarkers of interest. As direct-to-consumer POC IVMD systems mainly rely on an instrument/consumable business model, the combined effect of expected low consumable sales volumes with aforementioned increasing system complexity may translate into high cost-per-test, and therefore jeopardize system adoption and success. A recent workshop organized by the US National Institutes of Health (NIH) underlined the associated challenge of assessing "when, where and for whom mHealth devices, apps, and systems are efficacious " [104]. Among the issues at stake, the NIH highlighted the problems inherent to the rapid development and obsolescence of new mHealth technologies ; obsolescence that is obviously tightly linked to frequency-of-use in specifying the cost-effectiveness of a system over its life-expectancy.

As we saw at the beginning of this chapter, LOC-based mHealth must address the challenges and opportunities offered by the fast-pace evolution of its enabling technologies and by the increasing variety of possible applications: mHealth does not only stand as a means to decentralize routine IVMD testing to assess an individual's health status but also opens up a whole range of possible new use-cases: in pharmacogenomics and personalized medicine for instance, by allowing the predictive testing of response to a personalized pharmaceutical treatment at the point of need (i.e. companion diagnostics), or by enabling the home-based monitoring of the safety and efficacy of that treatment over a certain period of time after the initial prescription. This particular application shows promises for helping to "reboot" the pharmaceutical industry, for example by enabling the implementation of a "guaranteed-to-succeed model" for drugs: a pharmacological treatment would only be paid for based on documented efficacy via smartphone-based IVMD monitoring of relevant surrogate biomarkers [24]. Beyond the possibilities for remodelling our healthcare model, direct-to-consumer smartphone-based IVMD systems have started revolutionizing the way clinical research is carried out by enabling remote patient enrolment in clinical trials, etc. [79, 24]. The recent release of the Apple ResearchKit Application Programming Interface (API) is a testament of the interest of major technology players in mHealth and of the growing accessibility of tools allowing the implementation of new mHealth use-cases, including in the field of IVMD [105].

1.3 Summary

We acknowledged in this chapter that the advent of personalized medicine, together with the decentralization of IVMD testing to the consumer-market, hold great promise for the future of preventive and prescriptive medicine around the globe. We argued that the emergence of IVMD at the POC was driven by the fast-pace evolution and growing complexity of LOC devices and of their enabling technologies. Although smart-mobile devices stand as a great vector for facilitating the decentralization of IVMD to the point of need, the design of next-generation smartphone-based IVMD systems will need to anticipate for the emergence of new healthcare and clinical research IVMD applications. It will also need to account for the uncertainties associated with the incomplete, yet growing knowledge-

base in human health brought by advances in the varied *-omics* disciplines. It finally will need to solve the issues raised by the expected low-frequency of use and threat of fast obsolescence of next-generation systems.

The collection of requirements and the evaluation of the sociotechnical context that should initiate the design of new LOC/smartphone systems may thus reveal to be complex, incomplete and uncertain. We suggest that de-novo system design should be avoided, promoting instead reuse and incremental system evolution in order to limit implementation and validation costs as well as to reduce time-to-market [106] (appendix A). We recommend LOC/smartphone systems to demonstrate system *evolvability*: "the design characteristic that facilitates more manageable transitions between system generations via the modification of an inherited design" [107]. System evolvability should thus minimise the costs and efforts associated with system redesign and with development iterations of hardware, software and interfaces. It should facilitate system maintainability and mitigate the risks of fast system obsolescence. We elaborate extensively on system evolvability in the ensuing chapters.

Chapter 2

Introduction

In view of the challenges hindering the generalization of direct-to-consumer POC systems as discussed in the previous chapter, we articulate this thesis around the central concept of *system evolvability* in LOC/smart-device systems. More specifically, we investigate how system evolvability can be incorporated early in system design, in anticipation of needed change in successive system generations. We strive to:

1. Identify candidate frameworks that may support the design and development of evolvable systems.

2. Investigate the mechanisms, within this/these framework(s), which may promote system evolvability.

3. Develop an evolvable system prototype illustrating the implementation of such mechanisms. This particular instantiation should enable the interfacing of LOCs embedding Silicon-Nanowire biological Field Effect Transistors (SiNW-bioFETs), a promising biosensing technology for the detection of biological analytes at ultra low concentrations.

4. Choose or derive metrics allowing the assessment of system evolvability. These metrics should support decision-making early in the system concept exploration phase of the system development process. They should enable cost/utility tradeoff analysis of system evolvability.

Chapter 2 deals with question 1 and 2 and present the opportunities offered by model-based systems engineering for facilitating the design of evolvable systems. Chapter 3 presents extensively the SiNW-bioFET technology and the challenges associated with its sensitive instrumentation. Chapter 4 describes the design of an evolvable system satisfying the requirements of question 3. It also contains details on how the mechanisms of question 2 were implemented in the prototype. Finally chapter 5 presents a metric and methodology for assessing some of the intrinsic properties of system evolvability.

Chapter 3

An engineering systems framework for evolvable Point-of-Care In-Vitro Diagnostics

The portrait we drew of the IVMD paradigm shift in the first chapter of this thesis depicted three important trends: First, the emergence of new POC applications and the decentralization of targeted IVMD market segments make the field of IVMD an extremely *dynamic marketplace*. The concurrent personalization of medicine reinforces this dynamism with the customization of IVMD POC products and services. Second, the fast-pace *technological evolution* animating the foundational building blocs of POC IVMD systems, i.e. biosensing, microfluidics, LOC technologies and smart-devices, challenges the design of comprehensive and competitive systems which must keep pace with the short half-life of their constituents. Third, the *complexity* of POC systems and the *variety of environments*, e.g. the amount of inter-connected systems in which POC technologies must be integrated (Electronic or Personal Health Records, clinical trial data repositories, etc.) are increasing (figure 3.1.1).

As such, next-generation instrumented LOC/smartphone-based systems for POC IVMD do not qualify for traditional engineering approaches. Their vast socio-technical complexity calls for the adoption of an engineering framework that will not only help addressing their large multidisciplinary nature: from biophysics and life science to cloud-computing services, but also account for the dynamism and uncertainties that characterize POC consumer needs and the direct-to-consumer POC IVMD operational environment. Business and engineering strategies are needed not only to streamline system development and reduce initial time to market, but also to ensure that systems in operation or successive generations of systems can accommodate for change in stakeholder needs, adopt new technology or adapt to changing environments in a timely and cost-effective manner [108, 109].

This chapter introduces the field of systems engineering (SE) and presents the opportunities it offers for guiding the development of complex engineering systems. We will argue on the adequacy of the SE framework for engineering LOC-based POC systems and elaborate on some of the SE strategies that may help addressing the changing contexts and requirements that characterize the field of IVMD testing. We will particularly discuss system evolvability and how to incorporate it within the initial system design.

3.1 Systems engineering and engineering systems

3.1.1 Engineering systems

The International Council of Systems Engineering (INCOSE) defines a *system* as a "combination of interacting elements organized to achieve one or more stated purposes" [110]. After the end of the industrial revolution and the epoch of the *great inventions and artifacts*, many systems started to combine advances made in various technological fields together, in an attempt to offer increased performances or sometimes entirely new functionality. The epoch of *complex systems* had begun. Kossiakoff defines *complex systems* as systems "composed of a multiplicity of intricately interrelated diverse elements and requiring systems engineering to lead their development " [111, chap. 1]. Multiple views exist how to characterize the degree of complexity of complex systems [112]: from the amount of information they contain [113], to the number of tasks they involve [114] or the *connectivity* existing between the various elements composing them. [115, 116].

Figure 3.1.1: POC system context illustrating the need for inter-connectivity with a variety of external systems (e.g. Personal Health Record, etc.) and the sociotechnical complexity implied by the variety of stakeholders directly or indirectly associated with the POC system.

Complex systems are thus defined from the complexity of their *internal* composition and arrangements. More recently, the large societal challenges that arose at the end of the previous century led to the expansion of the concept of complex systems to that of *engineering systems*. Engineering systems possess the attributes of complex systems but are developed in the realm of the vast scope and complexity of their *interactions* with their users and their environment (figure 3.1.1). *Engineering systems* are defined as the "class of systems characterized by a high degree of technical complexity, social intricacy, and elaborate processes, aimed at fulfilling important functions in society" [117, chap 2].

One of the questions that arises at this stage is: Do LOC-based POC systems qualify as engineering systems? First of all, we believe that the socio-technical complexity we attempted to demonstrate in the previous chapter is a clear indicator that they do. In order to confirm this assertion, we may investigate whether these systems present the five

characteristics that De Weck et al. consider for the identification of engineering systems [117, chap 2]: First, LOC-based POC systems are "real-world" systems: they possess physical parts such as the LOC or the very biological sample they intend to extract information from. Second, they are heavily artificial - since engineered - systems. Third, they possess numerous dynamic properties: properties such as the biological target of interest, or the instrumentation sequences for various LOCs are bound to change over time if a new analyte is investigated. Fourth, they comprehend hybrid states: the continuous range of concentration that an analyte may take, versus the discrete decision tree branches that may result from an IVMD test. Finally fifth, they are under the control of human actors: the end-user. We can consequently consider LOC-based POC systems as *engineering systems*. As such we justifiably investigate their design process with a systems engineering perspective.

3.1.2 Systems engineering

The first definitions of SE were formalized in the 1970's, together with the first U.S military standard [118]. Although not all consensual, these definitions all share, what are still today, the view of SE as a holistic, synthetic, interrelationship-oriented discipline aiming at "guiding the engineering of complex systems" [111, chap. 1]. INCOSE cemented the definition of SE as "an interdisciplinary approach and means to enable the realization of successful systems. [SE] focuses on defining customers needs and required functionality early in the development cycle, documenting requirements, and then proceeding with design synthesis and system validation while considering the complete problem" [110].

From an *engineering systems* standpoint, "a fundamental goal for systems engineering is to maximize the perception of system success by stakeholders" [119]. SE differs from traditional engineering in its holistic perspective on system design. The focus is put on the interactions and interfaces between the various sub-systems and components rather than on the detailed design of the individual components themselves. SE therefore bridges conventional engineering disciplines together. It also encompasses the analysis of the relations between complex systems and their environment, users and interconnected systems. It relates the internal articulations of a complex system to its intended operation.

SE practices are today often adopted to address the same challenges they were originally conceptualised for:

Advancing technology: SE is key in designing and implementing *innovative* complex systems that rely on sub-systems or components that offer novel, superior functionality or performances. It then helps mitigate the known and unknown risks associated with incorporating elements that may still bear unknown properties or behaviours or that simply do not benefit from lessons-learned form past designs.

Competition: SE must help designing competitive systems often by leveraging incomplete knowledge, and sometimes conflicting objectives. Tradeoff-analyses are therefore central to SE practices. They can be applied in various scenarios: early in the project initiation, they may offer valuable decision-making support for the selection between system design alternatives. Later on in the design process, they can help balancing some of the essential requirements of system development such as cost versus time or performances.

Specialization: One of the most significant objectives of SE is to help analyze, specify and validate the interfaces articulating the different sub-systems or components: SE oversees system partitioning, and functional allocation. These activities alone justify SE practices since handling interfaces goes beyond the expertise and prerogatives of traditional specialty engineers. Good SE practices are thus essential in order to subdivide a complex system into independent building blocks or *modules*. As we will see later on in this manuscript, *modularity* is a key design principles to abide by in order to ease system integration, testing, operational maintenance or system upgrade.

3.1.3 Conceptual design of engineering systems

The SE framework applies from the very early phases of a system's lifecycle. The identification of a societal need or of a technological opportunity may already fall under the prerogatives of the systems engineer. From that point on, SE tools and methodologies exist to support all system development phases until the system is put in operation, further upgraded or maintained, until it eventually becomes obsolete and is withdrawn or replaced. There are numerous conceptual frameworks describing the system development process and the system lifecycle. From public government agencies to large companies and across different system domains, these frameworks can yet often be mapped to the life cycle model presented in figure 3.1.2. This overarching model is often referred to as the *systems engineering life cycle model* [111, chap 4].

Figure 3.1.2: Systems engineering life cycle. Reproduced from [111, chap 4]

Within the context of complex systems though, this model presents several limitations. One of the most important is that it does not reflect the feedback loops and typical developmental iterations that often characterise complex and software-intensive systems (e.g. the spiral development model [120]). We will from now on assume that these iterations are an integral part of the development process when needed and will not elaborate more on iterative design frameworks. We must underline that even though the focus of this thesis will mainly be put on *concept development*, we will tackle issues specific to the iterative redesign process, that may occur later on in the *post-development* stage.

The *concept development* stage of the SE life cycle encompasses the analytical and feasibility studies that should initiate the design of a complex system. This includes exploring candidate concepts and defining performance requirements. Importantly, this stage also embodies the *system architecting* activities: the translation of needs (including but not limited to requirements) into specifications and the development of an abstracted concept of the system [121, 122].

3.1.3.1 Systems architecting

The development of a system architecture is crucial for highly complex systems, especially if the system concept is not familiarly known among the developing team. System architecture is a broader concept than that of a system structure: it does not only inform about the composition of elements and interconnections between the elements of a system but also represents the assignments of functions to the systems elements [117, chap 2]. System architecting will not foster a complete set of specifications for the system, but it will determine some of its most substantial characteristic features such as its level of

integration, or on the contrary, the *modularity* that the system will exhibit [112]. Architectural patterns such as *layers* (e.g. the seven layers of the Open Systems Interconnections model [123, 124]) or *hierarchical trees* are also some of the most common concepts used in systems architecting. These features may characterise the overall system or, down in the detailed system decomposition, some of its sub-systems or components.

Modelling tools and methodologies can provide significant support for the architecting process (they will be discussed in an ensuing section): they can provide standardized constructs and rules for the initial formal definition and communication of the system concept (including its architecture). Such models can be subsequently further refined and assist the *engineering development* and *post-development* stages of the systems engineering life cycle.

3.1.4 The *-ilities* of engineering systems

The *epoch of engineering systems* puts significant importance on what are referred to as the *non-functional -ilities* of engineering systems. *-Ilities* are defined as the "desired properties of systems [...], that often manifest themselves after a system has been put to its initial use. These properties are not the primary functional requirements of a system's performance, but typically concern wider system impacts with respect to time and stakeholders than are embodied in those primary functional requirements."[117, chap 4].

The late days of the industrial revolution and the early years of the epoch of complex systems witnessed the emergence of the first *-ilities*: *safety* and *quality* which today still constitute a significant part of the overall engineering efforts when designing modern engineering systems. Hand in hand with the development of safer and more reliable complex systems, *usability* and *maintainability* appeared in the list of what are today considered the *traditional -ilities* of engineering systems. The latter two properties are still today the object of vivid research and developement.

Although it is the growing internal complexity of systems that conditioned the advent of the traditional *-ilities*, it is the accelerating rate of change [125] combined with the great social challenges of the last century that has led to the emergence of "advanced" *-ilities*. In a complex socio-technical context, POC-systems cannot succeed by the sole satisfaction of their design-time testing sensitivity and specificity requirements (n.b. linked to *quality*). Next generation IVMD POC system will need to keep up with fast pace technological progress animating the field of life science discussed in chapter 1. They will need to do so while managing the *uncertainties* intrinsic to our incomplete knowledge of human health in its broadest sense. IVMD system designers will also need to account for the shift in focus towards direct-to-consumer applications [106]. It is for such uncertain and somehow chaotic environments that advanced *-ilities* are increasingly accounted for through the design of today's engineering systems.

Advanced *-ilities* aim at "maximizing the net perceived value for the system stakeholders" while the system will need to respond to *perturbations*: disturbances, context shifts, or changes in stakeholder needs [119]. Two key system development strategies have been investigated in this perspective: *robustness* (survivability) which consists in ensuring a system will maintain its value in a changing context *without changing itself* and change-related *-ilities* that aim at providing systems with the ability to change. Change-related *-ilities* should increase system responsiveness to needed change, reduce long-term costs associated to system upgrade when dealing with changing requirements or a changing environment. We will focus our interest on this latter strategy.

Figure 3.1.3: Frequency of *-ilities* mentioned in journal articles and Google hits on the internet for a search extending from 1884 to 2010. Reproduced from [126]

3.1.4.1 Change-related *-ilities*

Change-related *-ilities* constitute means to mitigate the risks associated with uncertain environments. Their objectives is to help engineering systems "sustain value throughout their life-cycle in foreseen uncertain context and requirements" [119, 107]. Despite this noble purpose, change-related *-ilities* do not come without a certain number of challenges. Their consideration entails vaster design spaces and often a greater number of design alternatives to pick from in the earlier phases of system development. Change-related *-ilities* such as flexibility or modularity therefore increase the design complexity at the origin of some of the issues they intend to tackle in the first place. Furthermore, the formalisms and systems engineering concepts, tools, and methods addressing change-related *-ilities* rely on terminologies that are not consistent throughout the systems engineering community and require clarification:

As we mentioned before, change-enabling strategies for engineering systems all aim at *value sustainment/robustness* in the eyes of the various engineering systems stakeholders. *System robustness* only represents one of the strategies to achieve value robustness. We assume that in a fast technological evolution paradigm, robustness is not likely to sustain value throughout the system life cycle. We instead focus on a system's ability to change or rather to be changed [107].

De Weck et al. recently collected the number of occurences of the most common traditional and change-related *-ilities* and drew the associations commonly made between each of them [126]. Their study unravelled the intricacy of the semantics of these *-ilities* and the incongruence and misconceptions behind their use and prescriptions in the literature.

An interesting conclusion of De Weck's study is the establishment of formal means-to-end relationships between these *-ilities*, and the derivation of a hierarchy of *-ilities* according to Ross's semantics [119]. The ultimate *end* was set to be value robustness. Among the means to reach this end, system *evolvability* appeared as one of the most enabling, vastly overarching *-ility*, at the top of a hierarchy of other *means -ilities* such as modularity, scalability or reconfigurability (figure 3.1.4). These lower-level properties can be thought of as *architectural -ilities*: they constitute mitigation or exploitation means that can help reaching higher-level *system effectiveness -ilities* such as evolvability or

Table 3.1: MIT Systems Engineering Advancement Research Initiative (SEARi) group definitions of change-related *-ilities*

-ility	Definition
Value Robustness	ability of a system to maintain value delivery in spite of changes in contexts or needs
Robustness	ability of a system to maintain its level and set of specification parameters in the context of changing system external and internal forces
Changeability	ability of a system to alter its form, and consequently possibly its function, or operations, at an acceptable level of resource expenditure
Flexibility	ability of a system to be changed by a system-external change agent with intent
Adaptability	ability of a system to be changed by a system-internal change agent with intent
Evolvability	ability of an architecture to be inherited and changed across generations (over time)
Survivability	ability of a system to minimize the impact of a finite duration disturbance on value delivery
Versatility	ability of a system to satisfy diverse needs for the system without having to change form (measure of latent value)
Scalability	ability of a system to change the current level of a system specification parameter
Modifiability	ability of a system to change the current set of system specification parameters
Interoperability	ability of a system to effectively interact with other systems
Reconfigurability	ability of a system to change its configuration (component arrangement and links)
Agility	ability of a system to change in a timely fashion
Extensibility	ability of a system to accommodate new features after design

changeability [119]. Consequently, the frameworks, methodologies and tools of systems engineering need to promote the embedding of low-level architectural *-ilities* in engineering systems, which will, as a consequence, favour value robustness.

Although De Weck et al.'s proposed hierarchies are not consensual, they fostered a series of research work that later on complemented their study. Beesemyer [107] and Fulcoly [127] both investigated the promises of *evolvability* from an empirical and normative standpoint respectively. Their comprehensive semantical and ontological research led to prescriptive semantics on which this thesis work heavily relies (table 3.1).

3.1.4.2 System evolvability

Semantics Beesemyer and Fulcoly's works cemented a definition and ontology for system *evolvability* that present the system property as a cross-generational design-stage characteristic. Their view (that of the Systems Engineering Advancement Research initiative (SEARi) group at the Massachusetts Institute of Technology) joins that of McManus and Hastings who saw system *evolvability* as the "ability of the system to serve as the basis of new systems (or at least generations of the current system) to meet new needs and/or attain new capability levels" [128], and that of Butterfield et al. who see system architecture evolvability as "the ability of the architecture to handle future upgrades" [129]. This cross-generational perspective mainly differs from one alternative view, such as that of Christian et al. [130], who see evolvability as a single-generation life-cycle property: "the capacity of a system to adapt to changing requirements throughout its lifecycle without compromising the integrity of the system" .

We will from now on consider system *evolvability* as conceptualized by Beesemyer as "a design characteristic that facilitates more manageable transitions between system generations via the modification of an inherited design and can be defined by the ability of an architecture to be inherited and changed across generations [over time]" [107].

(a) (b)

Figure 3.1.4: Two propositions of means-end hierarchies of change-related *-ilities*. Reproduced from [126]

Ontology Adopting clearly defined semantics is a required first step to the design of value robust systems. The definitions proposed by the SEARi group for change-related *-ilities* enable the consideration of suggested ontologies for the development of evolvable systems (table 3.1) [126, 107]. System *evolvability* overarches most of the *means -ilities* favoring *changeability* (figure 3.1.4a). It can also be viewed as a parent-property of *changeability* (figure 3.1.4b). For either of both ontologies, the *means -ilities* of *changeability* should thus also promote system *evolvability*.

Beesemyer reviewed some of the most important earlier works on the design principles suggested for favouring changeability and/or evolvability. One of the first is that of Steiner, who emphasized the importance of "enduring architectures" for favouring the evolvability of products over several generations [131]. Steiner accentuated the crucial importance for enduring architectures to accommodate for "addition, extension, and adaption for use" from one generation to the next. The author thus introduced some of the key abilities we mentioned earlier: *modularity, extensibility*, and *scalability*. Fricke and Schultz later on proposed design principles for promoting changeability [108]. Although the semantics of the *-ilities* the authors relied on, i.e. flexibility, agility, robustness and adaptability, differ from that of the SEARi group, the definition of the overarching *changeability* property is similar to De Weck's. The design principles suggested by Fricke and Schultz to accomplish changeability are therefore relevant for enabling evolvability as well.

The authors distinguish a set of basic principles: *ideality/simplicity, independence, modularity/encapsulation* from a set of extended principles: *integrability, autonomy, scalability, non-hierarchical integration, decentralization, redundancy* (table 3.2). The suggested candidate *means -ilities* of Beesemyer (based on the semantical clarification of De Weck) partly overlap with these basic and extended principles.

The *means -ilities* of evolvability Among the intersecting subset just discussed, one of the most intuitive candidate is probably *modularity*. *Modularity* is one of the foundational principles for the design of systems capable of change. Laying at the bottom of the

ontological tree of value robustness, modularity has been widely acknowledged as a key enabler to engineering change [132]. Design for modularity consists in clustering systems functions into modules. The coupling between these modules should be minimized while the integrity within modules should be maximized. This coupling most often depends on the design of the *interfaces* associating modules between them. *Modularity* is powerful concept only if change to a module does not propagate to other modules and if the interfaces between modules remained unaltered. Changing interfaces especially, is often cumbersome and costly [112]. Lindermann et al. differentiate "local changes " which are changes confined within a module from "interface-overlapping changes " which often become inevitable as the system grows in complexity and connectivity [133]. When properly designed, modular system architectures can facilitate the re-use, exchange or adaptation of modules to perform new or more performant functionalities. *Modularity* is tightly coupled with the principle of *independence*: one of the axiomatic principles of design [113, 108]. If system modules are independent, i.e. if there is no coupling between them, then change to a module will not propagate to other modules, limiting the scope and hence costs and efforts of reengineering. The design of modular systems can be supported by the use of Design Structure Matrices (DSM) to first assess and then attempt to reduce the dependencies (and therefore coupling) between the structural or functional blocks composing a system [134, 135, 136, 116, 137]. We elaborate on DSMs and their potential role in assessing both *modularity* and *scalability* in chapter 5. Another useful tool for assessing modular independence is Change Propagation Analysis (CPA). CPA supports the evaulation of the impact of changing a component or module on the scope of the reengineering effort that will be required for changing the other affected system modules [138, 139]. Eckert et al. introduced CPA with the useful concepts of change-*absorbers*, *carriers* and *multipliers*, which represent the behavior of a system module affected by change [140].

Table 3.2: Candidate design principles favoring evolvability

Fricke and Schultz's design principles for changeability [108]	Beesemyer's candidate *means -ilities* for evolvability [107]
Ideality/Simplicity	Targeted modularity
Independence	Integrability
Modularity/Encapsulation	Scalability
Integrability	Decentralization
Autonomy	Redundancy
Scalability	Reconfigurability
Non-hierarchical integration	Leverage ancestry
Decentralization	Mimicry
Redundancy	Disruptive architectural overhaul
	Resourceful exaptation

Integrability is a vast topic of discussion: it engulfs *compatibility* and *interoperability*, which call for generic, standardized, open interfaces between structural or functional modules. *Scalability* will be discussed further in chapter 5. It presents one of the most easily quantifiable metric for the evaluation of evolvability: the span of one or more system specifications. *Decentralization* implies the distribution of control, information and resources, and finally *redundancy* involves duplicating resources or information.

3.1.5 Design frameworks and methodologies

The semantics and ontology discussed in this section have pinpointed several key design principles to be embedded in engineering systems in need of evolvability. Although we mentioned a few tools in order to facilitate the implementation of these principles, these still need to be incorporated within formal frameworks and methodologies in order to systemize the process of designing systems for evolvability. Cloutier and Verma [141], define *framework* as "a logical structure or an organizational skeleton used to classify concepts, terminology, data, artifacts, etc." The semantical and ontological works we have been relying on so far disclose any attempt to formalize such framework for designing systems for evolvability, perhaps an illustration of the intricacy of the matter [107].

Aside from the frameworks, few *methodologies* have been developed with the objective of delivering changeable or evolvable systems. Friedenthal et al. [142], define methodology as "a set of related activities, techniques, and conventions that implement one or more processes and is generally supported by a set of tools". Cardin et al. propose a comprehensive review of design procedures aimed at incorporating *flexibility* in engineering systems, where flexibility is defined as "enabling a system to change easily in the face of uncertainty", which suggests a partial overlap with our goals [143]. The authors acknowledge the contribution of several researchers aiming at similar objectives but highlight the incomplete nature of existing frameworks. Cardin et al. thus introduce a five-step methodology which they claim covers the all necessary phases of design process. One of the peculiarity of their approach is that it initially requires the consideration of a *baseline design*, that does *not* embed flexibility but should merely serve as a reference point in the design space to nourish the design process for flexibility in the subsequent development phases.

The authors, just like several of the work they reference, then insist on the evaluation of a priori knowledge and the use of *uncertainty recognition* techniques such as scenario planning in order to steer the design process towards the systems that show the greatest promise to overcome the challenges of their dynamic and uncertain environments [144, 143]. A review of uncertainty recognition techniques is out of the scope of our investigation.

The third phase of Cardin et al.'s methodology is that of *concept generation* which, according to the author, should comprise a strategy and enablers of flexibility. We will neglect the strategic aspects of concept generation but rather focus on the enablers, which are nothing but implementations of the *means -ilities* we discussed earlier. Cardin et al. suggest to rely on the early design specifications, or on an early *model* of the system to analyze the interfaces or dependencies between sub-systems, components or design variables and unravel opportunities for incorporating flexibility in the system. We undertook that modelling effort and elaborate on it later on in this chapter.

Cardin et al. then elaborate on the vast topic of *design space exploration* which should help select the best design alternative according to pre-defined optimality criteria, one of which here being flexiblity. Among the methods reviewed by the author, a particularly interesting approach is the Multi-Attribute Tradespace Exploration (MATE). MATE allows the computationally-efficient determination of the Pareto set of designs that will maximize system *utility* (defined for specific stakeholders) as a function of *costs* [145]. The method is particularly relevant to the the selection of changeable design alternatives: the *filtered outdegree* metric, introduced by Ross, represents the quantification of changeability for a given design in the trade space [146, 147]. Ross also developed metrics for the quantification of scalability, modifiability and robustness which, just as the filtered outdegree, require the prior *valuation* of the change-related *-ilities* using *utility theory* [145].

The fifth and last phase of Cardin's methodology addresses *process management* and will not be discussed further.

Several methodologies, including Cardin's, thus propose a systematic approach to design for flexibility and may constitute the basis of a formal approach to design for evolvability. We concur with Cardin's beliefs on the importance of adopting formal uncertainty recognition methods before undertaking conceptual design, on the need for metrics and valuation methods for change-related *-ilities* as well as the criticality of using decision-support methods in order to trade off the costs of changeability against value (e.g. utility) (figure 3.1.5).

Figure 3.1.5: Degree of changeability vs. sources of cost. Reproduced from [148]

Although we acknowledge the relevance of these considerations, we will not address uncertainty recognition or design space exploration within the scope of this thesis work. Instead, we assume prior assessment of the arguments in favour of design for evolvability and focus on *conceptual design*. In particular, since established systems engineering frameworks heavily rely on *models*, we investigate how model-based systems engineering (MBSE) and more particularly model-based design (MBD) can be leveraged to manage development complexity while promoting and help assessing evolvability.

3.2 Model-based systems engineering for evolvable systems

3.2.1 Model-based systems engineering

MBSE aims at formalizing the development of systems through the use of models. According to Rumbaugh et al. a model is a "representation of a selected part of the world, the domain of interest, that captures the important aspects, from a certain point of view, simplifying or omitting the irrelevant features" [149]. MBSE can be applied throughout the multiple development phases of the systems engineering life cycle. It often requires relying on multiple physics in order to represent the system comprehensively, via multiple viewpoints. MBSE's primary objectives are to manage system complexity, mitigate risks and assess system function via virtual prototyping (i.e. modelling), and facilitate system-related communication among stakeholders. MBSE appeared with and matured since the early days of computing, progressively enabling systems architects and systems engineers to distance themselves from document-based SE to embrace a model-based approach to system development. In that perspective MBSE has thus been following the footsteps of some of the traditional engineering disciplines it overarches, such as mechanical or electrical engineering. MBSE today addresses a wide variety of application domains, from defence and aerospace to energy systems and transportation. Within healthcare, MBSE

was recently acknowledged as one of the the keys to achieve high assurance in system software, interoperability, context-aware intelligence, autonomy, security and privacy, and device certifiability for life-critical, context-aware, networked systems of medical devices [150].

Most of the current modelling tools targeting MBSE practices include graphical modelling modules, partly to address the communicative objectives we were just referring to. Among them, the SysML language stands as one of the most institutionalized to this day. SysML finds its origin in the advent of object-orientation in computing in the 1970's [118, 151]. Object-orientation translated a few years later in the standardization of the Unified Modelling Language (UML), which became central for the analysis and design of software. In 2007, SysML was released as an extension of UML in an attempt to address the limitations of the software-specific language for modelling engineering systems [152]. SysML is not bound to any specific system development framework, which has probably contributed to its institutionalization and growing acceptance. Several SysML modelling methodologies, even though not standardized, are relatively widespread in the MBSE community. A brief review is given in [118]. The modelling activities undertaken within the scope of the third chapter of this thesis were pursued following Wielkiens' SYSMOD methodology [152]. These activities are described later on but mainly addressed the *conceptual design phase* of the SE life cycle.

Although its lack of semantics brought SysML some justified criticism [153], several academics and industrials have leveraged the language to carry out the development of engineering systems of various sizes and complexity. Numerous SysML-based SE projects are still rooted in defence and aerospace [154, 155, 156]. Several initiatives have yet demonstrated the growing interest of the mechatronics and Cyber-Physical Systems (CPS) communities for MBSE, including SysML-based development. Such interest is perhaps due to the increasing level of integration of SysML environments with simulation tools such as Matlab or Simulink [157, 158, 156]. These domains are obviously very relevant to instrumented LOC-based systems: CPS extending from biological compounds and living organisms to cloud computing and Artificial Intelligence (AI).

3.2.1.1 SysML modelling of cyber-physical systems

CPS are defined as the "systems that offer integrations of computation, networking, and physical processes" [159]. The interest of the research and industrial communities for new MBD methodologies for CPS originates from the ever increasing demands for performances and the complex usage patterns of closely interacting cyber- and physical components [160]. LOC-based POC systems as they are envisioned for integrated and connected direct-to-consumer applications present most of the characteristic attributes of CPS: high degree of automation, networking at multiple scales, integration at multiple temporal and spatial scales; and reorganizing/reconfiguring dynamics [161].

The modelling, simulation, model-checking and design synthesis of CPS is accompanied by a number of challenges including the heterogeneous nature of CPS, the concurrence of different cyber- and physical processes, and potentially real-time requirements. Existing CPS modelling tools are also suffering from poor realism on the mechanisms for interacting with sensors and actuators [162]. To this day a unifying framework overarching all the activities associated with the design of CPS and addressing the aforementioned issues is still lacking. Numerous recent promising advances are reviewed in [160]. Such a unifying framework will not only need to enable the satisfaction of the performances and traditional *-ilities* of CPS (e.g. schedulability, reliability, security, etc.) but should ideally also address the change-related *-ilities* that characterize the dynamism and uncertainty that surround modern CPS.

Among the significant number of tools, methodologies and frameworks that have been

developed over the past few years, SysML holds a peculiar place. As we mentioned earlier, its lack of semantics prevents it from enabling the refined modelling of advanced CPS. However, SysML engulfs several Models of Computations (MOC) (e.g. discrete-events, finite state-machines, etc.) and an increasing level of integration with numerical modelling tools (e.g. Mathematica, Matlab, Modellica) which confers it a significant advantage to address heterogenous domains modelling: SysML can be a valuable tool for systems architecting.

Efforts dedicated to the integration of SysML with numerical simulation tools date back to the early days of the OMG standard. Johnson et al. pioneered in presenting a way to represent continuous time phenomena in SysML through integration with Modellica [157]. More recently, Sakairi et al. demonstrated the advanced integration capabilities of the IBM Rhapsody SysML modelling suite with Simulink and showed the potential for applications in control systems [158]. Nakajima et al. proposed an elegant way of circumventing the inadequacies of SysML current specification for the modelling and simulation of hybrid systems: those that require the concurrent representation of continuous and event-based processes. The authors developed a co-analysis framework where the SysML SE capabilities are joined with the strong analytical capabilities of Simulink for continuous systems [157].

These developments illustrate the awareness and urgency of the CPS community in integrating MBSE practices with control, mechatronics, and other traditional engineering disciplines. This brief review of recent works on the topic shows the promises for formally-designed, model-checked, validated cyber-physical engineering systems.

As the main focus of this thesis goes to systems architecting and design principles for change-related *-ilities* in instrumented-LOC CPS systems, little importance was given to the issues of under-constraint and non-determinism of the SysML language. Rather, we relied on the various MOC and representative capabilities of SysML to architect a POC system based on the Silicon Nanowire biological Field Effect Transistor (SiNW-bioFET) technology. Although we acknowledge the potential for integrated design (i.e. including model checking, validation, synthesis), our main interest goes to the conceptual design phase of the systems engineering life cycle and more particularly to the systems architecting activities supported by MBSE. We introduce the technological opportunity offered by SiNW-bioFETs in the following chapter and present our SysML architectural endeavours for the design of a *platform* allowing the interfacing of the sensing technology in chapter 4.

3.2.2 Evolvable platform-based design

The personalization of medicine we were referring to in chapter 1 can be considered as the extension of the mass customization paradigm to the healthcare sector. Mass customization started in the 1990s. It aimed at serving the needs of individual customers (or patients) through high product variety [163]. In response for the increasing demand for personalization, to marketplace globalization, to the proliferation of niche markets, and to increased competitive pressures, industries had to adopt new frameworks and methodologies in order to decrease product development time and the costs of customization [164]. This period corresponds to the emergence of both lean manufacturing [165] and *product platforming* [132, 166, 167]. Product platforming aims at saving costs by sharing a common set of components, modules, and/or subsystems across various product derivatives [168]. The *product family* encompasses the group of related product variants. The core elements shared across the platform family constitute the "islands of architectural stability" that Percivall so elegantly formulated [169].

There is thus an intrinsic tension in Platform-Based Design (PBD): the will to reuse as

many core components as possible between product variants and the wish for the widest panel of variants which requires singular product characteristics [163]. PBD therefore comprehends all the challenges of system design while adding the complexity of platform-specific tradeoffs such as maximizing commonality with minimizing performance loss, minimizing costs, maximizing variety etc. [164]. The variants of a platform family should also be thoughtfully designed not to compete with one another on a given market segment. Worse, the standardization of the core elements of a platform may actually "limit the innovation and creativity by locking the platform/system into specific suppliers and technologies" [170]. Finally change of a platform component may propagate to several product variants, resulting in extended costs [112]. PBD can thus be a powerful strategy to address new market segments rapidly and capitalize on existing developments, but need to be carefully thought through if system evolvability is needed in perspective of an uncertain context or requirements.

A comprehensive review of PBD frameworks and methodologies is given in [168]. A narrower set of research works addresses PBD *under uncertainty*, hence better matching the contextual conditions (i.e. environment and requirements) we expect for the design of LOC-based POC IVMD systems. Suh et al. summarize what they identified as the principal contributions on the topic in [163]. In the same document, the authors present what they claim is a more comprehensive methodology for designing platforms under uncertainty. Their strategy includes the systematic mapping of uncertainties to (functional) product attributes, design variables, physical components, flexible designs, and finally to relevant costs for economic evaluation. Suh et al.'s flexible PBD design methodology incorporated a significant number of the concepts and methods later discussed by Cardin et al. [143]. Madni et al. address the methods and tools required in order to trade platform standardization against flexibility in [170]. The authors elaborate on what they refer to as *adaptable platform-based engineering*. Adaptability here is not to be confounded with adaptability as defined by the SEARi group (table 3.1) but is conceptually closer to flexibility or changeability. Madni et al. put forward the benefits of model-based design for developing evolvable platform, mainly on the account of the possibilities offered for sensitivity and cost-effectiveness analyses, uncertainty propagation analyses etc.

As Simpson et al. mention, PBD is sometimes viewed at a higher level of abstraction, depending on the technologies the platform is meant to be built on. In our case, the most appropriate view of PBD is probably through the CPS lens: Sangiovanni-Vincentelli famously conceptualized the foundation of PBD for embedded (arguably equivalent to today's cyber-physical) systems [171, 172]. The author defined an embedded platform as "an abstraction that covers several possible lower-level refinements. Every platform gives a perspective from which to map higher abstraction layers into the platform and one from which to define the class of lower-level abstractions that the platform implies". The PBD methodology for CPS involves the specification of the platform *function space* (figure 3.2.1) which should comprehend the set of elementary functions that the system will require in order to fulfill various use-cases. A given *function* (i.e. function instance) is meant to be allocated to a specific *architecture* falling within the *architectural space* supported by the platform. This allocation or *mapping* process is a key feature of CPS platforms and is often performed in search of meeting pre-defined optimality criteria (e.g. mapping a function on the architecture that will perform the fastest, etc.). The utmost successful example of a computational platform is the personal computer: computer programs (software) are nothing but functions (in the most generic meaning), allocated onto an architecture (e.g. x86, Linux, etc.). The operation of these functions is enabled by their mapping onto a target processor architecture, the mapping process of which is handled by the compiler.

CPS platform evolution can be conceptualized by the need to *expand* of the function space of a platform initial design, which may translate in the need for expanding the

Figure 3.2.1: CPS PBD in an uncertain environment. Changing context and requirements may necessitate the expansion of both the function and architectural space. Design principles for favouring platform evolvability are meant to facilitate that expansion at minimum costs and efforts.

architectural space as well (expansion arrows in figure 3.2.1). Change-related *means -ilities* associated with evolvability such as *scalability* or *targeted modularity* need to support the PBD design methodology in anticipation of unpredicted changes in order to help the smooth expansion of both function and architectural space at minimal costs and efforts. These *means -ilities* are nothing but the *adaptation mechanisms* that Madni advocates should be included in the initial platform design, with the help of *architectural patterns*.

3.3 Summary

The field of POC IVMD is characterized by a *dynamic marketplace*, *fast-pace technological innovation*, and a *variety of environments*. These properties contribute to the vast socio-economico-technical complexity that challenges the design of next-generation POC IVMD systems. MBSE frameworks and methodologies can be adopted in an attempt to handle that complexity. MBD, in particular, can assist the development of extensive CPS, by partly enabling the modelling, testing, and validation of engineering systems comprehending elements ranging from biological entities to cloud-computing and user-interactions.

Existing MBSE frameworks yet rarely accommodate for needed-change and do not intrinsically promote system evolvability: an overarching change-*ility* which should help to cope with the aforementioned challenges by facilitating incremental system evolutions in a cost-effective and timely manner. PBD can represent a legitimate entry-point for the design of *evolvable* CPS on the condition that it promotes the incorporation of change-enabling and change-absorbing mechanisms at key locations within the platform architecture, favoring the smooth expansion of its *function* and *architectural space*.

In the ensuing chapters, we present our efforts to carry out PBD for the development of a *evolvable* smartphone-based biosensing platform relying on the Silicon-Nanowire biological Field Effect Transistor (SiNW-bioFET) technology. Once again we disclose that although we acknowledge the crucial character of IVMD context dynamism and uncertainty,

and the importance cost-effectiveness assessment in platform-based design for evolvability, our efforts mainly addressed the conceptual design phase and in particular systems architecting and incorporation of *change-enabling/change-absorbing* mechanisms into the initial design. We thus attempted to answer two of Simpson's and Madni's key questions in [164, 170]: "Where to design platform flexibility?" and "How to prepare for new applications, new product lines, and new radical technologies?".

Chapter 4

Silicon-nanowire biological field-effect transistors for ultrasensitive Point-of-Care systems

We mentioned in chapter 1 that advances in micro- and nanotechnologies have been instrumental in the emergence of LOC devices, partly by allowing highly sensitive biosensing from submicroliters biological sample volumes. We also highlighted the integration challenges that engineers must face when developing complex instrumented LOCs systems be it for actuation or sensing. We proceed in this chapter to the in-depth investigation of one of the biosensor technologies that has generated the greatest hopes in providing LOC designers with ultra-high sensitivity detections while limiting instrumentation complexity: the Silicon Nanowire biological Field Effect Transistor (SiNW-bioFET). First, we introduce the parameters and relations dictating the behavior and specifying the sensitivity of SiNW-bioFETs. We then present lock-in amplification a suiting instrumentation technique for the recovery of the low- signals that may flow through high-sensitivity SiNW-bioFETs. We especially investigate in details how the intrinsic noise sources along the acquisition chain and the instrument specifications influence the limit of detection of the sensor for a given analyte.

This chapter illustrates where design for evolvability may find its root: here in the identification of a novel promising technology, and where to go from there.

4.1 SiNW-bioFET sensing technology

Metal-Oxide Semiconductor Field Effect Transistor (MOS-FET) technology is the pillar of the integrated analog and digital electronics industry. Although MOS-FET operation typically involves fast transients and binary outputs, it largely inspired the concept and design of SiNW-bioFET biosensors. SiNW-bioFETs present significant architectural and behavioral similarities with conventional MOS-FETs. The understanding of the basic structure and physical laws governing the behavior of a conventional MOS-FET is therefore essential in order to grasp the subtleties of SiNW-bioFET sensing. The architecture and the main principles of MOS-FET operation are presented in the ensuing sections.

4.1.1 MOS-FET architecture

The architecture of a n-type MOS-FET is shown in figure 4.1.1. It comprises a p-type semiconductor substrate within which two n-doped regions are implanted, and separated

by a distance L. The region separating these two n-type "wells" is the *channel* of the device. Both n-type regions are covered by metallic electrodes forming ohmic contacts. These electrodes constitute the *drain* and the *source* of the MOS-FET. Between source and drain, a *gate* electrode lays on top of the *channel*, electrically insulated from it by a thin dielectric (here SiO_2).

Figure 4.1.1: n-type MOS-FET architecture

The *gate* is often considered the key constituent of the MOS-FET. It is used to modulate the working *regime* of the transistor: an electrical potential applied at the gate will generate an electric *field effect* that will in turn affect the carriers implanted in the semiconductor below the gate insulator. Although a comprehensive discussion of all MOS-FET working regimes is out of the scope of this thesis, three of them are relevant for sensing applications using the SiNW-bioFET technology. Their principle is first introduced below for the conventional MOS-FET.

4.1.2 MOS-FET behavior

4.1.2.1 MOS-FET depletion regime

We consider a p-type substrate MOS-FET with n-doped regions below the source and drain ohmic contact electrodes. We assume that a certain drain-source potential difference $V_{DS} > 0V$ is applied between the source and drain. For a certain range of gate voltages $V_{fb} < V_{GS} < V_{th}$, the MOS-FET is considered to be turned off. V_{fb} is the flat-band voltage and corresponds to the gate electrode potential for which the surface potential ψ_S at the gate-insulator/silicon interface is null. V_{th} refers to the threshold voltage of the transistor, and will be elaborated on further in the next section.
In the *depletion* regime, the MOS-FET presents the dual n-p and p-n junction of its basal state (i.e. when $V_{GS} = V_{fb}$) preventing current flow between drain and source. A rising surface potential ψ_S at the gate-insulator/silicon interface will induce an energy band-bending in the underlying silicon that will in turn result in a depletion of the channel from its majority carriers (holes). By tuning up V_{GS} the depletion depth will increase. The relation between the depletion depth x_D and surface potential ψ_S at the gate in the *depletion* regime is given by:

$$x_D = \sqrt{\frac{2.\varepsilon_{Si}.\psi_S}{e.N_A}}, \tag{4.1.1}$$

where e is the elementary charge, N_A refers to the concentration of acceptor ions within the channel, and ε_{Si} is the permitivity of silicon. This relation is fundamental for the design of MOS-FETs an remains accurate until the transistor reaches the subthreshold regime

4.1.2.2 MOS-FET subthreshold regime

By further increasing V_{GS}, the MOS-FET, is intended to reach full-depletion. The surface potential ψ_S is increased until it equals the Fermi potential of the bulk of the p-doped substrate. When ψ_S reaches this so-called *threshold potential* ψ_{th}, the depletion depth is maximum: the channel is entirely deprived of its majority hole-carriers. The inverted gate-source p-n junction now allows charges to start flowing between the drain and source regions to form a conducting electron-channel. The transistor is said to be weakly-inverted. The electron transport in this early conducting regime is then mainly diffusive and exponentially dependent on the gate voltage V_{GS}. An expression of the subthreshold current $I_{DS_{\text{sub}}}$ is then given by:

$$I_{DS_{\text{sub}}} \simeq I_{DS_0}.e^{\left(\frac{q.\eta}{k_B.T}.(V_{GS}-V_{\text{TH}})\right)} \tag{4.1.2}$$

I_{DS_0} (drain-source current) is constant for a given temperature. The parameter η characterizes the relation between the applied gate-voltage V_{GS} and the surface potential ψ_S built-up at below the gate electrode, at the surface of the semiconductor channel. η is given by the expression:

$$\eta = \frac{\partial \psi_S}{\partial V_{GS}} = \frac{1}{1 + \frac{C'_{\text{D+it}}}{C'_{\text{ox}}}} \tag{4.1.3}$$

C'_{ox} is the gate insulating oxide capacitance per unit area and $C'_{D+it}$is the sum of the capacitances of depletion region and trapped charges at the Si/gate oxide interface. For a high gate quality, the amount of trapped charges at the gate-insulator/silicon interface is negligible. This in turns, mean that C'_{D+it} will be significantly smaller than C'_{ox}. η will then approach unity and the rate of subthreshold current $I_{DS_{sub}}$ change for a small variation in gate voltage V_{GS} will tend to a maximum. This transient behavior for which the transistor passes from an off to an on state is characterised by the subthreshold swing *subthreshold swing* of the transistor and is expressed as:

$$S_{\text{swing}} = \frac{\partial V_{GS}}{\partial \log I_{DS}} \approx \frac{k_B.T}{e}.\ln 10.\left(1 + \frac{C'_{\text{D+it}}}{C'_{\text{ox}}}\right) \approx 2.3.\frac{k_B.T}{\eta.e} \tag{4.1.4}$$

S_{swing} has a theoretical maximum value for silicon of 59.2 mV/decade at 25°C when $\eta = 1$. It translates the ability of the transistor to turn on and off as fast as possible, metrics of the utmost importance in logic applications.

4.1.2.3 MOS-FET linear regime

As the MOS-FET transitioned from the depletion to the sub-threshold regime, a conducting electron-channel has started to form in the semiconductor substrate right below the gate electrode surface. The threshold voltage V_{th}, corresponds to the gate electrode voltage V_{GS} for which the concentration of minority carriers within the inversion channel equals that of the majority carriers (holes concentration) deeper in the silicon substrate. For $V_{GS} > V_{th}$ the MOSFET is considered turned on: the I_{DS} current can increasingly flow between the source and drain, as a linear function of the gate voltage V_{GS}.

The so-called inversion layer will build-up at the surface of the substrate to neutralize the increasing positive charge at the gate electrode: the transistor is said to be inverted. This behavior justifies the interchangeable terminology for this regime either refered to as *linear* or *inversion* regime. If $V_{DS} \ll V_{\text{sat}}$, the drain-source current I_{DS} flowing through the channel will vary proportionally to V_{DS} (figure 4.1.2). The saturation voltage V_{sat} arises from so called short-channel effects and is illustrated in figure 4.1.3. Below V_{sat}, the

Figure 4.1.2: n-type MOS-FET linear regime: an inversion (-n type) layer forms the conduction channel between the source and drain of the transistor.

Figure 4.1.3: n-type MOS-FET saturation region: the drain-source potential V_{DS} is high enough to affect the distribution of the electric field induced by the gate. This translates in a pinching-off of the the conduction channel at the drain and a saturation of the I_{DS} current for increasing values of V_{DS}.

device works in the *linear region* and one can then use Ohm's law and Drude's model of conductivity to calculate the resistance R_{ch} formed along the inversion layer:

$$R_{\text{ch}} = \frac{1}{\sigma}.\frac{L}{W.x_{\text{ch}}}, \text{ with : } \sigma = n.\mu_n.e \tag{4.1.5}$$

where W refers to the channel width, x_{ch} stands for the channel height, L for the channel length, and σ refers to the conductivity of the carriers within the channel. The expression of conductivity is in turn determined by the carrier concentration within the channel, denoted n, on the mobility of these electron carriers μ_n. e stands for the elementary charge. In this linear region, the variations of the DC drain-source current I_{DS} are proportional to both the drain-source voltage V_{DS} and the applied gate voltage V_{GS}.

4.1.3 The Silicon Nanowire biological Field Effect Transistor (SiNW-bioFET)

SiNW-bioFETs for chemical or biosensing applications were introduced in the early 2000's. They were largely inspired from the physical principles and design of conventional MOS-FETs [173]. They have shown promise for the real-time, low-cost, label-free, high throughput analysis of a variety of chemical and biological markers present at low-concentration in low sample processing volumes [174]. SiNW-bioFETs often differ structurally from their MOS-FET counterparts in that they usually do not present the typical complementary dual n-p and p-n junction between their source and drain. They are in this case essentially "gated-resistors" doped with a unique type of acceptor or donor ions (either n- or p-type) from source to drain. These SiNW-bioFETs are sometimes referred to as "junctionless"

Figure 4.1.4: Illustration of the conventional architecture of a SiNW-bioFET

Figure 4.1.5: SiNW chip developed at the NanoBio Integrated Systems (NaBiS) group

biological transistors. As such they have benefited from the increasing interest of the solid-state physics and microelectronics communities: junctionless transistors are considered a promising alternative to the common MOS-FET transistor now reaching the limitations pinpointed by Moore's law[1]. Several design guidelines for SiNW-bioFETs thus originate from advances made through junctionless transistor research [175, 176, 177]

SiNW-bioFETs can be used for detecting biological species in solution by leveraging the field effect generated by the electrical charges of targeted molecules located in the vicinity of the SiNW-bioFET channel: the device's channel conductivity is most often modulated by the selective binding of target molecules in solution at the exposed gate insulator surface 4.1.4. The adoption of specific surface chemistry functionalization schemes is essential in that process as it will enable the specific targeting of biological entities as varied as antibodies, viruses, single stranded nucleic acids, etc. Charges borne by surface bound molecules at the gate but also ions in solution will affect the SiNW internal charge carriers' concentration by electrostatic interactions. The specific binding of a positively charged molecule should translate in the *depletion* of an unbiased p-doped SiNW channel from its majority carriers, hence resulting in a decrease in conductivity, while the binding of negatively charged molecules should result in the *accumulation* of holes in the channel and thus increase conductivity. This change in conductivity can be monitored by the use of appropriate electrical I-V measurement techniques.

[1]The JL-transistor (JL-FET) provides an answer to the challenges specific to sub-20nm fabrication processes, such as increased device variability or costly and intricate fast-thermal annealing.

Figure 4.1.6: Illustration of the intended field effect of a negatively charged analyte bound to the gate insulator's surface functionalization layer. If the SiNW is p-doped, then the selective binding of the target analyte will induce an accumulation of acceptor ions below the gate insulator, resulting in an increase in conductivity. This in turn will be measurable via appropriate electrical measurement of the I_{DS} current flowing between the source and drain of the SiNW-bioFET

SiNW-bioFETs have shown promise for a wide variety of applications ranging from the quantification of ions [178], nucleic acids [179, 180], proteins [181] to the detection of single-virus particles [182]. Limits of detection (LOD) in the picomolar range have been steadily reported since the early days of the technology, and applications for which LODs leveled down to femtomolar concentrations have already been demonstrated [183].
The sensitivity of SiNW-bioFETs is commonly defined as the relative change in drain-source current over the corresponding change in surface charge at the gate-insulator/SiNW surface:

$$S = \frac{\frac{I_{DS_1} - I_{DS_0}}{I_{DS_0}}}{\psi_1 - \psi_0} \tag{4.1.6}$$

A more comprehensive definition of sensitivity will yet take into account the efficacy of the biological assay which results in a gate-surface potential. Details on the the phenomena determining how efficacious the biofunctional layer is are elaborated on further on in the next section. Attempts to numerically predict the electrical behavior and assist the design of ultra sensitive SiNW-bioFETs are based on the consideration of the physical laws affecting and linking both the SiNW channel and the solution containing the analyte of interest. Analytical and numerical investigations have allowed the determination of which design variables impact most SiNW-bioFET sensitivity. We review these parameters in the following section.

4.1.3.1 SiNW-bioFETs sensitivity

The consideration of equation 4.1.5 can be used as an entry point to identify the variables influencing the sensitivity of a SiNW-bioFET. Although the expression of the carrier mobility μ_n takes part in the definition of the conductivity σ, it yet does not impact sensitivity [184]. De Vico et al. explain that carrier mobility influences the absolute change in conductance but not the relative change in conductance that defines sensitivity. This property brought a significant number of research group to develop non-crystalline silicon biological field effect transistors (i.e. polysilicon) in an attempt to reduce the average cost per sensing chip [184, 185, 186, 187, 188, 189, 190]. The further consideration of equation 4.1.5 then highlights two important design parameters influencing SiNW-bioFET sensitivity: *dimensionality* and *charge carrier dopant concentration*: smaller cross-sectional areas and lower dopant concentrations will result in variations in gate-surface charges to translate into greater I_{DS} fluctuations.

Nair et al. were among the first to investigate comprehensively their influence on sensitivity together with several other design properties [191]. They demonstrated that decreasing SiNW dopant concentration could only help scaling up SiNW-bioFET sensitivity until a certain level, beyond which inherent statistical fluctuations of the baseline conductance level, caused by a very low discrete number of dopant ions in each SiNW, would result in unacceptable behavioral inhomogeneity between several devices on the same substrate. The influence of the dopant concentration is a consequence of the Thomas-Fermi screening phenomenon, taking place within the SiNW: the charges borne by the dopant ions in the semiconductor screen the electrical field generated by the charges in the vicinity of the gate-insulator surface. This phenomenon can be quantified by a length metric, translating the distance from the silicon surface to the bulk of the material beyond which the electrical field does not influence the charge carriers anymore. For dopant concentrations inferior to $1 \times 10^{19}\,\mathrm{cm}^{-3}$, this metric is better approximated by the Debye-Hückel theory. An expression of the Debye-Hückel screening -or carrier screening- length $\lambda_{D_{\mathrm{SiNW}}}$ for the silicon wire can then be derived as:

$$\lambda_{D_{\mathrm{SiNW}}} = \sqrt{\frac{\epsilon_{Si} K T}{p e^2}}, \tag{4.1.7}$$

where p is the dopant concentration, ϵ_{Si} is the permittivity of silicon, K is the Boltzmann constant, T the temperature in Kelvin and e is the elementary charge.

Figure 4.1.7: Illustration of effect of the presence of ionic species in solution. Ionic charges screen those borne by the analytes bound to the surface functionalization layer. This phenomena is characterized by the Debye-Hückel screening length, here corresponding to the distance within which the SiNW channel undergoes the influence of the analytes electrostatic field at the gate-insulator surface. It is here represented by the height of the white zone on top of the gate.

The SiNW cross-sectional area and its geometry play a significant role in defining device sensitivity, whereas the SiNW length has theoretically no impact [192, 193, 191, 194, 195]. This is the result of gate-insulator surface charges effectively affecting the SiNW carriers distribution throughout a greater section of their conducting pathway. More recent investigations have confirmed these original conjectures. Buitrago et al. even showed that the sensitivity SiNWs with cross-sections down to $10\,\mathrm{nm} \times 10\,\mathrm{nm}$ did not change significantly when increasing dopant concentrations [175]. Elfstöm et al.'s experimental work supports these results, and highlights how a high surface-to-volume ratio (such as for cylindrical versus single-planar SiNWs) and low cross-sectional areas may prevent from the need to consider dopant concentration as a determining factor for sensitivity. Alternative geometries, such as the FIN-FET design (SiNW exhibiting a height/width ratio superior to one) have also been demonstrated to show significant advantage in terms of sensitivity [196].

