Fast Analysis of Nanomechanical Resonant Sensing Systems based on Canonical Models

by

Tarık Kargıoğlu

A Thesis Submitted to the
Graduate School of Sciences and Engineering
in Partial Fulfillment of the Requirements for
the Degree of

Master of Science

in

Electrical and Electronics Engineering

KOÇ ÜNİVERSİTESİ

September 23, 2022

Fast Analysis of Nanomechanical Resonant Sensing Systems based on Canonical Models

Koç University

Graduate School of Sciences and Engineering

This is to certify that I have examined this copy of a master's thesis by

Tarık Kargıoğlu

and have found that it is complete and satisfactory in all respects,
and that any and all revisions required by the final
examining committee have been made.

Committee Members:

Prof. Alper Demir (Advisor)

Prof. Alper Erdoğan

Assist. Prof. Önder Şuvak

Date:

ABSTRACT

Fast Analysis of Nanomechanical Resonant Sensing Systems based on Canonical Models
Tarık Kargıoğlu
Master of Science in Electrical and Electronics Engineering
September 23, 2022

We present NEMSUITE, a software tool for the analysis and simulation of nanoelectromechanical sensing systems based on resonance frequency tracking. NEMSUITE consists of a custom MATLAB® SIMULINK® block library and associated MATLAB® scripts that run in the background. NEMSUITE can be used to build and analyze NEMS based sensing schemes with the familiar graphical SIMULINK® interface. Custom models included in the block library capture various nonidealities and noise. NEMSUITE is designed as a purely software equivalent to widely used lock-in amplifier based setups. Analysis and simulation results that can be obtained with NEMSUITE can be directly compared with experimental measurements. NEMSUITE features user selectable analysis modes, including very fast analysis-based model evaluations as well as detailed stochastic simulations that include various sources of noise and customizable sensor events. Analysis and simulation results are directly mapped to Allan Deviations or fractional resonance frequency power spectral densities for ease of interpretation. NEMSUITE offers two analysis modes. Full stochastic simulation mode produces the most accurate results but takes hours to run. Fast analysis-based model evaluation mode avoids lengthy stochastic simulations and produces results that very well match the ones produced by expensive simulations. The fast analysis mode is based on a detailed analytical theory and canonical models that we extend to cover hybrid setups. We have implemented NEMSUITE models corresponding to almost all resonance frequency tracking schemes studied in the literature and currently in use, e.g., the feedback-free, frequency-locked loop, and self-sustained oscillator schemes, and other hybrid ones. These models are included in NEMSUITE as examples. We believe that NEMSUITE can serve as a trusted platform through which various sensor conceptions can be rigorously evaluated and compared before they are tested in expensive and time consuming experiments, as

well as to corroborate experimental data.

ÖZETÇE

Nanomekanik Rezonans Algılama Sistemlerinin Kanonik Modellere Dayalı Hızlı Analizi
Tarık Kargıoğlu
Elektrik ve Elektronik Mühendisliği, Yüksek Lisans
23 Eylül 2022

Rezonans frekans takibine dayalı nanoelektromekanik algılama sistemlerinin analizi ve simülasyonu için bir yazılım aracı olan NEMSUITE'i sunuyoruz. NEMSUITE, özel bir MATLAB® SIMULINK® blok kütüphanesinden ve arka planda çalışan ilişkili MATLAB® komut dosyalarından oluşur. NEMSUITE, tanıdık SIMULINK® grafik arayüzü ile NEMS tabanlı algılama şemalarını oluşturmak ve analiz etmek için kullanılabilir. Blok kütüphanesine dahil edilen özel modeller, çeşitli ideal olmayan durumları ve gürültüyü içerir. NEMSUITE, yaygın olarak kullanılan kilitli amplifikatör tabanlı kurulumlara eşdeğer bir yazılım olarak tasarlanmıştır. NEMSUITE ile elde edilebilecek analiz ve simülasyon sonuçları deneysel ölçümlerle karşılaştırılabilir. NEMSUITE, çok hızlı analize dayalı model değerlendirmelerinin yanı sıra çeşitli gürültü kaynakları ve özelleştirilebilir sensör olaylarını içeren ayrıntılı stokastik simulasyonlar dahil olmak üzere kullanıcı tarafından seçilebilen analiz modlarına sahiptir. Analiz ve simülasyon sonuçları, yorumlama kolaylığı için doğrudan Allan Sapmalarına veya fraksiyonel rezonans frekans güç spektral yoğunluklarına eşlenir. NEMSUITE iki analiz modu sunar. Tam stokastik simülasyon modu en doğru sonuçları verir ancak çalışması saatler sürer. Hızlı analize dayalı model değerlendirme modu, uzun stokastik simülasyonlardan kaçınır ve pahalı simülasyonlar tarafından üretilenler ile çok iyi eşleşen sonuçlar üretir. Hızlı analiz modu, ayrıntılı bir analitik teoriye ve hibrit kurulumları kapsayacak şekilde genişlettiğimiz standart modellere dayanmaktadır. Literatürde incelenen ve şu anda kullanımda olan hemen hemen tüm rezonans frekansı izleme şemalarına karşılık gelen NEMSUITE modellerini ürettik, örn., geri beslemesiz, frekans kilitli döngü ve kendi kendine devam eden osilatör şemaları ve hibrit olanlar. Bu modeller NEMSUITE'te örnek olarak yer almaktadır. NEMSUITE'in, çeşitli sensör tasarımlarının, pahalı ve zaman alıcı deneylerde test edilmeden önce titizlikle değerlendirilebileceği, karşılaştırılabileceği ve ayrıca

deneysel verileri doğrulamak için güvenilir bir platform olarak hizmet edebileceğine inanıyoruz.

ACKNOWLEDGMENTS

I would like to thank my advisor Prof. Alper Demir for his guidance and support, also to committee members Prof. Alper Erdoğan and Assist. Prof. Önder Şuvak for their comments and contributions. This work is built upon the previous theoretical investigations by Prof. Demir. I was supported by Scientific and Technological Research Council of Turkey (TÜBİTAK) 2211 Domestic Graduate Scholarship Program (In Turkish: Çalışmalarım TÜBİTAK 2211 Yurt İçi Lisansüstü Burs Programı ile desteklendi).

TABLE OF CONTENTS

List of Figures **xi**

Abbreviations **xiii**

Chapter 1: Introduction **1**

1.1 Resonance Frequency based Sensing 2

1.2 NEMSUITE Overview 3

Chapter 2: Theory **4**

2.1 Oscillatory Cores 4

2.1.1 Feedback-free Resonator Model 5

2.1.2 Resonator Phase Response 8

2.1.3 Self-oscillating Core Model 8

2.1.4 Hybrid Core Oscillator Model 10

2.1.5 Hybrid Core Frequency of Oscillations 13

2.1.6 Injection Lock in the Hybrid Core 14

2.1.7 Hybrid Core Phase Response 16

2.2 Frequency Detection Methods 19

2.2.1 Demodulator 20

2.2.2 Phase-locked Loop (PLL) 20

2.2.3 Frequency-locked Loop (FLL) 22

2.3 Noise 23

2.3.1 Thermomechanical Noise 23

2.3.2 Detection Noise 23

2.3.3 Noise in the Self-Oscillating Core 24

2.3.4 Noise in the Hybrid Core 24

2.4 Phase Domain Models 24
2.4.1 Resonator 25
2.4.2 Self-Oscillating Core 26
2.4.3 Hybrid Core 26
2.4.4 Demodulator 26
2.4.5 PLL 27

Chapter 3: Schemes **28**

3.1 Feedback-free Scheme (FF) 29
3.2 Frequency Locked Loop (FLL) 29
3.3 Self-sustaining Oscillator (SSO) with Demodulator 31
3.4 Self-sustaining Oscillator (SSO) with PLL 33
3.5 Hybrid Open-loop (HOP) 34
3.6 Hybrid Closed-loop (HCL) 36

Chapter 4: The Tool: NEMSUITE **41**

4.1 Custom Simulink® Blocks 41
4.1.1 Resonator 42
4.1.2 Self-Oscillating Core 42
4.1.3 Hybrid Core 44
4.1.4 Demodulator 44
4.1.5 Controller 45
4.1.6 Controlled Oscillator 45
4.1.7 Local Oscillator 46
4.1.8 White Noise 47
4.1.9 Maestro 48
4.2 Simulation Types 48
4.2.1 Full Simulation 48
4.2.2 Fast Analysis 50

Chapter 5: Results **52**

5.1 Validation of the Fast Analysis 52

5.1.1 Accuracy 53

5.1.2 Speed 53

5.2 Results Supporting the Theory Obtained with NEMSUITE 53

Chapter 6: Conclusion **58**

Bibliography **60**

LIST OF FIGURES

2.1 A diagram of the self-oscillating core. 9

2.2 A diagram of the hybrid core. 11

2.3 A diagram of the demodulator. 20

2.4 A diagram of the phase-locked loop. 21

2.5 A diagram of the negative feedback loop. 22

2.6 A diagram of the phase domain equivalent of the resonator. 25

2.7 A diagram of the phase domain equivalent of the self-oscillating core. 26

2.8 A diagram of the phase domain equivalent of the hybrid core. 27

3.1 A diagram of the FF scheme (adapted from [1]). 30

3.2 A diagram of the phase domain equivalent of the FF scheme (adapted from [1]). 30

3.3 A diagram of the FLL scheme (adapted from [1]). 31

3.4 A diagram of the phase domain equivalent of the FLL scheme (adapted from [1]). 32

3.5 A diagram of the SSO with demodulator scheme (adapted from [1]). . 33

3.6 A diagram of the phase domain equivalent of the SSO with demodulator scheme (adapted from [1]). 34

3.7 A diagram of the SSO with PLL scheme. 35

3.8 A diagram of the phase domain equivalent of the SSO with PLL scheme. 36

3.9 A diagram of the HOP scheme. 37

3.10 A diagram of the phase domain equivalent of the HOP scheme. 38

3.11 A diagram of the HCL scheme. 39

3.12 A diagram of the phase domain equivalent of the HCL scheme. 40

4.1 Block parameters interface of the resonator block. 43

4.2 Internal structure of the self-oscillating core block. 43
4.3 Internal structure of the phase domain version of the self-oscillating core block. 44
4.4 Internal structure of the hybrid core block. 44
4.5 Internal structure of the phase domain version of the hybrid core. . . 45
4.6 Internal structure of the demodulator block. 45
4.7 Internal structure of the controlled oscillator block. 46
4.8 Block parameters interface of the local oscillator block. 46
4.9 Internal structure of the local oscillator block. 47
4.10 Block parameters interface of the white noise block. 47
4.11 Interface of the maestro block. 48

5.1 Sample Allan Deviation plots showing the correspondence between the full simulation and the fast analysis results. All plots are generated from sensing schemes built with NEMSUITE and contain both thermomechanical and detection noise sources. (a) FF scheme Q=50, (b) FF scheme Q=1000, (c) FLL scheme, (d) SSO with demodulator, (e) HOP scheme $\alpha = 0.1, \beta = 0.9$, (f) HOP scheme $\alpha = 0.5, \beta = 0.5$. . 54
5.2 Sample frequency step response plots. (a) FF scheme, (b) FLL scheme, (c) SSO with demodulator, (d) HOP scheme 55
5.3 Allan Deviations at the output of an FF setup, as Q varies. 56
5.4 Allan Deviations at the output of an HOP setup, as α and β vary. . . 57

ABBREVIATIONS

NEMS	Nano Electromechanical Systems
FF	Feedback-Free
FLL	Frequency-Locked Loop
PLL	Phase-Locked Loop
SSO	Self-Sustained Oscillator
HOP	Hybrid Open Loop
HCL	Hybrid Closed Loop
PI	Proportional Integral

Chapter 1

INTRODUCTION

Nanomechanical resonant sensing has proved to be a promising technology with various areas of application such as mass spectrometry and atomic force microscopy. Although many great achievements including single-protein real-time mass spectrometry [2] and yoctogram mass resolution have been recorded [3], nanomechanical resonant sensing is still a highly active area of research and development in academia as well as industry. This multidisciplinary endeavor would be more efficient and impactful, if supported with proper computer-aided analysis and testing.

Research and development of nanomechanical resonant sensing typically involves plenty of carefully conducted experiments requiring complicated and expensive equipment such as nanoscale components and lock-in amplifiers. A software tool that provides accurate simulations or fast analysis that can corroborate these money and time-consuming experiments would accelerate the design an development process. NEMSUITE has been developed to fulfill this need. NEMSUITE is a software tool that enables users to analyze and test nanomechanical resonant sensing schemes before engaging in sophisticated experiments and to corroborate and explain experimental results.

Previous studies include experimental results [4, 5, 6, 7, 8] obtained from nanomechanical resonant sensing setups as well as detailed analysis of the setups [1] with comparisons between the theoretical, experimental and simulated results [9, 10]. We have developed an end-user software tool, NEMSUITE, which is built upon an extended version of previous theoretical investigations [1], to make the numerical simulations and the theoretical analysis accessible to anyone working in the field.

This thesis provides the theoretical background with some extensions, and serves as the design and user manual of NEMSUITE. In chapter 2, the theoretical analysis is presented, mathematical models for the physical modules used in nanomechanical sensing setups are produced. The resonance frequency tracking schemes are studied in chapter 3. Chapter 4 summarizes NEMSUITE. In chapter 5, the results obtained using NEMSUITE are shown. And finally, chapter 6 contains a summary and conclusions. We next briefly go over resonator-based sensing and a short overview of NEMSUITE in the following two sections.

1.1 Resonance Frequency based Sensing

The core component of a nanomechanical resonant sensor is the resonator. An event of interest causes shifts in characteristics of the resonator, such as the resonance frequency or the quality factor. Events are quantified by identifying the shifts in the characteristics. We consider only resonance frequency shifts hence resonance frequency based sensing in this thesis.

To detect events, the resonance frequency of the resonator must be monitored. One important aspect here is that, the resonance frequency of the resonator may be different from the frequency of oscillations at the output of the resonator. Therefore, measuring the resonance frequency is not necessarily as straightforward as measuring the frequency of a signal.

There are several schemes for monitoring resonance frequency of a resonator. These schemes incorporate different methods to track the resonance frequency with some pros and cons. The schemes are evaluated in terms of both accuracy and speed, and a detailed comparison of the most common schemes has been recently performed in the literature [1]. The scope of this thesis excludes comparisons of the schemes, rather we present the schemes and the consistency of the results obtained using our software tool without commenting on the comparative performance of the schemes.

1.2 NEMSUITE Overview

We call our simulation and analysis software tool NEMSUITE. NEMSUITE is a MATLAB® SIMULINK® toolbox that enables users to build nanomechanical resonant sensing models in the SIMULINK® environment, and produce easily interpretable results and characterizations either through full simulation or fast analysis. In this way, expensive and time-consuming physical experiments can be conducted in a fully software environment.

NEMSUITE offers a custom block library that has several blocks corresponding to modules in a physical setup, e.g. a resonator, a demodulator, a controller, and noise sources. These custom blocks can be combined with any built-in SIMULINK® block in the models. Custom blocks of NEMSUITE are described in Chapter 4 in detail.

NEMSUITE models can be run in two distinct modes, namely the full simulation and the fast analysis. In the full simulation mode, SIMULINK® performs a numerical solution of the model. High accuracy can be achieved in this mode, although it may take several hours to run. The fast analysis mode offers approximate results in much shorter runtimes. In Chapter 4 we go through these two modes in more detail. And in Chapter 5 we present the results obtained by the full simulation and the fast analysis modes.

Chapter 2

THEORY

In this chapter, we analyze various components of nanomechanical resonant sensing schemes and obtain phase domain equivalent models. This chapter is built upon recent theoretical investigation of the nanomechanical resonant sensors [9, 1]. For integrity purposes relevant parts of these previous works are summarized in some sections.

A typical nanomechanical resonant sensing scheme consists of an oscillating core and a frequency detection module. We analyze various oscillating cores and frequency detection methods that have been or can be employed in nanomechanical resonant sensing schemes in physical experiment setups in the next two sections. The physical setups suffer from various noise sources, two of which are included in our treatment and examined in Section 2.3. Finally in Section 2.4, we present the phase domain equivalent models of the components we analyze together with the noise sources. The analysis in this chapter establishes the foundation for the next chapters.

2.1 Oscillatory Cores

Resonators lie at the heart of every nanomechanical resonant sensing setup, while there are different ways of obtaining oscillations from a resonator to use it as a sensing element. We call each of such different configurations of a resonator an oscillatory core, and our aim in this section is to examine different oscillatory cores. In the following subsections, we present and analyze three cores, namely the feedback-free resonator, self-oscillating core and hybrid core. Subsections on the feedback-free resonator and the self-oscillating core are based on Demir's recent work [1] and included in this thesis for completeness and a self-contained treatment.

Introducing the cores we briefly mention that the feedback-free resonator is modeled with a damped driven harmonic oscillator fed by a force excitation input and is also an essential component in other oscillatory cores. In the self-oscillating core the oscillations are generated with a positive feedback loop; the resonator is fed by a phase-shifted and amplified version of its output. The hybrid core [4] can be considered as a blend of the feedback-free resonator and self-oscillating core, that is the resonator in the hybrid core is fed by a linear combination of an external force excitation and phase-shifted and amplified version of its own output, similar to feedback-free resonator and the self-oscillating core respectively.

2.1.1 Feedback-free Resonator Model

This subsection is adapted from [1].

In this work we only consider resonators operating in the linear regime and use a damped harmonic oscillator model for the resonators as follows:

$$\frac{d^2}{dt^2}x(t) + \frac{\omega_r}{Q}\frac{d}{dt}x(t) + \omega_r^2 x(t) = \frac{F(t)}{m} \tag{2.1}$$

where ω_r is the resonance frequency, Q is the quality factor, m is the mass, finally $F(t)$ and $x(t)$ are the input and the output of the resonator and represent the force excitation and displacement respectively.

