Modeling and optimization of a MEMS membrane-based acoustic-wave biosensor

Submitted in Partial Fulfillment of the Requirements for

the degree of

Doctor of Philosophy

in

Biomedical Engineering

Jane E. Valentine

B.S., Mathematical Sciences, Carnegie Mellon University

Carnegie Mellon University

Pittsburgh, Pennsylvania

August 2013

Abstract

Rapid, reliable, and inexpensive detection of biological and chemical species is highly advantageous in numerous situations. The ability to simultaneously detect multiple targets, for example in medical or environmental testing settings, in areas where modern laboratory equipment is not widely available, is especially desirable. The combination of acoustic wave sensing and MicroElectroMechanical Systems (MEMS) technology leads to a sensor with these capabilities. In this thesis we describe the modeling and optimization of such a membrane-based acoustic wave MEMS biosensor.

Starting from an analytical model of the vibration behavior of an unloaded membrane, we model the vibration behavior of a mass-loaded membrane both computationally (using Finite Element Methods) and by using matrix perturbation analysis to develop a computationally efficient approximate analytical solution. Comparing the two methods, we find that our two models show excellent agreement for the range of mass loadings we expect to see.

We then note that we can alter sensor performance by controlling the placement of chemically or biologically functionalized regions on the membrane. Our approximate analytical model lets us efficiently predict the effects of functionalization geometries, and so we can optimize performance according to a number of metrics. We develop several optimization objectives to take advantage of our ability to control sensitivity and to multiplex. We develop precise formulations for the objective functions and for constraints, both physical and design-related. We then solve our optimization problems using two complementary

methods. The first is an analytical approach we developed, which is feasible for simpler problems, while the second is a stochastic optimization routine using genetic algorithms for more complex problems. Using this method we were able to confirm the solutions given by our analytical approach, and find solutions for more complicated optimization problems. Our solutions allow us to examine the tradeoffs involved in deciding where to place regions of added mass, including tradeoffs between patches and between modes. This helps to elucidate the dynamics of our system, and raises questions for further research.

Finally we discuss future research directions, including further optimization possibilities for single sensors as well as for systems of multiple sensors.

Dedication

In loving memory:

John Leroy Valentine

Stephen Gregory Szczepaniak

Ethel Patricia Laura O'Reilly Szczepaniak

Elma Jean Hall

Acknowledgements

I would not have been able to complete this research without the support and help of my colleagues, family, and friends. I am grateful to Todd Przybycien and Steinar Hauan for their advice and guidance, academic and otherwise. To members of my thesis committee, Jelena Kovačević, Jim Antaki, Jonathan Wickert, and Burak Ozdoganlar; for their valuable input and suggestions. To Russell Walker for his support during my entire tenure at Carnegie Mellon University. To my doctors, particularly Ernest Manders, Maurice Cerul, and John Helfrich, for supporting my health. I gratefully remember Loki and Gypsy for never judging me, and thank their successors, Elster and Ben, for the same.

I thank friends too numerous to list for always being there for me, and assure them that though I do not name them here, I appreciate their support and friendship whole-heartedly.

I am deeply grateful to my family for all their love and support. Thank you Brighten. Thank you Mom, Dad, Laura, Greg, John, Kristin, Molly, Ellen, Nat, Sarah, Christa, Dave, and everyone else.

Finally, I gratefully acknowledge the financial support of the Department of Homeland Security: Science and Technology Directorate, from whom I received both a Graduate Fellowship and a Dissertation Fellowship; and of the National Institutes of Health, whose grant No. 1 R21 EB00735-01 provided funding for my project.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

1.1 Biochemical sensors and applications

Sensors can be used to detect and measure a broad range of physical quantities or phenomena, from a planet orbiting a distant star [88] to blood glucose levels in a diabetic [23, 58, 72]. Of particular interest to us are biochemical sensors: sensors which detect the presence of a chemical or biological species in a sample. These sensors can be used in areas including environmental or agricultural monitoring[22, 67, 20, 71], clinical or research medicine [3, 78, 55, 1], clinical or research chemistry [42, 23], drug development or discovery [16, 39], food or water supply safety [41], or food processing monitoring [49, 60], among others.

Biochemical sensors may be based on a large number of different physical, mechanical, chemical, or biological principles, or systems comprised of combinations thereof. Due to the wide range of sensor systems in use, it is infeasible to present a comprehensive list of technologies. However, we can place sensor technologies into broad categories based on the measured property.

1.2 Biochemical sensor types and characteristics

Electrical sensors rely on a specific chemical target changing an electrical property of the sensor, sometimes via affinity binding. Examples include Metal Oxide Silicon (MOS) gate sensors [37], Ion Sensitive Field Effect Transistors (ISFETs) [37] and ion conducting polymer systems [54]. For an ISFET the measurand, or measured quantity, of the sensor is the current across the transistor. Field effect transistors rely on electric fields to control the conductivity across a channel; ISFETs in particular use the field effects due to ion concentrations in a sample to measure those concentrations. Impedance sensors [21] measure the change in the surface impedance (or current:voltage ratio under an applied steady-state AC) of an interface when target molecules are bound. Electrical impedance is a complex quantity, and is the AC circuit analogue of resistance in DC circuits.

Optical sensors take advantage of the physics of light absorption, emission, reflection, and refraction. To do so, they may rely on the optical properties of the sample and the target analyte, as Fourier Transform Infrared (FTIR) spectroscopy [76], in which the amount of light absorbed at various wavelengths is measured and absorption bands are compared to libraries of empirically known measurements. Other techniques, such as fluorescence micropscopy, rely on directly measuring the light emitted by flurophores in the sample. These fluorophores are functional groups within molecules, and different fluorophores are excited by different (but specific) wavelengths of light, and likewise emit light at specific and known wavelengths. Therefore an emission spectrum can be compared to spectra for known fluorophores and associated molecules. Alternatively, optical sensors may rely on the effects of the target on the optical properties of the sensor substrate, as in Total Internal Reflection Fluorescence (TIRF) [9, 10]. In TIRF, an evanescent wave is used to excite flourophores in a sample, and the resulting spectra examined. Optical sensors may exploit sample preparation techniques such as fluorescent tagging of targets, and use these artificially created or enhanced optical properties of the target as the basis for detection.

Enzyme-based catalytic sensors such as blood glucose monitors [23, 25] depend on an

enzyme-catalyzed reaction to convert the target in a sample to a detectable product. In the context of this categorization catalytic sensors are often used with another sensing technique in order to effect detection of the target molecules via detection of the product of catalysis. Exceptions include chemical test strips for blood glucose monitoring which are to be examined visually for changes in color after a blood sample is applied, and lateral flow immunoassay devices [87] such as home pregnancy tests. In the latter, the sample of interest moves along a solid substrate by capillary action; it first encounters and is mixed with a colored reagent, and then subsequently encounters test regions pre-treated with antibodies or antigens specific to the target of interest. The colored reagent will bind to these test regions depending on the presence or absence of the target species, and the substrate is then visually examined for the presence of the reagent in the test regions.

Mechanical sensors measure a change in position of a structural sensor element due to an adsorbed mass, such as microcantilevers used in bending or static mode [24], as well as acoustic wave sensors [29, 27, 28], which detect changes in the resonant frequency of a vibrating structural element due to the adsorption of mass. The measurand for a mechanical sensor may be the output frequency of a vibrating structure being externally driven [51], the optically detected shift in the position of a structural element [47], or the electrical resistance of a piezoresistive structural element [80], among others. Devices such as the Quartz Crystal Microbalance [89] measure not only frequency of the resonator by the dissipation as well; this information is used to quantify the damping of the system. Mechanical sensors may use affinity-binding to target specific analytes.

Due to the wide range in the fundamental operating principles of biochemical sensors, there is no single fabrication technique used to manufacture biosensors. One technique which is well-suited to the production of some biosensor technologies, including ISFETs, microfluidic devices incorporating catalytic sensor, and acoustic wave devices (all of which have been produced in this manner) is MicroElectroMechanical Systems (MEMS) fabrication [43]. MEMS technology uses microfabrication techniques, including many of those used in the semiconductor industry, to create devices with integrated structural and electronic

components. The components of MEMS devices are generally on the scales of ones or tens of microns, and are produced by applying patterning, etching, deposition, and milling steps to a substrate. MEMS devices are small, lightweight, and inexpensive to produce, and the ability to include on-board electronics reduces the need for external hardware. They are also easily integrated into larger systems of electronics, and require small sample sizes. These properties make MEMS technology a good candidate for the production of biochemical sensors.

1.3 Improving sensor performance and developing sensor arrays

Biochemical sensors vary widely in design, purpose, and fundamental operating principle. Depending on its use, such a sensor may be designed for robustness to noise [19], sensitivity [19], specificity [19], rapid response time [83], operating frequency range [27, 83], or the ability to detect multiple analytes simultaneously [74]. In addition, sensors may be integrated into sensor arrays to rapidly screen samples for large number of targets [85, 79, 59, 68, 47]. We expect that to be competitive, any new sensor must significantly improve over alternative existing sensors designed for the same purpose. Our group has developed a MEMS acoustic wave sensor for which the resonator is a membrane. We expect this sensor to compete in at least two of the above arenas. First, the high surface area to mass ratio of our membrane leads to increased sensitivity [75, 5]. Second, the ability of our sensor to detect multiple targets simultaneously means that a single membrane sensor can function as a sensor array. Further, both of these properties can be improved upon by optimizing the distribution of adsorbed mass on the membrane surface.

The effects of non-uniform mass distribution on the resonant frequencies of a resonating element have been addressed for several systems. The Quartz Crystal Microbalance has been studied by Ward et al. [33] and Vig et al. [77], who show that device sensitivity depends on the exact locations of mass changes. Microcantilever sensors are sometimes mass-loaded

at the tip to maximize sensitivity [53, 34], and Dufour et al. have modeled cantilevers designed with a small plate at the sensitive free end to increase the area available for mass binding [24]. Laugeson [40] has modeled the effects on eigenvalues of inhomogeneous circular membranes with radial mass density constraints, but does not solve the inverse problem. McCarthy [46] has solved the problem of recovering an approximate density distribution function on inhomogeneous membranes, but requires axial symmetry in both directions. In this thesis we will develope a model of the vibrational behavior of a membrane under unevenly distributed mass loading, and will use that model to optimize the sensitivity of our device, and to permit easy implementation of multi-plexing, or detection of multiple analytes simultaneously.

The development of sensor arrays for multiple chemical or biochemical targets ("electronic noses") [4, 2, 11] has focused in part on sensor heterogeneity [85], i.e. on incorporating different sensors, sensor mechanisms, or sensor characteristics (such as size or material) for target molecules such as hydrogen [11], primary alcohols [4, 2], lysozymes [15], and metals [53, 11]. We envision an array distinct from these in that it will use a homogeneous array of sensors, or even a single sensor, distinguished only by the appropriate sizes and locations of specific binding regions. Such a device would also be distinct from the multi-plexed paper microfluidic devices designed for use in use in developing regions [45, 44]. While these sensors can give quantitative measures of multiple analytes simultaneously, they are limited to analytes for which traditional lateral flow immunoassay techniques are appropriate and for which the paper devices can be prepared well in advance [45].

1.4 Thesis overview and contributions

This thesis discusses the mathematical modeling and optimization of a MEMS membrane-based acoustic wave biosensor. The main contributions of this thesis are:

- The development of a reduced-order model of the membrane response to mass-loading, which is highly accurate for the range of mass-loading we expect to see.

- An optimization framework comprising objective functions, constraints, and optimization methods. This framework allows us to optimize sensor performance under a wide range of conditions, and can be modified for use with different models or additional objective functions.

- A description of our sensor's response surface behavior under different optimization conditions, elucidating the complex tradeoffs in the optimization search space.

The thesis is organized as follows:

Chapter 2 In this chapter we discuss acoustic wave sensors in general, and give details on our specific design. We also discuss MicroElectroMechanicalSystems technology, including an overview of the manufacturing process and the MEMS elements that are used in our design.

Chapter 3 In this chapter we discuss our efforts to model the vibrational behavior of our sensor under variable mass-loading, including critical assumptions we made and their implications. We first developed a computational model, and subsequently an analytical approximation model. We discuss both, and show that they agree. We proceed to show that the analytical model can be inverted, allowing us to move from predicting device behavior to designing device behavior. Finally, we talk about a performance measure we define, called eigenvalue measure, and mention its implications for optimization.

Chapter 4 This chapter covers optimization in detail. We give a broad outline of the types of optimization possible, and specify our areas of focus based on several posited real-world cases. We discuss key assumptions about our system and their implications, and qualitatively describe the objective functions we will consider in the ensuing chapters. We then discuss in detail the development and mathematical formulations of our objective functions and constraints. We formulate objective functions and attendent constraints for cases where we are optimizing the placement of either two or three

regions of added mass to our membrane, and distinguish between those objectives which are generalizable to systems with more regions or more modes and those which are not.

Chapter 5 This chapter begins with a discussion of possible solution methods for our optimization problems. The methods we chose are described more fully, and we give detailed descriptions of our implementations of these methods. We then apply the methods to our optimization problems, and dicuss the significance of the results, paying particular attention to issues including the possibility that we are not solving to optimality, and the tradeoffs involved in multi-patch and multi-mode systems.

Chapter 6 In the final chapter we review the contributions we have made, and discuss further research directions related to the project, including a sensitivity analysis, additional objective functions, and multi-membrane optimization.

Appendix A This appendix contains examples of code used in the thesis.

Chapter 2

Background

2.1 Acoustic wave sensors

Acoustic wave sensors work by exploiting the fact that the resonant frequency of a structure decreases as its mass increases. It is therefore possible to detect target molecules by measuring the changes in the resonant frequencies of vibrating structural elements (resonators) induced by the adsorption of targets. Acoustic wave sensors can be customized to detect particular ligands by fabricating the resonators using materials chosen to bind the ligand, or by coating the resonators with ligates specific to the ligands in question

The choice of resonator varies across devices, and types in use include bulk-wave devices such as quartz crystal plates[89, 64], Love-wave devices using a bulk crystal substrate and interdigitated transducers [31], silicon plates with zinc oxide coatings, [30, 82], and cantilevered beams of silicon or silicon nitide [69, 63]. We have developed a MEMS acoustic wave biochemical sensor [32, 5] for which the resonater is a membrane. The high surface area to volume ratio of the membrane compared to competing technologies leads to increased sensitivity, and, coupled with our abilities to selectively functionalize regions of the membrane for target species and to excite higher-order modes, enables us to detect multiple targets simultaneously or to integrate redundancy in a single sensor. This capability invites the possibility of optimizing the placement of functionalized regions with respect to

characteristics such as sensitivity or accuracy.

Acoustic wave sensors can, in theory, have a resonant structure of any shape. In practice, however, the range of possible geometries is limited by fabrication techniques and electronics limitations. The most common acoustic wave sensors currently being studied are Bulk Acoustic Wave (BAW) devices such as the Quartz Crystal Microbalance (QCM) [89] and Surface Acoustic Wave (SAW) or Shear Horizontal Surface Acoustic Wave (SH-SAW) sensors [86, 62], Flexural Plate Wave (FPW) devices [26, 82, 81], and microcantilevers [53, 24]. Even within these categories, resonator geometries can vary; microcantilevers may be simple beams, beams with plates at the tip to increase the surface area available for binding, or V-shaped,among other possibilities. Table 2.1 summarizes the key properties of these sensors, including the key physical and material parameters determining frequency, typical operating frequencies f observed in the literature, and the range of noise levels n noted in the literature.

Resonator Type	**Key Parameters**	**Typical f**	**Typical n**
Bulk Acoustic Wave	resonator thickness film thickness density	31 - 400 MHz	0.2 - 55Hz
Flexural Plate Wave	flexural rigidity tension plate thickness mass density characteristic geometric parameters	5.5 MHz	0.3 - 2.1 Hz
Microcantilever	mass length inertial moment	0.005 - 5 MHz	0.01 - 2.1 Hz

Table 2.1: Comparison of some acoustic wave devices and their key charateristics.

Some work on membrane based acoustic wave sensors has been performed by Clark et al. [15], whose work focused on detecting changes in membrane tension. While their work did not allow for target specific sensing, it could be modified to do so by using molecular

imprinting when fabricating the polymer membrane.

2.2 Membrane-based acoustic wave sensors

Acoustic wave sensors take advantage of the fact that for a mechanical resonator, and increase in mass induces a decrease in the resonant frequency. For a simple harmonic oscillator (SHO), the relationship between mass m and a harmonic frequency f is dictated by the equation $f = \sqrt{\frac{k}{m}}$ where k is a constant determined by the physical shape and material properties of the resonator, as well as by the order of the chosen harmonic. In a mass-spring SHO, m is the inertial mass and k is the spring constant of the massless spring. Assuming that k is constant, we see that:

$$f(m) = \sqrt{\frac{k}{m}} = \frac{\sqrt{k}}{\sqrt{m}}. \tag{2.1}$$

A first order Taylor Series Expansion then shows

$$\frac{df}{dm} = \frac{-\sqrt{k}}{2m^{3/2}}. \tag{2.2}$$

And hence, by discretizing and rearranging, the change in resonant frequency may be expressed as

$$\Delta f = \frac{-\Delta m\sqrt{k}}{2m^{3/2}} = \frac{-\Delta m f}{2m}. \tag{2.3}$$

If we assume that the adsorbed mass is evenly distributed over the entire exposed area, then the change in mass Δm will be directly proportional to the surface area of the resonator. D'Amico et al. [19] define *sensitivity* as the fractional change in frequency, $\frac{\Delta f}{f}$, and use it as a figure of merit for acoustic-wave sensors. This definition allows direct comparison of sensors with different fundamental frequencies. By this definition, a sensor operating in the kHz range may be more sensitive than one operating in the MHz range, even if its total frequency shift is much smaller.

The analysis above implies that a high surface area to mass ratio, as is seen in a membrane, is desirable in improving the sensitivity of an acoustic wave sensor. Previous work in our group indicates that our proposed sensor could be several orders of magnitude more sensitive than competing acoustic wave devices such as microcantilevers and bulk acoustic wave devices [5]. Table 2.2, using data from Michael Bartkovsky's 2006 doctoral thesis, gives the fractional frequency changes, measuring the relative frequency shift per unit mass, for various competing devices.

Device	S_m
QCM	0.011
SAW	0.20
SH-SAW	0.18
FPW	0.38
Cantilever	0.09
MEMS Membrane	0.90

Table 2.2: Fractional frequency changes S_m for acoustic wave devices. Units are $\frac{\frac{Hz}{MHz}}{\frac{ng}{cm^2}}$.

Further, studies show that the sensitivity of a membrane sensor can be increased by an unequal distribution of adsorbed mass [73, 40]. We can select a particular distribution by choosing which regions of a membrane to functionalize with ligands for the target species. This opens up the possibility of using a single membrane to simultaneously detect multiple species, and the question of optimal mass distribution on the membrane surface.

2.3 MicroElectroMechanical Systems

There are several fabrication methods that could be used to make a membrane. We have chosen a method that is particularly well suited to the production of small-scale devices with both mechanical and electronic components.

MicroElectroMechanical Systems, or MEMS, refers to technology in which micron-scale electronic and mechanical components are produced using the same processes, often silicon microfabrication or micromachining, to form a single device with fully integrated components[43]. Examples of these components include electronic circuit components such as transistors, resistors, logical operators, and multiplexers; as well as simple structural components like suspended beams, cavities, wells, and walls; and complex structural elements like interdigitated combs, valves, and hinges.

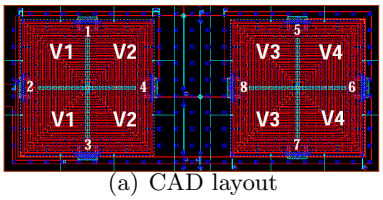
Traditional silicon micromachining, used to produce integrated circuits, requires the repeated use of three steps to modify silicon wafers: patterning of the exposed surface, etching of the surface, and deposition of materials onto the wafer. To allow for the formation of mechanical structures, additional etching steps are performed to release specific elements from the underlying silicon substrate. This permits the production of suspended mechanical structures which can act in conjunction with the embedded electronic circuitry.

The advantages of MEMS technology in general, and for sensing applications in particular, are myriad. Because MEMS devices can be produced using the same batch silicon manufacturing processes used for the production of computer chips and other electronics, the fabrication cost per device is typically low. Due to their small size, MEMS devices are also lightweight, leading to potential increases in sensitivity, as discussed above. As they require little to no ancillary electronics, they have low power consumption and are portable or easily integrated into larger systems. Finally, MEMS sensors would require only small volumes of sample solutions.

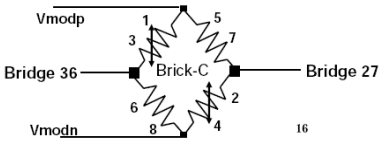
2.4 Our sensor

Our sensor is manufactured using silicon micromachining techniques. Each sensor is one of sixteen on a chip, and consists of a aluminum-oxide mesh structure suspended over a cavity in the silicon substrate of the chip and clamped on all four sides. This mesh structure is the transducer element for the sensor, converting electrical energy into mechanical energy,

and we model it as a membrane under tension. While the mesh structures for the current generation chip are square, measuring approximately 156 μm by 156 μm, in this paper we will consider only membranes with an aspect ratio of 1.5 : 1. This will allow us to avoid the degenerate case where multiple modes have the same frequency and eigenvalues. Two main structural mesh designs have been implemented for our membranes [5]; a grid design and a brick design, shown in Fig. 2.4. Each design has four variations. The designs have different void areas and total masses, but are functionally similar.



(a) CAD layout



(b) resistor configuration

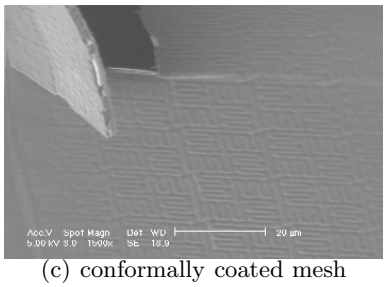
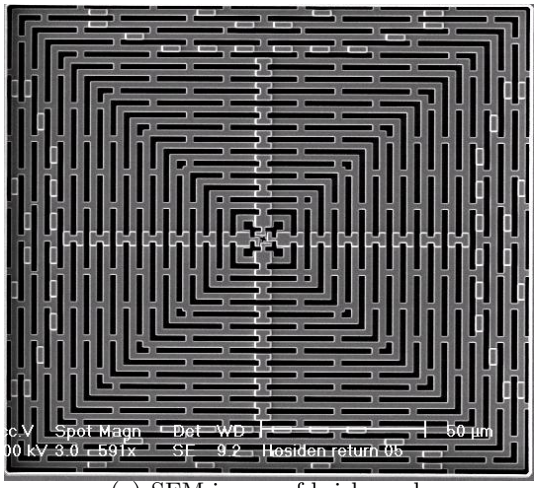

(c) conformally coated mesh

Figure 2.1: Sensor elements.


(a) SEM image of brick mesh

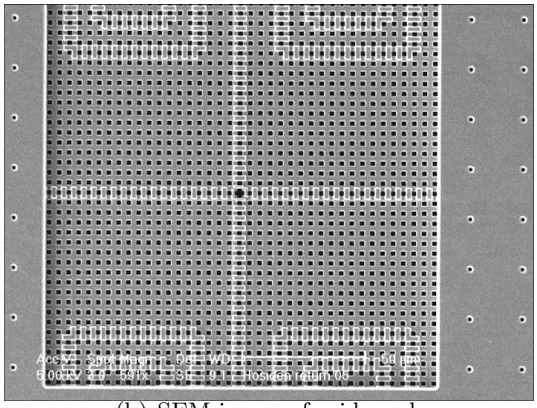

(b) SEM image of grid mesh

Figure 2.2: Images of different mesh designs. Photo credit: M. Bartkovsky

The electronic components necessary for actuating the sensor and for detecting the resultant

frequencies are integrated with the mechanical components. The membranes are actuated electrostatically via the application of a potential to the mesh, and their motion is independently detected using piezoresistive sensing elements embedded in the perimeter of the mesh.[5] Piezoresistive sensors take advantage of the fact that some materials change their electrical resistance in response to a mechanical stress. This type of sensing element is especially well suited to the detection of vibrations as the magnitudes of changes in resistance are not important as long as they are sufficient to allow for changes in frequency to be ascertained. Therefore precise calibration of piezoresistive behavior is unnecessary.