In [191], Nair et al. considered charge carrier transport through a cylindrical SiNW-bioFET channel by solving the 3D drift-diffusion Poisson's equation. They investigated the theoretical behavior of their device for electrostatic potentials present at the surface of the surrounding all-around gate-insulator. They considered the SiNW dimensions, dopant

concentration as well as the effect of the *ionic-strength of the solution* submerging the SiNW. This latter parameter plays a prominent role in the definition of the Debye-Hückel screening length *within* the electrolytic solution this time (eq. 4.1.8).

$$\lambda_{D_{\text{sol}}} = \sqrt{\frac{\epsilon_0 \epsilon_{sol} KT}{2N_A e^2 I}} \tag{4.1.8}$$

The Debye-Hückel screening length $\lambda_{D_{sol}}$ is a metric translating the relation between the ionic strength of the solution in contact with the gate with the *effective* distance beyond which the charges borne by the molecules of interest in solution are not contributing to the surface potential at the gate-liquid interface: charged analyte molecules beyond $\lambda_{D_{sol}}$ will be screened by the surrounding ions in solution, leaving unaffected the surface potential effectively modulating the SiNW conductance (figure 4.1.7). Nair et al. drew important conclusions on the effect of the SiNW channel *dimensionality* and its relation with the Debye-Hückel screening length on SiNW-bioFET sensitivity: SiNW cross-sectional dimensions in the range or inferior to the Debye-Hückel screening length are recommended in order for the target surface charges, partially screened by other ionic compounds in solution, to have a greater influence on SiNW conductivity [192, 193, 191, 195].

In addition to the tuning of these static design parameters, the sensitivity of SiNW-bioFET is dependent on the dynamic regime in which the transistor is operated. As mentioned earlier, junctionless SiNW-bioFETs function similarly to enhancement MOSFETs: they allow a drain-source current I_{DS} to flow without requiring a biasing gate voltage. SiNW-bioFETs have thus been studied under various regimes, specifically accumulation, depletion and sub-threshold. The latter has been demonstrated to offer the greatest sensitivity [197]. Indeed, for a given dopant concentration profile and given dimensionality, biasing a junctionless SiNW-bioFET to the subthreshold regime is equivalent to depleting it from its majority carriers until the current running through its channel is almost entirely shut off. This can for instance be achieved either via the use of an electrolytic gate that will set the electric potential of the species in solution at the top of the SiNW or via the use of an independent back-gate that will allow the biasing of the semiconducting channel through the capacitance formed by the underlying substrate.

Gao et al. relied on the principle that maximal sensitivity for SiNW-bioFETs could be achieved when the carrier screening length $\lambda_{D_{SiNW}}$ was much larger than the cross-sectional dimensions of the channel. They argued that this condition could be verified by dynamically biasing the transistor to the subthreshold regime, at which point the carrier concentration in the channel is drastically reduced, translating in a much greater screening length. They demonstrated this hypothesis experimentally and showed that even though sensing in the subthreshold regime generates smaller absolute conductance changes, the sensitivity -related to the *relative* change from baseline- could be multiplied by up to a factor of 3 [197]. Their analytical derivation highlights the fact that the conductance varies linearly with the gate surface potential in the linear regime, it show an exponential dependency in the subthreshold regime. Gao et al. demonstrate that higher-sensitivity could help achieve lower LOD, reducing the detection limit for the Prostate Specific Antigen from 0.75 pM in the linear regime to 1.5 fM in the subthreshold regime. Importantly, Lieber's group illustrate the impact of these behaviors on the signal-to-noise ratio: as sensor sensitivity increases, the signal-to-noise ratio of a conductance change follows the same trend. Nevertheless when the channel conductance tends to zero, *extrinsic* noise becomes limiting since the absolute current changes that need to be recovered also tend to infinitesimals. In other words the sensitivity analysis carried out by Gao et al. for the subthreshold regime or the comprehensive numerical analyses later developed by De Vico et al. for devices operated linearly do not account for the systems-level parameters

that would impact both detection sensitivity and LOD in an integrated sensor-instrument system.

Properties of the gate of SiNW-bioFET devices play a substantial role in defining device sensitivity and require careful consideration. The dielectric material of the gate will screen the surfaces charges potentially accumulating at the gate-liquid interface. Reducing the thickness of the insulating layer will therefore allow surface charges to influence the channel carriers more effectively, and thus increase sensitivity [198]. Interface trapped charges between the gate-insulator and the channel would similarly be detrimental to device performances. They are specifically problematic when the gate native silicon oxide is used as a dielectric.

Figure 4.1.8: Illustration of various strategies for optimizing SiNW-bioFET sensitivity at the biofunctional interface layer. From left to right: 1- homogeneous and short insulator linker molecules to maximize the surface coverage of the receptor probes and bring the target molecule closer to the gate insulator surface; 2- receptor probe alignment in order to favour the binding of target ligand to the receptor probles ; 3- use of a reporter molecule in order to increase the charge borne by the target molecule ; 4- Point-charge model of De Vico et al. computational model [184]

As for most biosensors, a thorough design of the biofunctionalization layer is key in enhancing the sensitivity of SiNW-bioFETs. Strategies for the functionalization of the bare-silicon or gate-oxide layers constitute only one of many opportunities for optimisation. Specific bio-conjugation mechanisms between gate-receptor molecules and their target analytes, the spatial alignment of the receptor probes or the use of reporter molecules have proven to offer potential for increasing sensor performances (figure 4.1.8). An extensive review of these research endeavors is available in [199]. De Vico et al. were the first to include some of the most significant properties of the biofunctional layer in a computational model allowing the parametric study of SiNW-bioFET sensitivity [184]. Their comprehensive model took into account design variables such as the gate-insulator permittivity, thickness and a series of user-defined parameters that helped circumvent the intricacy of defining the biological interface layer by defining some key generic properties. They for instance abstracted the linker-receptor-ligand geometries by specifying a point-charge for the target molecule of interest, and the distance between that point-charge and the gate-insulator surface. Another interesting feature is the capability to specify the surface coverage of the gate by a specific receptor molecule and to take into account the dissociation constant of that receptor with its target ligand in order to compute the cumulative surface charge covering the entire gate. The authors finally implemented an interface between their model and several biochemistry databases. This enabled them to investigate the influence of the SiNW geometry, and of the charge distribution and orientation of receptor-ligand complexes on SiNW-bioFET sensitivity, with varying pH and ionic strength.

SiNW-bioFET sensitivity is thus determined by the intricate physical relations between the device and its surrounding medium. The SiNW-bioFET and biological assay designers have relative control over a few design variables to attempt to maximise that sensitivity

(a) (b)

Figure 4.1.9: (a) Functionalization of a hydrogen-terminated silicon surface and immobilization of protein-decorated gold nanoparticle (b) Atomic Force Microscopy image of the resulting silicon surface coverage, letting appear a surface coverage density of about 66 nanoparticles per μm^2. Reproduced from [200]

and lower the LOD for a given biomolecular target: dimensionality, dopant concentration and working regime but also gate-related properties, and biofunctional layer must be thoroughly evaluated if one wishes to leverage the promise of ultra sensitivity offered by the technology. Other considerations such as the design advanced multi-channel and nano-patterned SiNW structures have also been documented with this same objective [201, 202, 203].

The ultra sensitivity of SiNW-bioFETs makes them unquestionably fit for biosensing applications where low detection limits are of prime importance. Other specifications such as specificity, robustness, reproducibility, affordability or integrability yet contribute just as much to determining the fate of a successful sensing technology. The integrability of SiNW-bioFETs has been put forward as another of their key advantages. We discuss this point in the following section.

4.1.3.2 SiNW-bioFETs integrability

SiNW-bioFETs stand as strong candidates for the development of multiplexed, high-throughput, multi-target biological assays: their micro- to nanometer scale and their silicon wafer-based fabrication processes offer the opportunity for a large number of sensors to be patterned on the same small silicon chip. This concept finds its utility if each sensor or group of sensors can be functionalized individually towards a different biological target. Multiplexing is valuable in that it allows the detection of a panel of biological markers in a single analytical step, often translating in decreased time-to-result, lower sample processing volumes and the possibility to leverage multivariable analytical scores.
Zheng et al. pioneered in developing the first multiplexed SiNW-bioFET chip [204]. They sensitively and selectively recovered the concentration of three distinct cancer biomarkers in the femtomolar range in a single step. Zhang et al. then reported the detection of three cardiac biomarkers using a similar approach [205]. A year later they revised their design and provided their measurement setup with an Application Specific Integrated Circuit (ASIC) chip made responsible for the biosensors excitation and readout [206]. Their ASIC and SiNW-bioFETs chip were interfaced via a supporting Printed Circuit Board (PCB) enabling the routing between the biosensors electrical terminals to the corresponding ASIC Input/Outputs (IO). Although several research groups have successfully developed CMOS-compatible SiNW-bioFET LOCs, none of them have, to our knowledge, actually initiated

the integration effort that would lead to a single chip embedding sensing, excitation and readout on the same silicon substrate [207, 208, 209, 188]. This observation undoubtedly reflects the difficulties lying ahead of the development of high-throughput SiNW-bioFET LOCs. Designers from multidisciplinary teams will need to address system-level considerations beyond the design, layout and manufacturing of the sole biosensors. High-throughput systems will require protocols enabling targeted biofunctionalization of individual sensors, many-channel microfluidics, decontamination protocols allowing reuse, and much likely highly-parallel and sensitive excitation/readout interfaces. Similarly to Micro Electrode Array (MEA) technology, high-throughput SiNW-bioFET LOCs will necessitate custom, embedded ASIC technology to provide the massively parallel processing capability required to acquire, filter, analyze and transmit sensing digital data to connected systems.

In the emerging POC diagnostics context, these integrability issues imply other considerations. The contamination of the LOC most often necessitates to dispose of the device after is has been used. The design of single-use high-throughput, hybrid SiNW-bioFET/ASIC testing chip would thus require throwing away not-only the contaminated assay but also the embedded signal acquisition and processing electronics. In an attempt to bring down the average cost per chip, this solution does not appear optimal, especially if the number of target analytes for a single-test remains sufficiently low so that the design and integration of a custom ASIC can be avoided: the integrability of SiNW-bioFET is tightly coupled to their affordability.

This empirical integrability versus affordability tradeoff analysis drove us to discard ASIC technology as an interface of choice for SiNW-bioFETs and rather consider a *modular* instrument/sensor system approach. This choice brings several key issues starting with the difficulty to design a non-integrated instrumentation accurate and precise enough to be able to leverage the ultrasensitivity the SiNW-bioFETs. Furthermore, design principles for system changeability recommend to give the instrument the ability to *scale* together with biosensor performances if those were to be improved.
We saw in section 4.1.3.1 that the sensitivity of SiNW-bioFETs was one of their most interesting attributes. Unfortunately, this property comes at a price. As Gao et al. demonstrated, nanoscale dimensions and subthreshold regime operation are required for optimal biosensing sensitivity [197]. Under these circumstances, the device drain-source current variation amplitudes δI_{DS} drop below the nanoampere. The importance of signal acquisition and processing techniques in such case is crucial. Although many high-impedance/low-current measurement technique are available to harness sensitivities in this range, few offer the capabilities of the lock-in amplifier. We elaborate on this technique in the next section.

4.2 Lock-in synchronous detection

Numerous research groups involved in the design and fabrication of SiNW-bioFETs have relied on lock-in synchronous detection to carry out their investigations [181, 182, 198, 199, 203, 204, 210]. Lock-in synchronous detection or lock-in amplification is a technique used in many engineering and scientific disciplines as a solution to varied AC-signals measurement problems. It offers both the possibility to recover signals buried in high levels of noise (up to several thousands times that of the signal of interest) and to reliably quantify relatively clean signals that may vary in amplitude or frequency over several orders of magnitude [211]. Both these functions can be of use for recovering the low currents flowing through SiNWs, possibly drown in instrumentation and extrinsic noise. A brief appraisal of the principles and main constituents of lock-in amplification are introduced below.

4.2.1 Fundamental principles of lock-in amplification

4.2.1.1 Input signal and synchronous references

Figure 4.2.1: The lock-in amplifier. Reproduced from [212]

The lock-in amplification technique aims at recovering the magnitude and phase of a periodic AC signal with a known main harmonic pulsation ω by converting it to DC. The simplified schematic of a lock-in amplifier is provided in figure 4.2.1. The lock-in amplification process is sometimes referred to as *synchronous detection* because it requires the generation of a *reference signal*, oscillating at the exact same AC pulsation ω and possibly presenting a fixed phase-offset to it. The expression of a reference sinusoidal voltage signal $V_{\text{ref}}(t)$ can thus be denoted as:

$$V_{\text{ref}}(t) = A_{\text{ref}} \sin(\omega t + \phi_{\text{ref}}) \tag{4.2.1}$$

Figure 4.2.2: Lock-in amplification SysML block diagram

A reference signal internally generated by the lock-in amplifier may be used simultaneously both for the physical excitation of an active sensor and for the internal signal processing tasks required by the synchronous detection technique. In this case the reference signal is called an *internal reference*. The use of an internal reference for external sensor excitation most often requires the conditioning of the analog signal derived from the reference (figure 4.2.1). Such conditioning might be needed in order to scale the signal to the appropriate level, to filter it from unwanted AC signal harmonics, or simply to buffer it e.g. to ensure that it can drive a high current or low-impedance load. A reference voltage signal, output from a digital-to-analog converter for instance, will traditionally go through a reconstruction filter followed by a low output impedance buffer amplifier.

Since SiNW-bioFET are usually not polarized devices (they present the same behavior if their source and drains are inverted), the buffered reference signal can be applied to the drain or source terminal of the sensor while the other terminal is connected to a fixed reference potential V_{ref}. Let us consider that the SiNW-bioFET source potential is maintained at a potential $V_{\text{ref}} = 1.8\,\text{V}$. The reference signal is an AC signal of main harmonic ω, with a baseline DC potential of $1.8\,\text{V}$ and amplitude A_{ref}. Applying this reference signal to the biosensor's drain terminal is equivalent to exciting the SiNW-bioFET with a signal of amplitude $|V_{DS}|$, with:

$$|V_{DS}| = |V_D - V_S| = A_{\text{ref}} \tag{4.2.2}$$

We will only consider the case where $|V_{DS}| < |V_{sat}|$, i.e. that we are working in the linear region of the transistor. We can therefore assume proportionality between the applied excitation voltage amplitude V_{DS} and the drain-source current I_{DS}.

As we will see in an ensuing section, a careful design of the signal conditioning block (figure 4.2.2) is of primordial importance to ensure that the lock-in amplifier can accurately recover subnanoampere I_{DS} current variations. Following the signal conditioning block, the Phase-Sensitive Detection block (PSD) comprehends the two main processing steps of lock-in amplification. In a noise-free scenario, the signal at the entry of the PSD block $V_{\text{PSD}_{\text{in}}}(t)$ is a linear function of $I_{DS}(t)$, possibly phase-shifted with $V_{\text{ref}}(t)$, depending on the transfer functions of the SiNW-bioFET and of the amplifiers and filters constituting the signal conditioning block. We can write:

$$V_{\text{PSD}_{\text{in}}}(t) = A_{\text{PSD}_{\text{in}}} \sin\left(\omega t + \phi_{\text{PSD}_{\text{in}}}\right) \tag{4.2.3}$$

4.2.1.2 Phase-Sensitive Detection

Figure 4.2.3: Phase-Sensitive Detection SysML internal block diagram. An AC-coupling block is responsible for removing any DC offset component in the signal before demodulation is performed. Any DC error would result in a spurious harmonic term at $f = f_{\text{exc}}$ after the demodulation block.

The performances of lock-in amplification for the recovery of noisy signals is greatly influenced by the design of its PSD block (figure 4.2.3). This functional block often first comprises an *AC-coupling* block, in charge of removing the DC baseline offset of the AC signal of interest. We will for now ignore the influence of the AC-coupler as it does not affect the fundamental principles of PSD. PSD always involves the parallel dual-*demodulation* of the input signal $V_{\text{PSD}_{\text{in}}}(t)$ with the *in-line* reference signal $V_{\text{ref}}(t)$ and with a second reference signal $V_{\text{ref}_{90}}(t)$ in *quadrature* with the first, i.e. $\phi_{\text{ref}_{90}} = \phi_{\text{ref}} + 90°$. In a noise-free scenario, the output of the *in-line* demodulator can be derived as follows:

$$V_{\text{demod}}(t) = V_{\text{PSD}_{in}}(t).V_{\text{ref}}(t)$$
$$V_{\text{demod}}(t) = A_{\text{PSD}_{\text{in}}} \sin\left(\omega.t + \phi_{\text{PSD}_{\text{in}}}\right) \times A_{\text{ref}} \sin\left(\omega t + \phi_{\text{ref}}\right)$$

Using standard trigonometry, we can rewrite that expression as:

$$V_{\text{demod}}(t) = \frac{A_{\text{PSD}_{\text{in}}} A_{\text{ref}}}{2}\left(\cos\left(\phi_{\text{PSD}_{\text{in}}} - \phi_{\text{ref}}\right) - \cos\left(2\omega t + \phi_{\text{ref}} + \phi_{\text{PSD}_{\text{in}}}\right)\right) \tag{4.2.4}$$

A similar computation for the demodulation with the quadrature reference signal gives:

$$V_{\text{demod}}(t) = V_{\text{PSD}_{\text{in}}}(t).V_{\text{ref}_{90}}(t)$$
$$V_{\text{demod}}(t) = A_{\text{PSD}_{\text{in}}} \sin\left(\omega.t + \phi_{\text{PSD}_{\text{in}}}\right) \times A_{\text{ref}_{90}} \sin\left(\omega t + \phi_{\text{ref}} + 90\right)$$
$$V_{\text{demod}}(t) = \frac{A_{\text{PSD}_{\text{in}}} A_{\text{ref}_{90}}}{2}\left(\sin\left(\phi_{\text{PSD}_{\text{in}}} - \phi_{\text{ref}}\right) + \sin\left(2\omega t + \phi_{\text{ref}} + \phi_{\text{PSD}_{\text{in}}}\right)\right)$$

Both the expressions of the in-line and quadrature demodulators output let appear a DC component and a harmonic term pulsating at twice the excitation frequency. The DC term comprehends information about both the amplitude and phase of the PSD input signal $V_{\text{PSD}_{\text{in}}}$. The key of the PSD process resides in the low-pass filtering of both in-line and quadrature time signals, which gives in an ideal case:

$$\begin{cases} V_{\text{LPF}}(t) = \dfrac{A_{\text{PSD}_{\text{in}}} A_{\text{ref}}}{2} \cos\left(\phi_{\text{PSD}_{\text{in}}} - \phi_{\text{ref}}\right) \\ V_{\text{LPF}_{90}}(t) = \dfrac{A_{\text{PSD}_{\text{in}}} A_{\text{ref}_{90}}}{2} \sin\left(\phi_{\text{PSD}_{\text{in}}} - \phi_{\text{ref}}\right) \end{cases} \tag{4.2.5}$$

Since the reference signal is set by design, we can assign $A_{\text{ref}} = 1$ and $\phi_{\text{ref}} = 0$. We eventually obtain:

$$\begin{cases} V_{\text{LPF}}(t)^2 + V_{\text{LPF}_{90}}(t)^2 = \dfrac{\left(A_{\text{PSD}_{\text{in}}}\right)^2}{4} \\ \dfrac{V_{\text{LPF}_{90}}(t)}{V_{\text{LPF}}(t)} = \tan\left(\phi_{\text{PSD}_{\text{in}}}\right) \end{cases} \tag{4.2.6}$$

or in other terms we can now recover the amplitude and phase of $V_{\text{PSD}_{\text{in}}}(t)$ as:

$$\begin{cases} \left|V_{\text{PSD}_{\text{in}}}\right| = A_{\text{PSD}_{\text{in}}} = 2\sqrt{V_{\text{LPF}}(t)^2 + V_{\text{LPF}_{90}}(t)^2} \\ \angle V_{\text{PSD}_{\text{in}}} = \phi_{\text{PSD}_{\text{in}}} = \arctan\left(\dfrac{V_{\text{LPF}_{90}}(t)}{V_{\text{LPF}}(t)}\right) \end{cases} \tag{4.2.7}$$

The I_{DS} current phase is accordingly given by $\angle I_{DS} = \phi_{\text{PSD}_{\text{in}}}$ and its magnitude $\left|I_{DS}\right|$ is a function of the various gains and attenuations applied throughout the signal conditioning block.

In the non-ideal case, the input signal $V_{\text{PSD}_{\text{in}}}(t)$ will be accompanied by interfering signals such as intrinsic component noise, extrinsic noise, a DC offset, or a low-frequency drift error. The demodulation process also applies to all these signal components. Let

us consider the PSD of a noise-free $V_{\mathrm{PSD_{in}}}(t)$ signal biased with a DC offset. The in-line demodulation of $V_{\mathrm{PSD_{in}}}(t)$ will thus result with:

$$
\begin{aligned}
V_{\mathrm{demod}}(t) &= V_{\mathrm{PSD_{in}}}(t).V_{\mathrm{ref}}(t) \\
V_{\mathrm{demod}}(t) &= \Big(A_{\mathrm{PSD_{in}}} \sin\left(\omega.t + \phi_{\mathrm{PSD_{in}}}\right) + C\Big) \times A_{\mathrm{ref}} \sin\left(\omega t + \phi_{\mathrm{ref}}\right) \\
V_{\mathrm{demod}}(t) &= \frac{A_{\mathrm{PSD_{in}}} A_{\mathrm{ref}}}{2}\Big(\cos\left(\phi_{\mathrm{PSD_{in}}} - \phi_{\mathrm{ref}}\right) - \cos\left(2\omega t + \phi_{\mathrm{ref}} + \phi_{PSD_{\mathrm{in}}}\right)\Big) \\
&\quad + C A_{\mathrm{ref}} \sin\left(\omega t + \phi_{\mathrm{ref}}\right)
\end{aligned}
$$

We can observe from the previous equations that the DC offset component characterizing $V_{\mathrm{PSD_{in}}}(t)$ has shifted to the pulsation ω. This simple scenario illustrates the importance of the low-pass filtering, in this case for ensuring that the wide-band image of the DC offset at ω rad/s is for the majority rejected and contributes as little as possible to the reconstruction of the $|I_{DS}|$ and $\angle I_{DS}$ values. Filter design will thus be determinant on the quality of the PSD output signals.

4.2.1.3 Digital lock-in amplification

Lock-in amplification can be implemented in the analog, digital or a combination of both domains. Most of today's available commercial lock-in amplifiers heavily rely on digital technologies. Digital lock-in amplifiers carry out PSD in the digital domain, leveraging the advantages offered by digital signal processing (DSP), notably in terms of result predictability, and execution speed. Internal signal references are generated digitally, for instance using Look Up Tables (LUTs) in association with digital to analog conversion, or by advanced techniques involving Direct Digital Synthesis (DDS) [213].

The digital approach prevents the need for costly and performance-limited pre-digitization filters and DC-coupled analog components. It requires downstream digital low-pass filtering often based on Finite or Infinite Impulse Response (FIR/IIR) filtering techniques. This characteristic constitutes one of the main advantages offered by the digital lock-in amplifier: sharp roll-off, low cut-off digital filters can be implemented at no cost, can be revised programmatically and can be hosted by powerful digital signal processors or Field Programmable Gate Arrays (FPGA). One must yet recognize the important tradeoffs arising with the design of such filters, and the critical need to balance sharp filter roll-offs against long time-constant and memory-consuming filter banks.

This brief introduction to the fundamental principles of lock-in amplification does not account for one of the major rationale justifying the use of the technique: noise rejection. As we will see in the ensuing sections, this consideration is paramount for assessing and optimizing the sensitivity of SiNW-bioFET/lock-in amplifier systems.

4.3 Model-based sensitivity analysis of lock-in amplified SiNW-bioFET signals

The SiNW-bioFETs design space is large. As we saw in section 4.1.3 several key factors contribute to defining the sensitivity of the biosensor. Although some of them are under the control of the system designer, others such as the charge borne by the target analyte are merely system or context-specific properties that yet need to be accounted for. The design of a performance lock-in amplifier similarly requires to take into consideration both the sensing signal characteristics and those of predictable interfering noise and offsets

sources. These are function of the various components along the signal conditioning acquisition chain. These are in turn dependent on the input and output characteristics of the SiNW-bioFET they are meant to interface with. These tight dependencies, even though addressable analytically, can be better observed and analyzed using adequate numerical models of the complete sensor-instrumentation subsystem.

We derive both an analytical Matlab model and a Matlab Simulink/Simscape model engulfing the design variables and intrinsic system properties discussed so far in sections 4.1.3 and 4.2.1. Parts of these models are presented in appendices F, H and G. These appendices are referenced to in the text when most relevant. In addition, a complete SysML model of the most relevant structures and their behaviors is introduced. SysML parametric diagrams representing the physical laws and relations between the various model elements are given as examples throughout the section. These parametric diagrams call Matlab functions to update system-level specification upon the change of any design variable.

Figure 4.3.1: SiNW-bioFET SysML block definition diagram

4.3.1 Analytical and behavioral SiNW-bioFET/instrumentation models

We investigate the properties and behavior of a SiNW-bioFET model as previously described by De Vico et al. [184] and of the lock-in synchronous recovery of its current signals as previously introduced.

4.3.1.1 SiNW-bioFET model

Figure 4.3.2: SiNW-bioFET SysML parametric diagram

We consider a SiNW-bioFET sensor fabricated according to the protocol described in appendix E. We consider the patterning of a polycrystalline silicon channel of mobility $\mu = 2\,\mathrm{cm}^2/\mathrm{V\,s}$. We set the initial SiNW length to $L = 100\,\mu\mathrm{m}$ and we define its rectangular cross-section with width $W = 400\,\mathrm{nm}$ and thickness $H = 30\,\mathrm{nm}$. The SiNW is homogeneously p-doped with boron acceptor ions originally specified with a concentration $N_\mathrm{A} = 1 \times 10^{24}\,\mathrm{m}^{-3}$. In an unbiased state i.e. $V_{GS} = V_\mathrm{fb}$, we assume that the carrier concentration within the channel remains at its intrinsic value N_A. Ohm's law and Drude's model of conductivity then allow us to specify the SiNW's intrinsic resistance as:

$$R_\mathrm{SiNW} = \frac{1}{\sigma}.\frac{L}{W.H} \tag{4.3.1}$$

$$R_\mathrm{SiNW} = \frac{1}{\mu.e.N_\mathrm{A}}.\frac{L}{W.H}, (e = 1.602 \times 10^{-19}\,\mathrm{C}) \tag{4.3.2}$$

$$R_\mathrm{SiNW} \simeq 2.6 \times 10^8\,\Omega \tag{4.3.3}$$

Following up on De Vico's work, we focus our interest on SiNW-bioFETs functionalized with the ABL tyrosine-kinase receptor, specific to the Adenosine Triphosphate (ATP) molecule, pillar of the cell metabolism. We assimilate the ATP/ABL complex with a single charge $Q_c = -3\,e$, distant from $l = 2.4\,\mathrm{nm}$ to the gate surface. We assume that Q_c is yet entirely borne by the ATP molecule and thus that variations in coverage of the SiNW-bioFET gate surface by ABL-tyrosine kinase receptor molecules do not change the intrinsic resistance of the SiNW.

The main specificity of our model is the rectangular cross-section geometry the SiNW channel compared to De Vico's cylindrical design (figure 4.3.3). This distinction is meant to better reflect the geometry of the SiNW-bioFET developed within the NaBiS group. We assume that since the height-to-width ratio of our SiNW is low, surface charges at the gate are likely to influence charge carriers deep in the bulk of the channel, should the Debye-Hückel lengths for the solution and for the semiconducting channel allow it. In such

a case we expect our rectangular cross-section device of thickness H to exhibit a similar behavior to that of a cylindrical SiNW of radius R if $R \approx H$ (figure 4.3.3).

Parametric studies can be undertaken in order to assess the numerical error caused by our rectangular cross-section approximation. Better still, a new physical model of the SiNW-bioFET could be derived to better match the geometries of our conventional top-down fabricated devices. As our main interest goes to system-level behaviors and properties, we did not pursue these steps.

Figure 4.3.3: Circular versus rectangular SiNW-bioFET channel cross-section.

The *dissociation constant* of the ATP/ABL-tyrosine kinase complex is $K_d = 62.9\,\text{nmol/L}$. K_d determines θ: the fraction of adsorbed protein at the surface of the gate as a function of the concentration of ATP in solution so that:

$$\theta = \frac{[\text{ATP}]}{[\text{ATP}] + K_d} \tag{4.3.4}$$

Equation 4.3.4 translates a behavior well known in the field of enzymology. Only a fraction θ of the receptors bound to the gate are coupled with their conjugate ligand to form a protein complex. The proportion of receptor/ligand complexes among all available receptors is defined by a sigmoïd function of the ligand concentration in solution.

The gate was assumed to be made of $h = 5\,\text{nm}$ of aluminium dioxide ($\varepsilon_{\text{ox}} = 9.34$) and the ionic strength of the surfacing solution was fixed at $1 \times 10^{-12}\,\text{mol}\,\text{m}^{-3}$. The surface coverage density s_d, of the gate by ABL-tyrosine kinase receptors was specified according to results from on-going surface functionalization studies carried out within the NaBiS group (figure 4.1.9). s_d was fixed at at $65 \times 10^{12}\,\text{receptors/m}^2$, which enabled us to approximate the number of receptor molecules N_{rcp} covering the entire surface area of the gate oxide A_{gate} by:

$$N_{\text{rcp}} \approx s_d \times A_{\text{gate}}$$

$$N_{\text{rcp}} \approx 65 \times 10^{12} \times L \times \Big(2(H + h) + W + 2h\Big)$$

We included in our model the possibility to specify the average cross-sectional surface area of the ligand/receptor complex A_{cplx}, in order to derive an approximation of the maximum number of possible receptor/ligand complexes forming at the gate $N_{\text{cplx.max}}$, so that:

$$N_{\text{cplx.max}} = \min\left(N_{\text{rcp}}, \frac{A_{\text{gate}}}{A_{\text{cplx}}}\right) \tag{4.3.5}$$

This feature is important: should the size of the ligand be greatly superior to that of the receptor, then the maximum number of ligand/receptor complexes at the gate may be limited by the space necessary for these complex to form.

From $N_{\text{cplx.max}}$ we can calculate the approximate number of complexes N_{cplx} present at the gate using the adsorbed fraction θ. We obtain:

$$N_{\text{cplx}} = N_{\text{cplx.max}} \times \theta \tag{4.3.6}$$

The overall surface charge density σ_b, corresponding to the average charge per unit of surface area from all ligand/receptor complexes bound to the gate at an average distance l to the gate surface can then be calculated as:

$$\sigma_b = \frac{N_{\text{cplx}} \times Q_c}{A_{\text{gate}}} \tag{4.3.7}$$

$$\sigma_b = \frac{N_{\text{cplx}} \times Q_c}{L \times \Big(2(H+h) + W + 2h + l\Big)} \tag{4.3.8}$$

Q_c represent the single point charge model we were referring to in section 4.1.3.1. Like De Vico we specified $Q_c = -3\,e$, and the ATP/ABL-tyrosine complex distance to the SiNW gate surface as $l = 2.4 \times 10^{-9}\,\text{m}$. The relative change in conductance of SiNW-bioFETs is then specified as [184]:

$$\frac{\Delta G}{G0} = -\frac{2}{Rep_0}\Gamma\Big(\Gamma_l\sigma_b + \sigma_s\Big) \tag{4.3.9}$$

Like De Vico, we will assume σ_s, the surface charge density directly at the gate surface, to be null. R represents the radius of the equivalent circular cross-section SiNW which we approximated with a thin rectangular cross-section channel. R is therefore assimilated to H according to our notation. p_0 correspond to the dopant concentration N_A and e is the elementary charge. The parameter Γ translates the sensitivity of the device according to the formula [184, 214]:

$$\Gamma = \frac{\varepsilon_{\text{Si}} K_0(B) \dfrac{\lambda_{D_{\text{sol}}}}{\lambda_{D_{\text{SiNW}}}} I_1\Big(\dfrac{R}{\lambda_{D_{\text{SiNW}}}}\Big)}{\left[K_0(B)\left(\dfrac{1}{B}\right) + \ln\left(\dfrac{R+\Delta R}{R}\right) K_1(B)\dfrac{\varepsilon_{\text{Sol}}}{\varepsilon_{\text{ox}}}\right]\varepsilon_{\text{Si}}\Big(\dfrac{R}{\lambda_{D_{\text{SiNW}}}}\Big) I_1\Big(\dfrac{R}{\lambda_{D_{\text{SiNW}}}}\Big) + \varepsilon_{\text{Sol}} K_1(B) I_0\Big(\dfrac{R}{\lambda_{D_{\text{SiNW}}}}\Big)} \tag{4.3.10}$$

$$\text{with: } B = \frac{R + \Delta R}{\lambda_{D_{\text{sol}}}} \tag{4.3.11}$$

with I_0, I_1, K_0, K_1 the modified Bessel functions of the first and second kind ; $\varepsilon_{\text{Si}}, \varepsilon_{\text{ox}}, \varepsilon_{\text{Sol}}$ stand for the relative permittivities of the SiNW, the aluminium gate oxide and the solution respectively [184, 214].

Finally, Γ_l is a dimensionless parameter quantifying the effect of σ_b and is defined by:

$$\Gamma_l = 2\frac{R}{R+l}\left(1 + \sqrt{\frac{R}{R+l}}\exp\left(l/\lambda_{D_{\text{Sol}}}\right)\right)^{-1} \tag{4.3.12}$$

As we saw in section 4.1.3.1 and equation 4.1.5, the expression of the intrinsic resistance of an unbiased SiNW-bioFET can be calculated as:

$$R_{\text{SiNW}} = \frac{1}{\sigma}.\frac{L}{W.H}, \text{ with : } \sigma = N_A.\mu_p.e \tag{4.3.13}$$

If the amplitude of the potential applied between source and drain is known then we can calculate the intrinsic current value I_{DS_0}:

$$I_{DS_0} = V_{DS}.\sigma.\frac{W.H}{L}, \tag{4.3.14}$$

From 4.3.9, we can thus deduce:

$$I_{DS} = I_{DS_0}\left(1 - 2\Gamma\left(\frac{\Gamma_l\sigma_b + \sigma_s}{HeN_A}\right)\right) \tag{4.3.15}$$

From equation 4.3.15 we can derive the expression of the impedance Z_{sens} by simple proportionality with V_{DS}. The results of our investigation of $\Delta Z_{\text{sens}} = |Z_{\text{sens}} - Z_0|$ for various concentrations of ATP, Z_0 representing the baseline sensor impedance magnitude for [ATP] = 0, are presented in figure 4.3.4. The SiNW-bioFET design specifications (e.g. N_A, L, W, etc.) were assigned the values discussed throughout this section.

Figure 4.3.4: SiNW-bioFET response to various ATP concentrations: the affinity law describing the ATP/ABL tyrosine kinase complex is characterized by a dissociation constant K_d. For ATP concentrations significantly larger than K_d, the SiNW gate is saturated by ATP molecule: the impedance of the SiNW stabilizes and sensitivity decreases. Sensitivity is increased for SiNW channels only lightly doped

The graph illustrates the dependency of the sensor sensitivity on N_A: the absolute change in impedance increases with lighter doping. This behavior is observed from infinitely low ATP concentrations up until they become significantly higher than the dissociation constant K_d. As the receptors bound to the gate surface become saturated, so is the resistive response of the sensor, setting the upper limit for sensitive quantitation to roughly 2 orders of magnitude above K_d.

We were also able to numerically confirm the independence of SiNW-bioFET sensitivity on the SiNW channel length: although the absolute intrinsic conductance and variations in conductance of SiNWs of various lengths differ from one another, the normalized change over length remains constant (figure 4.3.5). This property highlights SiNW length as an interesting parameter for leading system-level sensitivity optimizations without affecting

the sensitivity of the sensor itself. SiNW length only matters to the sensor designer when it is short enough to necessitate special alignment procedures during fabrication. On the other hand, SiNW length significantly impacts the instrumentation requirements as it is proportionally related to the intrinsic resistance of the wire, and thus to the amplitude of the current signals that will need to be amplified.

Figure 4.3.5: Absolute versus normalized change in impedance as a function of SiNW-bioFET channel length: although the absolute intrinsic conductance and variations in conductance of SiNWs of various lengths differ from one another, the normalized change over length remains constant. This characteristic makes SiNW length a valuable variable for performing system-level optimizations without affecting sensor sensitivity.

With resistance values over $1 - 10\,\mathrm{M}\Omega$, AC current measurements under low-biasing potentials need to be carefully thought through. In particular, one must be wary of the influence of the shunt capacitance C_{shunt} on sensor response. C_{shunt} denotes the equivalent capacitance resulting from the parasitic capacitive components existing between the SiNW-bioFET drain and source terminals. These parasitic capacitances may result from poor sensor design or simply from the non-idealities of the materials and processes used for device fabrication. The influence of C_{shunt} may also be supplemented by capacitive coupling taking place between the various metal electrodes patterned on the sensor's substrate. We will not investigate the elements responsible for C_{shunt} any further in this thesis. C_{shunt} can be modelled by a capacitance placed in parallel to the SiNW-bioFET (figure 4.3.8). For highly resistive SiNW channels, C_{shunt} may corrupt AC current measurements, by bypassing i.e. *shunting* the sensor, if the frequency of the AC excitation signal is high in comparison to the cutoff frequency of the SiNW-bioFET/C_{shunt} electrical network. We elaborate on this behavior below.

The complex impedance of the SiNW-bioFET/C_{shunt} network is given by:

$$Z_{\text{sens}}(\omega) = \frac{R_{\text{SiNW}}}{1 + j\omega C_{\text{shunt}} R_{\text{SiNW}}} \tag{4.3.16}$$

Figure 4.3.9 shows a Bode plot of the normalized SiNW-bioFET/C_{shunt} impedance magnitude (i.e. $20\log(|Z_{\text{sens}}|/R_{\text{SiNW}})$) and phase $\angle Z_{\text{sens}}$ for $C_{\text{shunt}} = 100 \times 10^{-15}\,\text{F}$ and $R_{\text{SiNW}} = 2.6 \times 10^8\,\Omega$. For excitation frequencies beyond the cutoff $f_{\text{sens}} = \omega_{\text{sens}}/2\pi = 1/2\pi R_{\text{SiNW}} C_{\text{shunt}} \simeq 6112\,\text{Hz}$, the impedance magnitude rolls off at 20dB/decade: the pulsating I_{DS} current signal is partly bypassed by C_{shunt} and measurements become inaccurate. AC methods for the recovery of SiNW-bioFET currents/resistance will thus be limited in bandwidth to frequencies significantly lower than f_{sens}.

4.3.1.2 Signal conditioning

Let us consider the AC voltage excitation of a SiNW-bioFET using a sinusoidal waveform generator and the collection of the I_{DS} current at the input of a lock-in amplifier, as depicted in figure 4.3.6. If we consider the SiNW-bioFET design specifications of the previous section, we obtain the intrinsic SiNW resistance $R_{\text{SiNW}} = 2.6 \times 10^8\,\Omega$. Under a low-frequency V_{DS} excitation signal of amplitude 1 V, we should expect to measure baseline currents of amplitude $I_{DS} \approx 1/2.6 \times 10^8 = 0.385\,\text{nA}$. In order to recover the *variations* in currents resulting from changes in the concentration in the target analyte at the gate, we may possibly need an instrumentation presenting current sensitivity several orders of magnitude lower than this baseline value. The design of pico-ampere resolution instrumentation can be a considerable enterprise.

Let us now focus on the signal conditioning block (figure 4.3.7) of the lock-in amplifier represented in figure 4.2.2. We refer to the input stage to the signal conditioning block as the *analog front-end* of the instrument. The SysML model of this analog front end is presented in figure 4.3.7 and its electrical schematics are detailed in figure 4.3.8. It consists of a single-stage voltage feedback transimpedance amplifier (TIA). This TIA serves to collect and amplify the drain-source current I_{DS}, converting it at the TIA output into the voltage signal $V_{\text{TIA}_{\text{out}}}$ [215].

The objective of this TIA stage is to bring the peak-to-peak amplitude of the TIA output voltage signal $V_{\text{TIA}_{\text{out}}}$ as close as possible to the input span acceptable by the following stage of the signal conditioning block.

TIA is a voltage feedback operational amplifier configuration that allows high-gain current amplifications, enabling the recovery of low-currents such as for high-impedance sensing applications. It thus appears quite appropriate for our purpose. In its simplest form i.e. for DC current inputs, the gain of a TIA is set by the value of the amplifier's feedback resistor R_f. An ideal TIA, presenting an infinite input impedance, would force the input current $I_{\text{in}} = I_{DS}$ to run through the feedback resistor R_f, translating at the amplifier's output with a potential:

$$V_{\text{TIA}_{\text{out}}} = -I_{DS} \times R_f \tag{4.3.17}$$

The R_f gain selection is critical: too low and the ensuing signal conditioning stages and PSD may result in poor accuracy; too high and the output of the TIA will saturate, ruining any chance to retrieve sensible values of I_{DS} and thus of the concentration in target analyte.

Unfortunately a number of factors usually prevent the direct application of equation 4.3.17. Among those factors a number of the TIA non-idealities must be accounted for. The amplifier's input bias current I_b and input offset voltage V_{os} for instantce, may translate at the output of the TIA stage by non-negligible *offsets*, impeding the accuracy of the current-to-voltage conversion. We will set aside offset-related issues for now.

Figure 4.3.6: SysML parametric diagram of a SiWN-bioFET interfaced to a lock-in amplifier.

Figure 4.3.7: SysML internal block diagram of the Signal Conditioning block. We consider an analog front end constituted of a single-stage TIA. A passive RC driver circuit precedes the Analog to Digital Converter. The output of the ADC, in turn, can be considered as the entry point to the PSD block.

Figure 4.3.8: Ideal DC transimpedance amplification of the drain-source current I_{DS} flowing through a SiNW-bioFET: the DC current I_{DS} is forced through the feedback resistor R_f, translating at the output of the amplifier by a potential $V_{DS} = -R_f \times I_{DS} + V_{\text{ref}}$

Also, various *noise* sources may corrupt signal integrity along the signal acquisition chain and impair the *precision* of the reconstructed R_{SiNW} value. This may be especially the case when the sensor is not tightly integrated together with the signal conditioning block. Long copper traces connecting the SiNW-bioFET source to the TIA input for instance, are more likely to pick-up environmental noise i.e. *extrinsic noise* by various electromagnetic coupling mechanisms. But even if this interface is carefully designed, which we will assume from now on, signal integrity may be deteriorated just as much by *intrinsic noise* i.e. noise originating from the system components themselves. We will investigate the role of intrinsic noise on an important systems-level specification later on in this chapter.

4.3.1.3 Bandwidth consideration

Equation 4.3.17 restrictively applies to DC and low AC frequency measurements. For wide-band AC applications, the transfer function of the TIA, i.e. the function relating the TIA output potential to its input current, becomes more complex. For instance, the sensor

Figure 4.3.9: Normalized impedance magnitude and phase Bode plot of a SiNW-bioFET and its shunt capacitance. The operating range for the sensor spans from DC to a frequency that must remain significantly smaller than the cutoff frequency f_{sens}, with $f_{\text{sens}} = 1/2\pi R_{\text{SiNW}} C_{\text{shunt}} \simeq 6112\,\text{Hz}$ if $R_{\text{SiNW}} = 2.6 \times 10^8\,\Omega$ and $C_{\text{shunt}} = 100 \times 10^{-15}\,\text{F}$

shunt capacitance C_{shunt} will add in parallel with the common-mode capacitance of the TIA inputs C_{cm}. We denote this equivalent input capacitance as C_{i}. C_{i} is detrimental to the phase-margin of the amplifier: it may cause instability and result in dramatic signal oscillations at the output of the TIA stage. To prevent this phenomenon, a feedback capacitor C_f is added in parallel to R_f to form the complex feedback network of impedance Z_f. C_f contributes to increase the amplifier's phase margin but also adds a zero to the transfer function of the TIA, incidentally creating a single-pole low-pass filter helping to reject high-frequency noise. The TIA's transimpedance transfer function $H_{\text{TIA}}(\omega)$ can be derived as follows:

$$H_{\text{TIA}}(\omega) = \frac{V_{\text{TIA}_{\text{out}}}}{I_{\text{TIA}_{\text{in}}}} = \frac{Z_f}{1 + \dfrac{1}{A_0(\omega)\beta(\omega)}} \tag{4.3.18}$$

, with according to control theory:

$$\begin{cases} A_0(\omega) = \dfrac{1}{1 + j\dfrac{\omega}{\omega_0}} \\ \beta(\omega) = \dfrac{Z_{\text{sens}}}{Z_{\text{sens}} + Z_f} = \dfrac{1 + j\dfrac{\omega}{\omega_f}}{1 + j\dfrac{\omega}{\omega_{i+f}}} \end{cases}$$

$$\begin{cases} \omega_0 \text{ is the amplifier's open-loop gain cut-off frequency} \\ \omega_f = \dfrac{1}{R_f C_f} \\ \omega_{i+f} = \dfrac{1}{R_f(C_i + C_f)} \ , \ C_i = C_{\text{shunt}} + C_{\text{cm}} \end{cases}$$

$$\begin{cases} f_0 = \omega_0 / 2\pi \\ f_f = \omega_f / 2\pi \\ f_{i+f} = \omega_{i+f} / 2\pi \end{cases}$$

The cutoff frequency f_f corresponds to the frequency marking the beginning of the roll-off of the TIA's *signal gain* profile (green curve in figure 4.3.11). In a similar way to what we deducted in the previous section for f_{sens}, f_f should set the ultimate upper excitation frequency limit for the operation of the TIA. The practical maximum bandwidth for AC operation of a high-impedance SiNW-bioFET interfaced to this TIA is thus restricted to the range in which both sensor and amplifier operate in their flat-band region or $f_{\max} = \min(f_f, f_{\text{sens}})$.

Figure 4.3.10: Transimpedance amplification of the drain-source current I_{DS} flowing through a SiNW-bioFET. This electric model takes into account the *input current noise* $i_n = i_{nn} = i_{np}$ of the amplifier and its *input voltage noise* e_n. These intrinsic noise components are characterized by their Power Spectral Density (PSD). Various noise harmonics will be modulated differently, depending on the transfer function $H_{\text{TIA}}(\omega)$ of the TIA.

In our search for a low input-bias and low-voltage noise TIA, we consider the AN8608 quad amplifier [216]. The specifications of the SiNW-bioFET discussed throughout in this

section and those of the AN8608 allow the numerical investigation of the transfer function $H_{\mathrm{TIA}}(\omega)$ (appendix F) and the visualization of its closed-loop magnitude and phase Bode plots as depicted in figure 4.3.11. The latter was obtained by setting R_f to 100 MΩ. The 260 MΩ intrinsic resistance of the SiNW-bioFET channel we are considering requires such amplification: a low-frequency excitation potential of peak amplitude $V_{DS} = 0.3\,\mathrm{V}$ would generate I_{DS} currents merely reaching $I_{DS} = 0.3/2.6 \times 10^8 \simeq 11.54 \times 10^{-9}\,\mathrm{A}$. These would translate at the output of the TIA with peak values of $V_{\mathrm{TIA_{out}}} = -I_{DS_{\mathrm{peak}}} \times 1 \times 10^8 = 1.154\,\mathrm{V}$. This output voltage amplitude is equivalent to a peak-to-peak output AC signal of 2.308 V, a mere 70% of the maximum output span allowed by a rail-to-rail amplifier powered at 3.3 V, leaving enough reserve to avoid signal saturation.

With such a high gain and a small feedback capacitance $C_f = 1.5\,\mathrm{pF}$, the cutoff frequency f_f is limited to $f_f = 1/2\pi R_f C_f = 1.06 \times 10^3\,\mathrm{Hz}$, a pale number in comparison to the theoretical maximum 9 MHz of Gain-Bandwidth Product (GBP) offered by the amplifier. In most cases, parasitic capacitances will add to C_f and reduce the effective bandwidth of the TIA even further.

This example illustrates well one of the limitations of AC measurement techniques for high-impedance applications: the need for high transimpedance gains will limit the AC bandwidth over which measurements can be performed. Low excitation frequency measurements will in turn extend the settling time of lock-in amplified signal, which relies on sharp roll-off, and thus long time-constant, low-pass filtering. This will eventually limit the measurement acquisition dynamics, forcing long time intervals between each measurement point.

4.3.1.4 Signal conditioning noise analysis

Noise is the single most critical factor to account for when designing high-impedance sensing instrumentation. As we mentioned earlier, noise will sum along the signal acquisition pathway, until the signal is digitized. We assumed earlier that extrinsic noise sources were negligible, leaving the need to consider *intrinsic* noise sources and their influence on signal integrity. This section presents in details the steps necessary to carry out a comprehensive intrinsic noise analysis of the signal conditioning block which precedes the PSD stage of the lock-in amplifier. We will, in particular, start by investigating the noise contribution of the TIA stage we have been discussing thus far.

TIA stage noise analysis The noise analysis of the signal conditioning block is probably most complex for the TIA stage. As we saw in section 4.3.1.1, achieving high sensitivity in SiNW-bioFET-based biosensing require small cross-sectional dimensions and either low dopant concentrations or sensor biasing to the sub-threshold regime where the device conductivity tends to zero. High-sensitivity SiNW sensing will thus require high TIA amplification gains, which are inherently associated with significant noise levels. TIA-associated noise sources are three-fold. They consist of the *input voltage noise* e_n of the AN8608 amplifier itself, of its *input current noise* i_n and of the noise intrinsic to the resistive components of the circuit. These TIA input noise sources are additive, and translate at the TIA output by $e_{\mathrm{TIA_{out}}}$, the noise level *referred to the output* (RTO) which will corrupt output signal integrity and propagate to the following stages of the signal conditioning block and eventually deteriorate the precision of the PSD signal recovery.

We start by investigating the contribution of the amplifier's input voltage noise e_n. We denote e_v the TIA's output noise originating from of e_n. We can write:

Figure 4.3.11: Closed-loop signal gain magnitude and noise gain Bode plot. The *signal gain* displays a simple single-pole low-pass filter behavior. Past $f_f = 1/2\pi R_f C_f$ the signal gain rolls-off at -6dB/octave. The *noise gain* instead shows a complex profile as a result of the relations between the amplifier, sensor, shunt capacitance and amplifier feedback network.

$$e_v = e_n \frac{A_0(\omega)}{1 + A_0(\omega)\beta(\omega)} \tag{4.3.19}$$

The expression of e_v is thus dependent on the transfer function of the TIA: an input noise component at a given frequency will be modulated in amplitude as a function of the *noise gain* of the TIA for that specific frequency. For frequencies ranging from DC to the f_{i+f} cutoff, the *noise gain* N_f, represented by the blue curve in figure 4.3.11, is expressed as $N_f = 1 + R_f/R_{\text{SiNW}}$. At f_{i+f}, the noise transfer function reaches its first pole. Voltage noise components beyond f_{i+f} will leak though the input capacitor network C_i and N_f is increasing at $6\,\text{dB}\,\text{octave}^{-1}$. At f_f the noise gain is stabilized by the zero formed by the $R_f C_f$ feedback network of the TIA. Until reaching the second zero in $H_{\text{TIA}}(\omega)$, we have $N_f = 1 + C_i/C_f$. Finally, beyond the cutoff frequency $f_{\text{roll-off}} = \text{GBP}.1/(1 + C_i/C_f)$ the noise gain starts to roll-off at $-6\,\text{dB}\,\text{octave}^{-1}$ [217]. We can thus derive:

$$\begin{cases} \omega_{i+f} = \dfrac{1}{R_f(C_i + C_f)} \simeq 962\,\text{rad}\,\text{s}^{-1} \\ \omega_f = \dfrac{1}{R_f C_f} \simeq 6.67 \times 10^3\,\text{radian/s} \\ \omega_{\text{roll-off}} = \text{GBP}.\dfrac{C_f}{C_f + C_i} \simeq 8.15 \times 10^6\,\text{rad}\,\text{s}^{-1} \end{cases} \qquad \begin{cases} f_{i+f} \simeq 153\,\text{Hz} \\ f_f \simeq 1.06 \times 10^3\,\text{Hz} \\ f_{\text{roll-off}} \simeq 1.3 \times 10^6\,\text{Hz} \end{cases}$$

The *noise gain* profile N_f serves for calculating e_v, together with the Power Spectral Density (PSD) exhibited by e_n.

Figure 4.3.12 shows the PSD provided for the AN8608 amplifier [216]. It reveals the $1/f$ dependency characteristic of many active electronic components at low frequencies: below a certain cutoff frequency f_{nc}, the voltage noise PSD is inversely proportional to the frequency under consideration. The impact of $1/f$ noise is detrimental to a lot of low bandwidth applications.

We can rely both on figure 4.3.12 and on the nominal average PSD at high-frequencies (i.e. $e_{\text{high-freq}} = 8\,\text{nV}/\sqrt{\text{Hz}}$ according to the device manufacturer [216]) to determine the total Root Mean Square (RMS) output noise e_v originating from e_n.

Generally speaking, for a given frequency interval $[f_L - f_H]$, i.e. bandwidth, the expression of the TIA output RMS noise $e_{\text{TIA}_{\text{out}}}$ is obtained from the integration of the amplifier's input noise PSD from f_L to f_H, taking into account the *noise gain* profile over this particular bandwidth [218]. If we consider a noise gain $N_f = 1$, we obtain:

$$e_{\text{TIA}_{\text{out}}} = \int_{f_L}^{f_H} \big(\text{PSD} \times N_f\big).df = \int_{f_L}^{f_H} \text{PSD}.df \tag{4.3.20}$$

If the PSD spectrum is flat over the $f_H - f_L$ bandwidth, (white-noise spectrum) we have:

$$e_{\text{TIA}_{\text{out}}} = \int_{f_L}^{f_H} C.df = C(f_H - f_L) \tag{4.3.21}$$

Conversely if the PSD spectrum consists purely of $1/f$ noise we get:

$$e_{\text{TIA}_{\text{out}}} = \int_{f_L}^{f_H} \frac{K^2}{f}.df = K^2 \ln \frac{f_H}{f_L} \tag{4.3.22}$$

with K^2 a characteristic device constant.

Let us consider a noise PSD constituted of $1/f$ noise for frequencies inferior to f_{nc}, and displaying white noise for frequencies superior to f_{nc}. We obtain $K^2/f_{\text{nc}} = C$ and the expression of $e_{\text{TIA}_{\text{out}}}$ becomes:

$$e_{\text{TIA}_{\text{out}}} = C\left(f_{\text{nc}} \ln \frac{f_H}{f_L} + (f_H - f_L)\right) \tag{4.3.23}$$

At 10 Hz, the AN8608 input voltage noise PSD is $29.8 \times 10^{-9}\,\text{V}/\sqrt{\text{Hz}}$. It can be shown that $K^2 = \big((29.8 \times 10^{-9})^2 - e^2_{\text{high-freq}}\big) \times 10$ [219] and thus that $f_{\text{nc}} = K^2/C^2 \simeq 128\,\text{Hz}$.

We have $f_{\text{nc}} < f_{i+f}$, meaning that the $1/f$ noise components will be amplified by $N_f = 1 + R_f/R_{\text{sens}}$ which simplifies the calculations. We can segment the integral of equation 4.3.21 and derive:

$$\begin{cases} e_{v\left[0.01:f_{i+f}\right]} = 8 \times 10^{-9}\left(1 + \dfrac{R_f}{R_{\text{sens}}}\right)\sqrt{f_{\text{nc}} \ln\left(\dfrac{f_{i+f}}{0.01}\right) + f_{i+f} - 0.01} \\ e_{v\left[f_{i+f}:f_f\right]} = 8 \times 10^{-9}\left(\dfrac{1}{f_{i+f}}\right)\left(1 + \dfrac{R_f}{R_{\text{sens}}}\right)\sqrt{\dfrac{f_f}{3} - \dfrac{f_{i+f}}{3}} \\ e_{v\left[f_f:+\infty\right[} = 8 \times 10^{-9}\left(1 + \dfrac{C_i}{C_f}\right)\sqrt{\left(\dfrac{\pi}{2} f_{\text{roll-off}} - f_f\right)} \end{cases}$$

$$\begin{cases} e_{v\left[0.01:f_{i+f}\right]} = 1.872 \times 10^{-6}\,\text{V} \\ e_{v\left[f_{i+f}:f_f\right]} = 8.795 \times 10^{-10}\,\text{V} \\ e_{v\left[f_f:+\infty\right[} = 1.404 \times 10^{-5}\,\text{V} \end{cases}$$

Figure 4.3.12: Voltage noise spectral density of the AN8608 amplifier. The PSD shows the typical $1/f$ dependency of many active electronic components at low frequency.