We obtain the baseband equivalent of the resonator by expressing the high frequency components with complex amplitude representation. We begin with $x(t)$ and express

$$x(t) = \Re\{s(t)e^{j\omega_o t}\} = \frac{1}{2}[s(t)e^{j\omega_o t} + s^*(t)e^{-j\omega_o t}], \tag{2.2}$$

where $.^*$ denotes a complex conjugate and $\Re\{.\}$ is the real part operator. ω_o is the carrier frequency which is close but not necessarily equal to the resonance frequency ω_r, holding $|\omega_r - \omega_o| < \frac{\omega_r}{Q} \ll \omega_r$. So, $s(t)$ is the slowly varying complex amplitude of $x(t)$, with a maximum bandwidth of $\frac{\omega_r}{Q}$. Substituting (2.2) into (2.1) we get

$$\frac{1}{2}[e^{j\omega_o t}\mathbf{S}(t) + e^{-j\omega_o t}\mathbf{S}^*(t)] = \frac{F(t)}{m}, \tag{2.3}$$

where

$$\mathbf{S}(t) = \frac{d^2}{dt^2}s(t) + \left(\frac{\omega_r}{Q} + 2j\omega_o\right)\frac{d}{dt}s(t) + \left(\omega_r^2 - \omega_o^2 + j\frac{\omega_r\omega_o}{Q}\right)s(t). \tag{2.4}$$

In (2.3), we see two sinusoidal components with frequencies ω_o and $-\omega_o$. To obtain the baseband equivalent of the resonator equation, we multiply both sides of (2.3) with $e^{-j\omega_o t}$ then integrate over a time interval of $\frac{2\pi}{\omega_o}$ expressed as

$$\frac{1}{2}\int_t^{t+\frac{2\pi}{\omega_o}} [\mathbf{S}(t') + e^{-j2\omega_o t'}\mathbf{S}^*(t')]dt' = \int_t^{t+\frac{2\pi}{\omega_o}} e^{-j\omega_o t'}\frac{F(t')}{m}dt'. \tag{2.5}$$

We assume a reasonably high Q for the resonator, resulting in a small bandwidth ($\frac{\omega_r}{Q}$) for $s(t)$ so that we approximate $s(t)$ as constant for one period of oscillations, which leads to

$$\mathbf{S}(t') \approx \mathbf{S}(t) \quad \text{for} \quad t \le t' \le t + \frac{2\pi}{\omega_o}. \tag{2.6}$$

Using (2.6) we evaluate the integral at the left-hand side of (2.5) and obtain

$$\frac{\pi}{\omega_o}\mathbf{S}(t) \approx \int_t^{t+\frac{2\pi}{\omega_o}} e^{-j\omega_o t'}\frac{F(t')}{m}dt'. \tag{2.7}$$

Substituting $\mathbf{S}(t)$ from (2.4) into (2.7) and multiplying both sides with $\frac{\omega_o}{\pi}$ we get

$$\begin{aligned}\frac{d^2}{dt^2}s(t) + \frac{\omega_r}{Q}\frac{d}{dt}s(t) + 2j\omega_o\frac{d}{dt}s(t) + \left(\omega_r^2 - \omega_o^2 + j\frac{\omega_r\omega_o}{Q}\right)s(t) \\ \approx \frac{\omega_o}{\pi}\int_t^{t+\frac{2\pi}{\omega_o}} e^{-j\omega_o t'}\frac{F(t')}{m}dt'.\end{aligned} \tag{2.8}$$

We observe that the first two terms in (2.8) have amplitudes Q times smaller than the rest of the terms in the left-hand side, because the derivative of $s(t)$ causes a maximum amplification of $\frac{\omega_r}{Q}$. Continuing the high-Q approximation we neglect these first two terms and multiply both sides with $-j\frac{Q}{\omega_r\omega_o}$ to get a first order equation as

$$\frac{2Q}{\omega_r}\frac{d}{dt}s(t) + \left(-j\frac{Q}{\omega_r\omega_o}(\omega_r^2 - \omega_o^2) + 1\right)s(t) \approx \frac{\omega_o}{\pi}\int_t^{t+\frac{2\pi}{\omega_o}} e^{-j\omega_o t'}\frac{F(t')}{m}dt'. \tag{2.9}$$

Since ω_o is close to ω_r

$$\omega_r^2 - \omega_o^2 \approx 2\omega_o(\omega_r - \omega_o), \tag{2.10}$$

which can be substituted into (2.9) to get

$$\frac{2Q}{\omega_r}\frac{d}{dt}s(t)+\left(j\frac{2Q}{\omega_r}(\omega_o-\omega_r)+1\right)s(t)\approx\frac{\omega_o}{\pi}\int_t^{t+\frac{2\pi}{\omega_o}}e^{-j\omega_o t'}\frac{F(t')}{m}dt'. \tag{2.11}$$

We define $\tau_r=\frac{2Q}{\omega_r}$ as the intrinsic resonator time constant so (2.11) can be expresses as

$$\tau_r\frac{d}{dt}s(t)+\left(j\tau_r(\omega_o-\omega_r)+1\right)s(t)\approx\frac{\omega_o}{\pi}\int_t^{t+\frac{2\pi}{\omega_o}}e^{-j\omega_o t'}\frac{F(t')}{m}dt'. \tag{2.12}$$

We express the force excitation $F(t)$ with complex amplitude representation similar to $x(t)$ in (2.2) as

$$F(t)=\Re\{f(t)e^{j\omega_o t}\}=\frac{1}{2}[f(t)e^{j\omega_o t}+f^*(t)e^{-j\omega_o t}], \tag{2.13}$$

where $f(t)$ is a slowly varying complex amplitude analogous to $s(t)$. Assuming

$$f(t')\approx f(t)\quad\text{for}\quad t\leq t'\leq t+\frac{2\pi}{\omega_o}, \tag{2.14}$$

similarly to the assumptions for $\mathbf{S}(t)$ in (2.6), we substitute (2.13) into (2.12) and obtain

$$\tau_r\frac{d}{dt}s(t)+\left(1+j\tau_r(\omega_o-\omega_r)\right)s(t)\approx\frac{-j}{m}\frac{Q}{\omega_r\omega_o}f(t). \tag{2.15}$$

We can use (2.15) to evaluate the steady-state amplitude and phase response of the resonator when excited with a sinusoidal signal. For a excitation $F(t)=A_f cos(\omega_o t+\theta_f)$, the corresponding complex amplitude becomes $f(t)=A_f e^{\theta_f}$. The steady-state response can be obtained by substituting $f(t)$ into (2.15) and setting the time derivative of $s(t)$ to zero as

$$s(t)=\frac{-j}{m}\frac{Q}{\omega_r\omega_o}\frac{A_f e^{j\theta_f}}{[1+j\tau_r(\omega_o-\omega_r)]}=A_r e^{j\theta_r}, \tag{2.16}$$

where

$$A_r=A_f\frac{Q}{m\omega_r\omega_o}\frac{1}{\sqrt{1+\tau_r^2(\omega_o-\omega_r)^2}} \tag{2.17}$$

and

$$\theta_r=\theta_f-\frac{\pi}{2}-atan(\tau_r(\omega_o-\omega_r)), \tag{2.18}$$

The equations above imply a sinusoidal response $x(t)=A_r cos(\omega_o t+\theta_r)$ for the resonator, meaning that the resonator oscillates at the same frequency as the input (ω_o), with its amplitude and phase depending on the frequency difference ($\omega_o-\omega_r$).

2.1.2 Resonator Phase Response

This subsection is adapted from [1].

In resonance frequency based nanomechanical resonant sensors, an event of interest cause to a shift in the resonance frequency. The amount of shift in the resonance frequency can be calculated using phase of the resonator's output since the phase response of the resonator depends on the difference between the resonance frequency and the frequency of the force excitation as shown in the previous section. In fact the phase response is preferred for measuring frequency shifts due to its sharpness around the resonance frequency.

We consider a case that at $t = 0$ the resonance frequency jumps from ω_{r1} to ω_{r2} that is

$$\omega_r(t) = \begin{cases} \omega_{r1}, & t < 0, \\ \omega_{r2}, & t \geq 0, \end{cases} \tag{2.19}$$

and we assume that $\omega_{r1} = \omega_o$. We approximate the amplitude of the resonator response around the resonance frequency as constant so using (2.17)

$$A_{rss} = A(t) = A_f \frac{Q}{m\omega_r^2} \tag{2.20}$$

then the complex amplitude of the output of the resonator becomes

$$s(t) = A_{rss} e^{j\theta_r(t)}. \tag{2.21}$$

We express the phase of the resonator's response for $t < 0$ using (2.18) as

$$\theta_r(t) = \theta_f - \frac{\pi}{2} - atan(\tau_r(\omega_o - \omega_{r1})) = \theta_f - \frac{\pi}{2} \quad \text{for} \quad t < 0. \tag{2.22}$$

To find the phase response of the resonator for $t \geq 0$ we use (2.15) and substitute $s(t) = A_{rss}e^{j\theta_r}$, $f(t) = A_f e^{j\theta_f}$, $\omega_o = \omega_{r1}$ and $\omega_r = \omega_{r2}$ as follows

$$\tau_r \frac{d}{dt} A_{rss} e^{j\theta_r} + (1 + j\tau_r(\omega_{r1} - \omega_{r2}))\, A_{rss} e^{j\theta_r} \approx \frac{-j}{m} \frac{Q}{\omega_{r1}\omega_{r2}} f(t). \tag{2.23}$$

2.1.3 Self-oscillating Core Model

This subsection is adapted from [1].

In the self-oscillating core, the resonator is fed by an amplified and phase-shifted version of its output as shown in Figure 2.1. Therefore the force excitation $F(t)$ in the resonator model we have obtained in subsection 2.1.1 is set as

$$F(t) = e^{j\frac{\pi}{2}} h(x(t)) \tag{2.24}$$

where $h(.)$ is a nonlinear tanh-like odd function, $x(t)$ is the output of the resonator. With complex-amplitude representation we obtain the baseband equivalent equation for the self-oscillating core using (2.12) as

$$\tau_r \frac{d}{dt} s(t) + [1 + j\tau_r(\omega_o - \omega_r)]s(t) = \frac{Q}{m\omega_o\omega_r} h_D(|s(t)|)s(t), \tag{2.25}$$

where $s(t)$ is the complex amplitude of the resonator's output $x(t)$ as in (2.2) and $h_D(.)$ is a nonlinear function related but distinct from $h(.)$ known as a describing function.

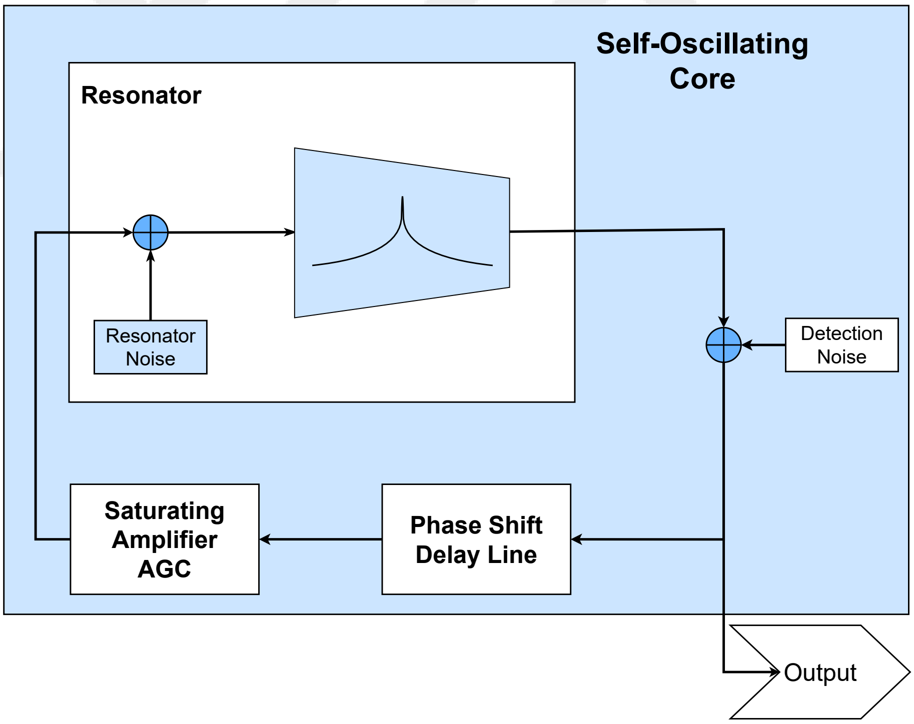

Figure 2.1: A diagram of the self-oscillating core.

Assuming a steady-state solution in the form $s(t) = A_r e^{j\theta_r}$ with time-invariant A_r and θ_r, (2.25) becomes

$$[1 + j\tau_r(\omega_o - \omega_r)]A_r e^{j\theta_r} = \frac{Q}{m\omega_o\omega_r} h_D(A_r)A_r e^{j\theta_r}. \tag{2.26}$$

We can divide both sides in (2.26) with $e^{j\theta_r}$ and get

$$[1 + j\tau_r(\omega_o - \omega_r)]A_r = \frac{Q}{m\omega_o\omega_r} h_D(A_r)A_r, \tag{2.27}$$

which means that there is no restriction on the value of θ_r; it is determined by the initial conditions. Separating the real and the imaginary parts of (2.27):

$$A_r = \frac{Q}{m\omega_o\omega_r} h_D(A_r)A_r \tag{2.28}$$

and

$$\tau_r(\omega_o - \omega_r)A_r = 0. \tag{2.29}$$

The imaginary part leads to $\omega_o = \omega_r$, meaning that the self-oscillating core oscillates at the resonance frequency (ω_r). $A_r = 0$ solution for the real part is unstable due to the positive feedback and the noise. Dividing both sides of the real part with A_r we get

$$1 = \frac{Q}{m\omega_o\omega_r} h_D(A_r), \tag{2.30}$$

solution of which gives the amplitude of the oscillations.

2.1.4 Hybrid Core Oscillator Model

In the hybrid core [4], the resonator is fed by both an external signal and the phase-shifted and amplified version of its output. The two signals are linearly combined in a device called balance network which multiplies the signals with variable coefficients. Figure 2.2 shows a diagram of the hybrid core.

We use the resonator model that we have obtained in subsection 2.1.1, setting the force excitation $F(t)$ as follows:

$$F(t) = \beta e^{j\frac{\pi}{2}} h(x(t)) + \alpha F_{ext}(t), \tag{2.31}$$

where $h(.)$ is the same nonlinear odd function that appeared in the self-oscillating core, $x(t)$ is the output of the resonator $F_{ext}(t)$ is the external force excitation, α and β are the linear combination coefficients in the balance network for the external

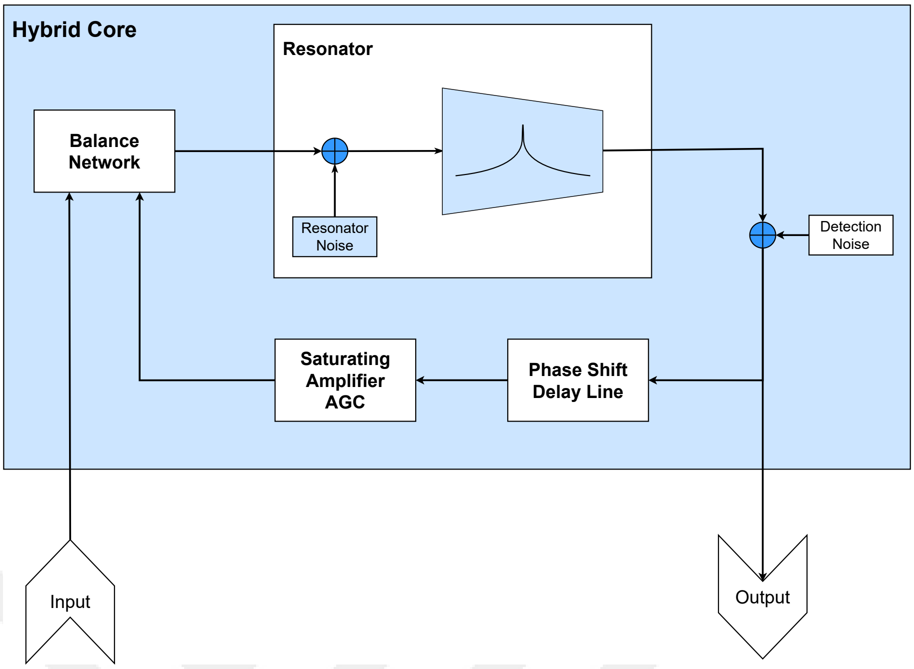

Figure 2.2: A diagram of the hybrid core.

input and the feedback signal respectively. We express $x(t)$ and $F_{ext}(t)$ in complex amplitude form as

$$x(t) = \Re\left(s(t)e^{j\omega_o t}\right) = \frac{1}{2}[s(t)e^{j\omega_o t} + s^*(t)e^{-j\omega_o t}] \tag{2.32}$$

and

$$F_{ext}(t) = \Re\left(f_{ext}(t)e^{j\omega_o t}\right) = \frac{1}{2}[f_{ext}(t)e^{j\omega_o t} + f_{ext}^*(t)e^{-j\omega_o t}], \tag{2.33}$$

where $s(t)$ and $f_{ext}(t)$ are the slowly varying complex amplitudes of $x(t)$ and $F_{ext}(t)$. With (2.31), (2.32) and (2.33), we obtain a baseband equivalent equation for the hybrid core using (2.12) of the feedback-free resonator model as

$$\tau_r \frac{d}{dt}s(t) + [1 + j\tau_r(\omega_o - \omega_r)]s(t) = \frac{Q}{m\omega_o\omega_r}(\beta h_D(|s(t)|)s(t) - j\alpha f_{ext}(t)), \tag{2.34}$$

where τ_r is the intrinsic time constant of the resonator, $h_D(.)$ is a describing function which behaves as a input dependent gain like in the self-oscillating core.

We assume a steady-state sinusoidal solution $s(t)$ and input $f_{ext}(t)$ to (2.34) as

$$s(t) = A_r e^{j\theta_r} \tag{2.35}$$

and

$$f_{ext}(t) = A_o e^{j\theta_o} \tag{2.36}$$

with time-invariant A_r, θ_r, A_o and θ_o. Substituting $s(t)$ and $f_{ext}(t)$ in (2.35) and (2.36) into (2.34) we get

$$[1 + j\tau_r(\omega_o - \omega_r)]A_r e^{j\theta_r} = \frac{Q}{m\omega_o\omega_r}(\beta h_D(A_r)A_r e^{j\theta_r} - j\alpha A_o e^{j\theta_o}), \tag{2.37}$$

dividing both sides with $e^{j\theta_r}$

$$[1 + j\tau_r(\omega_o - \omega_r)]A_r = \frac{Q}{m\omega_o\omega_r}(\beta h_D(A_r)A_r + \alpha A_o e^{j(\theta_o - \theta_r - \frac{\pi}{2})}). \tag{2.38}$$

We equate the real and the imaginary parts of equation (2.38) as

$$A_r = \frac{Q}{m\omega_o\omega_r}(\beta h_D(A_r)A_r + \alpha A_o cos(\theta_o - \theta_r - \frac{\pi}{2})) \tag{2.39}$$

and

$$\tau_r(\omega_o - \omega_r)A_r = \frac{Q}{m\omega_o\omega_r}(\alpha A_o sin(\theta_o - \theta_r - \frac{\pi}{2})). \tag{2.40}$$

Substituting $\frac{Q}{m\omega_o\omega_r}$ from (2.39) into (2.40) we get

$$\tau_r(\omega_o - \omega_r) = \frac{\alpha A_o sin(\theta_o - \theta_r - \frac{\pi}{2})}{\beta h_D(A_r)A_r + \alpha A_o cos(\theta_o - \theta_r - \frac{\pi}{2})}. \tag{2.41}$$

Equation (2.41) describes the relationship between the frequency difference $(\omega_o - \omega_r)$ and the phase difference at the resonator's response $(\theta_o - \theta_r)$ in the hybrid core. It becomes equivalent to self-oscillating core equations and to feedback-free resonator equations when $(\alpha = 0, \beta = 1)$ and $(\alpha = 1, \beta = 0)$ respectively, meaning that the hybrid core can be configured to behave like self-oscillating core or feedback-free resonator by adjusting the linear combination parameters. When $\beta > 0$, there is no restriction on the oscillation frequency, yet we assume in this subsection that both the external drive and the resonator output has the same frequency ω_o. In the next subsection, we validate this assumption by calculating the frequency of resonator response when fed with an arbitrary frequency.