Each chip also contains internal logic control circuitry, and a set of 25 bondpads, which allow the chip to be electronically connected to external devices for both input and output. The membranes are fabricated as mesh structures for two reasons; to prevent in-plane tension and resultant out-of-plane buckling, and to allow for release of the membrane structure from the underlying materials via a chemical etching process. During post-processing, the mesh is covered with a comformal polystyrene coating, sealing all open spaces in the mesh and resulting in a solid, uniform surface suitable for functionalization. Despite the composite design, these structures can be modeled as membranes; for the remainder of this thesis, we will use the term *membrane* to refer to the composite structure of the metal mesh conformally coated with polymer, except where the term is used in its stricly mathematical sense.

In order to electrically isolate the aluminum mesh of the membranes from the electronics of the chip, the mesh is composed of multiple layers: a metal 1 layer is connected to a metal 2 layer by a non-conductive oxide layer. The metal 2 layer is not continuous, but depends upon the oxide layer to form a cohesive structure. This allows connections to individual portions of the membrane to access or shield sensing and actuation mechanisms. Details on the physical characteristics of this membrane, including the masses and void areas of the material layers which make it up, are given in Table 2.4.

Detection of a target analyte is a multi-step process. The membrane surface is first patterned via photolithography and functionalized for the specific targets in question; the

		Brick A	**Grid A**
Metal 1	Area[μm^2]	12279.20	177383.20
	Mass[ng]	21.03	29.77
Metal 2	Area[μm^2]	1362.64	2187.20
	Mass[ng]	2.33	3.75
Oxide	Area[μm^2]	13641.84	19570.40
	Mass[ng]	26.42	37.41
Vias	Number	124	228
	Mass[ng]	0.11	0.20
Polystyrene	Area[μm^2]	24336.00	24336.00
	Mass[ng]	73.3	not given
Total	Area[μm^2]	24336.00	24336.00
	Mass[ng]	123.19	not given

Table 2.3: Masses and areas for the **Brick A** and **Grid A** meshes; numbers for the other brick and grid meshes are comparable. The area measurement for polystyrene is given for the top surface of the membrane, but it fills the void areas from the other layers as well. All data from [5]

photolithography allows us to functionalize specific regions of the membrane for particular targets, opening up the possibilities of increasing sensitivity or specificity of the system. It is then immersed in a solution which does not contain any molecules that would bind to the surface, either specifically or non-specifically, but which matches the sample solution in density, temperature, and viscosity. The membrane is actuated, and the resonant frequencies and attendant eigenvalues of interest are determined. This forms the baseline for later comparison. The sensor is then immersed in the sample solution for a time sufficient to allow binding of any targets present. Finally, the sensor is actuated once again and the new resonant frequencies and eigenvalues are ascertained. The eigenvalue shifts can then be used to determine the added mass for each functionalized region, via the model developed below. It is possible to actuate membranes in sets of two, with each functionalized membrane paired with an unfunctionalized membrane. This second membrane would be actuated and exposed to the sample solution along with the functionalized one, and the differential information could be used to control for variables such as sample temperature, viscosity, density effects, and non-specific binding. Using a second membrane could also eliminate the need to take a baseline frequency spectrum in solution, or even, potentially, the need for an initial baseline spectrum.

Because our model does not account for damping due to immersion in a liquid solution, it may be advisable to modify the detection procedure to minimize the effects of damping. In this modified procedure the baseline resonance measurements would be performed in air. The membrane would then be exposed to the sample solution and target analytes bound, and then the membrane would be drained and dried, and the new resonance measures taken again in air. This procedure would decrease the effects of liquid damping, but introduces challenges in terms of designing the drying process to ensure it does not effect the quantity of target bound to the surface.

During use, our sensor chip would be contained in a clamshell packaging which would create a reservoir for the sample material and contain an inlet and an oulet to allow for sample delivery to the surface of the membranes. This reservoir is discussed by Bartkovsky [5], and

will not be considered for the remainder of this thesis.

Chapter 3

Modeling mass-loaded membranes

3.1 Membrane model

Given that we have a prototype device, we need to ensure that we are using an appropriate model. Thin, sheetlike structures such as the one described in the previous chapter can be modeled either as membranes or as plates. A uniformly stretched membrane is the two-dimensional analogue of a stretched wire [70], while a plate is the two-dimensional analogue of a beam [70]; membranes and plates have very different behaviors and are governed by different physical properties. Membranes are thin structures conforming to a surface, possess no bending rigidity, and can only support tensile loads. The primary physical characteristics determing membrane behavior are tension per unit length, T (which has a second order effect), mass m, and geometry. In contrast, plates have a positive bending rigidity arising from their elasticity and thickness, are non-conformal, and undergo flexural deformation. The key characteristics governing plate vibrations are elasticity, flexural rigidity (possessing a fourth order effect), the Young's modulus, and geometry. When modeling our device, it is important to verify that we are using the model which most accurately reflects its real-life behavior. Clark et al. give the conditions under which each model applies [15]; a membrane model may be used where:

$$C = \frac{D}{(T/t)R^2} \leq 1 \tag{3.1}$$

where D is the flexural rigidity, t is the thickness, and R is a characteristic length. Neumann et al. [52] provide estimates for both D and T, and values for t and R are extracted from the design specifications. Using this equation, we find that a membrane model incorporating only second order effects most accurately describes our system [73].

It is still possible that this is not the best model for the actual device, for multiple reasons. If our group's estimates for D or T are inaccurate, or if the mesh structure changes the effective t or R value for our device, we could find that we do not satisfy the inequality, and that therefore an alternative model (either a plate model or a combined membrane/plate model incorporating both second and fourth order effects) would be preferable. In addition, if the conformally-coated mesh forming the resonator structure introduces new dynamics, e.g. acts not as a uniform tensile sheet but as a series of coupled units of rods and strings, we could need a different model for each mesh design to accurately incorporate these effects. However, the focus of this thesis is the development of a theoretical framework for optimizing membrane behavior. For this aim, it is important to move forward with a specific model of the membrane's response to non-uniform mass loadings. We believe that our membrane model captures the important behaviors of our device; should experiments later prove it to be inadequate, an alternate model can be developed which could be used in our optimization routines in a similar manner.

Rayleigh [56] gives the following model, derived from a force balance, for the position function of a membrane undergoing transverse (out of plane) vibrations:

$$T(\frac{\partial^2 u}{\partial x^2} + \frac{\partial^2 u}{\partial y^2}) = \rho \frac{\partial^2 u}{\partial t^2} \tag{3.2}$$

wherein $u(x, y, t)$ is the position function of the membrane, with the $x - y$ plane parallel to the membrane surface. T is the tension per unit length, and ρ is the mass per unit area of the membrane. The left hand side of this equation represents the restoring force imparted by local curvature of the membrane, while the right hand side represents the mass force

on the membrane at position (x, y). This model assumes an undamped system. For our modeling purposes, we assume a rectangular membrane, clamped at the edges, as shown in 3.1. We also assume constant ρ and T. Under these conditions, the solution to this equation is given by

$$u(x, y, t) = \sum_n \sum_m \psi_{mn} \sin \frac{m\pi x}{a} \sin \frac{n\pi y}{b}. \tag{3.3}$$

Here m and n represent the harmonic modes of a rectangular membrane, while a and b are the dimensions of the sides of the membrane. Each of the summands is an eigenmode of the position function, and their coefficients ψ_{mn}, which are functions of time, depend on the initial conditions. As the name suggests, the eigenmodes are orthogonal to each other and are each associated with an eigenvalue λ given by

$$\lambda_{mn} = \pi^2 \frac{T}{\rho} (\frac{m^2}{a^2} + \frac{n^2}{b^2}). \tag{3.4}$$

Since each eigenvalue is related to the angular frequency ω of the corresponding eigenmode by $\lambda = \omega^2$ and $f = \frac{\omega}{2\pi}$ this allows us to directly calculate the frequency of vibration for each eigenmode.

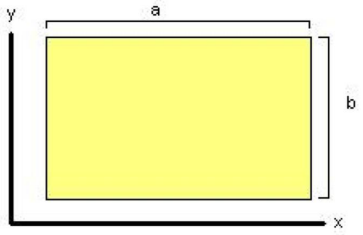
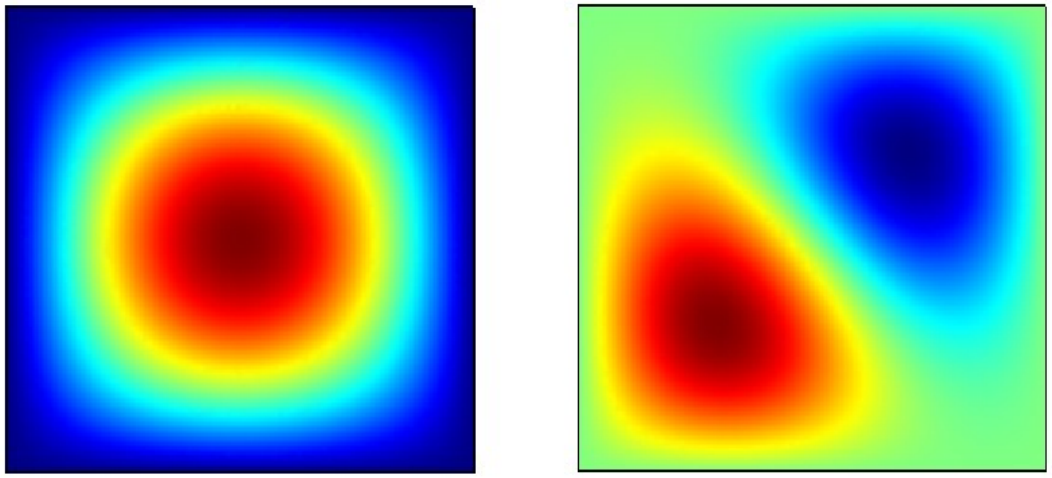


Figure 3.1: Membrane with length a and width b

Anti-nodes are points of maximal out-of-plane motion. The (m, n)th eigenmode will have $m \times n$ anti-nodes: m in the x-direction, and n in the y-direction. Modes other than the $(1, 1)$ mode will also have *nodal* lines, lines of zero out-of-plane motion. In a square membrane,

the (m, n)th and (n, m)th eigenmodes, while distinct, have identical eigenvalues. Therefore, a mode having $m \times n$ anti-nodes may be the (m, n)th mode, the (n, m)th mode, or a linear combination of the two.



(a) The (1,1) eigenmode. The anti-node, or region of maximal motion, is in the center

(b) Eigenmode resulting from (2,1) - (1,2). The nodal line is recognizable along the diagonal.

Figure 3.2: FEMLAB representation of two eigenmodes of a square membrane. Color indicates out-of-plane motion, with red having the largest positive value, and blue the largest negative value. Green indicates zero motion.

3.2 Differential mass loading

The derivation above is for a uniformly distributed mass across the entire surface of the membrane. However, mass sensitivity for a harmonic mode can be increased by concentrating the mass, and placing it on a specific region. The largest increase is seen when the region in question is located over an anti-node (a point of maximum motion) for the eigenmode being used. Table 3.2 shows the simulation results for a mass localization case study. In this study, we assume a square membrane with side lengths of $100\mu m$,an areal density of $4200\frac{kg}{m^2}$, and a uniform tension of $1.154 \times 10^6 \frac{N}{m}$ [7, 52]. A constant mass was distributed evenly over the entire membrane surface, and the resulting frequency shift from the unloaded case for

the first harmonic was calculated. The same mass was then distributed evenly over a $20\mu m$ by $20\mu m$ subregion placed in the center of the membrane, and again the frequency shift was calculated. This analysis was performed for two different levels of added mass, $0.07776ng$ and $0.15552ng$, using the methods described in 3.2.1. These levels were chosen by assuming binding of a single antibody (in the case of $0.07776ng$) or of an antibody-antibody complex to the membrane surface. In addition, we assume that the total mass bound was limited by the available surface area for the localized case, with the further restriction that only 54% of that area allows binding. This number is derived from Random Sequential Adsorption (RSA) and indicates the amount of surface area covered when spatially randomized sequential binding occurs [48]; the relative placement of adsorbed molecules can allow a significant space between them, but one which is insufficient to allow another molecule to occupy that space.

Because the fundamental frequency for the unloaded membrane is the same in all cases, the results summarized in Table 3.2 show that we can significantly increase our sensitivity (in this case by nearly a factor of four) by using an unequal distribution of mass.

Added Mass	Δf **Uniform Mass**	Δf **Localized Mass**
0.07776	50 Hz	190 Hz
0.15552	100 Hz	378 Hz

Table 3.1: Effect of mass localization on membrane frequency shift for the (1,1) mode.

3.2.1 Computational model

Membrane vibration is simulated using the modeling package COMSOL [35]. COMSOL uses finite element methods (FEM) to numerically model analytically intractable partial differential equations (PDEs) or ordinary different equations (ODEs). Finite element methods discretize the given continuous system modeled by PDEs or ODEs, reducing a problem of infinite degrees of freedom to one with a finite, although generally very large, number of degrees of freedom. The problem is formulated as a large system of coupled equations,

which may be linear or non-linear, and this system is then solved numerically.

COMSOL allows the user to choose the among pre-programmed models, as well as solver types, parameters, and the scale of the discretization, to match the requirements of the problem and ensure a sufficiently close approximation. COMSOL solutions depend on the system of equations entered, the boundary and initial conditions, and the solver, all of which are entered by the user. Our membrane vibration model was implemented as a two-dimensional partial differential equation model using the general form. The geometry of the problem was defined numerically, and the boundary conditions (membrane edges fixed on all four sides) entered as Dirichlet conditions. Values for tension, mass, added mass were entered as constants which were subsequently entered as parameters for each geometric region; this allows user-defined regions, or subdomains, to have different properties. In our case, we held the tension and mass coefficients constant over the entire membrane, and varied only the added mass coefficient for specific subdomains.

To insure a stable, accurate, and discretization-independent solution, we solve all problems at multiple levels of discretization. The lowest number of discretization points past which no measureable improvement or change in solution (within one part per million) is noted determines the level of discretization we use. In most cases we achieved an accurate solution for the required number of eigenvalues (between six and thirty) using the default discretization level for a triangular mesh; depending on the subdomains defined, this resulted in approximately 1000 mesh elements.

In the simulations performed for this paper, we use the parameters given in Table 3.2. The material properties are estimates based on preliminary membrane designs [7, 52]; and the membrane size was chosen to be close to that used for the actual device. For mathematical reasons (see Sec. 3.2.2), we chose a non-square membrane. The size of the functionalized regions was chosen based on the experimental work [6] on patterning the membrane for selective functionalization.

3.2.2 Matrix perturbation analysis

While the FEM simulations produce highly accurate approximations of the steady state membrane vibration problem, we must keep in mind that the goal is to optimize the performance of membranes and of arrays of membrane. Optimization at any level [8] requires an inner loop to describe the fundmental system behavior of interest, in this case the membrane eigenvalue shifts resulting from an uneven distribution of adsorbed mass. The COMSOL simulations provide numerical results, but no information as to the governing behavior of the system, such as gradient values. Running an optimization program with COMSOL simulations as the innermost loop would require a large number of individual simulations, each of which is computationally expensive. If instead we could analytically approximate the eigenvalue response of the system to added distributed masses, we could enjoy the advantages of additional information about the system, as well as decreased computational demands. Matrix perturbation analysis or perturbation theory [17] is a commonly used method to derive approximate solutions to perturbed systems of differential equations when the exact solution to the unperturbed system is known.

We follow the perturbation approach developed by Wickert et al. [84, 13] in our analysis. To perform the perturbation analysis we rewrite our differential equation in operator notation:

$$\rho(x,y)u_{tt} - T\bigtriangledown^2 u = 0 \tag{3.5}$$

This equation takes into account the fact that our density ρ may vary with x and y. For simplicity, we use u_{tt} to denote $\frac{\partial^2 u}{\partial t^2}$ and u to denote $u(x,y,t)$. Now we define the *mass operator* M, a linear function of the added mass in a particular region, as

$$M = \rho(x,y) = \rho_0 h + \epsilon m(x,y). \tag{3.6}$$

where $\rho_0 h$ is the areal density of the unloaded membrane, and $\epsilon m(x,y)$ is the perturbation in the density. Here ϵ is a small, dimensionless, scaling parameter, and $m(x,y)$ is an areal

density. In addition we define the *stiffness operator* K:

$$K = -T\bigtriangledown^2 \tag{3.7}$$

We note that the mass operator consists of an inital mass operator, corresponding to $\rho_0 h$ and an added, perturbed mass operator, corresponding to $\epsilon m(x, y)$, so we may write:

$$M = M_0 + \epsilon M_1 \tag{3.8}$$

and

$$(M_0 + \epsilon M_1)u_{tt} + Ku = 0 \tag{3.9}$$

We know that the solution to our unperturbed system has both a frequency component dependent on ω and a shape component ϕ, giving:

$$u(x, y, t) = e^{i\omega t}\phi(x, y) \tag{3.10}$$

The eigenvalue problem associated with our PDE is therefore:

$$K\phi = \omega^2 M\phi = \lambda(M_0 + \epsilon M_1)\phi \tag{3.11}$$

In this analysis we assume that the ϕ components are orthonormal with respect to each other.

Next we assume that our mode shapes and eigenvalues are perturbed in a manner similar to the mass operator: $\phi_{mn} = \phi_{mn_0} + \epsilon\phi_{mn_1}$ and $\lambda_{mn} = \lambda_{mn_0} + \epsilon\lambda_{mn_1}$. [1] We then substitute these two expressions into 3.11 and examine separately the terms which are zeroth and first order in ϵ.

The zeroth order terms return the unperturbed eigenvalue problem, and the first order terms give the following:

[1] Here m and n denote the examined mode; note that this m is an index, and is unrelated to the volume density expression $m(x, y)$.

$$K\phi_{mn_1} - \lambda_{mn_0} M_0 \phi_{mn_1} = \lambda_{mn_1} M_0 \phi_{mn_0} + \lambda_{mn_0} M_1 \phi_{mn_0} \tag{3.12}$$

There are two unknowns in this equation: ϕ_{mn_1} and λ_{mn_1}. However, if we examine the left hand side, we see that we use the same operator, $K - \lambda_{mn_0} M_0$ as in 3.11. This is a singular matrix operator. Using Fredholm's Alternative Theorem [50] we can solve 3.12 by restricting λ_{mn_1} so that the right hand side of 3.12 is in the null-space of $K - \lambda_{mn_0} M_0$. This is equivalent to setting $\lambda_{mn_1} M_0 \phi_{mn_0} + \lambda_{mn_0} M_1 \phi_{mn_0}$ orthogonal to each of the unperturbed mode shapes ϕ_{mn_0}. We observe that $\phi_{mn_0} = C_{mn} \sin \frac{m\pi x}{a} \sin \frac{n\pi y}{b}$. This observation leads to the solution:

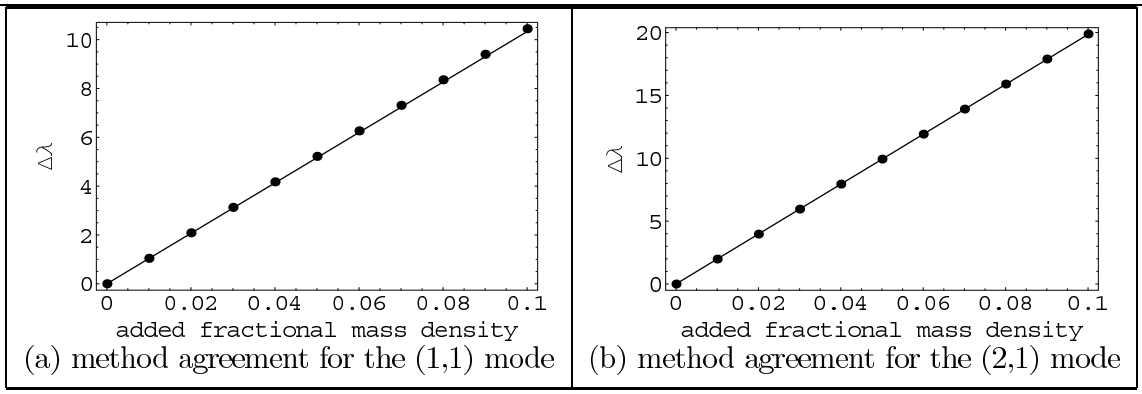
$$\lambda_{mn} - \lambda_{mn_0} = -\int_0^a \int_0^b \epsilon M C_{mn}^2 \sin^2(\frac{m\pi x}{a}) \sin^2(\frac{n\pi y}{b}) \mathrm{d}x\mathrm{d}y \tag{3.13}$$

From this equation it is clear that a selective spatial mass distribution can indeed increase the frequency sensitivity of our sensor.

The use of Fredholm's Alternative Theorem requires that the eigenvalues λ_{mn} all be distinct. This imposes a constraint on the membrane geometry, in that the side lengths a and b may not be equal. This condition is sufficient to prevent the degenerate case of multiple identical eigenvalues. However, if two eigenvalues are close, as in a nearly square membrane, we may observe a change in the order of eigenvalues when sorted by size (as opposed to being sorted by the (m, n) index) under differential mass loading conditions. That is, when eigenvalues are nearly the same, a significant change in one under a particular mass distribution may result in the smaller eigenvalue (for the unloaded membrane) becoming the larger eigenvalue (for the loaded case). Although it may be possible to take advantage of this behavior, it may also serve as a source of confusion, especially when interpreting (possible noisy) output from our device, and so we choose to eliminate it by considering only relatively large aspect ratios. The aspect ratio we choose for most of our calculations is 1.5:1; avoiding such issues.

3.2.3 Model agreement

To demonstrate the excellent agreement between FEM simulation results and the perturbation analysis results, we show in Tables 3.3 and 3.4 and in Figure 3.3 the calculated frequency shifts caused by mass loading for a scaled system. In Eq. 3.13, the constants C_{mn}^2 need to be appropriately set, or the eigenvalue shifts predicted by this equation will be in error by a constant factor. To appropriately calibrate the results, we find these constants by setting the predicted eigenvalue shift resulting from a uniform density increase over the entire membrane, and set it equal to the known analytical result for such a shift.



(a) method agreement for the (1,1) mode (b) method agreement for the (2,1) mode

Figure 3.3: Agreement between perturbation analysis (solid lines) and simulation (points) in predicting eigenvalue shifts due to added mass. Simulations used the parameter values given as Set II in Table 3.2. Regions of added mass were centered on anti-nodes for the modes being simulated.

We compared the simulation and analytical perturbation results for the (1,1) and (2,1) harmonics, centering rectangular regions of added mass spanning a range of anticipated magnitudes at an anti-node of each respective mode. Calculated eigenvalue shifts are listed in Tables 3.3, and 3.4; the parameter values are listed under Set I in Table 3.2.
Throughout this section we compare only the absolute changes in eigenvalues; as the baseline eigenvalues and the added masses are held constant for any particular case, it is not necessary to compare sensitivity values.