We arbitrarily chose a lower bound for the first segment at $f = 0.01\,\text{Hz}$ as the integral converges when f tends towards $0\,\text{Hz}$. The third segment is unbounded at high-frequencies i.e. $f_H = +\infty$, but we know that the noise gain profile limits system bandwidth beyond $f_{\text{roll-off}} = 1.3 \times 10^6\,\text{MHz}$ with a $-6\,\text{dB}\,\text{octave}^{-1}$ single-pole filter roll-off. The contribution of the attenuated noise components beyond that cut-off are incorporated by using by the expression of the Equivalent Noise Bandwidth (ENBW) of a single-pole filter for this segment [218], giving:

$$\text{ENBW} = \frac{\pi}{2}\left(f_{\text{roll-off}} - f_f\right)\ \text{Hz} \tag{4.3.24}$$

The total TIA RMS output noise e_v attributable to e_n is then:

$$e_v = \sqrt{\left(e_{v\left[0.01:f_{i+f}\right]}\right)^2 + \left(e_{v\left[f_{i+f}:f_f\right]}\right)^2 + \left(e_{v\left[f_f:+\infty\right[}\right)^2}$$

$$e_v = \sqrt{\left(1.872 \times 10^{-6}\right)^2 + \left(8.795 \times 10^{-10}\right)^2 + \left(1.404 \times 10^{-5}\right)^2}$$

$$e_v \simeq 1.416 \times 10^{-5}\,\text{V}_{\text{rms}}$$

The second input noise source we must consider for the TIA stage is the amplifier's current noise i_n. Just as the current signal I_{DS}, the current noise i_n will translate at the TIA's output by the RMS potential e_i so that:

$$e_i = \sqrt{\text{ENBW}\left(i_n R_f + i_n R_\text{b} N_f\right)} \tag{4.3.25}$$

Once again the expression of the noise gain N_f plays a role in the specification of the current-noise output. The current noise PSD is considered constant and according to the AN8608 manufacturer we have $i_n = 1 \times 10^{-14}\text{A}/\sqrt{Hz}$. Following a similar reasoning as the one we used for the calculation of e_v we can compute the piecewise contribution of the amplifier's current noise throughout the noise gain profile. We obtain:

$$\begin{cases} e_{i\left[0:f_{i+f}\right]} = 1\times10^{-14}\left(R_f + R_b\left(1+\dfrac{R_f}{R_{\text{sens}}}\right)\right)f_{i+f} \\ e_{i\left[f_{i+f}:f_f\right]} = 1\times10^{-14}\left(R_f + R_b\left(1+\dfrac{R_f}{R_{\text{sens}}}\right)\left(\dfrac{1}{f_{i+f}}\right)\right)\sqrt{\dfrac{f_f}{3}-\dfrac{f_{i+f}}{3}} \\ e_{i\left[f_f:+\infty\right[} = 1\times10^{-14}\left(R_f + R_b\left(1+\dfrac{C_i}{C_f}\right)\right)\sqrt{\dfrac{\pi}{2}(f_{\text{roll-off}}-f_f)} \end{cases}$$

$$\begin{cases} e_{i\left[0:f_{i+f}\right]} = 1.456\times10^{-5}\,\text{V} \\ e_{i\left[f_{i+f}:f_f\right]} = 1.682\times10^{-5}\,\text{V} \\ e_{i\left[f_f:+\infty\right[} = 2.531\times10^{-4}\,\text{V} \end{cases}$$

We can conclude:

$$e_i = \sqrt{\left(e_{i\left[0:f_{i+f}\right]}\right)^2 + \left(e_{i\left[f_{i+f}:f_f\right]}\right)^2 + \left(e_{i\left[f_f:+\infty\right[}\right)^2}$$

$$e_i = \sqrt{\left(1.456\times10^{-5}\right)^2 + \left(1.682\times10^{-5}\right)^2 + \left(2.531\times10^{-4}\right)^2}$$

$$e_i \simeq 2.541\times10^{-4}\,\text{V}_{\text{rms}}$$

Finally, the last of the TIA-associated noise sources to take into account is the so-called Johnson's or thermal noise. The latter is characteristic of resistive components. Thermal noise is conventionally considered uniform over the frequency spectrum. R_f, but also R_{sens} and R_b will contribute differently to the total thermal noise e_r at the output of the TIA. The noise induced by R_f will directly affect $V_{\text{TIA}_{\text{out}}}$ so that:

$$e_{R_f} = \sqrt{4k_BTR_f\text{ENBW}} = \sqrt{4k_BTR_f\frac{\pi}{2}f_f} = 5.236\times10^{-5}\,\text{V}$$

The high-impedance SiNW-bioFET sensor will itself add noise at the input of the amplifier. Several studies have demonstrated that the SiNW may exhibit a $1/f$ noise PSD [220, 221]. Nevertheless as the results supporting these investigations are mainly experimental, we assume that the sensor's thermal noise is purely uniform. The SiNW-bioFET noise undergoes the same amplification as the current signal I_{DS} and thus:

$$e_{R_{\text{sens}}} = \sqrt{4k_BTR_{\text{sens}}\left(\frac{R_f}{R_{\text{sens}}}\right)^2\text{ENBW}} = \sqrt{4k_BTR_{\text{sens}}\left(\frac{R_f}{R_{\text{sens}}}\right)^2\frac{\pi}{2}f_f} = 3.247\times10^{-5}\,\text{V}$$

The thermal noise originating from R_b will be amplified by the noise gain N_f and similarly to what we did for e_i we have:

$$\begin{cases} e_{R_b\left[0:f_{i+f}\right]} = \sqrt{4k_BTR_b\left(1+\dfrac{R_f}{R_{\text{sens}}}\right)^2 f_{i+f}} \\ e_{R_b\left[f_{i+f}:f_f\right]} = \sqrt{4k_BTR_b\left(1+\dfrac{R_f}{R_{\text{sens}}}\right)^2\left(\dfrac{1}{f_{i+f}}\right)^2\left(\dfrac{f_f}{3}-\dfrac{f_{i+f}}{3}\right)} \\ e_{R_b\left[f_f:+\infty\right[} = \sqrt{4k_BTR_b\left(1+\dfrac{C_i}{C_f}\right)^2\left(\dfrac{\pi}{2}(f_{\text{roll-off}}-f_f)\right)} \end{cases}$$

$$\begin{cases} e_{R_b\left[0:f_{i+f}\right]} = 8.176 \times 10^{-7}\,\mathrm{V} \\ e_{R_b\left[f_{i+f}:f_f\right]} = 4.46 \times 10^{9}\,\mathrm{V} \\ e_{R_b\left[f_f:+\infty\right[} = 7.119 \times 10^{-5}\,\mathrm{V} \end{cases}$$

By adding the contributions of R_b for these different segments we obtain $e_{R_b} \simeq 7.119 \times 10^{-5}\,\mathrm{V}$. We can add these values to those of the R_{sens} and R_f thermal noise components and get $e_R = 1.562 \times 10^{-4}\,\mathrm{V}$.

This calculation achieves to provide us with the different contributors to the TIA output RMS noise. Since these noise sources are supposedly uncorrelated, we can add their respective contributions and derive an expression of $e_{\mathrm{TIA_{out}}}$:

$$\begin{aligned} e_{\mathrm{TIA_{out}}} &= e_v + e_i + e_R \\ e_{\mathrm{TIA_{out}}} &= 1.416 \times 10^{-5} + 2.541 \times 10^{-4} + 1.562 \times 10^{-4} \\ e_{\mathrm{TIA_{out}}} &= 4.245 \times 10^{-4}\,\mathrm{V_{rms}} \end{aligned}$$

The noise analysis for the SiNW-bioFET signal conditioning circuitry may encompass other intrinsic noise sources along its path to the digital PSD block. It has been shown that single-stage amplification enhances signal-to-noise ratio compared to multi-stage amplification designs[2] [217]. We follow this design recommendation and limit the other noise sources on the signal conditioning pathway to the RC filter preceding the Analog to Digital Converter (ADC), entry point to our digital PSD (figures 4.3.7 and 4.3.13).

ADC driver circuit noise We demonstrated in this section that the AC recovery of the impedance magnitude and phase of SiNW-bioFETs will be bandwidth-limited either by the transfer function of the sensor or by the feedback network of the TIA. In any case the high-impedance of long-channel SiNW-bioFETs will set relatively soft requirements on signal sampling speed. With the aforementioned design choices, bandwidth is limited by the TIA with a cut-off frequency at $f_f = 1.06 \times 10^3\,\mathrm{Hz}$. Several ADC technologies can be envisaged for achieving sampling frequencies of that order while satisfying the Nyquist-Shannon criteria (i.e. $f_{\mathrm{s}} > 2f_{\mathrm{exc}}$) by one or more orders of magnitude. We consider the Successive Approximation Register ADC (SAR ADC) as a good candidate for our purpose (figure 4.3.13).

Although the investigation of the design characteristics and behaviors of the SAR ADC is out of the scope of this thesis work, we remind the reader of the its basic principle. At sampling time t_0 the S1 switch is closed which starts the charge of the sample and hold (SH) capacitor C_{SH}. Just prior to t_0, C_{SH} may retain a potential with a value laying in between the lower and upper rail power supply potentials. As S1 closes, the charge carried by C_{SH} will corrupt the incoming voltage signal which may result in an error of conversion. The external driving capacitor C_S is needed in order to prevent such spurious potential bursts or "kicks" and maintain the signal potential throughout the sampling acquisition

[2]This property comes from the fact that for an increase in gain in a single-stage design, when the signal linearly (i.e. proportionally) scales with the gain, thermal noise, and current-noise amplified by the noise gain only scale as a function of the square-root of the gain. A high-gain, single-stage design becomes preferable over a dual-stage design achieving the same signal amplification: as the noise at the output of the first stage is amplified proportionally, together with the signal, through stage 2, resulting in a deteriorated SNR

Figure 4.3.13: SAR ADC and RC driver circuit. The $R_S C_S$ network acts both as an anti-alias filter, cutting-off high-frequency noise to input the ADC and as a driving element preventing spurious potential bursts to deteriorate the AD conversion when the switch $S1$ is closed.

and conversion time. R_S should help ensure the stability of the TIA or buffer amplifier preceding the ADC stage. R_S and C_S form a single pole anti-aliasing filter, that will attenuate high-frequency noise beyond $f_{\text{anti-alias}} = 1/2\pi R_S C_S$ Hz..

An adverse effect of this driver circuit is that R_S will add thermal noise e_{R_S} to the $V_{\text{TIA}_{\text{out}}}$ signal so that:

$$e_{R_S} = \sqrt{4k_B T R_S \text{ENBW}} = \sqrt{4k_B T R_S \frac{\pi}{2} f_{\text{anti-alias}}}$$

The R_S and C_S values should be specified in order to allow sampling speeds that will satisfy the Nyquist-Shannon criteria i.e. $f_{\text{anti-alias}} > 2f_{s\ \max}$. We set $R_S = 1 \times 10^3\,\Omega$ and $C_S = 1 \times 10^{-9}\,\text{F}$, which defines the time constant of the RC filter to $1 \times 10^{-6}\,\text{s}$. The resolution of the ADC will set the minimum number of time constants for which the S1 switch must remain closed in between acquisitions, so that the C_S capacitor charge can reach a potential close enough to the input signal that it does not result in conversion errors. The difference between the input signal and the potential at C_S should be smaller than half the resolution of the Least Significant Bit (LSB) of the ADC. For a 12-bit ADC, 9 times constant are sufficient. A theoretical error-free conversion would in our case require $9 \times R_S C_S = 9\,\mu\text{s}$, limiting the signal bandwidth for the anti-alias filter to $1/2 \times 9/2\pi R_S C_S \simeq 55.56\,\text{kHz}$, way beyond the bandwidth allowed by our TIA stage (i.e. $1.06 \times 10^3\,\text{Hz}$), and thus not impacting the overall bandwidth of the signal conditioning block. With these specifications, the noise added by the $R_S C_S$ network is:

$$e_{R_S} = \sqrt{4k_B T R_S \text{ENBW}} = \sqrt{4k_B T 10^3 \left(\frac{\pi}{2}\frac{10^6}{2\pi}\right)} = 1.014 \times 10^{-3}\,\text{V} \tag{4.3.26}$$

ADC noise Finally we must consider the injection of noise by the quantization process itself. Quantization noise is characteristic of the discretization of the continuous input voltage signal into a digital representation of that signal. It depends on ADC resolution and on the voltage span over which conversion occurs. For a single-supply 12-bit ADC referenced at 3.3V, the expression of the quantization noise e_q can be shown to be :

$$e_q = \frac{\text{LSB}}{\sqrt{12}} = \frac{3.3/2^{12}}{\sqrt{12}} = 2.326 \times 10^{-4}\,\text{V} \tag{4.3.27}$$

This noise model relies on the assumption that the quantization noise is uncorrelated with the input signal: its power spectral density then spreads about uniformly over the Nyquist bandwidth $f_s/2$, f_s denoting the sampling frequency [222]. A non-ideal ADC will inevitably add other spurious noise harmonics to the quantization noise. We disregard the impact of ADC non-idealities as these are not always specified by device manufacturers. Their characterization then requires extensive experimental investigations, out of the scope of this formal analysis.

Summing up All previously calculated noise components will sum at the output of the signal conditioning block, materializing by the quantized noisy signal $V_{\text{ADC}_{\text{out}}} = V_{\text{PSD}_{\text{in}}}$. The unfiltered digital input signal at the input of the PSD block therefore includes *at minima* an equivalent RMS noise of amplitude $e_{\text{PSD}_{\text{in}}} = e_{\text{TIA}_{\text{out}}} + e_{R_S} + e_q = 1.671 \times 10^{-3}\,\text{V}_{\text{rms}}$.

Let us assume an ideal sceneario, in which the digital representation of the $|I_{DS}|$ signal amplitude is obtained from the lock-in amplification of a low-frequency AC signal so that $f_{\text{exc}} \ll f_f$. We consider an ideal transimpedance gain $-R_f$ at the TIA stage, the perfect signal buffering at the anti-aliasing filter stage and error-free ADC conversions. We call $S_{\text{PSD}_{\text{in}}}$ the signal at the input of the PSD block, resulting from the conditioning of I_{DS}. $S_{\text{PSD}_{\text{in}}}$ is assumed to be uncorrupted by any other source of noise or error other than $e_{\text{PSD}_{\text{in}}}$. We can thus write:

$$|V_{\text{PSD}_{\text{in}}}| = |S_{\text{PSD}_{\text{in}}}| + |e_{\text{PSD}_{\text{in}}}| \tag{4.3.28}$$

with:

$$\begin{aligned}
\left|S_{\text{PSD}_{\text{in}}}\right| &= \left| - R_f . I_{DS} \right| \\
\left|S_{\text{PSD}_{\text{in}}}\right| &= \left| - R_f . \frac{V_{DS}}{R_{\text{SiNW}}} \right| \\
\left|S_{\text{PSD}_{\text{in}}}\right| &= \left| - 1 \times 10^8 . \frac{0.3}{2.6 \times 10^8} \right| \\
\left|S_{\text{PSD}_{\text{in}}}\right| &\approx 1.154\,846 \times 10^{-1}\,\text{V}
\end{aligned}$$

From this rule of thumb calculation, we can now derive the relative noise level and SNR we may expect from the signal conditioning and digital acquisition process:

$$\text{relative noise level} = \frac{\left|e_{\text{PSD}_{\text{in}}}\right|}{\left|S_{\text{PSD}_{\text{in}}}\right|} \approx 1.4482\% \tag{4.3.29}$$

$$\text{SNR}_{\text{sig.cond.+acq.}} = 20 \times \log\left(\frac{\left|S_{\text{PSD}_{\text{in}}}\right|}{e_{\text{PSD}_{\text{in}}}}\right) \approx 36.78\,\text{dB} \tag{4.3.30}$$

As we discussed in section 4.2.1, the potency of the lock-in amplification process for rejecting noise and increase measurement precision mainly resides in the coupling of a dual-phase demodulation process, bringing the noisy $V_{\text{PSD}_{\text{in}}}$ signal to DC, and isolating signal from noise through a sharp low-pass filtering step. We evaluate the next section importance of the PSD block in reducing the noise level initially present at the PSD input,

raising the SNR level of the lock-in amplifier way beyond the level at the output of the signal conditioning block.

4.3.2 Phase-sensitive detection noise analysis

DC-coupling and demodulation As we discussed in section 4.2.1, PSD relies on the in-line and quadrature demodulations of the digital PSD input signal $V_{\mathrm{PSD_{in}}}(t)$ to bring the main input signal harmonic (at $f = f_{\mathrm{exc}}$) to DC. The magnitude and phase of $V_{\mathrm{PSD_{in}}}(t)$ can then be recovered and used to retrieve the magnitude and phase of the I_{DS} signal.

The precision of the PSD process yet depends on the level of noise in $V_{\mathrm{PSD_{in}}}(t)$. In particular, the demodulation of noise components at frequencies close to f_{exc} will fall in the transmission window of the filter and remain largely unattenuated. This in turn, will translate into spurious low-frequency noise in the output signal.

For systems exhibiting purely white noise, the improvement in SNR offered by PSD is specified by:

$$\text{improvement factor} = \frac{\mathrm{SNR_{out}}}{\mathrm{SNR_{in}}} = \frac{B_{\mathrm{in}}}{B_{\mathrm{out}}}, \tag{4.3.31}$$

where $\mathrm{B_{in}}$ is the equivalent noise bandwidth (ENBW) at the input of the PSD block and $\mathrm{B_{out}}$ is the ENBW at its output [211].

For many other cases though, the spectrum of the signal at the output of the signal conditioning block may contain correlated noise, including as we saw, $1/f$ noise. For applications allowing it, the choice of an excitation frequency beyond the $1/f$ noise cut-off f_{nc} is recommended. For the SiNW-bioFET, we saw that the operable bandwidth was likely to be limited either by the high-gain of the TIA stage or by the shunt capacitance of the sensor itself. AC measurements thus may need to lay within the $1/f$ region. In such case the expression of the improvement factor (eq. 4.3.31) does not apply and a more thorough investigation of the influence of the PSD stage on the output SNR is then needed.

Figure 4.3.14: Magnitude spectrum of a 10 Hz signal in $1/f$ noise

We consider a sinusoidal excitation signal at $f_{\mathrm{exc}} = 10\,\mathrm{Hz}$. At such a low pulsation, TIA amplification of the I_{DS} current introduces $1/f$ noise, rolling off at -3dB/octave.

Figure 4.3.14, depicts such a scenario for which $1/f$ noise has been artificially added to the main signal harmonic. The magnitude of the frequency spectrum at the input of the ADC shows a clear harmonic peak at 10 Hz. Sampling is carried out at 320 Hz, with a single-ended 12-bit SAR ADC powered with a single-supply +3.3 V source.

Figure 4.3.16 now illustrates the magnitude spectrum of the *demodulated signal*. The energy of the input signal initially oscillating at 10 Hz is split, and for half brought back to DC (that is the signal component we will recover) and for half transposed at 20 Hz, twice the excitation frequency. A 10 Hz pulsation remains as a result of the non-ideal AC-coupling of the signal at the demodulator's input: the biasing voltage (V_{ref} in figure 4.3.8) for the amplifiers manifests as a DC offset in the digital $V_{\text{PSD}_{\text{in}}}$ signal, and must be removed[3] (AC-coupler - figure 4.2.3).

Figure 4.3.16 also illustrates the challenge of rejecting the noise components specific to the demodulation process when the excitation frequency is low. $1/f$ noise, the projection of DC offsets to f_{exc} and the projection of half of the energy of the main harmonic signal at $2 \times f_{\text{exc}}$ will all add up and occupy a frequency spectrum still relatively close to DC. Therefore the lower the excitation frequency, the more stringent the requirements on the low-pass filtering stage will be. In the digital domain, careful design methods for FIR/IIR filtering are available. We elaborate on FIR filtering in the ensuing section. We investigate in particular what noise rejection performances are achievable and how these relate to the sensor/instrumentation system LOD for a given target analyte.

4.3.2.1 Low-Pass FIR Filtering

Many of the advantages of opting for digital over analog lock-in amplification come from the benefits offered by digital signal processing and in particular by digital filtering. Unlike their analog counterparts, digital filters are mostly immune to uncertainty on component specifications, drift, or temperature dependencies. The two most significant benefits of digital filters for our application are yet elsewhere. First, digital filters are usually not hardcoded, they can be reprogrammed relatively easily or collectioned in the form of a filter bank. Second, digital filters may achieve far sharper roll-offs than analog filters. The cost for these properties is that of the computational burden, memory requirements and the challenge of trading-off sharp, high-order filters against long-settling time.

Finite Impulse Response (FIR) digital filters are used in a wide variety of DSP applications. Two main properties often justify their choice over Infinite Impulse Response filters: they present a linear phase frequency response, and are intrinsically stable as they do not rely on previous results for the computation of a new output. The price of stability is usually a higher-filter order (i.e. a filter defined by a greater number of coefficients or "taps") to achieve the sharp roll-off of an equivalent IIR filter.

Both FIR and IIR filter design may be undertaken with the aid of various methodologies, depending on the requirements of the DSP application. A review of the theories and frameworks for digital filter design is out of the scope of this work. Instead, we identify the main DSP filtering requirements for digital lock-in amplification and we leverage the Parks-McClellan methodology to implement a filter bank like the one presented in figure 4.3.15.

The lock-in recovery of SINW-bioFET current signals will, depending on the intrinsic conductance and shunt capacitance of the sensor, be limited in bandwidth. At low excita-

[3] If the theoretical value of the ADC biasing voltage is subtracted, an offset error may yet still remain: on top of the theoretical offset bias, an additional offset error may be introduced, in part by the TIA stage. The amplifier's input offset V_{os} and the input bias current I_b are at fault. The effect of the latter may become non-negligible in a high-gain transimpedance configuration. In our case, $R_f = 100\,\text{M}\Omega$, which translates at the amplifier output by $(100 \times 10^{-12})(-100 \times 10^{8}) = 1 \times 10^{-4}$ V.

tion frequencies, the recovery of a single-tone AC signal may be corrupted by a significant amount of $1/f$ noise, which will necessitate low-cutoff, and sharp roll-off filters. High-order filters meeting drastic stop-band attenuation and sharp transition band requirements will comprehend a greater number of taps and therefore exhibit long time constants. These two limitations will work together to extend the measurement time for a single data point and thus limit the maximum output signal *throughput* of the lock-in amplification process: the lower the sensor/amplifier bandwidth, the longer the measurement time, additionally the lower the sensor/amplifier bandwidth, the sharper the filter for a given stop-band attenuation, the longer the measurement time. Thankfully, the recovery of the concentration of a target analyte at the gate of a SiNW-bioFET does not involve the sensor's dynamics. One may thus theoretically use significant amounts of time to carry out the measurement of a low-noise single-point measurement. In practice though, other phenomena such as sensor thermal drift may impede the performance of the measurement procedure if the latter gets prolonged unreasonably.

These concerns source the requirements for the low-pass FIR filter design: in a platform-based instrumentation system, sensors (e.g. SiNW-bioFETs) presenting intrinsic conductances spanning over a given range will tolerate measurements done over different AC bandwidths. In cases where the available AC measurement frequency range is relatively large, low-frequency noise will not be a concern and the filter requirements will soften. Conversely the continuous computation of FIR filtering operations will need to be performed faster, in order to keep pace with a higher AC excitation frequency. If non-adaptive FIR filtering is used, then these multiple scenarios will require different filters in order to satisfy higher-noise, low-bandwidth just as well as lower-noise, high-bandwidth measuring requirements.

To that end, we rely on the Parks-McClellan algorithm to design the six filters presented in figure 4.3.15. The algorithm uses the Remez exchange algorithm and Chebyshev approximation theory to optimize the fit between filter design specifications and those of the filter it iteratively computes [223]. The optimality criteria is defined so that the weighted approximation error between the actual frequency response and the desired frequency response is evenly distributed across both the passband and stopband of the filter, while minimising the maximum error. The FIR filters of figure 4.3.15 were designed with a tolerance of 0.01 dB of ripple in the passband and a large 60 dB of ripple allowed in the stopband, resulting in filters presenting attenuations downs to -90 dB in the stopband. The passband cutoff was specified as $(1 \times 10^{-20}).f_s$, with f_s, the sampling frequency. The stopband cutoffs were tailored in order to provide a filter bank with a number of taps ranging from $\sim$30 to $\sim$800.

Individual filter performances were assessed by evaluating analytically how much noise would remain at the output of the PSD stage in comparison to how much noise was present at the filter input. The technique consists once again in evaluating how the noise power spectral density at the *output* of the demodulator/input of the filter would translate at the output of the filter. The method involves a set of calculations very similar to those undertaken for the evaluation of the RMS noise level at the output of the TIA (section 4.3.1.4). The noise PSD this time, is nothing but the *noise PSD at the output of the ADC shifted to the left in the frequency domain by* f_{exc}. The power density spectrum of a 10 Hz demodulated signal overlapping $1/f$ noise is presented in figure 4.3.16. The spectrum is a mirroring image of the one presented in figure 4.3.14. Aside from this frequency shift, two main features distinguish the two spectrums. First, the demodulated 10 Hz main harmonic has been split and partly shifted to DC and partly to $2 \times 10 = 20$ Hz. Second, the demodulation process introduced a harmonic component at $2 \times f_{\text{exc}}$ that will contribute to impair signal integrity and should therefore be accounted for in the total RMS noise at the filter's output. We will refer to the amplitude of this spurious harmonic remaining

Figure 4.3.15: FIR filter bank frequency response

after filtering as $V_{2.f_{\text{exc}}\text{rms}}$.

Figure 4.3.16: Magnitude spectrum of a digitized 10Hz demodulated signal (in blue) and frequency response of the 228 taps FIR filter (in orange). The filter frequency response illustrates the noise components that will remain mostly unattenuated: those below $f_{\text{FIR}_{\text{cutoff}}}$

The total RMS noise $e_{\text{PSD}_{\text{out}}}$ at the PSD block output will mainly be determined by the very low-frequency noise components. These mostly correspond in figure 4.3.16 to the part of the signal spectrum (blue line) laying below by the pass-band of the FIR filter frequency response (orange line, here representing the 228 taps FIR filter).

A piecewise calculation of the contributions of the main noise sources can then be undertaken in a similar manner to what was used in subsection 4.3.1.4. For this particular case, the 10 Hz excitation frequency lies within the $1/f$ noise region and thus $N_f = 1 + R_f/R_{\text{SiNW}}$. If we assume the ideal cancellation of the signal DC offset before demodulation, the calculation for $e_{\text{PSD}_{\text{out}}}$ becomes:

$$e_{\text{PSD}_{\text{out}}} = e_{v\text{-ENBW}} + e_{i\text{-ENBW}} + e_{R_b\text{-ENBW}} + e_{R_S\text{-ENBW}} + e_{q\text{-ENBW}} + V_{2.f_{\text{exc}}\text{rms}}$$

, with $e_{v\text{-ENBW}}, e_{i\text{-ENBW}}, e_{R\text{-ENBW}}, e_{R_S\text{-ENBW}}, e_{q\text{-ENBW}}$ the contributions *after demodulation and FIR filtering* of the amplifier's voltage noise, current noise, of the thermal noise of R_b and R_f, of R_S, of the quantization noise and of the spurious harmonic remaining at $2.f_{\text{exc}}$ respectively. We can calculate these values:

$$
\begin{cases}
e_{v\text{-ENBW}} = 8 \times 10^{-9}\left(1 + \dfrac{R_f}{R_{\text{sens}}}\right)\sqrt{f_{\text{nc}} \ln\left(\dfrac{f_{\text{exc}} + \text{ENBW}}{f_{\text{exc}}}\right) + \text{ENBW}} \\
e_{i\text{-ENBW}} = 1 \times 10^{-14}\left(R_f + R_b\left(1 + \dfrac{R_f}{R_{\text{sens}}}\right)\right).\text{ENBW} \\
e_{R\text{-ENBW}} = \sqrt{4k_BT\left(R_f + R_b\left(1 + \dfrac{R_f}{R_{\text{sens}}}\right)^2 + R_{\text{sens}}\left(\dfrac{R_f}{R_{\text{sens}}}\right)^2\right).\text{ENBW}} \\
e_{R_S\text{-ENBW}} = \sqrt{4k_BTR_S.\text{ENBW}} \\
e_{q\text{-ENBW}} = \dfrac{(\frac{\text{LSB}}{\sqrt{12}})}{\sqrt{(\frac{f_s}{2})}}.\text{ENBW}
\end{cases}
$$

Note that the distribution for the quantization noise power spectral density is assumed to be uniform over the entire ADC bandwidth $f_s/2$ [222].

The expression of $V_{2.f_{\text{exc}}\text{rms}}$ corresponds to the FIR filtered harmonic at $2.f_{\text{exc}}$. As discussed previously, this harmonic corresponds to half the energy of the main harmonic signal at the input of the ADC, taking into account the possible attenuations/gain throughout the signal conditioning block. We can thus derive :

$$
V_{2.f_{\text{exc}}\text{rms}} = \frac{V_{\text{demod}_{\text{out}}}(2.f_{\text{exc}}).|H_{\text{FIR}}(2.f_{\text{exc}})|}{\sqrt{2}}
$$

$$
V_{2.f_{\text{exc}}\text{rms}} = \frac{V_{\text{exc}}.|H_{\text{TIA}}(f_{\text{exc}})|.|H_{\text{anti-alias}}(f_{\text{exc}})|.|H_{\text{FIR}}(2.f_{\text{exc}})|}{Z_{\text{sens}}2\sqrt{2}}
$$

For this specific case ($f_{\text{exc}} = 10\,\text{Hz}$), the expression of the $e_{\text{PSD}_{\text{out}}}$ gives:

$$
\begin{aligned}
e_{\text{PSD}_{\text{out}}} &= e_{v_{\text{ENBW}}} + e_{i_{\text{ENBW}}} + e_{R_{\text{ENBW}}} + e_{R_S\text{ENBW}} + e_{q_{\text{ENBW}}} + V_{2.f_{\text{exc}}\text{rms}} \\
e_{\text{PSD}_{\text{out}}} &= (2.591 \times 10^{-8}) + (6.337 \times 10^{-7}) + (9.573 \times 10^{-7}) + \\
&\quad (2.571 \times 10^{-9}) + (3.296 \times 10^{-5}) \\
e_{\text{PSD}_{\text{out}}} &= 3.457 \times 10^{-5}\,\text{V}
\end{aligned}
$$

By considering the RMS noise at the input of the PSD, we can compute the improvement factor for our particular example (228 taps FIR). We obtain:

$$
\text{Improvement factor} = \frac{e_{\text{PSD}_{\text{in}}}}{e_{\text{PSD}_{\text{out}}}} = \frac{1.671 \times 10^{-3}}{3.457 \times 10^{-5}} = 48.34 \simeq 33.7\,\text{dB} \tag{4.3.32}
$$

Importantly, $e_{\text{PSD}_{\text{out}}}$ corresponds to the RMS noise amplitude we may expect to recover in our lock-in amplified signal. From $e_{\text{PSD}_{\text{out}}}$, we may recover the theoretical LOD for our SiNW-bioFET/Lock-in amplifier system. The LOD of a detection system is commonly defined as:

$$
\text{LOD} = 3 \times \text{Noise floor}_{\text{RMS}} \tag{4.3.33}
$$

We can leverage the response curve of the SiNW-bioFET such as the one presented in figure 4.3.17. By interpolating the analyte concentration for which the corresponding variation in voltage signal change is $3 \times e_{\text{PSD}_{\text{out}}}$, we may retrieve the LOD value of the SiNW-bioFETs/Lock-in amplifier system. For our particular example we obtain:

$$\text{LOD} \simeq 7.12 \times 10^{-12}\,\text{mol}\,\text{m}^{-3} = 7.12 \times 10^{-15}\,\text{mol}\,\text{L}^{-1} \tag{4.3.34}$$

Figure 4.3.17: SiNW-bioFET response to ATP and Limit of Detection. From infinitesimal ATP concentration, spurious noise still appears at the output of the lock-in amplifier. The noise floor, equivalent to $3 \times e_{\text{PSD}_{\text{out}}}$ is represented by the horizontal purple line. Any signal variation observed at the output of the lock-in amplifier *below* this level should be considered as noise. Variations of amplitude *superior* to this level can be considered significant. We may interpolate the ATP concentration for which the noise floor is reached: this concentration corresponds to the SiNW-bioFET/Lock-in amplifier system LOD.

Obviously our methodology holds as long as the LOD is reached with a constant TIA gain $-R_f$. If the impedance variation upon the binding of the target analyte necessitates a readjustment of the gain (e.g. if the signal saturates at the TIA output), then the noise calculation needs to be reiterated. Another limitation of this approach is that the noise level varies with the ATP concentration: any change in the SiNW resistance will cascade changes in noise levels throughout the acquisition and signal processing chains. We assumed these changes were negligible and carried out our noise calculations by considering the intrinsic SiNW resistance R_{SiNW}

The previous analysis has enabled us to pinpoint the importance of system-level interactions in determining the LOD (and maximum resolution) of our SiNW-bioFET/Lock-in amplifier assembly. The sole intrinsic noise throughout the acquisition chain puts the theoretical high-sensitivity of the biosensor in jeopardy. We saw that the SiNW length could be tweaked without affecting the sensor's performances and that the noise-rejection capabilities of the synchronous detector were determined largely by the AC frequency of the excitation signal. We therefore propose to carry out a parametric study relating these variables together. Our attempt aims both at providing design guidelines to the sensor designer and at recommending an optimal operation point given the performance specifications of the lock-in synchronous detector.

4.3.3 Parametric sensitivity analysis

We rely on the derivations presented in the previous section to calculate the system-level LOD for the SiNW-bioFET and lock-in amplifier specifications we have been considering so far. We carry out the parametric evaluation of the LOD for ATP detection as a function of the SiNW length, of the dopant concentration (i.e. which can be assumed to be the proxy variable for the gate-biasing voltage/working regime of the transistor), of the FIR filter of the PSD stage, and of the excitation frequency f_{exc}. For each computation, the gain of the TIA $-R_f$ was set to the maximum value allowed by the input/output voltage span of the TIA amplifier: in this case the AN8608 is operated between ground and 3.3V. AC signals amplified to a maximum amplitude of $3.3/2 = 1.65\,\text{V}$ were tolerated. The gain specification was decided upon iteratively, starting at $R_f = 100\,\text{G}\Omega$, and decreased by one order of magnitude if saturation was detected. This process was necessary in order to ensure optimal SNR and therefore present the best-case scenario: the theoretical lowest achievable LOD. The Matlab scripts implemented for this study are provided in appendix H

Noise and Limit of Detection As we saw, longer SiNW lengths do not impact the SiNW-bioFET sensitivity but increase system-level noise. Short channels are therefore recommended. After considering the limits beyond which device fabrication and microfluidics integration becomes challenging, we limited our parametric investigations to SiNWs presenting lengths in the 5-20 µm range. Dopant concentrations were swept from $1 \times 10^{24}\,\text{m}^{-3}$ down to $1 \times 10^{21}\,\text{m}^{-3}$. Finally the excitation frequency was swept from $1 \times 10^{-2}\,\text{Hz}$ to three orders of magnitude beyond the TIA bandwidth cutoff, or $f_{\max} = (1 \times 10^3)/2\pi R_f C_f$ Hz. Results for acquisitions leveraging the 34 taps and the 856 taps FIR filters are presented in figure 4.3.18 and 4.3.19 respectively.

Several observations can be made from these results. The first is that, as we predicted, a general trend towards higher sensitivities (e.g lower LOD and smaller system resolution) is associated with lower dopant concentrations. The second is that sharper roll-off filters will generally help rejecting noise better which in turn favors lower LOD values. Beyond these elementary findings, this analysis provides with hints that in general, shorter SiNW lengths are preferable as they are associated with lower noise injection. Nevertheless, system-level considerations help identify discrepancies in this rule: as we may observe for both FIR34 and FIR856 for sensors presenting the 3 higher dopant concentrations ($1 \times 10^{24}, 1 \times 10^{23}$ and $1 \times 10^{22}\,\text{m}^{-3}$) a step of significantly higher LODs characterizes very short SiNWs ($L < 10\,\mu\text{m}$). This step is the consequence of the availability of a discrete, limited number of gains for the TIA stage. In this case, successive gains are one order of magnitudes apart from one another ($R_f = 1 \times 10^9, 1 \times 10^8\,\Omega$...). What these results depict is that for very short channels, a lower gain had to be selected in order to prevent signal saturation. Unfortunately, this lower gain entailed poorer SNR and translated in higher LODs.

Another observation that is worth explicating is the apparent absence of a $1/f$ noise dependency with system LOD. As a matter of fact, the detailed calculations of the RMS noise at the output of the PSD block reveals that the main noise contributor at low frequencies is not $1/f$ noise in our case, but simply the remnants of the demodulated signal harmonic at twice the excitation frequency (i.e. $V_{2.f_{\text{exc}}\text{rms}}$). A remarkable characteristic of the FIR low-pass filtering is that the cutoff frequency (and the entire frequency response of the filter) are normative, and related to the sampling frequency. As we performed synchronous detection at $f_s = 4 \times f_{\text{exc}}$, the ratio between the sampling frequency f_s and this demodulation signal harmonic remained $R = f_s/2f_{\text{exc}} = 2$. The attenuation of the FIR

Figure 4.3.18: Best-case scenario LOD for PSD with the FIR34 filter for various SiNW dopant concentrations. The best-case LOD is achieved for a SiNW dopant concentration of 1×10^{22} dopant/m^3 and for a SiNW length of about 10 µm. For lower dopant concentrations the maximum gain of the TIA is not sufficient to overcome the increasing noise level compare to the low signal amplitude. At 1×10^{22} dopant/m^3, measurements performed on SiNWs shorter than 10 µm will be done with a gain an order of magnitude lower than for longer channels, resulting in a poorer signal amplification compared to noise level relatively over the SiNW-length range.

filter for this harmonic therefore remains constant even when the AC excitation frequency is changed, which explains the relative flatness of the LOD curve at low frequencies.

Finally we must acknowledge the steep deterioration of LOD levels for frequencies beyond about 100 Hz. As we saw in section 4.3.1.4, the noise gain increases sharply after f_{i+f}. Concurrently, as the TIA cutoff approaches, the signal magnitude starts to roll-off. Both effects combined result in soaring decrease in LOD performances, and confirm the limits in operation range we discussed (i.e. $f < \min(f_f, f_{i+f})$).

Offset error We used the same parameters and their respective ranges to compute the minimum tangible offset error on the quantitation of the analyte concentration. This analysis is relevant if the computation of the SiNW impedance magnitude and phase values is based on the DC transfer function of the system. This postulate appears reasonable if we operate the synchronous detection at a pulsation $f_{\text{exc}} \ll f_f$. We will then expect to recover the impedance value of the SiNW-bioFET channel by using the output of the PSD stage so that:

Figure 4.3.19: Best-case scenario LOD for PSD with the FIR856 filter for various SiNW dopant concentrations. The observations made for the same parametric evaluation using FIR34 are valid. The only difference is the higher noise rejection of the FIR856 filter, which still needs to be traded off against longer filter settling times.

$$R_{\text{SiNW}} = V_{DS}.\frac{1}{I_{DS}} = V_{DS}.\frac{1}{\left(\frac{A_{\text{PSD}_{\text{in}}}}{R_f}\right)},$$

where $A_{\text{PSD}_{\text{in}}}$ is the amplitude of the quantized voltage signal at the ADC output, reconstructed by the PSD block. If this signal processing scheme is used (i.e. no particular calibration procedure is undertaken), then an error $\Delta V_{\text{ADC}_{\text{out}}}$ will result from the discrepancy between the assumed relation between I_{DS} and $V_{\text{ADC}_{\text{out}}}$ and how $V_{\text{ADC}_{\text{out}}}(\omega)$ is actually specified (equation 4.3.35). Indeed, the AC excitation potential signal V_{DS} translates successively: first at the output of the SiNW-bioFET sensor through the transfer function $H_{\text{sens}}(\omega)$. In turn, the transconductance of the sensor specifies the current signal at the TIA input, which is then amplified and responds to the transfer function $H_{\text{TIA}}(\omega)$. The amplified voltage signal at the TIA output goes through the anti-aliasing filter presenting a transfer function $H_{\text{anti-alias}}(\omega) = 1/1 + sR_SC_S$. $V_{\text{anti-alias}}(\omega)$ is finally sampled, and corrupted by quantization error e_q. These relations are mathematically equivalent to:

Figure 4.3.20: Best-case scenario uncalibrated error for PSD with the FIR34 filter for various SiNW dopant concentrations. The minimum error is attained at the point where the I_{DS} current increase resulting from the sensor transfer function gain attenuation annihilates with the gain attenuation of the TIA's transfer function. The error is dependent both on SiNW-length since SiNW-length, together with the dopant concentration, condition the gain at which the measurement is carried. These parametric surface plots can be used in order to determine whether a calibration procedure is needed, depending on the LOC one may hope to achieve.

$$\begin{cases} Z_{\text{sens}}(\omega) = R_{\text{SiNW}}.\dfrac{1}{1+j\left(\frac{\omega}{\omega_{\text{shunt}}}\right)} \\ I_{DS}(j\omega) = V_{DS}/Z_{\text{sens}}(\omega) \\ V_{\text{TIA}_{\text{out}}} = -I_{DS}(\omega).|H_{\text{TIA}}(\omega)| \\ V_{\text{anti-alias}}(\omega) = V_{\text{TIA}_{\text{out}}}.|H_{\text{anti-alias}}(\omega)| = V_{\text{ADC}_{\text{in}}}(\omega) \\ |V_{\text{ADC}_{\text{out}}}(\omega)| = |V_{\text{anti-alias}}(\omega)| + |e_q| \end{cases} \tag{4.3.35}$$

, with

$$e_q = \frac{\left(\frac{\text{LSB}}{\sqrt{12}}\right)}{\sqrt{\left(\frac{f_s}{2}\right)}}.\text{ENBW} \tag{4.3.36}$$

The expression of the offset error on the ADC input voltage is thus represented by:

$$\Delta V_{\mathrm{ADC_{out}}} = \left| V_{\mathrm{ADC_{out}}}(\omega) - V_{DS}.\frac{R_{\mathrm{SiNW}}}{R_f} \right|$$

We can then again look at how this offset voltage error would translate in terms of offset error from baseline for the analyte concentration: Since the SiNW-bioFET response to variations in analyte concentration displays a Boltzmann-Sigmoïd curvature, characteristic of receptor-ligand binding and dissociation laws, then the smallest offset error in concentration we may predict corresponds to the equivalent concentration change from a blank measurement (i.e. concentration = 0) for a voltage change equal to $\Delta V_{\mathrm{ADC_{out}}}$. The results for this analysis are given for a PSD executed with FIR34 and FIR856 in figure 4.3.20 and figure 4.3.21 respectively.

Figure 4.3.21: Best-case scenario uncalibrated error for PSD with the FIR856 filter for various SiNW dopant concentrations. The observations made for the uncalibrated error plots using FIR34 also apply here.

Although many observations can be made about these last results, most corroborate with what was observed and explained for the LOD analysis. An interesting specificity of this analysis is yet that it reveals the existence of a local optimum for minimizing the offset error. The sharp edge presenting low error values on figure 4.3.20 and figure 4.3.21 are simply the result of the annihilation of the TIA signal attenuation with the SiNW-

bioFET/C_{shunt} network signal gain. Beyond that point, signal attenuation prevails and the error rises.

4.4 Summary

The ultra-high sensitivity of SiNW-bioFETs should not be taken for granted. Many of the physical laws governing the sensor's behavior and its surrounding medium can be detrimental to sensitivity if overlooked and must therefore be carefully investigated during design. Dimensionality, carrier dopant concentration, gate oxide thickness and material, constitute the main variables in the SiNW-bioFET design space. They yet do not enable to predict accurately the performances of a POC diagnostic system relying on SiNW-bioFETs: As we saw, the theoretical ultra high sensitivity SiNW-bioFET often comes at the price of the need for intricate instrumentation for recovering high-impedance variations under low-voltage biasing conditions and truly exploit the potential of the SiNW-bioFET technology. In a POC setting, modular instrumentation is preferable. Non-integrated systems will alas require the thorough investigation of the instrument's noise-rejection capabilities for determining system-level sensitivity. Lock-in amplification is an interesting method in this regards, mainly for its signal recovery capabilities but also because it may be of use for a variety of impedimetric and current-based biosensing applications. The design of an evolvable system built around the SiNW-bioFET technology therefore requires to take this opportunity into consideration. We present in the next chapter how we leveraged model-based development (MBD) and design principles for promoting system changeability in order to do so.

Chapter 5

Evolvable Smartphone-based Platform for SiNW-bioFET-based Point-of-Care applications

As we discussed in chapters 1 and 2, direct-to-consumer systems for POC in-vitro applications could benefit from design methodologies and principles favoring *evolvability* [106]. This chapter presents the design and implementation of an evolvable platform aiming at *initially* supporting the interfacing of SiNW-bioFETs for various POC IVMD use-cases.

After considering the initial platform functional and performance requirements, we strived to favor the *extension* of the platform *function space* and *architectural space* in anticipation of needed-change, (figure 3.2.1): we followed Madni's methodological recommendations to: 1– identify the commonalities shared by the anticipated platform variants, 2 – to locate *where* in the platform architectures *change absorbers* needed to be implemented, and 3 – to leverage systems *architectural patterns* to incorporate *adaptation enablers*/change-related *means -ilities* into our platform reference architecture.

We specifically looked into the set of *means -ilities* supporting evolvability listed in table 3.2b [107] on top of which we considered some of the design principles suggested by Steiner [224], and Fricke and Schultz [108]. A SysML model of the platform functions, structures and behaviors was developed from the early conceptual design phase to the implementation of the platform prototype. Several diagrammatic representations of this model are presented throughout this chapter.

5.1 Conceptual design of evolvable platforms

The primary objective of our PBD endeavour is *to provide LOC application designers with the possibility to leverage SiNW-bioFET biosensing on LOC variants presenting structural, functional or operational specificities.* These variants may for instance differentiate from one another by the number of SiNW-bioFETs they embed, by the inclusion of sensors or actuators of various other types, or by the instrumentation settings for operating these sensors, etc.

5.1.1 Function space definition and change-enabling mechanisms

One of the key principles of PBD for CPS is the possibility of mapping a set of available *functions* onto an *architecture* instance or *variant* of the platform. The definition of the initial function space is determinant in promoting system-level evolvability. Several change-enabling *-ilities* can be considered at this stage:

Figure 5.0.1: LOC/smart-device CPS PBD in an uncertain environment. Changing context and requirements may necessitate the expansion of both the function and architectural space. Design principles for favoring platform evolvability are meant to facilitate that expansion at minimum costs and efforts. Adapted from [229].

Non-hierarchical integration Smartphone/HWA/LOC platforms display an obvious 3-layer architecture (figure 5.1.6). If we consider LOCs that do not embed any information processing capabilities (i.e. no embedded System-On-Chip (SOC)), two main functional allocation schemes can be envisaged: first, the programmatic functionality for specifying LOC operation can be allocated to the middle layer: the HWA embedded-firmware. In such case the mobile-software layer does not need to incorporate any advanced LOC-operation function but can simply be left responsible for elementary LOC operation start/stop functions and data recovery and analysis. The second main alternative is to implement interfaces for all LoC-related processes at the mobile-software layer, propagating the specification of LOC-variants architectures and functionality mapping at a higher level of abstraction. The HWA middle-layer merely relays and transforms signals or information flowing from LOC to smart-device. By integrating structures or functions at the same hierarchical level, links and communications between structures or functions can be implemented in a direct manner. This architectural pattern describes the *non-hierarchical integration* of a systems structures or functions.

A generic functional allocation scheme for a *non-hierarchically integrated* smartphone-/HWA/LOC architecture is presented in figure 5.1.1. It constitutes one of the key architectural feature of the platform we discuss throughout the rest of this thesis [106].

Composability and modularity Pursuing the design of a *non-hierarchically integrated* platform led us to make the functional distinctions as illustrated in figure 5.1.2. The *generic* functions represented in figure 5.1.1 have been replaced by examples of *specific*

Figure 5.1.1: Functional allocation SysML block definition diagram of a smartphone/HWA/LOC platform where LOC-operation-related functions are available at the mobile-software layer. This topology enables the specification of LOC-variants architectures and functionality mapping at a high level of abstraction. The HWA middle-layer merely relays information or signals form LOC to smartphone.

tasks in figure 5.1.2.

We refer to *Elementary Tasks* (ETs) to describe physical or computational functions performed at the HWA or LOC level. ETs were defined so as to be *composable* functions: they can be composed i.e. associated together while keeping their intrinsic properties, to help realizing LOC or HWA physical or computational functions of higher complexity/utility than each single ET can offer individually. We will refer to these compositions as *Low-Level Tasks* (LLTs) [225, 226] (appendix B). We finally defined *High-Level Tasks* (HLTs): functions that are computational-only, and realized in the higher-abstraction layers of the system: in mobile-software or in the cloud.

One of motivations behind these functional distinctions is that ETs and LLTs are pivotal in defining overall system operation. HLTs arguably form a vast function space, that may extend from data analytics, to machine learning, or cloud computing, whereas ETs and LLTs will be tied to HWA and LOC physical processes, which we expect will considerably constrict the *physical* function space offered by the platform.

The *modularity* and *composability* of ETs, LLTs, and HLTs also enable the specification of LOC *programs* i.e. sequences of tasks required for operating a given LOC variant in an appropriate manner (figure 5.1.3). As we explain in an ensuing section, these programs can be *composed* from the set of functions available in the initial design, or include LLTs realizing functions that were not available in the original function space. The platform

Figure 5.1.2: Platform functionality abstractions. Elementary tasks are defined as physical or computational functions performed at the low-abstraction HWA or LoC level. They can be composed at the mobile-software layer to form low-level tasks. These low-level tasks can be associated to high-level tasks, defined as computational-only, mobile-software level or cloud-computing tasks. This abstraction and ontology enables the non-hierarchical manipulation of low- and high-level task at the mobile-software API level, facilitating the specification of new LoC operations

mapping process then consists in associating each constituent function of a LOC program to the relevant structural component(s) (e.g. a SiNW-bioFET).

Ideality/simplicity Since ETs will much likely represent the limiting element of the platform function space (they are tied to physical processes) they must be carefully defined during initial design. In order to offer the greatest potential for composing LLT functions of enhanced utility, it may be tempting to specify ETs responsible for carrying out very fundamental operations. Let us for instance consider ETs enabling the control of each and every single register of an Analog to Digital Converter (ADC) embedded in the HWA. If ETs are adequately designed i.e if they are composable, then the composition of LLTs from these ETs will present the greatest potential in term of control of the signal acquisition process. Nevertheless the complexity associated with this functional decomposition may be burdening, since it requires handling low-level embedded software behavior at a higher level of abstraction. The system is highly flexible but at what cost? On the other hand, if the only ET associated with ADC control triggers an entire sequence of acquisitions with hard-coded settings, then utility diminishes and it may be difficult to design LLTs offering superior utility from that ET. Once again changeability is threatened.

Fricke and Schultz present the *ideality/simplicity* principle for guiding utility/complexity tradeoffs. *Ideality* aims at reducing system complexity by defining "only useful functions, which may be interpreted as establishing small, simple units/elements with a minimized number of interfaces (loose coupling among and strong cohesion within modules)". By defining *ideal/simple* ETs, we should promote changeability.

Figure 5.1.3: (a) Generic illustrative example. LOC program starts with a high-level task HLT1 carried out by in the cloud followed by a low-level task LLT1 performed by the HWA. A predicate on the execution of HLT2 conditions the next task to be executed: if the program branches to ET2 the program eventually ends by HLT3 otherwise it loops to the beginning. (b) Lock in amplification function performed for the recovery of the impedance magnitude and phase from a high-impedance Silicon Nanowire (SiNW) sensor [106]. Lock-in amplification is carried out by applying a sinusoidal voltage waveform between the source and drain of the sensor, amplifying the resulting current running between source and drain and performing phase sensitive detection. (PSD). PSD itself is composed of a mixing step using the excitation reference signal and of a low-pass filtering step, both function been made available in embedded software LOC program execution diagram for our design instantiation

The concepts discussed in this section offer an abstract framework for the early stage design of evolvable platforms. We detail the mechanisms by which we implemented these design principles in the ensuing sections.

5.1.2 Architecture variants and mapping

The definition of the platform initial function space comes hand-in-hand with the need to adequately abstract the platform architectural space. Specifically, the elementary *structural* components of a generic LOC i.e. the LOC *cells*, and the constituents of the interfacing HWA (figure 5.1.4) should be representable at the mobile-software layer. These abstracted representations (e.g. biosensors, actuators, embedded processor) should display standardized interfaces to enable their associations, composition, inter-communication etc. (e.g. the HWA electrical I/O terminals). These, in turn, should make it possible to the represent the complete physical architecture of any of the HWA/LOC variants *initially* considered, as well as alternative variants, unavailable at initial design time. We discuss *how* we addressed the specification of our platform architectural space in section 5.2.

The availability of structural elements and association or composition rules is fundamental to enable the *platform design-space export* of LOC/HWA variants (figure 5.0.1). As discussed in section 3.2.2, *platform design-space export* is key in PBD for CPS : it conditions the automated mapping of platform functions to the target architecture according to pre-defined optimality criteria [171, 172, 228]. Formal design-space export methods for optimal mapping were not investigated further within the scope of this thesis. We argue that our particular study-case did not justify the effort at this early development stage.

Platform mapping, together with the *non-hierarchical integration* of ETs, LLTs and HLTs thus required the "cyberization of the physical" [227] (figure 5.1.5). We elaborate

Figure 5.1.4: Abstraction of the LOC and HWA structural components at the mobile-software layer. The physical structures of the LOC and the interfaces enabling their associations are "cyberized" at the mobile-software layer [227] in order to enable the definition of LOC variants *architectures*.

on this aspect in the ensuing sections.

5.2 Platform implementation

5.2.1 Mobile software design

5.2.1.1 Object-oriented platform architecting

As we mentioned in section 5.1.2, the functionality of a cyber-physical platform is achieved by granting the application designer the possibility to map the available platform functions to a target architecture. The frameworks within which the platform architectural elements and functions are defined is therefore of critical importance, and so are the mechanisms by which the mapping and architecture-design exports processes are carried out (section 3.2.1).

In a CPS where the application software layer is highly-abstracted, Object-Oriented Programming (OOP) can be leveraged to abstract the software representations of the physical objects involved in system functionality, or as Edward A. Lee formulates, for "cyberizing the physical" [227]. Architecting then consists in abstracting physical objects into software classes and in using data structures intelligently in order to allow the composition or assembly of these objects so as to represent the imbrications of the real physical elements. We undertook that effort for the main structural elements of our POC platform: the LOC, its embedded components and its interfaces (figure 5.2.1).

Figure 5.1.5: Once both the platform function space and platform architectural space are defined, mechanisms need to be implemented in order to allow the functional allocation of ETs to the relevant structural components of the HWA or LOC: the application *mapping*.

The POCLOC class represents the abstraction of the LOC device. It is *compartmentalised* into LOC cells (i.e. POCCell class). These cells are referenced to in an array structure. POCCell is an abstract class that refers to all the embedded components on the LOC that play a role in device operation and that may interface to one another via *terminals* (i.e. POCTerminal class). Terminals may represent physical ports of various nature. Each sensor or actuator embedded in the LOC possesses at least one terminal: the former displays a terminal through which relevant sensing information is retrieved, whereas the latter exhibits a terminal through which actuation can be commanded. Terminals are conceptually inspired from the SysML *full ports*: they are abstraction of interfaces but they are still properties of a constituent: SiNW-bioFETs therefore have three to four terminals: their drain, source, top-gate and back-gate. At least 3 of them are of electrical nature, meaning the flow properties of each terminal represent electrical quantities. Since it offers the features of an interface, the same terminal may be referenced by several objects. A HWA I/O will need to refer to the LOC terminal object it is connected to, and that LOC terminal will point to the same object as the embedded sensor eventually targeted by the application.

5.2.1.2 Software composability mechanisms

OOP probably presents as many advantages for software architecting as it has drawbacks for functional specification. High-level languages such as Objective-C were designed so as to facilitate the implementation of higher abstraction concepts and behaviors than conventional procedural languages such as C. This specificity challenges the definition of

Figure 5.1.6: Platform overall architecture and its obvious 3-layer topology: Smart-device/HWA/LOC. The physical structures and functions of the LOC are "cyberized" at the mobile-software layer [227], e.g. the LOC sensors and actuators and their physical terminals are represented by arrays of POCCell objects in iOS.

Figure 5.2.1: iOS software classes for the main LOC architectural components and their relation to the HWA/LOC interfacing terminals

low-level functions at a high-level abstraction layer. Our endeavour has been meaning to foster a large functional space to the application designer. This involved defining elementary tasks (ETs) that are procedural in essence. As our Application Programming Interface (API) intends to facilitate the specification of LOC operation, we worked on the abstraction of such *procedures* into *objects*. Our objective was to eventually enable the *composition* of procedural task objects to form LOC *programs* (figure 5.2.2a).

As we mentioned in section 5.1, we differentiated *Low-Level Tasks* (LLTs) themselves composed of *Elementary Tasks* (ETs) from *High-Level Tasks* (HLTs). LLTs and ETs refer to the functions allocated either to the HWA or to the LOC. They can be either physical, computational or mixed, whereas HLTs are allocated to the smart-device or cloud computing layers and are computational-only.

As we detailed earlier in this chapter, we attempted to facilitate the expansion of the function space provided by the HWA and LOC by offering the possibility to compose new functionality from the set of ETs. This composition mechanism takes place within the POCLowLevelTask objects: each refers to an array of ETs (figure 5.2.2b). A LLT should ideally offer superior functionality than the sum of the functionalities offered by each of the individual ETs composing it. Evolvability can be enhanced by maximizing the composition possibilities and leverage as much as possible the ETs specified at initial design time *while* limiting complexity: the compositional mechanism by which *tasks* can be assembled and form *programs* is based on object references. POCTask objects may be composed serially by referencing or being referenced to by other POCTask objects (tasks parallelism is at this stage not supported by the platform). This mechanism should also optimally enable

(a) (b)

Figure 5.2.2: POCLOC and POCTask UML class diagrams. POCLoCProgram objects are referencing an array of tasks. These tasks constitute an array of linked objects, each referencing to the previous and following task in the array. The POCTask class is the parent class to the functionally distinct POCLowLevelTask, POCHighLevelTask and POCElementaryTask classes which implement the abstract concepts of LLT, HLT and ET previously discussed.

composition at the meta-level: LLTs composed of LLTs. This cascading mechanism further promotes enhanced functionality. An example of such functional cascade is presented in section 5.2.1.3.

Relevant tasks will precede or follow one another in accordance to the logical and temporal succession of the functions required for the operation the LOC (figure 5.2.2b). At runtime, and in accordance to the classical dataflow model, the execution of a task will be conditioned by the execution of previous tasks. Predicates as simple as the boolean assessment of the end-of-execution of the previous task or substantially more complex (e.g. machine learning algorithm, etc.) can be used to determine which task should be executed next and under what conditions (NB: program execution is handled in the POCApp and not by the iOS Operating System directly). The composition of a LOC program then becomes a *linked list* of tasks. The linking process will require the evaluation and/or inclusion of all predicates in order to either statically or dynamically route program execution to the correct task(s). We elaborate on this task linking algorithm in the next subsection.