2.1.5 Hybrid Core Frequency of Oscillations

In this section we calculate the frequency of the resonator response without any assumption on the frequency of the external drive. We express the frequency of the external signal with ω_o while using an unknown $\omega_x$for the frequency of the resonator response and obtain the relationship between ω_o and ω_x. We use the baseband equation (2.34) of the hybrid core and choose a steady state solution for the resonator output as

$$s(t) = A_r e^{j((\omega_x - \omega_o)t + \theta_r)}, \tag{2.42}$$

with time-invariant amplitude A_r but time-variant phase $(\omega_x - \omega_o)t + \theta_r$ to have a frequency of ω_x at the resonator's output while choosing a complex amplitude for the external drive as

$$f_{ext}(t) = A_o e^{j\theta_o}. \tag{2.43}$$

Substituting $s(t)$ and $f_{ext}(t)$ from the equations above into (2.34) we get

$$\begin{aligned} j\tau_r A_r(\omega_x - \omega_o)e^{j((\omega_x - \omega_o)t + \theta_r)} + [1 + j\tau_r(\omega_o - \omega_r)]A_r e^{j((\omega_x - \omega_o)t + \theta_r)} \\ = \frac{Q}{m\omega_o\omega_r}(\beta h_D(A_r)A_r e^{j((\omega_x - \omega_o)t + \theta_r)} - j\alpha A_o e^{j\theta_o}), \end{aligned} \tag{2.44}$$

dividing both sides with $e^{j((\omega_x - \omega_o)t + \theta_r)}$ we get

$$\begin{aligned} j\tau_r A_r(\omega_x - \omega_o) + [1 + j\tau_r(\omega_o - \omega_r)]A_r \\ = \frac{Q}{m\omega_o\omega_r}(\beta h_D(A_r)A_r - j\alpha A_o e^{-j((\omega_x - \omega_o)t + \theta_r - \theta_o))}), \end{aligned} \tag{2.45}$$

after some manipulations:

$$j\tau_r A_r(\omega_x - \omega_r) + A_r = \frac{Q}{m\omega_o\omega_r}(\beta h_D(A_r)A_r - j\alpha A_o e^{-j((\omega_x - \omega_o)t + \theta_r - \theta_o))}). \tag{2.46}$$

We observe that, the left-hand side of (2.46) is time-invariant while there is time-dependency on the right-hand side with a frequency of $(\omega_x - \omega_o)$. Therefore, we say that the equality $\omega_x = \omega_o$ must hold when $\alpha > 0$ to avoid the time dependence on the right-hand side. This means that the frequency of the resonator response is equal to the frequency of the external drive in the hybrid core. For the case where $\alpha = 0$, there is no external drive and equation (2.46) becomes identical to self-oscillating core equations implying that the frequency of the resonator response is equal to the resonance frequency ($\omega_x = \omega_r$). On the other hand, when α is negligibly small the external drive may possibly not be able to change the oscillation frequency of the resonator, in other words, the hybrid core oscillator may not lock to the injection frequency. We explore the injection lock thoroughly in the next subsection.

2.1.6 Injection Lock in the Hybrid Core

In the previous subsection, we have shown that the oscillation frequency of the hybrid core is equal to the frequency of the external drive. This phenomenon is called injection lock [11, 12, 13], as an oscillator (the hybrid core) locks to the frequency of the injection signal. Our aim in this subsection is to examine the injection lock for the hybrid core.

Similar to the previous subsection, we use the baseband equation (2.34) here. For the output of the resonator we assume a time-invariant amplitude but time-variant phase for the steady state as in

$$s(t) = A_r e^{j\theta_r(t)}, \tag{2.47}$$

and an external drive with time-invariant amplitude and phase as in

$$f_{ext} = A_o e^{j\theta_o}. \tag{2.48}$$

Substituting the expressions above for $s(t)$ and f_{ext} in (2.34) we get after some manipulations

$$\begin{aligned} j\tau_r A_r e^{j\theta_r(t)} \frac{d}{dt}\theta_r(t) + [1 + j\tau_r(\omega_o - \omega_r)]A_r e^{j\theta_r(t)} \\ = \frac{Q}{m\omega_o\omega_r}(\beta h_D(A_r)A_r e^{j\theta_r(t)} - j\alpha A_o e^{j\theta_o}). \end{aligned} \tag{2.49}$$

Dividing both sides with $e^{j\theta_r(t)}$ the equation above becomes

$$j\tau_r A_r \frac{d}{dt}\theta_r(t) + [1 + j\tau_r(\omega_o - \omega_r)]A_r = \frac{Q}{m\omega_o\omega_r}(\beta h_D(A_r)A_r - j\alpha A_o e^{j(\theta_o - \theta_r)}). \tag{2.50}$$

We separate the real and the imaginary parts of (2.50) as

$$A_r = \frac{Q}{m\omega_o\omega_r}(\beta h_D(A_r)A_r - \alpha A_o cos(\theta_o - \theta_r + \frac{\pi}{2})) \tag{2.51}$$

and

$$A_r\tau_r\frac{d}{dt}\theta_r(t) + A_r\tau_r(\omega_o - \omega_r) = -\frac{Q}{m\omega_o\omega_r}\alpha A_o sin(\theta_o - \theta_r + \frac{\pi}{2})). \qquad (2.52)$$

We manipulate the equation (2.51) to obtain an expression for $\frac{Q}{m\omega_o\omega_r}$ and substitute it into (2.52) and get

$$\tau_r\frac{d}{dt}\theta_r(t) + \tau_r(\omega_o - \omega_r) = -\frac{\alpha A_o sin(\theta_o - \theta_r + \frac{\pi}{2})}{\beta h_D(A_r)A_r - \alpha A_o cos(\theta_o - \theta_r + \frac{\pi}{2})} \qquad (2.53)$$

We consider the case where the amplitude of the injection signal is negligibly small compared to the feedback input to the resonator that is

$$\alpha A_o \ll \beta h_D(A_r)A_r \qquad (2.54)$$

which enables us to make the following approximation

$$\beta h_D(A_r)A_r - \alpha A_o cos(\theta_o - \theta_r + \frac{\pi}{2}) \approx \beta h_D(A_r)A_r. \qquad (2.55)$$

With the approximation above (2.53) becomes

$$\tau_r\frac{d}{dt}\theta_r(t) + \tau_r(\omega_o - \omega_r) = -\frac{\alpha A_o sin(\theta_o - \theta_r + \frac{\pi}{2})}{\beta h_D(A_r)A_r}, \qquad (2.56)$$

and after some manipulations we get

$$\frac{d}{dt}\theta_r(t) = \omega_r - \omega_o - \frac{\omega_r}{2Q}\frac{\alpha A_o}{\beta h_D(A_r)A_r}sin(\theta_o - \theta_r + \frac{\pi}{2}), \qquad (2.57)$$

notice that we have substituted τ_r with its equivalent $\frac{2Q}{\omega_r}$. Equation (2.50) is called Adler's equation [12] which is proposed by Adler to study injection locking in oscillators, we have obtained it through baseband analysis of the hybrid core.

For the injection locked state, phase of the resonator has to be constant, meaning that the derivative on the left-hand side of (2.57) is zero:

$$0 = \omega_r - \omega_o - \frac{\omega_r}{2Q}\frac{\alpha A_o}{\beta h_D(A_r)A_r}sin(\theta_o - \theta_r + \frac{\pi}{2}), \qquad (2.58)$$

equivalently

$$\omega_r - \omega_o = \frac{\omega_r}{2Q}\frac{\alpha A_o}{\beta h_D(A_r)A_r}sin(\theta_o - \theta_r + \frac{\pi}{2}). \qquad (2.59)$$

Amplitude of sine never exceeds 1 ($|sin(.)| \leq 1$), therefore equation (2.59) implies bounds for the frequency difference ($\omega_r - \omega_o$) as

$$|\omega_r - \omega_o| \leq \frac{\omega_r}{2Q} \frac{\alpha A_o}{\beta h_D(A_r) A_r}, \tag{2.60}$$

which can be interpreted as for injection locking to occur, injection frequency (ω_o) must be close to the resonance frequency (ω_r). Amplitudes of the injection signal (αA_o) and the resonator feedback ($\beta h_D(A_r) A_r$) determines how close the two frequencies must be for injection lock. Nevertheless, there is the assumption (2.54) that enabled us to go from (2.53) to (2.56), which is the injection amplitude is very small compared to the resonator's feedback. As we have seen in the previous section there is no frequency difference limit when the injection amplitude is at least comparable with the resonator's feedback. The inequality that is expressed in (2.60) is valid for negligibly small injection amplitudes.

2.1.7 Hybrid Core Phase Response

Similar to feedback-free resonator and self-oscillating core, we calculate the phase response of the hybrid core to shifts in the resonance frequency. We assume an external drive (injection signal) as

$$f_{ext} = A_o e^{j\theta_o} \tag{2.61}$$

with time-invariant amplitude and phase and that the resonance frequency jumps from ω_{r1} to ω_{r2} at $t = 0$ that is

$$\omega_r(t) = \begin{cases} \omega_{r1}, & t < 0, \\ \omega_{r2}, & t \geq 0. \end{cases} \tag{2.62}$$

also that the frequency of the injection signal is equal to the resonance frequency at $t < 0$ that is

$$\omega_o = \omega_{r1}, \quad \text{for all } t. \tag{2.63}$$

Then we have a steady state solution of

$$s(t) = A_{r1} e^{j\theta_{r1}} \tag{2.64}$$

for $t < 0$ where A_{r1} is the solution of

$$A_{r1} = \frac{Q}{m\omega_{r1}^2}(\beta h_D(A_{r1})A_{r1} + \alpha A_o) \tag{2.65}$$

and

$$\theta_{r1} = \theta_o - \frac{\pi}{2} \tag{2.66}$$

which can be obtained using equations (2.39), (2.40) and (2.41).

To find the transient phase response of the hybrid core to jumps in the resonance frequency, we use equation (2.34) substituting ω_o with ω_{r1}, ω_r with ω_{r2}, and f_{ext} as in (2.61), also assuming a solution in the form of $s(t) = A_r(t)e^{j\theta_r(t)}$:

$$\begin{aligned}\tau_r \frac{d}{dt} A_r(t) e^{j\theta_r(t)} + [1 + j\tau_r(\omega_{r1} - \omega_{r2})]A_r(t)e^{j\theta_r(t)} \\ = \frac{Q}{m\omega_{r1}\omega_{r2}}(\beta h_D(A_r)A_r(t)e^{j\theta_r(t)} - j\alpha A_o e^{j\theta_o}),\end{aligned} \tag{2.67}$$

evaluating the derivative at the left-hand side and dividing both sides with $e^{j\theta_r(t)}$

$$\begin{aligned}\tau_r \frac{d}{dt} A_r(t) + j\tau_r A_r(t) \frac{d}{dt}\theta_r(t) + [1 + j\tau_r(\omega_{r1} - \omega_{r2})]A_r(t) \\ = \frac{Q}{m\omega_{r1}\omega_{r2}}(\beta h_D(A_r)A_r(t) - \alpha A_o e^{j(\frac{\pi}{2} + \theta_o - \theta_r(t))}).\end{aligned} \tag{2.68}$$

Separating the real and the imaginary parts of the equation above we get

$$\begin{aligned}\tau_r \frac{d}{dt} A_r(t) + A_r(t) \\ = \frac{Q}{m\omega_{r1}\omega_{r2}}(\beta h_D(A_r)A_r(t) - \alpha A_o cos(\frac{\pi}{2} + \theta_o - \theta_r(t)))\end{aligned} \tag{2.69}$$

and

$$\begin{aligned}\tau_r A_r(t) \frac{d}{dt}\theta_r(t) + \tau_r(\omega_{r1} - \omega_{r2})A_r(t) \\ = -\frac{Q}{m\omega_{r1}\omega_{r2}}\alpha A_o sin(\frac{\pi}{2} + \theta_o - \theta_r(t)).\end{aligned} \tag{2.70}$$

We observe that equations (2.69) and (2.70) are coupled nonlinear differential equations and get simplified to self-oscillating core equations when $\alpha = 0, \beta = 1$ and to feedback-free resonator equations when $\alpha = 1, \beta = 0$. Here we make two assumptions: the amplitude is stable and not changing with time ($A_r(t) \approx A_r$) and the phase difference induced by the resonator is very close to $-\frac{\pi}{2}$ that is $(\theta_r(t)-\theta_o \approx -\frac{\pi}{2})$. The phase assumption enables us to make the following approximation

$$cos(\frac{\pi}{2}+\theta_o-\theta_r(t)) = sin(\theta_r(t)-\theta_o) \approx sin(-\frac{\pi}{2}) = -1. \tag{2.71}$$

Applying the approximations, (2.69) becomes

$$A_r = \frac{Q}{m\omega_{r1}\omega_{r2}}(\beta h_D(A_r)A_r + \alpha A_o). \tag{2.72}$$

We obtain an expression for $\frac{Q}{m\omega_{r1}\omega_{r2}}$ from (2.72) to substitute it into (2.70) and get

$$\tau_r \frac{d}{dt}\theta_r(t) + \tau_r(\omega_{r1}-\omega_{r2}) = -\frac{\alpha A_o sin(\frac{\pi}{2}+\theta_o-\theta_r(t))}{\beta h_D(A_r)A_r + \alpha A_o}. \tag{2.73}$$

Argument of the sine at the right-hand side of the equation above is very small due to the phase assumption that we have made above $(\theta_r(t)-\theta_o \approx -\frac{\pi}{2})$. Therefore, we make the following approximation:

$$sin(\frac{\pi}{2}+\theta_o-\theta_r(t)) = cos(\theta_r(t)-\theta_o) \approx \theta_r(t)-\theta_o+\frac{\pi}{2} \tag{2.74}$$

then (2.73) becomes

$$\tau_r \frac{d}{dt}\theta_r(t) + \tau_r(\omega_{r1}-\omega_{r2}) = -\frac{\alpha A_o(\theta_r(t)-\theta_o+\frac{\pi}{2})}{\beta h_D(A_r)A_r + \alpha A_o}. \tag{2.75}$$

We define injection ratio R_{inj} as the ratio of the amplitude of the injection signal to the total amplitude of the input to the resonator, that is

$$R_{inj} = \frac{\alpha A_o}{\beta h_D(A_r)A_r + \alpha A_o} \tag{2.76}$$

then we write (2.75) as

$$\tau_r \frac{d}{dt}\theta_r(t) + \tau_r(\omega_{r1}-\omega_{r2}) = -R_{inj}(\theta_r(t)-\theta_o+\frac{\pi}{2}). \tag{2.77}$$

When $\alpha = 0$ there is no injection and injection ratio equals to zero ($R_{inj} = 0$), then (2.77) becomes

$$\frac{d}{dt}\theta_r(t) = -(\omega_{r1}-\omega_{r2}), \tag{2.78}$$

evaluating the derivative

$$\theta_r(t) = -(\omega_{r1}-\omega_{r2})t + \theta_{r1}, \tag{2.79}$$

We write the phase of the resonator response as

$$\theta_r(t) = \Delta\theta_r(t) + \theta_{r1} \tag{2.80}$$

then

$$\Delta\theta_r(t) = -(\omega_{r1} - \omega_{r2})t, \tag{2.81}$$

which is the phase response of the self-oscillating core, meaning that our analysis of the hybrid core captures the self-oscillating core. Next we assume that $\alpha \neq 0$, divide both sides of (2.77) with R_{inj}, and substitute $\theta_r(t)$ as in (2.80) and (2.66)

$$\frac{\tau_r}{R_{inj}}\frac{d}{dt}\Delta\theta_r(t) + \Delta\theta_r(t) = -\frac{\tau_r}{R_{inj}}(\omega_{r1} - \omega_{r2}) \tag{2.82}$$

The equation above describes the change in the phase of the resonator's response to change in resonance frequency as the output of a low-pass filter with a time constant $\frac{\tau_r}{R_{inj}}$ when the input is the frequency difference multiplied with the time constant. We define the time constant of the hybrid core as

$$\tau_h = \frac{\tau_r}{R_{inj}} \tag{2.83}$$

then (2.82) becomes

$$\tau_h\frac{d}{dt}\Delta\theta_r(t) + \Delta\theta_r(t) = -\tau_h(\omega_{r1} - \omega_{r2}). \tag{2.84}$$

We note that when $\beta = 0$, $R_{inj} = 1$ and consequently $\tau_h = \tau_r$ making (2.84) equivalent to the feedback-free resonator equation.

2.2 Frequency Detection Methods

In resonance frequency based sensing, any event of interest causes a shift in the resonance frequency of the resonator. There are multiple techniques to track the resonance frequency. In this section we go over three of them, namely demodulator, phase-locked loop and negative feedback loop.

2.2.1 Demodulator

The demodulator we use in nanomechanical resonant sensing models can be considered as a phase detector. The output of the demodulator is the phase difference of its two input signals. The phase difference can be mapped to difference in the resonance frequency according to the characteristics of the feedback-free resonator, self-oscillating core and hybrid core.

The demodulator comprises two mixers, two low-pass filters and a nonlinearity. Figure 2.3 shows a diagram of the demodulator.

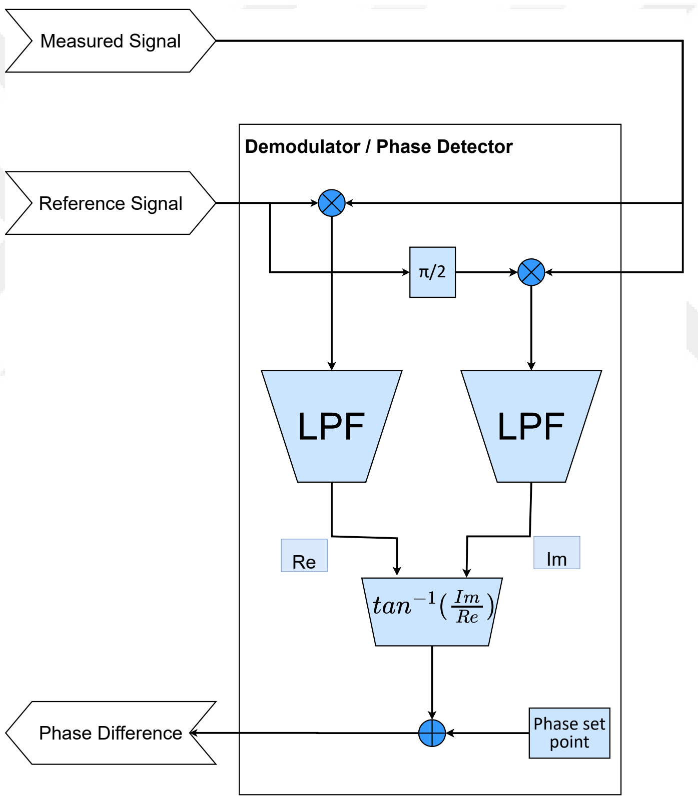

Figure 2.3: A diagram of the demodulator.

2.2.2 Phase-locked Loop (PLL)

A phase-locked loop is a negative feedback loop consisting of a demodulator, a controller and a controlled oscillator. The demodulator works as a phase difference detector and the controller converts the phase difference signal into the frequency error signal. The oscillator is controlled by this frequency error signal thereby having an output signal with the same frequency of the input.

Phase-locked loop can be used to track the resonator's frequency of oscillations, while this frequency may be different from the resonance frequency in feedback-free resonator and hybrid core. On the other hand as we have calculated in the previous section, the self-oscillating core oscillates at the resonance frequency. Thus, as a resonance frequency detection method, phase-locked loop is applicable only to self-oscillating core. Figure 2.4 shows a diagram of the phase-locked loop.