Parameter	**Value: Set I**	**Value: Set II**
Tension T	$1.154 \times 10^6 N/m^2$	$1.0 force/length^2$
Density ρ	$3600 kg/m^2$	$0.0036\ mass/length^2$
Membrane length x	$150 \mu m$	$1.5\ length$
Membrane width y	$100 \mu m$	$1.0\ length$
Region length Δx	$10 \mu m$	$0.1\ length$
Region width Δy	$10 \mu m$	$0.1\ length$
Added density $\Delta \rho$	$0 - 0.1 \times \rho$	$0 - 0.1 \times \rho$

Table 3.2: Simulation parameter sets I and II. Set I is based on the expected material and geometric properties of the device. Set II uses a scaled system of units, allowing for easy calculation and scaling of results [7, 52].

$\Delta \rho$	$\Delta \lambda_{11}$ **FEM**	$\Delta \lambda_{11}$ **Pert. Anal.**	**%Error**
0.01	1.043	1.0332	0.94
0.02	2.087	2.0664	0.99
0.03	3.131	3.0996	1.00
0.04	4.176	4.1328	1.03
0.05	5.221	5.1661	1.05
0.06	6.265	6.1993	1.05
0.07	7.311	7.2325	1.07
0.08	8.356	8.2657	1.08
0.09	9.402	9.2989	1.10
0.10	10.448	10.3321	1.11

Table 3.3: Calculation agreement for eigenvalue shifts using FEM simulations and perturbation analysis - in this example there is a single patch of size 0.1 x 0.1 units, centered on the membrane. The sensitivity for this position and mode is 2.61

$\Delta\rho$	$\Delta\lambda_{21}$ **FEM**	$\Delta\lambda_{21}$ **Pert. Anal.**	**%Error**
0.01	1.985	1.985	0.02
0.02	3.972	3.970	0.04
0.03	5.959	5.955	0.06
0.04	7.946	7.940	0.08
0.05	9.935	9.925	0.10
0.06	11.924	11.910	0.12
0.07	13.914	13.895	0.14
0.08	15.905	15.880	0.16
0.09	17.896	17.865	0.18
0.10	19.889	19.850	0.20

Table 3.4: Calculation agreement for eigenvalue shifts using FEM simulations and perturbation analysis. This example uses a single patch of size 0.1 x 0.1 units, centered over one of the anti-nodes for the (2,1) mode. The sensitivity for this position and mode is 2.61

In addition, we have performed a more detailed analysis of a probable added mass, to ensure that our calculations cover an appropriate range. We assume that our bound molecules are generic antibodies, with a molecular weight of $180,000$ and an equivalent spherical radius of $6.2nm$ [65]. We estimate the surface area SA, in $\mathring{A}^2$, of the antibody, as:

$$SA = 6.3mw^{0.73} \tag{3.14}$$

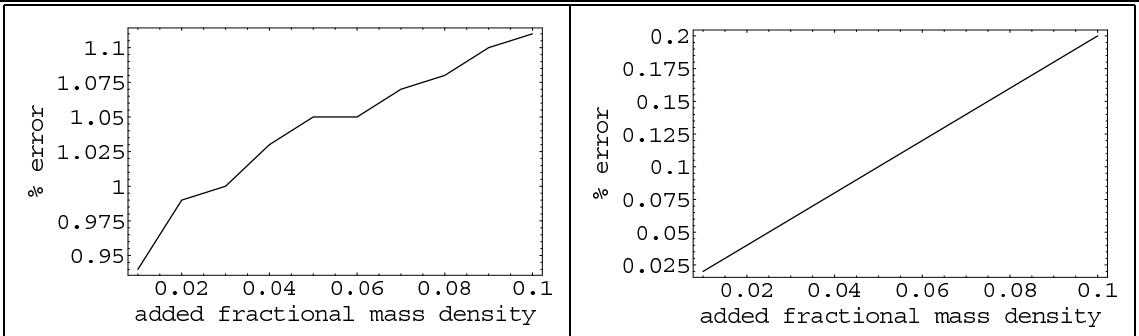
where mw is the molecular weight of the antibody [38].We then assume that a uniform one-molecule layer of water is bound to the surface of the molecule, and will move with it. This gives us an estimate of the total effective molecular weight of the antibody. Craig et al. [18] show that bound water molecules may significantly increase the mass, adding as much as five times the molecular weight to the effective mass. This is therefore a conservative estimate of the bound water mass. Thus it is necessary to determine if the error will lie within

acceptable limits when such masses are bound. We assume random sequential absorption coverage of the binding regions [12, 48], and calculate the number of antibodies bound if we model the antibody as a sphere, and assume a "footprint" of $\pi(6.2nm)^2$. The effect of the increased hydrodynamic radius is counteracted by the reduced number of molecules capable of binding to the membrane surface, so this case is covered by the work shown in Figure 3.3 and in Tables 3.3, and 3.4.

In Figure 3.3 we show the agreement between the perturbation analysis results (solid line) and the simulation results (points) for a scaled system. We use the parameter values listed under Set II in Table 3.2. These are scaled versions of the parameters in Set I, and the scaling was chosen to minimize numerical scale problems arising in FEM solutions using COMSOL. For both the (1,1) and the (2,1) mode, we calculate the change in eigenvalue resulting from increasing the areal mass density on a discrete region of added mass. We performed simulations and perturbation analysis calculations for added density values between 0.0 and 0.1, with a 0.01 step size; this interval covers the expected range of probable added mass values. For the (1,1) mode the region of added mass was centered at (0.75, 0.5), the anti-node for that mode. For the (2,1) mode, the region was centered at (0.5, 0.5), one of two anti-nodes for that mode.

In Figure 3.4 we show the error between the eigenvalue calculations using the two methods, given as the difference between the simulation and perturbation analysis results divided by the perturbation analysis results. The error increases as the added mass increases, which is expected due to the increase in magnitude of the higher order terms neglected in the perturbation analysis.

So far we have shown that we can accurately model the effects of a single region of added mass on the eigenvalue of a particular mode. We wish to further show that our approximation formula is accurate when we change the position, size, and number of regions of added mass; these can then be used as design variables in future optimization efforts. To demonstrate that Eq. 3.13 correctly predicts eigenvalue shifts resulting from patches of different sizes, we performed FEM simulations for two different system configurations. In



(a) error between methods for the (1,1) mode

(b) error between methods for the (2,1) mode

Figure 3.4: Error between perturbation analysis and simulation methods for predicting eigenvalue shifts. In both modes, the error increases roughly linearly with the added fractional mass density, but remains sufficiently small over the range of expected densities.

one, we again placed a patch of size 0.1 by 0.1 units on the anti-node for the (1,1) mode, and calculated the resultant eigenvalue shifts for two different added mass densities, representative of expected densities for bound antibodies. In the second, we placed a larger patch off-center so that for the same added density, we would acheive the same eigenvalue shift; we again calculated the eigenvalue shifts for both density levels. The results of these calculations are shown in Table 3.5 and show that changes in patch size and position do not affect the accuracy of the perturbation analysis approximation.

To show that the eigenvalue shifts resulting from multiple regions of added mass are additive, we simulated the eigenvalue response of the (1,1) mode to two discrete regions of added mass each considered separately, as well as to the case wherein both regions had simultaneous density increases. The results, shown in Table 3.6 show that for the expected density levels, the resulting error is very low, and so we may assume additivity.

Limitations and binding range

To ensure that our approximate analytic solution will be accurate for the expected range of added masses we first perform a more detailed analysis of a probable added mass. We

Patch	$\Delta\rho$	$\Delta\lambda_{11}$ **FEM**	$\Delta\lambda_{11}$ **Pert. Anal.**	**%Error**
1	0.054	5.626	5.579	0.008
1	0.076	7.937	7.854	0.010
2	0.054	5.629	5.564	0.011
2	0.076	7.938	7.832	0.013

Table 3.5: Calculation agreement for eigenvalue shifts in the (1,1) mode using FEM simulations and perturbation analysis. Patch 1 is 0.1 x 0.1 units, with the lower left corner at (0.7, 0.45) and a sensitivity of 2.61, while Patch 2 is 0.145 x 0.1398 units with the lower left corner at (0.935, 0.235), and a sensitivity of 1.28

.

$\Delta\rho$	$\Delta\lambda$ **Patch 1**	$\Delta\lambda$ **Patch 2**	$\Delta\lambda$ **Total (FEM)**	$\Delta\lambda$ **Summed**	**%Error**
0.054	5.626	2.676	8.306	8.302	0.04
0.076	7.937	3.787	11.708	11.724	0.14

Table 3.6: Calculation agreement for eigenvalue shifts in the (1,1) mode using FEM simulations and showing that the eigenvalue shifts are additive for discrete patches. Patch 1 is 0.1 x 0.1 units, with the lower left corner at (0.7, 0.45) and Patch 2 is 0.1 x 0.1 units with the lower left corner at (0.935, 0.235). These are the same positions as in the previous simulation, but the patch size for Patch 2 has been reduced.

assume that our bound molecules are generic antibodies, with a molecular weight of $180,000$ and a diameter of $12.4nm$ [65]. We estimate the surface area SA, in $\mathring{A}^2$, of the antibody, as:

$$SA = 6.3mw^{0.73}. \tag{3.15}$$

where mw is the molecular weight of the antibody [38].We then assume that a uniform one-molecule layer of water is bound to the molecule, and will move with it. This gives us an estimate of the total effective molecular weight of the antibody. We again assume RSA coverage of the binding regions, and calculate the number of antibodies bound if we model the antibody as a sphere, and assume a "footprint" of $\pi(6.2nm)^2$. We then use this information to calculate the total added mass on a region when an antibody-antibody complex (the largest ligand-ligate complex we expect to see) is bound to the surface. We find that our added mass density is approximately $0.00034\ kg/m^2$, which is less than one tenth of our baseline mass density of $0.0036\ kg/m^2$; this means that in practice the maximum value of ϵ, our dimensionless scaling parameter, will be less than 0.1; for our purposes we assume $\epsilon \leq 0.1$.

Now we wish to ensure that our matrix perturbation analysis result will be sufficiently accurate for an added mass density of 10%. To do so we note that we could express the eigenvalue shift as an infinite sum:

$$\Delta\lambda = \gamma_1\epsilon + \gamma_2\epsilon^2 + \gamma_3\epsilon^3 + \dots \tag{3.16}$$

where the γ_i are dependent on the mass distribution function. Our matrix perturbation analysis solved for the γ_1 term, neglecting higher order terms. We can extract the expression for γ_1 from Eq. 3.13:

$$\lambda_j - \lambda_{(j_0)} = -\int_0^a \int_0^b MC_{mn}^2 \sin^2(\frac{m\pi x}{a}) \sin^2(\frac{n\pi y}{b}) \mathrm{d}x\mathrm{d}y \times \epsilon = \gamma_1\epsilon \tag{3.17}$$

If we assume that the γ_i are all of the same order, than we can approximate the relative

error of our first order matrix perturbation analysis by ϵ^2 for small ϵ, as higher order terms would contribute negligibly. As the maximum value for ϵ is 0.1, we expect the relative error to be on the order of 0.01, which is acceptable. Therefore we assume that our perturbation analysis results are accurate for our expected range of binding conditions.

3.3 Model inversion

In the analysis above, we determined the effect of a distributed mass load on the eigenvalues, and hence the frequencies, of the membrane motion. However, in practice we wish to do the reverse: determine the mass added (in a known spatial distribution) when a particular shift in the eigenvalues is measured. This inverse problem is easily solved. From above, we know that the shifts in eigenvalues are linear in the added masses m_j, i.e.

$$\Delta\lambda_i = c_{i1}m_1 + c_{i2}m_2 + ... + c_{in}m_n \tag{3.18}$$

or more compactly,

$$\Delta\hat{\lambda} = \hat{\mathbf{c}} \bullet \hat{m} \tag{3.19}$$

where $\hat{\mathbf{c}}$ is the correction matrix determined by the perturbation analysis. Because $\hat{\mathbf{c}}$ is invertible, which fact is clear from its form (see Eq.3.13), we can solve for $\hat{m}$ given only the matrix $\hat{\lambda}$, provided the number of measured harmonics is greater than or equal to the number of regions of added mass, and that all of the individual equations are linearly independent. This latter condition will be an important constraint in some of the optimization problems we will consider. When it is not satisfied our system of equations is underdetermined, and only partial solutions will be available.

3.4 Positions and level curves

The perturbation analysis results demonstrate that for a given arbitrary mass distribution we can calculate the resultant eigenvalue shifts, and that given the spatial distribution of mass, absent magnitudes, and the eigenvalue shifts generated by the system, we can determine the added masses. However, during the design process we can choose which regions of the membrane will be functionalized. Some functionalization schemes will perform better than others. With this in mind we aim to formulate an optimization procedure to maximize device performance in terms of discrimination, sensitivity, or signal strength. There are two fundamental ways to approach this in our optimization schemes.

One option is to consider each arrangement of functionalized regions in physical space, where each patch is associated with a length, width, and a point on the membrane surface. This permits us to directly implement constraints on patch positions, both relative to the membrane and to other patches. However, due to symmetry and continuity, a specific position for a region of a given size is only one of many such positions that would yield the same eigenvalue shifts, in a given mode, for a given density factor m_j. We can therefore consider families of equivalent positions for each mode.

This leads to the second approach, where we seek to identify which functionalization patterns will result in the same frequency response for a given mode, as measured by the change in eigenvalue. We first attempt to understand how the choice of mass placement affects the eigenvalue shift for the case of a single region of fixed size and mass placed on the membrane. This will permit us to define our objective function in terms of the relative frequency shifts desired in each mode, instead of in terms of region placement.

The membrane provides a continuum of possible placements, several of which will result in the same eigenvalue shift. We desire a measure which approximates the relationship between the position of the added mass and the induced eigenvalue shift. For the first harmonic, one simple approach is to consider the planar Euclidean distance from the center of the added region to the anti-node. We also considered a measure in which two regions

are equivalent if their centers lie on points which, in the unperturbed solution, have the same maximum amplitude during vibration. Both of these measures are straightforward to calculate.

In a perfectly circular membrane, these two conditions are equivalent, and furthermore, can be used as an eigenvalue measure. This is due to the rotational symmetry of the system. However, a rectangular membrane such as ours has only two axes of symmetry. This means that the eigenvalue shifts resulting from two regions which are placed equivalently according to either of these two measures may be significantly different. As the aspect ratio of each modal subregion increases, these measures diverge more and more severely from each other and from the true solution.

We propose instead to invert the perturbation analysis result. Because the original result is a many-to-one function, the inversion will generate a set of points. These points are equivalent according to an eigenvalue measure, as opposed to a distance measure, as suggested above. Therefore, for each eigenvalue shift there exists a level curve (or, in the case of higher modes, a set of non-contiguous level curves) in the spatial domain; each curve is comprised of points which are the same *eigenvalue distance* away from the anti-node. Placement of the region of added mass anywhere on the same level curve will result in the same eigenvalue shift.

From the form of Eq. 3.13, reproduced here,

$$\lambda_j - \lambda_{(j_0)} = -\int_0^a \int_0^b \epsilon M C_{mn}^2 \sin^2(\frac{m\pi x}{a}) \sin^2(\frac{n\pi y}{b}) \mathrm{d}x\mathrm{d}y \tag{3.20}$$

We can see that inverting it for given mass distribution function M (a function of both x and y) requires integrating the right hand side. For an arbitrary mass distribution function this is not analytically feasible, but by placing appropriate restrictions on M we can guarantee an integrable function.

For the case of a rectangular region of added mass, i.e. the case where M is identically zero everywhere but on a rectangular subsection of the membrane, denoted by $[x_1, x_2] \times [y_1, y_2]$ and where the density of the added mass is uniform across that subsection, Eq. 3.13 becomes separable and can be integrated analytically, yielding an even less computationally intensive

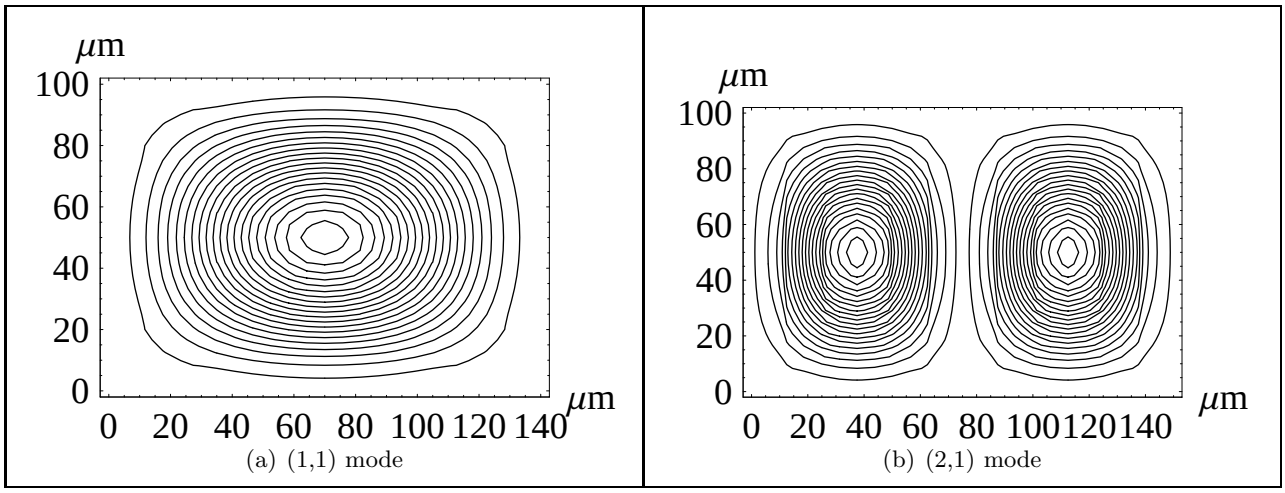
formula for the eigenvalue shift:

$$\lambda_j - \lambda_{(j_0)} = -\epsilon\, C_{mn}^2 \left[\frac{x}{2} - \frac{\sin(\frac{2m\pi x}{a})}{\frac{4m\pi x}{a}} \right]_{x_1}^{x_2} \cdot \left[\frac{y}{2} - \frac{\sin(\frac{2n\pi y}{b})}{\frac{4n\pi y}{b}} \right]_{y_1}^{y_2} \tag{3.21}$$

This result can also be generalized for arbitrary patch shapes, provided a one-to-one mapping from the given mass distribution to an equivalent rectangular one; in this thesis we restrict ourselves to the rectangular case. Because the eigenvalue shifts due to different weighted regions are additive in this approximation, we can calculate the impact of each region individually.

Eq. 3.21 allows us to calculate the aforementioned level curves for the response surface of each mode; sets of such level curves for the (1,1) and (2,1) response surfaces are shown in Fig. 3.5; simulation parameters are those for Set II listed in 3.2. While these response surfaces strongly resemble the unperturbed mode shapes for their respective modes, there are important differences. The response surfaces have uniformly non-positive values, since they represent the eigenvalue shifts instead of movement into or out of the plane of motion. The response surfaces will also change with changes in the size or shape of the functionalized region; each one is specific to the particular functionalization scheme. However, nodes and anti-nodes for a mode will coincide with their counterparts on a response surface.

This approach allows us to frame our objective in terms of the relative frequency shifts desired in each mode. Using multiple modes will constrain our problem and allow us to efficiently find sets of equivalent functionalization geometries.



(a) (1,1) mode (b) (2,1) mode

Figure 3.5: Level curves for the eigenvalue shifts induced by regions of added mass.

Chapter 4

Optimizing ligand patterning

4.1 Single membrane optimization

The development of sensor arrays for multiple chemical or biochemical targets ("electronic noses") [4, 2, 11] is often based on two sometimes complementary strategies. The first is sensor heterogeneity [85], wherein different sensors, sensor mechanisms, or sensor characteristics (such as size or material) are incorporated for the detection of target molecules such as hydrogen [11], primary alcohols [4, 2], lysozymes [15], and metals [53, 11]. The second is the use of at least one individual sensor per target analyte, with additional sensors sometimes used for redundancy. We propose instead a pseudo-array consisting of a single sensor capable of simultaneous detection or built in redundancy. This capability arises from the acoustic characteristics of our sensor and from our ability to functionalize appropriate sizes and locations of specific binding regions. A similar system has been implemented for surface plasmon resonance sensors [90] and lateral flow immunoassays [45] but not, to our knowledge, for any acoustic wave sensors.

We wish to optimize our pseudo-array to maximize its performance, but performance may be measured by different objectives, including maximizing the eigenvalue shifts resulting from binding, either of individual targets or of the sum total shifts; maximizing the sensitivity in specific modes or a weighted average of the sensitivity of multiple modes; or maximizing

discrimination. Each of these objectives presents difficulties and raises questions about the limitations of the single-sensor pseudo-array. We will primarily consider cases where we aim to maximize the sum of the eigenvalue shifts we observe, where we aim to maximize the minimum of the eigenvalue shifts, and where we aim to maximize the probability of discrimination between signals when we are not measuring enough modes to guarantee such discrimination. In the first case we wish to maximize the probability that we will see a signal shift from all of our target analytes. In the second, we wish to equalize contributions from multiple analytes, ensuring that shifts from smaller analytes will make significant contributions to the overall signal. In the third, by carefully choosing the positions of functionalized regions, we can (sometimes) make our system act as if it were giving us information from more modes than it actually was.

4.1.1 Critical assumptions

For all optimization in this paper, we assume the following: ease of measurement is not dependent on absolute eigenvalue shifts, provided the shifts themselves are sufficiently large to discount the possibility that they arise solely from nonspecific binding or electronic noise. In other words, if we are certain that the eigenvalue shifts are due to specific binding by our desired targets, we do not have to consider the magnitude of any *absolute* eigenvalue shifts, but instead can consider *relative* eigenvalue shifts, normalized with respect to the unperturbed eigenvalue for each mode. This approach allows us to weight information received from each mode equally, decoupling the effect a particular mode has from the magnitude of its eigenvalue. Alternatively, we can consider the *fractional* shift, wherein we compare the shift induced by a patch at a particular position to the maximum shift it could induce, taken over all possible positions on the membrane. The fractional shift is therefore a measure of how good a position is, independent of the specific mass-loading. We will sometimes refer to objective functions in which we consider fractional shifts as *normalized* objective functions.

This assumption is valid provided that noise in the system will remain below some constant

level significantly below the expected level of meaningful signal. We therefore expect this assumption to hold for lower order modes, including the fundamental mode, for which the expected eigenvalue shifts are large compared to the noise levels observed [5].

We further assume that if a target is present in solution, it will bind irreversibly, and binding will not be mass-transport-limited. In practice, we expect that the quantity of bound mass and the quantity of target mass in the bulk phase will be related by an equilibrium isotherm such as the Langmuir or Hill isotherm. These equations will allow us to convert the quantity of bound mass to the concentration of target in solution, except in cases where the concentration is above the level that will saturate the functionalized region; concentrations from saturation level up are indistinguishable from each other. However, the nature of the conversion from bound mass to solution concentration is unimportant in terms of the optimization of the surface functionalization. For simplicity we choose irreversible binding, but note that this choice can easily be revised if necessary.

In all cases, we will assume that if an analyte is present, it will bind either at a specified fraction of its total binding capacity, or within a specified fractional range of the binding capacity. (The total binding capacity is determined by the size of the ligands and ligates and by the area available, as described in Section 3.2.) This is a general form of the problem for determining the presence or quantity of multiple targets; the case in which we have no advance knowledge of the range of fractional binding is included. This assumption is valid for any case where ligands, if present in any amount, will be present above or below certain thresholds.

We assume that any eigenvalue shift arising from non-specific binding will be negligible compared to the noise of the system. The validity of this assumption will depend on the target species, competitor species, and the binding chemistries used, as well as on the electrical and mechanical noise of the sensor. Without experimental evidence, it is difficult to estimate the sizes of the various factors, and so we choose to work with the optimistic and idealized case of no non-specific binding, with the understanding that this parameter will have to be adjusted for specific real-world cases.