Task list linkage LOC program execution may be entirely specified at run-time: the execution of a single task should then foster return values that will dynamically determine which tasks need to be executed next. This scenario could be envisaged for non-deterministic LOC program execution. If on the other hand a priori knowledge of the functions that need to be carried out at the HWA and LOC level is sufficient to specify a fully deterministic program then the tasks can be linked at compilation time instead.

The architecture of our POCTask class and its subclasses was defined so that a given task on the non-terminal ends of the program (i.e. program start and program completion tasks) would refer both to the following and the previous task in the list. The subtasks arrays of LLTs can therefore be considered as a sub-program: the execution of a LLT involves the successive execution of its ETs (figure 5.2.3). The task preceding the LLT will need to point to the first task of the ET subtasks array whereas the last ET of the array will need to point to the task following the LLT. The source code for the POCLowLevelTask class is provided in appendix I.

Figure 5.2.3: Tasks linking scheme for static LOC program specification

5.2.1.3 Specification of SiNW-bioFET acquisitions

The availability of these composability mechanisms and linking method enabled us to fulfil the functional specification for the acquisition of SiNW-bioFET impedance values using lock-in amplification via the specification of a LLT from a reduced set of ETs (figure 5.2.4).

The POCSiNWAcquisition LLT subclass is formed from a cascade of LLTs: it includes a reference to a POCLockInAmplificationAcquisition LLT object which itself includes a reference to a POCDSPCurrentReadoutAcquisition LLT object which includes two ETs. The POCLockInAmplificationAcquisition specifies the settings necessary for carrying out the lock-in synchronous recovery of the I_{DS} currents from the POCSiNWSensor object the POCSiNWSensorAcquisition object refers to. The `excitationTerminal` of the POCWaveformGeneration ET will thus need to point to the `drain` terminal of the POCSiNWSensor object, while the `acquisitionTerminal` of the POCCurrentAcquisition ET will need to point to its `source` terminal: these references constitute the outcomes of the *mapping* process for a single SiNW-bioFET acquisition. On top of the POCLockInAmplificationAcquisition, the SiNW sensor acquisition has the potential to specify a `backgateDCBiasing` POCWaveformGeneration ET object for the biasing of the SiNW-bioFET.

Obviously the composability of a LLT requires to be able to implement all the desired functionality through the set of referenced ETs. The SiNW-bioFET acquisition settings discussed in chapter 3 therefore had to find their equivalent design variables through the POCWaveformGeneration, POCCurrentAcquisition and POCDSP ETs (table 5.1). In this specific case the equivalence is straightforward: the drain-source biasing voltage amplitude corresponds to the amplitude of the sinusoïdal waveform specified by POCWaveformGeneration. Properties like the FIR filter, input gain or number of acquisition points per signal periods are similarly propagated from the parent LLT to the ET properties. One specific concern is the representation of the mechanism specifying the associations between various ETs. For lock-in amplification, the synchronicity of excitation and acquisition must be represented together with the need for carrying out PSD. Specific opcodes may help encoding that information. In our case the information is carried by the `Channel mode` byte, part of the instruction packet sent from iOS to the HWA (section 5.2.3.4).

The encoding of property values and of the association mechanisms for the various ETs is pursued within the *encoder* singleton object (source file available in appendix J).

Figure 5.2.4: SiNW-bioFET acquisition LLT decomposition: the POCSiNWAcquisition LLT subclass is formed from a cascade of LLTs: it includes a reference to a POCLockInAmplificationAcquisition LLT object which itself includes a reference to a POCDSPCurrentReadoutAcquisition LLT object which includes two ETs. The POCLockInAmplificationAcquisition specifies the settings necessary for carrying out the lock-in synchronous recovery of the I_{DS} currents from the POCSiNWSensor object the POCSiNWSensorAcquisition object refers to. A POCWaveformGeneration ET object serves to specify the voltage biasing of the SiNW-bioFET.

ETs passage through the encoder is an obligatory step in *program execution*.

5.2.1.4 LOC program execution handling

User-interaction oriented software such as the one that prevails in smart-devices heavily relies on the Model-View-Controller or Model-View-Presenter (MVC/MVP) software *architectural pattern* [230, 231]. MVC implies the separation of the software *model*: the domain and data of the information that is being represented, from the *view*: the sensitive (e.g. visual) representation of the information, from the *controller*: the actor in charge of relating the model to the view. Under this paradigm, all the classes and methods presented in this section so far belong to the *model*. The integration of our *model* (e.g. architecting, function classes, etc.) within a user-centric smart application therefore necessitates the design of appropriate mechanisms and relationships both to the *views* (i.e. human-machine interactions) and to the *controllers*.

Although a detailed discussion of design options are out of the scope of this thesis, we must mention the overall articulation of our *model* to the common *controller* classes offered by iOS (and those conceptually equivalent for other mobile-application platforms such as Androïd). These relations are important since our PBD approach entails LOC execution handling at the mobile-software layer: the role of the mobile-software *controller* objects will be just as critical as the *model* we have been presenting thus far.

Table 5.1: Functional composition of a lock-in amplification acquisition LLT. The acquisition settings i.e. design variables of the function are specified on the left of the figure. They can be mapped to the different parameters available through the ETs offered by the platform (bottom of the table): Voltage Waveform Generation, Current Acquisition, and DSP.

	Initialisation potential	Initialisation duration	Waveform {sine,square,triangle}	Offset potential	Frequency	Amplitude	Number of excitation periods	Ramp start/stop potentials	Duty cycle	Sampling rate	Number of acq. points	Amplifier sensitivity	Sample averaging	Sampling window	FIR LPF	Decimation	Phase Sensitive Detection
Excitation frequency	—	—	—	—	X	—	—	—	—	—	—	—	—	—	—	—	X
Excitation amplitude	—	—	—	—	—	X	—	—	—	—	—	—	—	—	—	—	X
DC offset potential	—	—	—	X	—	—	—	—	—	—	—	—	—	—	—	—	X
Demodulation frequency	—	—	—	—	—	—	—	—	—	X	X	—	—	—	—	—	X
Sampling order	—	—	—	—	—	—	—	—	—	X	X	—	—	—	—	—	X
Frames per period	—	—	—	—	—	—	—	—	—	X	X	—	—	—	—	—	X
Nb. acq. periods	—	—	—	—	—	—	X	—	—	X	X	—	—	—	—	—	X
Sensitivity	—	—	—	—	—	—	—	—	—	—	—	X	—	—	—	—	X
Filtering	—	—	—	—	—	—	—	—	—	—	—	—	—	—	X	—	X
Output sampling rate	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	X	X
	Voltage Waveform Generation									Current Acquisition					DSP		

The relations established between POCLoCProgram objects and native Apple Cocoa application classes is given in figure 5.2.5a. The main application controller (POCAppDelegate) references a set of LOC programs possibly through relations to a POCLOC object. When the POC test is to be launched, the relevant POCLoCProgram object relies on its *controller* object to execute its tasks. LLTs and ETs are assigned the same *controller* object through a *delegation* mechanism (figure 5.2.5b) [231]. It is this controller that calls the `execute()` method for these tasks.

(a) (b)

Figure 5.2.5: Mobile application general architecture and LOC program delegation mechanism. The main application controller (POCAppDelegate) references to a set of LOC programs. When the POC test is to be launched, the relevant POCLoCProgram object relies on its *controller* object to execute its tasks. LLTs and ETs are assigned the same *controller* object through a *delegation* mechanism [231]. It is this controller that calls the `execute()` method for these tasks.

Whether the LOC program is statically defined via pre-linking of the task list or whether it is dynamically defined, all LLTs and ETs references are passed to a singleton *encoder* object when the time comes for their execution. The POCEncoder class singleton object is in charge of formatting the LLTs and ETs instructions so as to match the packet

structures both understood by iOS and the HWA. The UML sequence diagram of the execution of a LOC program containing a single LLT is given in figure 5.2.6.

The call to the `execute()` method of the POCLoCProgram object successively calls the individual ETs `execute()` operation. The latter calls the encoding of the ETs properties into instructions before the BLE controller transfers these instructions down to the HWA. As we mentioned before the role of the encoder is pivotal for a platform-based system. The implementation details for our encoder class is provided in appendix J. After the execution of each ET is completed, the HWA notifies the iOS layer by an End-Of-Transmission (EOT) message. In our example, the sole EOT message triggers the execution of the following task.

Figure 5.2.6: LOC Program sequence diagram for the execution of a LLT

5.2.2 Hardware design

5.2.2.1 Digital design

The instantiation of the HWA within the PBD framework may present medium to high complexity, depending on the instrumentation and control requirements set by the family of LOCs one is interested in interfacing (e.g. optical readout for fluorescence biosensors, on-chip magnets, EWOD digital microfluidics, high-impedance sensors, etc.). Parallelization or multiplexing requirements will add to that complexity by enlarging the initial functional, architectural and mapping opportunities of the platform. In our case the platform should allow the interfacing of several SiNW-bioFET LOC-*variants and* favour the adoption of new use-cases such as the interfacing of LOCs embedding other types of biosensors or actuators, or LOCs for which the operation of these sensors/actuators would require different settings. The initial product family: the set of compatible LOC-variants, was defined early on during concept definition.

In order to allow multiplexing, an initial number of 12 SiNW-bioFETs was agreed upon: this number corresponds to maximum number of SiNW-bioFETs that LOC-variants may embed if the LOC does not incorporate multiplexing capabilities itself. We implemented this capability through two channels enabling 3-terminal electrochemical measurements and ten additional channels for two-terminal current-sensing only. We enabled lock-in amplification on any of the twelve acquisition channels in search of *scalability*: as we saw in chapter 3, the increase in SiNW-bioFET sensitivity will require the associated scaling in current-readout instrumentation sensitivity and resolution. We identified that lock-in amplification was a strong candidate for this upscaling.

Considerations such as timeliness and schedulability requirements can constitute the entry points to a formal design method for embedded systems. This was not pursued within the scope of this work. We empirically considered schedulability and timeliness of our ETs to determine whether a Real-Time Operating System (RTOS) would be needed for the supervision of the HWA embedded software processes. The requirement for continuous sensor monitoring conditioned our parallel embedded hardware architecture.

Once our embedded architecture was defined, we specified the bidirectional communication protocol that would carry the LOC program tasks instructions to the HWA's communication module and upstream data from that communication module to the mobile-software layer. The details for our hardware platform and its embedded firmware are presented below.

The functional and performance requirements set by our elementary tasks led us to design the HWA depicted in figure 5.2.7. Hardware is powered at 3.3 V, enabling the single-supply operation of its embedded analog and digital components. It hosts a Bluetooth Low Energy (BLE) module (BLE112, Bluegiga Inc., Finland) enabling wireless communication with iOS. The BLE112 hosts its own CC2540 System-on-Chip (SOC), procuring it signal acquisition and digital communication capabilities including several Universal Asynchronous Receiver Transmitter (UART) modules, each configurable as a Serial Peripheral Interface (SPI). The BLE112 is supplemented by an API and a scripting language to facilitate its programming. Further details on the BLE firmware is elaborated further on in this chapter. The BLE112 is configured as a master to a dual-core, C2000 Digital Signal Controller (DSC - TMS320F28277D, Texas Instruments Inc., TX, USA) so far interfaced to via a development breakout board [106].

The DSC embeds all the peripherals required for implementing the waveform generation and synchronous acquisitions needed in order to carry out phase-synchronous detection: It hosts four separate ADCs operable either in 12-bit single-ended or 16-bit differential mode and four independent Digital to Analog to Converters (DAC). These peripherals are addressable by either of the two cores of the DSC [232]. Each core possesses a main Central Processing Unit (CPU) (proprietary F28x family processor, clocked at 200 MHz), as well as a proprietary hardware accelerator (Control Law Accelerator (CLA)) working independently from the F28x and granted access to some of the shared peripherals of the processor, including to the twelve Pulse Width Modulation (PWM) and the four DAC modules.

Both F28x processors have access to a Global Shared Random Access Memory (GSRAM) with 204 kB of capacity, on top of which they each own 72 kB of Local Shared RAM (LSRAM) not accessible by the other core. 1024 kB of Flash memory is also available to both F28x processors but was not used for our implementation in an attempt to ease firmware development efforts. A Direct Memory Addressing (DMA) module enables fast memory transfers from several relevant peripherals (e.g. ADC, PWM, SPI) to GSRAM. We must finally mention the Peripheral Interrupt Expansion (PIE) block, responsible for handling hardware interrupt priorities.

Figure 5.2.7: Hardware accessory SysML internal block diagram

5.2.2.2 Analog design

The analog peripherals of the DSC are interfaced to the *analog front end* of the HWA. The latter includes twelve buffering operational amplifier stages connected to the output of both of the DSCs DACs and to ten of its PWM modules. These amplifiers were referenced at 1.8 V (REF3318, Texas Instruments Inc. USA) to offer a theoretical AC output span of up to 1.5 V. Out of these twelve blocks, the 10 amplifiers interfaced to the PWM modules of the DSC operate in a Sallen-Key filter configuration figure 5.2.8. The other two output conditioning amplifiers are configured as potentiostatic cells. They are interfaced to the output of both DACs channels as described by others [78]. Each one of these channels uses two single-pole single-throw (SPST) switches (TS3A4742, Texas Instruments Inc., USA) to select either a two or three electrode electrochemistry configuration (figure 5.2.9).

The analog front-end inputs are designed as described in chapter 3: they materialize by twelve TIA stages followed by passive RC drivers for the F28377 ADC inputs. AD8608 quad amplifier Surface Mount Devices (SMD) (Analog Devices Inc., MA, USA) enabled the integration of the twelve TIA stages in only 3 distinct components. We provided each TIA with four selectable feedback networks each presenting a different feedback resistor value R_f in order to satisfy various sensitivities for various applications. The four gains were set to $1 \times 10^4, 1 \times 10^5, 1 \times 10^6$, and $3.6 \times 10^6\,\mathrm{V\,A^{-1}}$. The selection of these

Figure 5.2.8: Unity gain Sallen-Key reconstruction filter

Figure 5.2.9: Potentiostatic cell

Figure 5.2.10: Transimpedance amplifier offering four selectable gains in a Kelvin switching configuration [217]

feedback networks is made possible by the interfacing two SPST switches to the same General Purpose IO (GPIO) line of the DSC in a Kelvin switching configuration (figure 5.2.10). This double-switch design is recommended for limiting the gain error offset in transimpedance configurations relying on high feedback resistors [217]. In the attempt to reduce the number of on-board Integrated Circuits (IC), we laid out four TS3A4751 (Texas Instruments Inc. USA) quad switches around each of our AD8608 quad amplifier in order to accommodate for the four selectable gains of each TIA stage.

All the analog front-end components, the BLE module as well as SMA connectors for the inputs to the TIAs were embedded in a four-layer 174×104 mm PCB, the grounded-top layer of which is presented in figure 5.2.11. The DSC on the other end is, at this prototyping stage, still not integrated on that PCB but interfaced to via a development board made available by the manufacturer [233].

This hardware configuration allowed us to implement synchronous acquisitions, coupling any of the excitation channels to any of the acquisition channels. These twelve excitation channels and twelve acquisition channels constitute the possible electrical interfaces for mapping current-acquisitions to LOC-embedded sensors. The details of firmware implementation and specifications are presented in the next section.

Figure 5.2.11: Top layer of the PCB embedding the analog front end components and BLE module

5.2.3 Embedded software design

5.2.3.1 Overview

The embedded software allocation to the various embedded processors of the HWA is presented in figure 5.2.12 and the state-machine of the DSC is given in figure 5.2.13.

The dual-core architecture of the DSC is leveraged as follows: CPU1 is responsible for the communication with the BLE112 module. As it receives encoded iOS instructions it translates them into settings for each ET. Upon loading of each ET setting, CPU1 triggers task initialization and execution on CPU2. The second core of the DSC and its independent CLA are left responsible for voltage waveform generation functions and digital acquisitions by use of the DSC's embedded 12-bit DACs or PWM modules, and its four embedded 12-bits ADCs.

A degree of parallelism allows appropriate signal processing routines to be carried out on the F28x and CLA of CPU1, while CPU2 keeps running its excitation/acquisition pro-

Figure 5.2.12: Embedded software allocation

cesses. This feature will be elaborated on as it presents significant benefits for continuous sensor monitoring applications.

ADC acquisitions can only be performed on one of the available four ADC modules of the DSC, forcing current or impedimetric sensing to be carried out *sequentially*. Sensor excitation or biasing can be led, to some extent, in parallel: several DC voltages can be output simultaneously via the PWM modules, which is notably useful for the control of multi-terminal sensors/actuators that require several excitation signals (e.g. gate-biasing of three-terminal biosensors such as SiNW-bioFETs [234, 235]. The signal processing routines carried out by CPU1 allow digital filtering, decimation and current calibration of the sampled signals. Their output triggers the upstream transmission of the processed data to the BLE module that in turn relays it to iOS.

5.2.3.2 Waveform generation and synchronous detection

DAC-enabled waveform generation The voltage waveform required for the excitation of active sensors such as electrochemical cells or SiNW-bioFETs may be implemented

Figure 5.2.13: Embedded software simplified state-machine diagram

in numerous ways. Digital designs such as Direct Digital Synthesis or methods relying on Look-up Tables (LUTs) are commonly adopted in high-end waveform generation solutions. They provide significant benefits in terms of programmability and control and offer increasing bandwidth as digital processors keep on operating at higher clocks speeds. Although off-the-shelf (OTS) components are available and provide an integrated option for waveform generation, they do not help to avoid implementation complexity for our specific purpose. Our requirements for Phase-Synchronous Detection imply a waveform generation tightly coupled with synchronous analog to digital conversion. A stand-alone waveform generator IC would therefore necessitate the implementation of a frequency-tracking module (e.g. based on Phase-Locked Loop) in order to synchronise ADC conversion with the externally generated reference signal.

(a) (b)

Figure 5.2.14: Synchronous signal acquisition for $n = 4$ and $m = 2$. (a) Excitation signal (b) Noisy and phase shifted acquisition signal at the ADC input

In order to avoid such complexity and considering the low-bandwidth requirements for the AC recovery of SiNW-bioFET signals, we discarded the external IC option. We rely instead, on an internally generated digital waveform through the use of Direct Look-Up Tables (DLT).

These DLTs feed either the DSC's embedded DACs or PWM channels: depending on the waveform required by the ET under consideration, the values of a sinusoïdal, triangular or square- waveform over a single period are calculated at task-initialization-time and loaded in the LSRAM of DSC core 2. The number of values for this one signal period is set to $N_{\text{period}} = 2^n$, with N_{period} a multiple of 4. An illustration of DLT-based waveform generation with $n = 4$ is provided in figure 5.2.14a. Since phase-synchronous detection requires both an in-line and a quadrature demodulation, a minimum of 4 points need to be acquired over the N_{period} points per period of the excitation signal . We can define the number of acquisition points per period as $N_{\text{acqs/period}} = 2^m$, with $m <= n - 2$. Synchronization between excitation and acquisitions can be achieved by monitoring the index of the DLT value $i_{\text{DLT}_{\text{out}}}$ output by the targeted DAC or PWM module. We consider for now the DAC channels. We trigger ADC acquisition once $i_{\text{DLT}_{\text{out}}}$ mod $(2^n/2^m) \equiv 0$.

This coarse synchronicity can be refined by several methods, including increasing the number of points defining a period of the reference signal, or by timing more precisely when to trigger the ADC's sample and hold circuitry within the time lapse when $i_{\text{DLT}_{\text{out}}}$ mod $(2^n/2^m) \equiv 0$. We implemented the latter solution as follows (figure 5.2.15

and 5.2.16a): The 32-bit TIMER1 of DSC Core2 was set to loop to zero when reaching $t_{\mathrm{TIMER1}} = T/N_{\mathrm{period}} = T/2^n$, with T the excitation signal period. When TIMER1 reaches t_{TIMER1}, an interrupt triggers the CLA software routine responsible for updating the output potential of the DAC $V_{\mathrm{DAC_{out}}}$. Should the condition $i_{\mathrm{DLT_{out}}} \bmod (2^n/2^m) \equiv 0$ be met, the CLA triggers in its turn a second timer (TIMER2) via an Interrupt Sub-Routine (ISR) executed on the F28x processor. TIMER2 delays the ADC Sample and Hold (S/H) circuitry by another $t_{\mathrm{S/H_{delay}}}$, with $t_{\mathrm{S/H_{delay}}} \ll T/2^n$. When TIMER2 reaches its target value, the ADC S/H finally occurs. Timing of the S/H circuitry is theoretically as precise as the TIMER2 clock allows. If the DSC main system clock running at 200 MHz is used, then we may hope to achieve synchronicity with a precision of 5ns. In practice, this of course requires the fine tuning of the TIMER2 counter limit, so as to account for the delay between the moment TIMER2 interrupts the processor until the S/H is actually performed. Even in this best case scenario, clock jitter can be expected to limit synchronous detection accuracy. These latter concerns were not investigated further within the scope of this work.

Figure 5.2.15: Activity diagram for the continuous lock-in synchronous detection

Our implementation offers the possibility to oversample and average: TIMER0 can be started within a few clock cycles after the S/H has been triggered via TIMER2, and iteratively trigger ADC sampling so that $n_{\mathrm{oversampl.}}$ samples can be acquired and averaged over a period of time $t_{\mathrm{oversampl.}} = (n_{\mathrm{oversampl.}} - 1) \times t_{\mathrm{TIMER0}}$ seconds (figure 5.2.16a). Oversampling and averaging may be used to increase the Effective Number Of Bits (ENOB) of the digitization process, although it cannot be conveniently applied for synchronous detection [236].

These timing mechanisms also enabled us to implement synchronous acquisitions *windowing* (figure 5.2.16b): should pulse-waveforms be selected as an excitation signal, we gave the application designer the possibility to define acquisition points for both the positive and negative pulses, irrespectively of the duty cycle of the pulse-wave. This feature implies a periodic sampling that triggers twice during an excitation period. This feature is particularly useful for studying capacitive decays such as for Differential Pulse Voltammetry (DPV) on which we will elaborate further in section 5.3.3.

(a) (b)

Figure 5.2.16: Oversampling and windowing

The DLT method for digital sinusoïdal waveform generation is arguably one of the simplest to implement. It yet translates in relatively large errors, which increase if the number of points per signal period is reduced. Figure 5.2.17 illustrates this characteristic: the relative error over a single signal period is a low as 2% of the theoretical sine amplitude for DLT with 256 points when it goes up to about 30% when the number of points is decimated to 16 points per period. For our specific application, this error will propagate along the signal recovery chain and bias the values retrieved for the sensor impedance magnitude and phase. The derivation of the error propagation is yet not trivial as it will greatly depend on the acquisition (i.e. S/H, oversampling, etc.) settings. We can yet safely derive it if a 4 acquisition points per period scheme is used and if these points are synchronous to the maximum-zero-minimum-zero of the excitation signal (as represented in figure 5.2.14a): in the absence of any further errors (e.g. noise, distortions) along the excitation and acquisition channels, then the theoretical absolute value of the relative error on impedance recovery would equal that of the maximum relative error on waveform generation.

Numerous techniques provide with significantly better numerical approximations of trigonometric waveforms than DLT. Some rely on interpolation while others are specifically designed for hardware efficiency such as the COordinate Rotation DIgital Computer (CORDIC) algorithm [237]. These techniques are recommended when the requirements on the output signal Total Harmonic Distortion (THD) of the signal are stringent. The quality of these numerical approximations plays a significant role in the quality of the reconstructed analog signal. Although we acknowledge the importance of generating a high-quality reference signal in order to limit lock-in acquisition errors or for applications requiring low THD excitation signals, we were compelled to balance signal quality against implementation efforts. We therefore restricted our design to the DLT method for our

Figure 5.2.17: Relative error for Direct Look-up Table sinusoïdal waveform generation

DAC excitation channels. In the next section we present how we leveraged the PWM channels with the same purpose.

Pulse width modulation-enabled waveform generation We argued previously that instrumented LOC devices are increasingly designed so as to allow the multiplexed detection of a panel of analytes in a single test. For LOCs embedding current- or impedimetric biosensors, the multiplication of the number of on-chip sensors can only benefit the application if the interfacing instrumentation matches the highly-paralleled detection array. If timing requirements are low, then electronic multiplexers can be envisaged at the output of the excitation channels and, if the sensitivity budget allows it, at the input of the front end amplifiers. If instead strict timing and true parallelism are required, the instrumentation front-end and processing modules may need to be multiplied accordingly to the number of sensors and operate simultaneously. Such design prevails in large Micro-Electrode Array-based systems (MEA), sometimes comprehending several thousands of electrodes and electrical readout channels on a single die [238]. MEA-based systems present tremendous potential for micro- and nanoscale investigations of impedimetric or current based-biosystems. From an evolvability standpoint, MEAs therefore benefit from several indisputable advantages over OTS components-based systems incapable of providing such levels of *redundancy*. Substantial care often also goes to make these MEA highly *reconfigurable*.

Our design and implementation efforts have obviously not intended to compete with MEA on these aspects. In an attempt to address comprehensively all change-related *-ilities* we yet attempted to provide our platform with the capability to output voltage waveforms on several independent channels simultaneously. The Pulse Width Modulation (PWM) modules of the DSC offer the opportunity to do so without adding extra components and interfaces to the design.

PWM is a commonly used technique in automation and control, most notoriously for driving electrical DC motors. PWM consists in delivering electrical potential pulses periodically to a load. The modulation of the pulse width (i.e the duty cycle of the pulse) serves as a way to efficiently control the average potential delivered to the load over time, while keeping the power consumption low (power consumption should near zero during the low part of the duty cycle). PWM is a functionality offered on numerous MCUs dedicated to industrial automation applications. PWM may yet also be used as an alternative to conventional DAC for low-bandwidth, precision-limited applications [239]:

the finely-tuned low-pass filtering (LPF) of a a periodic PWM output signal gives a DC output. By modulating the duty cycle of the pulse waveform output by the PWM, we may therefore achieve the output of an arbitrary waveform at the output of the LPF. The ideal output potential is defined by:

$$V_{\text{avg}} = D(V_{\text{high}} - V_{\text{low}}) + V_{\text{offset}}, \tag{5.2.1}$$

with D the duty cycle of the pulse waveform. The F28377D presents a high reference potential $V_{\text{high}} = 3.3\,\text{V}$ and a low reference $V_{\text{low}} = 0\,\text{V}$. We will neglect the V_{offset} as the latter could arguably be compensated for by an appropriate software calibration routine. D is limited in accuracy by the resolution of the digital counter of the PWM module. The ePWM modules of the F28377D have a 16-bit resolution. For a desired V_{avg} (i.e. the value pointed to by the DLT index), the PWM duty on register should thus be:

$$\text{REG}_{\text{PWM}_{\text{ON}}} = \frac{V_{\text{avg}}}{V_{\text{high}} - V_{\text{low}}}.2^{16}, \tag{5.2.2}$$

The output of a waveform can then be achieved by updating the value of the $\text{REG}_{\text{PWM}_{\text{ON}}}$ register at a periodic time interval just like one would do for a conventional DAC channel. Our DLT waveform generation was thus leveraged in the same way both for two DAC channels and ten PWM channels of the F28377D.

The modulation frequencies $f_{\text{PWM}_{\text{mod}}}$ for all PWM module is the main frequency harmonic at which the pulses are delivered. It sensibly conditions the quality of the output waveform. Each of the PWM modules on the processor may generate pulses at different modulation frequencies f_{PWM}. Unfortunately though, PWM waveform generation relies on timer interrupts (i.e. TIMER1) triggered at fixed time steps in order to update the output voltage V_{avg} to the appropriate value. This in turn means that in order to simultaneously output waveforms with different main frequency harmonic we would need to trigger interrupts at various points in time for each channels. In most cases (i.e. if the desired excitation frequencies for all channels are not all multiples of one another) the execution of some interrupts would be delayed by the completion of the execution of the previous, compromising the shape of the output waveform. A non-problematic scenario is to set the $\text{REG}_{\text{PWM}_{\text{ON}}}$ constant and only allow the output of DC potentials for the various PWM modules (i.e. do not require timer interrupts for waveform output value update).

Although other algorithms may be investigated for the optimisation of multichannel independent and simultaneous waveform generation, we limited ourselves to these specifications, allowing the output of ten DC voltages simultaneously while one of the DAC channel may be used for the output of any triangular, square, sine or DC waveform over a bandwidth of about 50 kHz.

Several dependencies condition the bandwidth and THD performances of the waveform generators both for the PWM and DAC channels: our DLT approach means that bandwidth can be increased by the reduction of the number of points per excitation period, although at the cost of greater error and THD. Fixing the number of points per excitation period is a way to limit the amount of THD. Maximum excitation signal bandwidth will then be limited by the execution time of the ADC acquisition interrupt routines or by the execution time of the DAC or PWM register update routine themselves.

The *scalability* of our waveform generators bandwidth may be an interesting research point, directly related to overall system *evolvability*: as we will illustrate in the next section with the example of impedance flow cytometry, the ability of the system to scale up the *excitation* signal bandwidth may open up new biosensing applications, if the *acquisition* bandwidth scales up accordingly. We support our research on system *scalability* with the investigation of our system continuous *acquisition* bandwidth in chapter 5.

5.2.3.3 Digital signal processing for continuous sensor monitoring

The capability of continuously performing lock-in synchronous detection over an unbounded period of time represents a considerable opportunity. Many biosensors and diagnostic methods rely on the the monitoring of AC signal amplitude and phase over significant periods of time. Some of them require to do so in order to detect what are often considered *discrete* events throughout the entire acquisition time. A good example of such applications is flow cytometry: biological cells are guided through a small microfluidic channel while an excitation source and detection unit continuously probes the channel in order to detect single cells passing through, and to classify each individual cell according to the properties of the acquired signal (e.g. amplitude, phase offset, composite score). Flow cytometry is a vast domain, comprehending many excitation/detection technologies, the gold standard of which remains optical-based methods. More recently though, impedance-based flow cytometry has generated a growing interest as a cheaper alternative to optical detection for a subset of common cytometric applications [240]. The technique is based on the continuous monitoring of impedance variations within a microfluidic channel. Events such as the crossing of the detection zone by a small particle will manifest by a drop or increase in impedance magnitude and/or a shift in signal phase. The specific features of this signal change may help in further assessing the shape, type, etc. of the particle and eventually draw various conclusions on the overall sample.

In our platform-based approach to POC system design, we must acknowledge the potential of such technique: although our starting point i.e. the SiNW-bioFET does not imply any sort of requirement for continuous monitoring, other biosensing technologies embedded in LOCs designed for other applications may make use of that functionality.

The term *continuous* monitoring needs to be carefully considered. Digital lock-in synchronous detection obviously implies the processing of *discrete* signals. By *continuous*, we here refer to the uninterrupted monitoring of the sensing signal. Continuous lock-in sychronous detection will therefore be characterized by the signal *throughput* rate: the average number of impedance measurement points/per seconds streamed to the mobile-software layer. This throughput rate will be dependant both on the excitation and acquisition settings (e.g. the AC excitation frequency) and on the upstream DSP and communication specifications of the system.

The parametric determination of the maximum excitation frequency for which this scheme executes successfully is not trivial. Dependencies between the size of the input buffers, the excitation frequency, the PSD algorithm execution time and the bandwidth of the wireless communication module (BLE112) contribute to the complexity of the issue. We discuss below the main implementation details involved:

Acquisition buffers We implemented the frame-based processing of the digitized input signal $x(n)$, where n represent the index of the 12-bit ADC sample from time $t = 0$, corresponding to the beginning of the acquisition process. `signalBufferStage1A` and `signalBufferStage1B` are the two digitized signal input buffers allocated in GSRAM. They are alternatively filled with the (averaged) digitized input signal $x(n)$, once the relevant ISRs have executed (figure 5.2.15). Along with the digitized input signal, the values of both I- and Q- reference signals corresponding to the acquisition sample n are also stored in GSRAM. When the running buffer is full (or if the acquisition is complete), the fullDMABufferIntISR (or endTask) routine triggers the execution of the first stage of the Phase-Sensitive Detection process on DSC Core1 (figure 5.2.15). This same triggering event entails the swapping of the pointer to the alternate input buffer (and similarly of the pointer to the I- and Q- alternate buffers), in a negligible time span, so that new samples can be digitized while others are being processed. The repetition of this crude parallel execution allows continuous synchronous detection if the DSP (i.e. PSD, calibration) and

data transfer to the higher-abstraction mobile-software layer of the system are completed *within the time frame in which the running input buffer is filled.* Larger buffers will tend to increase maximum excitation frequency since they will lead to calling the subroutines for buffer swapping and PSD initialisation less often, thereby reducing overhead execution time. The memory architecture of the F28377D DSC brought us to specify the size of each input buffer to 1024×4 bytes/float = 4 kB.

FIR filter order and decimation As we discussed in chapter 3, the sensitive recovery of SiNW-bioFET impedance magnitude and phase using lock-in amplification depends on the FIR filter selected for low-pass filtering the output of the demodulator. Sharper roll-off filters will exhibit a greater number of taps, which in turn will extend the computational time for the PSD.

Decimation is often used to remedy DSP filtering execution time issues: low-pass decimation filtering consists in computing the convolution of the downsampled digitized input signal. If we consider the input signal $x(n)$ then the FIR low-pass filtering output of $x(n)$ by a N tap filter is given by:

$$\hat{x}(n) = h(n) * x(n) = \sum_{i=1}^{N} x(n-i)h(i) \tag{5.2.3}$$

The low-pass *decimation* filtering of $x(n)$ is defined so that:

$$\hat{x}_{(\downarrow M)}(n) = \hat{x}(nM) = \sum_{i=1}^{N} x(nM-i)h(i), \tag{5.2.4}$$

where M is the *decimation factor*: the factor by which the input sample rate is reduced. One can identify that the introduction of M in the convolution allows the discarding of $M-1$ out of M samples for the filtering process. In practice this often translates in computational gains in the order of magnitude of M as only a small fraction of the overall acquired samples are convoluted. In order for decimation not to violate the Shannon-Nyquist criteria, the input signal to the low-pass decimator filter should have been pre-filtered in order to remove signal harmonics laying beyond twice the desired new sampling rate. Should the input signal $x(n)$ be specified over a bandwidth f_b, then in order to decimate $x(n)$ by M, $x(n)$ must be low-pass filtered so that $f_{b\,\mathrm{LPF}} < f_b/2M$. For PSD, this extra filtering step is not necessary as the demodulation has brought the signal of interest to baseband and the decimation process can be carried out together with the FIR low-pass filtering.

Decimation thus represents a way to increase the maximum excitation frequency for which uninterrupted impedance signal monitoring can be achieved: even though the acquisition buffers will fill up faster at increasing frequencies, fewer input samples will be processed and eventually transmitted over wireless communication. An important point yet, is that decimation comes at the price of signal bandwidth reduction: an impedance signal recovered for an acquisition frequency f_{acq} and decimated by a factor M will only exhibit a bandwidth of $f_{(\downarrow M)} = f_{\mathrm{acq}}/2M$ which may result in the loss of signal information.

We may use the electrical impedance spectroscopy to illustrate this property. Let us consider that the method is used to count cells flowing through a microfluidic channel by monitoring impedance variations for an acquisition frequency $f_{\mathrm{exc}} = 10\,\mathrm{MHz}$ and sampling rate 40 MSPS (Mega-Samples Per Seconds). If the instrumentation allows the non decimated transfer of all measurement points, then signal variations with a main frequency of up to 20 MHz may be recovered. Signals from cells passing through the detector every $1/(20\times10^6)$ seconds may in theory be picked up by the monitor. Conversely, if decimation is used so that for the same excitation frequency, the output data throughput rate is down

to 2 MSPS, then the cell throughput will also need to be reduced by a factor of ten in order not to miss particle counts. Decimation therefore cannot make up for the intrinsic bandwidth performance limitations of the system.

Dual-stage filtering Decimation and low-pass filtering can be performed over several stages both in order an attempt to better reject noise and/or to diminish computational load by decimating at each stage. As our DSC Core2 is in charge of ensuring synchronous excitation and sampling, DSC Core1 is responsible for carrying out the AC-coupling, the demodulation, the FIR filtering and decimation and post-processing necessary in order to retrieve the impedance amplitude and phase values. It does so when interrupted by Core2 once an acquisition buffer is full or when the acquisition is completed (figure 5.2.15). The first processing stage, PSD Stage 1, first performs AC-coupling on the digitized input signal: the 12-bit ADC samples reflecting the $V_{\mathrm{ADC_{in}}}$ potential fluctuations are averaged over 3 signal periods. The resulting mean average value is an estimate of the DC biasing offset over which the AC signal was carried throughout the acquisition chain. As we saw in section 4.3.2, this offset needs to be subtracted to the input signal before demodulation. After this AC-coupling follows demodulation and the first stage of FIR low-pass decimation filtering on both the in-line and quadrature demodulation products. The demodulation of the I- and Q- signal components are carried out serially. The output of the FIR filter is then calibrated to retrieve the current amplitude of the sensor in a floating-point format.

Calibration As we discussed in section 5.2.2.1 the digitization of the amplified input signal is carried out using the single-supplied embedded ADC of the F28377D processor. In an ideal scenario, the reference potential used for biasing the TIA amplifier could be used to compute the current signal I_{sens} that we are interested in recovering. Our TIA is biased at $V_{\mathrm{ref}} = 1.8\,\mathrm{V}$. If I_{sens} is strictly equal to zero, then the ideal 12-bit code at the output of the ADC should be:

$$\mathrm{Code}(I_{\mathrm{sens}} = 0) = 2^{12}.\frac{1.8}{V_{\text{full-scale}}} = 4096.\frac{1.8}{3.3} \simeq 2234 \tag{5.2.5}$$

In practice, the non-idealities of both analog components and of the digitization process itself will alter that baseline value. As we saw in chapter 3, the TIA input bias current and offset voltage are likely to alter the effective biasing voltage at the output of the amplifier. Furthermore, the ADC itself is non-ideal and may present an input offset voltage that will translate by an output count error.

Aside from this *offset* error, the acquisition system will exhibit a *gain* error. First of all the ideal transimpedance gain of the TIA only falls within 0.1-5% of the nominal values for the resistors in the feedback networks. The theoretical specification of the TIA gains are as accurate and precise as the tolerances on the feedback components allow it. We thus need to account for these uncertainties. Also, the ADC itself may exhibit a gain error, due to tolerances on its internal reference voltage.

Several calibration methods may be adopted in order to correct for these systemic errors, accounting for the varied offset and gain tolerances throughout the entire acquisition chain. Although hardware calibration may be useful for complex instrumentation, we satisfied with a 2-point software calibration scheme: in order to correct both offset and gain-related errors, we recovered the ADC output code for 2 distinct current values applied at the TIAs input. The results of this procedure are given in table 5.8. We used $I_{\mathrm{TIA_{in}}} = 0$ for the first point. The ADC output code gave us the offset of the entire acquisition chain (n.b. measurements were averaged over 1024 acquisitions). We repeated this operation this time with an input current that would bring the output of the TIA close to its full-scale potential. We made these measurements for all four available gains of each TIA

Table 5.2: Acquisition calibration data for the first TIA_1 on the PCB

		Current input low (A)	Measured ADC code low (12 bit)	Ideal ADC code low (12 bit)
TIA Gain (V/A)	1×10^4	0.0	2200	2234
	1×10^5	0.0	2226	2234
	1×10^6	0.0	2225	2234
	3.6×10^6	0.0	2211	2234

		Current input high (A)	Measured ADC code high (12 bit)	Ideal ADC code high (12 bit)
TIA Gain (V/A)	1×10^4	-1.2×10^{-4}	3713	3724
	1×10^5	-1.2×10^{-5}	3714	3724
	1×10^6	-1.2×10^{-6}	3703	3724
	3.6×10^6	-3×10^{-6}	3531	3708

		Gain (V^{-1})	Reciprocal gain (V)	Input offset (A)
TIA Gain (V/A)	1×10^4	-1.2608×10^{7}	-7.93126×10^{-8}	$-1{,}744878\times10^{-4}$
	1×10^5	-1.24×10^{8}	-8.06452×10^{-9}	$-1{,}795161\times10^{-5}$
	1×10^6	-1.23167×10^{9}	-8.11908×10^{-10}	$-1{,}806495E\times10^{-6}$
	3.6×10^6	-4.4×10^{9}	-2.27273×10^{-10}	$-5{,}025\times10^{-7}$

stage. The gain of the entire acquisition chain could thus be calculated as:

$$\text{Gain} = \frac{\text{Code}(I_{90\%\text{full-scale}}) - \text{Code}(I_{\text{sens}} = 0)}{I_{90\%\text{full-scale}} - 0} \tag{5.2.6}$$

By reversing this relation and defining the reciprocal gain of the acquisition chain as:

$$\text{Reciprocal Gain} = \frac{1}{\text{Gain}}, \tag{5.2.7}$$

we were now capable of matching ADC input codes to current values with

$$I_{sens} = \text{Reciprocal Gain} \times (\text{Code}_{\text{ADC}} - \text{Code}(I_{\text{sens}} = 0)) \tag{5.2.8}$$

This computation is performed on the output of the first FIR filtering and decimation stage. Once completed, a second stage FIR filtering and decimation is automatically started: PSD Stage 2 (figure 5.2.15).

Just like the input acquisition buffers are duplicated for avoiding to stall the ADC sampling process, the output buffers of PSD Stage 1 also have alternates. When PSD Stage 2 starts, it takes PSD Stage 1 running output buffer as input, and swaps the running *output* buffer with its alternate. Finally upon completion of the second filtering stage, the magnitude and phase of the sensor impedance can be computed from the calibrated current values and the amplitude of the excitation voltage waveform. The transfer of the impedance magnitude and phase to the BLE module can then start.

5.2.3.4 Wireless communication protocol

Change-absorbing smart-device/HWA interface Special care was taken in designing the wireless communication protocol specifying the interface between the mobile-software and embedded-firmware layers. As we argued in section 3.1.4.2, the definition of interfaces is critical in attempting to uncouple system modules. The wireless communication protocol defining the interface between iOS and the embedded firmware is probably the most sensitive interface in the entire design as it must vehicle the encoded information

of both LOC-program commands and retrieved data, virtually representing information relevant to the function space, architectural space, and mapping between the two. This interface bares the risk to *carry* or *multiply* changes made to one of the layers to the next, potentially in a cascaded manner, necessitating large reengineering efforts throughout the platform architecture. Its sensitivity to change can be assessed or predicted using techniques such as CPA as introduced earlier. In an attempt to limit the sensitivity of this wireless interface to change,

we strived to design our wireless protocol as a *change absorber*, with the objective *not* to propagate anticipated changes in the functional or architectural space to other system modules or layers across the interface.

Bluetooth Low Energy protocol implementation Bluetooth Low Energy (BLE) is a wireless Bluetooth specification mainly addressing applications with low bandwidth and low power consumption requirements [241]. Current trends towards an Internet of Things (IoT) have brought many BLE applications to the consumer market. Examples in the fields of wellness and healthcare include wearables such as heart rate monitors, glucose meters or even blood pressure sensors that can be readily interfaced to a smartphone for activity or basic health metrics tracking.

Figure 5.2.18: Custom GATT Profile SysML block definition diagram

Most BLE applications rely on Generic Attributes (GATT) *profiles*: GATT profiles are abstracted software constructs that enable BLE *servers* to deliver *services*, implemented according to the BLE communication protocol, to a number of different *clients*. Given the increasing number of available devices and manufacturers offering IoT solutions, an effort has been undertaken for the development of several standardized GATT profiles [242]. The HRP profile for instance, is specific to heart-rate monitors: the *server* should be a heart rate sensor while the *client* should be the data collector, most likely a smart-device. For applications falling outside of these standards, BLE developers have the opportunity to implement custom GATT profiles, including services and characteristics that will best fit their server-client data interchange requirements.

We relied on the BLE protocol to establish wireless communication between our HWA

and the iOS mobile-software layer of our platform. We undertook the development of a custom BLE profile with two main objectives: first, to allow the protocolar communication of smarphone-instructed commands to the HWA and second, to enable the upstream transfer of processed data from the HWA to the smart-mobile software layer (figure 5.2.18).

The transfer of processed data at the output of the DSP routines goes through the `AccessoryReadout` attribute. The latter is in charge of carrying acquisition data upstream to the iOS mobile-software layer

Table 5.3: `AccessoryReadout` attribute value packet structure

Byte 0	Byte 1	Byte 2-18	Byte 19
Ch. ID	Data Mode	Calibrated floating-point current/impedance values	EOT

The BLE specification requires the GATT attributes (e.g. characteristic values) to be specified over a maximum of 20 bytes. `AccessoryReadout` (and `AccessoryConfig`) include the channel ID over which the acquisition has been performed. An End-Of-Transmission opcode (EOT) carries a specific value when the transmission of acquisition data has been completed. The EOT tag allows the iOS software to assess the completion of the task under execution. The remaining bytes of the `AccessoryReadout` characteristic carry the floating point values of the calibrated data points reconstructed by our DSP routines. As we mentioned before, for the specific lock-in amplification case, the completion of PSD Stage 2 triggers the transfer of impedance magnitude and phase values. The `AccessoryReadout` packet structure is given in table 5.3. The short length of the attribute data packet puts a first limitation on the BLE protocol bandwidth. The throughput rate is further limited by the minimum duration that needs to separate each attribute transfer sent by the *server* (HWA) to the smarphone *client*. For iOS clients, attributes can only be sent every 20 ms or more, putting tremendous limitations on the maximum bandwidth achievable by BLE.

The `AccessoryConfig` attribute embeds the iOS-encoded instructions to the HWA. The packet structures for `AccessoryConfig` are presented in tables 5.4, 5.5 and 5.6. They were defined so as to allow the specification of all tuneable settings for the execution of the HWA embedded voltage excitation, acquisitions and DSP functions. Their definition was pivotal in our PBD design approach: the protocol defining `AccessoryConfig` articulates the interface between the mobile-software function space and the HWA architectural space: the platform *mapping* process, described in section 5.1.2 must allow the encoding of any function instance onto length-limited opcodes carried by `AccessoryConfig`. Conversely, the HWA will need to decode all the architectural information from these opcodes and interpret the desired functionality accordingly.

In order to absorb a potential expansion of the function or architectural space of the platform, the attributes carried by `AccessoryConfig` were represented with a provisional number of extra bits: the `Channel mode` opcode byte for instance, is designed to encode the operational mode of a target excitation or acquisition channel in a single byte. Although currently only 3 modes have been implemented, an entire byte encodes for the `Channel mode` attribute, making it possible for 255 modes to be represented without having to modify the protocol. If `Channel ID` points to an acquisition channel, then `Channel mode` will specify whether the targeted acquisition channel is synchronous to one of the HWA excitation channels and what configuration options are implied. Similarly, the `Channel ID` attribute is encoded over one byte, enabling the mapping of up to 255 acquisition channels or 255 excitation channels should hardware *redundancy* be increased without affecting the iOS-DSC firmware interface.

The low bandwidth of the BLE specification put relatively soft requirements on the

Table 5.4: `AccessoryConfig` attribute value packet structure for excitation waveform specifications

	Byte 0	Byte 1	Byte 2-3	Byte 4-7	Byte 8-11	Byte 12-15
Packet 1	Ch. ID	Channel Mode	Nb exc. periods	Exc. ampl.	Exc. freq.	Exc. DC offset

	Byte 0	Byte 1	Byte 2-5	Byte 6-9	Byte 10-13	Byte 14-15
Packet 2	Ch. ID	Channel Mode	Ramp incr.	Init V.	Final V.	Duty cycle

Table 5.5: `AccessoryConfig` attribute value packet structure for asynchronous acquisitions specifications

	Byte 0	Byte 1	Byte 2	Byte 3-4		Byte 5-8
Default	Ch. ID	Channel mode	NaN	Nb. acqs. points		Sampling freq.
	Byte 9 (0:3)	Byte 9(4:7)- 10	Byte 11 (0:3)	Byte 11 (4:7)	Byte 12	Byte 19
	NaN	NaN	Rf gain	FIR sel.	Decim.	Ch. status

Table 5.6: `AccessoryConfig` attribute value packet structure for synchronous acquisitions specifications

	Byte 0	Byte 1	Byte 2	Byte 3-4		Byte 5-8
Sync. acq.	Ch. ID	Channel mode	Paired exc.channel	Nb. acqs. periods		Sampling freq.
	Byte 9 (0:3)	Byte 9 (4:7)- 10	Byte 11 (0:3)	Byte 11 (4:7)	Byte 12	Byte 19
	Sampl. order	FPP	Rf gain	FIR sel.	Decim.	Ch. status

communication speeds between the DSC and BLE module. The BLE112 was configured as a master to the DSC. The processors were interfaced via a 3-wire SPI procotol. The implementation C code of the DSC and the BGScript scripting-language firmware for the BLE112 are given in appendix K

This section achieves to present the implementation of our initial platform. In the ensuing section, we illustrate how our mobile-software API can be used in order to specify the operation of a compatible LOC-variant [226] (appendix B).

5.2.3.5 Illustrative application

The set of functional and architectural classes and methods discussed in section 5.2.1.3 constitutes the foundation of an API for the specification of LOC operations. As an illustrative example of how these software elements can be used, we implemented an iOS application, including *models*, *views* and *controllers*. Our POCApp application intends to demonstrate how LOC architecture, applicable functions, the mapping of these functions onto the architectural elements and finally the composition and execution of LOC programs can be done *at run-time* through the simple use of a mobile Graphical User Interface (GUI). We must disclose that our iOS application has the sole purpose of illustrating the back-end capabilities of our API and does not illustrate what end-user solutions would look or behave like.

We once again rely on the specification of a LOC program aimed at operating a single SiNW-bioFET embedded on a passive (i.e. driven by capillary flow) microfluidic LOC. Figure 5.2.20 a to c illustrate the steps leading to the addition of a SiNW-bioFET sensor onto the cyber- representation of the LOC we would like to interface to the HWA. Figure 5.2.19d and 5.2.19f show how the functional definition of a SiNW-bioFET acquisition LLT can be carried out and directly mapped to the cell terminals defined in the first step. Figure 5.2.20a shows that a HLT can then be associated with the acquisition before the program is generated (figure 5.2.20b) by adding successively the LLT and appropriate HLT to the program task list. Figure 5.2.20c finally shows the initial experimental results

Figure 5.2.19: iOS LOC architecting, functional definition and mapping

obtained for the electrical gating of a SiNW-bioFET. The sensor is still currently under development and has not yet proven to be reliable enough to detect biological molecules in solution. The impedance versus time plot presented here was acquired on a SiNW of length $L = 10 \times 10^{-6}\,\mathrm{m}$, width $W = 2 \times 10^{-6}\,\mathrm{m}$ and height $H = 50 \times 10^{-9}\,\mathrm{m}$. Lock-in synchronous detection was performed at an excitation frequency of 20 Hz with a 0.3 V amplitude. The steps identifiable on the graph at regular time interval correspond to steps in back-gate potentials applied to the sensor via an external Keithley 2400 Sourcemeter

(a) (b) (c)

Figure 5.2.20: iOS configuration of high-level tasks, program generation and execution

(Tektronix, Oregon, USA).

5.3 Platform evolution

Electrochemical methods for biological sensing have aroused significant interest for the POC testing of electro-active compounds. The low-cost and relative simplicity of the instrumentation of electrochemical biosensors is often cited as one of the key advantages of electrochemical analysis [59, 243]. Techniques such as Cyclic Voltammetry (CV) rely on the electrical excitation of a sample using a potential waveform and the acquisition of the currents resulting from the presence of ionic species in solution. Electro-active compounds will react to the electrical potential difference applied between the Counter Electrode (CE) and the Working Electrode (WE) of an electrochemical cell (figure 5.2.9), and either release or uptake electrons depending on the nature of the chemical reaction animating them: oxidation vs. reduction. These chemical processes can be monitored most often by measuring the current flowing through the WE. The two-electrode configuration CE-WE is often complemented by a third Reference Electrode (RE): in order to better control the electrochemical reactions, one needs to ensure that the current sinking or sourcing into the solution will not alter the potential applied across the sample. The RE provides a high-impedance and highly stable reference potential without either sinking or sourcing currents: The CE endorses this responsibility.

Although an extensive discussion on electroanalytical chemistry or the subtleties of its specific instrumentation is out of the scope of this thesis work, we must take notice of the important *commonalities* shared between the SiNW-bioFET instrumentation and the basic architecture of a potentiostat: the instrument specific to electrochemical analysis. For potentiostatic measurements, the CE of an electrochemical cell must often be driven by triangular, or pulse voltage waveforms with relatively low bandwidth, while the WE requires to be interfaced to a current amplification analog front-end. The hardware archi-

Figure 5.3.1: Cyclic voltammetry acquisitions are carried out successively at decreasing sensitivities if current saturation is detected within the potential range where calibration is carried out. Cyclic voltammetry LLT1 is composed of two synchronous elementary tasks: triangular waveform generation and synchronous sampling

tecture required for the instrumentation of a SiNW-bioFET is therefore sensibly similar to that required for carrying out electrochemistry except for the capabilities offered by the RE. We identified these similarities during the functional analysis we initially carried out for our platform design (section 5.1.1) and provided both our DAC-channel excitation channels with the capability to be operated in a 3-electrode configuration (section 5.2.2.1).

Before we now elaborate on how PBD and the *reconfigurability* mechanisms implied by PBD allowed us to evolve our platform into one capable of electrochemical analysis, we must acknowledge that reconfigurability *alone* would not have sufficed to implement these new functionalities. Had we not anticipated the requirements for adding the RE-related circuitry at initial design time, we of course could not have avoided hardware and firmware modifications (e.g. switches, dedicated DSC GPIO, modification of opcodes interpretation in the BLE protocol, etc.) and subsequent testing, validation, etc. Nevertheless, design for evolvability, including composable ETs at the mobile-software layer and a change-absorbing BLE protocol, would still have been instrumental in *limiting* the implementation efforts associated with the inclusion of the RE hardware in the platform, merely requiring the addition of appropriate embedded code to the HWA rather than a complete remodelling of the mobile-software, BLE interface and embedded software modules altogether. The significance of the anticipated difference in reengineering effort of a platform vs. evolvable platform could be assessed by looking at how the addition of the RE channel and of its controlling elements would have propagated through the platform architecture. Methods

Figure 5.3.2: Electrochemical analysis acquisition LLTs decomposition

such as CPA may then have been of use to determine the relative degree of *modularity* of the platform vs. evolvable platform, partly contributing to the overall assessment of evolvability [139].

5.3.1 Electrochemical analysis functional composition

In a similar way to how we specified SiNW-bioFET acquisitions, we *composed* various potentiostatic electrochemistry acquisition tasks, using a waveform generation function coupled to a synchronous acquisition. The UML diagram of the resulting classes is presented in figure 5.3.2. Once again the prerequisite for this functional composition is that all the design variables that define each function finds an equivalent in the association of ETs (table 5.7) [226]. Note the inheritance of Differential Pulse Voltammetry (DPV) from Square-Wave Cyclic voltammetry. The two analytical schemes both rely on a pulse-waveform excitation. The only difference between them is that SWCV uses a 50% duty cycle whereas DPV may use custom settings.

Once implemented, these classes and the encoding/decoding schemes for their instructions enabled us to specify the basic programs necessary to operate electrochemical cells.

5.3.2 Electrochemical analysis validation

We demonstrated the electrochemical capabilities of our system by interfacing one of its potentiostatic channels to Screen-Printed Gold-Electrodes (SPGE) (C223AT, DropSens) and benchmarked against a commercially available AutoLab PGSTAT302N potentiostat (Metrohm, The Netherlands). We investigated a ferricyanide/ferrocyanide solution using cyclic voltammetry. The ferri/ferrocyanide redox couple is commonly used in electrochemistry and reacts so that:

Table 5.7: Functional composition of cyclic voltammetry (CV). The acquisition settings i.e. design variables for CV, can be mapped to the different parameters available through the *elementary tasks* offered by the platform.

	Initialisation potential	Initialisation duration	Waveform {sine,square,triangle}	Offset potential	Frequency	Amplitude	Number of excitation periods	Ramp start/stop potentials	Duty cycle	Sampling rate	Number of acq. points	Amplifier sensitivity	Sample averaging	Sampling window	FIR LPF	Decimation	Phase Sensitive Detection
Startup potential	X	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Startup time	—	X	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Low potential	—	—	—	X	X	X	—	—	—	—	—	—	—	—	—	—	—
High potential	—	—	—	X	X	X	—	—	—	—	—	—	—	—	—	—	—
Scan rate	—	—	—	—	X	—	—	—	—	—	—	—	—	—	—	—	—
Number of cycles	—	—	—	—	—	—	X	—	—	—	—	—	—	—	—	—	—
Acq. points/cycle	—	—	—	—	—	—	—	—	—	X	X	—	—	—	—	—	—
Sensitivity	—	—	—	—	—	—	—	—	—	—	—	X	—	—	—	—	—
Filtering	—	—	—	—	—	—	—	—	—	—	—	—	—	—	X	—	—
	Voltage Waveform Generation									Current Acquisition					DSP		

$$[\mathrm{Fe(CN)_6}]^{4-} \rightleftharpoons [\mathrm{Fe(CN)_6}]^{3-} + e^- \tag{5.3.1}$$

The oxidation of $[\mathrm{Fe(CN)_6}]^{4-}$ is obtained from the forward sweep (negative to positive potentials) of the triangular excitation waveform specific to CV. It results in an anodic current peak at the WE before decaying. Conversely the reduction of $\mathrm{Fe(CN)_6}]^{3-}$ takes place during the backward sweep (from positive to negative potentials). It fosters a negative cathodic current peak. The $[\mathrm{Fe(CN)_6}]^{4-}/[\mathrm{Fe(CN)_6}]^{3-}$ couple reaction is reversible: the transfer of electrons for both oxidation and reduction occurs relatively fast, which translates in cathodic and anodic peak currents of roughly equal amplitudes (figure 5.3.3).