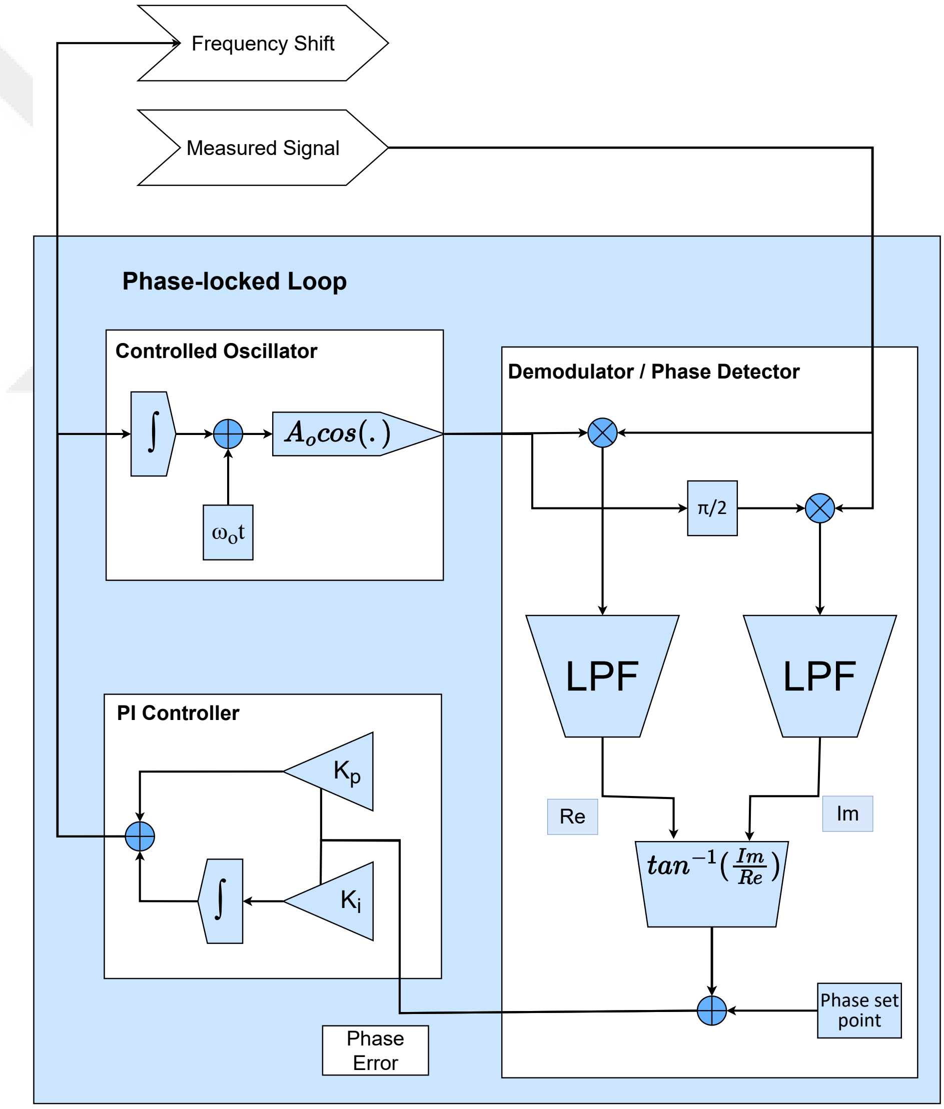

Figure 2.4: A diagram of the phase-locked loop.

2.2.3 Frequency-locked Loop (FLL)

As we have seen in the previous subsection, the PLL contains a controlled oscillator that has the same frequency at its output as its input. The controlled oscillator of the PLL can be used to drive the resonator so that the negative feedback loop guarantees the equality of the oscillation frequency and the resonance frequency. The frequency error signal at the output of the controller can be used to track the resonance frequency. This technique is applicable to the feedback-free resonator and the hybrid core. The figure in Fig. 2.5 shows a diagram of the frequency-locked loop without a resonator.

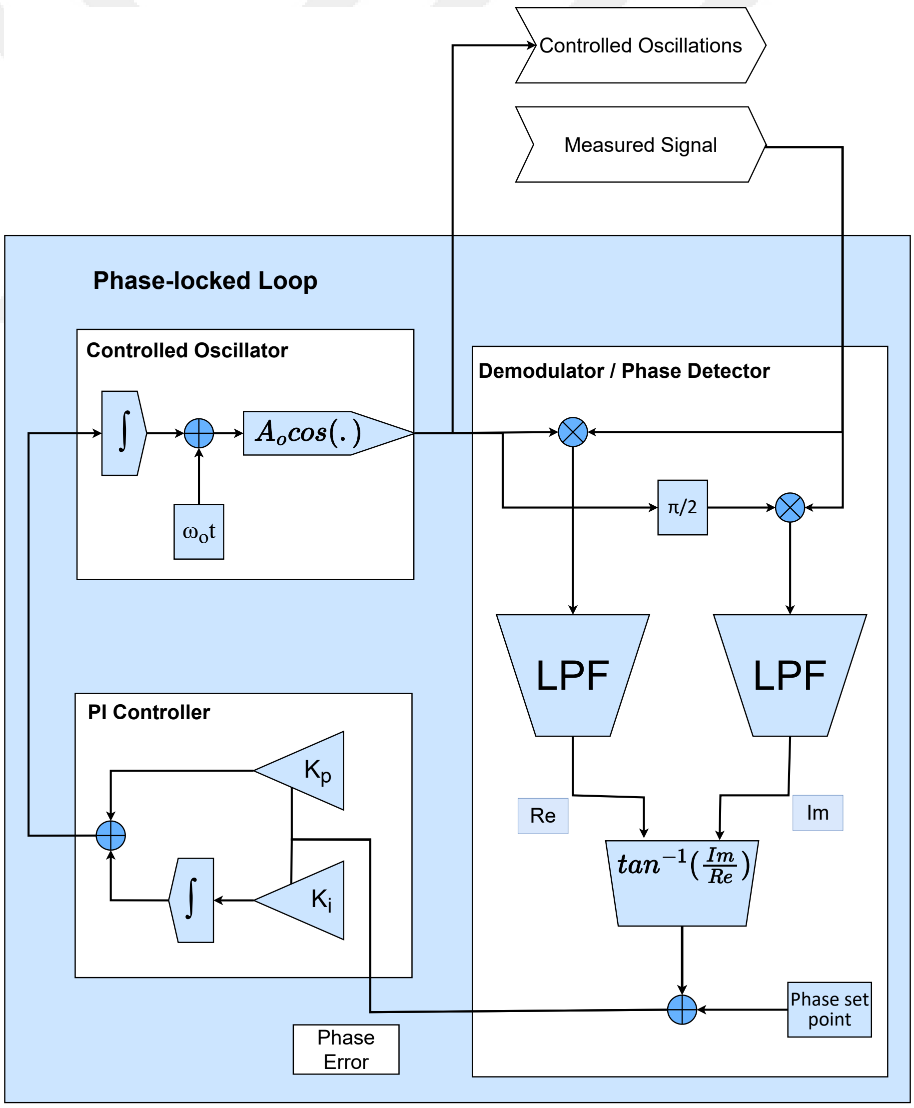

Figure 2.5: A diagram of the negative feedback loop.

2.3 Noise

In nanomechanical resonant sensor setups noise is the ultimate unavoidable cause of the performance limitation. We include two noise sources in our analysis, namely the thermomechanical noise and the detection noise. The thermomechanical noise is caused by the intrinsic thermal properties of the resonator while the detection noise comprises all noise signals produced during transduction and measurement stages.

2.3.1 Thermomechanical Noise

We model the thermomechanical noise of the resonator as a white noise source at the external drive, with a (two-sided) spectral density,

$$S_{thm}(\omega) = 2m\frac{\omega_r}{Q}k_B T, \tag{2.85}$$

where k_B is Boltzmann's constant and T is temperature [1]. Similar to the input and output signals of the resonator we use complex amplitude representation to the thermomechanical noise

$$N_{thm}(t) = \Re\{n_{thm}(t)e^{j\omega_o t}\} \tag{2.86}$$

where $n_{thm}(t)$ is the complex amplitude of the noise process $N_{thm}(t)$, and can be expressed as

$$n_{thm}(t) = n_{thm\Re}(t) + jn_{thm\Im}(t) \tag{2.87}$$

where $n_{thm\Re}(t)$ and $n_{thm\Im}(t)$ are uncorrelated white noise sources having twice of the spectral density in (2.85).

2.3.2 Detection Noise

Detection noise contains all noise sources that arise during detection stages e.g. transduction and amplification. We model the detection noise as white noise source similar to the thermomechanical noise but added at the output of the resonator. Using complex amplitude representation

$$N_d(t) = \Re\{n_d(t)e^{j\omega_o t}\} \tag{2.88}$$

and

$$n_d(t) = n_{d\Re}(t) + jn_{d\Im}(t). \tag{2.89}$$

2.3.3 Noise in the Self-Oscillating Core

The two noise sources are introduced into the self-oscillating core model equation (2.25) as

$$\tau_r \frac{d}{dt}s(t)+[1+j\tau_r(\omega_o-\omega_r)]s(t) = \frac{Q}{m\omega_o\omega_r}(h_D(|s(t)|)(s(t)+n_d(t))+n_{thm}(t)). \tag{2.90}$$

Separating the real/imaginary parts and assuming a time-invariant amplitude A_{rss} at the resonator's output we obtain

$$\frac{d}{dt}\Delta\theta_r(t) = \frac{1}{\tau_r}\frac{1}{A_{rss}}\frac{Q}{m\omega_r^2}(h_D(A_{rss})n_{d\Im}(t) + n_{thm\Im}(t)). \tag{2.91}$$

2.3.4 Noise in the Hybrid Core

We add the two noise sources into the hybrid core model by inserting $n_{thm}(t)$ and $n_d(t)$ into (2.34) as

$$\tau_r \frac{d}{dt}s(t)+[1+j\tau_r(\omega_o-\omega_r)]s(t) = \frac{Q}{m\omega_o\omega_r}(\beta h_D(|s(t)|)(s(t)+n_d(t))+n_{thm}(t)-j\alpha f_{ext}(t)). \tag{2.92}$$

Applying the same mathematical procedure as in the previous subsections, we obtain the following equation from (2.92)

$$\tau_h \frac{d}{dt}\Delta\theta_r(t) + \Delta\theta_r(t) = \frac{1}{\alpha A_{co}}(\beta h_D(A_{rss})n_{d\Im}(t) + n_{thm\Im}(t)) \tag{2.93}$$

which describes the change in the resonator's phase caused by the noise sources as the noise processes are applied to the same one-pole filter that we have found in the previous subsections.

2.4 Phase Domain Models

In this section we summarize the results obtained in this chapter in the form of so-called phase domain models of the oscillatory cores together with noise sources, and the frequency detection modules. These simplified and approximated versions of the modules are called to be in phase domain as they contain only the phase information through which the resonance frequency affected by the noise sources or an event of interest can be characterized.

Phase domain models are obtained by linearizing nonlinear models around an operating point. Then the effect of noise and small shifts in the resonance frequency of the resonator are described. This method is known as small signal analysis in electronics and used to linear(ized) equivalents of nonlinear devices like transistor around an operating point [14].

2.4.1 Resonator

The phase domain equivalent of the resonator is $H_R(s) = \frac{1}{\tau_r s+1}$, where $\tau_r = \frac{2Q}{\omega_r}$ [1]. The two noise sources are multiplied with some constants. Figure 2.6 shows a diagram of the phase domain equivalent of the resonator including the noise sources.

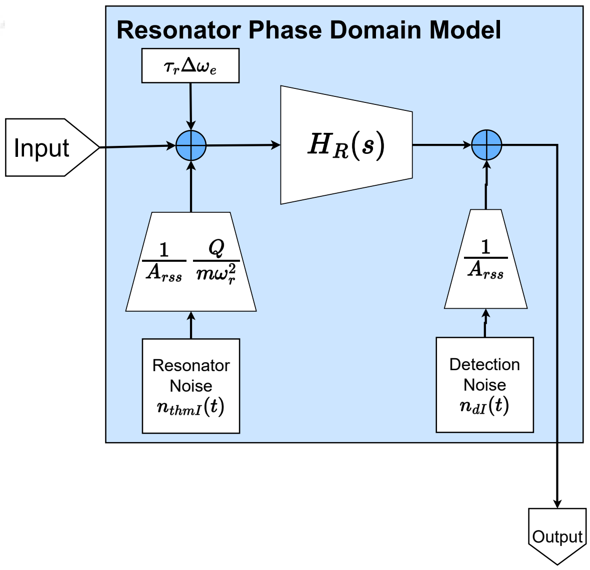

Figure 2.6: A diagram of the phase domain equivalent of the resonator.

2.4.2 Self-Oscillating Core

The phase domain equivalent of the self-oscillating core is just an integrator connected to the noise sources [1]. Figure 2.7 shows a diagram of the phase domain equivalent of the self-oscillating core.

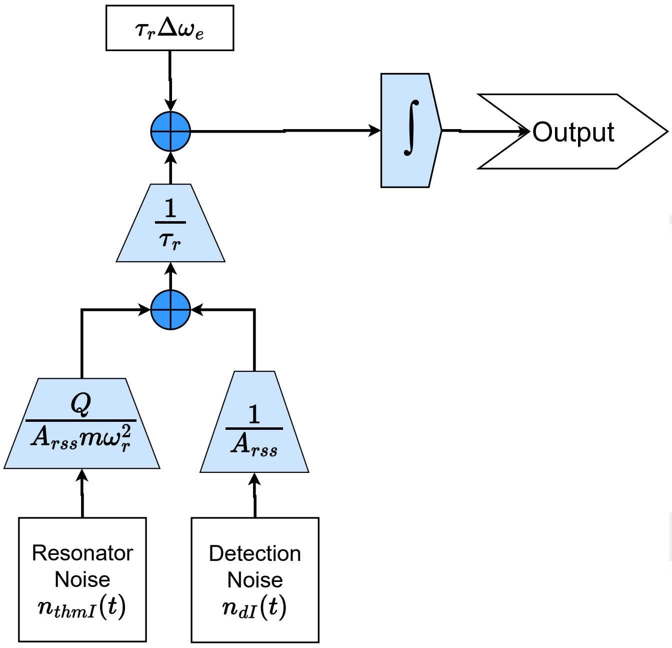

Figure 2.7: A diagram of the phase domain equivalent of the self-oscillating core.

2.4.3 Hybrid Core

The phase domain equivalent of the hybrid core is $H_H(s) = \frac{1}{\tau_h s+1}$ where $\tau_h = \frac{\tau_r}{R_{inj}} = \frac{2Q}{\omega_r R_i n j}$. The two noise sources are multiplied with some constants and added to the input and the output of the system. The detection noise is added to both at the input and the output of $H_H(s)$, due to the loop in the hybrid core.

2.4.4 Demodulator

The phase domain equivalent of the demodulator is the low-pass filter inside it ($H_L(s)$).

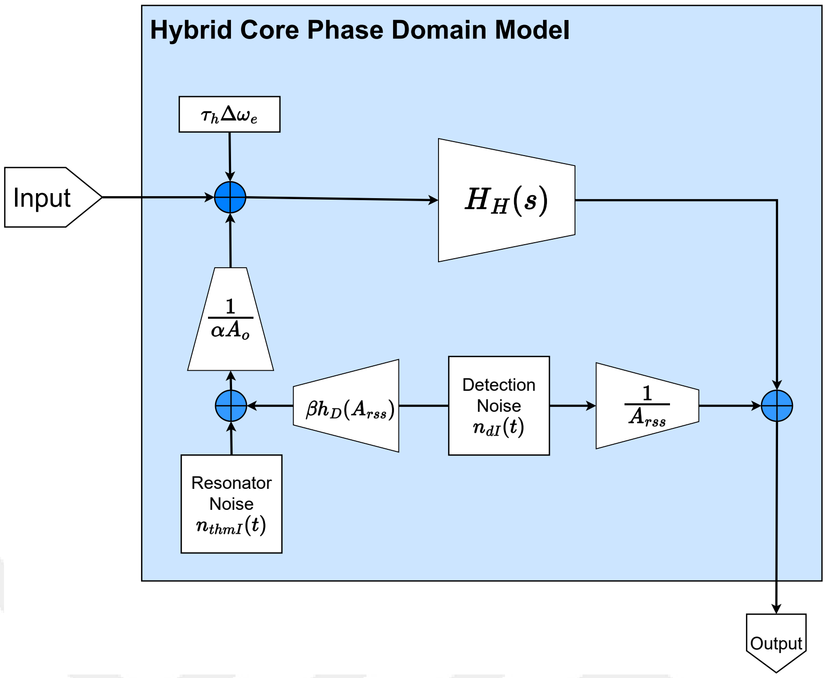

Figure 2.8: A diagram of the phase domain equivalent of the hybrid core.

2.4.5 PLL

The phase domain equivalent of the PLL can be obtained by replacing the demodulator with its phase domain equivalent ($H_L(s)$) and the controlled oscillator with an integrator.

Chapter 3

SCHEMES

Nanomechanical resonant sensing incorporates various schemes to employ a resonator as an event detector. To build a resonant sensor, a resonator together with a frequency detection module can be utilized in different configurations that are called the sensing schemes [1]. A resonance frequency based sensing scheme is basically composed of an oscillatory core containing a resonator and a frequency detection module. In the previous chapter we have seen three oscillatory cores and three frequency detection methods. Combining from these oscillatory cores and frequency detection methods, six different resonance frequency based sensing schemes can be formed.

The theoretical analysis presented in Chapter 2 has shown that the oscillatory cores can be simplified into the phase domain and still preserve the resonance frequency information, albeit approximately. Similar to the oscillatory cores, frequency detection modules can also be moved into the phase domain. Thus, each sensing scheme can be converted into its phase domain equivalent by replacing the oscillatory core and the frequency detection module with their phase domain counterparts. Phase domain models of the most common resonant sensing schemes have been studied recently in the literature [1].

Phase domain models offer a simplified quantification of the measurement process of the resonance frequency shifts caused by events of interest and affected by noise. A shift in the resonance frequency goes through the oscillatory core and the frequency detection module until appeared at the measurement. Likewise, the noise sources are shaped by the scheme before affecting the measurements. Linear(ized) phase domain models can be used to compute transfer functions from resonance frequency shift and noise sources to the output, allowing easy evaluation of the performance of a scheme in terms of speed and accuracy.

In the following sections we present the six resonance frequency based sensing schemes, which are the feedback-free (FF), frequency-locked loop (FLL), self-sustaining oscillator (SSO) with demodulator, SSO with phase-locked loop, hybrid open-loop (HOP) and hybrid closed-loop (HCL), together with their phase domain equivalents using diagrams. The first three of these schemes are analyzed in [1] and the last one is studied in [4] as "heterodyne phase-controlled oscillator", although without a detailed analysis.

3.1 Feedback-free Scheme (FF)

The scheme composed of a feedback-free resonator connected to a demodulator is called the feedback-free (FF) scheme [1], as there is no feedback loop in the oscillatory core or frequency detection module. In the FF scheme, the resonator is fed by a local oscillator oscillating at a fixed frequency close to the resonance frequency. The demodulator monitors the phase difference applied by the resonator, comparing the phase of the resonator's output and the local oscillator signal. Using this phase difference information, difference between the resonance frequency and the frequency of the oscillations is calculated using the phase response of the resonator.

In the phase domain equivalent of the FF scheme, the local oscillator is removed, the resonator and the demodulator is replaced by their phase domain equivalents obtained in Chapter 2. Figures 3.1 and 3.2 show diagrams of the FF scheme and the phase domain equivalent of it.

3.2 Frequency Locked Loop (FLL)

In the frequency locked loop, the resonator is fed by a controlled oscillator whose frequency is always maintained at the resonance frequency of the resonator [1], therefore, the resonator oscillates always at the resonance frequency. Shifts in the resonance frequency can be measured using the control signal produced by the controller.