For all cases, we assume that the ratio of bound mass on the more heavily loaded patch to the other will be between 1 and 10 for cases where both patches are mass-loaded. We realize that the expected range of bound masses would be considerably broader than one order of magnitude. For example, if two targets were antibodies of similar size with similar dissociation constants, we might expect their serum concentrations to vary by several orders of magnitude [66], leading the bound masses in their turn to differ by several orders of magnitude. However, we chose to restrict ourselves to the cases where the ratio lies between 1 and 10 because where the ratio becomes much larger the more massive patch will tend to dominate, regardless of position, and so covering those cases does not reveal more information about the objective functions. All of our methods can be extended to cover arbitrary ratios of bound mass, which can be useful in running speciic optimization cases. Because the eigenvalue shifts for a particular patch are linear with added mass (for small masses) we need only consider the ratio of masses, not the absolute masses.

Finally, we note that while we could in practice functionalize regions of any shape, we restrict our attention to rectangular patches the sides of which are parallel to the sides of the membrane. This is for mathematical reasons: so we can analytically integrate the results of our perturbation analysis, as discussed in Sec. 3.4. The use of other shapes would require numerical integration, negating the benefits of the perturbation analysis approach.

4.1.2 Objective functions and constraints

We are now in a position to formulate objective functions with attendant constraints. As mentioned above in Sec. 4.1, we propose three general objectives: the maximization of the sum of all induced eigenvalue shifts, the maximization of the minimum of all eigenvalue shifts, and a "signal overlap" discrimination objective. For each of the first two objectives, we will examine the cases of two distinct patches and of one, two, and three modes. We then formulate analogous objectives using fractional eigenvalue shifts, relative eigenvalue shifts, and combinations of the two, to further reveal the dynamics of the system. We then examine the signal overlap objective function, which allows discrimination even in underdetermined

systems.

The placement of functionalized regions is constrained by the geometry of our system. Functionalized patches may not overlap, and due to imprecision in placement, we enforce a minimum separation distance between patches, defined as the distance between corresponding corners of two patches. In addition, our formulation requires that all individual eigenvalue shifts must exceed a threshold, τ, to minimize the effects of noise [1].

The general form of our optimization problem is then as follows:

$$\begin{aligned} \max & f(x_1, y_1, x_2, y_2) \\ & \text{s.t. } x_1, x_2 \in [0, a] \\ & \quad y_1, y_2 \in [0, b] \\ & d((x_1, y_1), (x_2, y_2)) \geq \delta \\ & \quad \Delta\lambda_{i_{(j,k)}} \geq \tau \; \forall \; i, (j, k) \end{aligned} \tag{4.1}$$

where f is the objective function, (x_i, y_i) denotes the lower left corner of the ith patch, and the membrane dimensions are $a \times b$. The function d is the Euclidean distance metric, δ is the minimum separation distance between patches, and $\Delta\lambda_{i_{j,k}}$, a function of x_i, y_i, and the mass loading coefficient, is the eigenvalue shift from patch i in mode (j, k).

As is evident from this formulation, optimization in a particular scenario may involve two types of tradeoffs. Because functionalized regions must remain separated by a minimum distance δ, there may be cases where improving the position of one region comes at the cost of worsening the position of another. The second type of tradeoff is between modes: mode shapes vary significantly between modes; in fact, an anti-node for one mode may lie on the nodal line of another, such as is the case with the (1,1) and (2,1) modes, respectively. This means that improving the position of a region with respect to one mode may worsen it with respect to another. These two types of tradeoffs, occurring in combination, can lead

[1] In our simulations, for simplicity, we set $\tau = 0$, but recognize the importance in real-world cases of setting an appropriate value for the threshold constraint.

to interesting behaviors on the parts of the objective functions.

We have now introduced three general types of objective functions: a max-sum type where we maximize the sum of all such signal shifts; a max-min type where we maximize the minimum of all signal shifts arising from the combinations of patches and measured modes; and a signal overlap type, designed for underdetermined cases. Now we will go over each of these objective functions and their variants in detail, discussing why and when we want to use each, how our different normalization schemes affect them, and what they can tell us about our sensor's behavior.

4.2 Use of optimization

Let us first examine several possible scenarios we might encounter in the use of our sensor, and consider how to approach them. These hypothetical cases are not intended to cover the full range of scenarios we expect our sensor to address, but rather to motivate the discussion of the obective functions we have developed and demonstrate for each a straightforward application.

Case 0 We assume we can operate in only the primary (1,1) mode. We wish to detect a single target, and will use only a single functionalized region to do so.

Case 1 We assume we can operate in only the primary mode. We aim to detect a single target, and, for redundacy, will do so using two functionalized regions.

Case 2 Again operating in only the primary mode, we need to determine whether a sample contains either of two targets: one of high molecular weight, one of low molecular weight. We can devote one functionalized region to each target.

Case 3 We need to robustly detect a single target analyte with very high confidence. To minimize both false positives and false negatives, we will operate in both the primary and the secondary (2,1) modes, and use two distinct binding chemistries.

Case 4 Now considering only the primary mode, we wish to detect the presence and, if possible, quantities of two different target species.

Case 0 is simplest scenario we will face. With only one functionalized region, and only one operating mode, there are no tradeoffs to consider. The only candidate for optimization is the eigenvalue shift resulting from target binding. Because the signal-to-noise ratio increases for larger signal shifts, the larger can make our eigenvalue shift, the more accurately we can determine the quantity of target bound to the surface. So we maximize this quantity. To do so, we center the functionalized region over the anti-node for the operating mode.

When we consider **Case 1**, we find ourselves in an analogous situation. Although we have two functionalized regions, we do not need to worry about tradeoffs between them. Because they are functionalized for the same target they are not competing with each other; we can conceive of them as a single region functionalized for a single target. We then find that once again our only candidate for optimization is the eigenvalue shift resulting from target binding. To maximize this shift, we maximize the sum of the two component shifts, whose positions we allow to vary independently. While we cannot determine the optimal placement of the regions as easily as in the previous case, the same principle of placing the regions close to the anti-node applies, and in fact generalizes to the same problem with n functionalized regions used for redundancy.

Case 2 appears very similar; again we are operating only in the primary mode, using two functionalized regions. Instead of being used for redundancy, however, they are functionalized for two different targets, and this is a critical difference. Now the two functionalized regions are in competition for the optimal position: if one of them were to move closer to the anti-node, thus forcing the other, via the minimum separation distance constraint, to move further away, then the maximum eigenvalue shift acheivable by the latter region is decreased. This means that for this case we can no longer use our max-sum objective function, as there are straightforward instances in which its use would lead to placement schemes that would yield false negatives for samples for which we were capable of detecting the targets. In other words, the use of the max-sum objective function, in this case, would worsen the performance of our sensor.

To see why, recall that we are examining a sample in which we expect to find none, one, or both of two target species: one of a higher molecular weight than the other. If we were to use the max-sum objective function to determine the placement of our functionalized regions, it would place the region for the heavier of the species on the anti-node, where it can contribute most; the region for the lighter species would be placed at whichever of the set of points the minimum distance away gave the largest eigenvalue shift.

Now suppose that the heavier target is not at all present in the sample, but the lighter target is, in very low quantity. So low, in fact, that the eigenvalue shift induced by its binding is just below the threshold τ required to distinguish signal from noise. If the functionalized region for the lighter patch had been assigned a more advantageous position, the target bound to it would have resulted in a larger eigenvalue shift, avoiding a false negative. The key point here is that this result is asymmetric: the region for the heaver target was favored because it already produced a larger eigenvalue shift. Inequalities between regions can become amplified using the max-sum formulation, where we would prefer that they be minimized instead. To achieve this, we use a max-min formulation: we maximize the minimum of all the separate eigenvalue shifts. This objective function ensures that all of the functionalized regions perform well, even though it may penalize some naturally favored regions to gain such an assurance. The risk here is that a small increase in the objective function could be purchased at a high cost to the other regions; however, as we shall see in the next chapter, for cases with few patches and in the lower harmonics, this does not become an issue.

Case 3 appears very similar to **Case 1**, in that we have two regions functionalized for the same target. The key difference seems to be that we are now operating in two modes instead of only one. However, because we are using our two regions orthogonally, to minimize the probability of false positives or negatives, we cannot consider the two regions to be operating as a single region, as we did in **Case 1**; instead they are competing, as in **Case 2**. This is the case not only in the (1,1) mode, but in the (2,1) mode, as well, and so the dynamics become more complicated: a shift in position may bring an improvement with respect to

the primary mode at the cost of a worsening with respect to the secondary mode, or vice versa. In order to robustly detect a single target, even with multiple forms of redundancy built in, it is important to choose an appropriate objective function.

The final scenario to consider is **Case 4**, where again a small difference from the previous cases proves crucial. We are operating in the (1,1) mode. Faced with a sample possibly containing targets A and B we want to know whether the sample contains target A, target B, both, or neither. Further, if it contains only one of the targets, can we tell in what quantity or concentration? Our desire to distinguish the signal shifts resulting from the different target species is what sets this problem apart from the previous ones. The system of equations resulting from Eq. 3.19 for this case would consist of only a single equation (arising from the output of the single operating mode) having two unknowns (the quantities of mass bound to each of the functionalized regions), and so it seems that we are faced with an intractable problem. However, if we are fortunate enough to have *a priori* information about the likely quantities of each target, we can generate a second equation based on the positions of the functionalized regions. In the best case, this will allow us to always distinguish the contributions from targets A and B; in the worst, when we have no advance information, we will have no such ability. This limits the applicability of this objective function but not its usefulness when applicable.

The principle underlying this objective function is as follows: suppose we know that target A, if present, will be present in a specified fractional range of its total binding capacity, say in a concentration between A_{min} and A_{max}. Likewise, target B's concentration will either be 0 or within the range B_{min} to B_{max}. This means that the eigenvalue shifts produced by each target will also lie within a range, and, crucially, we can adjust the size and position of these eigenvalue ranges by moving the functionalized regions around the surface of the membrane. To demonstrate this, we restrict our eigenvalue shift function to patch positions on a single line from the membrane edge to the (1,1) anti-node, as shown in Fig. 4.1, and plot the eigenvalue shifts induced by different masses as their positions along this line change (see Fig. 4.2). We now consider three possible ranges in eigenvalue

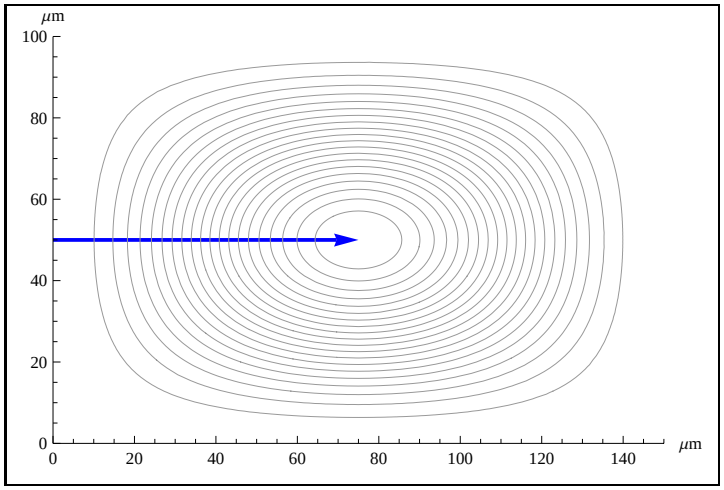


Figure 4.1: Shift measure contours for (1,1) mode overlaid with line along which values in Fig. 4.2 are taken.

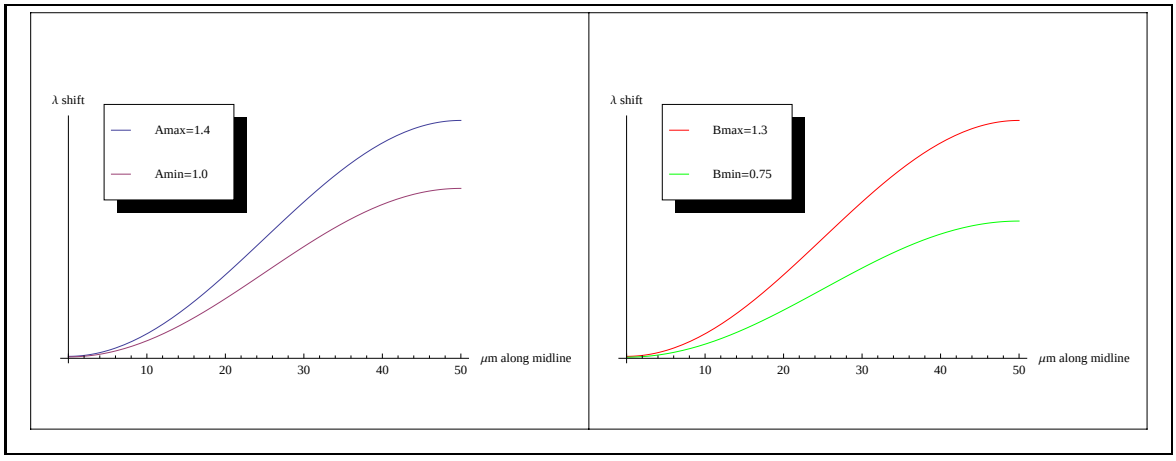


Figure 4.2: Eigenvalue ranges and range sizes change with membrane position, for two different mass concentration ranges A and B. The range at any point along the x-axis is given by the difference between the curves; by moving along the x-axis we can change both the size and position of the eigenvalue range.

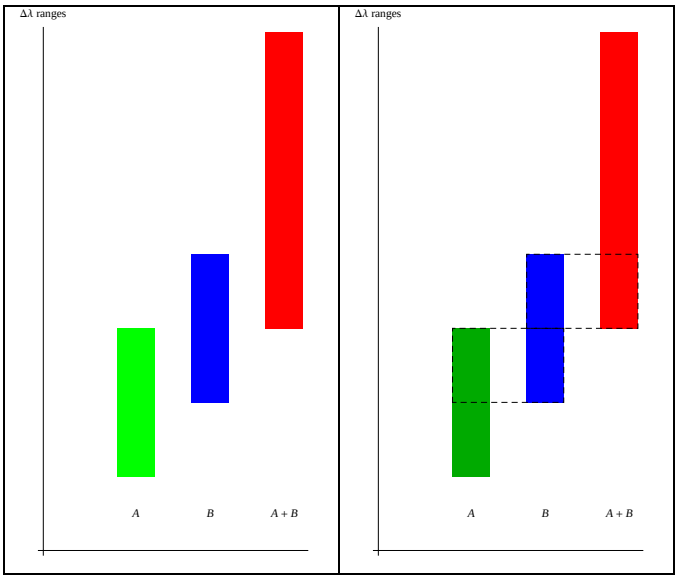


Figure 4.3: The eigenvalue ranges associated with different targets may overlap. In this example, part of the range for target A overlaps with part of the range for target B, and a different section of the range of target B overlaps with the summed output from targets A and B. Values in the dotted regions therefore cannot be conclusively assigned to a single case.

space. We have the ranges associated with targets A and B, and the third: generated by all possible combinations of the two (see Fig. 4.3) We can easily discern if our sample contains neither of the targets; to distinguish the remaining three cases (A, B, A and B) we wish to minimize the overlap between these three ranges. If there is no such overlap, we have complete discrimination. In some cases this will not be possible, and we will have to settle for the maximum discrimination possible under our constraints.

4.3 Maximizing the sum of all signal shifts

The simplest form of our first objective function f (which we maximize) is:

$$f = \sum_{i,(j,k)} \Delta\lambda_{i_{(j,k)}}, \tag{4.2}$$

where pairs (j, k) denote operating modes and i indexes functionalized regions. Each mode/region pair will contribute one summand, so if we functionalize two regions and operate in three modes we will have six total summands. Because of this, we should not be surprised if some of the individual summands are very small, especially when we consider that the anti-nodes for some modes coincide with the nodes or nodal lines for other modes. See, for example, Fig. 4.4, showing that the anti-node for the (1,1) mode, as well as one of the anti-nodes for the (3,1) mode, lie on the nodal line for the (2,1) mode (given by the line $x = 75$ in our canonical example). A position for a functionalized region may therefore result in a large contribution to our objective function from one of its associated summands, and in a small contribution from another summand.

Another factor affecting the sizes of contributions from different modes is that the resonant frequencies (and hence eigenvalues) of the modes differ significantly, as we saw in Subsec. 4.1.1. The shifts from eigenvalues of larger magnitudes will tend to be larger and therefore to contribute more to the objective function. To counteract this, we can divide each eigen-

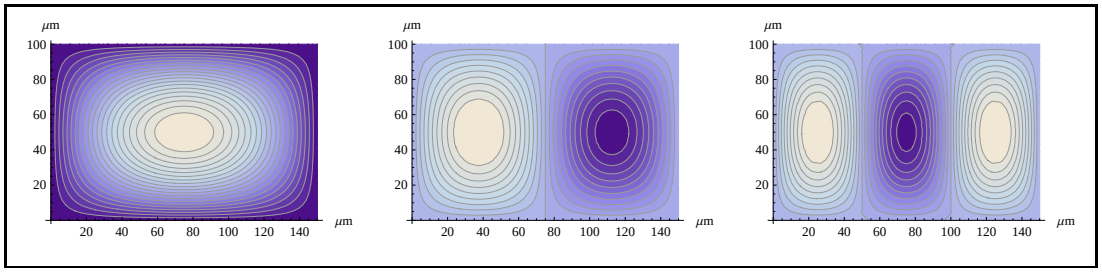


Figure 4.4: Contour plots showing the mode shapes of the (1,1), (2,1), and (3,1) modes.

value shift by the unperturbed eigenvalue of the associated mode.[2] Summing these *relative* eigenvalue shifts gives the information we receive from each mode equal value:

$$f = \sum_{i,(j,k)} \frac{\Delta\lambda_{i_{(j,k)}}}{\lambda_{i_0}}. \tag{4.3}$$

Finally, we note that we may also want to weight the summands relative to maximum quantity they could achieve in the absence of all constraints. If we let $\Delta\lambda^{max}_{i_{(j,k)}}$ denote the maximum possible value taken by $\Delta\lambda_{i_{(j,k)}}$ over all possible pairs (x_i, y_i), then we can add a third formulation of our objective function, the *fractional* formulation:

$$f = \sum_{i,(j,k)} \frac{\Delta\lambda_{i_{(j,k)}}}{\Delta\lambda^{max}_{i_{(j,k)}}}. \tag{4.4}$$

This formulation "pushes" each summand $\lambda_{i_{(j,k)}}$ towards its maximum, but since each functionalized region is associated with multiple summands (one for each mode), each individual region may be "pushed" in multiple directions simultaneously. The addition of the separation distance constraint will further contribute to the complexity of this system.

[2]If we wanted to give custom weights to each mode, for example, reflecting the reliability of the sensor's operation in different modes, we could do so by assigning the weights to each eigenvalue shift in this formulation. An example of such a case, where we assign custom weights to individual summands, is discussed in 5.

4.4 Maximizing the minimum signal shift

The forms of this objective function are analogous to those of the previous. The first formulation considers neither the relative eigenvalue shifts nor the fractional eigenvalue shifts; with this formulation contributions from the $(1,1)$ mode are expected to dominate:

$$f = \min_{i,(j,k)} \Delta\lambda_{i_{(j,k)}}. \tag{4.5}$$

With this objective function, however, the discrepancy between the sizes of the shifts is even more of an issue. When operating in more than one mode, the smallest shift may be negligible in comparison to one or several of the others, even when maximized, so optimizing that shift will do little to optimize the majority of the shifts, because the minimum shift will never pressure the others from beneath. [3] The relative formulation therefore becomes important when operating in multiple modes:

$$f = \min_{i,(j,k)} \frac{\Delta\lambda_{i_{(j,k)}}}{\lambda_{i_0}}. \tag{4.6}$$

The fractional formulation is as expected and requires no special explanation:

$$f = \min_{i,(j,k)} \frac{\Delta\lambda_{i_{(j,k)}}}{\Delta\lambda^{max}_{i_{(j,k)}}}. \tag{4.7}$$

4.5 Minimizing signal overlap

In the cases above, we have always required that the number of modes be equal or greater than the number of functionalized regions, as stated in Sec. 3.3; along with a small constraint on symmetry, this enables us to translate the device output in frequency form (or equivalently in eigenvalue form) to the quantity of mass bound to each functionalized region. This objective function is designed to allow us that same discrimination in the case

[3] This pressure is applied by the minimum shift, under maximization, exceeding another shift in value; the latter shift then becomes the minimum and is forced upward in value as well. The "floor" of the system is thus raised.

where we have one operating mode, but two functionalized regions. While this approach could potentially be used for more complicated scenarios, it is difficult to generalize and we limit ourselves here to the simplest case.

As in Sec. 4.2, we have two targets, A and B, each of which is either not present or present in the concentration range $[A_{min}, A_{max}]$ or $[B_{min}, B_{max}]$ respectively. For a given functionalized region placement (x_i, y_i) for one of the targets, say, A, the concentration range would correspond to a range in the possible resulting eigenvalue shifts. For simplicity, let $[S_{Amin}, S_{Amax}]$ be the eigenvalue shift range associated with target A, with the ranges for B and for the shift with both A and B present similarly defined. (We will have a third eigenvalue shift range, the sum of the other two, denoted by $[S_{ABmin}, S_{ABmax}]$.) As shown in Fig. 4.2 above, changing the choice of the functionalized region position not only changes the magnitude of the eigenvalue shifts, but the width of the resulting range. This means that each individual target will have an associated range, which can be stretched and translated by changing the position of the target's functionalized region. However, instead of seeing the output from each target individually, we will only see a single output - the total eigenvalue shift. Given this single point, and the ranges associated with each target, what can we say about the sample?

Obviously, if neither target is present our job is simple. Otherwise it must fall into at least one of the following three ranges: that associated with target A ($[S_{Amin}, S_{Amax}]$,), that associated with target B ($[S_{Bmin}, S_{Bmax}]$), or the eigenvalue shift range resulting from both targets A and B binding ($[S_{ABmin}, S_{ABmax}]$). If it falls into only one of these, then that range indicates which targets are present in the sample. However, if the point lies in multiple overlapping ranges, we have no way of determining what combination of target binding gave rise to the eigenvalue shift. To maximize discrimination, therefore, we have to minimize the regions where these ranges overlap; we do so by moving the functionalized regions around, thereby stretching and translating the eigenvalue ranges.

To determine the signal overlap we examine each of the three possible pairs of ranges, measure the overlap in each pair, and sum up. For an arbitrary pair of ranges $[P_{lower}, P_{upper}]$,

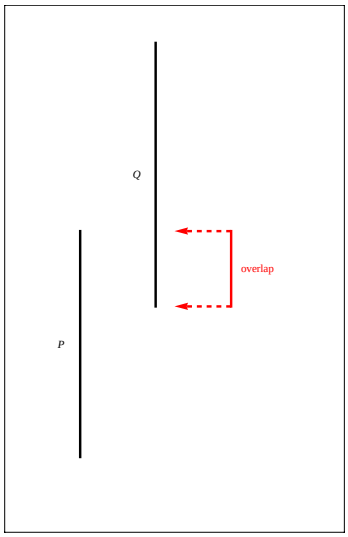


Figure 4.5: Overlap is given by a function of the upper and lower limits of the two ranges.

$[Q_{lower}, Q_{upper}]$, the overlap between them is given by $|\min P_{upper}, Q_{upper} - \max P_{lower}, Q_{lower}|$ (see Fig. 4.5). We can now define our objective function, which is to be minimized:

$$\begin{aligned} f = & |\min(S_{Amax}, S_{Bmax}) - \max(S_{Amin}, S_{Bmin})| \\ & + |\min(S_{Amax}, S_{ABmax}) - \max(S_{Amin}, S_{ABmin})| \\ & + |\min(S_{ABmax}, S_{Bmax}) - \max(S_{ABmin}, S_{Bmin})| \end{aligned} \tag{4.8}$$

Note that this function "triple-counts" some overlaps, so that if the eigenvalue ranges for target A, target B, and the total shift all include a particular value, that is counted in the objective function for each pair of eigenvalue ranges. This is a feature, not a bug: the triple overlap is especially detrimental to discrimination; by counting each individual overlap separately we are penalizing the triple overlap more heavily and helping to drive the solution to a less ambiguous state.