We carried out the CV of 70 µL of a ferri/ferrocyanide solution at $1 \times 10^{-4}\,\mathrm{mol\,L^{-1}}$ first on the AutoLab PGSTAT302N and repeated the measurement on our platform. For that we specified a POCCyclicVoltammetryAcquisition task, with three CV cycles (forward and backward sweep), each acquiring 256 points for potentials ranging from $E = -1\,\mathrm{V}$ to $E = 1\,\mathrm{V}$.

The comparison of the voltammograms obtained from both methods reveals mostly a current measurement offset and a peak-to-peak amplitude discrepancy of about 10%. We may conjecture that the two-point calibration of our current amplifiers is insufficiently accurate and that it combines with gain and offset errors introduced by the waveform generator itself. The error of the latter were also corrected using a two point calibration scheme but it seems that overall system accuracy still suffers from the absence of a potential monitor at the output of the DACs to rectify for any potential waveform error. Better still, we may envisage the analog to digital conversion of the output potential actually present at the RE, which would enable us to plot current versus RE potential instead of the biased current versus CE potential presented here. These improvements require re-iterating the electrical hardware design. They do not qualify as an evolvability-related change but are strictly related to performance limitations. We therefore set aside this concern and evaluated how these initial performances could be leveraged in real biosensing applications.

Figure 5.3.3: Comparison of CV results acquired on our platform versus on an AutoLab PGSTAT302N potentiostat

5.3.3 Application: electrochemical detection of dopamine

Dopamine (DA) is a widely known neurotransmitter molecule. Involved in a large variety of neurological disorders such as Parkinson's disease or schizophrenia, DA has also been demonstrated to have a role in learning, memory and attention span [244]. DA is the precursor molecule for the synthesis of noradrenaline and adrenaline. As such it is also implicated in mechanisms of stress response and is tested for in blood plasma together with other catecholamines to assess adrenal gland function. Although applications for direct-to-consumer POC testing are not straightforward here, DA is an interesting candidate for validating the usability and performances of our evolved platform.

DA is an electro-active compound, oxidating as follows:

Figure 5.3.4: Oxidation of dopamine

DA has ben widely investigated by electrochemical analysis [244]. The scope of this research work does not include the optimization of dopamine sensors, we therefore determined the LOD achievable using non-functionalized Screen-Printed Gold Electrodes (SPGE).

Method We proceeded to the electrochemical quantitation of dopamine (DA) solutions obtained from dopamine hydrochloride (Sigma-Aldrich, MO, USA). A stock solution at $0.1\,\mathrm{mol\,L^{-1}}$ and successive dilutions were prepared to obtain DA concentrations down to $5.9 \times 10^{-9}\,\mathrm{mol\,L^{-1}}$. We used distilled water (Millipore Milli-Q, Bedford, MA, USA) as a solvent.

Amperometry was carried out by setting the electrochemical cell at $E = 0.6\,\mathrm{V}$ for 4 seconds and then applying a potential $E = -0.3\,\mathrm{V}$ for 20 seconds during which the current decay was measured at 100 samples per seconds. Quantitation (HLT1 in figure 5.3.1) was obtained from the integration of the current signal (i.e. charge) over the entire acquisition time. CV was performed using a scan rate of $E = 100\,\mathrm{mV\,s^{-1}}$ for potentials ranging from $E = -1\,\mathrm{V}$ to $E = 1\,\mathrm{V}$. HLT1 was designed to detect the reduction current peak between -0.1 and 0.1 V. SWCV was configured with the same beginning and end potentials, with increments of $E_{\mathrm{incr}} = 5\,\mathrm{mV}$, pulse amplitudes of 80 mV at a frequency of $E = 20\,\mathrm{Hz}$. DPV was similarly executed from $E = 0.6\,\mathrm{V}$ to $E = -0.2\,\mathrm{V}$ with pulses duration of 0.04 s, an amplitude of $E = 50\,\mathrm{mV}$ and potential steps duration of 0.1 s. HLT1 for both SWCV and DPV implemented a differential reduction peak current detection algorithm. The local maximum differential current was probed for between $[-0.1 - 0.1]$ V for SWCV and between $[0 - 0.2]$ V for DPV. Current sensitivities (i.e. reciprocal of the current amplifier gain) were specified programmatically so as to maximize the signal-to-noise ratio for a given DA dilution. Each calibration point was obtained by averaging results from three measurements.

Once the calibration methods were available, we carried out the fully automated quantitation of two DA solutions with nominal concentrations of $4 \times 10^{-3}\,\mathrm{mol\,L^{-1}}$ and $40 \times 10^{-6}\,\mathrm{mol\,L^{-1}}$ respectively. Measurements were performed five times for each solution. Gain/sensitivity was initially set to its maximum value (Gain $= 3.6 \times 10^{6}\,\mathrm{V\,A^{-1}}$). HLT2 replaced HLT1 for this automated quantitation. HLT2 was implemented with a current-saturation detection algorithm in order to determine whether the acquisition should be repeated with a lower input current-gain. Current-saturation was probed for within the specified potential range for which we searched for the local maxima/minima. Should saturation be detected then the LLT was repeated at the directly inferior gain. This scheme was repeated iteratively until no saturation occured or once the lowest sensitivity was reached ($1 \times 10^{-4}\,\mathrm{A\,V^{-1}}$).

Results An illustration of the acquisition data retrieved at the mobile-software layer for each method is given in figure 5.3.5. We may notice on the voltammograms obtained from the CV acquisitions in figure 5.3.5a, that the oxidation of DA engenders significantly high currents in the high-potential range ($E > 0.5\,\mathrm{V}$), bringing the TIA to saturation. In the absence of an automatic gain selection module at the HWA accessory level, we are limited to evaluation of measurement data post-acquisition. For our particular application, this characteristic is advantageous since it allows us not to consider current saturation if it does not occur for potentials neighbouring DA's reduction potential $E_r \simeq -0.03\,\mathrm{V}$. We can also observe that the three cycles of CV for each acquisition do not superimpose perfectly: as the cycles are repeated, the oxidation and reduction peak currents sensibly decrease. This behavior has been previously documented and most likely results from the consecutive reactions of DA oxidation products leading to the formation aminochrome and the consequent deterioration of the electrode surface [245]. This signal degradation requires using a new SPGE for each measurement and to derive the analyte concentration from the first or first two CV cycles.

Figures 5.3.5b and 5.3.5c represent the difference between the *forward* and *reverse* currents for SWCV and DPV respectively. Windowed acquisitions as described in section 5.2.3.2 were used in order to retrieve data points in the last decile of both forward and reverse potentials. Oversampling was specified so that the average of 32 acquisition points would define the forward current value, and the average of 32 acquisition points would similarly define the reverse current value.

The amperometric acquisitions are zoomed in in figure 5.3.5d which allows the inspection of the fast reduction current decay once the startup oxidating potential $E = 0.6\,\mathrm{V}$ is

Figure 5.3.5: Acquisition data for the electrochemical detection of DA using CV, SWCV, DPV and amperometry

switched to the reducing potential $E = -0.3\,\mathrm{V}$. The integration of current values is yet performed from $t = 0\,\mathrm{s}$ until $t_{\mathrm{final}} = 20\,\mathrm{s}$

For each of the electrochemical schemes discussed above, and for each DA dilution, a calibration point was retrieved from either identifying a local current maxima/minima (i.e. for CV, SWCV and DPV) or by integrating current over the entire acquisition sequence (i.e. for amperometry). The averaging of three data points per DA concentration enables us to derive the calibration curves given in figure 5.3.6. As we investigated a wide range of DA concentrations, resulting in currents varying over several orders of magnitude, the data is presented in a log-log scale. The linearity of the response with DA concentration is reflected in the results of the log-log regression, presented in table 5.8.

The calibration curve for amperometry (figure 5.3.6d) was obtained by using a segmented log-log regression, each segment corresponding to the input current sensitivity (i.e. TIA gain) used for the acquisition: although the gain and offset errors are relatively low, our amperometric quantitation method relies on the piecewise integration of current values throughout the entire acquisition time. The systematic error therefore accumulates iteratively with each current measurement point. The charge (i.e. current integrated over

Figure 5.3.6: Calibration curves for the electrochemical detection of DA using CV, SWCV, DPV and amperometry

Table 5.8: Calibration of the various electroanalytical schemes for dopamine detection.

		Calibration solution concentration range		R^2	LOD ($\times 10^{-6}$ mol.L^{-1})
		min $\times 10^{-6}$mol.L^{-1}	max $\times 10^{-6}$mol.L^{-1}		
CV		97.6	50×10^3	0.996	0.446
SWCV		48.5	12.5×10^3	0.996	0.611
DPV		12.2	50×10^3	0.988	0.324
Amperometry	@ 1×10^5 V A^{-1}	12.2	0.1953×10^3	0.921	0.347
	@ 1×10^4 V A^{-1}	0.1953	50×10^3	0.977	—

time) is thus significantly affected by the various gain and offset errors for different TIA gains, as illustrated in figure 5.3.6d. A single regression would therefore lead to an obvious mathematical mismatch, whereas a piecewise regression provides very decent results. For all electrochemical methods, our calibration curves span over several orders of magnitude of DA concentrations with regression coefficients up to 0.996. Linear ranges were recovered for concentrations down to 20×10^{-6} mol L^{-1}.

Table 5.9: Accuracy of the automated detection of two DA solution at $4 \times 10^{-3}\,\mathrm{mol\,L^{-1}}$ and $40 \times 10^{-6}\,\mathrm{mol\,L^{-1}}$

		Accuracy (%)			
		Solution at $4 \times 10^{-3}\,\mathrm{mol\,L^{-1}}$		Solution at $40 \times 10^{-6}\,\mathrm{mol\,L^{-1}}$	
CV		97.7	97	42.2	65.3
SWCV		90	91.8	82.5	87.5
DPV		70	82.7	77.6	86
Amperometry	@ $1 \times 10^{5}\,\mathrm{V\,A^{-1}}$	—	—	—	46.7
	@ $1 \times 10^{4}\,\mathrm{V\,A^{-1}}$	—	86.9	—	—

Finally the fully automated quantitation of DA was carried out using each electrochemical scheme for two DA solutions at $4 \times 10^{-3}\,\mathrm{mol\,L^{-1}}$ and $40 \times 10^{-6}\,\mathrm{mol\,L^{-1}}$. with accuracies up to 97.7%. By applying a segmented regression (i.e. one for each current sensitivity) also for CV, SWCV and DPV, accuracies could be improved by up to 12.7% for DA at $4 \times 10^{-3}\,\mathrm{mol\,L^{-1}}$ and up to 23.1% for DA at $40 \times 10^{-6}\,\mathrm{mol\,L^{-1}}$. Although the achieved LODs do not reach levels documented on other sensing systems (e.g. [246, 247, 248]), they were obtained from generic commercial electrochemical sensors. This suggests potential for performance optimization, out of the scope of this research work.

5.3.4 Summary

We presented in this chapter a framework for the design of evolvable LOC/smart-device platforms and illustrated the application of this framework with the design and implementation of an evolvable platform allowing *at initial design time*, the interfacing of SiNW-bioFET sensors. Our approach relies on the intrinsic concepts of PBD for CPS, combined with the systematic consideration of design principles favoring system evolvability. Our research findings suggest key locations where these design principles should materialize. Relevant principles and their locations can be summarized as follows:

At the mobile-software layer, the *non-hierarchical integration* of the HWA and LOC functionality and the *modularity/composability* of *independent* ETs, LLTs, and HLTs, can enable the specification of LOC operation of increasing utility, favoring the expansion of the platform function space, and the expansion of the HWA or LOC architectural or functional space (section 5.2.3.4). The definition of ETs requires special attention: by using the *ideality/simplicity* principle, one can promote an appropriate balance of system complexity vs. system flexibility.

The interface linking the mobile-software layer and the HWA embedded firmware should be designed as a *change-absorber* in an attempt to prevent the propagation of anticipated change through the various platform abstraction layer.

At the HWA level, *redundancy* is often inevitable as it is generally required for the multiplexed interfacing of LOCs embedding numerous biosensors or actuators.

The implementation of these principles in our initial platform allowed us to evolve it so as to accommodate for electrochemical sensing. We could demonstrate the utility of the added functionality for the detection of dopamine using various electroanalytical schemes. Finally, we must insist that the identification of key system performance specifications early in the platform design should initiate a thorough evaluation of system *scalability* with regards to relevant performance metrics. We discuss scalability extensively in the following chapter.

Chapter 6

Evaluating evolvability

6.1 Model-based evaluation of modular scalability

As we discussed in chapter 2, *scalability* is considered by Ross et al. [119] as one of the key factors for providing a complex (e.g cyber-physical) system with the ability to sustain its value in a changing environment. Together with several other non-functional -*ilities* of systems engineering, *scalability* contributes to the *evolvability* of complex systems. From their extensive semantical and ontological research, Ross et al. and De Weck et al. [147, 126] argued that *scalability* could be defined as *the ability of a system to change the current value of one of its specification parameters.* Their definition broadly applies to complex engineering systems. It is yet conceptually closely related to Weinstock and Goodenough's concept of *scalability by extension* specific to computational systems (i.e. software) [249]. In this pioneering work, the authors distinguished two definitions for *scalability* in an attempt to answer semantical questions dating back from the early 90's [250]:

They first consider *scalability* defined as the "ability of a system to handle an increasing workload *without adding resources* to that system". The inclusion of the *workload* concept is obviously very biased to computing applications. Computational workload had been the focus of discussions on scalability before Weinstock and Goodenough published their manuscript and still influenced a succession of research works on *scalability* after that. Duboc et al. developed an extensive framework for the evaluation of *scalability*, according to this first definition for software systems [251]. Although domain-specific, this framework has the merit to tackle the scalability issue with concepts abstract enough to be applicable to complex engineering systems. Duboc et al. acknowledge, just like Weinstock and Goodenough that there is no such thing as "scalable systems" but merely systems scalable *with respect to a given system characteristic as several environment and system design aspects are varied within an expected range.* The authors suggest the evaluation of the impact of the variation of *factors*, within pre-defined *bounds*, on *dependent variables.* A system is judged scalable if factors variations within a given range translate in tolerable changes in the dependent variables. Duboc calls *independent variables* this set of factors. Some of these factors may be *non-scaling*: unchanged while other are *scaling*: varying over an expected range. The *dependent variables* are metrics, often related to performances or resource usage. The evaluation of scalability then consists in a *multi-criteria optimization problem* aiming at identifying what *configuration* of *non-scaling* factors results in the preferred *dependent variables* outcomes given the variations of the *scaling factors.* Since it is multi-criterial, this optimization problem needs to rely on mathematical constructs such as *preference* functions and *utility functions* in order to quantify the set of outcomes for the *dependent variables* with a single metric.

The investigation we present in this section differs from Duboc et al.'s in several aspects. First of all, we present a framework for assessing a *scalability* conceptually much closer to what Weinstock and Goodenough explicit in their *second* definition: *scalability by extension* is the "ability to handle increased workload by repeatedly applying a cost-effective strategy for *extending a system's capability*". This second *scalability* concept thus incorporates the notion of *added system resource* and *the ease with which this addition could be brought to the system* in response to an increasing workload/performance requirement.

We considered that since this concept includes the idea of *adding capacity* to a system *in an iterative manner*, it better fit within the ontology and concepts of *evovability* we have been investigating so far [126, 107, 147]: we are interested in *evaluating the capacity of a system to change the the current value of one of its specification parameters over several generations*. Second of all, our framework aims at identifying system bottlenecks with a *modular perspective* in comparison with the holistic scalability assessment that Duboc proposes. We intend not only to quantify how much a performance or resource *specification* (i.e. Duboc's *dependent variable*) can be improved by scaling up or down factors, but we want to *trace* where these factors originate from and determine how the scaling of a set of factors constricted to independent system *modules* may be beneficial to scalability: we are interested in *modular scalability*: *the ability of a system to scale the current value of one of its specifications by successively reengineering targeted system modules*.

The rationale for this attempt is cost-effectiveness, which as Weinstock and Goodenough suggest, is intrinsically related to how scalable a system should be considered. Given unlimited cost and resource budgets, systems specifications may be infinitely (or largely) scalable. In engineering systems though, costs cannot be overlooked and should even be an integral part of the evaluation of scalability. Our framework does not investigate cost-effectiveness formally, but nevertheless sets the methodological foundation to do so in further research. We must remember that *modularity* favours targeted changes as opposed to system-wide changes, which may translate in significant time, costs and resource savings. Our aim is thus to identify how targeted modular changes, made iteratively over several system generations, can scale a design specification.

We present in this chapter a framework for the evaluation of *modular scalability* [252] (appendix C). Although we clearly are interested in different definitions of *scalability*, our work can be closely related to Duboc's. We discuss the similarities of our approach with theirs throughout the next sections. We then illustrate our methological approach with a case-study: we investigate how the *continuous acquisition bandwidth* of our smartphone-based biosensing instrumentation platform could be scaled up from the successive re-engineering of relevant functionaly and structuraly independent modules. As presented in chapter 4, we mean by bandwidth *the maximum voltage waveform excitation frequency for which the system can operate continuously while streaming data to the mobile-software layer*.

Although Duboc's approach is multi-variable, our methodology addresses a single metric: the *span* of the system specification for which we want to assess the scalability. The investigation of a single-metric is limited in that it does not allow to account for potential inter-dependencies existing between various specifications. These dependencies will very often be of significant interest to the system designer when undertaking scalability analysis. Our single-metric approach yet shows the advantage of avoiding the use of utility functions and multi-variable optimization analyses, to the benefit of methodological clarity. We may foresee that our methodology can easily be expanded to multi-variables analyses by using preference and utility functions, just like Duboc advocates.

System specifications are determined by the structural, functional and/or operational

properties of the system: they are the *dependent variables* that must satisfy performance or technical requirements set for the system. These are specified by the *factors* we defined previously, including *design variables*, over which the system designers can exert control, and uncontrolled factors (e.g. environmental factors). In order to evaluate the *modular scalability* a system specification, we present the 4 steps of our methodology before illustrating its application with a concrete example in the next section. Our approach involves:

1. The identification of all the *factors* influencing the specification of interest. In the context of complex systems, several methods can be used, depending on which phase of the system life-cycle is considered. In the early conceptual design phase, factors including design variables and their associated functions, structures or operations can be identified with the collaborative help of all specialist engineers involved (i.e. mechanics, analog, digital, software, system, etc.). Model-Based Design (MBD) can then be leveraged to associate identified design variables to their realising high-abstraction functions, structures or operations. Further on in the design process, MBD may allow, at a lower level of abstraction this time, to identify relevant design variables that may have been overlooked, e.g. through model-based analysis and simulation. If MBD was not adopted during the conceptual design phase, a reverse-engineering modelling effort can still be undertaken later in the development process or when the system is in operation. The design variables can then be identified with a greater level of refinement, directly related to that of the model itself. System-level MBD, such as for engineering cyber-physical or mechatronic systems, may require formal holistic modelling languages, engulfing different models of computations (e.g. SystemC, SysML). Past this identification step, our methodology necessitates:

2. To *identify and trace the dependencies* existing between all identified factors. There, the system model can be leveraged in different ways: dependencies can be assessed by specialist engineers and *then* informed in the model. If the analytical capabilities of the model allow it, these dependencies can be recovered via model-based simulations and analysis. This identification process is necessary in order to *aggregate design variables and uncontrolled factors that show dependencies with one another.* Aggregates of inter-dependent variables and uncontrolled factors should be defined so as to be independent from one another. The next step of our methodology then requires:

3. To assess the independence of the functions, structures, or operations implementing these aggregates of factors. This step leads to the *identification of the various system modules* (e.g. hardware, software, mixed, architectural, functional, etc.) responsible for the design specification under consideration. This process should be undertaken using a refined model i.e presenting a low level of abstraction: a highly-abstracted model will underline the independence of large sub-systems (figure 6.1.1), whereas lower-abstraction representations (i.e. with finer structural and behavioral details) will more likely reveal finer interactions between modules or components *within* these large subsystems. Independent system modules, i.e. those that do now show dependencies with one another, will be of smaller size (figure 6.1.2). This in turn will enable a minimalistic evaluation of modular scalability: we only intend to re-engineer modular system components affecting the design specification of interest, while leaving unassociated components (i.e. these that would have been comprehended in a higher-abstraction module representation) untouched. The scalability of the design specification determined by these independent modules can be then be quantitatively evaluated and demands:

Figure 6.1.1: Module identification at a high-level of abstraction: the factors influencing the specification of interest have a red overlay. The incremental system evolution by the replacement of the independent modules involved in the specification will lead to reengineering efforts that will extend to the entire system.

Figure 6.1.2: Module identification at a refined level of abstraction: the factors influencing the specification of interest have a red overlay. The incremental system evolution by the replacement of the independent modules involved in the specification will lead to reengineering efforts constricted to modules 1.2, 1.3, 2.3, 3.2

4. To iteratively identify the system module(s) representing the bottlenecks for the system specification under consideration. Once a bottleneck is identified, it should be replaced by a module that is *not limiting* for scaling the design specification. This step is equivalent to successively nulling the influence of each factor aggregate defined at the previous step, and allows the recovery of value for the specification for successive system generations, each one being an evolution of the previous by the replacement of the bottleneck module.

We illustrate how to apply this methodology with the evaluation of the scalability of our platform's continuous acquisition bandwidth.

Figure 6.1.3: Methodology steps for evaluating the modular scalability of a system specification

6.2 Scalability analysis for the continuous acquisition bandwidth specification

As we saw in chapter 4, the capability to monitor current or impedance signals continuous using lock-in amplification presents a significant interest as it may allow advanced signal processing and analysis to be allocated to the mobile-software layer. This in turn, offers benefits such as limiting re-engineering efforts to high-level software when tuning signal analysis for a specific use-case. This would for instance be the case for electrical impedance spectroscopy applications such as cell counting [240], or bacterial detection [253]. The former of these applications is a good illustration of why bandwidth scalability is important here: the detection or non-detection, as well as the further classification of living cells using impedance-based cytometry is dependent on the frequency at which the measurements are carried out. The higher the continuous acquisition bandwidth the more analytical capabilities will be available at the mobile-layer software. Also, the higher the continuous acquisition bandwidth, the higher the maximum cell-throughput, the faster the completion of the cytometry analysis [240].

Our model-based methodology enables scalability assessment relatively early in the conceptual design phase, offering the chance to compare various design alternatives for their ability to scale one of its specifications before a candidate system is selected for implementation. For this case-study though, the modelling effort was carried out post-implementation: first of all, to assess our current system bandwidth and identify the current bottleneck for its specification, and second of all, in order to assess the *potential* for increasing our system bandwidth to allow cell cytometry applications relying on signal processing carried out at the mobile-software layer.

We analytically identified the factors involved in acquisition bandwidth specification. We implemented SysML representations of the system functions, structures and behaviors at a level of abstraction that would allow the definition of parametric relationships between the identified factors.

The structural model of our platform HWA is given in figure 5.2.7 and a functional view of the continuous lock-in acquisition process is given in the activity diagram in figure 5.2.15. We completed this functional representation with a behavioral model of the system with state-machines for all subsystems influencing bandwidth specification. The state-machine of the DSC was presented figure 5.2.13.

Factors influencing system bandwidth were identified by the electrical, embedded software, and mobile-software designer. They were then informed in the various diagrams of the SysML model (e.g. figure 6.2.1) and are summarized in the dependency matrix given in figure 6.2.2 (step 1). The same set of factors, those relevant to system bandwidth, fill both the rows and columns of the matrix. The matrix columns also comprehend dependable software routines. The dependencies are read from row to column. Arrows for a given matrix row represents a dependency of the factor corresponding to the arrow's cell row on the factor in the arrow's cell column. psdStage1ExecutionTime is thus depending on the psdStage1OverheadExecutionTime and avgExecutionTimePerTap factors (darker color matrix cell, including an arrow). These dependencies are relatively obvious in this case: for each PSD DSP routine, the total execution time is dependent on the number of taps of the FIR filter, the processor's clock frequency, the average number of clock cycles per processed tap and the overhead execution time. Similarly, the spi8BitWordTransferIntervalDuration represents the necessary delay between each DSC to BLE SPI transfer to allow for the BLE module to process each word.

As mentioned in section 6.1, these dependencies can be used to define *aggregates* of factors showing interdependencies and their associated system functions, structures or operations (step 2). If an arrow displays a dependency between two factors in a ma-

Figure 6.2.1: Dependencies of the DSP variables on DSC Core 1 (represented by the dashed arrows): the total execution time of the PSD routine depends on the FIR filter number of taps, on the intrinsic DSC clocking frequency, the sampling frequency, the decimation factor and the overhead execution time of the routines

trix cell, then the corresponding row and column factors are aggregated together. These aggregates correspond in figure 6.2.2 to the groups of factors with overlays of the same color or, in the case of independent variables, to single rows with no overlay. The psdStage1OverheadExecutionTime and avgExecutionTimePerTap variables, both influencing psdStage1ExecutionTime are presented in the matrix rows with the same yellow-overlay that is covering their dependent variable: psdStage1ExecutionTime.

The next step of our methodology (step 3) demanded us to identify independent system modules (functional, structural or operational) *specifying or encompassing* the independent factors aggregates from the previous step: the green-overlay aggregate could thus be attributed to the ADC sampling interrupt subroutines module; the red-overlay aggregate to the PSD stage 1 activity (figure 5.2.15) and its target processor module (figure 5.2.7) (i.e. a PSD stage1 hardware-software module) ; the blue-overlay aggregate to the PSD stage 2 hardware-software module; the yellow-overlay aggregate to the BLE-DSC communication module (i.e. software-only). The independent dmaTransferISRExecutionTime variable is solely associated with the acquisitionBufferFullDMAInt software interrupt subroutine as shown by the single dependency arrow in its matrix row.

To make greater sense of the influence of opting for BLE technology on system bandwidth, we aggregated the attributeNotificationUpdateInterval together with the factors associated with the BLE-DSC data transfers (yellow-aggregate factors). The latter variable is a property of the BLE protocol specification. This meant considering an entire independent BLE module (i.e. hardware-software-communication protocol).

The final step of our methodology consists in successively replacing these independent

Figure 6.2.2: Dependency matrix presenting dependencies between all the factors influencing the specification of interest: the system bandwidth. The same set of factors fill both the rows and columns of the matrix. The dependencies are read from row to column: arrows for a given matrix row represents a dependency *of* the factor corresponding to the arrow's cell row *on* the factor in the arrow's cell column. factors presenting a dependency relation are aggregated together. These aggregates are represented with overlays of the same color

modules by ones that are *not* bandwidth limiting (step 4). This process was achieved using a state-machine-based fault-detection simulation model and is explicited further in the next section.

6.2.1 Model-based estimation of continuous acquisition bandwidth

We carried out the last step of our methodology by relying on a refined state-machine model enabling fault-detection. Part of this model (that of the DSC) is presented in figure 6.2.3. Faulty states were specified so as to be able to identify the independent module responsible for bandwidth limitation for a given trial excitation frequency. For instance, the trigger of the StartDSPStage2 event while the CLA hardware accelerator is still processing samples (i.e. is in the DSP Stage 2 state) is a faulty operation that would result in the input data for PSD Stage 2 being wrongly overwritten. Such event trigger is thus redirected in the model to the bwLimitedByPSDStage2Duration faulty state (one the red states in figure 6.2.3), pinpointing PSD Stage 2 as the responsible module for bandwidth limitation, since we asserted in the previous steps of the methodology that the factors associated to bandwidth specification in DSP Stage 2 were independent from the other modules. Executing a simulation for a specific voltage waveform excitation frequency *above* our system bandwidth would thus bring the state-machine to a given faulty-state, whereas a simulation at an excitation frequency *below* the bandwidth would run continuously, recovering the impedance measurement in the mobile software layer model independently

Figure 6.2.3: The value of the successive system's generations bandwidth specification is retrieved by simulation of the system's state machines and the recovery of faulty executions. Entry to any of the predefined faulty states (red states in the figure) translates a fault in execution and effectively stops the simulation. More specifically, simulations at various AC excitation frequencies are carried out by dichotomy: If a simulation is faulty, then the excitation frequency is set as an upper bound for the minimum system bandwidth. If the simulation is successful, it is set as a lower bound for the maximum system bandwidth. Simulations are repeated until an excitation frequency within the specified tolerance interval outputs a successful simulation.

of the acquisition duration. We considered that the simulation were fault-free if the model could successfully fill at least two acquisition buffers while transmitting and recovering data in the mobile-software layer model.

In order to recover the bandwidth of our initial system via simulation, we derived an iterative dichotomous algorithm (figure 6.2.4) made responsible for attempting to simulate the state-machine model at excitation frequencies redefined iteratively in the following maner: The initial condition for running this algorithm was to set the initial excitation frequency to an arbitrary value, in this case 100 kHz. Should this first simulation be successful, this frequency would be set as the lower bound for the minimum system bandwidth at the next iteration where the trial frequency would be doubled. If instead the simulation ended in a faulty state, then the excitation frequency would be set as an upper bound for maximum bandwidth and the next trial frequency would be halved. By repeating this scheme iteratively, the algorithm converges towards the system's bandwidth value within a given tolerance. Given the possibility of a large number of iterations and related computational load, we opted for replicating our state-machine model in MathWorks Simulink StateFlow and implemented our algorithm in Matlab.

Figure 6.2.4: Principle of the iterative dichotomous algorithm for the estimation of system bandwidth. An initial condition sets the original excitation frequency to an arbitrary value, in this case 100 kHz. The state-machine simulation ends in faulty state A. 100 kHz is thus set as an upper bound for the maximum system bandwidth and the next trial frequency is halved to 50 kHz. At this frequency the simulation runs successfuly setting 50 kHz as the lower bound for the minimum system bandwidth. By repeating this scheme iteratively, the algorithm converges towards the system's bandwidth value: it ends with a successful run at frequency 68.75 kHz with a lower bound at 62.5 kHz and upper bound of 75 kHz. The difference between which is 12.5 kHz, inferior to the maximum uncertainty tolerance set for the bandwidth estimate.

6.2.2 System scalability over generations

Our initial system i.e. *Gen1* on figure 6.2.5 successfully passed all simulations at excitation frequencies below 3.1624 Hz. The first faulty-state in which the model would end up beyond this value was that associated with a trigger event received for initiating the PSD routines while DSC Core1 was still transferring data to the BLE module. As the simulation enters in the faulty-state it is aborted, thereby pinpointing the BLE module to be the bottleneck for *Gen1*'s bandwidth. We determined in step 3 and 4 that three factors influencing system bandwidth have been confined to the BLE system communication module, namely spi8BitWordTransferIntervalDuration, attributeNotificationUpdateInterval, and bleSPIWordProcessingTime. The first of these variables is dependent on the last

Figure 6.2.5: Successive systems are evolved so that the bottleneck module identified of a given generation is replaced by a non-limiting module in the next generation.

(BLE-DSC communication software dependencies). attributeNotificationUpdateInterval does not show dependencies with the other two but also influences the system bandwidth. It is a property of the BLE protocol specification.

We were able to shunt the influence of the BLE module on bandwidth by setting these three factors to infinitesimals. This, in turn, helped us identify which system module would be the bottleneck for bandwidth in *Gen2*: the evolved system corresponding to *Gen1* but *replacing the BLE module and its relevant factors* by non-limiting implementations. The execution of our iterative dichotomous algorithm now converged towards 27.498 Hz, this time limited in bandwidth by the PSD Stage2 hardware-software module. By using the same reasoning we could investigate which modules would be successively limiting system bandwidth, each time circumventing the bottleneck of the previous system generation. The corresponding successive systems generations could thus be determined and consist of:

- *Generation 1*: Original system such as described throughout chapter 4.
- *Generation 2*: System evolved so that the BLE communication module (i.e. the yellow-overlay factor aggregate + the attributeNotificationUpdateInterval variables) is not limiting to system bandwidth: the duration factors were set to infinitesimals.
- *Generation 3*: Generation 2 evolved so that PSD stage 2 is not limiting to bandwidth (i.e. the blue-overlay factors aggregate): stage2PSDExecutionTime set to an infinitesimal.
- *Generation 4*: Generation 3 evolved so that PSD stage 1 is not limiting to bandwidth (i.e. the red-overlay factors aggregate): stage1PSDExecutionTime set to an infinitesimal.

– *Generation 5*: Generation 4 evolved so that the ADC sample and hold duration is not limiting (i.e. the green-overlay factors aggregate): adcSamplingISRExecutionTime set to an infinitesimal

– *Generation 6*: Generation 5 evolved so that the independent dmaTransferIsrExecutionTime variable is not limiting: set to an infinitesimal.

6.3 Discussion

Our methodology is valuable in that it can not only help identifying system bottlenecks, but it also provides a means to evaluate the focus and scope of the re-engineering effort needed to improve the system specification under consideration to an upgraded *known* level. The analysis of dependencies helps certify that the benefit of scaling a specification by changing a system module will not necessitate the re-engineering of other modules dependent on the former in any way. The validity of our approach of course necessitates a thorough dependency evaluation methodology. A review of the state-of-the-art for the assessment of system dependencies is out of the scope of this research work.

Our strategy of iterative system evolution, circumventing bottleneck system modules with non-limiting implementations enables to predict which unchanged module will become the new specification bottleneck in the evolved system. Our methodology differs substantially from Duboc's approach in this regards, which advocates the estimation of *bounds* for the scaling factors, and deducing how the specification of interest will scale accordingly. Importantly, the quantitative results of our methodology cannot be trusted for the system's last generation as they will reflect how the non-limiting conditions were set for the design variables involved in the specification under scrutiny. This is illustrated in the case study: the bandwidth specification given for *Gen6* is a function of the infinitesimals set for all relevant design variables (e.g. processing times, buffer sizes, etc.) in previous system generations.

Although our strategy is valuable for bottleneck identification, one can also consider not relying on non-limiting conditions but rather to replace a module with another presenting known design properties/variables, or follow more closely Duboc's approach with a modular system perspective. This approach will provide a more concrete assessment of scalability should a given implementation be considered for re-engineering and will help certifying whether the newly implemented module will truly benefit the system specification or whether another module will limit overall system performances, effectively wasting the re-engineering effort. The accurate definition of a replacement module may yet require modifying the model so as to accommodate for additional or alternative factors, potentially establishing new parametric relationships with the other system modules, new dependencies, or affecting other specifications. A strong case could be made for using this approach for our case-study: we could have opted for swapping the BLE module and its associated factors with a another wireless communication module (incl. hardware, software, etc.) specified in the model with a fine level of details. The adoption of a technology such as Bluetooth High Speed or Wifi would have translated in an increase in bandwidth which, at best, would have resulted in the bandwidth specification exhibited by *Gen2*. Since impedance based cytometry requires a bandwidth specification close or above what *Gen6* can provide, we may conclude that the modular re-engineering approach of identified bottlenecks in system generations 1 to 6 will not suffice and that greater architectural changes, and corresponding re-engineering efforts need be undertaken.

Finally, we must discuss the absence of the decimation factor as a factor influencing bandwidth: since sample decimation allows the reduction of system throughput, we could have included the decimation factor in the factors. The scalability law is straightforward:

doubling the decimation factor means that the voltage excitation frequency can also be doubled. This apparent scaling yet relates to another important specification of the system: the *signal* bandwidth (not to be confused with the system data throughput/bandwidth we have been discussing so far). We therefore excluded decimation from our analysis, which comes down to specifying a decimation factor of 1.

6.4 Summary

We presented in this chapter a 4-step methodology, enabling the evaluation the *modular scalability* of complex systems by leveraging model-based design and analysis. We defined *modular scalability* as *the ability of a system to scale the current value of one of its specifications by successively reengineering targeted system modules.* Our methodology involves the successive identification of factors, the assessment of their inter-dependencies, and the determination of the independent system modules engulfing these factors. Following these steps the system designer will be able to consider how the successive re-engineering of each of the identified modules will scale the design specification of interest. Our systems engineering approach enables scalability assessment to be carried out for complex (e.g. cyber-physical) systems where specifications are likely to arise from factors spanning over various engineering disciplines and where system modules may comprehend anything from a mechanical add-on to a software toolbox. It can be adapted to multi-variable analyses through the use of utility functions and multi-variable optimization.

Chapter 7

Discussion

7.1 On the integrability of biosensors

At an abstract level, LOCs can be considered as grids of sensors and actuators interconnected by microfluidic channels through which information flows within a biological sample. Under this perspective, next-generation POC system designers may be able to leverage engineering design principles for changeability from emerging fields such as sensor networks or smart grids. In order to do so, they yet will need to overcome one of the main challenges pertaining to LOC-based system design: *the lack of integrability of most of biosensing technologies*. Surely the *substrates* on which biological signal transductions take place are increasingly integrated with communication nodes (e.g. SiNW embedded on a SoC, possibly comprehending I2C or SPI communication modules, MEA technologies, etc.) but biosensors generally suffer from poor integrability if the biological elements of the system are also taken into consideration. State-of-art in this regards is probably best represented by LOC cartridges such as that developed by McDevitt and colleagues [41]. These cartridges often embed reagents under blisters or freeze-dried. These reagents are released (or eluted) and motivated to a sensing area until finally information is retrieved through external instrumentation. The size (certainly not at the microscale), the limited stability of the biological species or interface layers in harsh environments, and the related reproducibility and quality issues of integrated biosensing devices have so far prevented their standardization and availability as OTS components.

This lack of integrability justifies in itself the use of comprehensive SE design frameworks, tools and methodologies for engineering (evolvable) systems. The SiNW-bioFET technology on which we extensively elaborated on in chapter 3 provides a good example. As we mentioned, the possibilities for SiNW-bioFET gate surface chemistries are numerous, which could ideally translate in integrated and modular SiNW-bioFET-based components each targeting a given biomarker, and interfaced to via a standard inter-circuit communication technology. In such scenario, "conventional " CPS design methodologies, including PBD, would be simplified to a great extent to rather well known *platform design-space export* and *platform mapping* steps and their optimization techniques. Unfortunately, the limitations of current modelling formalisms for depicting accurately the mechanisms for interacting with sensors and actuators (section 3.2.1.1 [162]) take a whole new dimension with biosensing technologies. System designer cannot at this stage avoid hybrid-modelling or multi-tool and multiphysics modelling, with the inter-tool compatibility challenges this entails and the complications and delays associated with the integration of modules developed in isolation. The search for *where* and *how* to build-in evolvability in next-generation POC IVMD systems will likely still expand from the biological sample microfluidic inlet of the LOC to the high-abstraction cloud-computing layer, a rather vast design space.

7.2 Model-based platform design, hybrid-systems and evolvability

7.2.1 Towards a unifying design framework for IVMD systems

We stipulated in section 3.2.2 that PBD merely offers a possible entry point for the design of evolvable systems. It appears particularly appropriate for high-throughput, multiplexed IVMD system requiring a high degree of *parallelism/redundancy*. McDevitt and colleagues' arguments on the adequacy of PBD for next-generation POC systems [41] seem to support our approach. Although we must acknowledge once again that we omitted formal *uncertainty recognition*, and the *valuation* and *cost-effectiveness* assessment of evolvability for this particular design and implementation work, we still believe the possibilities and modalities of PBD should be considered early in the design process of any next-generation POC IVMD system.

We must yet acknowledge the missing elements that to this day prevent the complete model-based PBD of hybrid-systems comprehending microfluidics and biosensing elements. We for instance cannot at this stage pursue *design synthesis* since our *platform design-space export* and *platform mapping* are not automated but manually hard-coded. The implementation of these processes will require to address challenges such as the definition of optimization mapping techniques targeting LOC-specific requirements e.g. optimized sample processing volumes, best LOD, minimum time-to-result etc. Optimization processes towards such performance requirements will probably reveal quite challenging, considering the non-triviality of optimizing power comsumption or computational time when PBD is used for conventional CPS design. In order to reach that point though, a number of intermediary steps need to be completed, such as the development of a unifying and overarching semantics, a key research topic of the CPS community [172].

We may yet argue that despite the lack of a unifying framework, lots can be done. We for instance could model mobile software, embedded firmware and the parametric relationships representing the physical laws governing the behavior of SiNW-bioFETs using SysML. We could then complete our investigations by modelling and analyzing the time-based behavior of our sensor and instrumentation in Simulink. Although neither SysML nor Simulink allowed us to fully model our hybrid-system, interfaces between varied modelling tools enabled us to comprehensively assess system behavior and performances.

7.2.2 Platform architecting for evolvability

The ontology we discussed in chapter 2 enables us to draw partial conclusions on the *potential* of our prototype system to demonstrate evolvability. The *non-hierarchical integration* of the platform functionality at the mobile-software layer provides significant freedom of action as to how to post-process the LOC acquisition data in higher-level abstraction layers: in mobile-software or potentially in the cloud. This property is valuable in a context where *-omics* sequencing technologies are increasingly coupled to cloud supercomputing analytics and AI [254, 255]. Furthermore, we must not forget that POC IVMD systems are *user-centric* systems. The smart-device is pivotal in handling user interaction and in articulating the information from chip to cloud. Similarly, the composition of LOC programs intertwining *modular* low- and high-level tasks offers significant LOC programmability advantages: high-level tasks return values can be used to modify ensuing LOC program instructions at run time. LOC program conditional branching could thus rely on the evaluation of user-queried or cloud-computing return properties.

This non-hierarchical design yet puts a certain number of constraints on the system requirements, notably on the mobile-software/embedded firmware wireless interface. As we mentioned earlier, we strived to design the iOS/embedded firmware protocol as a

change *absorber* to the expansion of the software functional space, or to the expansion of the HWA or LOC architectural or functional space (section 5.2.3.4). That interface itself must be robust enough to maintain LOC operation even when the smart-device (and often its user) is remotely located from the LOC. This in turns may imply the wireless transmission of mobile-software commands to the HWA via wireless networks and reciprocally the transmission of acquisition data from the HWA to the mobile-software layer over these networks. This scenario entails security and privacy concerns, as well as response-time considerations should any of the LOC processes be time-critical. We deliberately set these issues aside during our investigation but we acknowledge that they are of paramount importance.

Our search for *modularity* enabled us to implement electrochemical capabilities for our platform by the sole functional composition of existing *elementary tasks* at the mobile-software layer and did not require change to any other system components. The composability of ETs is essential for enabling the incorporation of new physical functionality on the platform. It is yet not easily implemented. Furthermore, the definition of ETs in consideration of the *ideality/simplicity* can itself represent a challenge [108]: ETs allowing a fine level of control may translate into a burdening complexity (e.g. mobile-software control of every HWA CPU register) whereas higher abstraction levels will limit functionality and therefore flexibility. Once again, the complexity/flexibility tradeoff here can be subject to a refined uncertainty and cost-effectiveness analysis.

Redundancy is probably the most obvious of the design principles favoring evolvability. Although our current platform allows the interfacing of up to 12 current-readout channels, many more could have been included with a hardware design inspired from MEA systems. One of the arguments against a high level of redundancy/parallelism are the obvious costs of duplicating hardware structures and design controllers capable of running in parallel or in a fast multiplexed manner. In our specific case, duplicating the analog front end current-amplifiers alone would at a certain point have required shifting from OTS to technologies offering higher levels of integration, such Application Specific Integrated Circuits (ASIC), a much costlier design choice.

7.2.3 Modular scalability

We saw in chapter 5 to which extent our system can *scale* with regards to continuous acquisition bandwidth, just as we could identify the relevant modules involved in bandwidth specification. Although our modularity assessment did not include change propagation analysis, it constitutes a potent tool to relate structures to functions and performances. Our investigation of *modular scalability*, as we defined it, can be further completed by a valuation process (e.g. using utility and cost functions) and be taken into account in the refined valuation of evolvability. Our endeavor addressed some of the shortcomings associated with the assessment of change-related *-ilities* in the engineering of complex-systems. It acknowledges that engineering *change* has become predominant over design from blank-canvas when dealing with complex systems, and that system-wide tools and methodologies for easing the evaluation (and valuation) of change-related *-ilities* are still needed.

7.3 Is it worth it?

The vast panel of considerations addressed throughout this thesis certainly attests of the intricacy of designing evolvable systems and raises again the fundamental question: Are *potential* benefits of design-for-evolvability worth the troubles? The simple assessment of Fricke and Schultz's criteria, i.e. dynamic marketplace, technological innovation, variety

of environments [108], can merely be used as a rule of thumb for initiating the more formal *uncertainty recognition* advocated by Cardin et al. [143] and the *valuation* of evolvability in various design alternative as recommended by Suh et al. [163]. These methods can only help the decision-making process by ensuring that the probability of cost-effectiveness of design-for-evolvability is favorable. Since we did not formally go through these steps, we cannot claim that our design methodology or our POC platform implementation (chapter 2, 3, 4) resulted in a cost-effective evolvable system. Our main objective was not so much to ensure cost-effectiveness than to propose a framework including design principles and some implementation mechanisms that would allow to build-in evolvability in smartphone/LOC systems.

From a more general perspective, the arguments we put forward for justifying design for evolvability for smartphone/LOC POC systems may themselves be questioned. As we mentioned earlier, the consideration of non-functional *-ilities* and specifically *change-related -ilities* are often examplified for large engineering systems such as in the fields of aerospace or defence. The "size" of POC IVMD systems is thus in no way comparable to that of the "super-systems" Ramos et al. refer to [118]. Yet the *trends* for which we presented tangible evidence: *increase in complexity, fast-pace innovation, dynamic marketplace, variety of environments* are becoming more apparent and more prominent and will soon erase any doubt that evolvability is needed in order to ensure the long-lasting success of direct-to-consumer IVMD systems.

Among the indications that system complexity will increase more sharply than it has so far, we may cite the efforts to bring mass spectrometry methods to the POC, with the related hardware and software integration complexity this entails [256, 257, 258]. System complexity is also likely to increase with new emergent applications requiring a high degree of interconnectivity between individual POC IVMD systems. This could soon materialize in the form of nation-wide, or worldwide databases and associated predictive and prescriptive systems. Such system-of-systems (SoS) has already been envisioned [24]: it would be determinant, first for improving our knowledge-base of general human-health and answering the remaining questions pertaining to our biology such as the association or origins of specific phenotypes with particular *-omics* profiles. Subsequently, this interconnected SoS could be used to leverage the comprehensive general knowledge-base *and* identified similarities between individuals (e.g. common gene variants, transcription profiles, etc.) to the benefit of the individual. The scope and complexity of such application may seem confined to high-abstraction functionality layers i.e. software, data analytics, cloud computing, AI. It may yet not be the case, and extensive cloud-to-LOC connectivity could make a lot of sense in some scenarios (e.g automated reprogramming and execution of LOC operation based on population-wide risk assessment, etc.).

The plausibility of reaching such level of system *complexity* is reinforced by the likelihood for the *fast-pace technological innovation* animating the field of IVMD technologies to keep accelerating. Since the main focus of this thesis has been put on the recovery of diagnostic or predictive information from biological samples, little has been said on the advances in synthetic biology and their potential in healthcare, including at the point of need. The convergence of synthetic chemistry and biology with sophisticated automation surely holds promises for the future of home-based IVMD, which may soon expand to unconventional applications.

At this point one may only venture ideas and opinions as to the feasibility of revolutionary applications combining *in-situ*, on-chip, chemical or biological synthesis and interventional/experimental procedures on one's own biological specimens (e.g circulating DNA, RNA, induced pluripotent stem cells, etc.). An example of such applications could involve personalized, closed-loop processes for home-based automated drug screening: synthetic compounds could be tested on a panel of *-omics*, in order to evaluate the

best course of action should the *user* be infected by a given pathogen. This panel of *-omics* could proxy a variety of different tissues. It would be generated from the individual's own induced pluripotent stem cells (iPS). These iPS could be harvested, cultured and reprogrammed on the same LOC where candidate drugs/gene vectors would be synthesized and tested for. Such screening could be carried preemptively, before the detection of symptomatic events, solely based on a the identification of environmental risks from population-wide data (e.g. the sudden increase in the amount of a given pathogen in the local population). Although such scenario appears futuristic, and certainly does not account for the regulatory, ethical, political and operational challenges it would entail, it is not unrealistic in consideration of currently available technologies and of the exponential rate of progress in synthetic biology and computing [125, 259, 260, 261].

The realization of these trends would thus transform design for evolvability from strategic choice to essential requirement. Answering the question "is it worth it?" should thus keep on becoming easier.

Chapter 8

Conclusion

Smart-device/LOC systems for POC IVMD applications represent one of the key vectors for the decentralization and remodelling of our healthcare systems. They carry the potential to rebalance the daunting dissymmetry in the availability of medical information existing between medical practitioners and non-medical professionals, including patients. Smart-device/LOC systems are today essential in palliating to the lack of resources and medical infrastructures in low- and middle-income countries. They are just as much pivotal to the emergence of a preventive and personalized medicine by enabling the convergence of a variety of health related information, including of multi-*omics* data, anywhere, and at anytime [1, 24].

Substantial operational, regulatory, ethical, economical and social barriers remain to be overcome before the growing potential of POC technologies is fully exploited. From a socio-technico-economical standpoint, two of the main interrelated challenges are probably: 1 – the increasing system complexity and low frequency-of-use of POC systems, synonymous of poor added-value to the end-user and low return-on-investment to the buyer ; and 2 – the high risk of rapid system obsolescence in consideration of the accelerating pace of innovation that is driving IVMD technologies.

System evolvability can contribute to solving these particular issues by limiting the re-engineering costs and efforts associated with the successive incremental necessary evolutions of a system. It can ideally support the adoption and long-lasting success of next-generation instrumented POC IVMD systems. Our investigation of the design frameworks promoting evolvability enlightened us as to the potential offered by PBD as formal entry point to the design of evolvable systems, under the condition it is undertaken within the realm of the uncertainties and dynamism associated with the POC IVMD system context.

Within the framework of evolvable platform design, our principal objectives were to define *where* and *how* to incorporate change-enablers, i.e. *means* change-related *-ilities* and change-absorbers into instrumented smart-device/LOC system architectures in order to promote evolvability. Our conceptual study findings suggest, in particular, the need to carefully define the initial function space of the platform, making it available at the mobile-software layer, and incorporating *modularity* and *composability* mechanisms in order to enable its smooth expansion. The incorporation of change-absorption mechanisms, is conversely critical for the main smart-device/HWA and HWA/LOC interfaces. It should prevent the propagation or multiplication of anticipated changes from one layer to the next, and constrict reengineering efforts to targeted modules. By following these principles, our design and implementation efforts resulted in a platform prototype capable of evolving smoothly from interfacing LOCs embedding SiNW-bioFETs to one accommodating for electrochemical biosensors.

Although the lack of a unifying MBD framework is still hindering the emergence of design automation and synthesis for CPS including biological elements, system-level de-

sign tools and languages such as SysML can still offer valuable assistance for the design and assessment of evolvability in engineering systems. Our investigation of the tools and metrics for assessing evolvability confronted us to their limitations for relating the architectural features of a system to its performance specifications. The model-based method we developed for the evaluation of modular scalability contributes to coping with these insufficiencies.

The research findings presented throughout this thesis constitute elements of an extensive framework enabling design for evolvability in engineering systems, including for consumer-based IVMD solutions. Our conceptual and methodological work has highlighted many of the issues to be addressed and given hints of a possible solution through evolvable PBD. Our case-study allowed us to suggest possible implementations of some of the discussed change-enabling/absorbing mechanisms.

Following our efforts, significant research is yet still needed in order formalize a more comprehensive SE framework for the design of evolvable systems, extending throughout and *across* the life-cycle of several systems generations. This framework should assist the evaluation and incorporation of change-enabling/change-absorbing mechanisms promoting evolvability early in the conceptual design phase, and target considerations specific to CPS. Our approach has missed this level of formalism and has merely resulted in a model and implementation for which we could *partially* determine the evolvability *a posteriori*. Finally we must once again acknowledge the lack of cost-effectiveness assessment in our investigation and the consideration of a single metric in the methodology we proposed for evaluating modular scalability. Among the interesting research questions that arise from our investigation we may thus wonder: how to guide the design of a system towards evolvability with limited a priori knowledge on the system's internal working? how to formalize such a framework for CPS systems, including for those comprehending biological and microfluidic constituents? how to draw a more comprehensive portrait of evolvability through relevant metrics relating system structures, functions, behaviors and performances?

This thesis provides a comprehensive portrait of the technological challenges and opportunities laying ahead of the decentralization of POC IVMD systems. We presented system evolvability as an increasingly valuable *-ility* to consider for the design of tomorrow's complex LOC-based systems, in what could be the *epoch of exponential technologies* [125]. Although engineers will need to further address where and how to embed evolvability in engineering systems, answering the question of whether they should will become obvious. The adoption of proper design frameworks and methodology for promoting evolvability may then be determinant for the fate engineering systems and for their societal impact on the generations to come.

Bibliography

[1] E. Topol, *The Patient Will See You Now: The Future of Medicine is in Your Hands.* Basic Books, 2015. [Online]. Available: http://books.google.dk/books?id=Kr2DBQAAQBAJ

[2] M. Prince, A. Wimo, M. Guerchet, A. Gemma-Claire, Y.-T. Wu, and M. Prina, "World Alzheimer Report 2015: The Global Impact of Dementia - An analysis of prevalence, incidence, cost And trends," p. 84, 2015.

[3] C. W. Yancy, M. Jessup, B. Bozkurt, J. Butler, D. E. Casey, M. H. Drazner, G. C. Fonarow, S. A. Geraci, T. Horwich, J. L. Januzzi, M. R. Johnson, E. K. Kasper, W. C. Levy, F. A. Masoudi, P. E. McBride, J. J. V. McMurray, J. E. Mitchell, P. N. Peterson, B. Riegel, F. Sam, L. W. Stevenson, W. H. W. Tang, E. J. Tsai, and B. L. Wilkoff, "2013 ACCF/AHA guideline for the management of heart failure: A report of the american college of cardiology foundation/american heart association task force on practice guidelines," *Circulation*, vol. 128, no. 16, 2013.

[4] J. Bryce, C. Boschi-Pinto, K. Shibuya, and R. E. Black, "WHO estimates of the causes of death in children." *Lancet*, vol. 365, no. 9465, pp. 1147–1152, 2005.

[5] G. . M. Collaborators and C. of Death, "Global, regional and national levels of age-specific mortality and 240 causes of death, 1990-2013 : A systematic analysis for the Global Burden of Disease Study 2013," *The Lancet*, vol. 385, no. 9963, pp. 1990–2013, 2015. [Online]. Available: http://dx.doi.org/10.1016/S0140-6736(14)61682-2

[6] A. Mills, "Health Care Systems in Low- and Middle-Income Countries," *New England Journal of Medicine*, vol. 370, no. 6, pp. 552–557, 2014. [Online]. Available: http://www.nejm.org/doi/abs/10.1056/NEJMra1110897

[7] E. Comission, "Directive 98/79/EC of the european parliament and of the council of 27 october 1998 on in vitro diagnostic medical devices," *Official Journal of the European Communities*, no. L331, p. 7.12, 1998. [Online]. Available: http://scholar.google.com/scholar?q=related:aXn63vgYeKAJ:scholar.google.com/{\&}hl=en{\&}num=30{\&}as{_}sdt=0,5$\backslash$npapers2:// publication/uuid/97E7B316-AC2E-431E-A1D5-393FA7C84C65

[8] R. Parloff, "This CEO's out for blood - Fortune." [Online]. Available: http://fortune.com/2014/06/12/theranos-blood-holmes/

[9] E. K. Clifford Goodman, Eric Faulkner, Ciara Gould, A. Smith, C. Aguiar, C. Nelson, A. Grover, A. Berlin, R. Phillips, A. Horan, and The_Lewin_Group, "The Value of Diagnostics Innovation, Adoption and Diffusion into Health Care," Tech. Rep. July, 2005. [Online]. Available: http://earlydetect.com/Graph/TheValueofDiagnostics0.pdf

[10] "Investing in the Future of Medicine: A Investor's Guide to the In Vitro Diagnostics Market - Forbes," 2016. [Online]. Available: http://www.forbes.com/sites/zackmiller/2014/02/12/investing-in-the-future-of-medicine-a-investors-guide-to-the-in-vitro-diagnostics-market/{\#}45d4a8cd5feb

[11] L. J. Kricka, T. G. Polsky, J. Y. Park, and P. Fortina, "The future of laboratory medicine - A 2014 perspective," *Clinica Chimica Acta*, vol. 438, pp. 284–303, 2015. [Online]. Available: http://dx.doi.org/10.1016/j.cca.2014.09.005

[12] I. H. G. S. Consortium, "Initial sequencing and analysis of the human genome," *Nature*, vol. 409, no. 6822, pp. 860–921, feb 2001. [Online]. Available: http://dx.doi.org/10.1038/35057062http://www.nature.com/nature/journal/v409/n6822/suppinfo/409860a0{_}S1.html

[13] E. Check Hayden, "The $1,000 genome," *Nature*, pp. 4–5, 2014.

[14] F. Collins and H. Varmus, "A New Initiative on Precision Medicine," *New England Journal of Medicine*, vol. 372, no. 9, pp. 2012–2014, 2015.

[15] E. J. Topol, "Individualized medicine from prewomb to tomb," *Cell*, vol. 157, no. 1, pp. 241–253, 2014. [Online]. Available: http://dx.doi.org/10.1016/j.cell.2014.02.012

[16] K. M. Boycott, M. R. Vanstone, D. E. Bulman, and A. E. MacKenzie, "Rare-disease genetics in the era of next-generation sequencing: discovery to translation," *Nat Rev Genet*, vol. 14, no. 10, pp. 681–691, oct 2013. [Online]. Available: http://dx.doi.org/10.1038/nrg3555http://10.1038/nrg3555http://www.nature.com/nrg/journal/v14/n10/abs/nrg3555.html{\#}supplementary-information

[17] M. H. McLean, D. Dieguez, L. M. Miller, and H. a. Young, "Does the microbiota play a role in the pathogenesis of autoimmune diseases?" *Gut*, vol. 64, no. 2, pp. 332–341, 2015. [Online]. Available: http://www.ncbi.nlm.nih.gov/pubmed/25416067

[18] P. A. Smith, "The tantalizing links between gut microbes and the brain," *Nature*, vol. 526, no. 7573, pp. 312–314, 2015.