The resonator and the demodulator are replaced by $H_R(s)$ and $H_L(s)$ respectively in the phase domain similar to the FF scheme, and the feedback loop converted to

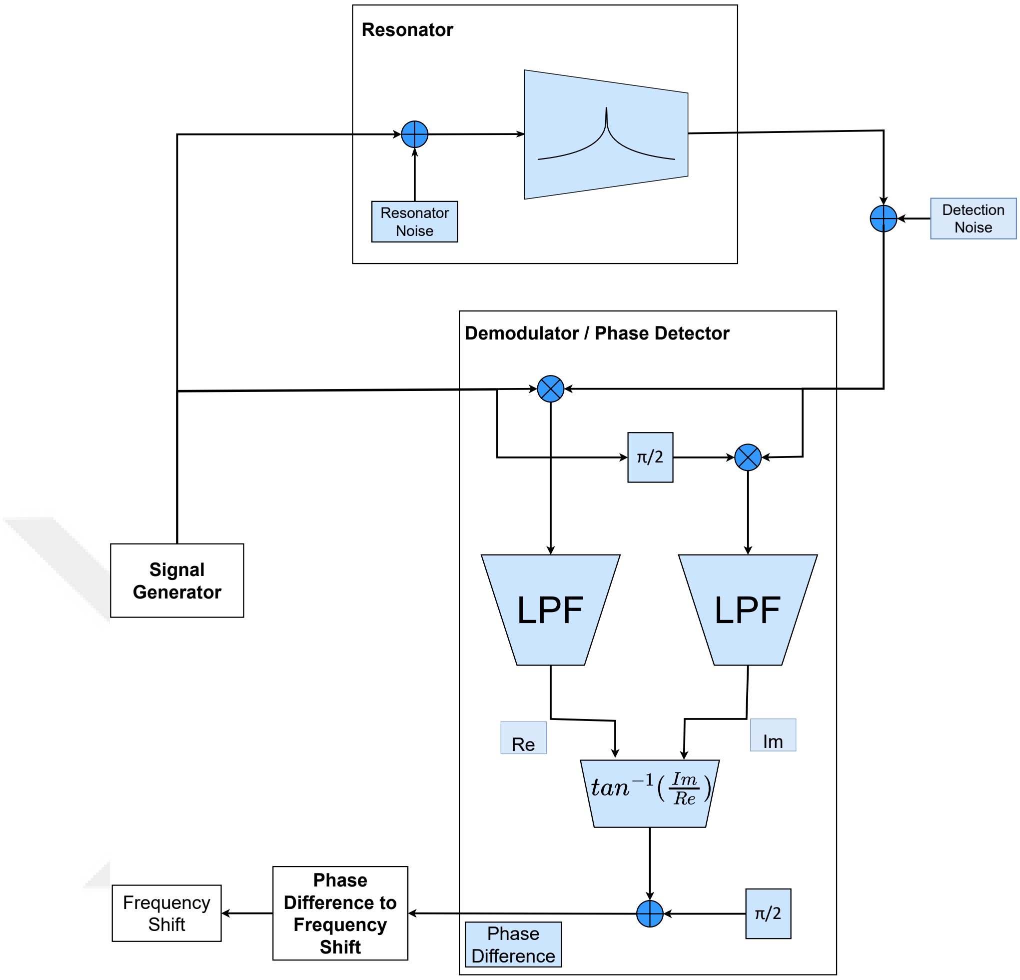

Figure 3.1: A diagram of the FF scheme (adapted from [1]).

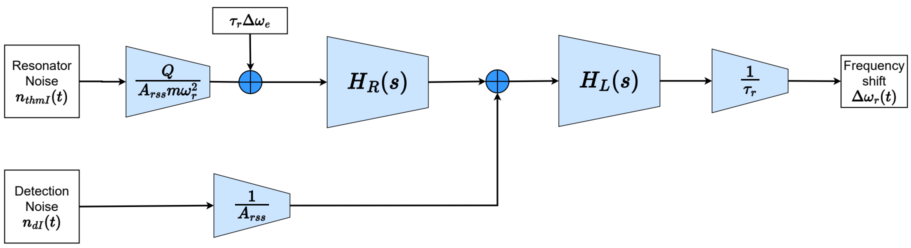

Figure 3.2: A diagram of the phase domain equivalent of the FF scheme (adapted from [1]).

a phase difference detector. The controlled oscillator is converted to a integrator while the controller is left untouched. Figures 3.3 and 3.4 show diagrams of the FLL scheme and the phase domain equivalent of it.

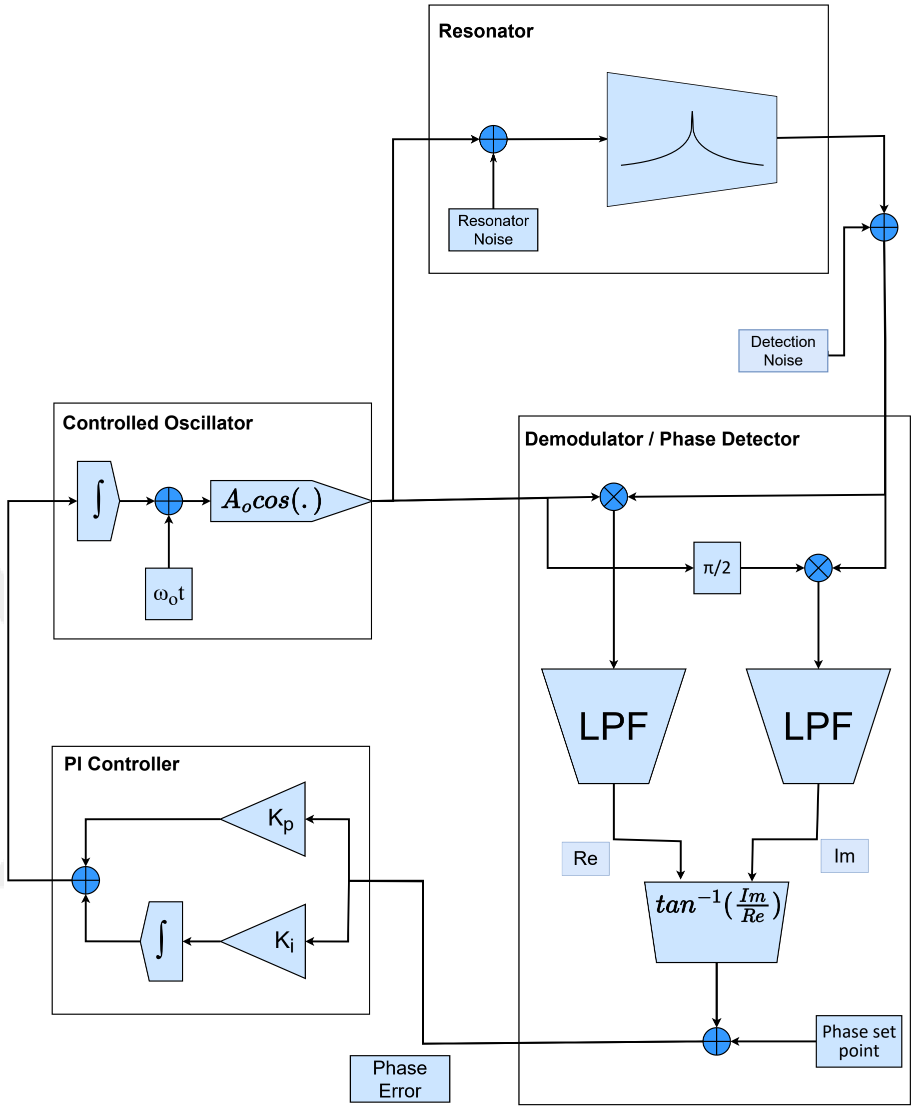

Figure 3.3: A diagram of the FLL scheme (adapted from [1]).

3.3 *Self-sustaining Oscillator (SSO) with Demodulator*

The SSO with demodulator scheme is composed of a self-oscillating core and a demodulator [1]. As we have mentioned in Chapter 2, in the self-oscillating core the resonator is fed by phase-shifted and amplified version of its output, and frequency of oscillations is always equal to the resonance frequency. Therefore, measuring the frequency of the oscillations is sufficient to track the resonance frequency of the resonator, unlike other schemes. For instance, in the FF scheme the resonator oscillates at the frequency of the local oscillator and resonance frequency can be

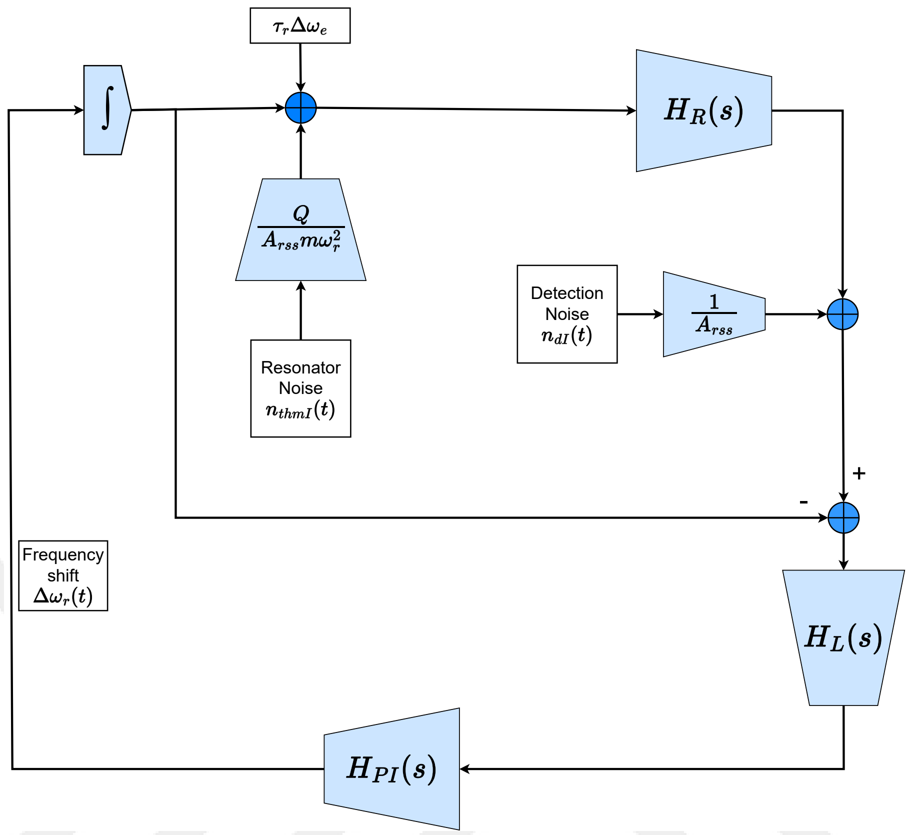

Figure 3.4: A diagram of the phase domain equivalent of the FLL scheme (adapted from [1]).

tracked by measuring the phase difference applied by the resonator and using the phase response of the resonator. Hence, the usage of the demodulator is different in the SSO scheme than the FF scheme, here the demodulator measures the phase difference between two signals that may have different frequencies. To obtain the frequency difference between the two signals the phase difference measured by the demodulator is differentiated. Figure 3.5 shows a diagram of the SSO scheme.

In the phase domain version of the SSO scheme, the oscillatory core is replaced by an integrator and the demodulator is replaced by $H_L(s)$ similar to the other schemes. Figure 3.6 shows a diagram of the phase domain version of the SSO scheme.

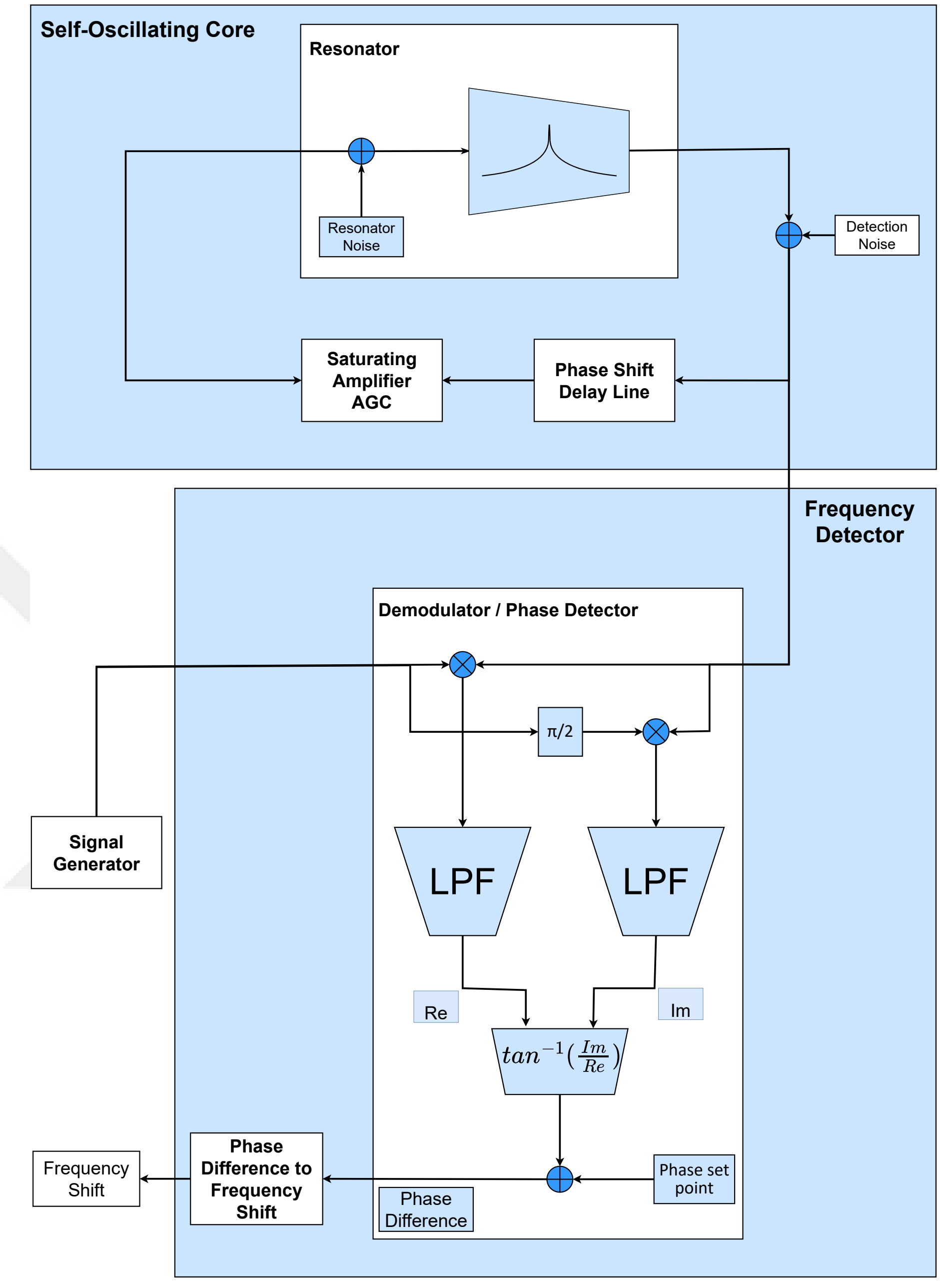

Figure 3.5: A diagram of the SSO with demodulator scheme (adapted from [1]).

3.4 *Self-sustaining Oscillator (SSO) with PLL*

The SSO with PLL scheme is composed of a self-oscillating core and a phase-locked loop as the frequency detector. As we have mentioned previously, the self-oscillating core oscillates at the resonance frequency of the resonator, therefore we need a frequency detection module to track the resonance frequency. A PLL contains a

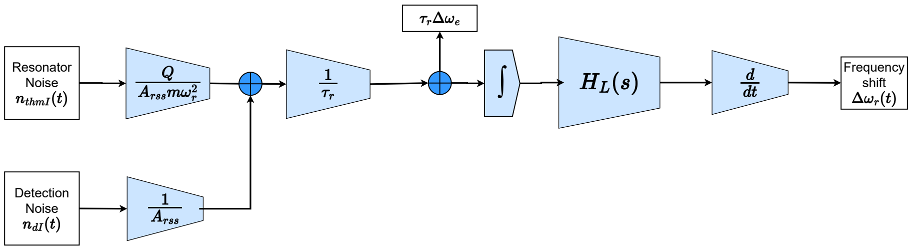

Figure 3.6: A diagram of the phase domain equivalent of the SSO with demodulator scheme (adapted from [1]).

controlled oscillator that oscillates at the same frequency as the input signal. Hence, in the SSO with PLL scheme the control signal feeding the controlled oscillator contains the resonance frequency information. Figure 3.7 shows a diagram of the SSO with PLL scheme.

In the phase domain equivalent of the SSO with PLL scheme, the self-oscillating core and the PLL are replaced with their phase domain counterparts. Phase domain counterpart of the self-oscillating core is an integrator as we have obtained in Chapter 2. To get the phase domain counterpart of the PLL, we replace the controlled oscillator with an integrator and the demodulator with $H_L(s)$. Figure 3.8 shows a diagram of the phase domain equivalent of the SSO with PLL scheme.

3.5 Hybrid Open-loop (HOP)

The hybrid open-loop scheme contains a hybrid core connected to a demodulator. As we have explained in Chapter 2, in the hybrid core the resonator is fed by a linear combination of the phase-shifted and amplified version of its output and an external signal. The HOP scheme can be obtained by replacing the resonator in the FF scheme with the hybrid core. The external input to the hybrid core is a pure sinusoid at a fixed frequency, in the HOP scheme, similar to the FF scheme. The demodulator measures the phase difference applied by the resonator, which is used to calculate the difference between the resonance frequency and the frequency of the oscillations according to the phase response of the resonator. Figure 3.9 shows a diagram of the HOP scheme.

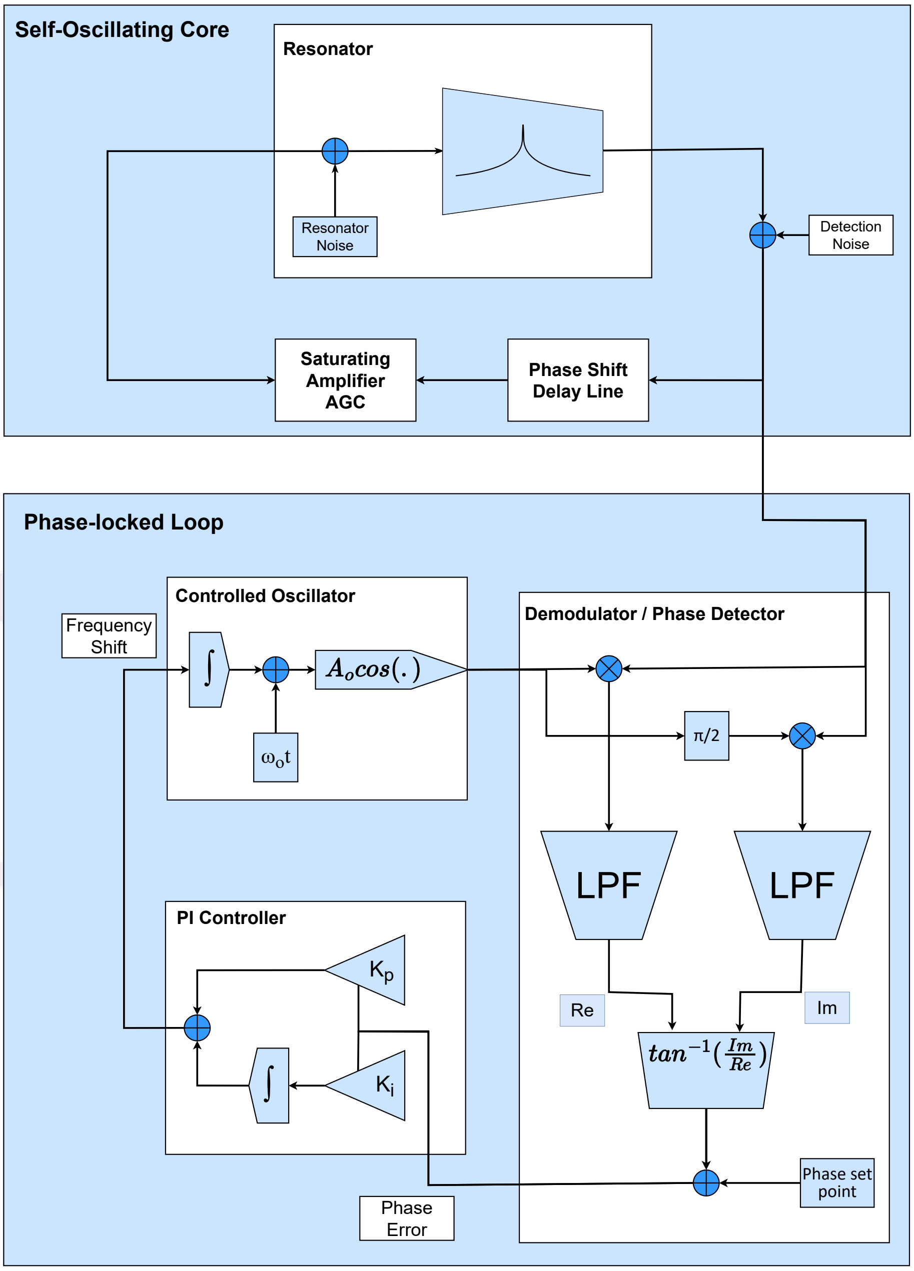

Figure 3.7: A diagram of the SSO with PLL scheme.