Another feature is that this function calculates not only the signal overlap, but the separation between signal ranges as well. This separation indicates that one or another of the ranges could be larger than it is, without sacrificing discrimination. Minimizing the signal separation in addition to the signal overlap will help to ensure that in our optimization we do not drive the solution to the opposite but equally unpalatable situation where one

eigenvalue shift range is vanishingly small. [4] To avoid related problems (such as making *all* the eigenvalue shift ranges small and tightly packed) we rely on choosing appropriate running conditions for the genetic algorithms we use to solve the optimization problems.
This objective function can be generalized to the case of multiple modes, but the conditions for discrimination become intricate and we do not consider them here. As we are only working with a single mode, there is no need to consider our eigenvalue shifts relative to the unperturbed eigenvalue or to consider fractional eigenvalue shifts. Therefore our initial formulation, Eq. 4.7, is the sole version of this objective function we will use.

4.6 Constraints

Our objective functions operate in a common environment; they are all minimized or maximized while subject to various constraints. Some of these constraints are imposed by physical law or the physical and geometrical design of our sensor and ancillary components, such as the photolithographs used to pattern the surface; these constraints are immutable, and if they are not implemented in our optimization our results will not reflect what is physically feasible. The second set of constraints are those we choose to impose; the question here is not of actual feasibility but of restricting the search space to eliminate regions which would return undesireable or unacceptable solutions.
Physical, or *hard* constraints, include constraints on where functionalized regions may be placed, how large or small they may be, and what shapes they may take. However, for simplicity we have chosen to use a fixed patch size, and as discussed in Subsec. 4.1.1, we restrict ourselves to rectangular functionalized regions for ease of integration; therefore these two constraints are self-imposed *soft* constraints, stricter versions of their physical counterparts. An additional soft constraint is derived from the precision of the patterning technology used to functionalize regions; to ensure that discrete regions do not overlap,

[4] We also recognize that there can be an advantage to small but non-zero gaps between the signal ranges, as this may enable us to avoid confusion as to which range we are in under noisy conditions; this objective function can easily be modified to require a given separation between ranges.

we prescribe a minimum separation distance between them, as described in Subsec. 4.1.2. We also often want to ensure that all of the signal shifts exceed a tuneable minimum threshold value. In practice, we may also want to enforce a constraint on solutions that are symmetric (or that would, given the noise of the system, be functionally symmetric) in all the considered modes simultaneously, as this makes it impossible to determine which patch contributed which portion of the signal. However, in this thesis we are more interested in elucidating the behavior of the different objective functions, and choose to examine them in the absence of this additional constraint. We will see in Chapter 5 that often the returned solution is symmetric in all eigenmodes, and therefore when we desire this discrimination explicitly including this constraint will be necessary.

The eigenmodes of a membrane of uniform density have a natural order in eigenvalue space, determined by the equation

$$\lambda_{mn} = \pi^2 \frac{T}{\rho}\left(\frac{m^2}{a^2} + \frac{n^2}{b^2}\right) \tag{4.9}$$

(see Sec. 3.1). When we, by manipulating the distribution of mass on the surface of the membrane, induce shifts in the eigenvalues λ_{mn}, we open up the possibility that the order of the eigenmodes will be altered, i.e., that some of the eigenvalues will cross over each other. The risk is that output frequencies could then be mistakenly identified with the wrong eigenmodes, leading to inaccurate system outputs. To avoid this, we need to be able to identify mode crossings when they occur, to impose a constraint on the system prohibiting mode crossings, or to be certain in advance that mode crossings will not occur. Table 4.6 shows the relative sizes of the fundamental eigenvalues for several of the lower order modes, given a nominal eigenvalue of 1 for the (1,1) mode; choosing modes that are more widely separated reduces the chances of mode crossings. However, mode choices are also constrained by measureability. The separation in frequency space of the modes we use, coupled with the projected sizes of the induced signal shifts, indicates that for our system we do not need to worry about mode crossings. Should any of our simulation parameters change or prove inaccurate, we will need to re-evaluate this possibility.

Mode (m,n)	$\frac{\lambda_{mn}}{\lambda_{11}}$
(1,1)	1.0
(1,2)	1.923
(1,3)	3.462
(2,1)	3.077
(2,2)	4.0
(2,3)	5.538
(3,1)	6.538
(3,2)	7.462
(3,3)	9.0

Table 4.1: Relative sizes of the fundamental eigenvalues of the (m, n) modes for each $m \in \{1, 2, 3\}$ and $n \in \{1, 2, 3\}$.

4.7 Optimization methods

4.7.1 Potential solution methods

Our optimization problems are formulated as non-linear programming problems (NLPs) with continuous variables. These are optimization formulations in which either the objective function or the set of constraints contains non-linear parts.

Solution methods for NLPs can be broadly categorized as deterministic, in which a given input will always yield the exact same solution, or stochastic, in which random processes are used, and therefore a given input may produce different results each time it is used. Examples of deterministic methods include gradient-based methods such as steepest descent or Newton's method, and interior-point methods which depend on the linearization of convex functions. Stochastic methods include evolutionary algorithms such as hill-climbing and genetic algorithms, and probabilistic-based methods such as Monte Carlo simulations and simulated annealling.

Typically, deterministic solution methods, such as those mentioned above, explicitly use some underlying properties or dynamics of the system in question. The steepest descent method will move from an initialization point to its final output by always stepping in the direction of the maximally negative gradient for which a step will result in a position inside the feasible region. Interior point methods use the convexity of the objective function to generate a linearized version of it, which is then amenable to solution methods for linear programming problems. In contrast, typical stochastic solvers can treat the objective function as a black box. The hill-climbing method will move from one position in the solution space to the next by comparing the objective function value at its current position to one or several nearby points, and choosing the optimal one among the set. Simulated annealling uses a similar approach, but one where the probability of moving to the optimal point is not always equal to one.

This means that deterministic solution methods are well-suited to problems for which the function's underlying behavior can easily be found, either analytically or computationally, and for problems with few local optima. Genetic algorithms are useful when the system dynamics are not easily accessible, or when a large number of convexities mean that rapidly finding and comparing many feasible solutions is desirable.

In our formulations the signal shift function is the product of squared sinusoidal functions in two dimensions. This means that many of our objective functions, as well as the threshold constraints, are non-convex. Therefore we require solution methods which can handle local optima (sometimes in large numbers) and non-convex feasible regions.

4.7.2 Linear proxy

In Chapter 3 we introduced the concept of eigenvalue distance, a representation of the mapping from a position on the membrane to an eigenvalue shift for a given patch size and mass. We now use the eigenvalue distance function to simplify our optimization problems, ridding them of some of the non-convexities, both in the objective functions and the constraints. The fundamental idea is to reduce our solution search space from all possible

patch positions on the membrane surface to a smaller set of possible patch positions still representing the full range of eigenvalue distances. We do this by choosing a line that intersects all level curves of the system. If we choose this line appropriately we can move from a two-dimensional search space for each patch to a one-dimensional search space. Then, instead of optimizing for a set of patch positions on the membrane, we optimize for a set on the line. After optimization, we use the level curves to generate sets of positions equivalent in eigenvalue space to the given solution.

For this method to work, several conditions must be met. While the line acting as the reduced search space need not, technically, be a straight line, it must be a smooth, continuous, non-self-intersecting curve. For simplicity, in this thesis we work only with straight lines as optimization proxies, although the use of curves may open up the possibility of using such proxies for the optimization of higher modes. As mentioned above, the line must intersect each level curve; if any are neglected a portion of the search space will be left out and the optimization will be flawed. Crucially, this applies in each mode, and, when using the max-sum objective function, to the "effective" modes created by the sums of the component modes; this functionally limits the applicability of this method to single-mode optimization cases for the most part.

Converting the separation distance constraint on our patches from the full membrane geometry to a linear subspace is challenging; two positions that are proximal on the proxy line correspond to many pairs of positions on the membrane, none or many of which may be far apart. We have chosen to use a conservative separation distance constraint, explained below, in our linear proxy solution method. This formulation guarantees satisfaction of the separation distance constraint for all solutions, but is only applicable to two patches and may exclude some feasible solutions. We are mindful that careful consideration should give rise to more refined constraints.

Any straight line meeting the conditions described above will function as a linear proxy. To further simplify the optimization, we require a one-to-one correspondance between points on the line and level curves; for the (1,1) mode, this means the proxy line runs from the

edge to the center (where the anti-node lies), but not beyond. As mentioned above, this constraint restricts the modes to which this method can be applied; in some of the higher modes, there is no straight line fulfilling all the conditions we have defined for the proxy line. Curved paths satisfying these conditions may be constructed, leaving additional modes amenable to this method, but we do not explore this avenue.

To formulate our separation distance constraint, we double the length of the line, as if extending it to the opposite edge. We then sum the width of each patch, the position of each patch on the line (i.e. the distance from the edge to the left side of each patch) and the separation distance - this sum must be less than or equal to the doubled length. If it is, there is a feasible patch positioning with the separation distance constraint enforced, because the patches and separation distance may be laid down on the double line without overlap. [5] This constraint formulation is imperfect, however; a pair of eigenvalue level curves may be sufficiently separated on some proxy lines, and not on others. Therefore, to choose the angle of the proxy line, we consider the slope in eigenvalue space; given the placement of the first patch, the separation distance will determine the placement of the second. If our proxy line is perpendicular to a very steep slope, then the separation distance in physical space will correspond to a large distance in eigenvalue space. If, however, our proxy line is perpendicular to a shallow slope, we will see a much smaller change in eigenvalue shifts. As we see the maximum shift at the anti-node, and want to maximize our eigenvalue shifts, we will want to choose a proxy line following a shallow gradient rather than a steep one so as to minimize the decrease in eigenvalue space given by traveling a fixed distance along the linear proxy. For the (1,1) mode, this corresponds to a line running from one membrane corner to the anti-node (see Fig. 4.6). For ease of visualization, and because the expressions for these lines are simpler (due to separability), in this thesis we restrict ourselves to horizontal and

[5] Note that the patch position in this constraint is determined by the left edge of the patch, while in implementations the position x_1 may be coded to indicate the left side, the center, or the right side of the patch. This explains discrepancies between the form of this constraint here and how it is shown in some pieces of code.

vertical proxy lines. For a real application the optimal, diagonal proxy line could be used; this implementation is equally straightforward but slighter more computationally intensive.

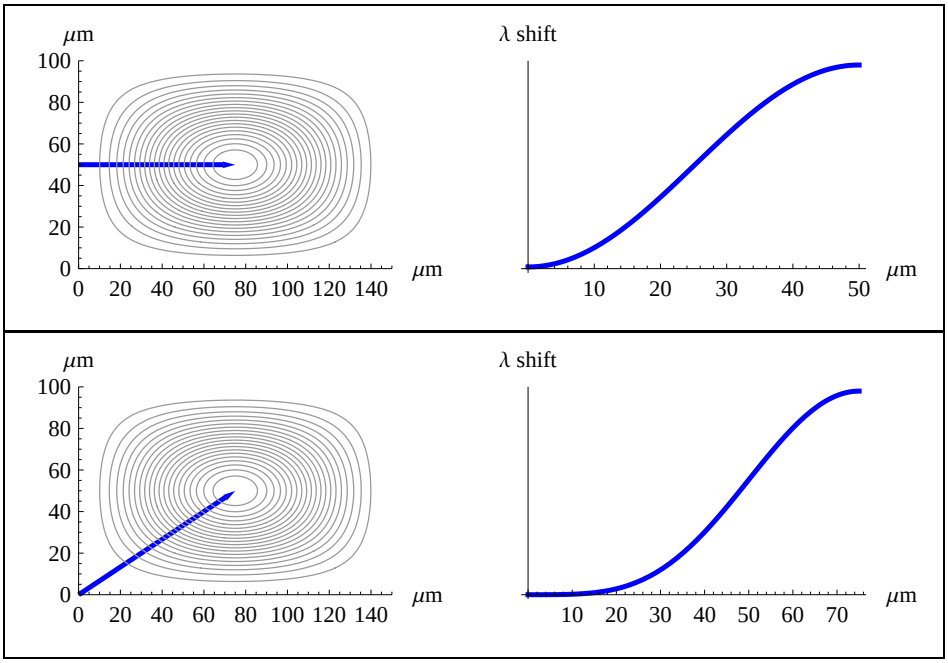


Figure 4.6: Two options for linear proxies.

We perform our linear proxy optimization using Mathematica [57], a software program which allows the use of both symbolic and numerical computation. Mathematica's built-in optimization methods allow us to choose from global and local searches, and from symbolic methods and various numerical methods. Because we are dealing with monotone, smooth, functions the symbolic methods give accurate results.

An example formulation, for a linear proxy taken along the line $y = b/2$, is given below. Here, $f_{lp}(x_1, x_2) := f((x_1, b/2), (x_2, b/2))$.

$$\begin{aligned} &\max f_{lp}(x_1, x_2) \\ &\text{s.t. } x_1, x_2 \in [0, a/2] \\ &x_1 + x_2 + \delta + 2 \times \Delta x \leq a \end{aligned} \tag{4.10}$$

4.7.3 Genetic algorithms

For objective functions which can not be simplified sufficiently to use the linear proxy approach, we use genetic algorithms as our solution method. Genetic algorithms allow us to rapidly find and compare large numbers of potential solutions. Our search space includes multiple convexities, but the stochastic nature of the solution method keeps us from getting stuck in local optima.

Genetic algorithms work by imitating the process of biological evolution. The general framework is as follows: after the generation of an initial population of potential solutions, each member is evaluated for fitness. Lower-fitness members of the population are then eliminated, and a new population is generated by creating combinations of different members of the previous generation, creating variants of individual members, and by directly preserving (some) highest-fitness members. In this way, populations of high fitness are bred.

We use a program called the Genetic Algorithm Toolbox [14], a Matlab [36] toolbox, for all of our optimization work in this area. In this program, each potential solution is represented by a *chromosome*, a binary encoding of the position. The chromosome is the data structure which undergoes mutations to generate new potential solutions. From each chromosome, a *phenotype* is generated; in our case, the phenotype for each potential solution is its position on the membrane. Our convention for positions is that (x, y) denotes the lower-left corner of a patch of preset size. An objective function is applied to each phenotype to evaluate its *fitness*. While the Genetic Algorithm Toolbox is designed to support evaluation of and selection based on multiple objective functions used simultaneously, we have not used this capability.

We use a program called the Genetic Algorithm Toolbox [14], a Matlab [36] toolbox, for all of our optimization work in this area. In this program, each potential solution is represented by a *chromosome*, a binary encoding of the position. The chromosome is the data structure which undergoes mutations to generate new potential solutions. From each chromosome, a *phenotype* is generated; in our case, the phenotype for each potential solution is its position

on the membrane. Our convention for positions is that (x, y) denotes the lower-left corner of a patch of preset size. An objective function is applied to each phenotype to evaluate its *fitness*. While the Genetic Algorithm Toolbox is designed to support evaluation of and selection based on multiple objective functions used simultaneously, we have not used this capability.

For all of our optimization work, we use *stochastic universal sampling* (SUS) to select members of each population for the breeding phase, in which different population members are combined to form members of the new generation. In SUS, potential solutions are mapped to contiguous line segments, where each member's segment is proportional to its fitness. Fitter members of the population therefore occupy a larger fraction of the line. The line is then divided into N_{sel} equal sections, where N_{sel} is the number of members of the population to be chosen. N_{sel} equally spaced points are then chosen, starting from a random point in the first section. The population members associated with these points are the ones selected for breeding. Using this scheme, it is possible for individual members to be selected more than once.

The selected members are bred using a single-point crossover technique. Two parent members are chosen and a random point between 0 and the length of the chromosome is generated. For each parent, the portion of the chromosome before the randomly selected point is the head, and the portion after is the tail. Two offspring are then created by swapping the tails of the chromosomes.

Once all of the offspring have been created, a mutation function is applied. This function randomly switches bits, with the default probability of any bit being switched of 0.007.

Parameter settings, such as the number of individuals per generation, the number of generations per simulation, and the fraction of best members retained after each generation, vary with the individual optimization cases, and will be included in the variable tables for each case.

We test for convergence by tracking the fitness of the best member of each generation over time; when this fitness is unchanged over 100 generations, and over at least 3 simulations, we

judge that the simulations are converged to a steady state solution. The test for convergence relies on the fitness value rather than on the set of patch positions corresponding to the best member, because multiple population members, denoting multiple sets of patch positions may all be equivalent in eigenvalue space (and therefore be equal according to the fitness metric). Since we are testing convergence over multiple simulations, we have to allow for different best members, but use fitness as our metric to ensure that we arrive at equivalent solutions.

Our constraints require that the total signal from each target exceed a certain threshold, and we want to enforce a minimum separation distance between patches. The Genetic Algorithm Toolbox can implement linear constraints directly, but not non-linear constraints[14]. We implement our non-linear signal threshold constraint using a penalty function. This changes our optimization problem from the form shown previously to the following:

$$\begin{aligned} \max f(x_1, y_1, x_2, y_2) - M & \\ \text{s.t. } x_1, x_2 \in [0, a] & \\ y_1, y_2 \in [0, b] & \\ d((x_1, y_1), (x_2, y_2)) \geq \delta & \\ M = \begin{cases} 0 & \text{if } \Delta\lambda_{i_{(j,k)}} \geq \tau, \\ 1000 & \text{if } \Delta\lambda_{i_{(j,k)}} < \tau. \end{cases} & \end{aligned} \tag{4.11}$$

where M is set to 0 if $\Delta\lambda_{i_{(j,k)}} \geq \tau$, and to a large positive number (commonly 1000, as here, but sometimes set to a different value for a specific case) if $\Delta\lambda_{i_{(j,k)}} < \tau$. The number must be large compared to a typical objective function value; we choose a positive number because we are maximizing our function (and *subtracting* the penalty value); if we were minimizing it we would choose a negative one.

4.8 Tradeoffs

In Sec. 4.2, we discussed the issue of patches competing with each other. Two patches can jostle to occupy the same, optimal spot; past a certain point, an improvement in the position of one will force the other to move further away, via the separation distance constraint. A gain for one patch is traded for a loss for another. This is not, however, the only tradeoff we will see in our optimization. Equally important will be tradeoffs between modes: a patch position giving rise to a large signal shift in one mode can give rise to a small shift in another, or a change in position which increases the signal shift from one mode can decrease the shift from another mode. We will see how our choice of objective function and target analyte weights will lead to different types of tradeoffs between patches and between modes.

Chapter 5

Optimization results

5.1 Linear Proxy results

In Section 4.7.2 we describe our linear proxy solution method, whereby we map the the two-dimensional membrane surface to a one-dimensional search space using the eigenvalue distance metric. This method, when applicable, is much faster than using genetic algorithms.

Recall that we define a *relative* eigenvalue shift as one normalized with respect to the unperturbed eigenvalue: $\Delta\lambda_{relative} := \frac{\Delta\lambda}{\lambda_0}$ In contrast, a *fractional* eigenvalue shift is normalized with respect to the eigenvalue shift that patch could achieve were it placed optimally with respect to the anti-node: $\Delta\lambda_{fractional} := \frac{\Delta\lambda}{\Delta\lambda_{max}}$. Each fractional shift value is therefore associated one-to-one with a level curve in eigenvalue space. When we discuss *normalized* objective functions, we mean those which are formulated with fractional eigenvalue shifts. As mentioned in Subsec. 4.1.1, in our analysis we show the ratio of the masses of the functionalized patches captures their behavior even if the absolute masses are not known. In addition, we work with a fixed patch size, which means we can bundle mass and density together. Therefore, in all our simulations, we vary the value for one parameter for patch density, k_1, and fix the other at $k_2 = 1$. Values for k_1 are $0.0, 0.1, 0.2, ..1.0$ (usually; exceptions will be noted). The $k_1 : k_2$ ratio therefore varies from 0 to 1. We refer to the patch

with density k_1 as *patch 1* or *the first patch*, and to the patch with density k_2 analogously. The standard values we use for important variables are as follows:

Membrane length: 150 μm

Membrane height: 100 μm

Patch size: 10 μm by 10 μm

Minimum separation distance[1]: 20 μm

Noise threshold: $\tau = 0$ [2]

Sum of non-normalized signal shifts in the (1,1) mode

We demonstrate our linear proxy method using the objective function resulting from summing the effects of two patches in the (1,1) mode. In this mode, we know that we have only one optimal patch position, centered over the anti-node. As we have two patches of added mass, we expect competition between them for the ideal location.

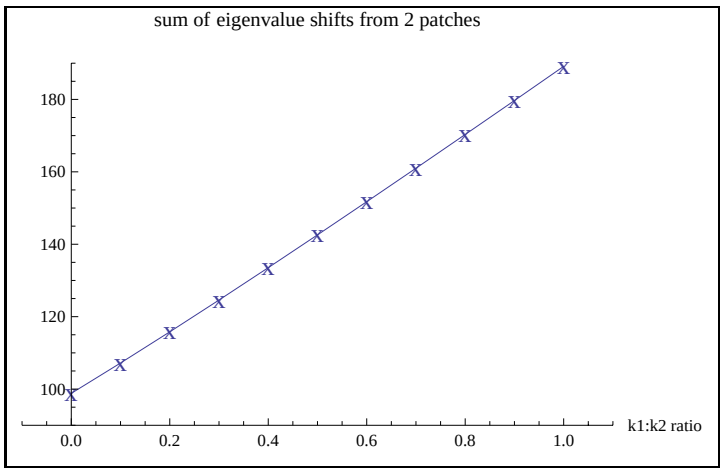


Figure 5.1: Function values plotted against $k_1 : k_2$, sum of non-normalized signal shifts in the (1,1) mode.

In Fig. 5.1 we see the objective function value plotted against the $k_1 : k_2$ ratio. From this we

[1]The minimum required distance between corresponding corners of two patches.

[2]This represents the idealized case of no nonspecific binding.

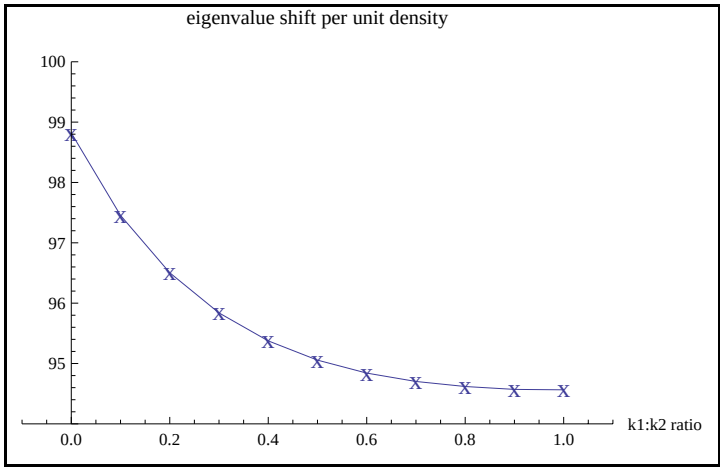


Figure 5.2: Function values per unit mass, sum of non-normalized signal shifts in the (1,1) mode

can see that, as expected, the objective function value increases as the $k_1 : k_2$ approaches one (and simultaneously as the sum $k_1 + k_2$ increases). However, from this figure it is difficult to see any tradeoffs between the two patches. To illuminate these, we consider the objective function value per unit of added mass, i.e. the objective function value divided by the sum $k_1 + k_2$. We again plot this against the mass ratio, resulting in Fig. 5.2. This shows us that as the ratio increases, the contribution to the objective function per unit mass decreases. In Figs. 5.3 and 5.4, which show the positions of each patch on the linear proxy line (x-axis) as well as the objective function value resulting from each patch (y-axis) for three pairs of patches ($(k_1, k_2) = (0.1, 1.0), (0.5, 1.0), (1.0, 1.0)$, we can see why this is so - as the patch masses become equal, the patches are moving to a more symmetric arrangement with respect to the anti-node. These dynamics are explored more fully when we consider the same objective function in Sec. 5.2.