[19] M. I. Lefterova, C. J. Suarez, N. Banaei, and B. A. Pinsky, "Next-Generation Sequencing for Infectious Disease Diagnosis and Management: A Report of the Association for Molecular Pathology." *The Journal of molecular diagnostics : JMD*, vol. 17, no. 6, pp. 623–34, 2015. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S1525157815001725

[20] A. Szpechcinski, J. Chorostowska-Wynimko, R. Struniawski, W. Kupis, P. Rudzinski, R. Langfort, E. Puscinska, P. Bielen, P. Sliwinski, and T. Orlowski, "Cell-free DNA levels in plasma of patients with non-small-cell lung cancer and inflammatory lung disease," *Br J Cancer*, vol. 113, no. 3, pp. 476–483, jul 2015. [Online]. Available: http://dx.doi.org/10.1038/bjc.2015.225http://10.1038/bjc.2015.225

[21] K. R. Chi, "The Tumour Trail Left In Blood," *Nature*, vol. 532, pp. 569–571, 2016.

[22] N. Crosetto, M. Bienko, and A. van Oudenaarden, "Spatially resolved transcriptomics and beyond," *Nature Reviews Genetics*, vol. 16, no. 1, pp. 57–66, 2015. [Online]. Available: http://www.nature.com/nrg/journal/v16/n1/full/nrg3832.html$\backslash$nhttp://www.nature.com/nrg/journal/v16/n1/pdf/nrg3832.pdf

[23] S. J. Bielinski, J. E. Olson, J. Pathak, R. M. Weinshilboum, L. Wang, K. J. Lyke, E. Ryu, P. V. Targonski, M. D. Van Norstrand, M. A. Hathcock, P. Y. Takahashi, J. B. McCormick, K. J. Johnson, K. J. Maschke, C. R. Rohrer Vitek, M. S. Ellingson, E. D. Wieben, G. Farrugia, J. A. Morrisette, K. J. Kruckeberg, J. K. Bruflat, L. M. Peterson, J. H. Blommel, J. M. Skierka, M. J. Ferber, J. L. Black, L. M. Baudhuin, E. W. Klee, J. L. Ross, T. L. Veldhuizen, C. G. Schultz, P. J. Caraballo, R. R. Freimuth, C. G. Chute, and I. J. Kullo, "Preemptive genotyping for personalized medicine: Design of the right drug, right dose, right timedusing genomic data to individualize treatment protocol," *Mayo Clinic Proceedings*, vol. 89, no. 1, pp. 25–33, 2014.

[24] E. J. Topol, *The Creative Destruction of Medicine: How the Digital Revolution Will Create Better Health Care*. Basic Books, 2012. [Online]. Available: http://books.google.dk/books?id=5Q-O9vnNqPkC

[25] B. S. Ferguson, D. A. Hoggarth, D. Maliniak, K. Ploense, R. J. White, N. Woodward, K. Hsieh, A. J. Bonham, M. Eisenstein, T. E. Kippin, K. W. Plaxco, and H. T. Soh, "Real-Time, Aptamer-Based Tracking of Circulating Therapeutic Agents in Living Animals," *Science Translational Medicine*, vol. 5, no. 213, pp. 213ra165—-213ra165, 2013. [Online]. Available: http://stm.sciencemag.org/content/5/213/213ra165

[26] T. T. Ruckh and H. a. Clark, "Implantable nanosensors: Toward continuous physiologic monitoring," *Analytical Chemistry*, vol. 86, no. 3, pp. 1314–1323, 2014.

[27] G. Poste, "Bring on the biomarkers." *Nature*, vol. 469, no. 7329, pp. 156–157, 2011.

[28] W. Koh, W. Pan, C. Gawad, H. C. Fan, G. a. Kerchner, T. Wyss-Coray, Y. J. Blumenfeld, Y. Y. El-Sayed, and S. R. Quake, "Noninvasive in vivo monitoring of tissue-specific global gene expression in humans." *Proceedings of the National Academy of Sciences of the United States of America*, vol. 111, no. 20, pp. 7361–6, 2014. [Online]. Available: http://www.ncbi.nlm.nih.gov/pubmed/24799715

[29] "New Queen Elizabeth Hospital Birmingham Opens State-Of-The-Art Blood Sciences Laboratory." [Online]. Available: http://www.laboratorynetwork.com/doc/new-queen-elizabeth-hospital-birmingham-opens-state-of-the-art-blood-sciences-laboratory-0001

[30] E. C. Hayden, "No Out of regulatory limbo, 23andMe resumes some health tests and hopes to offer more," oct 2015.

[31] "Theranos - Wired." [Online]. Available: http://www.wired.com/2016/02/theranos-is-hiring-a-writer-to-solve-its-problems/

[32] "Theranos' Lab Problems Go Way Deeper Than Its Secret Tech — WIRED." [Online]. Available: http://www.wired.com/2016/04/theranos-lab-problems-go-way-deeper-secret-tech/

[33] G. J. Kost, "Guidelines for point-of-care testing: Improving patient outcomes," 1995.

[34] S. K. Vashist, P. B. Luppa, L. Y. Yeo, A. Ozcan, and J. H. T. Luong, "Emerging Technologies for Next-Generation Point-of-Care Testing," *Trends in Biotechnology*, vol. 33, no. 11, pp. 692–705, 2015. [Online]. Available: http://dx.doi.org/10.1016/j.tibtech.2015.09.001

[35] "Point-Of-Care Diagnostic Market worth $27.5 Billion by 2018." [Online]. Available: http://www.marketsandmarkets.com/PressReleases/point-of-care-diagnostic.asp

[36] T. Hänscheid, M. Rebelo, and M. P. Grobusch, "Point-of-care tests: Where is the point?" *The Lancet Infectious Diseases*, vol. 14, no. 10, p. 922, 2014. [Online]. Available: http://dx.doi.org/10.1016/S1473-3099(14)70914-4

[37] C. F. Weber, K. Gorlinger, D. Meininger, E. Herrmann, T. Bingold, a. Moritz, L. H. Cohn, and K. Zacharowski, "Point-of-care testing: a prospective, randomized clinical trial of efficacy in coagulopathic cardiac surgery patients," *Anesthesiology*, vol. 117, no. 3, pp. 531–547, 2012. [Online]. Available: http://www.ncbi.nlm.nih.gov/pubmed/22914710

[38] K. R. King, L. P. Grazette, D. N. Paltoo, J. T. McDevitt, S. K. Sia, P. M. Barrett, F. S. Apple, P. A. Gurbel, R. Weissleder, H. Leeds, E. J. Iturriaga, A. K. Rao, B. Adhikari, P. Desvigne-Nickens, Z. S. Galis, and P. Libby, "Point-of-Care Technologies for Precision Cardiovascular Care and Clinical Research," *JACC: Basic to Translational Science*, vol. 1, no. 1-2, pp. 73–86, 2016. [Online]. Available: http://linkinghub.elsevier.com/retrieve/pii/S2452302X16000139

[39] C. M. Lowndes, E. Sherrard-Smith, C. Dangerfield, Y. H. Choi, N. Green, M. Jit, R. D. Marshall, C. Mercer, E. Harding-Esch, A. Nardone, R. Howell-Jones, J. Bason, O. A. Johnson, C. P. Price, C. A. Gaydos, S. T. Sadiq, and P. J. White, "Point-of-care testing versus standard practice for chlamydia: a new approach to assessing the public health effect of rapid testing and treatment at local level," *The Lancet*, vol. 384, p. S47, 2014. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0140673614621735

[40] "Roche cobas 6800/8800." [Online]. Available: https://www.cobas68008800.com/

[41] M. P. McRae, G. W. Simmons, J. Wong, B. Shadfan, S. Gopalkrishnan, N. Christodoulides, and J. T. McDevitt, "Programmable bio-nano-chip system: a flexible point-of-care platform for bioscience and clinical measurements," *Lab Chip*, vol. 15, no. 20, pp. 4020–4031, 2015. [Online]. Available: http://xlink.rsc.org/?DOI=C5LC00636H

[42] "Alere Determine HIV-1/2 Ag/Ab." [Online]. Available: http://www.alere.com/en/home/product-details/determine-1-2-ag-ab-combo.html

[43] I. V. Jani, D. Ph, and T. F. Peter, "How Point-of-Care Testing Could Drive Innovation in Global Health," *The New England journal of medicine*, vol. 368, no. 24, pp. 2319–2324, 2013.

[44] R. Steven and M. Steinhubl, "Can Mobile Health Technologies Transform Health Care?" *JAMA : the journal of the American Medical Association*, vol. 92037, no. 1, pp. 1–2, 2013. [Online]. Available: http://list.umassmed.edu/read/attachment/136612/1/Steinhubl{_}JAMA{_}MHTTransformHealthCare{_}2013.pdf

[45] J. G. Peter, L. S. Zijenah, D. Chanda, P. Clowes, M. Lesosky, P. Gina, N. Mehta, and G. Calligaro, "Effect on mortality of point-of-care, urine-based lipoarabinomannan testing to guide tuberculosis treatment initiation in HIV-positive hospital inpatients: a pragmatic, parallel-group, multicountry, open-label, randomised controlled trial," *The Lancet*, vol. 387, no. 10024, pp. 1187–1197, 2016. [Online]. Available: http://dx.doi.org/10.1016/S0140-6736(15)01092-2

[46] World Health Organization. Global Malaria Programme, "Scaling up diagnostic testing , treatment and surveillance for malaria," *Who*, pp. 1–12, 2012.

[47] I. V. Jani, N. E. Sitoe, E. R. Alfai, P. L. Chongo, J. I. Quevedo, B. M. Rocha, J. D. Lehe, and T. F. Peter, "Effect of point-of-care CD4 cell count tests on retention of patients and rates of antiretroviral therapy initiation in primary health clinics: An observational cohort study," *The Lancet*, vol. 378, no. 9802, pp. 1572–1579, 2011.

[48] R. W. Peeling and A. Ronald, "Diagnostic challenges of sexually transmitted infections in resource-limited settings." *Future microbiology*, vol. 4, no. 10, pp. 1271–1282, 2009.

[49] "Low-tech microfluidics — Science Practice." [Online]. Available: http://www.science-practice.com/blog/2015/01/29/low-tech-microfluidics/

[50] "Engineering material." [Online]. Available: http://www.materialsforengineering.co.uk/engineering-materials-features/what-role-will-materials-play-in-making-microfluidics-a-success/58365/

[51] "Continuous flow microfluidics." [Online]. Available: https://www.asme.org/engineering-topics/articles/bioengineering/using-microfluidics-to-diagnose-hiv

[52] "Stephen Quake's Group @ Stanford University." [Online]. Available: http://thebigone.stanford.edu/

[53] "Wheeler Microfluidics Laboratory: Research." [Online]. Available: http://microfluidics.utoronto.ca/research{_}dmfchannel.php

[54] "Droplet Microfluidics." [Online]. Available: http://www3.imperial.ac.uk/portal/page/portallive/polymersandmicrofluidics

[55] L. Gervais, N. De Rooij, and E. Delamarche, "Microfluidic chips for point-of-care immunodiagnostics," *Advanced Materials*, vol. 23, no. 24, pp. 151–176, 2011.

[56] C. D. Chin, T. Laksanasopin, Y. K. Cheung, D. Steinmiller, V. Linder, H. Parsa, J. Wang, H. Moore, R. Rouse, G. Umviligihozo, E. Karita, L. Mwambarangwe, S. L. Braunstein, J. van de Wijgert, R. Sahabo, J. E. Justman, W. El-Sadr, and S. K. Sia, "Microfluidics-based diagnostics of infectious diseases in the developing world." *Nature medicine*, vol. 17, no. 8, pp. 1015–1019, 2011. [Online]. Available: http://dx.doi.org/10.1038/nm.2408

[57] K. F. Lei, "Microfluidic systems for diagnostic applications: a review." *Journal of laboratory automation*, vol. 17, no. 5, pp. 330–47, 2012. [Online]. Available: http://www.ncbi.nlm.nih.gov/pubmed/22893635

[58] H. Zhu, S. O. Isikman, O. Mudanyali, A. Greenbaum, and A. Ozcan, "Optical imaging techniques for point-of-care diagnostics," *Lab on a Chip*, pp. 51–67, 2012.

[59] J. Wang, "Electrochemical biosensors: Towards point-of-care cancer diagnostics," *Biosensors and Bioelectronics*, vol. 21, no. 10, pp. 1887–1892, 2006.

[60] C. Wyatt Shields IV, C. D. Reyes, and G. P. López, "Microfluidic cell sorting: a review of the advances in the separation of cells from debulking to rare cell isolation," *Lab Chip*, vol. 15, pp. 1230–1249, 2015. [Online]. Available: http://xlink.rsc.org/?DOI=C4LC01246A

[61] Y. Chen, P. Li, P.-H. Huang, Y. Xie, J. D. Mai, L. Wang, N.-T. Nguyen, and T. J. Huang, "Rare cell isolation and analysis in microfluidics." *Lab on a chip*, vol. 14, no. 4, pp. 626–45, 2014. [Online]. Available: http://www.ncbi.nlm.nih.gov/pubmed/24406985

[62] S. Lindström and H. Andersson-Svahn, "Overview of single-cell analyses: microdevices and applications." *Lab on a chip*, vol. 10, no. 24, pp. 3363–3372, 2010.

[63] Y. Zhao, D. Chen, H. Yue, J. B. French, J. Rufo, S. J. Benkovic, and T. J. Huang, "Lab-on-a-chip technologies for single-molecule studies," *Lab on a Chip*, vol. 13, no. 12, p. 2183, 2013. [Online]. Available: http://xlink.rsc.org/?DOI=c3lc90042h

[64] I. E. Araci and P. Brisk, "Recent developments in microfluidic large scale integration," *Current Opinion in Biotechnology*, vol. 25, pp. 60–68, 2014. [Online]. Available: http://dx.doi.org/10.1016/j.copbio.2013.08.014

[65] S.-Y. Teh, R. Lin, L.-H. Hung, and A. P. Lee, "Droplet microfluidics." *Lab on a chip*, vol. 8, no. 2, pp. 198–220, 2008.

[66] M. J. Jebrail, M. S. Bartsch, and K. D. Patel, "Digital microfluidics: a versatile tool for applications in chemistry, biology and medicine," *Lab on a Chip*, vol. 12, no. 14, p. 2452, 2012. [Online]. Available: http://xlink.rsc.org/?DOI=c2lc40318h

[67] I. E. Araci and S. R. Quake, "Microfluidic very large scale integration (mVLSI) with integrated micromechanical valves," *Lab on a Chip*, vol. 12, no. 16, p. 2803, 2012.

[68] N. S. G. K. Devaraju and M. a. Unger, "Pressure driven digital logic in PDMS based microfluidic devices fabricated by multilayer soft lithography," *Lab on a Chip*, vol. 12, no. 22, p. 4809, 2012.

[69] J. a. Weaver, J. Melin, D. Stark, S. R. Quake, and M. a. Horowitz, "Static control logic for microfluidic devices using pressure-gain valves," *Nature Physics*, vol. 6, no. 3, pp. 218–223, 2010. [Online]. Available: http://dx.doi.org/10.1038/nphys1513

[70] M. Rhee and M. a. Burns, "Microfluidic pneumatic logic circuits and digital pneumatic microprocessors for integrated microfluidic systems." *Lab on a chip*, vol. 9, no. 21, pp. 3131–3143, 2009.

[71] P. Pop, M. Alistar, E. Stuart, and J. Madsen, *Fault-Tolerant Digital Microfuidic Biochips*. Springer US, 2015.

[72] A. Rival, D. Jary, C. Delattre, Y. Fouillet, G. Castellan, A. Bellemin-Comte, and X. Gidrol, "EWOD-based microfluidic chip for single-cell isolation, mRNA purification and subsequent multiplex qPCR," *Lab on a Chip*, vol. 14, pp. 3739–3749, 2014. [Online]. Available: http://pubs.rsc.org/en/Content/ArticleLanding/2014/LC/C4LC00592A

[73] D. Witters, K. Knez, F. Ceyssens, R. Puers, and J. Lammertyn, "Digital microfluidics-enabled single-molecule detection by printing and sealing single magnetic beads in femtoliter droplets," *Lab on a Chip*, vol. 13, no. 11, p. 2047, 2013. [Online]. Available: http://xlink.rsc.org/?DOI=c3lc50119a

[74] J. V. Jokerst and J. T. Mcdevitt, "Programmable nano-bio-chips: multifunctional clinical tools for use at the point-of-care," *Nanomedicine*, vol. 5, no. 1, pp. 143–155, 2010.

[75] D. Erickson, D. O'Dell, L. Jiang, V. Oncescu, A. Gumus, S. Lee, M. Mancuso, and S. Mehta, "Smartphone technology can be transformative to the deployment of lab-on-chip diagnostics." *Lab on a chip*, pp. 3159–3164, 2014. [Online]. Available: http://www.ncbi.nlm.nih.gov/pubmed/24700127

[76] "Lab-on-a-chip said to perform 1,024 simultaneous tests — EE Times." [Online]. Available: http://www.eetimes.com/document.asp?doc{_}id=1171478

[77] "Smartphones set to become the fastest spreading technology in human history — ExtremeTech." [Online]. Available: http://www.extremetech.com/computing/129058-smartphones-set-to-become-the-fastest-spreading-technology-in-human-history

[78] A. Nemiroski, D. C. Christodouleas, J. W. Hennek, A. a. Kumar, E. J. Maxwell, M. T. Fernández-abedul, and G. M. Whitesides, "Universal mobile electrochemical detector designed for use in resource-limited applications," *Proceedings of the National Academy of Sciences*, vol. 111, no. 33, pp. 1–6, 2014.

[79] S. R. Steinhubl, E. D. Muse, and E. J. Topol, "The emerging field of mobile health," *Science Translational Medicine*, vol. 7, no. 283, pp. 283rv3–283rv3, 2015. [Online]. Available: http://stm.sciencemag.org/cgi/doi/10.1126/scitranslmed.aaa3487

[80] A. C. Brewer, D. C. Endly, J. Henley, M. Amir, B. P. Sampson, J. F. Moreau, and R. P. Dellavalle, "Mobile Applications in Dermatology," *JAMA Dermatology*, vol. 149, no. 11, p. 1300, 2013. [Online]. Available: http://archderm.jamanetwork.com/article.aspx?doi=10.1001/jamadermatol.2013.5517

[81] L. Dayer, S. Heldenbrand, P. Anderson, and P. O. G. B. C. Martin, "Smartphone medication adherence apps: Potential benefits to patients and providers," *J Am Pharm Assoc*, vol. 53, no. 2, pp. 172–181, 2013.

[82] K. Yang, H. Peretz-Soroka, Y. Liu, and F. Lin, "Novel developments in mobile sensing based on the integration of microfluidic devices and smartphones," *Lab Chip*, vol. 16, pp. 943–958, 2016. [Online]. Available: http://xlink.rsc.org/?DOI=C5LC01524C

[83] H. Im, C. M. Castro, H. Shao, M. Liong, J. Song, D. Pathania, L. Fexon, C. Min, M. Avila-Wallace, O. Zurkiya, J. Rho, B. Magaoay, R. H. Tambouret, M. Pivovarov, R. Weissleder, and H. Lee, "Digital diffraction analysis enables low-cost molecular diagnostics on a smartphone," *Proceedings of the National Academy of Sciences*, vol. 112, no. 18, p. 201501815, 2015. [Online]. Available: http://www.pnas.org/lookup/doi/10.1073/pnas.1501815112

[84] J. C. Contreras-naranjo, Q. Wei, and A. Ozcan, "Mobile Phone-Based Microscopy , Sensing , and Diagnostics," *IEEE Journal of selected topics in quantum electronics*, vol. 22, no. 3, 2016.

[85] V. Oncescu, D. O'Dell, and D. Erickson, "Smartphone based health accessory for colorimetric detection of biomarkers in sweat and saliva." *Lab on a chip*, vol. 13, no. 16, pp. 3232–8, 2013. [Online]. Available: http://www.ncbi.nlm.nih.gov/pubmed/23784453

[86] S. Sumriddetchkajorn, K. Chaitavon, and Y. Intaravanne, "Mobile-platform based colorimeter for monitoring chlorine concentration in water," *Sensors and Actuators, B: Chemical*, vol. 191, pp. 561–566, 2014. [Online]. Available: http://dx.doi.org/10.1016/j.snb.2013.10.024

[87] S. K. Vashist, E. Marion Schneider, R. Zengerle, F. von Stetten, and J. H. Luong, "Graphene-based rapid and highly-sensitive immunoassay for C-reactive protein using a smartphone-based colorimetric reader," *Biosensors and Bioelectronics*, vol. 66, pp. 169–176, 2015. [Online]. Available: http://linkinghub.elsevier.com/retrieve/pii/S0956566314009063

[88] K. Su, Q. Zou, J. Zhou, L. Zou, H. Li, T. Wang, N. Hu, and P. Wang, "High-sensitive and high-efficient biochemical analysis method using a bionic electronic eye in combination with a smartphone-based colorimetric reader system," *Sensors and Actuators B: Chemical*, vol. 216, pp. 134–140, 2015. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0925400515005043

[89] A. I. Barbosa, P. Gehlot, K. Sidapra, A. D. Edwards, and N. M. Reis, "Portable smartphone quantitation of prostate specific antigen (PSA) in a fluoropolymer microfluidic device," *Biosensors and Bioelectronics*, vol. 70, pp. 5–14, 2015. [Online]. Available: http://linkinghub.elsevier.com/retrieve/pii/S0956566315001499

[90] H. C. Koydemir, Z. Gorocs, D. Tseng, B. Cortazar, S. Feng, R. Y. L. Chan, J. Burbano, E. McLeod, and A. Ozcan, "Rapid imaging, detection and quantification of Giardia lamblia cysts using mobile-phone based fluorescent microscopy and machine learning," *Lab Chip*, vol. 15, no. 5, pp. 1284–1293, 2015. [Online]. Available: http://xlink.rsc.org/?DOI=C4LC01358A

[91] A. M. Nicolini, C. F. Fronczek, and J.-Y. Yoon, "Droplet-based immunoassay on a sticky nanofibrous surface for multiplexed and dual detection of bacteria using smartphones," *Biosensors and Bioelectronics*, vol. 67, pp. 560–569, 2015. [Online]. Available: http://linkinghub.elsevier.com/retrieve/pii/S0956566314007271

[92] H. Zhu, I. Sencan, J. Wong, S. Dimitrov, D. Tseng, K. Nagashima, and A. Ozcan, "Cost-effective and rapid blood analysis on a cell-phone." *Lab on a chip*, vol. 13, no. 7, pp. 1282–8, 2013. [Online]. Available: http://www.pubmedcentral.nih.gov/articlerender.fcgi?artid=3594636{\&}tool=pmcentrez{\&}rendertype=abstract

[93] A. García, M. M. Erenas, E. D. Marinetto, C. a. Abad, I. De Orbe-Paya, A. J. Palma, and L. F. Capitán-Vallvey, "Mobile phone platform as portable chemical analyzer," *Sensors and Actuators, B: Chemical*, vol. 156, no. 1, pp. 350–359, 2011.

[94] E. H. Doeven, G. J. Barbante, A. J. Harsant, P. S. Donnelly, T. U. Connell, C. F. Hogan, and P. S. Francis, "Mobile phone-based electrochemiluminescence sensing exploiting the 'USB On-The-Go' protocol," *Sensors and Actuators B: Chemical*, vol. 216, pp. 608–613, 2015. [Online]. Available: http://linkinghub.elsevier.com/retrieve/pii/S0925400515005390

[95] E. Lebiga, R. Edwin Fernandez, and A. Beskok, "Confined chemiluminescence detection of nanomolar levels of H ₂ O ₂ in a paper–plastic disposable microfluidic device using a smartphone," *The Analyst*, 2015. [Online]. Available: http://xlink.rsc.org/?DOI=C5AN00720H

[96] M. Zangheri, L. Cevenini, L. Anfossi, C. Baggiani, P. Simoni, F. Di Nardo, and A. Roda, "A simple and compact smartphone accessory for quantitative chemiluminescence-based lateral flow immunoassay for salivary cortisol detection," *Biosensors and Bioelectronics*, vol. 64, pp. 63–68, 2015. [Online]. Available: http://linkinghub.elsevier.com/retrieve/pii/S0956566314006368

[97] B. Berg, B. Cortazar, D. Tseng, H. Ozkan, S. Feng, Q. Wei, R. Y. L. Chan, J. Burbano, Q. Farooqui, M. Lewinski, D. Di Carlo, O. B. Garner, and A. Ozcan, "Cellphone-Based Hand-Held Microplate Reader for Point-of-Care Testing of Enzyme-Linked Immunosorbent Assays," *ACS Nano*, vol. 9, no. 8, pp. 7857–7866, 2015.

[98] T. Laksanasopin, T. W. Guo, S. Nayak, A. A. Sridhara, S. Xie, O. O. Olowookere, P. Cadinu, F. Meng, N. H. Chee, J. Kim, C. D. Chin, E. Munyazesa, P. Mugwaneza, A. J. Rai, V. Mugisha, A. R. Castro, D. Steinmiller, V. Linder, J. E. Justman, S. Nsanzimana, and S. K. Sia, "A smartphone dongle for diagnosis of infectious diseases at the point of care," *Journal of translational medicine*, vol. 7, no. 273, 2015.

[99] J. Jiang, X. Wang, R. Chao, Y. Ren, C. Hu, Z. Xu, and G. L. Liu, "Smartphone based portable bacteria pre-concentrating microfluidic sensor and impedance sensing system," *Sensors and Actuators, B: Chemical*, vol. 193, no. 2014, pp. 653–659, 2014. [Online]. Available: http://dx.doi.org/10.1016/j.snb.2013.11.103

[100] B. Li, L. Li, A. Guan, Q. Dong, K. Ruan, R. Hu, and Z. Li, "A Smartphone Controlled Handheld Microfluidic Liquid Handling System," *Lab on a Chip*, vol. 14, pp. 4085–4092, 2014. [Online]. Available: http://pubs.rsc.org/en/Content/ArticleLanding/2014/LC/C4LC00227J

[101] "Mobile Health (mHealth) Technologies and Global Markets." [Online]. Available: http://www.bccresearch.com/market-research/healthcare/mobile-health-hlc162a.html

[102] "QuantuMDx — The Future of Diagnostics." [Online]. Available: http://www.quantumdx.com/

[103] "Biomeme, Inc." [Online]. Available: http://www.biomeme.com/

[104] S. Kumar, W. J. Nilsen, A. Abernethy, A. Atienza, K. Patrick, M. Pavel, W. T. Riley, A. Shar, D. Spruijt-metz, D. Hedeker, V. Honavar, C. Quinn, V. Shusterman, and D. Swendeman, "Mobile Health Technology Evaluation," *Am J Prev Med*, vol. 45, no. 2, pp. 228–236, 2013.

[105] "Apple - ResearchKit." [Online]. Available: https://www.apple.com/researchkit/

[106] F. Patou, M. Dimaki, W. E. Svendsen, K. Kjaegaard, and J. Madsen, "A Smart Mobile Lab-on-Chip-Based Medical Diagnostics System Architecture Designed for Evolvability," *2015 Euromicro Conference on Digital System Design*, pp. 390–398, 2015. [Online]. Available: http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=7302301

[107] J. Beesemyer, "Empirically characterizing evolvability and changeability in engineering systems," MSc thesis., Massachusetts Institute of Technology, 2012. [Online]. Available: http://dspace.mit.edu/handle/1721.1/76092

[108] E. Fricke and A. P. Schulz, "Design for changeability (DfC): Principles to enable changes in systems throughout their entire lifecycle," *Systems Engineering*, vol. 8, no. 4, pp. 342–359, 2005.

[109] P. D. Borches, "On the Origin of Evolvable Systems," pp. 1–12, 2008.

[110] INCOSE, *Systems engineering handbook*, C. Haskins, Ed., 2006. [Online]. Available: http://smslab.kaist.ac.kr/Course/CC532/2012/LectureNote/2012/INCOSESystemsEngineeringHandbookv3.15-Sep-2007.pdf

[111] A. Kossiakoff, W. N. Sweet, S. J. Seymour, and S. M. Biemer, *Systems Engineering Principles and Practice*, 2011, vol. 102, no. 3. [Online]. Available: http://www.ncbi.nlm.nih.gov/pubmed/20576311$\backslash$nhttp://books.google.com/books?id=MRZoj0yAm9oC{\&}pgis=1

[112] T. a. W. Jarratt, C. M. Eckert, N. H. M. Caldwell, and P. J. Clarkson, "Engineering change: An overview and perspective on the literature," *Research in Engineering Design*, vol. 22, no. 2, pp. 103–124, 2011.

[113] N. P. Suh, *The principles of design*, 1990. [Online]. Available: http://scholar.google.com/scholar?hl=en{\&}btnG=Search{\&}q=intitle:The+Principles+of+Design{\#}0

[114] S. D. Eppinger, D. E. Whitney, R. P. Smith, and D. A. Gebala, "A model-based method for organizing tasks in product development," *Research in Engineering Design*, vol. 6, no. 1, pp. 1–13, 1994.

[115] J. Johnson, *Networks in Action: Communication, Economics and Human Knowledge*. Berlin, Heidelberg: Springer Berlin Heidelberg, 1995, ch. The Multid, pp. 49–79. [Online]. Available: http://dx.doi.org/10.1007/978-3-642-57843-4{_}3

[116] M. E. Sosa, S. D. Eppinger, and C. M. Rowles, "A Network Approach to Define Modularity of Components in Complex Products," *Journal of Mechanical Design*, vol. 129, no. 11, p. 1118, 2007.

[117] O. L. De Weck, D. Roos, and C. L. Magee, *Engineering Systems: Meeting Human Needs in a Complex Technological World*, ser. Engineering systems. MIT Press, 2011. [Online]. Available: https://books.google.dk/books?id=lGaGjE2i0S0C

[118] A. L. Ramos, J. V. Ferreira, and J. Barceló, "Model-based systems engineering: An emerging approach for modern systems," *IEEE Transactions on Systems, Man and Cybernetics Part C: Applications and Reviews*, vol. 42, no. 1, pp. 101–111, 2012.

[119] A. M. Ross and D. H. Rhodes, "Architecting systems for value robustness: Research motivations and progress," *2008 IEEE International Systems Conference Proceedings, SysCon 2008*, pp. 216–223, 2008.

[120] B. Boehm and W. J. Hansen, "Spiral Development : Experience , Principles , and Refinements," Special Report, Carnegie Mellon Software Engineering Institute, Tech. Rep. July, 2000.

[121] M. W. Maier and E. Rechtin, *The Art of Systems Architecting, Third Edition*, ser. Systems Engineering, C. Press, Ed. CRC Press, 2009. [Online]. Available: https://books.google.dk/books?id=AWLMBQAAQBAJ

[122] M. Emes, P. A. Bryant, M. K. Wilkinson, P. King, A. M. James, and S. Arnold, "Interpreting "Systems Architecting"," *Systems Engineering*, vol. 15, no. 4, pp. 305–326, 2012. [Online]. Available: http://ezproxy.lib.ucf.edu/login?url=http://search.proquest.com/docview/926281441?accountid=10003$\backslash$nhttp://sfx.fcla.edu/ucf?url{_}ver=Z39.88-2004{\&}rft{_}val{_}fmt=info:ofi/fmt:kev:mtx:journal{\&}genre=article{\&}sid=ProQ:ProQ:civilengineering{\&}atitle=Model+Based+Systems

[123] H. Zimmermann, "OSI Reference Model - The ISO Model of Architecture for Open Systems Interconnection," pp. 425–432, 1980.

[124] P. M. Shames and M. A. Sarrel, "A modeling pattern for layered system interfaces," in *25th Annual INCOSE International Symposium*, 2015.

[125] R. Kurzweil, *The Singularity Is Near*, 2005. [Online]. Available: http://books.google.com/books?id=9FtnppNpsT4C{\&}printsec=frontcover{\&}dq=intitle:The+Singularity+Is+Near{\&}hl={\&}cd=1{\&}source=gbs{_}api$\backslash$npapers2://publication/uuid/A9C61AE5-996F-4801-854E-15236451BC0C

[126] O. L. De Weck, A. M. Ross, and D. H. Rhodes, "Investigating Relationships and Semantic Sets amongst System Lifecycle Properties (Ilities)," *Third International Engineering Systems Symposium CESUN 2012, Delft University of Technology, 18-20 June 2012*, no. June, pp. 18–20, 2012.

[127] D. O. B. Fulcoly, "a Normative Approach To Designing for Evolvability: Methods and Metrics for Considering Evolvability in Systems Engineering," MSc thesis., 2012.

[128] H. McManus and D. Hastings, "A framework for understanding uncertainty and its mitigation and exploitation in complex systems," *IEEE Engineering Management Review*, vol. 34, no. 3, pp. 81–94, 2006.

[129] M. L. Butterfield, J. S. Pearlman, and S. C. Vickroy, "A system-of-systems engineering GEOSS: Architectural approach," *IEEE Systems Journal*, vol. 2, no. 3, pp. 321–332, 2008.

[130] J. A. Christian and J. R. Olds, "A Quantitative Methodology for Identifying Evolvable Space Systems," *Aiaa*, no. February, pp. 1–11, 2005. [Online]. Available: http://smartech.gatech.edu/bitstream/handle/1853/8375/AIAA{_}2005{_}2543JChristian.pdf?sequence=1

[131] R. Steiner, "System Architectures and Evolvability: Definitions and Perspective," *Systems Engineering*, p. 5, 1998.

[132] K. Hölttä-Otto, *Modular Product Platform Design*, 2005, vol. 18, no. 1. [Online]. Available: http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.98.739{\&}rep=rep1{\&}type=pdf

[133] U. Lindemann, R. Kleedörfer, and M. Gerst, "The Development Department and Engineering Change Management," in *Designers*, 1998, pp. 169–182. [Online]. Available: http://link.springer.com/10.1007/978-1-4471-1268-6{_}16

[134] D. V. Steward, "Design Structure System: A Method for Managing the Design of Complex Systems," *IEEE Transactions on Engineering Management*, vol. EM-28, no. 3, pp. 71–74, 1981. [Online]. Available: http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=6448589

[135] R. Keller, T. Eger, C. Eckert, and P. J. Clarkson, "Visualising change propagation," pp. 1–12, 2005. [Online]. Available: http://oro.open.ac.uk/13257/

[136] S. Ferguson, A. Siddiqi, K. Lewis, and O. L. de Weck, "Flexible and Reconfigurable Systems: Nomenclature and Review," *33rd Design Automation Conference*, vol. 6, pp. 249–263, 2007.

[137] M. Kwak and H. Kim, "Advances in Product Family and Product Platform Design," *Engineering Optimization Journal*, vol. 43, no. 3, pp. 707–735, 2011. [Online]. Available: http://link.springer.com/10.1007/978-1-4614-7937-6

[138] P. Clarkson, C. Simons, and C. Eckert, "Predicting change propagation in complex design," pp. 1–10, 2004. [Online]. Available: http://oro.open.ac.uk/18626/

[139] N. Ahmad, D. C. Wynn, and P. J. Clarkson, "Change impact on a product and its redesign process: A tool for knowledge capture and reuse," *Research in Engineering Design*, vol. 24, no. 3, pp. 219–244, 2013.

[140] C. Eckert, P. J. Clarkson, and W. Zanker, "Change and customisation in complex engineering domains," *Research in Engineering Design*, vol. 15, no. 1, pp. 1–21, 2004.

[141] R. J. Cloutier and D. Verma, "Applying the concept of patterns to systems architecture," *Systems Engineering*, vol. 10, no. 2, pp. 138–154, 2007.

[142] S. Friedenthal, A. Moore, and R. Steiner, *A practical guide to SysML: The systems modeling language*, 2008.

[143] M. A. Cardin, G. L. Kolfschoten, D. D. Frey, R. De Neufville, O. L. De Weck, and D. M. Geltner, "Empirical evaluation of procedures to generate flexibility in engineering systems and improve lifecycle performance," *Research in Engineering Design*, vol. 24, no. 3, pp. 277–295, 2013.

[144] C. J. Roberts, M. G. Richards, A. M. Ross, D. H. Rhodes, and D. E. Hastings, "Scenario planning in dynamic multi-attribute tradespace exploration," *2009 IEEE International Systems Conference Proceedings*, pp. 366–371, 2009.

[145] A. M. Ross, D. E. Hastings, J. M. Warmkessel, and N. P. Diller, "Multi-Attribute Tradespace Exploration as Front End for Effective Space System Design," *Journal of Spacecraft and Rockets*, vol. 41, no. 1, pp. 20–28, 2004.

[146] A. Ross and D. Hastings, "Assessing Changeability in Aerospace Systems Architecting and Design Using Dynamic Multi-Attribute Tradespace Exploration," *American Institute of Aeronautics and Astronautics*, vol. 7255, no. September, 2006.

[147] A. M. Ross, D. H. Rhodes, and D. E. Hastings, "Defining changeability: Reconciling flexibility, adaptability, scalability, modifiability, and robustness for maintaining system lifecycle value," *Systems Engineering*, vol. 11, no. 3, pp. 246–262, 2008.

[148] E. Fricke, B. Gebhard, H. Negele, and E. Igenbergs, "Coping with Changes: causes, Findings and Strategies," *Systems Engineering*, vol. 3, no. 4, pp. 169–179, 2000.

[149] J. Rumbaugh, I. Jacobson, and G. Booch, *The Unified Modeling Language Reference Manual*, 1999. [Online]. Available: http://www.amazon.com/s/ref=nb{_}sb{_}noss?url=search-alias{\%}3Daps{\&}field-keywords=9780201309980

[150] I. Lee, O. Sokolsky, S. Chen, and J. Hatcliff, "Challenges and Research Directions in Medical Cyber–Physical Systems," *Proceedings of the ...*, vol. 100, no. 1, pp. 75–90, 2012. [Online]. Available: http://ieeexplore.ieee.org/xpls/abs{_}all.jsp?arnumber=6051465

[151] D. W. Oliver, J. F. Andary, and H. Frisch, "Model-based systems engineering," *Handbook of Systems Engineering and Management*, no. December, pp. 1361–1400, 2009. [Online]. Available: http://www.knovel.com/web/portal/basic{_}search/display?{_}EXT{_}KNOVEL{_}DISPLAY{_}bookid=2817

[152] T. Weilkiens, *Systems Engineering with SysML/UML: Modeling, Analysis, Design*, ser. Morgan Kaufmann OMG press. Morgan Kaufmann OMG Press/Elsevier, 2007. [Online]. Available: https://books.google.dk/books?id=HMcqlAEACAAJ

[153] P. Derler, E. a. Lee, and A. Sangiovanni Vincentelli, "Modeling cyber-physical systems," *Proceedings of the IEEE*, vol. 100, no. 1, pp. 13–28, 2012.

[154] M. Rao, S. Ramakrishnan, and C. Dagli, "Modeling and simulation of net centric system of systems using systems modeling language and Colored Petri-nets: A demonstration using the Global Earth Observation System of Systems," *Systems Engineering*, vol. 11, no. 3, pp. 203–220, 2008.

[155] P. Shames, M. Anderson, S. Kowal, M. Levesque, O. Sindiy, K. Donahue, and P. Barnes, "NASA Integrated Network Monitor and Control Software Architecture," *The 12th International Conference on Space Operations*, pp. 1–21, 2012.

[156] D. Kaslow, G. Soremekun, H. Kim, and S. Spangelo, "Integrated model-based systems engineering (MBSE) applied to the Simulation of a CubeSat mission," *IEEE Aerospace Conference Proceedings*, 2014.

[157] S. Nakajima, S. Furukawa, and Y. Ueda, "Co-analysis of SysML and simulink models for cyber-physical systems design," *Proceedings - 18th IEEE International Conference on Embedded and Real-Time Computing Systems and Applications, RTCSA 2012 - 2nd Workshop on Cyber-Physical Systems, Networks, and Applications, CPSNA*, no. Figure 2, pp. 473–478, 2012.

[158] T. Sakairi, E. Palachi, C. Cohen, Y. Hatsutori, J. Shimizu, and H. Miyashita, "Designing a control system using SysML and Simulink," *SICE Annual Conference (SICE), 2012 Proceedings of*, pp. 2011–2017, 2012. [Online]. Available: http://ieeexplore.ieee.org/xpls/abs{_}all.jsp?arnumber=6318343

[159] L. Sha, S. Gopalakrishnan, X. Liu, and Q. Wang, "Cyber-Physical Systems: A New Frontier," in *2008 IEEE International Conference on Sensor Networks, Ubiquitous, and Trustworthy Computing (sutc 2008)*, 2008, pp. 1–9. [Online]. Available: http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=4545732

[160] S. K. Khaitan, S. Member, and J. D. Mccalley, "Design Techniques and Applications of Cyberphysical Systems : A Survey," *IEEE Systems Journal*, vol. 9, no. 2, pp. 350–365, 2015.

[161] L. Miclea and T. Sanislav, "About dependability in cyber-physical systems," in *Proceedings of IEEE East-West Design and Test Symposium, EWDTS'2011*, 2011, pp. 17–21.

[162] E. Lee, "Cyber Physical Systems: Design Challenges," *2008 11th IEEE International Symposium on Object and Component-Oriented Real-Time Distributed Computing (ISORC)*, 2008.

[163] E. S. Suh, O. L. De Weck, and D. Chang, "Flexible product platforms: Framework and case study," *Research in Engineering Design*, vol. 18, no. 2, pp. 67–89, 2007.

[164] T. W. Simpson, Z. Siddique, and J. Jiao, *Product platform and product family design: Methods and applications*, 2006.

[165] J. P. Womack, D. T. Jones, and D. Roos, *The Machine that Changed the World: The Story of Lean Production*, 1990.

[166] M. H. Meyer and A. P. Lehnerd, "The power of product platforms - building value and cost leadership - meyer,mh, lehnerd,ap," *Journal of Product Innovation Management*, vol. 14, pp. 526–529, 1997.

[167] E. Suk Suh, "Flexible Product Platforms," Ph.D. dissertation, 2005.

[168] T. W. Simpson, T. Marion, M. Kokkolaras, and A. Arbor, "Platform-Based Design And Development: Current Trends And Needs In Industry," in *Proceedings of IDETC/CIE 2006 ASME 2006 International Design Engineering Technical Conferences & Computers and Information in Engineering Conference*, 2006.

[169] G. S. Percivall, "SYSTEM ARCHITECTURE FOR EVOLUTIONARY SYSTEM DEVELOPMENT," *INCOSE International Symposium*, vol. 4, no. 1, pp. 532–536, 1994. [Online]. Available: http://dx.doi.org/10.1002/j.2334-5837.1994.tb01754.x

[170] A. M. Madni, "Adaptable platform-based engineering: Key enablers and outlook for the future," *Systems Engineering*, vol. 15, no. 1, pp. 95–107, 2012. [Online]. Available: http://ezproxy.lib.ucf.edu/login?url=http://search.proquest.com/docview/926281441?accountid=10003$\backslash$nhttp://sfx.fcla.edu/ucf?url{_}ver=Z39.88-2004{\&}rft{_}val{_}fmt=info:ofi/fmt:kev:mtx:journal{\&}genre=article{\&}sid=ProQ:ProQ:civilengineering{\&}atitle=Model+Based+Systems

[171] A. Sangiovanni-vincentelli and G. Martin, "Platform-Based Design and Software Design Methodology for Embedded Systems," *IEEE design and test of computers*, no. December, pp. 23–33, 2001.

[172] A. Sangiovanni-Vincentelli, "Quo vadis, SLD? Reasoning about the trends and challenges of system level design," *Proceedings of the IEEE*, vol. 95, no. 3, pp. 467–506, 2007.

[173] Y. Cui, Q. Wei, H. Park, and C. M. Lieber, "Nanowire nanosensors for highly sensitive and selective detection of biological and chemical species." *Science (New York, N.Y.)*, vol. 293, no. 5533, pp. 1289–1292, 2001.

[174] F. Patolsky, G. Zheng, and C. M. Lieber, "Nanowire-Based Biosensors," *Nature Protocol*, vol. 1, no. 4, pp. 1711–1724, 2006.

[175] E. Buitrago, G. Fagas, M. F. B. Badia, Y. M. Georgiev, M. Berthomé, and A. M. Ionescu, "Junctionless silicon nanowire transistors for the tunable operation of a highly sensitive, low power sensor," *Sensors and Actuators, B: Chemical*, vol. 183, pp. 1–10, 2013.

[176] L. Ansari, B. Feldman, G. Fagas, J. P. Colinge, and J. C. Greer, "Subthreshold behavior of junctionless silicon nanowire transistors from atomic scale simulations," *Solid-State Electronics*, vol. 71, pp. 58–62, 2012. [Online]. Available: http://dx.doi.org/10.1016/j.sse.2011.10.021

[177] J.-P. Colinge, C.-W. Lee, A. Afzalian, N. D. Akhavan, R. Yan, I. Ferain, P. Razavi, B. O'Neill, A. Blake, M. White, A.-M. Kelleher, B. McCarthy, and R. Murphy, "Nanowire transistors without junctions." *Nature nanotechnology*, vol. 5, no. 3, pp. 225–229, 2010. [Online]. Available: http://dx.doi.org/10.1038/nnano.2010.15

[178] M. Wipf, R. L. Stoop, A. Tarasov, K. Bedner, W. Fu, I. A. Wright, C. J. Martin, E. C. Constable, M. Calame, and C. Schönenberger, "Selective Sodium Sensing with Gold-Coated Silicon Nanowire Field-Effect Transistors in a Differential Setup," *ACS Nano*, vol. 7, no. 7, pp. 5978–5983, jul 2013. [Online]. Available: http://dx.doi.org/10.1021/nn401678u

[179] J.-i. Hahm and C. M. Lieber, "Direct Ultrasensitive Electrical Detection of DNA and DNA Sequence Variations Using Nanowire Nanosensors," *Nano Letters*, vol. 4, no. 1, pp. 51–54, 2004. [Online]. Available: http://pubs.acs.org/doi/abs/10.1021/nl034853b

[180] G. J. Zhang, J. H. Chua, R. E. Chee, A. Agarwal, and S. M. Wong, "Label-free direct detection of MiRNAs with silicon nanowire biosensors," *Biosensors and Bioelectronics*, vol. 24, no. 8, pp. 2504–2508, 2009.

[181] A. Kim, C. S. Ah, H. Y. Yu, J. H. Yang, I. B. Baek, C. G. Ahn, C. W. Park, M. S. Jun, and S. Lee, "Ultrasensitive, label-free, and real-time immunodetection using silicon field-effect transistors," *Applied Physics Letters*, vol. 91, no. 10, 2007.

[182] F. Patolsky, G. Zheng, O. Hayden, M. Lakadamyali, X. Zhuang, and C. M. Lieber, "Electrical detection of single viruses." *Proceedings of the National Academy of Sciences of the United States of America*, vol. 101, no. 39, pp. 14 017–14 022, 2004.

[183] S. Regonda, R. Tian, J. Gao, S. Greene, J. Ding, and W. Hu, "Silicon multi-nanochannel FETs to improve device uniformity/stability and femtomolar detection of insulin in serum," *Biosensors and Bioelectronics*, vol. 45, no. 1, pp. 245–251, 2013. [Online]. Available: http://dx.doi.org/10.1016/j.bios.2013.01.027

[184] L. De Vico, M. H. Sørensen, L. Iversen, D. M. Rogers, B. S. Sørensen, M. Brandbyge, J. Nygård, K. L. Martinez, and J. H. Jensen, "Quantifying signal changes in nano-wire based biosensors." *Nanoscale*, vol. 3, no. 2, pp. 706–717, 2011.

[185] Y. L. Wu, P. Y. Hsu, and J. J. Lin, "Polysilicon wire glucose sensor highly immune to interference," *Biosensors and Bioelectronics*, vol. 26, no. 5, pp. 2281–2286, 2011. [Online]. Available: http://dx.doi.org/10.1016/j.bios.2010.09.052

[186] G. Wenga, E. Jacques, a. C. Salaün, R. Rogel, L. Pichon, and F. Geneste, "Bottom-gate and step-gate Polysilicon nanowires field effect transistors for ultrasensitive label-free biosensing application," *Procedia Engineering*, vol. 47, pp. 414–417, 2012.

[187] C.-H. Lin, C.-Y. Hsiao, C.-H. Hung, Y.-R. Lo, C.-C. Lee, C.-J. Su, H.-C. Lin, F.-H. Ko, T.-Y. Huang, and Y.-S. Yang, "Ultrasensitive detection of dopamine using a polysilicon nanowire field-effect transistor." *Chemical communications (Cambridge, England)*, no. 44, pp. 5749–5751, 2008.

[188] G. Wenga, E. Jacques, a. C. Salaün, R. Rogel, L. Pichon, and F. Geneste, "Step-gate polysilicon nanowires field effect transistor compatible with CMOS technology for label-free DNA biosensor," *Biosensors and Bioelectronics*, vol. 40, no. 1, pp. 141–146, 2013.

[189] T. Adam, U. Hashim, T. S. Dhahi, K. L. Foo, P. L. Leow, and P. S. Chee, "Polysilicon Nanowire Fabrication as a Transducer for Fast Reaction Assays in Nano Lab-on-Chip Domain," *Sensor Letters*, vol. 11, no. 2, pp. 333–336, 2013. [Online]. Available: http://openurl.ingenta.com/content/xref?genre=article{\&}issn=1546-198X{\&}volume=11{\&}issue=2{\&}spage=333

[190] M. M. a. Hakim, M. Lombardini, K. Sun, F. Giustiniano, P. L. Roach, D. E. Davies, P. H. Howarth, M. R. R. De Planque, H. Morgan, and P. Ashburn, "Thin film polycrystalline silicon nanowire biosensors," *Nano Letters*, vol. 12, no. 4, pp. 1868–1872, 2012.

[191] P. R. Nair, S. Member, and M. a. Alam, "Design Considerations of Silicon Nanowire Biosensors," *IEEE Transactions on Electron Devices*, vol. 54, no. 12, pp. 3400–3408, 2007.

[192] E. Stern, R. Wagner, F. J. Sigworth, R. Breaker, T. M. Fahmy, and M. a. Reed, "Importance of the debye screening length on nanowire field effect transistor sensors," *Nano Letters*, vol. 7, no. 11, pp. 3405–3409, 2007.

[193] P. R. Nair and M. a. Alam, "Performance limits of nanobiosensors," *Applied Physics Letters*, vol. 88, no. 23, 2006.

[194] N. Elfström, A. E. Karlström, and J. Linnros, "Silicon nanoribbons for electrical detection of biomolecules," *Nano Letters*, vol. 8, no. 3, pp. 945–949, 2008.

[195] P. R. Nair and M. a. Alam, "Screening-limited response of NanoBiosensors," *Nano Letters*, vol. 8, no. 5, pp. 1281–1285, 2008.

[196] E. Buitrago, M. Fernández-Bolaños, and a. M. Ionescu, "Vertically stacked Si nanostructures for biosensing applications," *Microelectronic Engineering*, vol. 97, pp. 345–348, 2012. [Online]. Available: http://dx.doi.org/10.1016/j.mee.2012.03.017

[197] X. P. a. Gao, G. Zheng, and C. M. Lieber, "Subthreshold regime has the optimal sensitivity for nanowire FET biosensors," *Nano Letters*, vol. 10, no. 2, pp. 547–552, 2010.

[198] Y. L. Bunimovich, Y. S. Shin, W. S. Yeo, M. Amori, G. Kwong, and J. R. Heath, "Quantitative real-time measurements of DNA hybridization with alkylated nonoxidized silicon nanowires in electrolyte solution," *Journal of the American Chemical Society*, vol. 128, no. 50, pp. 16 323–16 331, 2006.

[199] M. Y. Shen, B. R. Li, and Y. K. Li, "Silicon nanowire field-effect-transistor based biosensors: From sensitive to ultra-sensitive," *Biosensors and Bioelectronics*, vol. 60, pp. 101–111, 2014. [Online]. Available: http://dx.doi.org/10.1016/j.bios.2014.03.057

[200] A. Pfreundt, "Functionalization and microfluidic integration of silicon nanowire biologically gated field effect transistors," Ph.D. dissertation, Technical University of Denmark, 2015.

[201] D. Wang, B. A. Sheriff, and J. R. Heath, "Silicon p-FETs from Ultrahigh Density Nanowire Arrays," *Nano Letters*, vol. 6, no. 6, pp. 1096–1100, jun 2006. [Online]. Available: http://dx.doi.org/10.1021/nl052558g

[202] B.-R. Li, Y.-J. Hsieh, Y.-X. Chen, Y.-T. Chung, C.-Y. Pan, and Y.-T. Chen, "An Ultrasensitive Nanowire-Transistor Biosensor for Detecting Dopamine Release from Living PC12 Cells under Hypoxic Stimulation," *Journal of the American Chemical Society*, vol. 135, no. 43, pp. 16 034–16 037, oct 2013. [Online]. Available: http://dx.doi.org/10.1021/ja408485m

[203] B.-R. Li, C.-W. Chen, W.-L. Yang, T.-Y. Lin, C.-Y. Pan, and Y.-T. Chen, "Biomolecular recognition with a sensitivity-enhanced nanowire transistor biosensor." *Biosensors & bioelectronics*, vol. 45, pp. 252–9, jul 2013. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0956566313001073

[204] G. Zheng, F. Patolsky, Y. Cui, W. U. Wang, and C. M. Lieber, "Multiplexed electrical detection of cancer markers with nanowire sensor arrays." *Nature biotechnology*, vol. 23, no. 10, pp. 1294–1301, 2005.

[205] G. J. Zhang, Z. H. H. Luo, M. J. Huang, J. J. Ang, T. G. Kang, and H. Ji, "An integrated chip for rapid, sensitive, and multiplexed detection of cardiac biomarkers from fingerprick blood," *Biosensors and Bioelectronics*, vol. 28, no. 1, pp. 459–463, 2011. [Online]. Available: http://dx.doi.org/10.1016/j.bios.2011.07.007

[206] G. J. Zhang, K. T. C. Chai, H. Z. H. Luo, J. M. Huang, I. G. K. Tay, A. E. J. Lim, and M. Je, "Multiplexed detection of cardiac biomarkers in serum with nanowire arrays using readout ASIC," *Biosensors and Bioelectronics*, vol. 35, no. 1, pp. 218–223, 2012.

[207] E. Stern, D. a. Routenberg, A. Vacic, N. K. Rajan, J. M. Criscione, J. Park, T. M. Fahmy, and M. Reed, "CMOS compatible nanowires for biosensing," *2011 48th ACM/EDAC/IEEE Design Automation Conference (DAC)*, pp. 718–722, 2011.

[208] T. Kong, R. Su, B. Zhang, Q. Zhang, and G. Cheng, "CMOS-compatible, label-free silicon-nanowire biosensors to detect cardiac troponin I for acute myocardial infarction diagnosis," *Biosensors and Bioelectronics*, vol. 34, no. 1, pp. 267–272, 2012.

[209] M. C. Chen, H. Y. Chen, C. Y. Lin, C. H. Chien, T. F. Hsieh, J. T. Horng, J. T. Qiu, C. C. Huang, C. H. Ho, and F. L. Yang, "A CMOS-compatible poly-si nanowire device with hybrid sensor/memory characteristics for system-on-chip applications," *Sensors*, vol. 12, no. 4, pp. 3952–3963, 2012.

[210] X. T. Vu, R. GhoshMoulick, J. F. Eschermann, R. Stockmann, a. Offenhäusser, and S. Ingebrandt, "Fabrication and application of silicon nanowire transistor arrays for biomolecular detection," *Sensors and Actuators, B: Chemical*, vol. 144, no. 2, pp. 354–360, 2010.

[211] M. L. Meade, *Lock-in Amplifiers: Principles and Applications*, ser. IEE electrical measurement series. P. Peregrinus, 1983. [Online]. Available: https://books.google.dk/books?id=cg9TAAAAMAAJ

[212] J. Leis, C. Kelly, and D. Buttsworth, "Sampling , Quantization and Computational Aspects of the Quadrature Lock-In Amplifier," in *Signal Processing and Communication Systems (ICSPCS), 2012 6th International Conference on*, 2012, pp. 1–7.

[213] N. Li, W. Wang, H. Xu, H. Yu, J. Diao, and D. D.-U. Li, "Wide-Bandwidth Biological Impedance Spectroscopy System Based on the Digital Lock-In Technique," *Spectroscopy Letters*, vol. 46, no. 7, pp. 476–482, 2013. [Online]. Available: http://www.tandfonline.com/doi/abs/10.1080/00387010.2012.705801

[214] M. H. Sørensen, N. A. Mortensen, and M. Brandbyge, "Screening model for nanowire surface-charge sensors in liquid," *Applied Physics Letters*, vol. 91, no. 10, 2007.

[215] G. Ferrari, F. Gozzini, and M. Sampietro, "Transimpedance amplifiers for extremely high sensitivity impedance measurements on nanodevices," *Analog Circuit Design: High-speed Clock and Data Recovery, High-performance Amplifiers, Power Management*, pp. 193–207, 2006.

[216] A. Device, "Analog Devices AN8608 Data Sheet," pp. 1–24, 2003.

[217] L. Orozco, "Programmable-Gain Transimpedance Amplifiers Maximize Dynamic Range in Spectroscopy Systems," *Analog Dialogue*, vol. 47, no. 05, pp. 1–5, 2013.

[218] A. devices, "MT-048 - Op Amp Noise Relationships: 1/f Noise, RMS Noise, and Equivalent Noise Bandwidth," Tech. Rep., 2010.

[219] Texas Instruments, "Noise Analysis in Operational Amplifier Circuits," Tech. Rep., 2007.