The phase domain equivalent of the HOP scheme is obtained by replacing the hybrid core and the demodulator with their phase domain counterparts, which are $H_H(s)$ and $H_L(s)$ respectively. The detection noise is connected to both before and after $H_R(s)$ due to the loop in the hybrid core. Figure 3.10 shows a diagram of the phase domain equivalent of the HOP scheme.

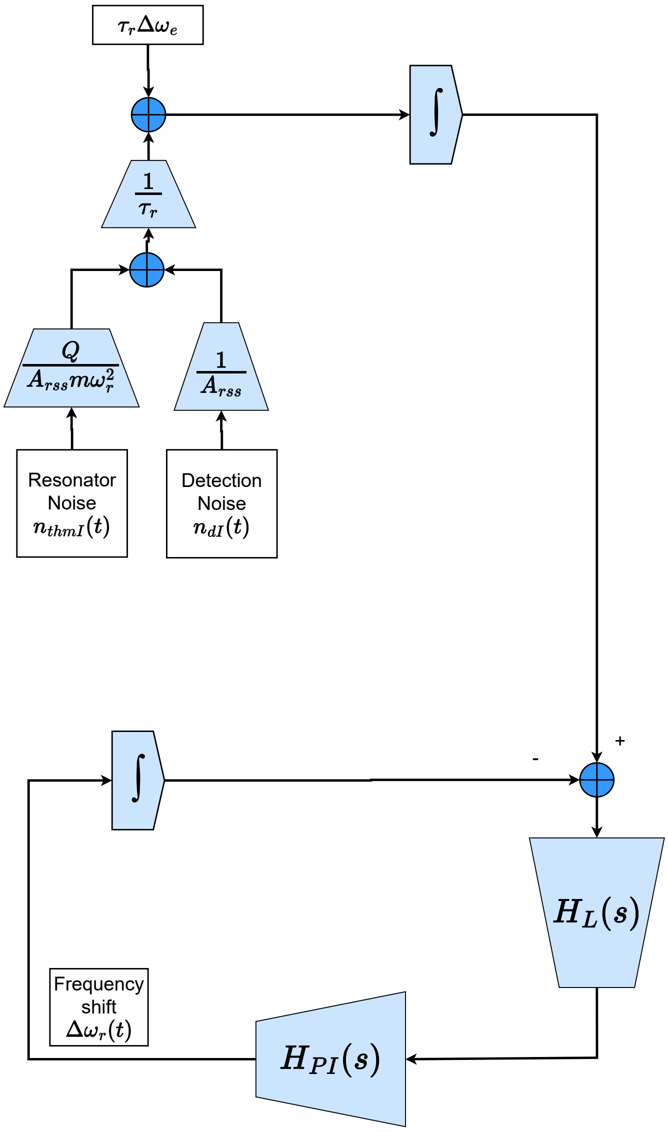

Figure 3.8: A diagram of the phase domain equivalent of the SSO with PLL scheme.

3.6 Hybrid Closed-loop (HCL)

The hybrid closed loop scheme consists of a hybrid core and a negative feedback loop [4]. The HCL scheme can be obtained by replacing the resonator in the FLL scheme with the hybrid core. The frequency of the external input of the hybrid core is controlled by the negative feedback loop, so that the resonator oscillates at the resonance frequency all time, similar to the FLL scheme. The control signal generated inside the negative feedback loop contains the resonance frequency information,

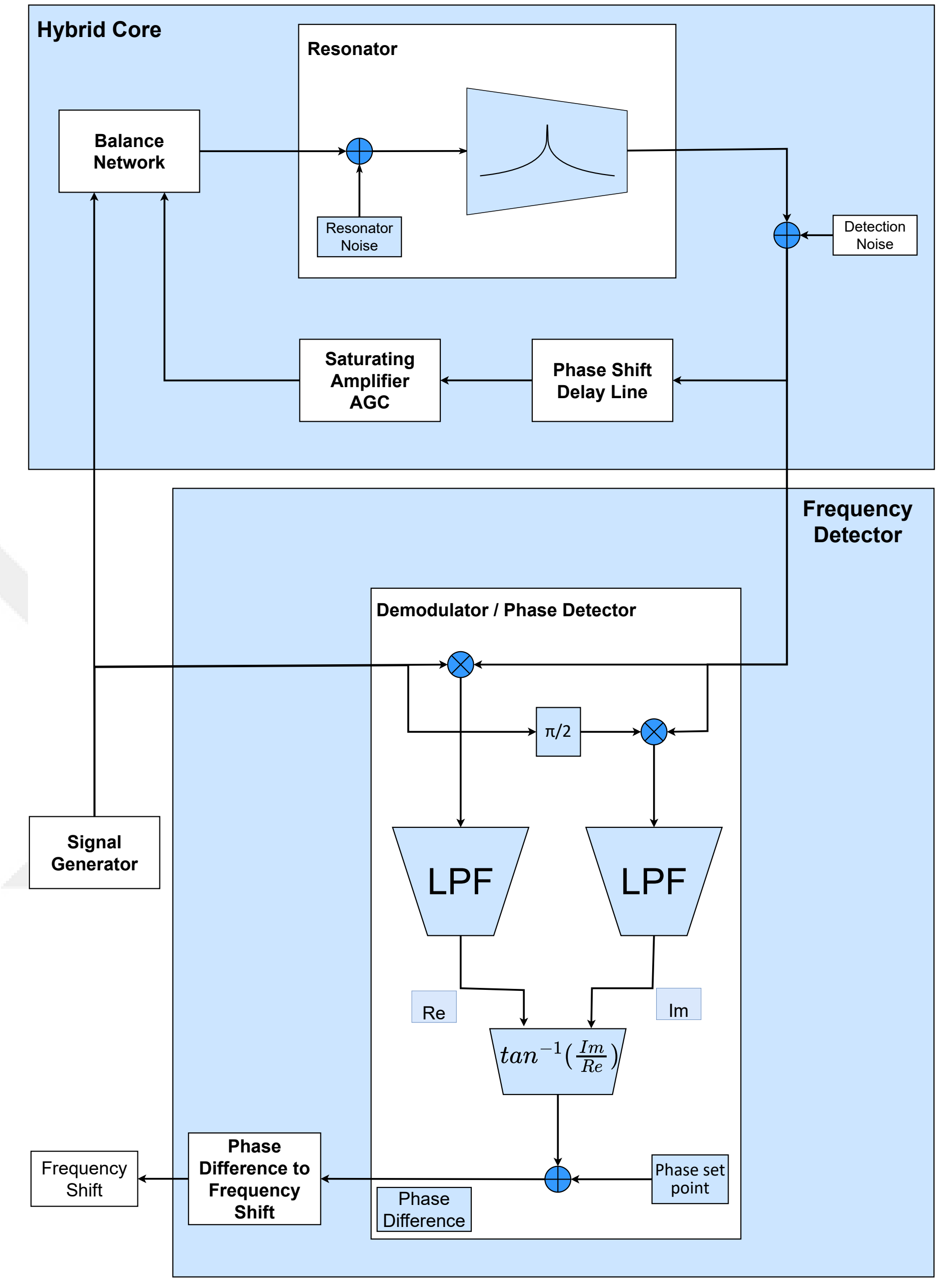

Figure 3.9: A diagram of the HOP scheme.

thereby enabling resonance frequency tracking. Figure 3.11 shows a diagram of the HCL scheme.

The phase domain equivalent of the HCL scheme can be obtained by replacing the hybrid core and the negative feedback loop with their phase domain counterparts. We have shown in Chapter 2 that the phase domain equivalent of the hybrid core is

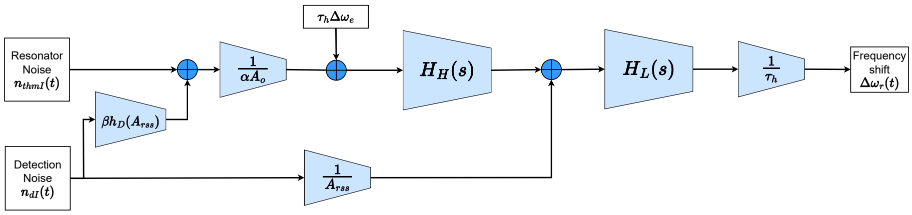

Figure 3.10: A diagram of the phase domain equivalent of the HOP scheme.

$H_H(s) = \frac{1}{\tau_h s+1}$ where $\tau_h = \frac{\tau_r}{R_i nj}$ and $R_i nj$ is the ratio of injection which is equal to the ratio of the amplitudes of the external input to the feedback input in the hybrid core. The detection noise contributes at both before and after $H_H(s)$ as we have mentioned in the previous section. The negative feedback loop can be converted into the phase domain by replacing the demodulator and the controlled oscillator with $H_L(s)$ and an integrator similar to the FLL scheme. Figure 3.12 shows a diagram of the phase domain equivalent of the HCL scheme.

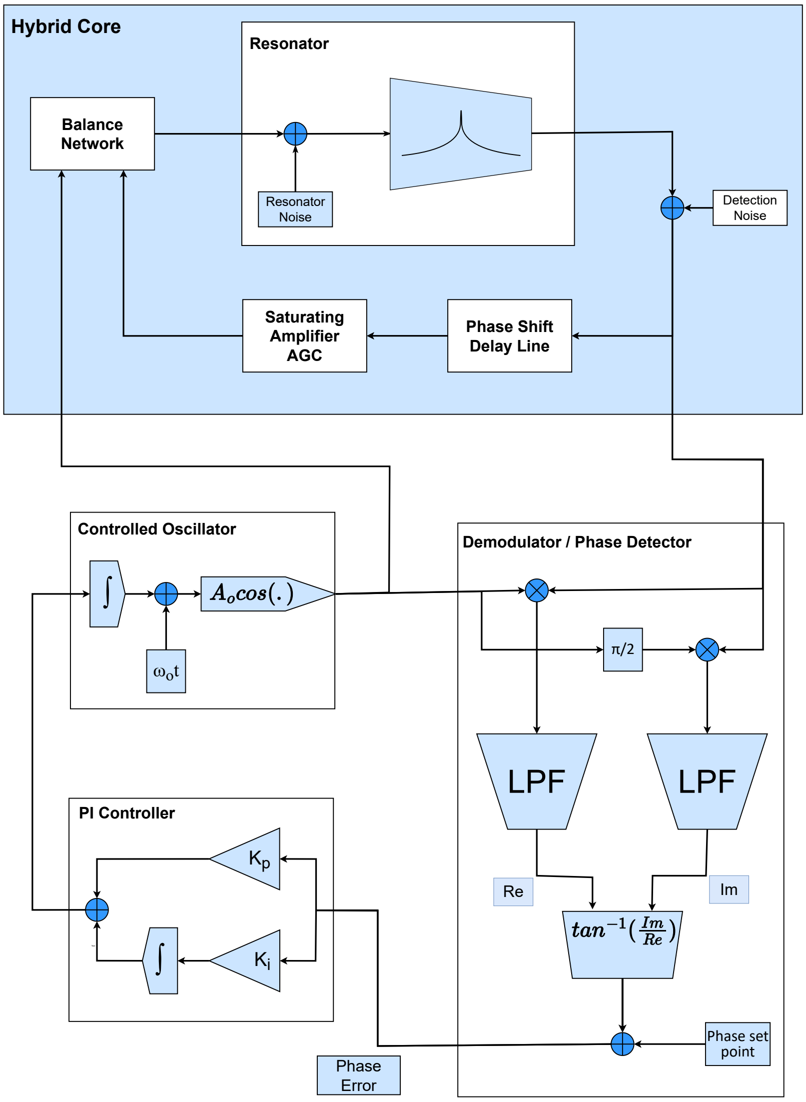

Figure 3.11: A diagram of the HCL scheme.

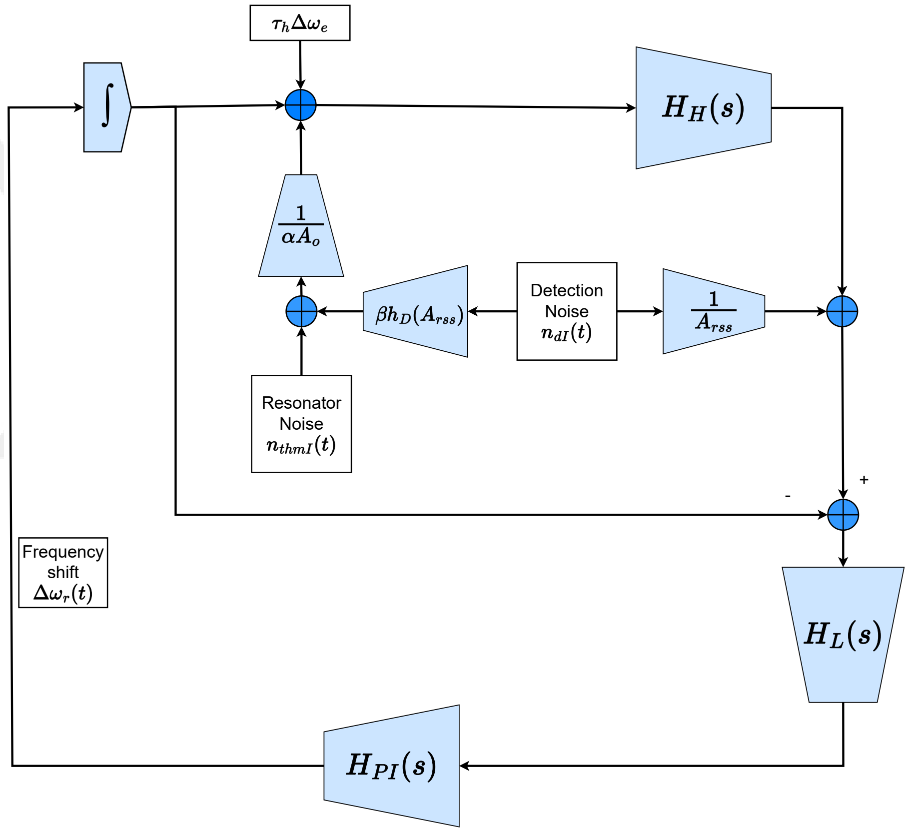

Figure 3.12: A diagram of the phase domain equivalent of the HCL scheme.

Chapter 4

THE TOOL: NEMSUITE

In this chapter, we focus on the simulation and analysis tool that we have developed. NEMSUITE is an end-user software tool designed to assist the nanomechanical resonant sensing studies by enabling experiments in a pure software environment.

NEMSUITE consists of a SIMULINK® custom block library, MATLAB® scripts and example models of nanomechanical resonant sensing setups. Users can build their models in SIMULINK® environment using the custom blocks as well as any other SIMULINK® block. The models can be run in either the full simulation or the fast analysis mode where the full simulation mode offers more accurate results with the expense of time and in the fast analysis mode on the other hand, users can obtain good approximation to the full simulation results in seconds rather than hours.

In the next section we go over the custom blocks of NEMSUITE, then in Section 4.2 we examine the full simulation and fast analysis modes.

4.1 *Custom* Simulink® *Blocks*

SIMULINK® offers the capability of creating custom blocks as well as a library containing them [15]. We use built-in SIMULINK® blocks to develop our block models then convert the models into subsystems. A subsystem in SIMULINK® is a visual abstraction that displays a group of interconnected blocks as a single block, hiding the internal details of the group [16]. Moreover a subsystem can be masked to have its own block parameters and custom interface to set and view the parameters [17]. In short, we have created our custom blocks in SIMULINK® by firstly building models, then converting the model into subsystems, and thirdly masking the subsystems. Masking phase includes the configuration of the parameter setting interface and parameter mapping between the interface and the internal model.

In NEMSUITE, there are two sets of custom blocks, namely the full simulation blocks and the fast analysis blocks. These two sets have one-to-one correspondence between them, in other words, there are two versions of each block; one for the full simulation and the other for the fast analysis. In the following subsections we go over each block in detail.

4.1.1 Resonator

We model the resonator with a damped harmonic oscillator formula as we have mentioned in chapter 2. The second order resonator model (2.1) is reproduced here for convenience

$$\frac{d^2}{dt^2}x(t) + \frac{\omega_r}{Q}\frac{d}{dt}x(t) + \omega_r^2 x(t) = \frac{F(t)}{m} \tag{4.1}$$

where $x(t)$ and $F(t)$ are the output and the input signals respectively.

We use a built-in state-space block for the resonator. The state-space model corresponding to (4.1) is

$$\begin{aligned} \frac{dX}{dt} &= AX + BU \\ Y &= CX + DU \end{aligned} \tag{4.2}$$

where $Y = x(t)$, $U = F(t)$, $X = \begin{bmatrix} x(t) \\ x'(t) \end{bmatrix}$, $A = \begin{bmatrix} 0 & 1 \\ -\omega_r^2 & \frac{-\omega_r}{Q} \end{bmatrix}$ $B = \begin{bmatrix} 0 \\ 1 \end{bmatrix}$ $C = \begin{bmatrix} 1 & 0 \end{bmatrix}$ $D = 0$. Figure 4.1 shows the block parameters interface of the resonator block.

The phase-domain model for the feedback-free resonator is

$$H_R(s) = \frac{1}{\tau_r s + 1}, \tag{4.3}$$

where $\tau_r = \frac{2Q}{\omega_r}$ is the intrinsic time-constant of the resonator. We use a built-in transfer fcn block for the fast analysis version of the resonator [18].