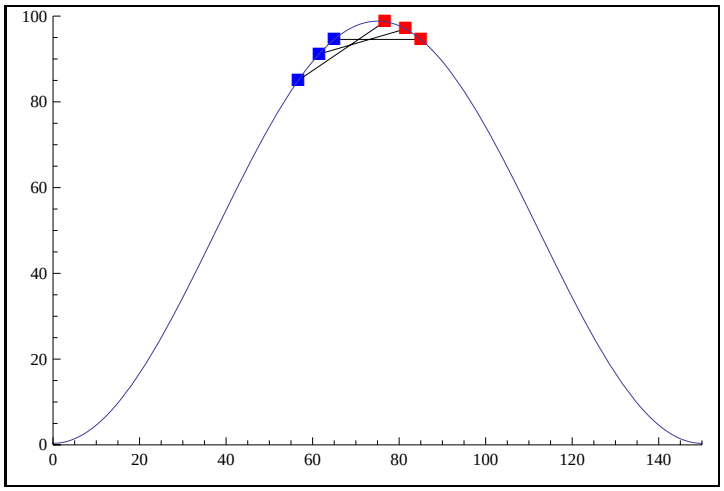


Figure 5.3: Patch positions and objective function values for three pairs of (k_1, k_2) (blue, red) values, superimposed on the curve showing the objective function value for each point on the linear proxy line. Black lines connect the two patches for each (k_1, k_2) pair.

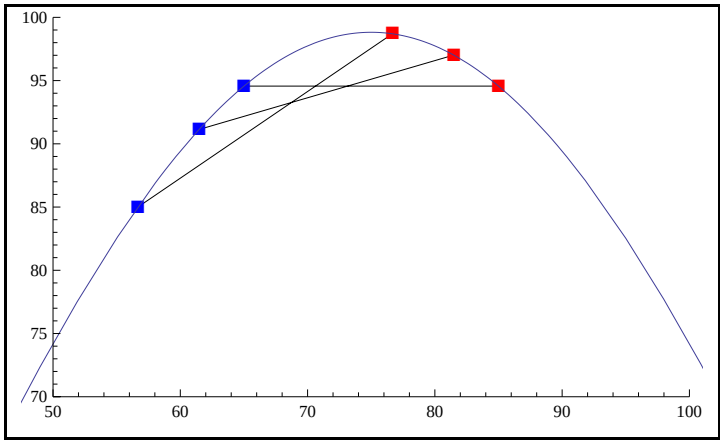


Figure 5.4: Close up of Fig. 5.3. Increasing k_1 corresponds to pairs moving towards the right.

5.2 Genetic Algorithm results

For cases in which the linear proxy is not applicable, i.e. for cases in which a single proxy line cannot intersect all the level curves, we use genetic algorithms. This stochastic method allows us to broadly search the solution space and thereby to consider more complicated optimization cases.

The genetic algorithm results, unless otherwise indicated, were generated using the same membrane simulation parameters as used for the linear proxy results. For parameters specific to the genetic algorithms, such as number of generations, we used the following settings:

Number of individuals per generation: 900

Number of generations: 125

Fraction of best members retained after each generation: 0.2

5.2.1 Summation formulations

Sum of non-normalized signal shifts in the (1,1) mode

In this objective function we consider the (1,1) mode exclusively, and consider two patches of added mass. The purpose of this objective is to maximize the sum of two eigenvalue shifts, one arising from each patch. As this function is based on the summation of two eigenvalue shifts in a mode with only one anti-node, we expect to see tradeoffs between the two patches, even for low (but non-zero) relative patch densities.

If we consider Fig. 5.5 we see that as our mass ratio increases, so does our objective function value. This increase is sub-linear, and decreases as our mass ratio increases. As the mass of patch 1 increases, it is gradually pushing patch 2 further and further off of the anti-node. Fig. 5.6 shows the value of the objective function per unit density. In this figure we can clearly see that the fraction of the maximum that we are able to acheive drops as the two patches become comparable in density. When only patch 2 is present (i.e. $k_1 = 0$), it

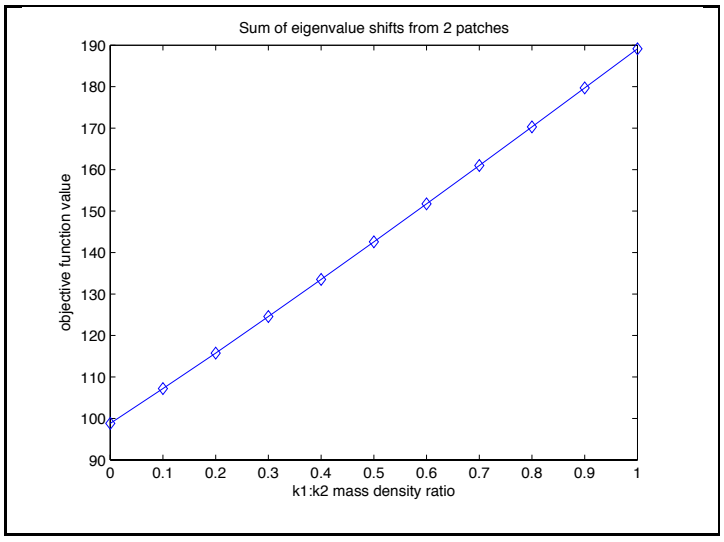


Figure 5.5: Function values plotted against $k_1 : k_2$, sum of non-normalized signal shifts in the (1,1) mode.

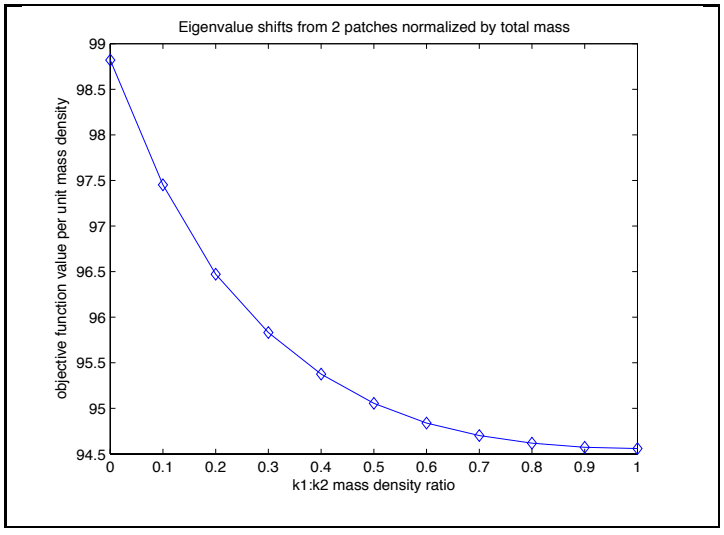


Figure 5.6: Function per unit mass, sum of non-normalized signal shifts in the (1,1) mode.

occupies the optimal position; as patch 1 is added and increases in mass, the fraction of the signal coming from the optimal position is reduced - more and more of the eigenvalue shift is resulting from less well-placed mass. Partly this is due to patch 2 moving to a less advantageous position, and partly to patch 1 contributing an increasing portion of the total signal due to its increasing mass and improving position.

Because our objective function is symmetric with respect to the two patches, we expect to see a "bottoming out" of the decrease as $k_1 -> k_2$; here the slope of the curve must be equal to 1, as adding density to either patch is equivalent.

This symmetric behavior, as well as the competition for the best position, is shown more explicitly in Fig. 5.7, which shows the fractional shifts achieved by each patch. When $k_1 = 0$, the second patch, optimally placed, attains a fractional shift value of 1. Patch 1, being unweighted, is randomly placed and its fractional shift is irrelevant. As k_1 increases, even to only 0.1, it is already affecting the position of patch 2; this means the objective function overall does better when patch 2 is moved to a less-optimal position than before, its loss being offset by the gain from patch 1. This indicates that the magnitude of the gradient of the eigenvalue shift function in the neighborhood of patch 1 (in the direction of patch 1's movement as we increase k_1 from 0 to 0.1) is greater than the magnitude of the gradient in the neighborhood of patch 2 (defined in the direction of patch 2's movement as k_1 is shifted). In fact, because the former gradient includes a factor of k_1 and the latter a factor of k_2, and $k_1 < k_2$, we can see that the difference in the gradients of the fractional shift function is even larger. In essence, we are moving the light patch 1 up a steep slope, and thereby gaining a large value in the objective function, while moving the heavy patch 2 down a shallow slope, losing a lesser amount due to its worsened position. Again, we see that as the mass of patch 1 approaches that of patch 2, we approach a solution that is symmetric in eigenvalue space.

Because there is a one-to-one correspondance between fractional shift values and eigenvalue level curves, with a fractional shift value of 1 corresponding to a position over an anti-node and a value of 0 corresponding to a position over a node or nodal line, we can interpret

Fig. 5.7 as representing the individual level curves on which our patches are sitting. And because we know what our level curves look like in the (1,1) mode (see Fig. 3.5) we can directly see how altering the relative patch masses alters their positions on the membrane.

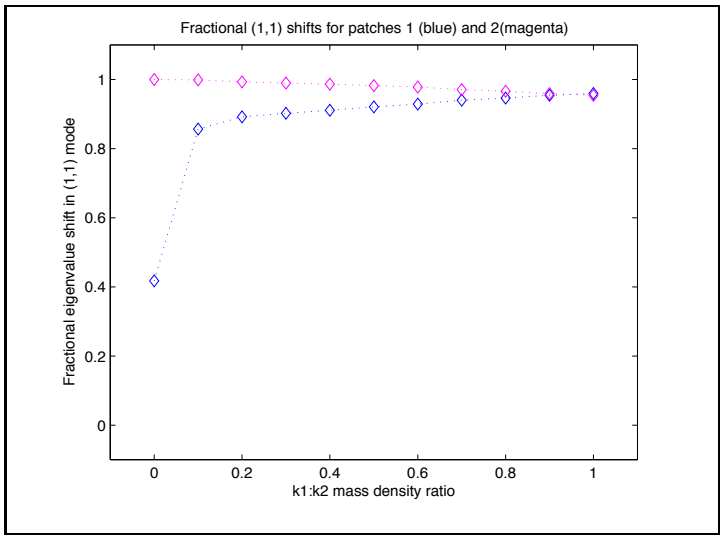


Figure 5.7: Fractional signal strength in (1,1) mode, for sum of non-normalized signal shifts in the (1,1) mode.

These three functions of the ratio $k_1 : k_2$ (the objective function value, the objective function value per unit mass, and the fractional signal strength of each of the objective function summands) will be important for comparing objective functions to one another, for evaluating tradeoffs between modes and patches, and for comparing the relative contributions of different modes and patches.

In most cases we will compare each non-normalized objective function to its normalized counterpart. Normalizing an objective function multiplies the contributions from each mode by a specified constant, different for each mode. Objective functions concerning only one mode are merely scaled when normalized, so it is not necessary to compare the two cases. When multiple modes are involved the functions are not directly scaled, so normalization produces substantively different objective functions.

Because this objective function, the sum of non-normalized signal shifts in one mode, is

a single mode function, we will not consider its normalized version. The next objective function is the sum of non-normalized signal shifts in two modes.

Sum of non-normalized signal shifts in the (1,1) and (2,1) modes

In this objective function we consider the (1,1) and (2,1) modes simultaneously, and look at two patches of added mass. The purpose of this objective is to maximize the sum of four eigenvalue shifts, one arising from each patch/mode pair. As this formulation is a summation of the non-normalized eigenvalue shifts, we expect to see tradeoffs between the two patches even for low (but non-zero) relative patch masses. These tradeoffs are complicated by the facts that we are now considering two modes, and that in the (1,1) mode there is only one anti-node while in the (2,1) mode there are two.

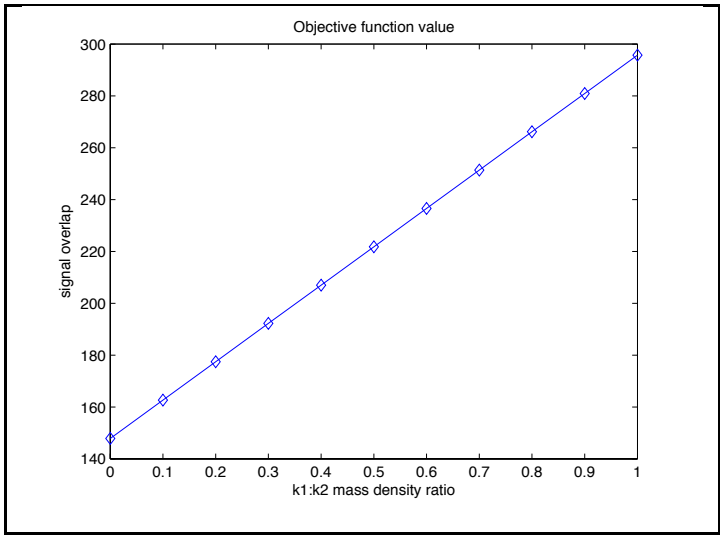


Figure 5.8: Function values plotted against $k_1 : k_2$, sum of non-normalized signal shifts in the (1,1) and (2,1) modes.

However, when we consider Fig. 5.8, which shows the objective function value plotted against the ratio $k_1 : k_2$, we see that the increase in objective function value is linearly proportional to the total mass, indicating that tradeoffs between patches are not important. This is verified in Fig. 5.9, showing the objective function value per unit mass; this value

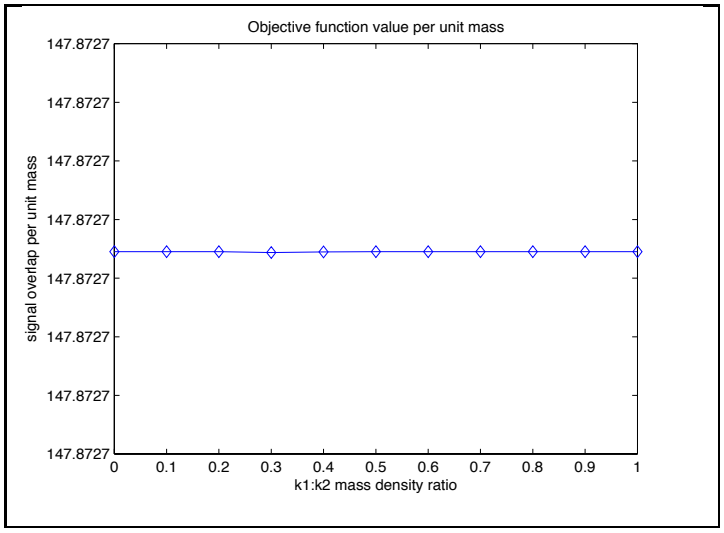


Figure 5.9: Function per unit mass, sum of non-normalized signal shifts in the (1,1) and (2,1) modes.

is constant, which means that changing the ratio of masses does not change the efficiency of the system, implying that the two patches are not changing their eigenvalue positions regardless of their masses. What has happened is that in adding in considerations of the (2,1) mode we have moved from a system in which there is only one optimal position, to one in which there are two. Positioning a patch over the anti-node for the (1,1) mode will result in the single largest eigenvalue shift of the four contributing to this objective function. However, as the anti-node for the (1,1) mode lies over the nodal line for the (2,1) mode, that large contribution will be paired with one of nearly zero from the same patch in the (2,1) mode. In contrast, placing a patch on one of the (2,1) anti-nodes leads to substantial contributions from both modes, making this position better overall. We are not dealing with tradeoffs between the patches, but between the modes. To simplify this system, consider our objective function, written in an expanded form:

$$f = \Delta\lambda 1_{(1,1)} + \Delta\lambda 1_{(2,1)} + \Delta\lambda 2_{(1,1)} + \Delta\lambda 2_{(2,1)} \tag{5.1}$$

where each $\Delta\lambda_{i_{(j,k)}}$ is the integral of the product of a mass term and and a mode shape

function representing the response surface (recall Eq. 3.21). Because the mass terms for each patch are identical across the modes, we can regroup these terms, and consider the sum of the two response surface functions as a single response surface function, affected by two different mass terms, one from each patch. (Recall from Chap. 3 that this function is the result of an *integration* of a mass distribution function and a mode shape function; we convert this to a *product* of the mass distribution function and a response surface function by restricting our mass distribution to rectangular regions, this making our integration seperable and allowing us to generate (by integrating the mode shapes) new response surfaces specific to the mass distribution parameters.)

We can think of this new response surface function as being the response surface of the *effective mode* of the system, and consider its nodes, anti-nodes, and gradients just as we would for the response surface of a single mode. Fig. 5.10 shows the level curves for the response surfaces corresponding to the (1,1) mode, the (2,1) mode, and the combined (1,1)+(2,1) effective mode, for membranes and added patches using the same parameters as in our genetic algorithm calculation.

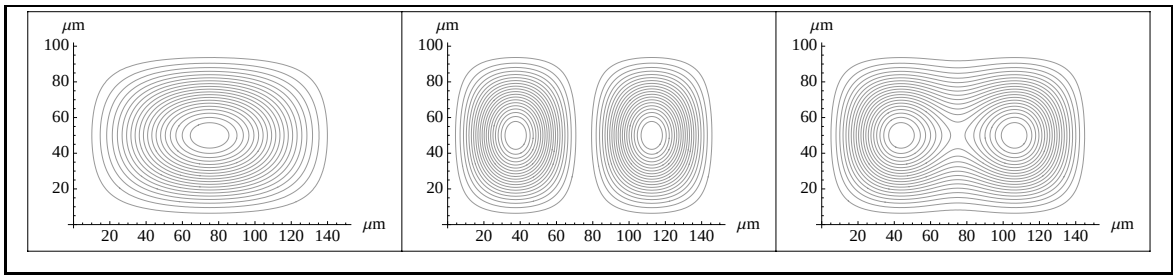


Figure 5.10: Response surfaces for the (1,1) mode, (2,1) mode, and (1,1) + (2,1) effective mode.

From this figure we can see that the effective mode generated by the (1,1) and (2,1) modes has two anti-nodes. As they are separated by more than the minimum separation distance we have assigned, placing one patch on each of them results in a feasible solution. As we are not enforcing discrimination constraints (see Sec. 4.6), we do not penalize a symmetric

solution; therefore this configuration yields the maximum value for this objective function, regardless of the ratio of patch masses. This is why the objective function value rises linearly with total added mass.

An important question is whether we can use characteristics of the objective function (such as the number of modes or patches used) combined with characteristics of the effective mode (such as the number of local maxima) to predict the general behavior under optimization of our objective functions. For example, do all summation functions using the (1,1) and (2,1) modes with 2 patches show the same behavior at the gross level, regardless of normalization or weighting schemes?

Sum of normalized signal shifts in the (1,1) and (2,1) modes

This is the normalized version of the previous objective function, so we are again considering the (1,1) and (2,1) modes, and two patches of added mass. The purpose of this objective is to maximize the sum of all four normalized eigenvalue shifts, one arising from each patch in each mode. The eigenvalue shift from each patch is normalized by the maximum eigenvalue shift possible in the considered mode if each patch were considered in isolation. This maximum possible shift is achieved by centering the patch over the anti-node for the mode in question. As there is only one anti-node in the (1,1) mode, and two patches, we will not be able to acheive this maximum for both patches simultaneously in this mode. Although the corresponding maximum for the sum of the two individual signal shifts is not feasible, it is a highly useful metric; it allows us to compare our results to the naive case where each patch is used on a single membrane in isolation, and therefore to monitor the cost of multiplexing.

In Fig. 5.11 we see that as our mass ratio increases, so does our objective function value. This increase is linear with the total added mass. For $k_1 = 0$, equivalent to a case of only one patch, we see that we achieve the maximum shift possible in the (2,1) mode, but not the (1,1) mode. Because the anti-nodes do not overlap, we cannot achieve the maximum for both modes simultaneously. In Figs. 5.12, 5.13, and 5.14 we observe that for cases where

$k_1 > 0$, i.e. cases where we are truly dealing with two patches, there is no change in the positions or eigenvalue shifts per unit mass as we increase k_1. This is what we saw with the previous objective function; normalization has not changed the results. This is because we are working with the effective mode generated by the (normalized) (1,1) and (2,1) modes; although this effective mode is somewhat different from the one in the previous case because the contributing modes are scaled by different factors [3] The distortion due to the differential scaling is not sufficient to change the relevant properties of the effective mode; it has two anti-nodes separated by more than the minimum separation distance, so the two patches always sit on the anti-nodes.

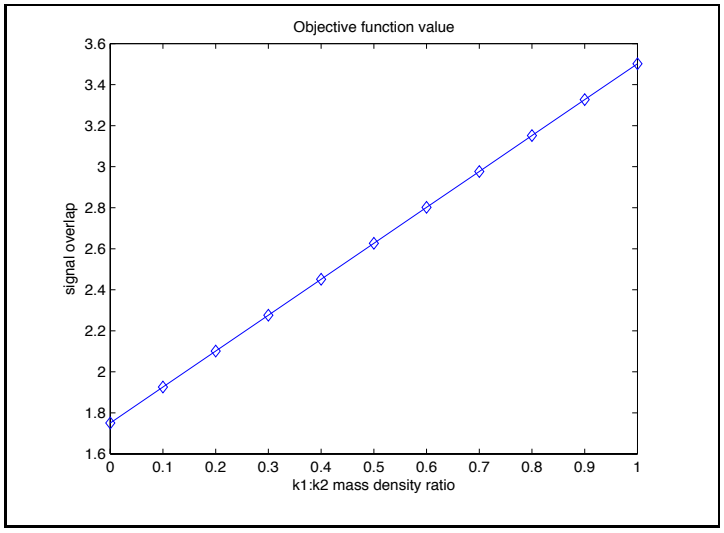


Figure 5.11: Function values plotted against $k_1 : k_2$, sum of normalized signal shifts in the (1,1) and (2,1) modes.

[3] In the non-normalized case, the effective mode is given by $(1,1) + (2,1)$; in the normalized case, by $\frac{(1,1)}{max\Delta\lambda_{11}} + \frac{(2,1)}{max\Delta\lambda_{21}}$.

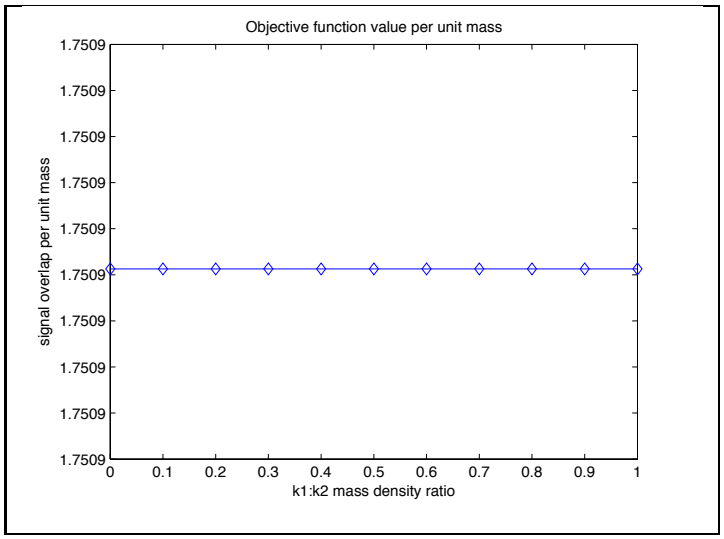


Figure 5.12: Function per unit mass, sum of normalized signal shifts in the (1,1) and (2,1) modes.