[220] S. Reza, G. Bosman, M. S. Islam, T. I. Kamins, S. Sharma, and R. S. Williams, "Noise in silicon nanowires," *IEEE Transactions on Nanotechnology*, vol. 5, no. 5, pp. 523–528, 2006.

[221] N. K. Rajan, D. A. Routenberg, J. Chen, and M. A. Reed, "1/f noise of silicon nanowire BioFETs," *IEEE Electron Device Letters*, vol. 31, no. 6, pp. 615–617, 2010.

[222] W. Kester, "Taking the Mystery out of the Infamous Formula," SNR= 6.02 N+ 1.76 dB," and Why You Should Care," *MT-001 Tutorial*, pp. 1–7, 2005. [Online]. Available: http://qt.tn.tudelft.nl/{~}schouten/linkload/adc-tutorial.pdf

[223] J. G. Proakis and D. G. Manolakis, *Digital Signal Processing (3rd Ed.): Principles, Algorithms, and Applications*. Upper Saddle River, NJ, USA: Prentice-Hall, Inc., 1996.

[224] R. Steiner, "System Architectures and Evolvability: Definitions and Perspective," *System*, p. 5, 1998.

[225] G. Gossler and J. Sifakis, "Composition for component-based modeling," in *Science of Computer Programming*, vol. 55, no. 1-3 SPEC. ISS., 2005, pp. 161–183.

[226] F. Patou, M. Dimaki, W. E. Svendsen, C. Kjægaard, and J. Madsen, "Smartphone-based biosensing platform evolution : implementation of electrochemical analysis capabilities ." in *International Symposium on Medical Communication and Information Technologies*, 2016, p. [In Press].

[227] E. A. Lee, "CPS foundations," *Design Automation Conference (DAC), 2010 47th ACM/IEEE*, pp. 737–742, 2010.

[228] P. Nuzzo, J. Finn, M. Mozumdar, and A. Sangiovanni-vincentelli, "Platform-Based Design Methodology and Modeling for Aircraft Electric Power Systems," *Proceedings of Green Energy and Systems Conference*, pp. 1–7, 2013.

[229] A. Sangiovanni-vincentelli, "Platform-Based Design and Software Design Methodology for Embedded Systems," no. December, pp. 23–33, 2001.

[230] M. Fowler, *Patterns of Enterprise Application Architecture*, 2002, vol. 48, no. 2. [Online]. Available: http://www.amazon.com/Patterns-Enterprise-Application-Architecture-Martin/dp/0321127420

[231] E. Gamma, R. Helm, R. Johnson, and J. Vlissides, "Design Patterns: Elements of Reusable Software," *Addison-Wesley Professional Computing Series*, p. 395, 1996.

[232] Texas Instruments, "F28377D Technical Reference Manual," 2008.

[233] "C2000 Delfino F28377S LaunchPad." [Online]. Available: http://www.ti.com/tool/launchxl-f28377s

[234] G. Zheng, F. Patolsky, and C. Lieber, "Nanowire biosensors: a tool for medicine and life science," *Nanomedicine: Nanotechnology, Biology and Medicine*, vol. 2, no. 4, p. 277, 2006. [Online]. Available: http://dx.doi.org/10.1016/j.nano.2006.10.050

[235] D. Kwasny, M. Dimaki, K. B. Andersen, A. Zulfiqar, Z. Tümer, and W. E. Svendsen, "Nanoscaled Biological Gated Field Effect Transistors for Cytogenetic Analysis," in *Proceedings of the 9th IEEE International Conference on Nano/Micro Engineered and Molecular Systems*, 2014, pp. 130–134.

[236] M. Pachchigar, "Increase dynamic range of SAR ADCs," 2015.

[237] R. Andraka, "A survey of CORDIC algorithms for FPGA based computers," *Proceedings of the 1998 ACMSIGDA sixth international symposium on Field programmable gate arrays FPGA 98*, pp. 191–200, 1998. [Online]. Available: http://portal.acm.org/citation.cfm?doid=275107.275139

[238] V. Viswam, J. Dragas, A. Shadmani, Y. Chen, A. Stettler, J. Müller, and A. Hierlemann, "Multi-Functional Microelectrode Array System," in *IEEE International Solid-State Circuits Conference*, 2016, pp. 394–396.

[239] D. M. Alter, "Using PWM output as a Digital-to-Analog converter on a TMS320F280x Digital Signal Controller," 2008.

[240] Y. Xu, X. Xie, Y. Duan, L. Wang, Z. Cheng, and J. Cheng, "A review of impedance measurements of whole cells," *Biosensors and Bioelectronics*, vol. 77, pp. 824–836, 2016. [Online]. Available: http://dx.doi.org/10.1016/j.bios.2015.10.027

[241] Bluegiga, "Bluegiga Bluetooth Smart Software V.1.2 API documentation," pp. 1–202, 2014.

[242] "Bluetooth Profiles." [Online]. Available: https://developer.bluetooth.org/TechnologyOverview/Pages/Profiles.aspx

[243] N. J. Ronkainen, H. B. Halsall, and W. R. Heineman, "Electrochemical biosensors." *Chemical Society reviews*, vol. 39, no. 5, pp. 1747–1763, 2010.

[244] K. Jackowska and P. Krysinski, "New trends in the electrochemical sensing of dopamine," *Analytical and Bioanalytical Chemistry*, vol. 405, no. 11, pp. 3753–3771, 2013.

[245] U. E. Majewska, K. Chmurski, K. Biesiada, A. R. Olszyna, and R. Bilewicz, "Dopamine Oxidation at Per(6-deoxy-6-thio)-α-Cyclodextrin Monolayer Modified Gold Electrodes," *Electroanalysis*, vol. 18, no. 15, pp. 1463–1470, 2006. [Online]. Available: http://dx.doi.org/10.1002/elan.200603556

[246] F. Alatraktchi, T. Bakmand, M. Dimaki, and W. Svendsen, "Novel Membrane-Based Electrochemical Sensor for Real-Time Bio-Applications," *Sensors*, vol. 14, no. 11, pp. 22 128–22 139, 2014. [Online]. Available: http://www.mdpi.com/1424-8220/14/11/22128/

[247] T. Bakmand, D. Kwasny, M. Dimaki, and W. E. Svendsen, "Fabrication and Characterisation of Membrane-Based Gold Electrodes," *Electroanalysis*, vol. 27, no. 1, pp. 217–224, 2015. [Online]. Available: http://doi.wiley.com/10.1002/elan.201400407

[248] S. Saha, P. Sarkar, and A. Turner, "Interference-Free Electrochemical Detection of Nanomolar Dopamine Using Doped Polypyrrole and Silver Nanoparticles Interference-free Electrochemical Detection of Nanomolar Dopamine Using Doped Polypyrrole and Silver Nanoparticles," *Electroanalysis*, no. 26, pp. 2197–2206, 2014.

[249] C. B. Weinstock and J. B. Goodenough, "On System Scalability," *Technical Note, Carnegie Mellon Software Engineering Institute*, no. March, 2006. [Online]. Available: http://repository.cmu.edu/sei/399/

[250] M. D. Hill, "What is scalability?" *ACM SIGARCH Computer Architecture News*, vol. 18, no. 4, pp. 18–21, 1990.

[251] L. Duboc, D. Rosenblum, and T. Wicks, "A framework for characterization and analysis of software system scalability," *Proceedings of the the 6th joint meeting of the European software engineering conference and the ACM SIGSOFT symposium on The foundations of software engineering*, pp. 375–384, 2007. [Online]. Available: http://portal.acm.org/citation.cfm?doid=1287624.1287679$\backslash$nhttp://dl.acm.org/citation.cfm?id=1287679

[252] F. Patou, M. Dimaki, W. E. Svendsen, and J. Madsen, "Model-Based Evaluation Of System Scalability : Bandwidth Analysis For Smartphone-Based Biosensing Applications," in *2016 Euromicro Conference on Digital System Design*, 2016.

[253] D. Zhang, Y. Lu, Q. Zhang, L. Liu, S. Li, Y. Yao, J. Jiang, G. L. Liu, and Q. Liu, "Protein detecting with smartphone-controlled electrochemical impedance spectroscopy for point-of-care applications," *Sensors and Actuators, B: Chemical*, vol. 222, pp. 994–1002, 2016. [Online]. Available: http://dx.doi.org/10.1016/j.snb.2015.09.041

[254] M. J. Puckelwartz, L. L. Pesce, V. Nelakuditi, L. Dellefave-Castillo, J. R. Golbus, S. M. Day, T. P. Cappola, G. W. Dorn, I. T. Foster, and E. M. Mcnally, "Supercomputing for the parallelization of whole genome analysis," *Bioinformatics*, vol. 30, no. 11, pp. 1508–1513, 2014.

[255] "IBM Watson and personalised medicine." [Online]. Available: http://www.businessinsider.com/sc/ibm-watson-and-personalized-healthcare-for-cancer-2016-1?r=US{\&}IR=T{\&}IR=T

[256] R. D. Espy, N. E. Manicke, Z. Ouyang, and R. G. Cooks, "Rapid analysis of whole blood by paper spray mass spectrometry for point-of-care therapeutic drug monitoring," *The Analyst*, vol. 137, no. 10, pp. 2344 – 2349, 2012.

[257] D. Gao, H. Liu, Y. Jiang, and J.-M. Lin, "Recent advances in microfluidics combined with mass spectrometry: technologies and applications." *Lab on a chip*, vol. 13, no. 17, pp. 3309–22, 2013. [Online]. Available: http://www.ncbi.nlm.nih.gov/pubmed/23824006

[258] C. R. Ferreira, K. E. Yannell, A. K. Jarmusch, V. Pirro, Z. Ouyang, and R. Graham Cooks, "Ambient ionization mass spectrometry for point-of-care diagnostics and other clinical measurements," pp. 99–110, 2016.

[259] J. Li, S. G. Ballmer, E. P. Gillis, S. Fujii, M. J. Schmidt, A. M. E. Palazzolo, J. W. Lehmann, G. F. Morehouse, and M. D. Burke, "Automated Process," *Org. Synth.*, vol. 347, no. 6227, p. 1221, 2015.

[260] S. C. C. Shih, G. Goyal, P. W. Kim, N. Koutsoubelis, J. D. Keasling, P. D. Adams, N. J. Hillson, and A. K. Singh, "A Versatile Microfluidic Device for Automating Synthetic Biology." *ACS Synthetic Biology*, p. 150615150549007, 2015. [Online]. Available: http://pubs.acs.org/doi/abs/10.1021/acssynbio.5b00062$\backslash$npapers3://publication/doi/10.1021/acssynbio.5b00062

[261] G. Linshiz, E. Jensen, N. Stawski, C. Bi, N. Elsbree, H. Jiao, J. Kim, R. Mathies, J. D. Keasling, and N. J. Hillson, "End-to-end automated microfluidic platform for synthetic biology: from design to functional analysis," *Journal of Biological Engineering*, vol. 10, no. 1, p. 15, 2016. [Online]. Available: http://www.jbioleng.org/content/10/1/3

List of Abbreviations

ADC Analog to Digital Converter

AI Artificial Intelligence

API Application Programming Interface

ATP Adenosine Triphosphase

ASIC Application Specific Integrated Circuit

BLE Bluetooth Low Energy

CMOS Complementary Metal Oxide Semiconductor

CPU Central Processing Unit

CLA Control Law Accelerator

CPA Change Propagation Analysis

CPS Cyber Physical Systems

CV Cyclic Voltammetry

DAC Digital to Analog Converter

DLT Direct Look-up Table

DMA Direct Memory Addressing

DDS Direct Digital Synthesis

DMF Digital Microfluidics

DPV Differential Pulse Voltammetry

DSC Digital Signal Controller

DSM Design Structure Matrix

DSP Digital Signal Processing

DUT Device Under Test

EHR Electronic Health Record

EIS Electrochemical Impedance Spectroscopy

EMI Electromagnetic Interferences

EOT End Of Transmission

ET Elementary Task

ELISA Enzyme-Linked Immunosorbent Assays

ENOD Effective Number of Bits

EWOD Electrowetting On Dielectric

FIFO First In First Ou

FPGA Field Programmable Gate Array

GATT Generic Attribute

GBP Gain Bandwidth Product

GFLOPS Giga Floating-point Operations Per Second

GUI Graphical User Interface

HLT High-Level Task

HWA Hardware Accessory

IC Integrated Circuit

INCOSE International Council on Systems Engineering

IO Input/Output

IoT Internet of Things

LLT Low-Level Task

LOC Lab-On-Chip

LPF Low-Pass Filter

LUT Look Up Table

MBD Model-Based Development

MBSE Model-Based Systems Engineering

MCU Micro Controller Unit

MOC Model Of Computation

MVC Model-View-Controller

mVLSI microfluidic- Very Large Scale Integration

MATE Multi-Attribute Tradespace Exploration

NIH National Institutes of Health

NPT Near-Patient Technology

OOP Object-Oriented Programming

OTS Off The Shelf

PCB Printed Circuit Board

PCR Polymerase Chain Reaction

PHR Personal Health Record

PIE Peripheral Interrupt Expansion

POC Point-of-Care

PSD Phase Sensitive Detection

PWM Pulse Width Modulation

RTI Referred to The Input

RTO Referred to the Output

RTOS Real-Time Operating System

SAR Successive Approximation Register

SEARi Systems Engineering Advancement Research initiative

S/H Sample and Hold

SE Systems Engineering

SiNW Silicon Nano-Wire

SiNW-bioFET Silicon Nano-Wire Biological Field Effect Transistor

SMD Surface Mount Device

SNR Signal to Noise Ratio

SOC System On Chip

SoS System-of-Systems

SPGE Screen Printed Gold Electrodes

SPI Serial Peripheral Interface

SPST Single Pole Single Throw

SWCV Square Wave Cyclic Voltammetry

TIA Transimpedance Amplifier

THD Total Harmonic Distortion

Appendices

Links to publications

Appendix A

F. Patou, M. Dimaki, W. E. Svendsen, K. Kjaegaard, and J. Madsen, "A Smart Mobile Lab-on-Chip-Based Medical Diagnostics System Architecture Designed for Evolvability," *2015 Euromicro Conference on Digital System Design, pp. 390–398, 2015.* [Online]. Available: http://ieeexplore.ieee.org/document/7302301/

Appendix B

F. Patou, M. Dimaki, W. E. Svendsen, C. Kjægaard, and J. Madsen, "Smartphone-based biosensing platform evolution : implementation of electrochemical analysis capabilities ." in *International Symposium on Medical Communication and Information Technologies, 2016* http://ieeexplore.ieee.org/document/7498881/

Appendix C

F.Patou, F.A. Alatraktchi, C. Kjægaard, M.Dimaki, J. Madsen, W.E. Svendsen, "Model-Based Evaluation Of System Scalability: Bandwidth Analysis For Smartphone-Based Biosensing Applications" 2016 Euromicro Conference on Digital System Design [in press]

Appendix D

F.Patou, F.A. Alatraktchi, C. Kjægaard, M.Dimaki, J. Madsen, W.E. Svendsen, "Evolvable Smartphone-Based Platforms for Point-of-Care In-Vitro Diagnostics Applications" *Diagnostics, 2016* http://www.mdpi.com/2075-4418/6/3/33

Appendix E

A.Zulfiqar, F.Patou, A.Pfreundt, W.E.Svendsen, M.Dimaki, "*In-situ doped junctionless polysilicon nanowires field effect transistors for low-cost biosensor", Sensing And Biosensing Research, 2016* http://www.sciencedirect.com/science/article/pii/S2214180416301052

Appendix F

Signal conditioning analytical model

```
PlotImprovementScript;
%% TIA MODEL
%load TIA_model_defaultSettings;
dop = 1e24;
length = 100e-6;
analyteC = 0;
Kd = 62e-9;
Qc = 0;% -4.80652986e-19;
l = 2.4e-9;

%% SINW MODEL

format long

%% CONSTANTS
pi = 3.14159265359;
e = 1.60217662e-19;
Kb = 1.38064852e-23;
m = 9.10938356e-31;
h = 6.62607004e-34;
epsilon0 = 8.854187817e-12;
Na = 6.0221409e23;
sigma_S = 0;


%% SINW PRE-DEFINED PARAMETER VALUES
width = 400e-9;                         % SiNW width
height = 30e-9;                         % SiNW height
oxthickness = 5e-9;                     % Gate oxide thickness
mobility = 0.0002;                      % Mobility
epsilon1 = 12;                          % SiNW rel. perm.
epsilon2 = 3.9;                         % Oxide rel. perm.
epsilon3 = 78;                          % Solution rel. perm.
temp = 298.15;                          % Temperature
funcLayerCoverage = 65e12;              % Num. receptor/m^2
approxSurfAreaTargetMol = 0;            % Approx. surf. covered by target molecule
ionicStrength = 1e-12;                  % Ionic strength
lambdaD = 0;                            % Debye-Huckel screening length
Inoisesinw_psd = 0;

%% SINW PARAMETRIC FUNCTIONS
lambdaTF = sqrt(epsilon0*epsilon1*Kb*temp/(dop*e^2));
if lambdaD == 0
    lambdaD = sqrt(epsilon0*epsilon3*Kb*temp/(2*Na*e^2*ionicStrength));
end
sinwAndGatePerimeter = (2*(height+oxthickness)+width+2*oxthickness);
gateSurfaceArea = sinwAndGatePerimeter*length;
if approxSurfAreaTargetMol == 0
    maxReceptorsOnGate = funcLayerCoverage*gateSurfaceArea;
else
    maxReceptorsOnGate = gateSurfaceArea/approxSurfAreaTargetMol;
end

adsorbedFraction = analyteC/(analyteC + Kd);
Nc = maxReceptorsOnGate*adsorbedFraction;
sigma_B = Nc*Qc/((2*(height+oxthickness)+width+2*oxthickness+l)*length);
tau = (epsilon1*besselk(0,(height+oxthickness)/lambdaD)*(lambdaD/lambdaTF)*...
    besseli(1,height/lambdaTF))/((besselk(0,(height+oxthickness)/lambdaD)*...
    (lambdaD/(height+oxthickness))+log(1+oxthickness/height)*...
    besselk(1,(height+oxthickness)/lambdaD)*epsilon3/epsilon2)*epsilon1*...
    (height/lambdaTF)*besseli(1,height/lambdaTF) + epsilon3*...
    besselk(1,(height+oxthickness)/lambdaD)*besseli(0,height/lambdaTF));
taul = 2*(height/(height+l))/(1+sqrt(height/(height+l))*exp(l/lambdaD));
Rsens = 1/((1-2*tau*(taul*sigma_B+sigma_S)/(height*e*dop))*dop*height*width*e*...
    mobility/length);



C_stray = 1e-13;
cutoffFreqForSiNW = 1/(2*pi*Rsens*C_stray);

%% SENSOR TRANSFER FUNCTION - BODE ANALYSIS
w_sens = 1/(2*pi*Rsens*C_stray);
figure;
set(gca,'FontSize',12);
 s = tf('s');
H_sensor = 1/(1 + s*Rsens*C_stray);
H_sensor_plot = bodeplot(H_sensor);
options = getoptions(H_sensor_plot);
options.Title.FontSize = 16;
options.XLabel.FontSize = 14;
options.YLabel.FontSize = 14;
options.TickLabel.FontSize = 12;
options.FreqUnits = 'Hz';
options.MagUnits = 'dB';
options.PhaseUnits = 'deg';
h = findobj(gcf,'type','line');
ax = findobj(gcf,'type','axis');
set(ax,'linewidth',1);
set(h,'linewidth',1.5);
setoptions(H_sensor_plot,options);
legend('boxoff')
grid on
title('SiNW-bioFET magnitude and phase Bode plot','FontSize',16)



%% ANALOG

VhighAdc = 3.3;
dacOffset = (2234/4096)*3.29;
Rf = 1e8;
Cf = 1.5e-12;


%% TIA
a0 = 9e6;            % Open-loop gain = 1000V/mV = 1e6
w1 = 2*pi*0.9;           % Opamp open-loop pole cutoff (in Hz);
H_tia = a0/(1+s/w1);     % laplace transfer function
I_noise2 = (1e-14)^2;          % A^2/Hz
I_noise = 1e-14;
V_noise2 = (8e-9)^2;           % V^2/Hz
V_noise = 8e-9;
I_bias = 100e-12;
V_offset = 80e-6;
C_in_CM = 8.8e-12;
C_in_DIFF = 2.6e-12;
slewRate = 5e6 ;              % V/s
Rout = 1;
BW = 9e6;
GBW = 9e6;


Cshunt = C_stray + C_in_CM;
Fz = 1/(2*pi*Rf*(Cf+Cshunt));
Fp = 1/(2*pi*Rf*Cf);
```

```
F0 = sqrt(GBW * Fz);
Q = F0/(GBW*(Fz/Fp)+Fz);
Kb = 1.38064852e-23;        % Boltzmann constant

Vnoise_out_tia = sqrt((V_noise2 + 4*Kb*298*Rsens)* F0*(1+GBW/Fz)*(pi/2)*Q +...
    (I_noise2*(Rf^2) + 4*Kb*298*Rf)*F0*(pi/2)*Q);

% save TIA_model_defaultSettings;


% figure;
H_tia_bode = bodeplot(H_tia,'r');
options = getoptions(H_tia_bode);
options.FreqUnits = 'Hz';
options.MagUnits = 'dB';
options.PhaseUnits = 'deg';
options.PhaseVisible = 'off';
options.Title.FontSize = 18;
options.XLabel.FontSize = 16;
options.YLabel.FontSize = 16;
options.TickLabel.FontSize = 14;
options.Xlim = [1e3 1e10];
options.Ylim = [-20 80];
setoptions(H_tia_bode,options);
h = findobj(gcf,'type','line');
set(h,'linewidth',2);
grid on
title('AD8608 Open-loop magnitude Bode plot');



%% CLOSED-LOOP ANALYSIS

F_noise = (1+s*Cf*Rf)/(1+s*(Cshunt+Cf)*(Rf*Rsens)/(Rf+Rsens));
H_signal = (Rf/Rsens)/(1+s*Rf*Cf);
save TIA_model_defaultSettings;


% For A*B >> 1; we can approximate A = 1/B
% NOISE GAIN
figure;
H_noise = feedback(H_tia,F_noise);
H_cl_bode = bodeplot(H_signal,'g',H_noise,'b');
options2 = getoptions(H_cl_bode);
options2.FreqUnits = 'Hz';
options2.MagUnits = 'dB';
options2.PhaseUnits = 'deg';
options2.PhaseVisible = 'off';
options2.Title.FontSize = 18;
options2.XLabel.FontSize = 16;
options2.YLabel.FontSize = 16;
options2.TickLabel.FontSize = 14;
options2.Xlim = [1 1e10];
options2.Ylim = [-50 50];
options2.Ylim
setoptions(H_cl_bode,options2);
h = findobj(gcf,'type','line');
set(h,'linewidth',2);
legend('Close-loop Signal Gain','Close-loop noise Gain')
grid on
legend('boxoff')
title('AD8608 Closed-loop signal magnitude and noise gain Bode plots')


I_noise2 = I_noise^2;           % A^2/Hz
Fz = 1/(2*pi*Rf*(Cf+Cshunt));
Fp = 1/(2*pi*Rf*Cf);
F0 = sqrt(GBW * Fz);
Q = F0/(GBW*(Fz/Fp)+Fz);
Kb = 1.38064852e-23;        % Boltzmann constant

%% Noise gain profile
noiseGainLow = 1+Rf/Rsens;
noiseGainCutoff = 1/(2*pi*((Rf*Rsens/(Rf+Rsens))*(Cf+Cshunt)));
noiseGainHigh = 1+(Cshunt/Cf);
R_ref = 1e5;

%% NOISE BANDWIDTH OF THE MODULATED SIGNAL:
tiaCutoff = 1/(2*pi*Rf*Cf);
K2 = ((29.8e-9)^2-V_noise^2)*10;     % opamp psd at 10Hz is 29.8nV/sqrt(Hz)
flickerNoiseCutoff = K2/(V_noise^2);
noiseGainBandwidth = sqrt(1e7/(2*pi*Rf*Cshunt));


% NOISE CALCULATION FULL TIA BANDWIDTH
en_pinkpluswhite2_low = sqrt((V_noise^2)*noiseGainLow^2*(flickerNoiseCutoff*...
    (log(noiseGainCutoff/0.01)+noiseGainCutoff-0.01)));
en_pinkpluswhite2_cutoff = sqrt((V_noise^2)*((1/noiseGainCutoff)*(1+Rf/Rsens))^2*...
    ((tiaCutoff)/3 - (noiseGainCutoff)/3));
en_pinkpluswhite2_high = sqrt((V_noise^2)*(noiseGainHigh^2)*(1.54*noiseGainBandwidth-
...
    tiaCutoff));
erms = sqrt(en_pinkpluswhite2_low^2 + en_pinkpluswhite2_cutoff^2 +...
    en_pinkpluswhite2_high^2);

in2_low = I_noise2*(Rf^2 + R_ref^2*noiseGainLow^2)*(noiseGainCutoff);
in2_cutoff = I_noise2*(Rf^2 + R_ref^2*((1/noiseGainCutoff)*(1+Rf/Rsens))^2)*...
    ((tiaCutoff)/3 - (noiseGainCutoff)/3);
in2_high =  I_noise2*(Rf^2 + R_ref^2*noiseGainHigh^2)*(1.54*noiseGainBandwidth-...
    tiaCutoff);
thermal_noise2_low = 4*Kb*298*(R_ref*noiseGainLow^2)*(noiseGainCutoff);
thermal_noise2_cutoff = 4*Kb*298*(R_ref*((1/noiseGainCutoff)*(1+Rf/Rsens))^2)*...
    ((tiaCutoff)/3 - (noiseGainCutoff)/3);
thermal_noise2_high = 4*Kb*298*(R_ref*noiseGainHigh^2)*(1.54*noiseGainBandwidth-...
    tiaCutoff);

thermal_noise_Rf = sqrt(4*Kb*298*Rf*tiaCutoff*1.57);
thermal_noise_Rsens = sqrt(4*Kb*298*Rsens*(Rf/Rsens)^2*tiaCutoff*1.57);

%Antialiasing filter
Raa = 1e3;
Caa = 1e-9;
H_aa = 1/(1+s*(2*pi*Raa*Caa));
ENBaa = 1.57/(2*pi*Raa*Caa);
Vnoise_aa = sqrt(4*Kb*298*Raa) * ENBaa;

% ADC
adcResolution = 12;
VhighAdc = 3.3;
lsbVEq = VhighAdc/(2^adcResolution);
Vnoise_refered_to_input_adc = 4*lsbVEq;            % Vrms
SNRquantization = 6.02*adcResolution + 1.76;       % Ideal SNR for quantization
Vnoise_quantization = lsbVEq/sqrt(adcResolution); % Vnoise quantization rms

% NOISE PSD
excitationFrequency = 10;
```

```
Vexc = 0.3;
nbSamplingPointsPerAcquisitionPeriod = 4;
samplingFrequency = excitationFrequency * nbSamplingPointsPerAcquisitionPeriod;
[Hfir,Hdemod,ENBW] = initFIRFilter(228,excitationFrequency,samplingFrequency);

wexc = 2*pi*excitationFrequency;
s = tf('s');
H_sensor = 1/(1 + s*Rsens*C_stray);
attSensor = bode(H_sensor,wexc);
H_signal = attSensor*(Rf/Rsens)/(1+s*Rf*Cf);
attExcSignal = bode(H_signal,wexc);

if excitationFrequency < flickerNoiseCutoff
    en_pinkpluswhite2_lpf = (V_noise^2)*noiseGainLow^2*(flickerNoiseCutoff*...
        log((excitationFrequency+ENBW)/excitationFrequency) + ENBW);
    in2_lpf = I_noise2*(Rf^2 + R_ref^2*noiseGainLow^2)*ENBW;
    thermal_noise2_lpf = 4*Kb*298*(R_ref*noiseGainLow^2)*ENBW;
elseif excitationFrequency < noiseGainCutoff
    if excitationFrequency + ENBW < noiseGainCutoff
        en_pinkpluswhite2_lpf = (V_noise^2)*noiseGainLow^2*ENBW;
        in2_lpf = I_noise2*(Rf^2 + R_ref^2*noiseGainLow^2)*ENBW;
        thermal_noise2_lpf = 4*Kb*298*(R_ref*noiseGainLow^2)*ENBW;
    else
        en_pinkpluswhite2_lpf = (V_noise^2)*noiseGainLow^2*(noiseGainCutoff-...
            excitationFrequency) +
(V_noise^2)*((1/noiseGainCutoff)*(1+Rf/Rsinw))^2*...
            (((excitationFrequency+ENBW))/3 - (noiseGainCutoff)/3);
        in2_lpf = I_noise2*(Rf^2 + R_ref^2*noiseGainLow^2)*(noiseGainCutoff-...
            excitationFrequency) + I_noise2*(Rf^2 + R_ref^2*((1/noiseGainCutoff)*...
            (1+Rf/Rsinw))^2)*(((excitationFrequency+ENBW))/3 - (noiseGainCutoff)/3);
        thermal_noise2_lpf = 4*Kb*298*(R_ref*noiseGainLow^2)*ENBW +
4*Kb*298*(R_ref*...
            ((1/noiseGainCutoff)*(1+Rf/Rsinw))^2)*(((excitationFrequency+ENBW))/3 -...
            (noiseGainCutoff)/3);
    end
elseif excitationFrequency < tiaCutoff
    if excitationFrequency + ENBW < tiaCutoff
         en_pinkpluswhite2_lpf = (V_noise^2)*((1/noiseGainCutoff)*(1+Rf/Rsinw))^2*...
             (((excitationFrequency+ENBW))/3 - (excitationFrequency^3)/3);
        in2_lpf = I_noise2*(Rf^2 + R_ref^2*((1/noiseGainCutoff)*(1+Rf/Rsinw))^2)*...
            (((excitationFrequency+ENBW))/3 - (excitationFrequency^3)/3);
        thermal_noise2_lpf = 4*Kb*298*(R_ref*((1/noiseGainCutoff)*(1+Rf/Rsinw))^2)*...
            (((excitationFrequency+ENBW))/3 - (noiseGainCutoff^3)/3);
    else
        en_pinkpluswhite2_lpf = (V_noise^2)*((1/noiseGainCutoff)*(1+Rf/Rsinw))^2*...
            ((tiaCutoff)/3 - (excitationFrequency)/3) +
(V_noise^2)*(noiseGainHigh^2)*...
            (excitationFrequency+ENBW-tiaCutoff);
        in2_lpf = I_noise2*(Rf^2 + R_ref^2*((1/noiseGainCutoff)*(1+Rf/Rsinw))^2)*...
            ((tiaCutoff)/3 - (excitationFrequency)/3) + I_noise2*(Rf^2 + R_ref^2*...
            (noiseGainHigh^2))*(excitationFrequency+ENBW-tiaCutoff);
        thermal_noise2_lpf = 4*Kb*298*(R_ref*((1/noiseGainCutoff)*(1+Rf/Rsinw))^2)*...
            (((excitationFrequency+ENBW))/3 - (noiseGainCutoff)/3) +
4*Kb*298*(R_ref*...
            noiseGainHigh^2)*(excitationFrequency+ENBW-tiaCutoff);

    end
else
    en_pinkpluswhite2_lpf = (V_noise^2)*(noiseGainHigh^2)*ENBW;
    in2_lpf = I_noise2*(Rf^2 + R_ref^2*noiseGainHigh^2)*ENBW;
    thermal_noise2_lpf =  4*Kb*298*(R_ref*noiseGainHigh^2)*ENBW;
end

thermal_noise_Rf = sqrt(4*Kb*298*Rf*ENBW);
thermal_noise_Rsens = sqrt(4*Kb*298*Rsens*(Rf/Rsens)^2*ENBW);

demodHarmonicNoiseRms = 0.5*Vexc*attExcSignal*Hdemod/(sqrt(2));
Vnoise_out_tia_lpf = sqrt(en_pinkpluswhite2_lpf +in2_lpf + demodHarmonicNoiseRms^ +
...
    thermal_noise2_lpf + thermal_noise_Rf^2 + thermal_noise_Rsens^2);

modulated_BW_aa = 1/(2*3.14159*Raa*Caa)-excitationFrequency;
    % the -excitationFreq comes from the demodulation
    rNoise_rms = calcResistorNoise(Raa,modulated_BW_aa);
    rNoise_filtered_rms = calcResistorNoise(Raa,ENBW);

Vnoiseadc_psd2 = lsbVEq^2/(12*samplingFrequency/2);
Vnoiseadc_filtered_rms = sqrt(Vnoiseadc_psd2*ENBW);


%% SENSITIVITY AND PHASE MARGIN ANALYSIS
% LoopGain = H_tia * F_noise
% Sensitivity = 1/(1+LoopGain)
% Sensitivity_fb = feedback(1,LoopGain)
% figure;
% sensitivity_bode = bodeplot(H_cl,'b',Sensitivity_fb,'g')
% legend('Closed-Loop Gain(A)', 'System Sensitivity(S)','Location','SouthEast')
% setoptions(sensitivity_bode,'FreqUnits','Hz','XLimMode','Manual','XLim',[100 1e10])
% grid on
% legend('boxoff')

%
% figure
% margin(LoopGain)
% title('Phase marging analysis');
% setoptions(sensitivity_bode,'FreqUnits','Hz')
%
% % STEP RESPONSE ANALYSIS CLOSE-LOOP
% figure;
% stepplot(H_cl)
```

Appendix G

SiNW-bioFET Simscape model

3
analyteConcentrationSubPort
S PS
Simulink-PS Converter1
2
sinwLengthSubPort
S PS
Simulink-PS Converter2
D
S
Simscape
Zout
sinw
lengthParam
analyteC
SiNW
PS S
PS-Simulink Converter1
zSens
To Workspace
f(x) = 0
Solver Configuration
Controlled Voltage Source
C_stray
Current Sensor
PS S
PS-Simulink Converter
1
Source current
1
Drain-Source Potential
S PS
Simulink-PS Converter
Electrical Reference

```
%% Simscape model of a SiNW-bioFET as specified by De Vico et al.

component sinw
    % SiNW
    nodes
        D = foundation.electrical.electrical;
        S = foundation.electrical.electrical;
    end
    inputs
        lengthParam = {0,'m'};
        analyteC = {0,'mol/l'}
    end
    outputs
        Zout = {0,'Ohm'};
    end

    parameters (Access=private)
        pi = 3.14159265359;
        e = {1.60217662e-19, 'c'};
        Kb = {1.38064852e-23, 'm^2*kg/(s^2*K)'};
        m = {9.10938356e-31, 'kg'};
        h = {6.62607004e-34,'m^2*kg/s'};
        epsilon0 = {8.854187817e-12, 'A^2*s^4/(kg*m^3)'};
        Na = {6.0221409e23,'mol^(-1)'};
        lambdaTF = {2.04e-9, 'm'};              % Thomas-Fermi length
        tau = 0;
        tau1 = 0;
        sigma_S = {0, 'c/m^2'}
        sinwAndGatePerimeter = {0, 'm'}

    end
    parameters
        width = {400e-9,'m'};                   % SiNW width
        height = {30e-9,'m'};                   % SiNW height
        oxthickness = {5e-9, 'm'};              % Gate oxide thickness
        dop = {1.11e24,'m^(-3)'};               % Dopant concentration
        mobility = {0.0002, 'm^2/V/s'};         % Mobility
        epsilon1 = 12;                          % SiNW rel. perm.
        epsilon2 = 9.34;                         % Oxide rel. perm.
        epsilon3 = 78;                          % Solution rel. perm.
        temp = {298.15, 'K'};                   % Temperature
        l = {3e-9, 'm'};                        % Average distance charge-ox.surface
        Qc = {0, 'c'};                          % Elementary charge of sensed molecule
        funcLayerCoverage = {50,'m^(-2)'};      % Num. receptor/m^2
        approxSurfAreaTargetMol = {0,'m^2'};    % Approx. surf. covered by target molecule
        Kd = {50e-9,'mol/l'};                   % Dissociation constant analyte-receptor
        ionicStrength = {1e-12, 'mol/m^3'}      % Ionic strength
        lambdaD = {0, 'm'};                     % Debye-Huckel screening length
    end

    variables
        length = {100e-6,'m'};;                 % SiNW length
        gateSurfaceArea = {6.296e-12, 'm^2'};     % Equivalent SiNW surface area
        maxReceptorsOnGate = 0;
        adsorbedFraction = 0;                   % Analyte adsorbed fraction
        sigma_B = {1,'c/m^2'};                  % Surface charge density
        ids0 = {0,'A'};
        ids = {0, 'A'};
        Nc = 0;                                 % Number of charges bound to gate
        Zsens = {0, 'Ohm'}
    end

    function setup
        lambdaTF = sqrt(epsilon0*epsilon1*Kb*temp/(dop*e^2));
        if lambdaD == 0
            lambdaD = sqrt(epsilon0*epsilon3*Kb*temp/(2*Na*e^2*ionicStrength));
        end
        sinwAndGatePerimeter = (2*(height+oxthickness)+width+2*oxthickness);
        tau = (epsilon1*besselk(0,(value(height,'m')+value(oxthickness,'m'))/...
            value(lambdaD,'m'))*(lambdaD/lambdaTF)*besseli(1,value(height,'m')/...
            value(lambdaTF,'m')))/((besselk(0,(value(height,'m')+value(oxthickness,'m'))/...
            value(lambdaD,'m'))*(lambdaD/(height+oxthickness))+log(1+value(oxthickness,'m')/...
            value(height,'m'))*besselk(1,(value(height,'m')+value(oxthickness,'m'))/...
            value(lambdaD,'m'))*epsilon3/epsilon2)*epsilon1*(height/lambdaTF)*...
            besseli(1,value(height,'m')/value(lambdaTF,'m')) +
epsilon3*besselk(1,(value(height,'m')+...
            value(oxthickness,'m'))/value(lambdaD,'m'))*besseli(0,value(height,'m')/...
            value(lambdaTF,'m')));
        taul = 2*(height/(height+l))/(1+sqrt(height/(height+l))*exp(l/lambdaD));

    end

    branches
        ids : D.i -> S.i
    end
    equations
        length == lengthParam;
        gateSurfaceArea == sinwAndGatePerimeter*length;
        if approxSurfAreaTargetMol == 0
            maxReceptorsOnGate == funcLayerCoverage*gateSurfaceArea;
        else
            maxReceptorsOnGate == gateSurfaceArea/approxSurfAreaTargetMol;
        end
        adsorbedFraction == analyteC/(analyteC + Kd);
        Nc == maxReceptorsOnGate*adsorbedFraction;
        sigma_B == Nc*Qc/((2*(height+oxthickness)+width+2*oxthickness+l)*length);
        ids0 == (D.v - S.v)*dop*height*width*e*mobility/length;
        ids == ids0*(1-2*tau*(taul*sigma_B+sigma_S)/(height*e*dop));
        Zsens == 1/((1-2*tau*(taul*sigma_B+sigma_S)/(height*e*dop))*...
            dop*height*width*e*mobility/length);
        Zout == Zsens;
    end

end
```

Appendix H

Parametric study of SiNW-bioFET Limit Of Detection source files

```
%% analyticalParametricStudyLOD
%
% Parametric study relating SiNW-bioFET Limit Of Detection (LOD)
% to biological sample medium, SiNW gate, SiNW, and instrumentation
% parameters.

%%
% Analyte concentration vector
analyteConcentrationVect = [0 logspace(-18, 0, 38)];
% Saturation flag
sat = 1;
VhighAdc = 3.3;
s = tf('s');

% SiNW dopant concentration vector
dopantConcentrationVect = fliplr(logspace(21,24,4));

% SiNW length vector
nwLengthVect = linspace(5e-6,20e-6,16);
sinwLength = nwLengthVect(end);                  % Default scenario.

% Baseline TIA gain and feedback network parameters
Rf_baseline = 1e10;
Rf = Rf_baseline;
Cf = 1.5e-12;
Cstray = 1e-13;
Ccm = 8.8e-12;
Cdiff = 2.6e-12;
Cshunt = Ccm+Cstray+Cdiff;

tiaCutoff = 1/(2*pi*Rf*Cf);      % max 1/10 the actual tia cutoff freq.
maxExcFreq = tiaCutoff;
prevPowOf10 = floor(log10(maxExcFreq));
excFreqVect = logspace(-2,prevPowOf10+3,8*(prevPowOf10+3)+2);
vds=0.3;
adcResolution = 12;

signal_vect = zeros(length(analyteConcentrationVect),1);
noise_rms_vect = zeros(length(analyteConcentrationVect),1);
noise_rms_analytical_vect = zeros(length(analyteConcentrationVect),1);
error_fir_vect = zeros(length(analyteConcentrationVect),1);
error_ampl_vect = zeros(length(analyteConcentrationVect),1);
deltaSignal_vect = zeros(length(analyteConcentrationVect),1);

analyticalParamVectTableRowIndex = 1;

nbRows = length(excFreqVect)*length(nwLengthVect)*length(analyteConcentrationVect);
nbColumns = 13;
analyticalParamVectTable = zeros(nbRows,nbColumns);


for filter=1:6

    switch filter
        case 1
        filterCoefficients = 34;
        case 2
        filterCoefficients = 86;
        case 3
        filterCoefficients = 137;
        case 4
        filterCoefficients = 228;
        case 5
        filterCoefficients = 685;
        otherwise
        filterCoefficients = 856;
    end
    analyticalParamVect(1) = filterCoefficients;
    for dop=1:length(dopantConcentrationVect)
         dopantConcentration = dopantConcentrationVect(dop);
         analyticalParamVect(2) = dopantConcentration;

         for exc=1:length(excFreqVect)
            excitationFrequency = excFreqVect(exc);
            analyticalParamVect(3) = excitationFrequency;
            samplingFreq = 4*excitationFrequency;
            [Hfir,hdemod,enbwFIR] =
initFIRFilter(filterCoefficients,excitationFrequency,samplingFreq);
            for nw = 1:length(nwLengthVect)
                Rf = Rf_baseline;
                nwLength = nwLengthVect(nw);
                analyticalParamVect(4) = nwLength;
                % Detect saturation. If so divide gain by 10.
                while (sat == 1 && Rf>=100)
                    sat=0;
                    for conc = 1:length(analyteConcentrationVect)

                        analyteConc = analyteConcentrationVect(conc);
                        analyticalParamVect(5) = analyteConc;
                        [r_sinw(conc), sigmax, quantized_signal_vect(conc),...
                            noise_rms_analytical_vect(conc),  error_ampl_vect(conc)] =
calcAnalyticalNoiseForSettings(...
                            analyteConc,62.9e-12,-4.80652986e-19,2.4e-
9,nwLength,dopantConcentration,Rf,Cf,...
                            Cstray,Cshunt,vds,excitationFrequency,4,adcResolution,enbwFIR,hdemod);
                        if sigmax*sqrt(2)>(VhighAdc/2)
                            sat = 1;
                            Rf = Rf/10;
                            analyticalParamVectTableRowIndex = analyticalParamVectTableRowIndex-(conc-
1);

                            break
                        else
                            signal_vect(conc) = sigmax;
                        end
                        analyticalParamVect(6) = signal_vect(conc);
                        analyticalParamVect(7) = quantized_signal_vect(conc);
                        analyticalParamVect(8) = noise_rms_analytical_vect(conc);
                        analyticalParamVect(9) = r_sinw(conc);
                        analyticalParamVect(10) = error_ampl_vect(conc);

                        idealImpedanceMagnitude = vds*Rf/(signal_vect(conc)*sqrt(2));
                        recoveredMagnitude =
vds*Rf/(signal_vect(conc)*sqrt(2)+noise_rms_analytical_vect(conc)+...
                            error_fir_vect(conc));
                        % nb: we do not use the quantized_signal vector for
                        % specifying recovered magnitude because
                        % quantization noise is accounted for in
                        % noise_rms_vect
                        analyticalParamVect(11) = idealImpedanceMagnitude;
                        analyticalParamVect(12) = recoveredMagnitude;
                        analyticalParamVect(13) = Rf;
                        analyticalParamVectTable(analyticalParamVectTableRowIndex,:) =
analyticalParamVect;
                        analyticalParamVectTableRowIndex = analyticalParamVectTableRowIndex+1;

                    end
                end
                sat = 1;
            end
         end

        analyticalParamVectTableRowIndex = 1;
        filename = ['SiNW_test' num2str(filterCoefficients) '_dop' num2str(dop) '.mat'];
        save(filename,'analyticalParamVectTable');
    end
end
```

```
%% calcAnalyticalNoiseForSettings(analyteC,Kd,Qc,l,length,dop,Rfb,Cf,Cstray,Cshunt,vds,...
% excitationFreq,framesPerPeriod,adcResolution,enbwFIR,hdemod)
% computes the noise level for a given biological sample - SiNW-bioFET -
% instrumentation system

function [r_sinw, signal_rms, quantized_signal_rms, noise_rms_analytical, error_ampl] =...
    calcAnalyticalNoiseForSettings(analyteC,Kd,Qc,l,length,dop,Rfb,Cf,...
    Cstray,Cshunt,vds,excitationFreq,framesPerPeriod,adcResolution,enbwFIR,hdemod)

format long

    %% CONSTANTS
    pi = 3.14159265359;
    e = 1.60217662e-19;
    Kb = 1.38064852e-23;
    m = 9.10938356e-31;
    h = 6.62607004e-34;
    epsilon0 = 8.854187817e-12;
    Na = 6.0221409e23;
    sigma_S = 0;

    %% SINW PRE-DEFINED PARAMETER VALUES
    width = 400e-9;                         % SiNW width
    height = 30e-9;                         % SiNW height
    oxthickness = 5e-9;                     % Gate oxide thickness
    mobility = 0.0002;                        % Mobility
    epsilon1 = 12;                          % SiNW rel. perm.
    epsilon2 = 9.34;                         % Aluminum oxide rel. perm.
    epsilon3 = 78;                          % Solution rel. perm.
    temp = 298.15;                          % Temperature
    funcLayerCoverage = 65e12;              % Num. receptor/m^2
    approxSurfAreaTargetMol = 0;            % Approx. surf. covered by target molecule
    ionicStrength = 1e-12;                  % Ionic strength
    lambdaD = 0;                            % Debye-Huckel screening length
    Inoisesinw_psd = 0;

    %% SINW PARAMETRIC FUNCTIONS
lambdaTF = sqrt(epsilon0*epsilon1*Kb*temp/(dop*e^2));
    if lambdaD == 0
        lambdaD = sqrt(epsilon0*epsilon3*Kb*temp/(2*Na*e^2*ionicStrength));
    end
    sinwAndGatePerimeter = (2*(height+oxthickness)+width+2*oxthickness);
    gateSurfaceArea = sinwAndGatePerimeter*length;
    if approxSurfAreaTargetMol == 0
        maxReceptorsOnGate = funcLayerCoverage*gateSurfaceArea;
    else
        maxReceptorsOnGate = gateSurfaceArea/approxSurfAreaTargetMol;
    end

    adsorbedFraction = analyteC/(analyteC + Kd);
    Nc = maxReceptorsOnGate*adsorbedFraction;
    sigma_B = Nc*Qc/((2*(height+oxthickness)+width+2*oxthickness+l)*length);
    tau = (epsilon1*besselk(0,(height+oxthickness)/lambdaD)*(lambdaD/lambdaTF)*...
        besseli(1,height/lambdaTF))/((besselk(0,(height+oxthickness)/lambdaD)*...
                (lambdaD/(height+oxthickness))+log(1+oxthickness/height)*...
                besselk(1,(height+oxthickness)/lambdaD)*epsilon3/epsilon2)*epsilon1*(height/lambdaTF)*...
                besseli(1,height/lambdaTF) + epsilon3*besselk(1,(height+oxthickness)/lambdaD)*...
                besseli(0,height/lambdaTF));
    taul = 2*(height/(height+l))/(1+sqrt(height/(height+l))*exp(l/lambdaD));
    Rsinw = 1/((1-2*tau*(taul*sigma_B+sigma_S)/(height*e*dop))*dop*height*width*e*mobility/length);
    r_sinw = Rsinw;

    %% TIA NOISE
    % Opamp parameters
    Inoiseamp_psd2 = (1e-14)^2;         % A^2/Hz
    Vnoiseamp_psd = 8e-9;
    GBW = 9e6;
    R_ref = 1e5;
    Inoisetia_psd = Inoisesinw_psd+sqrt(Inoiseamp_psd2);
    [~, Vnoisetiaout_modulated_filtered_rms] = calcTiaOutputNoiseRms(Cstray,Cshunt,Rfb,Cf,R_ref,GBW,...
            Inoisetia_psd,Vnoiseamp_psd,Rsinw,excitationFreq,enbwFIR,hdemod,vds);

    %% ANTI-ALIASING FILTER NOISE
    Raa = 1e3;
    Caa = 1e-9;
    % Antia-alias filter noise bandwidth after modulation:
    rNoise_filtered_rms = calcResistorNoise(Raa,enbwFIR);

%     The aliased noise could be taken into consideration:
%     samplingFreq = excitationFreq*framesPerPeriod;
%     G1=calcGainMagnitude(accuracy,aaFilterOrder,enbwAA,samplingFreq);
%     aliasedNoiseAA_rms = G1*Vnoisetiaout_psd * (samplingFreq/enbwAA);
%     Vnoiseaadirect_rms=sqrt(enbwAA)*Vnoisetiaout_psd;

%   Vnoiseaaout_rms = sqrt(Vnoiseaa_modulated_rms^2 + Vnoisetiaout_modulated_rms^2);
    Vnoiseaa_modulated_filtered_rms = rNoise_filtered_rms;
    Vnoiseaaout_filtered_rms = sqrt(Vnoiseaa_modulated_filtered_rms^2 + Vnoisetiaout_modulated_filtered_rms^2);

    %% ADC NOISE
    VhighAdc = 3.3;
    samplingFreq = excitationFreq*framesPerPeriod;
    lsbVEq = VhighAdc/(2^adcResolution);
    Vnoiseadc_psd2 = lsbVEq^2/(12*samplingFreq/2);
    Vnoiseadc_filtered_rms = sqrt(Vnoiseadc_psd2*enbwFIR);
    Vnoiseadcout_filtered_rms = sqrt(Vnoiseaaout_filtered_rms^2 + Vnoiseadc_filtered_rms^2);

    %% FIR NOISE FILTERING
    % SIGNAL
    wexc = 2*pi*excitationFreq;
    s = tf('s');
    H_sensor = 1/(1 + s*Rsinw*Cstray);
    attSensor = bode(H_sensor,wexc);
    H_signal = attSensor*(Rfb/Rsinw)/(1+s*Rfb*Cf);
    attExcSignal = bode(H_signal,wexc);
    signalTiaOut = vds*attExcSignal;
    H_aa = 1/(1 + s*Raa*Caa);
    signalAdcIn = signalTiaOut * bode(H_aa,wexc);
    signalAdcOut = signalAdcIn + Vnoiseadc_filtered_rms;
    signal_rms = signalAdcOut/sqrt(2);
    quantized_signal = round((signalAdcIn/VhighAdc)*(2^adcResolution)) * VhighAdc/(2^adcResolution);
    quantized_signal_rms = quantized_signal/sqrt(2);

    % TOTAL OUTPUT NOISE
    noise_rms_analytical = Vnoiseadcout_filtered_rms;

    % ERROR
    error_ampl = abs(signalAdcOut-(vds/Rsinw)*Rfb);

end
```

```
%% calcTiaOutputNoiseRms(Cstr,Cshunt,Rf,Cf,R_ref,GBW,I_noise,V_noise,Rsinw_baseline,...
% excitationFreq,enbwFIR,hdemodFIR,Vexc)
% Computes TIA noise

%%
function [Vnoise_out_tia, Vnoise_out_tia_lpf] = calcTiaOutputNoiseRms(Cstr,Cshunt,...
    Rf,Cf,R_ref,GBW,I_noise,V_noise,Rsinw_baseline,...
    excitationFreq,enbwFIR,hdemodFIR,Vexc)

I_noise2 = I_noise^2;           % A^2/Hz
Fz = 1/(2*pi*Rf*(Cf+Cshunt));
Fp = 1/(2*pi*Rf*Cf);
F0 = sqrt(GBW * Fz);
Q = F0/(GBW*(Fz/Fp)+Fz);
Kb = 1.38064852e-23;        % Boltzmann constant

%% Noise gain profile
noiseGainLow = 1+Rf/Rsinw_baseline;
noiseGainCutoff = 1/(2*pi*((Rf*Rsinw_baseline/(Rf+Rsinw_baseline))*(Cf+Cshunt)));
noiseGainHigh = 1+(Cshunt/Cf);

%% NOISE BANDWIDTH OF THE MODULATED SIGNAL:
tiaCutoff = 1/(2*pi*Rf*Cf);
K2 = ((29.8e-9)^2-V_noise^2)*10;     % opamp psd at 10Hz is 29.8nV/sqrt(Hz)
flickerNoiseCutoff = K2/(V_noise^2);
noiseGainBandwidth = sqrt(9e6/(2*pi*Rf*Cshunt));

%% NOISE PSD INPUT NOISE FROM TIA :
if excitationFreq < flickerNoiseCutoff
    en_pinkpluswhite2_low = (V_noise^2)*noiseGainLow^2*(flickerNoiseCutoff*...
        log(flickerNoiseCutoff/excitationFreq)+(flickerNoiseCutoff-excitationFreq));
    en_pinkpluswhite2_cutoff = (V_noise^2)*((1/noiseGainCutoff)*(1+Rf/Rsinw_baseline))^2*...
        ((tiaCutoff)/3 - (noiseGainCutoff)/3);
    en_pinkpluswhite2_high = (V_noise^2)*(noiseGainHigh^2)*(1.54*noiseGainBandwidth-tiaCutoff);
elseif excitationFreq < noiseGainCutoff
    en_pinkpluswhite2_low = (V_noise^2)*noiseGainLow^2*((noiseGainCutoff-excitationFreq));
    en_pinkpluswhite2_cutoff = (V_noise^2)*((1/noiseGainCutoff)*(1+Rf/Rsinw_baseline))^2*...
        ((tiaCutoff)/3 - (noiseGainCutoff)/3);
    en_pinkpluswhite2_high = (V_noise^2)*(noiseGainHigh^2)*(1.54*noiseGainBandwidth-tiaCutoff);
elseif excitationFreq < tiaCutoff
    en_pinkpluswhite2_low = 0;
    en_pinkpluswhite2_cutoff = (V_noise^2)*((1/noiseGainCutoff)*(1+Rf/Rsinw_baseline))^2*...
        ((tiaCutoff)/3 - (excitationFreq)/3);
    en_pinkpluswhite2_high = (V_noise^2)*(noiseGainHigh^2)*(1.54*noiseGainBandwidth-tiaCutoff);
else
    en_pinkpluswhite2_low = 0;
    en_pinkpluswhite2_cutoff = 0;
    en_pinkpluswhite2_high = (V_noise^2)*(noiseGainHigh^2)*(1.54*noiseGainBandwidth-excitationFreq);
end
en_pinkpluswhite2 = en_pinkpluswhite2_low + en_pinkpluswhite2_cutoff + en_pinkpluswhite2_high;


%% RMS OUTPUT NOISE AT TIA OUTPUT
if excitationFreq < noiseGainCutoff
    in2_low = I_noise2*(Rf^2 + R_ref^2*noiseGainLow^2)*(noiseGainCutoff-excitationFreq);
    in2_cutoff = I_noise2*(Rf^2 + R_ref^2*((1/noiseGainCutoff)*(1+Rf/Rsinw_baseline))^2)*...
        ((tiaCutoff)/3 - (noiseGainCutoff)/3);
    in2_high =  I_noise2*(Rf^2 + R_ref^2*noiseGainHigh^2)*(1.54*noiseGainBandwidth-tiaCutoff);
    thermal_noise2_low = 4*Kb*298*(R_ref*noiseGainLow^2)*(noiseGainCutoff-excitationFreq);
    thermal_noise2_cutoff = 4*Kb*298*(R_ref*((1/noiseGainCutoff)*(1+Rf/Rsinw_baseline))^2)*...
        ((tiaCutoff)/3 - (noiseGainCutoff)/3);
    thermal_noise2_high = 4*Kb*298*(R_ref*noiseGainHigh^2)*(1.54*noiseGainBandwidth-tiaCutoff);

elseif excitationFreq < tiaCutoff
    in2_low = 0;
    in2_cutoff = I_noise2*(Rf^2 + R_ref^2*((1/noiseGainCutoff)*(1+Rf/Rsinw_baseline))^2)*...
        ((tiaCutoff)/3 - (excitationFreq)/3);
    in2_high =  I_noise2*(Rf^2 + R_ref^2*noiseGainHigh^2)*(1.54*noiseGainBandwidth-tiaCutoff);
    thermal_noise2_low = 0;
    thermal_noise2_cutoff = 4*Kb*298*(R_ref*((1/noiseGainCutoff)*(1+Rf/Rsinw_baseline))^2)*...
        ((tiaCutoff)/3 - (excitationFreq)/3);
    thermal_noise2_high = 4*Kb*298*(R_ref*noiseGainHigh^2)*(1.54*noiseGainBandwidth-tiaCutoff);
else
    in2_low = 0;
    in2_cutoff = 0;
    in2_high =  I_noise2*(Rf^2 + R_ref^2*noiseGainHigh^2)*(1.54*noiseGainBandwidth-excitationFreq);
    thermal_noise2_low = 0;
    thermal_noise2_cutoff = 0;
    thermal_noise2_high = 4*Kb*298*(R_ref*noiseGainHigh^2)*(1.54*noiseGainBandwidth-excitationFreq);
end
thermal_noise_Rf = sqrt(4*Kb*298*Rf*tiaCutoff*1.57);
thermal_noise_Rsens = sqrt(4*Kb*298*Rsinw_baseline*(Rf/Rsinw_baseline)^2*tiaCutoff*1.57);

in2 = in2_low + in2_cutoff + in2_high;
thermal_noise2 = thermal_noise2_low + thermal_noise2_cutoff + thermal_noise2_high +
thermal_noise_Rf^2 + thermal_noise_Rsens^2;

Vnoise_out_tia = sqrt(en_pinkpluswhite2 +in2 + thermal_noise2);

%% RMS OUTPUT NOISE AT FIR INPUT (AFTER DEMODULATION - ENBW OVER DEMODULATED SPECTRUM)
wexc = 2*pi*excitationFreq;
s = tf('s');
H_sensor = 1/(1 + s*Rsinw_baseline*Cstr);
attSensor = bode(H_sensor,wexc);
H_signal = attSensor*(Rf/Rsinw_baseline)/(1+s*Rf*Cf);
attExcSignal = bode(H_signal,wexc);

if excitationFreq < flickerNoiseCutoff
    en_pinkpluswhite2_lpf = (V_noise^2)*noiseGainLow^2*(flickerNoiseCutoff*...
        log((excitationFreq+enbwFIR)/excitationFreq) + enbwFIR);
    in2_lpf = I_noise2*(Rf^2 + R_ref^2*noiseGainLow^2)*enbwFIR;
    thermal_noise2_lpf = 4*Kb*298*(R_ref*noiseGainLow^2)*enbwFIR;
elseif excitationFreq < noiseGainCutoff
    if excitationFreq + enbwFIR < noiseGainCutoff
        en_pinkpluswhite2_lpf = (V_noise^2)*noiseGainLow^2*enbwFIR;
        in2_lpf = I_noise2*(Rf^2 + R_ref^2*noiseGainLow^2)*enbwFIR;
        thermal_noise2_lpf = 4*Kb*298*(R_ref*noiseGainLow^2)*enbwFIR;
    else
                en_pinkpluswhite2_lpf = (V_noise^2)*noiseGainLow^2*(noiseGainCutoff-excitationFreq)
                +...
                (V_noise^2)*((1/noiseGainCutoff)*(1+Rf/Rsinw_baseline))^2*(((excitationFreq+enbwFIR))/
                3 -...
                (noiseGainCutoff)/3);
                        in2_lpf = I_noise2*(Rf^2 + R_ref^2*noiseGainLow^2)*(noiseGainCutoff-
                excitationFreq) +...
                I_noise2*(Rf^2 + R_ref^2*((1/noiseGainCutoff)*(1+Rf/Rsinw_baseline))^2)*...
            (((excitationFreq+enbwFIR))/3 - (noiseGainCutoff)/3);
                thermal_noise2_lpf = 4*Kb*298*(R_ref*noiseGainLow^2)*enbwFIR +...

                4*Kb*298*(R_ref*((1/noiseGainCutoff)*(1+Rf/Rsinw_baseline))^2)*(((excitationFreq+enbwF
                IR))/3 -...
            (noiseGainCutoff)/3);
    end
        elseif excitationFreq < tiaCutoff
    if excitationFreq + enbwFIR < tiaCutoff
                en_pinkpluswhite2_lpf = (V_noise^2)*((1/noiseGainCutoff)*(1+Rf/Rsinw_baseline))^2*...
             (((excitationFreq+enbwFIR))/3 - (excitationFreq)/3);
                in2_lpf = I_noise2*(Rf^2 + R_ref^2*((1/noiseGainCutoff)*(1+Rf/Rsinw_baseline))^2)*...
            (((excitationFreq+enbwFIR))/3 - (excitationFreq)/3);
                thermal_noise2_lpf =
                4*Kb*298*(R_ref*((1/noiseGainCutoff)*(1+Rf/Rsinw_baseline))^2)*...
            (((excitationFreq+enbwFIR))/3 - (noiseGainCutoff)/3);
```

```
        else
              en_pinkpluswhite2_lpf = (V_noise^2)*((1/noiseGainCutoff)*(1+Rf/Rsinw_baseline))^2*...
           ((tiaCutoff)/3 - (excitationFreq)/3) +...
           (V_noise^2)*(noiseGainHigh^2)*(excitationFreq+enbwFIR-tiaCutoff);
              in2_lpf = I_noise2*(Rf^2 + R_ref^2*((1/noiseGainCutoff)*(1+Rf/Rsinw_baseline))^2)*...
           ((tiaCutoff)/3 - (excitationFreq)/3) + I_noise2*(Rf^2 +...
           R_ref^2*(noiseGainHigh^2))*(excitationFreq+enbwFIR-tiaCutoff);
              thermal_noise2_lpf =
              4*Kb*298*(R_ref*((1/noiseGainCutoff)*(1+Rf/Rsinw_baseline))^2)*...
           (((excitationFreq+enbwFIR))/3 - (noiseGainCutoff)/3) +...
           4*Kb*298*(R_ref*noiseGainHigh^2)*(excitationFreq+enbwFIR-tiaCutoff);

    end
else
    en_pinkpluswhite2_lpf = (V_noise^2)*(noiseGainHigh^2)*enbwFIR;
    in2_lpf = I_noise2*(Rf^2 + R_ref^2*noiseGainHigh^2)*enbwFIR;
    thermal_noise2_lpf =  4*Kb*298*(R_ref*noiseGainHigh^2)*enbwFIR;
end

thermal_noise_Rf = sqrt(4*Kb*298*Rf*enbwFIR);
thermal_noise_Rsens = sqrt(4*Kb*298*Rsinw_baseline*(Rf/Rsinw_baseline)^2*enbwFIR);
demodHarmonicNoiseRms = 0.5*Vexc*attExcSignal*hdemodFIR/(sqrt(2));
Vnoise_out_tia_lpf = sqrt(en_pinkpluswhite2_lpf +in2_lpf + demodHarmonicNoiseRms^2 + ...
thermal_noise2_lpf + thermal_noise_Rf^2 + thermal_noise_Rsens^2);

end
```