4.1.2 Self-Oscillating Core

We use the resonator block in self-oscillating core. There is no external input to the resonator except the two noise sources: thermomechanical and detection noise. The

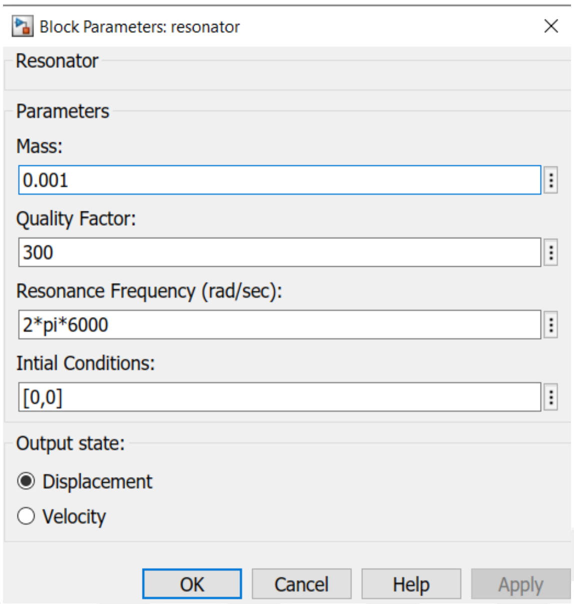

Figure 4.1: Block parameters interface of the resonator block.

resonator is fed by an nonlinearly amplified and phase-shifted version of its output and the oscillations in the self-oscillating core are created by this positive feedback loop. Figure 4.2 shows the internal structure of the self-oscillating core. The phase

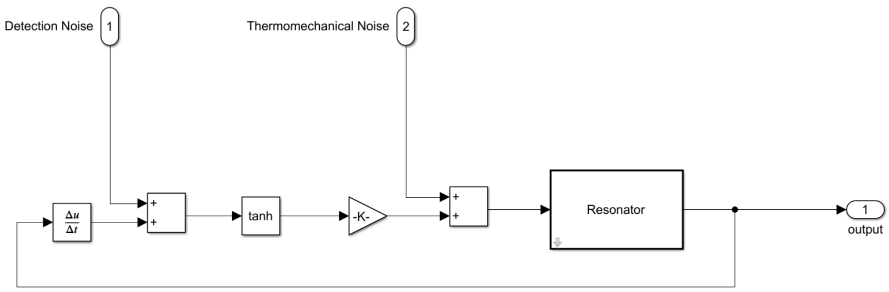

Figure 4.2: Internal structure of the self-oscillating core block.

domain equivalent of the self-oscillating core is basically an integrator, Figure 4.3 shows the internal structure of the phase domain equivalent of the self-oscillating core.

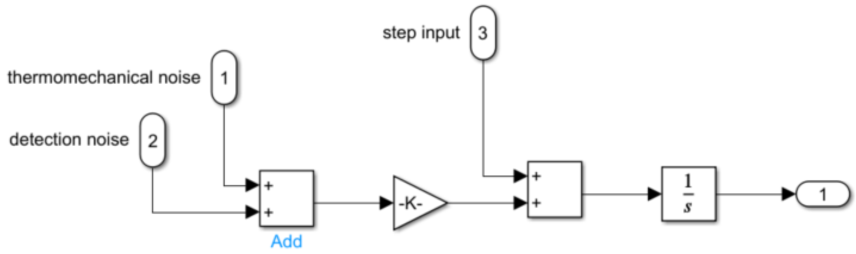

Figure 4.3: Internal structure of the phase domain version of the self-oscillating core block.

4.1.3 Hybrid Core

Similar to the self-oscillating core, the hybrid core is build upon the resonator model. Unlike the self-oscillating core, the resonator is fed by a linear combination of amplified and phase-shifted version of its output and an external signal. This linear combination is conducted by a custom balance network block. Figure 4.4 shows the internal structure of the hybrid core block.

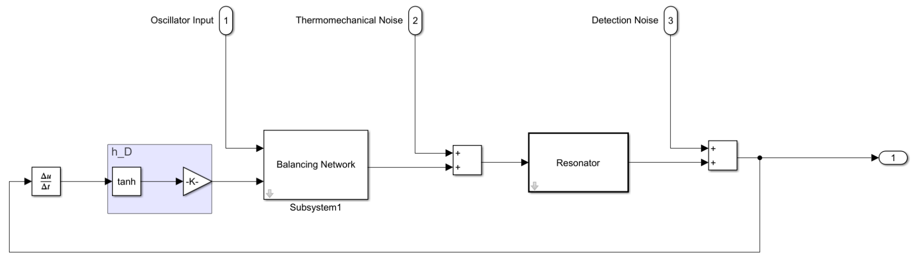

Figure 4.4: Internal structure of the hybrid core block.

At the center of the phase domain model of the hybrid core is the transfer function of the one pole filter that we have found in Chapter 2: $H_H(s) = \frac{1}{\tau_h s+1}$. Figure 4.5 shows the internal structure of the phase domain version of the hybrid core.

4.1.4 Demodulator

The demodulator block contains multipliers, low-pass filters, atan nonlinearity and other basic built-in SIMULINK® blocks. Figure 4.6 shows the internal structure

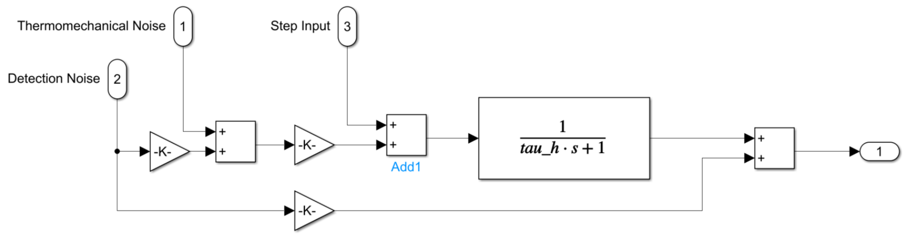

Figure 4.5: Internal structure of the phase domain version of the hybrid core.

of the demodulator. The three inputs are the resonator signal, the in-phase and the quadrature versions of the reference signal. The output is the phase difference between the resonator and the reference signals.

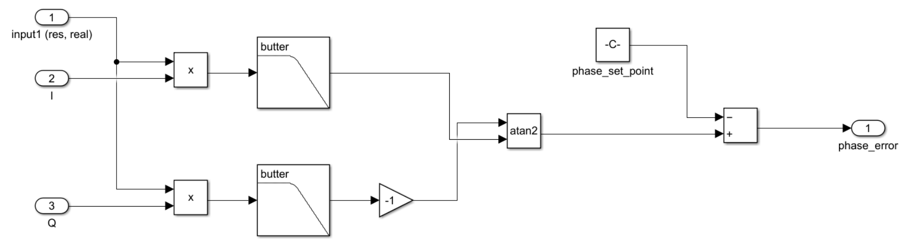

Figure 4.6: Internal structure of the demodulator block.

The phase-domain equivalent of the demodulator is the low-pass filter inside it.

4.1.5 Controller

We prefer using PI controller in NEMSUITE, other types of controllers can be added easily. For the PI controller we use built-in transfer fcn block [18]. The phase domain model of the controller is itself.

4.1.6 Controlled Oscillator

We have a numerically-controlled oscillator among NEMSUITE blocks. The internal structure of the block is shown in Figure 4.7. The phase-domain version of the numerically controlled oscillator is just an integrator.

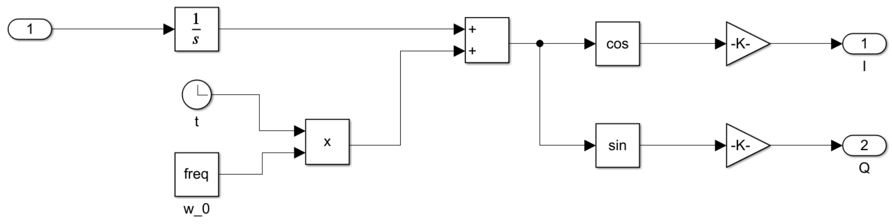

Figure 4.7: Internal structure of the controlled oscillator block.

4.1.7 Local Oscillator

Local Oscillator is a NEMSUITE custom block that frequency, amplitude and phase information can be set through its block parameters interface and generates a pure sinusoidal signal accordingly at its output. Figures 4.8 and 4.9 show the block parameters interface and the internal structure of the Local Oscillator block.

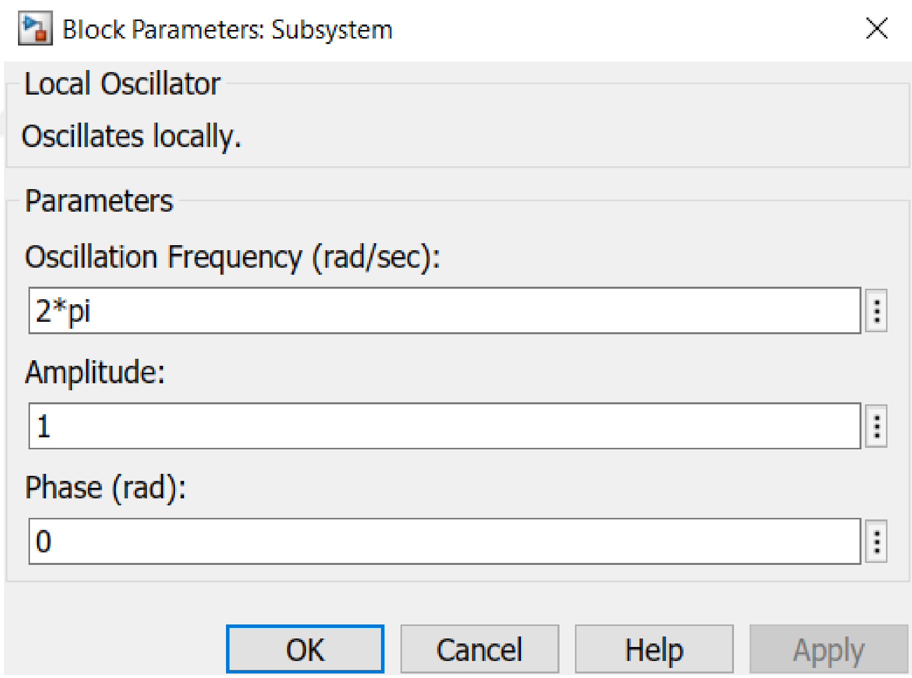

Figure 4.8: Block parameters interface of the local oscillator block.

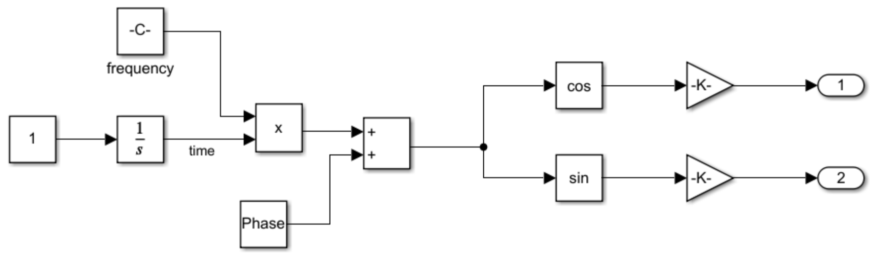

Figure 4.9: Internal structure of the local oscillator block.

4.1.8 White Noise

White Noise block is a wrapper around SIMULINK®'s built-in Band-limited White Noise block [19], offering a more convenient parameters interface. This wrapping is required for NEMSUITE scripts to detect the noise sources in a model. Figure 4.10 shows the parameters interface of the white noise block.

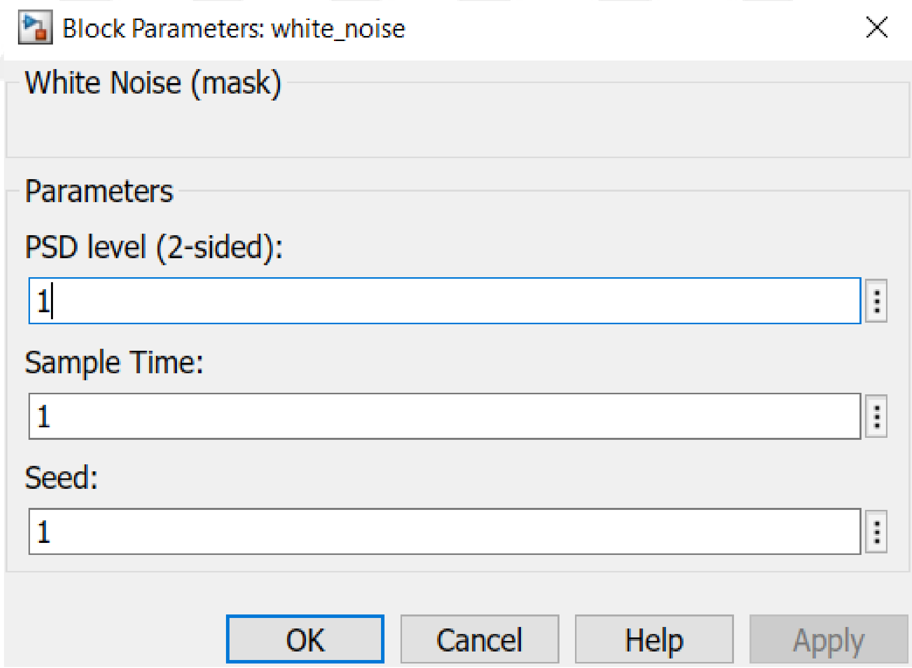

Figure 4.10: Block parameters interface of the white noise block.

4.1.9 Maestro

Maestro is a simulation control block that enables users to select the simulation type as full simulation or fast analysis and start/stop the simulation. Figure 4.11 shows the interface of the Maestro block.

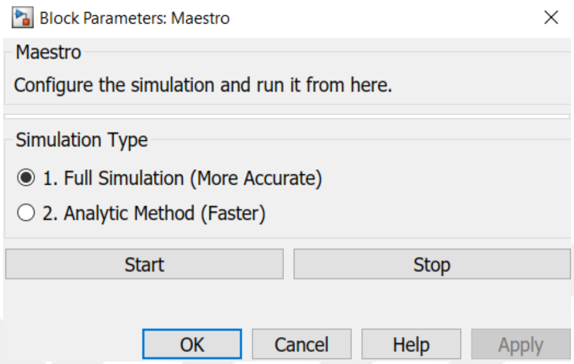

Figure 4.11: Interface of the maestro block.

4.2 Simulation Types

There are two simulation types available in NEMSUITE, to run the models and get results in various forms. The full simulation mode offers more accurate results with the expense of time, whereas the fast analysis mode promises much shorter runtime with little loss of accuracy. In this section we go through these simulation types.

4.2.1 Full Simulation

Models built by using NEMSUITE blocks can be run in MATLAB®/SIMULINK® environment similar to any other SIMULINK® model. NEMSUITE full simulations give more accurate results but take more time compared to NEMSUITE fast analysis; to get accurate results in an acceptable amount of time however, simulation settings must be configured correctly. We briefly mention a couple of these settings and some points to consider here.

A computer has only a finite number of bits to represent a number. Therefore, not all -infinitely many- numbers can be represented in the memory. This means that, the numbers that a computer is able to represent constitutes a finite set A, which is called machine numbers [20]. Numbers that are not present in A must be rounded to one of the elements of A, so that, a number $x \notin A$ that should have theoretically shown up in the calculations, is represented with a machine number $x' \in A$. A computer, most of the time, picks x' so that the difference $|x - x'|$ is minimized. Nevertheless, there is a deviation in the calculated result from the theoretical one, which is called round-off error [20].

Discretization error is another type of error that is caused by the finite processing power of a computer, which necessitates finite number of steps in a numerical computation [21]. During simulation, calculations occur at discrete time-points, producing only an approximation to the theoretical result. Discretization error can be reduced by decreasing the intervals between the time-points also known as time-step size.

Decreasing the time-step size can reduce the discretization error, however at the same time it increases the amount of calculations and consequently the accumulated round-off error. As the time-step size decreases, round-off error becomes more apparent and at some point it starts to dominate over the discretization error [21].

One way of minimizing the round-off error is working with machine numbers as much as possible. In a resonance frequency based sensing simulation, time is the most important quantity to know precisely throughout the simulation, since phase and frequencies are calculated based on the simulation time. To have a precise time information at each step of a simulation, time steps must be machine numbers so that after each step time quantity ends up being a machine number.

In simulations, time steps are adjusted according to the highest frequency. Time steps must be smaller than the smallest period of change, to make sure that the simulation embodies all of the dynamism the model have. For instance, if the fastest dynamic of a model has a frequency f Hz, time step size can be picked to be $\frac{1}{10f}$ seconds to have ten points in each period. However, setting the time step size like this has little chance to result in time steps and consequently time points belonging to the set of machine numbers.

NEMSUITE offers optional frequency normalization to avoid rounded time points problem. When the frequency normalization is activated, the resonance frequency is reduced so that the required time step size becomes an integer power of two, thereby having all time points throughout the simulation in the set of machine numbers. In some cases the smallest available power of two may turn out to be zero, depending on the frequencies and number of time steps in one period.

SIMULINK® offers two types of solvers: variable-step solvers and fixed-step solvers [22]. Variable-step solvers adjust the time-step size in a range during the simulation according to the requirements imposed by model dynamics as well as error limits, while fixed-step solvers work with a predetermined time step size disregarding any requirement or any error limit. Variable-step solvers are generally considered to be superior to fixed-step solvers in terms of accuracy and time-efficiency, however setting time-steps for frequency normalization is possible with fixed-step solvers. Therefore, picking a variable-step solver may increase the error by inhibiting frequency normalization and causing round-off errors in timing. There are reasons for selecting both types of solvers. In our experiments we generally get better results with fixed-step solvers.

4.2.2 Fast Analysis

Fast analysis is one of the simulation types available in NEMSUITE, besides the full simulation. In fact, fast analysis is a frequency domain evaluation of the linear(ized) phase domain equivalent of a model, rather than a time-stepping based numerical simulation. Through fast analysis, users can get good approximation to the results obtained with the full simulation in considerably shorter time.

NEMSUITE enables users to run their resonance frequency based sensing models in fast analysis mode through the interface of the maestro block as seen in Figure 4.11. The fast analysis procedure are controlled by the scripts running in the background, and starts with saving a copy of the model with a different name. All of the operations are conducted on the copied file to protect the original file. As the first step of operations, all blocks in the model are iterated through and replaced with their phase domain counterparts if available. As we have mentioned in the previous section, all NEMSUITE blocks have their phase domain counterparts. By replacing the components with their phase domain counterparts, we get a linear(ized) phase domain equivalent of a model. These linear(ized) equivalents are then used for characterizations of the model against noise effects and small resonance frequency shifts. As we have mentioned in Section 2.4, this analysis technique is called small signal analysis in electrical engineering.

Next, transfer functions from noise sources and step input to phase error output are obtained using MATLAB®'s `linmod` routine [23]. `linmod` is capable of linearizing nonlinear SIMULINK® models before generating the state-space representations of them, however in NEMSUITE, `linmod` is utilized only to get the state-space representations of SIMULINK® models that are already linearized when converted into the phase domain by the scripts contained in NEMSUITE. Using the state-space representations generated by `linmod`, several transfer functions are obtained. For the case where there are thermomechanical and detection noise sources in the model, three transfer functions are sufficient for speed and accuracy characterizations. The transfer functions multiplying the noise sources are used to calculate Allan deviation, while the transfer function for the step input is used to calculate the step response of the model. Hence all of the results that can be obtained with full simulation can be obtained through fast analysis too. As we discuss in the next chapter, in our experiments the results from the full simulation and the fast analysis are overlapped to a large extent.

Chapter 5

RESULTS

In this chapter we present the results obtained with NEMSUITE. Although NEMSUITE is developed to evaluate models in terms of speed and accuracy, comparing different schemes are out of the scope of this thesis. However, we present some results supporting the theory obtained with NEMSUITE. Also, we try to demonstrate that results obtained from full simulation and fast analysis match well.

In the following section we validate the fast analysis method with Allan Deviation and frequency step response plots. The fast analysis method has been demonstrated to match with physical experiments [10], and our future plans include verification of NEMSUITE with physical experiments. Then in the next section we exhibit some results obtained with the Fast Analysis method supporting the theory.

5.1 Validation of the Fast Analysis

The theoretical foundations that the fast analysis is based on are derived rigorously in [1] and in Chapter 2 of this thesis. Although the mathematical flow is precise and provable, it contains some approximations thereby needing validation. The theory of the fast analysis method has been validated against precise computer simulations [9, 1] as well as physical experiments [10]. In our work we validate the fast analysis method by comparing its results on the sensing schemes with the full simulation results in terms of accuracy and speed.