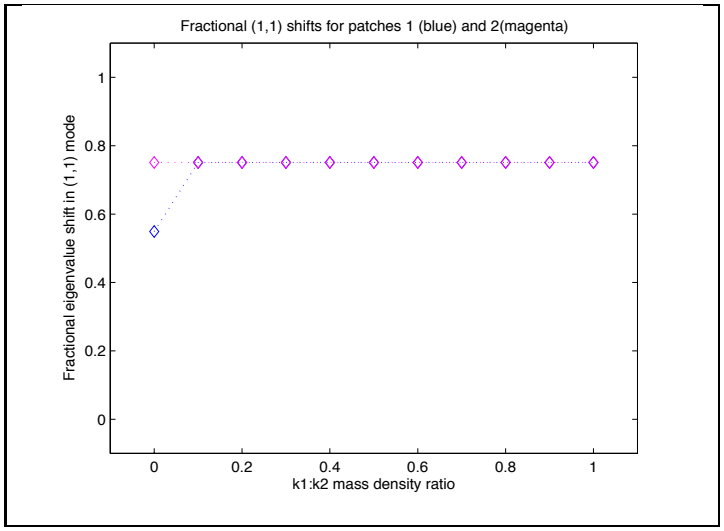


Figure 5.13: Fractional signal strength in (1,1) mode, for sum of normalized signal shifts in the (1,1) and (2,1) modes.

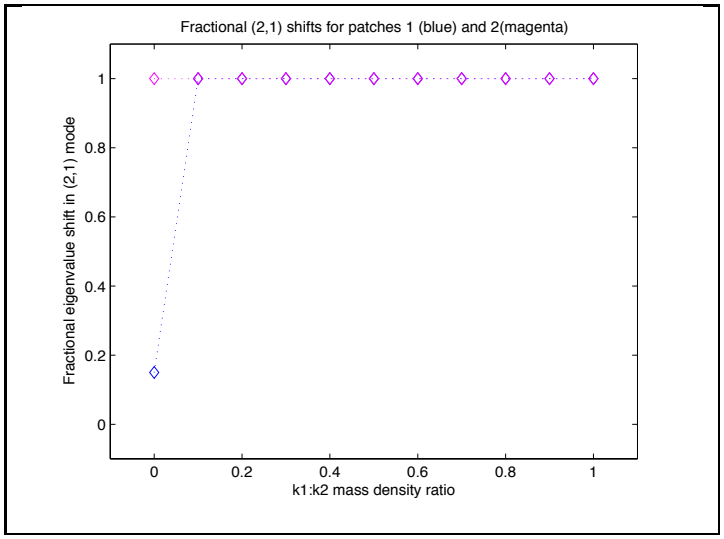


Figure 5.14: Fractional signal strength in (2,1) mode, for sum of normalized signal shifts in the (1,1) and (2,1) modes.

Sum of non-normalized signal shifts in the (1,1), (2,1), and (3,1) modes

As we have seen in the previous two cases, when considering max-sum objective functions in multiple modes, it can be helpful to consider the effective mode created by summing the component modes. In Fig. 5.15 we see the response surfaces of the three component modes of this objective function, as well as that of the effective mode they generate. Of particular note is that one of the anti-nodes for the (3,1) mode coincides with the anti-node for the (1,1) mode and with the nodel line for the (2,1) mode, while the nodal lines for the (3,1) mode lie over the anti-nodes for the (2,1) mode. Therefore we expect to see tradeoffs between modes, as in the previous cases, and that, again, no patch can achieve the maximum in all modes simultaneously. However, the coincident anti-nodes for the (1,1) and (3,1) modes make that position the single best position on the membrane; thus we expect the two patches to compete for this optimal position.

Once again we begin by considering the raw objective function value, shown in Fig. 5.16. The increase in objective function value is approximately linear in added mass, indicating

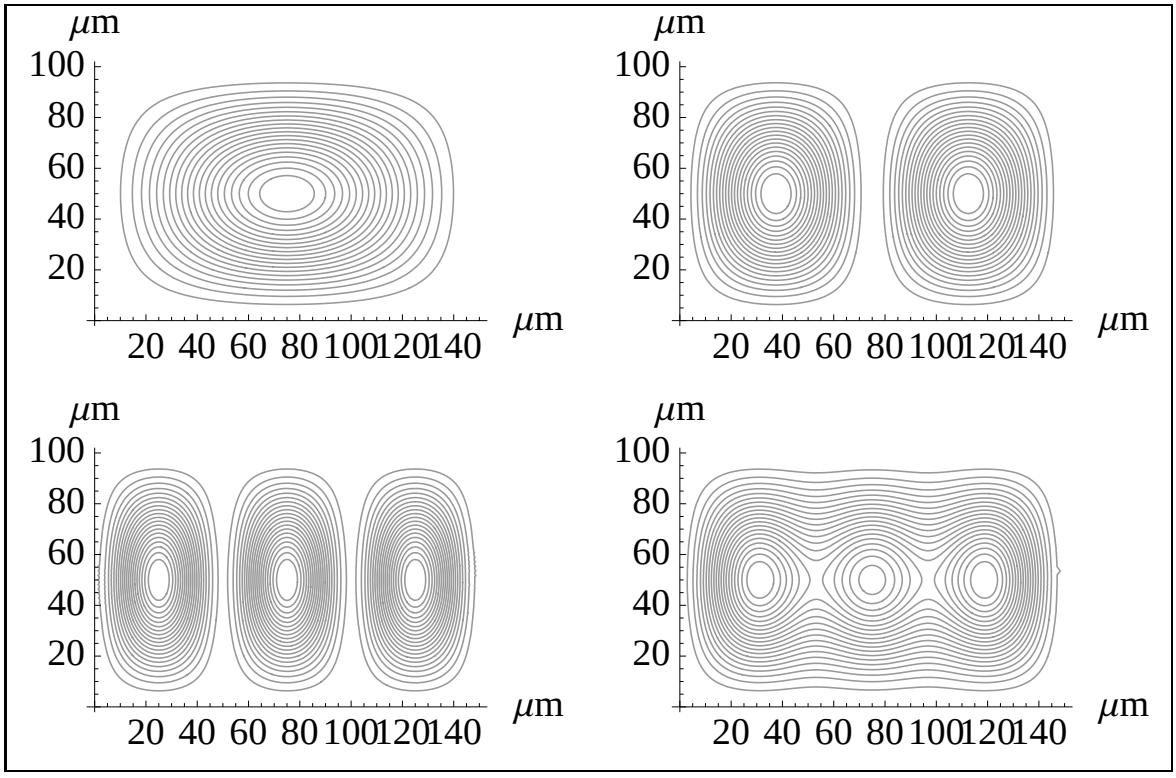


Figure 5.15: Response surfaces for the (1,1), (2,1), and (3,1) modes, and for the (1,1) + (2,1) effective mode.

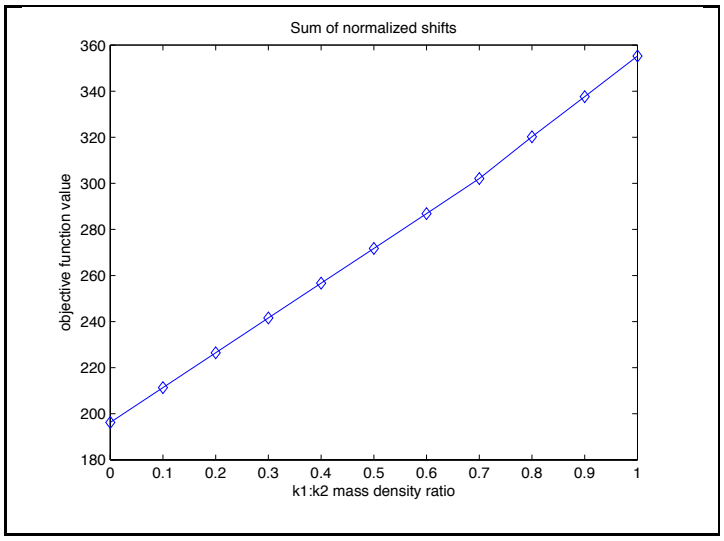


Figure 5.16: Function values plotted against $k_1 : k_2$, sum of non-normalized signal shifts in the (1,1), (2,1), and (3,1) modes.

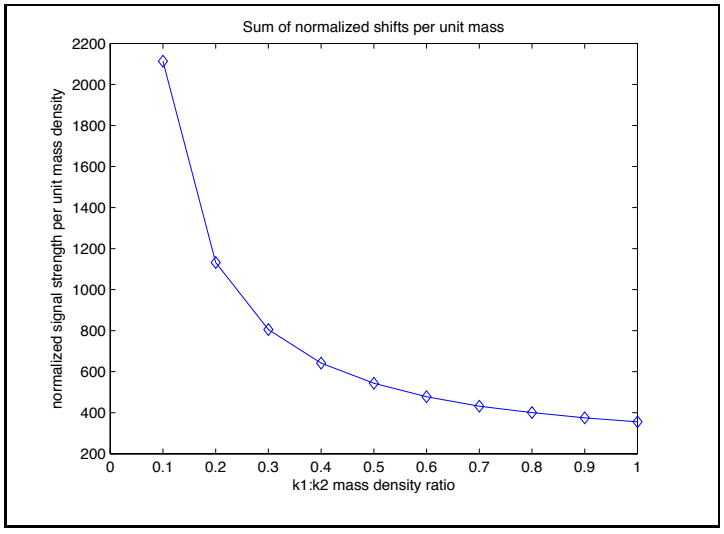


Figure 5.17: Function per unit mass, sum of non-normalized signal shifts in the (1,1), (2,1), and (3,1) modes.

that each unit of mass added to the lighter patch is contributing approximately the same amount to the objective function. However, as we can see in Fig. 5.17, the increase is not linear in total mass. This is due to the shape of the response surface of the effective mode: it has one global maximum. When only one patch contributes to the eigenvalue shifts (i.e. $k_1 = 0$), that patch is placed on the global maximum, and we see a large combined eigenvalue shift per unit mass. However, when patch 1 has non-zero density ($k_1 > 0$), it is placed not on the global maximum, but on a less advantageous position. Patch 2, which is both heavier and optimally places, therefore has a larger effect per unit mass than patch 1, and as the latter increases in mass, the average contribution to the objective function per unit mass decreases.

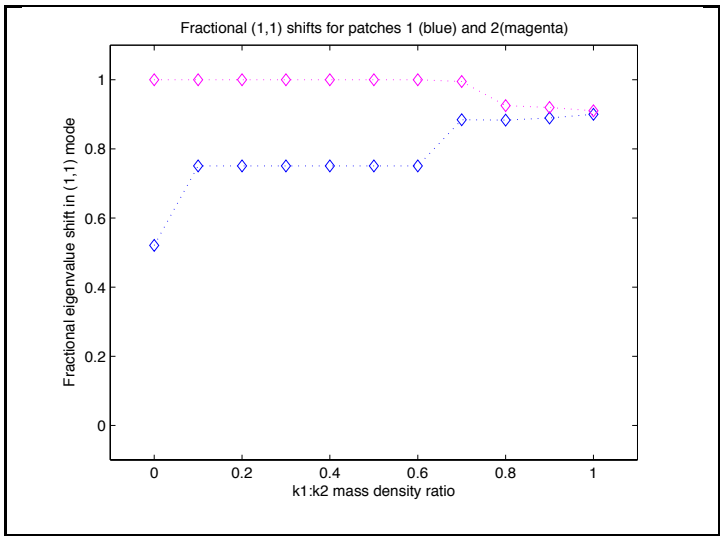


Figure 5.18: Fractional signal strength in (1,1) mode, for sum of non-normalized signal shifts in the (1,1), (2,1), and (3,1) modes.

When we look at the fractional signal shifts achieved in each modes, in Figs. 5.18, 5.19, and 5.20, we see behavior that changes in all three modes as k_1 increases.

For patch 2, the heavier patch, the initial placement yields a fractional signal strength of 1 in both the (1,1) and (3,1) modes, and of 0 in the (2,1) mode, indicating placement on the anti-node of the former modes and on the nodal line of the latter. This is as expected; in

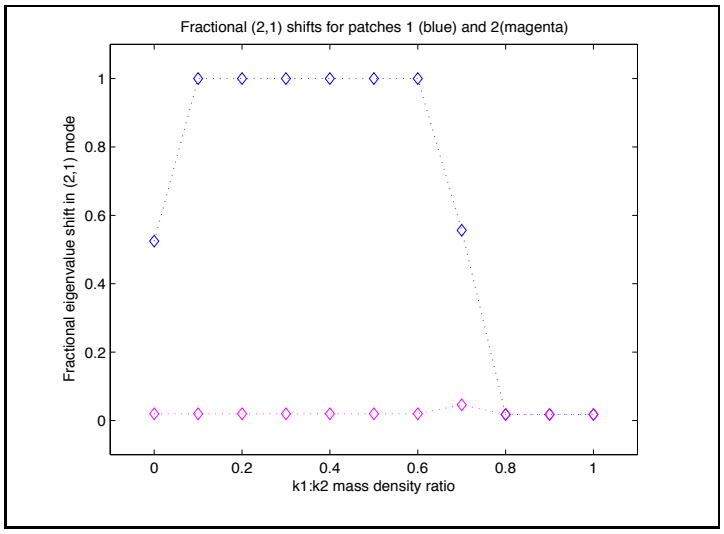


Figure 5.19: Fractional signal strength in (2,1) mode, for sum of non-normalized signal shifts in the (1,1), (2,1), and (3,1) modes.

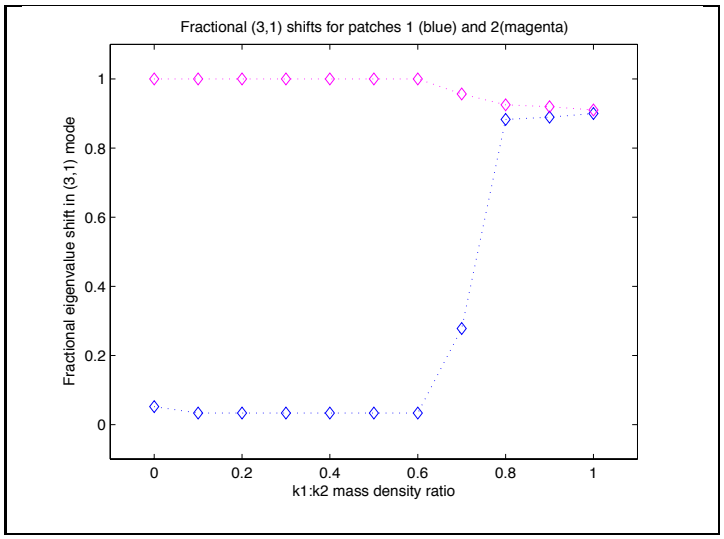


Figure 5.20: Fractional signal strength in (3,1) mode, for sum of non-normalized signal shifts in the (1,1), (2,1), and (3,1) modes.

the absence of a competing, second patch (as patch 1 initially has a mass of 0; its placement is random), patch 2 lies on the maximum for the effective mode.

When we considered a previous case, the sum of non-normalized signal shifts in the (1,1) mode, we saw that even changing the density on patch 1 from $k_1 = 0$ to $k_1 = 0.1$ alters the position of patch 2. This is due to the difference in the slopes of the eigenvalue shift function in the neighborhoods of the patches. In this case, however, we do not see a change in the position of patch 2 until our k_1 value reaches 0.7. This is because now the changes in contribution from patch 1 and patch 2 are each coming from changes in three modes; while in the (1,1) mode the change for patch 1 resulting from a shift towards the optimal position outweighs that of patch 2 as it moves away from the optimum, and in the (2,1) mode the change for patch 2 is positive, the gradient of the fractional shift function in the (3,1) mode is nearly as steep in the neighborhood of patch 2 than in that of patch 1. Therefore the patch density ratio $k_1 : k_2$ must be significantly higher for the gains from patch 1 to offset the losses from patch 2. From this point on, the two patches are competing for the single optimal position; in Figs. 5.21 and 5.22, which show the absolute eigenvalue shifts induced by each patch in each mode, we can see that when $k_1 : k_2 = 1$, the two patches behave equivalently with respect to all three modes. Further, we note that for $k_1 \geq 0.8$, the (2,1) mode does not contribute significantly for either patch.

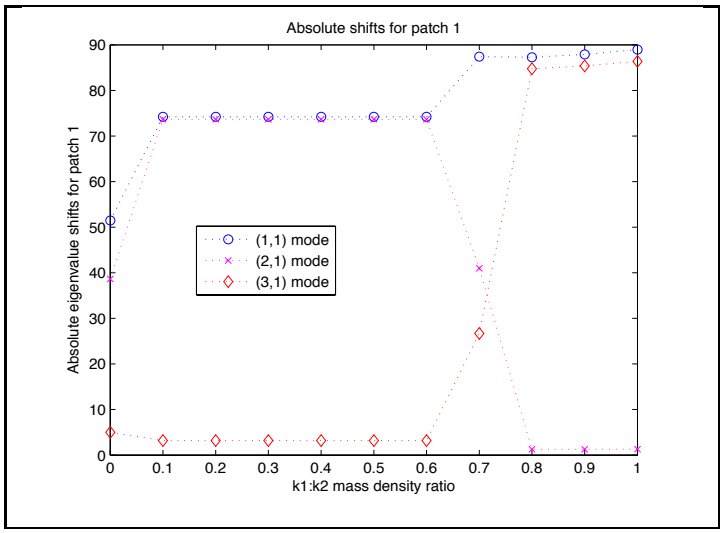


Figure 5.21: Absolute eigenvalue shifts for patch 1.

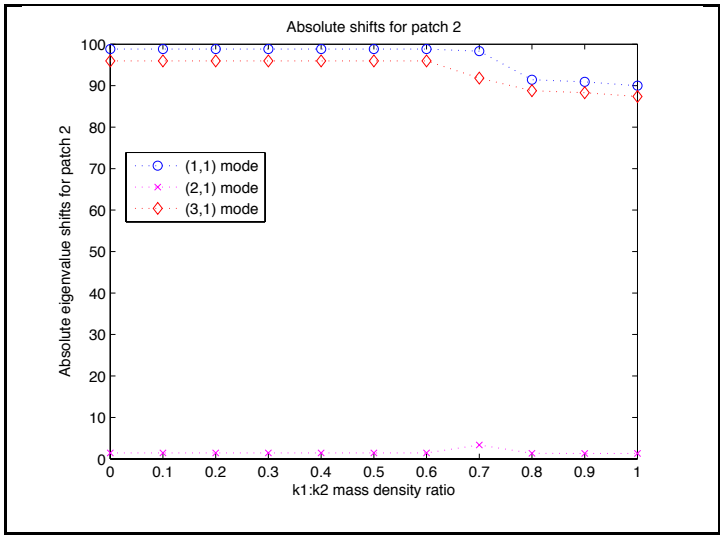


Figure 5.22: Absolute eigenvalue shifts for patch 2.

Sum of normalized signal shifts in the (1,1), (2,1), and (3,1) modes

The normalized version of the previous case shows some similarities. In Fig. 5.23, we see that again our objective function value increases linearly in added mass. In addition, in Fig. 5.24 we see that, as in the previous case, the increase is not linear in total mass. This is due to the response surface for the combined normalized modes, which is qualitatively similar to that of the non-normalized system in that it has one global maximum and additional local maxima. When patch 2 is the only one in play, it occupies the global maximum and we see a large eigenvalue shift per unit mass. When patch 1 is added (i.e. $k_1 > 0$), it is placed on a local maximum, with a smaller total shift value than the global. So patch 2 has a larger effect per unit mass than patch 1.

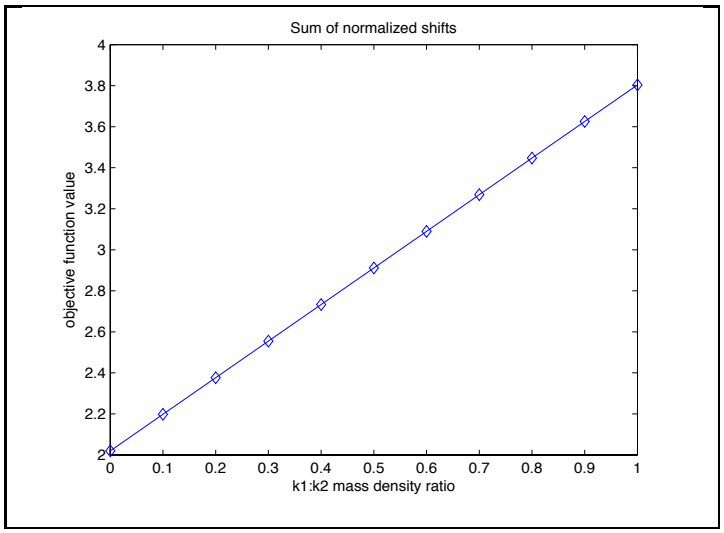


Figure 5.23: Function values plotted against $k_1 : k_2$, sum of normalized signal shifts in the (1,1), (2,1), and (3,1) modes.

When we look at the fractional signal shifts achieved in each mode, in Figs. 5.26, 5.27, and 5.28, we see that for $k_1 > 0$, our patches are fixed in eigenvalue space for all three modes. This stands in contrast to what we saw in the non-normalized case, despite some gross similarities in the effective mode (see Fig. 5.25) This tells us that measures such as the ratios of global optima to patches, or of local optima to patches, or of modes to patches,

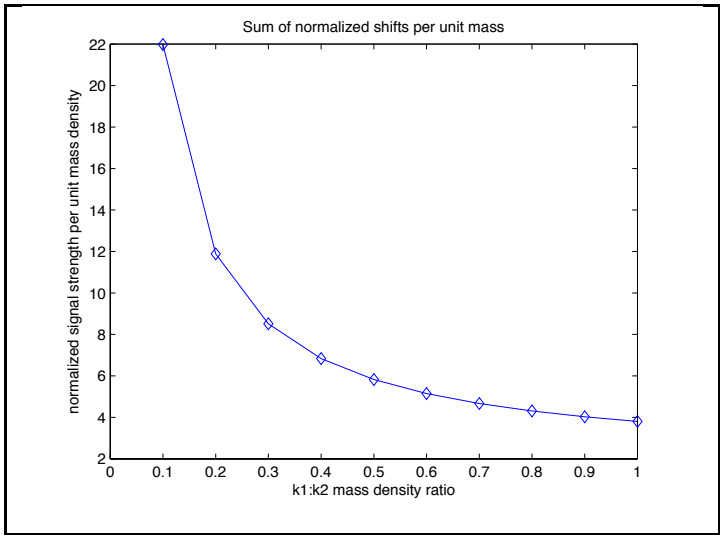


Figure 5.24: Function per unit mass, sum of normalized signal shifts in the (1,1), (2,1), and (3,1) modes.

are not sufficient to capture the behavior in a multi-mode system; the weighting attached to each mode can significantly influence the behavior of the system. In the previous case, for some weights of k_1, the loss of moving patch 2 off of the global maximum was offset by the gain of moving patch 1 closer to that global maximum. In this case, the latter gains are not sufficient to offset the former loss, and we see instead a solution that is static in the ratio $k_1 : k_2$. [4] That is, because the size of the membrane means that the separation of the anti-nodes is greater than the minimum separation distance between patches, the patches remain located on anti-nodes regardless of their masses.

[4] Actually, as in all our cases, for the special case of $k_1 = k_2$ the two patches are equivalent and can switch between positions for different instances of our GA program. Otherwise the statement holds.