Appendix I

POCLowLevelTask source file

```
//
//  POCComplexLowLevelTask.m
//  PoCAppBLEv2
//
//  Created by Francois Patou on 06/01/15.
//  Copyright (c) 2015 Francois Patou. All rights reserved.
//

#import "POCLowLevelTask.h"
#import "POCCyclicVoltametryAcquisition.h"

@implementation POCLowLevelTask

static NSString * cmplxLLTaskStringId = @"CPLX_LLT";

+ (NSString *)lowLevelTaskStringId{
    return cmplxLLTaskStringId;
}



- (id)init{
    self = [super init];
    if(self){
        _sequentialSubtasksArray = [[NSMutableArray alloc] init];
    }
    return self;
}



// LINK SUBTASKS
- (void)linkSubtasks{

    for(NSUInteger subtaskIndex= 0; subtaskIndex < self.
        sequentialSubtasksArray.count; subtaskIndex++) {
        POCElementaryTask *subtask = [[self sequentialSubtasksArray]
            objectAtIndex:subtaskIndex];
        // Recursive call to linkSubtasks in case the subtask would itself be
            a POCComplexLowLevelTask.

        subtask.parentTask = self;

            // SUBTASK IS A COMPLEX TASK
        if ([subtask isKindOfClass:[POCLowLevelTask class]]) {
            POCLowLevelTask *complexSubtask = (POCLowLevelTask *)subtask;
            [complexSubtask linkSubtasks];  // Recursive call to
                linkSubtasks.. All subtasks are linked down the tree.
            if (subtaskIndex > 0) {
                POCElementaryTask *previousSubtask = [_sequentialSubtasksArray
                    objectAtIndex:(subtaskIndex-1)];
                if ([previousSubtask isKindOfClass:[POCLowLevelTask class]]) {
                    POCLowLevelTask *complexPreviousSubtask = (POCLowLevelTask
                        *)previousSubtask;
                    [complexSubtask.sequentialSubtasksArray.firstObject
                        setPreviousTask:complexPreviousSubtask.
                        sequentialSubtasksArray.lastObject];
                } else {    // Subtask is a simple POCLowLevelTask
                    [complexSubtask.sequentialSubtasksArray.firstObject
                        setPreviousTask:previousSubtask];
                }
            }
            if (subtaskIndex < self.sequentialSubtasksArray.count-1) {
                POCElementaryTask *followingSubtask =
                    [_sequentialSubtasksArray objectAtIndex:(subtaskIndex+1)];
                if ([followingSubtask isKindOfClass:[POCLowLevelTask class]])
                    {
                    POCLowLevelTask *complexFollowingSubtask =
                        (POCLowLevelTask *)followingSubtask;
                    [complexSubtask.sequentialSubtasksArray.lastObject
                        setFollowingTask:complexFollowingSubtask.
                        sequentialSubtasksArray.firstObject];
                } else {    // Subtask is a simple POCLowLevelTask
                    [complexSubtask.sequentialSubtasksArray.lastObject
                        setFollowingTask:followingSubtask];
                }
            } else if (subtaskIndex == self.sequentialSubtasksArray.count-1){
                [complexSubtask.sequentialSubtasksArray.lastObject
                    setFollowingTask:self.followingTask];
            }

            // SUBTASK IS NOT A COMPLEX TASK
        } else {
            if (subtaskIndex > 0) {
                POCElementaryTask *previousSubtask = [_sequentialSubtasksArray
                    objectAtIndex:(subtaskIndex-1)];
                if ([previousSubtask isKindOfClass:[POCLowLevelTask class]]) {
                    POCLowLevelTask *complexPreviousSubtask = (POCLowLevelTask
                        *)previousSubtask;
                    [subtask setPreviousTask:complexPreviousSubtask.
                        sequentialSubtasksArray.lastObject];
                } else {    // Subtask is a simple POCLowLevelTask
                    [subtask setPreviousTask:previousSubtask];
                }
            }
            if (subtaskIndex < self.sequentialSubtasksArray.count-1) {
                POCElementaryTask *followingSubtask =
                    [_sequentialSubtasksArray objectAtIndex:(subtaskIndex+1)];
                if ([followingSubtask isKindOfClass:[POCLowLevelTask class]])
                    {
                    POCLowLevelTask *complexFollowingSubtask =
                        (POCLowLevelTask *)followingSubtask;
                    [subtask setFollowingTask:complexFollowingSubtask.
                        sequentialSubtasksArray.firstObject];
                } else {    // Subtask is a simple POCLowLevelTask
                    [subtask setFollowingTask:followingSubtask];
                }
            }
            else if (subtaskIndex == self.sequentialSubtasksArray.count-1){
                [subtask setFollowingTask:self.followingTask];
            }
        }
    }
    [[_sequentialSubtasksArray lastObject] setTaskIsLastOfASetOfSubtasks:YES];
    [self setFollowingTask:[[self.sequentialSubtasksArray lastObject]
        followingTask]];
}



// OVERRIDING THE EXECUTION OF SIMPLE LOW-LEVEL TASKS
- (void)execute{
    if (self.taskIsUnderExecution == NO) {
        self.taskIsUnderExecution = YES;
    }
    [self linkSubtasks];
    [_sequentialSubtasksArray.firstObject execute];
}

// OVERRINDING SET_LOW_LEVEL_TASK_DELEGATE METHOD FOR ALL SUBTASKS
- (void)setElementaryTaskExecutionDelegate:(id<
    POCElementaryTaskExecutionDelegate>)lowLevelTaskExecutionDelegate{
    [super setElementaryTaskExecutionDelegate:lowLevelTaskExecutionDelegate];
    for (POCElementaryTask *subTask in _sequentialSubtasksArray) {
        [subTask setElementaryTaskExecutionDelegate:
            lowLevelTaskExecutionDelegate];
    }
    // Should propagate to subTasks sustasks array, in case the subtasks are
        themselves POCComplexLowLevelTasks.
}

- (id)copyWithZone:(NSZone *)zone{
    id copy = [super copyWithZone:zone];
    if (copy){
        [copy setSequentialSubtasksArray:[[NSMutableArray alloc] init]];
        // IDEALLY WE NEED TO CREATE A DEEP-COPY OF THE SEQUENTIAL SUBTASK
            ARRAY. THIS REQUIRES
        // TO HAVE ALL OBJECTS ABIDE BY THE NSCODING PROTOCOL. WE WILL THUS
            FOR NOW RECREATE THE SUBTASK ARRAY IN ALL
        // COMPLEXTASKS CHILD CLASSES COPY METHOD.
    }
    return copy;
}

@end
```

Appendix J

POCEncoder source file

```
//
//  POCHighLevelSoftwareCompiler.m
//  PoCAppBLEv2
//
//  Created by Francois Patou on 17/12/14.
//  Copyright (c) 2014 Francois Patou. All rights reserved.
//

#import "POCEncoder.h"
#import "POCTasksCustomClasses.h"
#import "POCAccessoryConstants.h"
#import "POCFIRFilter.h"
#import "POCElementaryTask.h"
#import "POCLoCProgram.h"
#import "POCDefines.h"


@implementation POCEncoder

static POCEncoder *pocHighLevelSoftwareCompiler = nil;


// HIGH-LEVEL SOFTWARE ENCODER SINGLETON
+ (POCEncoder *)sharedEncoder{
    if (pocHighLevelSoftwareCompiler == nil) {
        pocHighLevelSoftwareCompiler = [[super alloc] init];
    }
    return pocHighLevelSoftwareCompiler;
}

// ENCODING METHODS

#pragma mark - ENCODE LOW-LEVEL TASK

+ (NSArray *)encodeElementaryTask:(POCElementaryTask *)task{

    NSMutableArray *instructionAppendedSequence = [[NSMutableArray alloc] init
        ];
    if(task != nil){

            // ****************************************************
            //  LOW-LEVEL/COMPLEX-LOW-LEVEL TASK COMPILATION
            // ****************************************************

                uint8_t instructionByteArray
                    [HIGH_LEVEL_COMPILER_INSTRUCTION_SET_BYTE_LENGTH] = {0x00};
                NSData * instruction = [[NSData alloc] init];

                // ***********************
                // VOLTAGE EXCITATION
                // ***********************
                if ([task isKindOfClass:[POCWaveformGeneration class]]){
                    POCWaveformGeneration *voltExc = (POCWaveformGeneration *)
                        task;

                    NSMutableArray *excitationInstructions = [[NSMutableArray
                        alloc] init];

                    // OUTPUT TERMINAL
                    uint16_t outTerm = [[[voltExc excitationTerminal]
                        valueForKey:@"Port"] unsignedShortValue];
                    // CHANNEL MODE -- Theoretically, in order to be
                        consistent, the mode should be retrieve from taskMode
                        for the POCVoltageExcitation it is the
                        POCVoltageExcitation class that
                    // should be responsible to make sure that its taskMode
                        property is updated as a function of its waveform
                        subclass.
                    uint8_t chMode = 0x00;
                    uint16_t dutyCycleOn = 50;
                    uint8_t electrochemistryOn = 0x00;
                    uint16_t nbExcPeriods = voltExc.nbExcitationPeriods.
                        unsignedShortValue;
                    uint16_t startupTimeInMilliSec = 0x00;

                    // AMPLITUDE
                    float voltAmpl = 0.0f;
                    if (voltExc.waveform.amplitudeInVolts != nil) {
                        voltAmpl = voltExc.waveform.amplitudeInVolts.
                            floatValue;
                        // Deprecated :
                        // voltAmpl = (uint16_t)((([voltExc.waveform
                            amplitudeInVolts] floatValue] / (float)DAC_SPAN) *
                            (int)NB_DAC_POINT_FULL_RESOLUTION);
                    }
                    // EXC_FREQUENCY
                    float excFreq = 0.0f;
                    if (voltExc.waveform.frequencyInHertz != nil) {
                        excFreq = [[voltExc.waveform frequencyInHertz]
                            floatValue];
                    }
                    // OFFSET VOLTAGE / RAMP START VOLTAGE
                    float offsetVolt = 0.0f;
                    if (voltExc.dcOffset != nil){
                        offsetVolt = voltExc.dcOffset.floatValue;
                        // Deprecated :
                        // (uint16_t)([[voltExc dcOffset] floatValue] *
                            (float)NB_DAC_POINT_FULL_RESOLUTION /
                            (float)DAC_SPAN) + (int)VREF_DAC_CODE;
                    }
                    // RAMP INCREMENT VOLTAGE
                    float rampIncrement = 0.0f;
                    if (voltExc.waveform.rampIncrement != nil) {
                        rampIncrement = voltExc.waveform.rampIncrement.
                            floatValue;
                    }
                    //  INIT VOLTAGE
                    float initialVoltage = 0.0f;
                    if (voltExc.waveform.initialVoltage != nil) {
                        initialVoltage = voltExc.waveform.initialVoltage.
                            floatValue;
                    } else {
                        initialVoltage = voltExc.dcOffset.floatValue;
                    }
                    //  FINAL VOLTAGE
                    float finalVoltage = 0.0f;
                    if (voltExc.waveform.finalVoltage != nil) {
                        finalVoltage = voltExc.waveform.finalVoltage.
                            floatValue;
                    } else {
                        finalVoltage = voltExc.dcOffset.floatValue;
                    }
                    if (voltExc.startupTime != nil) {
                        startupTimeInMilliSec = voltExc.startupTime.
                            unsignedShortValue;
                    }

                    // ******* ADVANCED WAVEFORM SETTINGS *******
                    // We pass the ADVANCED_WAVEFORM_SETTINGS opcode to the
                        HWA
                    // The advanced settings are only transmitted for NON
                        DC_EXC_MODE

                    if ([voltExc.waveform isKindOfClass:[POCPulseWaveform
                        class]]){
                        POCPulseWaveform *pulseWaveform = (POCPulseWaveform *)
                            voltExc.waveform;
                        if (pulseWaveform.dutyHigh != nil){
                            dutyCycleOn = pulseWaveform.dutyHigh.
                                unsignedIntegerValue;  }
                    }
                    //if (((chMode >> 2) & 0x0F) != DC_EXC_MODE) {

                        chMode = (ADVANCED_WAVEFORM_SETTINGS << 2) | (0x03 &
                            TASK_HOLD);

                        // ENCODE ADVANCED WAVEFORM INSTRUCTION 2
                        memcpy(&instructionByteArray[0], &outTerm,1);
                                // Byte  0      - output terminal #ID
                        memcpy(&instructionByteArray[1], &chMode, 1);
                                // Bytes 1      - channel Mode | Task status
                        memcpy(&instructionByteArray[2], &rampIncrement, 4);
                                // Bytes 2-5    - ramp increment voltage
                            (float)
                        memcpy(&instructionByteArray[6], &initialVoltage, 4);
                                // Bytes 6-9    - initial potential
                        memcpy(&instructionByteArray[10], &finalVoltage, 4);
                                // Bytes 10-13  - final potential
                        memcpy(&instructionByteArray[14], &dutyCycleOn, 2);
                                // Bytes 14-15  - duty cycle high (only for
                            square-wave excitations).
                        memcpy(&instructionByteArray[16], &
                            startupTimeInMilliSec, 2);      // Bytes 16-17  -
                            startup Time excitations).

                        instruction = [NSData dataWithBytes:
                            instructionByteArray length:
                            HIGH_LEVEL_COMPILER_INSTRUCTION_SET_BYTE_LENGTH];
                        NSLog(@"instruction : %@ appended to
                            instructionSequence", instruction);
                         [excitationInstructions addObject:instruction];
                         [instructionAppendedSequence addObject:instruction];
                    //}


                    // We do not TASK_HOLD anymore. If the voltage excitation
                        should be trigger TASK_START can be transmitted.
                    // ELECTROCHEMISTRY ON
                    chMode = 0x00;

                    if(voltExc.referenceElectrodeOn == TRUE){
                        electrochemistryOn = ELECTROCHEMISTRY_ON;};
                    chMode = chMode | electrochemistryOn;

                    if ([voltExc.waveform isKindOfClass:[POCPulseWaveform
                        class]]){
                        chMode = chMode | (SQUARE_WAVE_EXC_MODE << 2) | (0x03
                            & voltExc.taskStatus);                          //
                            Sine-Wave generator (2)
                    } else if ([voltExc.waveform isKindOfClass:
                        [POCSineWaveform class]]){
                        chMode = chMode | (SINE_WAVE_EXC_MODE << 2) | (0x03 &
                            voltExc.taskStatus);
                    } else if ([voltExc.waveform isKindOfClass:
                        [POCTriangularWaveform class]]){
                        chMode = chMode | (TRIANGULAR_EXC_MODE << 2) | (0x03 &
                            voltExc.taskStatus);
                    } else {    // If voltExc.waveform class is none of the
                        POCWaveform subclass then it corresponds to a constant
                        potential.
                        chMode = chMode | (DC_EXC_MODE << 2) | (0x03 & voltExc
                            .taskStatus);
                        dutyCycleOn = 0;
                    }


                    // ENCODE BASIC WAVEFORM INSTRUCTION 1
                    memcpy(&instructionByteArray[0], &outTerm,1);           //
                        Byte  0     - output terminal #ID
                    memcpy(&instructionByteArray[1], &chMode, 1);           //
                        Bytes 1     - channel Mode | Task status
                    memcpy(&instructionByteArray[2], &nbExcPeriods, 2);
                    memcpy(&instructionByteArray[4], &voltAmpl, 4);         //
                        Bytes 3:4   - amplitude
                    memcpy(&instructionByteArray[8], &excFreq, 4);          //
                        Byte  5:8   - frequency
                    memcpy(&instructionByteArray[12], &offsetVolt, 4);      //
                        Bytes 9:10   - offset voltage

                    instruction = [NSData dataWithBytes:instructionByteArray
                        length:HIGH_LEVEL_COMPILER_INSTRUCTION_SET_BYTE_LENGTH]
                        ;
                    NSLog(@"instruction : %@ appended to instructionSequence",
                        instruction);
                    [excitationInstructions addObject:instruction];
                    [instructionAppendedSequence addObject:instruction];

                    task.lowLevelTaskInstructions = [NSArray arrayWithArray:
                        excitationInstructions];
                }

                // ***********************
                // FILTER + DECIMATION CURRENT ACQUISITION
                // ***********************
                else if ([task isKindOfClass:[POCDSPCurrentReadoutAcquisition
                    class]]) {
                    POCDSPCurrentReadoutAcquisition *decimFiltCurrAcq =
                        (POCDSPCurrentReadoutAcquisition *)task;
                    // INPUT TERMINAL
                    uint16_t inTerm = [[[decimFiltCurrAcq acquisitionTerminal]
                        valueForKey:@"Port"] unsignedShortValue];
                    // SYNCHRONICITY
                    uint8_t synchronous = 0x00;
                    uint8_t pairedExcCh = 0xFF;
                    if(decimFiltCurrAcq.isSynchronous == TRUE){
                        synchronous = SYNCHRONOUS_ACQUISITION_ON;
                        pairedExcCh = [[decimFiltCurrAcq.
                            pairedWaveformGeneration.excitationTerminal
                            valueForKey:@"Port"] unsignedCharValue];
                    }
                    // CHANNEL MODE
                    uint8_t chMode = 0x00;
                    chMode = chMode | synchronous | (decimFiltCurrAcq.taskMode
                        << 2) | (0x03 & decimFiltCurrAcq.taskStatus);
                            // Default chMode value is 0xFF
                    // NUMBER OF ACQUISITION POINTS
                    uint16_t nbAcqPoints = [decimFiltCurrAcq.
                        nbAcquisitionPoints unsignedIntValue];
                    // SAMPLING FREQUENCY
                    float samplFreq = [decimFiltCurrAcq.samplingRate
```

```
        floatValue];
    // GAIN
    uint8_t currentGain = 0x00;
    NSUInteger index =[[[POCAccessoryConstants
        sharedAccessoryConstants]
        currentAmplifiersSensitivities] indexOfObject:
        decimFiltCurrAcq.sensitivity];
    currentGain = 0x0F & index;
    // FILTER
    uint8_t filterSelection = NO_FILTERING;
    for (POCFIRFilter *filter in [[POCAccessoryConstants
        sharedAccessoryConstants] firFilters]) {
        if ([decimFiltCurrAcq.lpf isEqual:filter]) {
            filterSelection = 0x000F &
                [[[POCAccessoryConstants
                sharedAccessoryConstants] firFilters]
                indexOfObject:filter];
            break;
        }
    }
    // GAIN-FILTER SELECTION BYTE
    uint16_t gainFilterSelection = 0x00FF & ((filterSelection
        << 4) | (0x0F & currentGain));
    // DECIMATION FACTOR : set by default to 0xFF if no-
        decimation should be applied.
    uint16_t decimFactor = NO_DECIMATION;
    for (NSNumber *decFact in [[POCAccessoryConstants
        sharedAccessoryConstants] decimationFactors]) {
        if ([decimFiltCurrAcq.decimationFactor isEqual:decFact
            ]) {
            decimFactor = 0x00FF & [[[POCAccessoryConstants
                sharedAccessoryConstants] decimationFactors]
                indexOfObject:decFact];
            break;
        }
    }

    uint16_t samplingWindow = 0x0000;
    samplingWindow = decimFiltCurrAcq.samplingWindow.
        unsignedIntegerValue;
    uint16_t windowingMode = 0x0000;
    uint8_t windowAtEndOfCycle = 0x00;
    if (decimFiltCurrAcq.samplingWindowAtEndOfCycle == YES) {
        windowAtEndOfCycle = 0x01;
    } else windowAtEndOfCycle = 0x00;
    uint8_t windowingPerHalfCycle = 0x00;
    if (decimFiltCurrAcq.samplingWindowPerHalfCycle == YES) {
        windowingPerHalfCycle = 0x01;
    } else windowingPerHalfCycle = 0x00;
    windowingMode = 0x00FF & ((windowAtEndOfCycle << 4) |
        (0x0F & windowingPerHalfCycle));

    // DATA OUTPUT
    uint8_t currentAcqOutput = 0x01;
                // 0x00 code : unprocessed data.
    // AVERAGED SAMPLING
    uint8_t avgSamples = 0x01;
    avgSamples = decimFiltCurrAcq.nbAvgSamples.intValue;

    // ENCODE INSTRUCTION
    memcpy(&instructionByteArray[0], &inTerm,1);
                // Byte  0      - input terminal #ID
    memcpy(&instructionByteArray[1], &chMode, 1);
                // Bytes 1      - channel Mode
    memcpy(&instructionByteArray[2], &pairedExcCh, 1);
                // Byte  2      - default paired channel : 0xFF
    memcpy(&instructionByteArray[3], &nbAcqPoints, 2);
                // Bytes 3      - Number of acquisition points
    memcpy(&instructionByteArray[5], &samplFreq, 4);
                // Bytes 5:8    - sampling frequency
    memcpy(&instructionByteArray[11], &gainFilterSelection, 1)
        ;       // Byte  11     - current gain #index
    memcpy(&instructionByteArray[12], &decimFactor, 1);
                // Bytes 12     - decimation factor
    memcpy(&instructionByteArray[13], &currentAcqOutput, 1);
                // Bytes 16     - output data transmission mode
    memcpy(&instructionByteArray[14], &samplingWindow, 1);
                // Bytes 16     - sampling window (in %)
    memcpy(&instructionByteArray[15], &windowingMode, 1);
                // Bytes 16     - sampling window mode
    memcpy(&instructionByteArray[16], &avgSamples, 1);
                // Bytes 16     - sampling window mode

    instruction = [NSData dataWithBytes:instructionByteArray
        length:HIGH_LEVEL_COMPILER_INSTRUCTION_SET_BYTE_LENGTH]
        ;
    NSLog(@"instruction : %@ appended to instructionSequence",
        instruction);
    [instructionAppendedSequence addObject:instruction];
    task.lowLevelTaskInstructions = [NSArray arrayWithObject:
        instruction];
}
else if ([task isKindOfClass:
    [POCLockInAmplificationAcquisition class]]) {

    POCLockInAmplificationAcquisition *lockIn =
        (POCLockInAmplificationAcquisition *)task;

    [instructionAppendedSequence addObjectsFromArray:[self
        encodeElementaryTask:[lockIn getWaveformGeneration]]];

    // INPUT TERMINAL
    uint16_t inTerm = [[lockIn.acquisitionTerminal
        valueForKey:@"Port"] unsignedCharValue];
    // CHANNEL MODE
    uint8_t chMode = 0x00;
    chMode = chMode | SYNCHRONOUS_ACQUISITION_ON |
        (LOCK_IN_ACQ_MODE << 2) | (0x03 & lockIn.taskStatus);
                                                // chMode code for
        lock-in amplification.
    // PAIRED EXCITATION FREQUENCY
    uint8_t pairedExcCh = [[lockIn.excitationTerminal
        valueForKey:@"Port"] unsignedCharValue];
    // NUMBER OF ACQUISITION PERIODS
    uint16_t nbAcqPeriods = [lockIn.nbAcquisitionPeriods
        unsignedShortValue];
    // DEMODULATION FREQUENCY
    float demodFreq = [lockIn.demodulationFrequency floatValue
        ];
    // SAMPLING ORDER
    uint8_t samplOrd = [lockIn.samplingOrder unsignedCharValue
        ];
    // FRAME-PER-PERIOD
    uint8_t fpp = [lockIn.framePerPeriod unsignedCharValue];
    // COMBINED SAMPLING-ORDER / FPP CODE BYTE
    uint8_t samplParam = (fpp << 4) | (0x0F & samplOrd);
    // GAIN
    uint8_t currentGain = 0x00;
    NSUInteger index =[[[POCAccessoryConstants
        sharedAccessoryConstants]
        currentAmplifiersSensitivities] indexOfObject:lockIn.
        sensitivity];
    currentGain = 0x0F & index;
    // FILTER
    uint8_t filterSelection = LOWEST_ORDER_FILTER;
    for (POCFIRFilter *filter in [[POCAccessoryConstants
        sharedAccessoryConstants] firFilters]) {
        if ([lockIn.firFilter isEqual:filter]) {
            filterSelection = 0x000F &
                [[[POCAccessoryConstants
                sharedAccessoryConstants] firFilters]
                indexOfObject:filter];
            break;
        }
    }
    // GAIN-FILTER SELECTION BYTE
    uint8_t gainFilterSelection = (filterSelection << 4) |
        (0x0F & currentGain);
    // DECIMATION FACTOR
    uint16_t decimFactorIndex = NO_DECIMATION;
    for (NSNumber *decFact in [[POCAccessoryConstants
        sharedAccessoryConstants] decimationFactors]) {
        if ([lockIn.decimationFactor isEqual:decFact]) {
            decimFactorIndex = 0x00FF &
                [[[POCAccessoryConstants
                sharedAccessoryConstants] decimationFactors]
                indexOfObject:decFact];
            break;
        }
    }
    // DATA OUTPUT
    uint8_t currentAcqOutput = 0x01;
                // 0x00 code : unprocessed data.
    // ENCODE INSTRUCTION
    memcpy(&instructionByteArray[0], &inTerm,1);
                // Byte  0      - input terminal #ID
    memcpy(&instructionByteArray[1], &chMode, 1);
                // Bytes 1      - channel Mode
    memcpy(&instructionByteArray[2], &pairedExcCh, 1);
                // Byte  2      - default paired channel : 0xFF
    memcpy(&instructionByteArray[3], &nbAcqPeriods, 2);
                // Bytes 3      - Number of acquisition points
    memcpy(&instructionByteArray[5], &demodFreq, 4);
                // Bytes 5:8    - sampling frequency
    memcpy(&instructionByteArray[9], &samplParam, 1);
                // Bytes 5:8    - sampling frequency
    memcpy(&instructionByteArray[11], &gainFilterSelection, 1)
        ;       // Byte  11 - current gain #index
    memcpy(&instructionByteArray[12], &decimFactorIndex, 1);
                        // Bytes 5:8    - sampling frequency
    memcpy(&instructionByteArray[13], &currentAcqOutput, 1);
                // Bytes 16   - output mode : unprocessed data.

    instruction = [NSData dataWithBytes:instructionByteArray
        length:HIGH_LEVEL_COMPILER_INSTRUCTION_SET_BYTE_LENGTH]
        ;
    NSLog(@"instruction : %@ appended to instructionSequence",
        instruction);
    [instructionAppendedSequence addObject:instruction];
    task.lowLevelTaskInstructions = [NSArray arrayWithObject:
        instruction];
}

// If compilation of a complex task : recurssive call to the
    compiling method.
else if ([task isKindOfClass:[POCLowLevelTask class]]){
    NSLog(@"Enter subtask sequence compilation for task with
        ID:%@ - of class: %@",task.taskId, [task class]);
    POCLowLevelTask * complexTask = (POCLowLevelTask *)task;
    NSArray *subtaskInstructions = [[NSArray alloc] init];
    for (POCElementaryTask *subtask in complexTask.
        sequentialSubtasksArray) {
        subtaskInstructions = [self encodeElementaryTask:
            subtask];
        subtask.lowLevelTaskInstructions = subtaskInstructions
            ;
        NSLog(@"Subtask program instruction: %@ appended to
            program instructions", subtaskInstructions);
        [instructionAppendedSequence addObjectsFromArray:
            subtaskInstructions];
    }
}
}
NSArray *programInstructions = [NSArray arrayWithArray:
    instructionAppendedSequence];
NSLog(@" (sub-)Program instructions: %@", programInstructions);
return programInstructions;
}


#pragma mark - ENCODE PROGRAM

+ (void)encodeProgram:(POCLoCProgram *)program{

    if (program != nil) {
        if (program.tasksArray.count > 0) {
            for (POCTask *task in program.tasksArray) {
                if ([task isKindOfClass:[POCElementaryTask class]]) {
                    POCElementaryTask *llTask = (POCElementaryTask *)task;
                    [llTask setLowLevelTaskInstructions:[self
                        encodeElementaryTask:llTask]];
                    NSLog(@"low-level task: %@ was ENCODEd with instructions:
                        %@",llTask,llTask.lowLevelTaskInstructions);
                } else if ([task isKindOfClass:[POCHighLevelTask class]]){
                    // NO COMPILATION REQUIRED.
                }
            }
        }
    }
}
@end
```

Appendix K

BLE Firmware

```
dim result
dim connected
dim temp_handle
dim temp_offset
dim temp_value_len
dim spi_channel
dim rlen
dim impedance_byte_count


####### BLE112 / HOST MCU COMMUNICATION SPI PROTOCOL ################################
####### SPI INTERFACE WITH HOST CONTROLLER - FEED THE Lockin Measurement Info attribute
####################
# Byte 1: spiSize will be the first byte transferred from the MCU to the BLE112. It will
contain information about the number of bytes to follow.
# Byte 2: attribute_type will permit the identification of the attribute to which data
received over SPI should belong:
#   "\x00": the following *spiSize bytes will correspond to the channel over which
measurement points have been acquired.
#   "\x03": the following bytes will correspond to lock-in -impedance- bytes ;
#   "\x05" ...
#   "\x07" Other codes will be implemented to identify data corresponding to calibration
values for the PGA-ADC.
#
#
#
# Byte 3 - Bytes 3+(*spiSize -1) data bytes of type attribute_type. (the minus one comes
from the attribute_type byte that is counted within the spiSize byte count.

####### DATA TRANSMITTED TO HOST PROTOCOL - LOCK-IN OPCODE + DATA
##################################################
# Make a table for clarity

# Byte 0: Opcode for the PoC Lock-In    (among the Opcodes: one for PGA and Current Gain
Control, One for default, One for Single-channel mode, One for
Calibration)
# Opcode 0x00 : Default mode: 40Hz excitation signal, 5 sec burst interval, 256 burst
points, automated-gain control, automated calibration. No following bytes required
# Opcode 0x03 : Advanced mode: Following bytes required:
#   Byte 1-2 : Nb acquisition points per burst (max 65535).
#   Byte 3-6 : Excitation frequency
#   Byte 7-8 : Number of samples per period // 0x00 if under sampling is enabled.
#   Byte 9 :   Interval between acquisitions (in seconds from 0 to 255)
#   Byte 10-11 : undersampling order
#   Byte 12-13 : undersampling Frames Per Period (FPP)
#   Byte 14 :    excitation voltage (coded over 1 byte)
#   Byte 15-19 :



##########################################################################################

###### LOCKIN MEASUREMENT INFO ATTRIBUTE - NOTIFIED #######################################

#   Byte 0 - Flags
#           bit 0 : Overloaded channels (Low-impedance sensor or short)
#           bit 1 : Unusable sensor : all necessary sensors are overloaded.
#           bit 2 :
#           bit 3 :
#           bit 4 : ?
#           bit 5 : ?
#           bit 6 : ?
#           bit 7 : ?
#   Byte 1-2- ??
#   Bytes 3-4 : ??
#   Bytes 5-8: Excitation frequency (limited resolution)
#   Bytes 9-12: Sampling frequency (limited resolution)
#   Byte 13 : PGA Gain (if 0xFF - Gain = 256)
#   Byte 14 : Current Gain
#   Bytes 15-19 : Calibration constants ??
#

###### LoC INFO ATTRIBUTE - READ-ONLY IN OUR CASE ##########################################
#   Byte 0 : LoC Flags
#       bit 0: LoC disconnected
#       bit 1: Microfluidic path complete
#       bit 2: ...
#       bit 3: ...
#       bit 4:
#       bit 5:
#       bit 6:
#       bit 7:
#   Byte 1-2 Number of channels on LoC
#   Byte 3-7 : Valid channels mask (no overload). Max 40 channels
#   Byte 8-12 : Reference channel mask (supposedly attributed from remote client
#   Byte 13-17 : Calibration channel mask.
#   Byte 18-19 : FREE
##########################################################################################
##

###### AckImpedance attribute -  ##########################################
#
#   Impedance attribute notification receive is acknowledged by the client by setting the
AckImpedance attribute to 0x39
#   Upon receive of acknowledgement: send SPI message - Opcode 0x10 - to host controller for
requesting new Impedance measurement;
#
#######################################################################


## USER CONTROLLABLE PARAMETERS ##
dim opcode(1)

## NOTIFIED AND INDICATED VALUES
dim lockin_measurement_info(20)
dim spi_message_length(1)
dim attribute_type(1)
dim acquisition_channel(2)
dim ack(2)


## SPI COMMUNICATION BUFFERS AND FLAGS
dim RX_flag
dim TX_flag
dim spi_TX_buffer(22)
dim spi_RX_buffer(22)
dim spi_RX_buffer_index
dim spi_TX_buffer_index
dim spi_TX_length(1)
dim spi_RX_length(1)
dim TX_counter
dim RX_counter
dim debug_index
dim debug_loc_attribute(22)
dim major_info
dim minor_info
dim patch_info
dim build_info
dim ll_version_info
dim protocol_version_info
dim hw_info

##########################################################################################
##########################################################################################

event system_boot(major, minor, patch, build, ll_version, protocol_version, hw)

    call sm_set_bondable_mode(1)
    #   Configure for output : port1 for output
    #   call hardware_io_port_config_pull(0,"\x80",0)               # Port 0 pins are all
    pulled down except for pin 0_7 (LED OUT)
    #   call hardware_io_port_config_pull(1,"\xB2",0)               # Port 1 pins are not
    pulled. except for P1_1, P1_6 and P1_7f

    call hardware_io_port_config_direction(1,"\xED")            # Pins 1_4 and P1_1 as inputs
    call hardware_io_port_config_function(1, "\x38")            # Pins 1_3,  1_4 and 1_5
    configured for peripheral function CLK, MI and MO respectively


    #Connection status:
    connected = 0

    # Enable interrupts on P1_1
    call hardware_io_port_irq_enable(1, "\x02")
    call hardware_io_port_irq_direction(1,0)            # enable interrupt on P1_1, rising
    edge

    #Set advertising pause intervals and start advertising
    call gap_set_adv_parameters(32,48,7)
    call gap_set_mode(gap_general_discoverable, gap_undirected_connectable)


    ############ DEBUG #######################
    #   call hardware_set_soft_timer(16000,1,1)
    ############ DEBUG #######################

    # Spi communication
    TX_flag = 0
    RX_flag = 1          # By default, the BLE module can interrupted by MCU for SPI
    communication: RX flag is set.
                         # Only when the Opcode or Acknowledgement attributes are written by
                         client is RX flag reset
    RX_counter = 0
    TX_counter = 0
    spi_TX_length(0:1) = 0
    spi_RX_length(0:1) = 0
    spi_TX_buffer(0:22) =
    "\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00"
    spi_RX_buffer(0:22) =
    "\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00"
    spi_message_length(0:1) = "\x00"
    attribute_type(0:1) = "\x00"
    ack(0:2) = "\x00\x00"
end

############################## timer
##########################################################################
event hardware_soft_timer(timer_handle)
    if timer_handle = 1 then
        call attributes_write(AccessoryReadout, 0, 16, spi_RX_buffer(0:16))

        ############ DEBUG #######################
        #   debug_index = debug_index + 1
        #   debug_loc_attribute(0:20) =
        "\x78\x78\x78\x78\x78\x78\x78\x78\x78\x78\x78\x78\x78\x78\x78\x78\x78\x78\x78\x78\xFF"
        #   debug_loc_attribute(0:1) = debug_index
        #   debug_loc_attribute(1:2) = result
        #   call attributes_write(HardwareInfo,0,20,debug_loc_attribute(0:20))
        ########################################

    end if
end

event attributes_status(handle, att_flags)
    #if  ((handle = HardwareInfo) && (att_flags = attributes_attribute_status_flag_notify))
    then
    #################  DEBUG ################
    #   debug_loc_attribute(0:20) =
    "\x31\x31\x31\x31\x31\x31\x31\x31\x31\x31\x31\x31\x31\x31\x31\x31\x31\x31\x31\x31\xFF"
    #   debug_loc_attribute(0:1) = result
    #   call attributes_write(HardwareInfo,0,20,debug_loc_attribute(0:20))
    #end if

    #if ((handle = Impedance) && (att_flags = attributes_attribute_status_flag_notify))then
    #   debug_loc_attribute(0:20) =
    "\xdd\xdd\xdd\xdd\xdd\xdd\xdd\xdd\xdd\xdd\xdd\xdd\xdd\xdd\xdd\xdd\xdd\xdd\xdd\xdd\xFF"
    #   debug_loc_attribute(0:1) = result
    #   call attributes_write(HardwareInfo,0,20,debug_loc_attribute(0:20))
    #end if
    ##################################
end


################################# attributes_value event handling
##################################################
event attributes_value(connection, reason, handle, offset, value_len, value_data)

   #################
    # if RX_counter != 0 then
    #    return
    # end if
    #################
    TX_flag = 1
    TX_counter = 0
    RX_flag = 0

    if handle = AccessoryConfig then                                        # a few
    users may want to write the opcode directly.
        # each SPI transfer will start with the msg_length byte. msg_length < 255.
        # We set the default spi_message_length to 20: maximum attribute length.
        #opcode(0:1) = value_data(0:1)

        #First byte transmitted to MCU is the msg length
            spi_TX_length(0:1) = value_len + 1      # The msg length is value_len (20 for
            the 20 bytes of the attribute) + one byte coding for the attributeId itself.
```

```
        #Second byte transmitted to MCU is the attributeId . for AccessoryConfig: 0x00
            spi_TX_buffer(0:1) = "\x00"
        #Following bytes correspond to the attribute bytes
            spi_TX_buffer(1:value_len) = value_data(0:value_len)
            ########### DEBUG #######################
            #   debug_loc_attribute(0:20) =
            "\x16\x16\x16\x16\x16\x16\x16\x16\x16\x16\x16\x16\x16\x16\x16\x16\x16\x16\x16\xFF"
            #   debug_loc_attribute(0:spi_TX_length(0:1)) =
            spi_TX_buffer(0:spi_TX_length(0:1))
            #   call attributes_write(HardwareInfo,0,20,debug_loc_attribute(0:20))
            ##########################################
            call hardware_io_port_write(1, $04, 0) # Select MCU : pull PORT 1_2 down
            call hardware_spi_transfer(0,1,spi_TX_length(0:1))
            call hardware_io_port_write(1, $04, $04) # Unselect MCU : pull PORT 1_2 up
            #### Debug #######################
            #   lockin_measurement_info(0:value_len) = value_data(0:value_len)
            #   call hardware_set_soft_timer(1024,1,1) # handle = 1 ; one-shot timer
            ##################################
            return
        #end if                                              # We assume only the default
        mode for now: no user-specified parameters. The 0x00 code will trigger
                                                            # acquisition with the
                                                            default parameters.
        # if opcode(0:1) = "\x03" then
            # spi_TX_length(0:1) = value_len
            # spi_TX_buffer(0:spi_TX_length(0:1)) = value_data(0:value_len)

            # ########### DEBUG #######################
                # debug_loc_attribute(0:20) =
                "\x78\x78\x78\x78\x78\x78\x78\x78\x78\x78\x78\x78\x78\x78\x78\x78\x78\x78\x78\
                xFF"
                # debug_loc_attribute(0:value_len) = spi_TX_buffer(0:spi_TX_length(0:1))
                # call attributes_write(HardwareInfo,0,20,debug_loc_attribute(0:20))
            # ##########################################
            # call hardware_io_port_write(1, $04, 0) # Select MCU : pull PORT 1_2 down
            # call hardware_spi_transfer(0,1,spi_TX_length(0:1))
            # call hardware_io_port_write(1, $04, $04) # Unselect MCU : pull PORT 1_2 up
            # return
        # end if

        # ########### DEBUG - RESET ALL VARIABLES #######################
        # if opcode(0:1) = "\x77" then
            # TX_counter = 0
            # RX_counter = 0
            # spi_TX_length(0:1) = 0
            # spi_RX_length(0:1) = 0
            # spi_TX_buffer(0:22) =
            "\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\
            x00\x00"
            # spi_RX_buffer(0:22) =
            "\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00"
            # attribute_type(0:1) = "\x00"
            # ack(0:2) = "\x00\x00"
            # return
        # end if
        ################################################################
    end if

    if handle = Acknowledgements then

            spi_TX_length(0:1) = 2
            ack(0:2) = value_data(0:2)
            if ack(0:2) = "\x39\x04" then
                #   call hardware_set_soft_timer(0,0,0)             # Disable timer to stop
                sending update notifications
                spi_TX_buffer(0:2) = ack(0:2)
                TX_counter = 0
                spi_RX_buffer(0:20) =
                "\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\
                x00"
                spi_RX_buffer_index = 0
                spi_RX_length(0:1) = 0
                RX_counter = 0
                call hardware_io_port_write(1, $04, 0) # Select MCU : pull PORT 1_2 down
                call hardware_spi_transfer(0,1,spi_TX_length(0:1))
                call hardware_io_port_write(1, $04, $04) # Unselect MCU : pull PORT 1_2 up
                return
            end if
            if ack(0:2) = "\x49\xb2" then
                call hardware_set_soft_timer(0,0,0)     # Disable timer to stop sending
                update notifications
                spi_TX_buffer(0:2) = ack(0:2)
                TX_counter = 0
                spi_RX_buffer(0:20) =
                "\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\
                x00"
                spi_RX_buffer_index = 0
                spi_RX_length(0:1) = 0
                RX_counter = 0
                call hardware_io_port_write(1, $04, 0) # Select MCU : pull PORT 1_2 down
                call hardware_spi_transfer(0,1,spi_TX_length(0:1))
                call hardware_io_port_write(1, $04, $04) # Unselect MCU : pull PORT 1_2 up
            end if
    end if
    if reason= attributes_attribute_change_reason_write_request_user
        call attributes_user_write_response(connection,"\x00")
    end if
end



event hardware_io_port_status(timestamp, port, irq, state)
    ##########################################################################
    #    SPI TX MODE
    ##########################################################################
    if TX_flag = 1 then
        if TX_counter < spi_TX_length(0:1) then
            call hardware_io_port_write(1, $04, 0) # Select MCU : pull PORT 1_2 down
            call hardware_spi_transfer(0,1,spi_TX_buffer(TX_counter:1))
            call hardware_io_port_write(1, $04, $04) # Unselect MCU : pull PORT 1_2 up
            ########### DEBUG #######################
                debug_loc_attribute(0:20) =
                "\xBF\xBF\xBF\xBF\xBF\xBF\xBF\xBF\xBF\xBF\xBF\xBF\xBF\xBF\xBF\xBF\xBF\xBF\xBF\
                xFF"
                debug_loc_attribute(0:1) = TX_counter
                debug_loc_attribute(1:1) = spi_TX_length(0:1)
                call attributes_write(HardwareInfo,0,20,debug_loc_attribute(0:20))
            ##########################################
            TX_counter = TX_counter + 1
        end if
        if TX_counter = spi_TX_length(0:1) then
            ########### DEBUG #######################
                debug_loc_attribute(0:20) =
                "\x95\x95\x95\x95\x95\x95\x95\x95\x95\x95\x95\x95\x95\x95\x95\x95\x95\x95\x95\
                xFF"
                debug_loc_attribute(0:1) = TX_counter
            #   debug_loc_attribute(1:1) = spi_TX_length(0:1)
            #   debug_loc_attribute(2:18) = spi_TX_buffer(0:18)
                call attributes_write(HardwareInfo,0,20,debug_loc_attribute(0:20))
            ##########################################
            if spi_TX_length(0:1) = 2 then
                call attributes_write(Acknowledgements,0,2,"\x09\x07")      # Reset the
                acknowledgement flags if needed
            end if
            spi_message_length(0:1) = 0
            attribute_type(0:1) = 0
            spi_RX_length(0:1) = 0
            TX_counter = 0
            RX_counter = 0
            TX_flag = 0
            RX_flag = 1
            spi_TX_length(0:1) = 0
            spi_TX_buffer(0:22) =
            "\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\
            x00\x00"
            spi_RX_buffer_index = 0
        end if
        return
    end if

    ##########################################################################
    #   SPI RX MODE
    ##########################################################################
    if RX_flag = 1 then
        if RX_counter = 0 then
            call hardware_io_port_write(1, $04, 0) # Select MCU : pull PORT 1_2 down
            call hardware_spi_transfer(0,2,"\xAC\xAC")(result, spi_channel, tlen,
            spi_RX_buffer(RX_counter:2))  # First byte corresponds to the message length in
            bytes..
            call hardware_io_port_write(1, $04, $04) # Unselect MCU : pull PORT 1_2 up
            spi_RX_length(0:1) = spi_RX_buffer(0:1) + 2  ## FB ???????
            attribute_type(0:1) = spi_RX_buffer(1:1)

            ########### DEBUG #######################
                debug_loc_attribute(0:20) =
                "\x00\x00\x01\x01\x01\x01\x01\x01\x01\x01\x01\x01\x01\x01\x01\x01\x01\x01\x01\
                xFF"
                debug_loc_attribute(0:1) = spi_RX_length(0:1)
                debug_loc_attribute(1:1) = attribute_type(0:1)
                call attributes_write(HardwareInfo,0,20,debug_loc_attribute(0:20))
            ##########################################
            RX_counter = RX_counter + 2
            if spi_RX_length(0:1) = 0 then
                RX_counter = 0
                spi_RX_length(0:1) = 0
                ########### DEBUG #######################
                    debug_loc_attribute(0:20) =
                    "\x02\x02\x02\x02\x02\x02\x02\x02\x02\x02\x02\x02\x02\x02\x02\x02\x02\x02\x02\
                    x02\xFF"
                    call attributes_write(HardwareInfo,0,20,debug_loc_attribute(0:20))
                ##########################################
            end if
            return
        else
             # if RX_counter = 1 then
                 # call hardware_io_port_write(1, $04, 0) # Select MCU : pull PORT 1_2 down
                 # call hardware_spi_transfer(0,1,"\xAC")(result, spi_channel, tlen,
                 spi_RX_buffer(RX_counter:1))   # First byte corresponds to the message
                 length in bytes..
                 # call hardware_io_port_write(1, $04, $04) # Unselect MCU : pull PORT 1_2 up
                 # attribute_type(0:1) = spi_RX_buffer(1:1)
                 # ########### DEBUG #######################
                 debug_loc_attribute(0:20) =
                 "\xA5\xA5\xA5\xA5\xA5\xA5\xA5\xA5\xA5\xA5\xA5\xA5\xA5\xA5\xA5\xA5\xA5\xA5\xA5\xA5
                 \xFF"
                 debug_loc_attribute(0:1) = RX_counter
                 debug_loc_attribute(1:1) = spi_RX_length(0:1)
                 debug_loc_attribute(2:1) = attribute_type(0:1)
                 call attributes_write(HardwareInfo,0,20,debug_loc_attribute(0:20))
                 ##########################################
                 # RX_counter = RX_counter + 1
                 # return
             # else
                if  RX_counter < spi_RX_length(0:1) then
                    #spi_RX_buffer_index = RX_counter-2
                    call hardware_io_port_write(1, $04, 0) # Select MCU : pull PORT 1_2 down
                    call hardware_spi_transfer(0,2,"\xAC\xAC")(result, spi_channel, tlen,
                    spi_RX_buffer(RX_counter:2))  # First byte corresponds to the message
                    length in bytes..
                    call hardware_io_port_write(1, $04, $04) # Unselect MCU : pull PORT 1_2 up
                    ########### DEBUG #######################
                    #   debug_loc_attribute(0:20) =
                    "\xAA\xAA\xAA\xAA\xAA\xAA\xAA\xAA\xAA\xAA\xAA\xAA\xAA\xAA\xAA\xAA\xAA\xAA\xAA\
                    xAA\xFF"
                    #   debug_loc_attribute(0:1) = RX_counter
                    #   debug_loc_attribute(1:2) = spi_RX_buffer(RX_counter:2)
                    #   debug_loc_attribute(3:1) = spi_RX_length(0:1)
                    #   call attributes_write(HardwareInfo,0,20,debug_loc_attribute(0:20))
                    ##########################################
                    RX_counter = RX_counter + 2

                    if RX_counter >= spi_RX_length(0:1) then
                    # RESET THE RX_counter ONCE MESSAGE IS FULL AND UPDATE THE ATTRIBUTES
                        ########### DEBUG #######################
                            debug_loc_attribute(0:20) =
                            "\xEE\xEE\xEE\xEE\xEE\xEE\xEE\xEE\xEE\xEE\xEE\xEE\xEE\xEE\xEE\xEE\xEE\xEE\xEE\
                            xEE\xEE\xEE\xFF"
                            debug_loc_attribute(0:1) = RX_counter
                            debug_loc_attribute(1:1) = spi_RX_length(0:1)
                            debug_loc_attribute(2:1) = attribute_type(0:1)
                            debug_loc_attribute(3:16) = spi_RX_buffer(0:16)
                            call attributes_write(HardwareInfo,0,20,debug_loc_attribute(0:20))
                        ##########################################

                        if ((spi_RX_length(0:1) = 2) && (attribute_type(0:1) = "\x00"))  then
                         #If attribute_type byte = "\x00", the following two bytes are
                        coding for the channel from which acquisition was performed
                            ########### DEBUG #######################
                            #   debug_loc_attribute(0:20) =
                            "\x00\x00\x00\x00\x00\x00\x00\x00\x99\x88\x00\x00\x00\x00\x00\
                            x00\x00\x00\xFF"
                            #   debug_loc_attribute(0:2) = spi_RX_buffer(0:2)
                            #   call
                            attributes_write(HardwareInfo,0,20,debug_loc_attribute(0:20))
                            ##########################################
```

```
                    spi_RX_buffer(0:22) =
                    "\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\
                    x00\x00\x00\x00\x00\x00\x00"
                    spi_RX_buffer_index = 0
                    spi_RX_length(0:1) = 0
                    RX_counter = 0
                    return
                end if
                if attribute_type(0:1) = "\x6B" then                #If
                attribute_type byte = "\x&B", the following 20 bytes are coding for
                the Channel Readout ############ DEBUG #######################
                    call attributes_write(AccessoryReadout, 0, 20, spi_RX_buffer(2:20
                    )) #Write from byte 2 till 22.
                    spi_RX_buffer_index = 0
                    spi_RX_length(0:1) = 0
                    RX_counter = 0
                    ##########################################
                        debug_loc_attribute(0:20) =
                        "\x03\x03\x03\x03\x03\x03\x03\x03\x03\x03\x03\x03\x03\x03\x03\x03\x03\x03\x03\
                        x03\x03\x03\x03\x03\xFF"
                        debug_loc_attribute(0:1) = RX_counter
                        debug_loc_attribute(1:1) = spi_RX_length(0:1)
                        call attributes_write(HardwareInfo,0,20,debug_loc_attribute(0
                        :20))
                    ##########################################
                    return
                end if

                ############ DEBUG #######################
                    debug_loc_attribute(0:20) =
                    "\xD7\xD7\xD7\xD7\xD7\xD7\xD7\xD7\xD7\xD7\xD7\xD7\xD7\xD7\xD7\xD7\xD7\xD7\xD7\
                    xD7\xD7\xD7\xFF"
                    debug_loc_attribute(0:1) = attribute_type(0:1)
                    call attributes_write(HardwareInfo,0,20,debug_loc_attribute(0:20))
                ##########################################

                spi_RX_buffer(0:22) =
                "\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\
                x00\x00\x00\x00\x00"
                spi_RX_buffer_index = 0
                spi_RX_length(0:1) = 0
                RX_counter = 0
                return
            end if
        else
            if RX_counter >= spi_RX_length(0:1) then
                    ############ DEBUG #######################
                        debug_loc_attribute(0:20) =
                        "\xe2\xe2\xe2\xe2\xe2\xe2\xe2\xe2\xe2\xe2\xe2\xe2\xe2\xe2\xe2\xe2\xe2\xe2\xe2\
                        xe2\xe2\xe2\xe2\xe2\xFF"
                        debug_loc_attribute(0:1) = RX_counter
                        debug_loc_attribute(1:1) = spi_RX_length(0:1)
                        call attributes_write(HardwareInfo,0,20,debug_loc_attribute(0
                        :20))
                    ##########################################
                RX_counter = 0
                spi_RX_length(0:1) = 0
            end if
        end if
    #end if
  end if
end if
end


############ connection_disconnected event handling ##############
event connection_disconnected(connection, reason)
    connected = 0
    call gap_set_mode(gap_general_discoverable, gap_undirected_connectable)
end

############ connection_status event handling #############
event connection_status(connection, flags, address, address_type, conn_interval, timeout,
latency, bonding)
    if flags = connection_completed
        connected = 1
    end if
end
```

Copyright: François Patou
All rights reserved

Published by:
DTU Nanotech
Department of Micro- and Nanotechnology
Technical University of Denmark
Ørsteds Plads, building 345C
DK-2800 Kgs. Lyngby