In the following subsections we first investigate the accuracy and then the speed. We see a good correspondence between the results obtained with the two methods in both characteristics, validating the fast analysis method.

5.1.1 Accuracy

As mentioned earlier, we use Allan Deviation to characterize the schemes in terms of accuracy. Allan Deviation [24] is a standard measure of frequency stability that is widely used in resonator characterizations [1]. Allan Deviation is the square-root of Allan variance which is computed with the following equation

$$\sigma_y^2(\tau) = \frac{4}{\pi\tau^2}\int_{-\infty}^{+\infty}\frac{[sin(\frac{\omega\tau}{2})]^4}{\omega^2}\mathcal{S}_{y_r}(\omega)d\omega, \tag{5.1}$$

where $\mathcal{S}_{y_r}$ is the spectrum of fractional frequency fluctuations calculated using the transfer functions from noise sources to the frequency deviation output, in fast analysis. In full simulation on the other hand, we use an overlapped variable τ estimator for Allan Deviation [25].

In dozens of experiments that we have conducted, we see a very good match in the Allan Deviation plots generated with the full simulation and the fast analysis. In Figure 5.1 we exhibit a number of these plots all obtained using NEMSUITE.

5.1.2 Speed

To characterize the schemes in terms of speed we regard settling times in the responses to jumps in the resonance frequency. In phase domain, the schemes are converted into multiple transfer functions one of which is from the frequency step input to the output. In the fast analysis mode NEMSUITE uses this transfer function of a scheme to generate a frequency step response providing the speed characterization.

Figure 5.2 shows a couple of frequency step responses each corresponding to a particular setup in which the resonance frequency increases with a fractional frequency of $\frac{1}{2\pi 3000}$ at $t = 0$. We observe a high degree of correspondence in the step response plots generated via the full simulation and the fast analysis.

5.2 Results Supporting the Theory Obtained with NEMSUITE

NEMSUITE is developed to aid the nanomechanical resonant sensing studies. The full simulation mode enables users to conduct experiments in a fully software environment, while the fast analysis mode offers very fast evaluation of models. In this

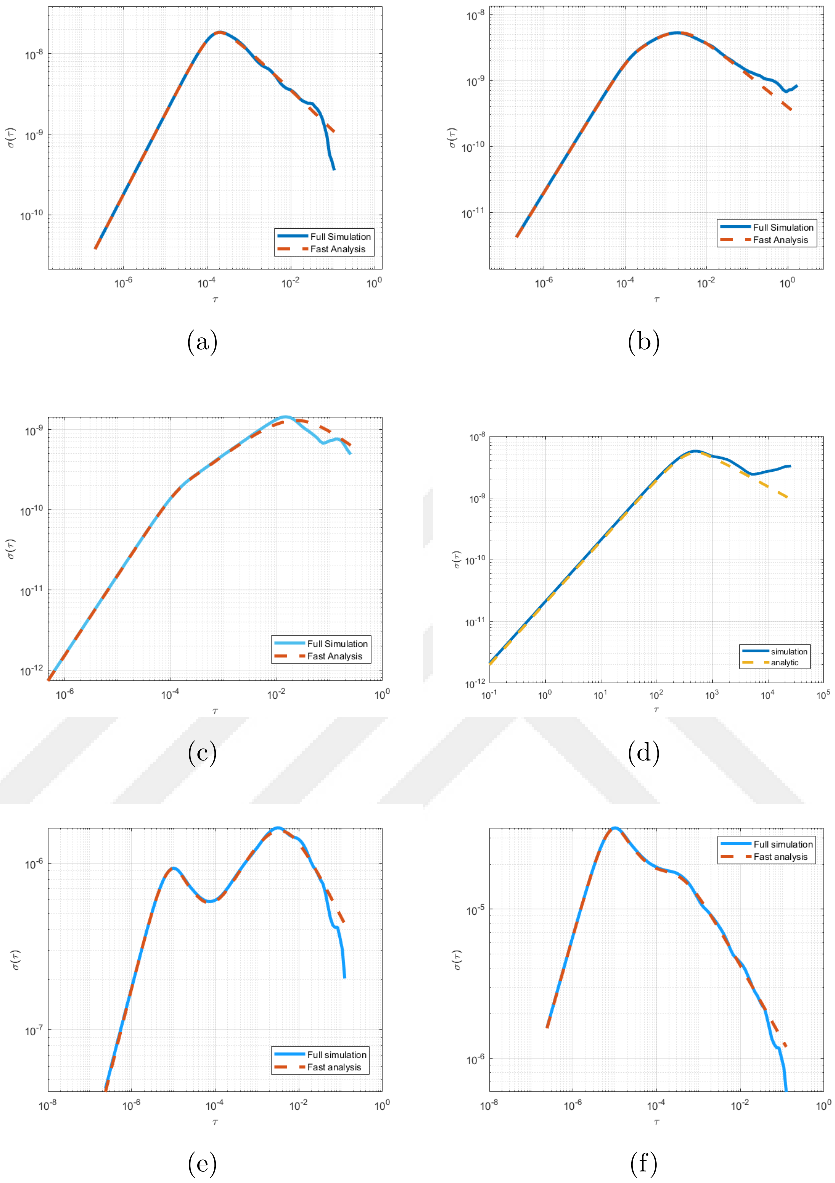

Figure 5.1: Sample Allan Deviation plots showing the correspondence between the full simulation and the fast analysis results. All plots are generated from sensing schemes built with NEMSUITE and contain both thermomechanical and detection noise sources. (a) FF scheme Q=50, (b) FF scheme Q=1000, (c) FLL scheme, (d) SSO with demodulator, (e) HOP scheme $\alpha = 0.1, \beta = 0.9$, (f) HOP scheme $\alpha = 0.5, \beta = 0.5$.

section we present some sample results obtained with the fast analysis demonstrating high potential of NEMSUITE.

Figure 5.3 shows the Allan Deviations calculated from the output of a setup of

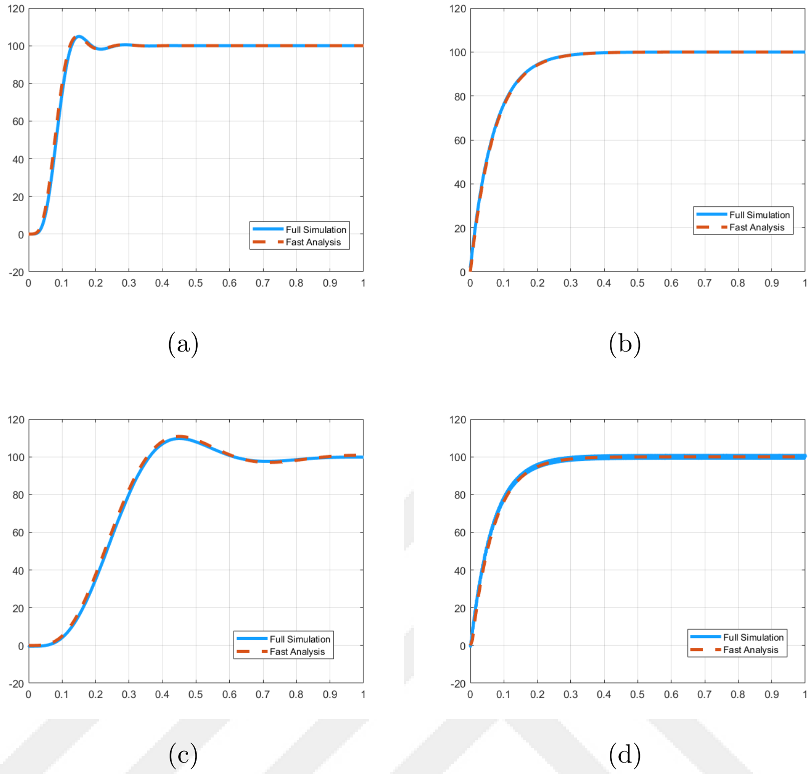

Figure 5.2: Sample frequency step response plots. (a) FF scheme, (b) FLL scheme, (c) SSO with demodulator, (d) HOP scheme

FF scheme with thermomechanical and detection noise sources. As Q increases, the noise level reduces as the PSD level of the thermomechanical noise is inversely proportional to the quality factor of the resonator. Another observation is that as Q increases two peaks move apart and become more apparent. These two peaks are due to the low-pass filtering imposed by the resonator and the demodulator, and as Q increases the time-constant of the resonator increases while the filtering effect of the demodulator stays constant.

HOP is a novel scheme that has not been studied in the literature. While the comparisons between the schemes are excluded from the scope this thesis, we present some results on HOP scheme characterizing it in terms of accuracy while demonstrating the potential of NEMSUITE.

Figure 5.4 shows Allan Deviation plots calculated from the output of HOP scheme. A number of α and β values are used, as mentioned earlier, α and β are

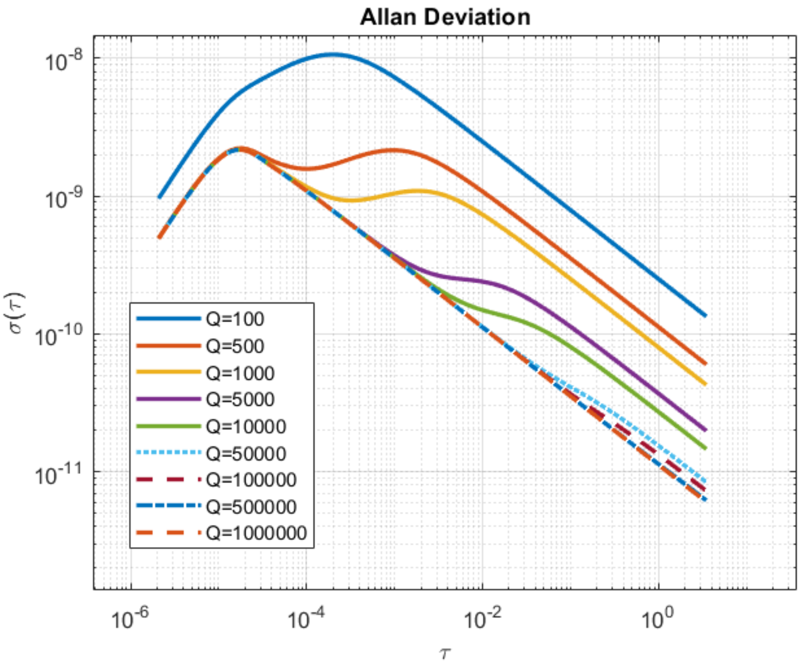

Figure 5.3: Allan Deviations at the output of an FF setup, as Q varies.

the coefficients in the hybrid core multiplying the external signal and the feedback signal respectively. In our experiment that results in Figure 5.4, we vary the α and β parameters in such a way that the amplitude of the input signal to the resonator is constant. This is necessary to get a correct interpretation from the results, as an input to the resonator with varying amplitude would lead to varying signal-to-noise ratio even if the noise level stays the same.

We observe from Figure 5.4 that, as α increases HOP scheme exhibits better noise performance. The best performance among the plots in Figure 5.4 is the case when $\alpha = 1$ and $\beta = 0$ in which HOP scheme becomes equivalent to FF scheme as $\beta = 0$ blocks any feedback input to the resonator. Therefore, as a result deduced only from Figure 5.4, we can say that FF scheme has better accuracy performance than HOP scheme. Nevertheless, more experiments should be conducted to fully understand the HOP scheme. NEMSUITE is highly convenient tool for such experiments, due to its features such as short runtimes and lack of any hardware necessity. After building the model, a figure similar to Figure 5.4 would take a couple of minutes to generate in the Fast Analysis mode.

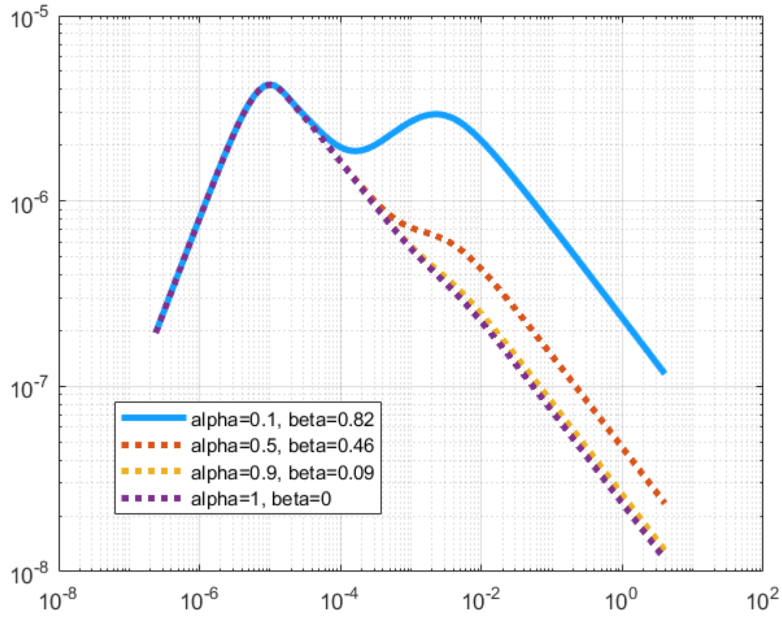

Figure 5.4: Allan Deviations at the output of an HOP setup, as α and β vary.

Chapter 6

CONCLUSION

Computers have been providing important aid to science and engineering for many decades, however this has been possible only with the presence of appropriate software programs running on the computers. NEMSUITE is such a software tool that has been developed to accelerate the research and development activities in the area of nanomechanical resonant sensing. This thesis can be considered as a design and usage manual of NEMSUITE.

We have presented the theoretical background of NEMSUITE in Chapter 2, the theory is developed upon a recent work [1]. In our treatment we have reached to results confirmed with established knowledge in the field of electrical engineering and nanomechanical resonant sensing.

In Chapter 3, we have introduced a number of resonance frequency based sensing schemes, most of which have appeared in the literature and have been used widely in the field. We have described the schemes and have presented block diagram models of the physical setups. These models are included in NEMSUITE as examples.

In Chapter 4, we have gone through NEMSUITE in detail. NEMSUITE is a MATLAB® SIMULINK® toolbox containing a custom block library and relevant MATLAB® scripts. NEMSUITE enables users to build their own nanomechanical resonant sensing models in SIMULINK® environment and offers two simulation modes for characterization of the models in terms of speed and accuracy. Most accurate results can be obtained with the full simulation mode but it may take hours to run. The fast analysis mode offers very good approximations to the full simulation results in much shorter runtimes.

We have presented some results in Chapter 5, in the form of Allan Deviation and frequency step response plots. We have observed a very good correspondence between the full simulation and the fast analysis results, validating the fast analysis method. Moreover, the results demonstrate the potential of NEMSUITE in the most common usage scenarios such as sweeping a parameter in the model.

We anticipate widespread adoption of simulation and analysis tools in nanomechanical resonant sensing studies in the near future. Our hope is that, with its great potentials, NEMSUITE will become a pioneering project in that direction.

BIBLIOGRAPHY

[1] A. Demir, "Understanding fundamental trade-offs in nanomechanical resonant sensors," *Journal of Applied Physics*, vol. 129, no. 4, p. 044503, 2021.

[2] M. S. Hanay, S. Kelber, A. Naik, D. Chi, S. Hentz, E. Bullard, E. Colinet, L. Duraffourg, and M. Roukes, "Single-protein nanomechanical mass spectrometry in real time," *Nature nanotechnology*, vol. 7, no. 9, pp. 602–608, 2012.

[3] J. Chaste, A. Eichler, J. Moser, G. Ceballos, R. Rurali, and A. Bachtold, "A nanomechanical mass sensor with yoctogram resolution," *Nature nanotechnology*, vol. 7, no. 5, pp. 301–304, 2012.

[4] U. Dürig, H. Steinauer, and N. Blanc, "Dynamic force microscopy by means of the phase-controlled oscillator method," *Journal of applied physics*, vol. 82, no. 8, pp. 3641–3651, 1997.

[5] J. R. Vig and Y. Kim, "Noise in microelectromechanical system resonators," *IEEE transactions on ultrasonics, ferroelectrics, and frequency control*, vol. 46, no. 6, pp. 1558–1565, 1999.

[6] A. N. Cleland and M. L. Roukes, "Noise processes in nanomechanical resonators," *Journal of applied physics*, vol. 92, no. 5, pp. 2758–2769, 2002.

[7] M. Sansa, E. Sage, E. C. Bullard, M. Gély, T. Alava, E. Colinet, A. K. Naik, L. G. Villanueva, L. Duraffourg, M. L. Roukes, *et al.*, "Frequency fluctuations in silicon nanoresonators," *Nature nanotechnology*, vol. 11, no. 6, pp. 552–558, 2016.

[8] K. L. Ekinci, Y. T. Yang, and M. L. Roukes, "Ultimate limits to inertial mass

sensing based upon nanoelectromechanical systems," *Journal of applied physics*, vol. 95, no. 5, pp. 2682–2689, 2004.

[9] A. Demir and M. S. Hanay, "Fundamental sensitivity limitations of nanomechanical resonant sensors due to thermomechanical noise," *IEEE Sensors Journal*, vol. 20, no. 4, pp. 1947–1961, 2019.

[10] P. Sadeghi, A. Demir, L. G. Villanueva, H. Kähler, and S. Schmid, "Frequency fluctuations in nanomechanical silicon nitride string resonators," *Physical Review B*, vol. 102, no. 21, p. 214106, 2020.

[11] R. Adler, "A study of locking phenomena in oscillators," *Proceedings of the IRE*, vol. 34, no. 6, pp. 351–357, 1946.

[12] B. Razavi, "A study of injection locking and pulling in oscillators," *IEEE journal of solid-state circuits*, vol. 39, no. 9, pp. 1415–1424, 2004.

[13] P. Bhushan, "A rigorous graphical technique for predicting sub-harmonic injection locking in lc oscillators," in *Proceedings of the 51st Annual Design Automation Conference*, pp. 1–8, 2014.

[14] A. Sedra and K. Smith, *Microelectronics Circuits.* Oxford University Press, 2015.

[15] MathWorks®, "Custom libraries." `mathworks.com/help/simulink/libraries.html`, 2022 (accessed August 1, 2022).

[16] MathWorks®, "Subsystems." `mathworks.com/help/simulink/subsystems.html`, 2022 (accessed August 1, 2022).

[17] MathWorks®, "Create block masks." `mathworks.com/help/simulink/block-masks.html`, 2022 (accessed August 1, 2022).

[18] MathWorks®, "Transfer fcn." `mathworks.com/help/simulink/slref/transferfcn.html`, 2022 (accessed August 1, 2022).

[19] MathWorks®, "Band-limited white noise." mathworks.com/help/simulink/slref/bandlimitedwhitenoise.html, 2022 (accessed August 1, 2022).

[20] R. Bulirsch, J. Stoer, and J. Stoer, *Introduction to numerical analysis*, vol. 3. Springer, 2002.

[21] G. H. Golub, J. M. Ortega, *et al.*, *Scientific computing and differential equations: an introduction to numerical methods.* Academic press, 1992.

[22] MathWorks®, "Solver." mathworks.com/help/simulink/gui/solver.html, 2022 (accessed August 1, 2022).

[23] MathWorks®, "linmod." mathworks.com/help/simulink/slref/linmod.html, 2022 (accessed August 1, 2022).

[24] D. W. Allan, "Statistics of atomic frequency standards," *Proceedings of the IEEE*, vol. 54, no. 2, pp. 221–230, 1966.

[25] Wikipedia, "Allan variance." en.wikipedia.org/wiki/Allan_variance, 2022 (accessed August 10, 2022).