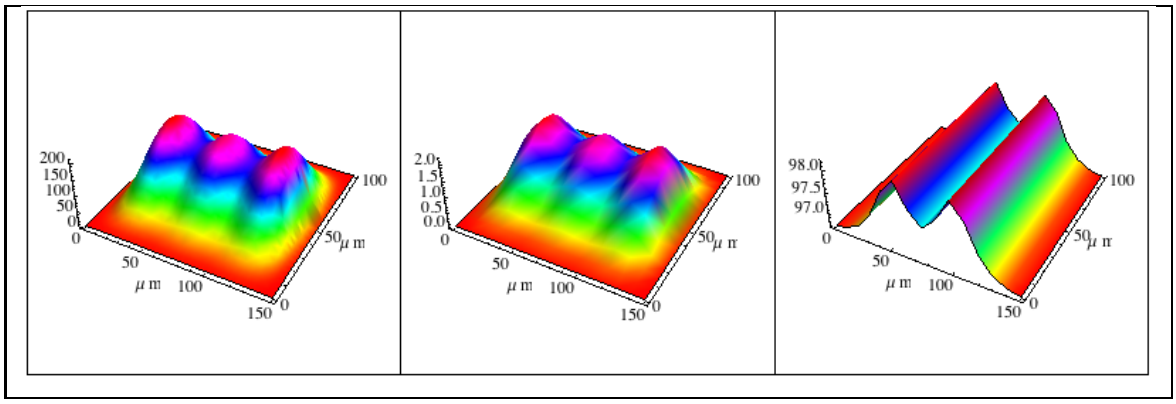


Figure 5.25: Effective mode for the (1,1) + (2,1) + (3,1) system; the normalized version of same; the ratio of the non-normalized to the normalized version of the effective mode, showing that scaling from the non-normalized to the normalized version varies locally across the membrane.

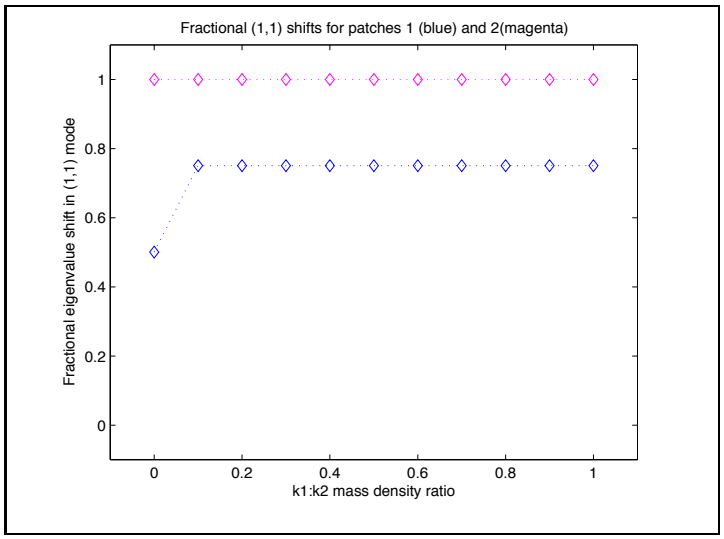


Figure 5.26: Fractional signal strength in (1,1) mode, for sum of normalized signal shifts in the (1,1), (2,1), and (3,1) modes.

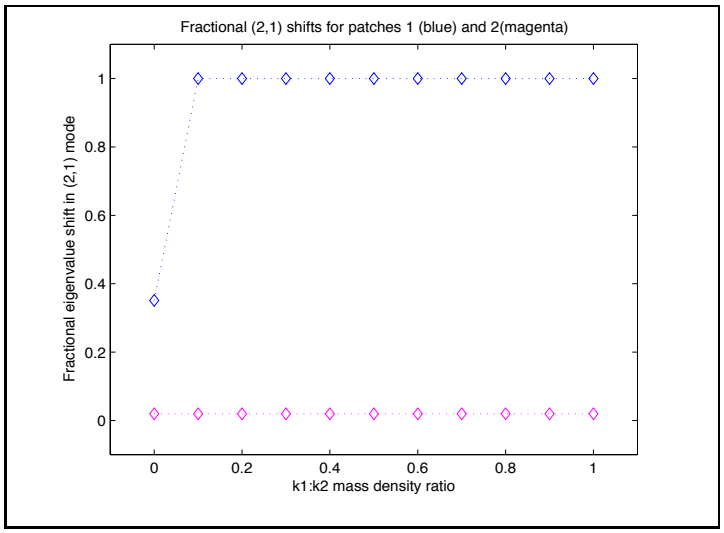


Figure 5.27: Fractional signal strength in (2,1) mode, for sum of normalized signal shifts in the (1,1), (2,1), and (3,1) modes.

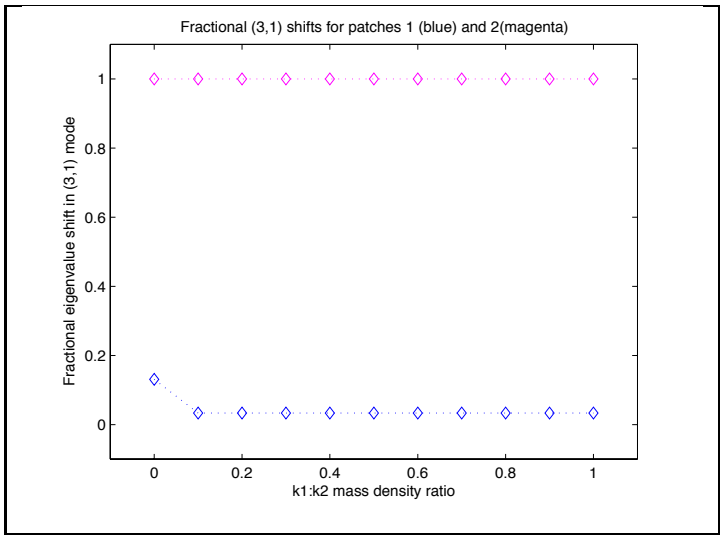


Figure 5.28: Fractional signal strength in (3,1) mode, for sum of normalized signal shifts in the (1,1), (2,1), and (3,1) modes.

5.2.2 Max-Min formulations

We now turn to the objective functions which maximize the minimum component shifts. As discussed in Secs. 4.4 and 4.1.1, the relative formulations and fractional formulations of these objective functions will be important.

Maximizing the minimum of non-normalized signal shifts in the (1,1) mode

In this objective function we consider the (1,1) mode exclusively, and look at two patches of added mass. The purpose of this objective is to maximize the minimum of two separate eigenvalue shifts, ensuring that the smaller of the two shifts is not "sacrificed" for the sake of the larger. Bear in mind that for this formulation the signal shift from the heavier patch does not directly contribute to the objective function - it contributes only via competing with the lighter patch. This is contrasted with the objectives we considered previously, where all eigenvalue shifts contributed directly and additively.

In Fig. 5.29 we see that as our mass ratio increases, so does our objective function value. This is because the minimum eigenvalue shift grows as the density of the lighter patch (patch 1) grows. (Recall that the signal shift induced by a patch scales linearly with the mass of the patch.) We initially see a linear increase as patch 1 grows heavier - this is because patch 1 is giving rise to the smaller signal shift, and so its position is optimized - centered on the optimal spot for a single patch. Increasing its mass therefore simply induces a corresponding increase in eigenvalue shift. However, as its mass nears that of patch 2, the patch 1 is no longer necessarily giving the smaller eigenvalue shift. Instead, patch positioning determines which patch gives rise to the smaller shift. Because of the minimum separation distance requirement, the two patches cannot both occupy the optimal vspot. Once this dynamic comes into play, the genetic algorithm will drive the two patches to positions where they give equal signal shifts.

In Fig. 5.30 we can see the results in terms of the objective function value per unit mass *directly contributing to the objective function.* (Recall that the objective function value is the

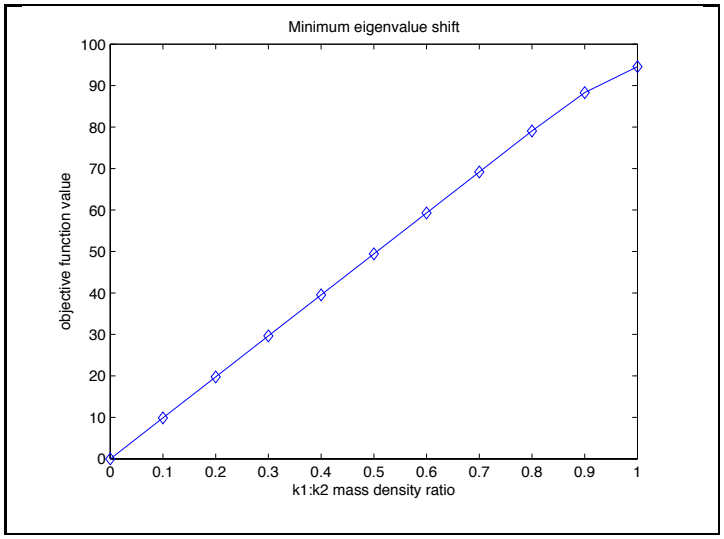


Figure 5.29: Function values plotted against k_1 : k_2, max-min of non-normalized signal shifts in the (1,1) mode.

smaller of the eigenvalue shifts induced, so it is directly considering only the contribution of one of the patches.) Therefore, this figure represents the eigenvalue shift per unit mass (i.e. the patch placement quality) of the patch with the smaller eigenvalue shift. When k_1 is low enough so that patch 1 induces a smaller eigenvalue shift than even a badly placed patch 2, patch 1's position is optimized. However, as the masses of the two patches converge, the shift per unit mass drops, because the two patches are competing for the same best spot; patch 1 is no longer occupying the single best position.

In Fig. 5.31 we see this more clearly. The fraction of the maximum that patch 1 is able to acheive drops as the two patches become comparable in density. When k_1 is small, the heavier patch has few constraints on position, so the eigenvalue shift it induces is somewhat arbitrary - rerunning the genetic algorithm would give different values for the fractional eigenvalues for these cases.

As was the case for the objective function summing the contributions of multiple patches in the (1,1) mode, this objective function is merely linearly scaled by normalization, because only one mode is involved. Therefore we do not separately consider the normalized version

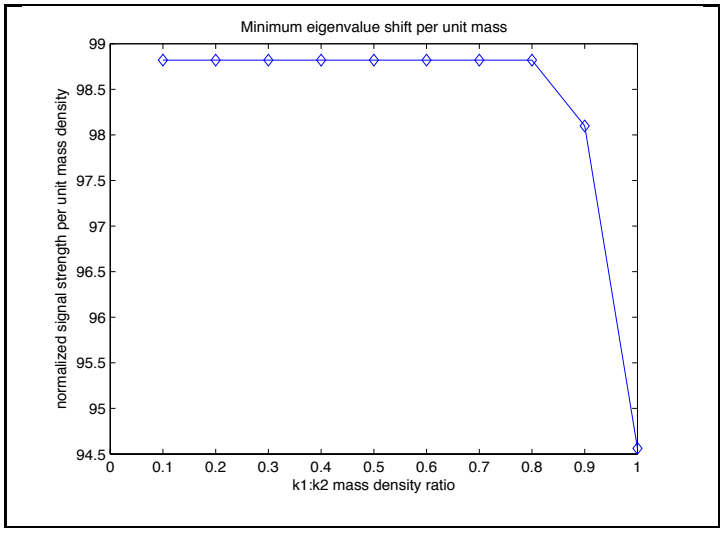


Figure 5.30: Function per unit mass, max-min of non-normalized signal shifts in the (1,1) mode.

of this function.

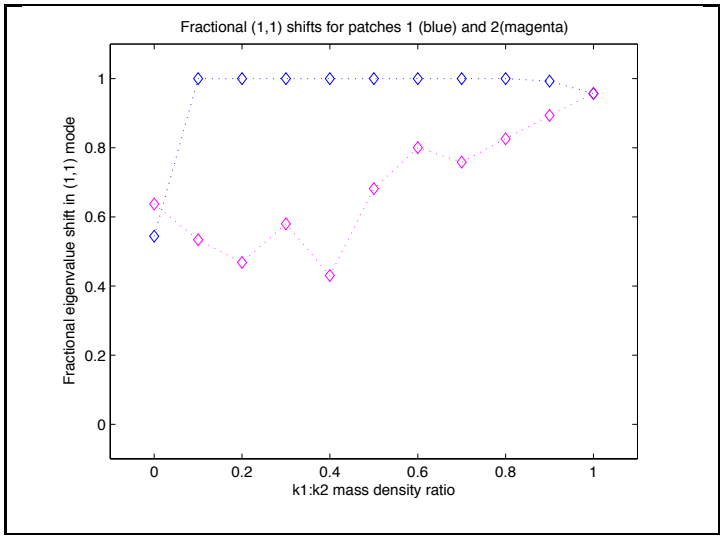


Figure 5.31: Fractional signal strength in (1,1) mode, for max-min of non-normalized signal shifts in the (1,1) mode.

Maximizing the minimum of non-normalized signal shifts in the (1,1) and (2,1) modes

In this objective function we consider the (1,1) and (2,1) modes, and two patches of added mass. The purpose of this objective function is to maximize the minimum of the four separate signal shifts induced. This case is more complicated than a single mode one, as we have to consider that improving an individual signal shift can impact the other shift for the associated patch directly, and not only via competition with the other patch for a better position.

In Fig. 5.32, we note that the objective function value rises linearly with the mass ratio (this is confirmed by Fig. 5.33). When we look at Figs. 5.34 and 5.35 we can see why: excluding the case where patch 1 is not present (i.e. has mass 0), it always occupies the same position in eigenvalue space for both the (1,1) and the (2,1) modes, even when the mass ratio equals 1. The position in question is one of the anti-nodes for the (2,1) mode. As we have not normalized our objective function components by their corresponding unperturbed

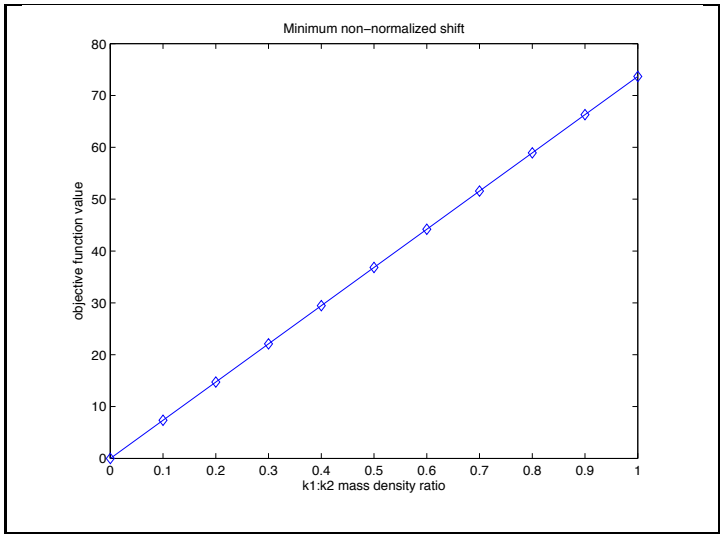


Figure 5.32: Function values plotted against $k_1 : k_2$, max-min of non-normalized signal shifts in the (1, 1) and (2,1) modes.

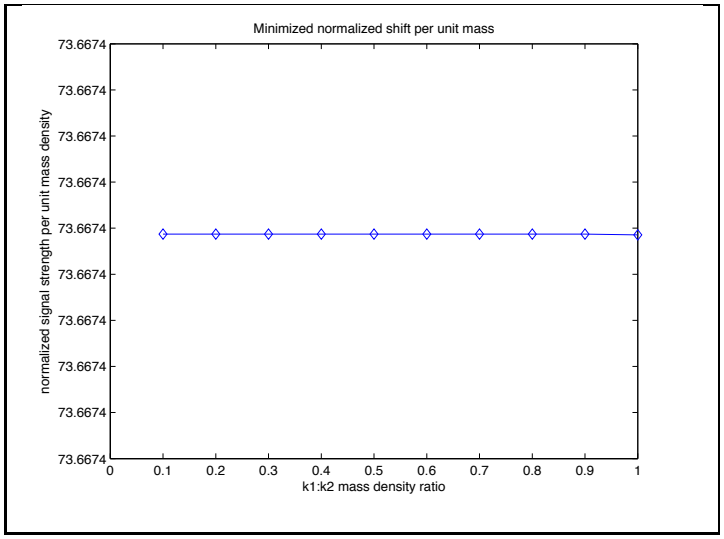


Figure 5.33: Function per unit mass, max-min of non-normalized signal shifts in the (1, 1) and (2,1) modes.

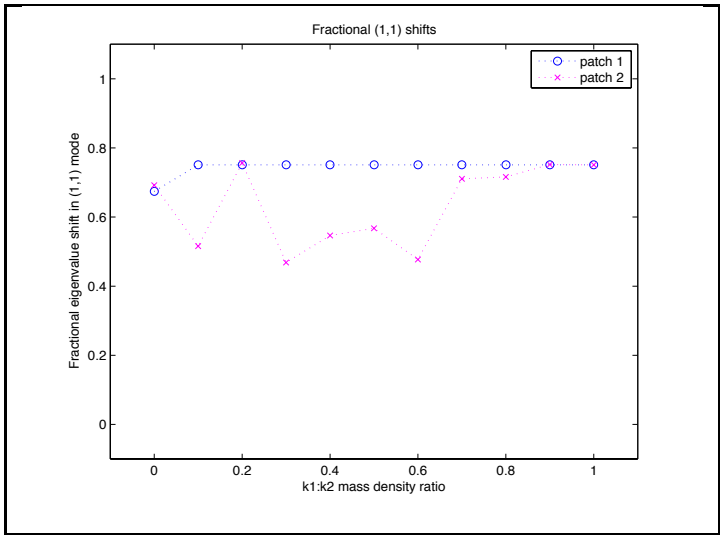


Figure 5.34: Fractional signal strength in (1,1) mode, for max-min of non-normalized signal shifts in the (1, 1) and (2,1) modes.

eigenvalues, and as we are not using a fractional formulation, the signal shift for patch 1, mode (2,1) is consistently the smallest shift, and hence is the one directly optimized. Fig. 5.39 does show that the signal shift for patch 2, mode (2,1) is indirectly optimized via the aforementioned method of "raising the floor".

When the masses of the two patches become equal, we see that both of them exhibit identical performance in both modes. This is because they are each centered on one of the (2,1) anti-nodes. If we had implemented a discrimination constraint, as discussed in Sec. 4.6, we would not see this symmetric solution.

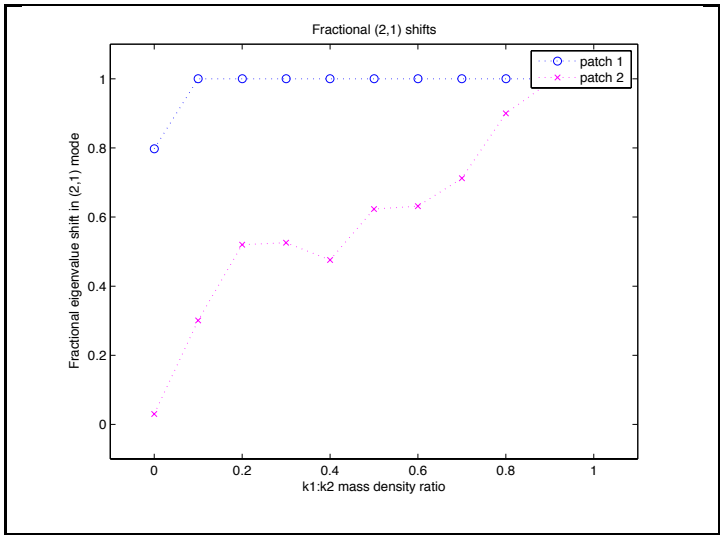


Figure 5.35: Fractional signal strength in (2,1) mode, for max-min of non-normalized signal shifts in the (1, 1) and (2,1) modes.

Maximizing the minimum of normalized signal shifts in the (1,1) and (2,1) modes

In this objective function we consider the (1,1) and (2,1) modes, and look at two patches of added mass. The purpose of this objective is to maximize the minimum of the four separate signal shifts induced. This case is the normalized version of the previous one.

In Fig. 5.36 we see that as our mass ratio increases, so does our function value. This is because the minimum signal shift grows as the mass of patch 1 grows. (Recall: the signal shift induced by a patch scales linearly with the mass of the patch.) We initially see a linear increase as patch 1 grows heavier - this is because patch 1, as the significantly lighter patch, is giving rise to the smallest signal shift, and so its position is optimized - centered on the optimal spot for a single patch. Increasing its mass therefore simply induces a corresponding increase in signal shift. However, as the mass of the lighter patch nears that of the heavier, the lighter patch is no longer necessarily giving the smallest signal shift - now which patch gives a smaller shift is dependent on positioning. Because of the minimum

separation distance requirement, the two patches cannot both occupy the optimal spot. Once this dynamic comes into play, the genetic algorithm will drive the two patches to positions where they give equal signal shifts.

In Fig. 5.37 this is seen more clearly. The fraction of the maximum that the lighter patch is able to achieve drops rapidly as the two patches become equal in mass.

In Figs. 5.38 and 5.39 we see that the because one of the signals from patch 1 acts as the minimum in most cases, our optimizer manipulates the position of patch 1 in order to maximize it. This leaves patch 2 mostly free in terms of position. Because we are dealing with the normalized cases, achieving 0.8 of the maximum possible signal in the (1,1) mode is no better than 0.8 in the (2,1) mode, so the normalized eigenvalue shifts for the lighter patch are driven to the same value in both modes.

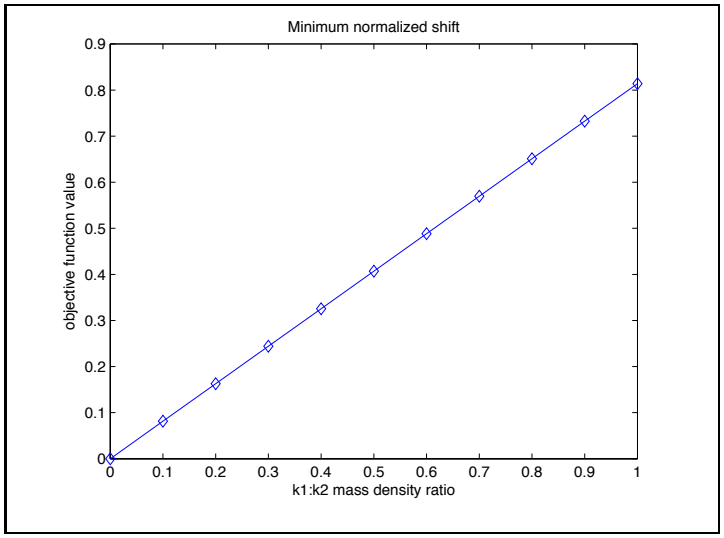


Figure 5.36: Function values plotted against $k_1 : k_2$, max-min of normalized signal shifts in the (1, 1) and (2,1) modes.

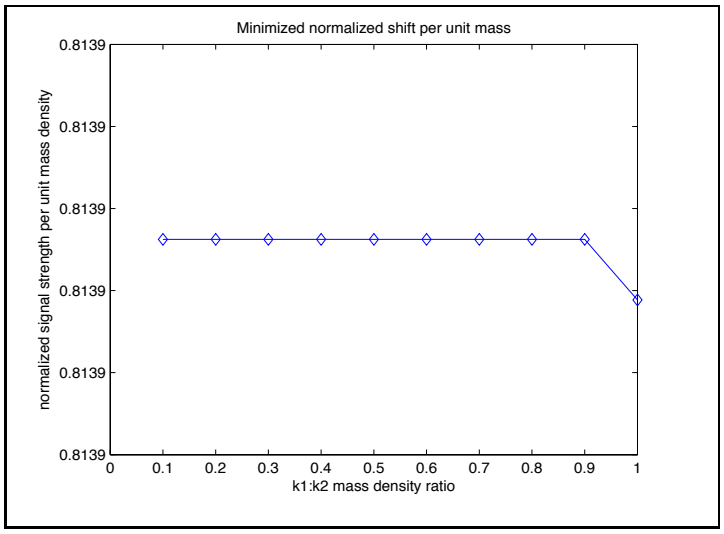


Figure 5.37: Function per unit mass, max-min of normalized signal shifts in the (1, 1) and (2,1) modes.

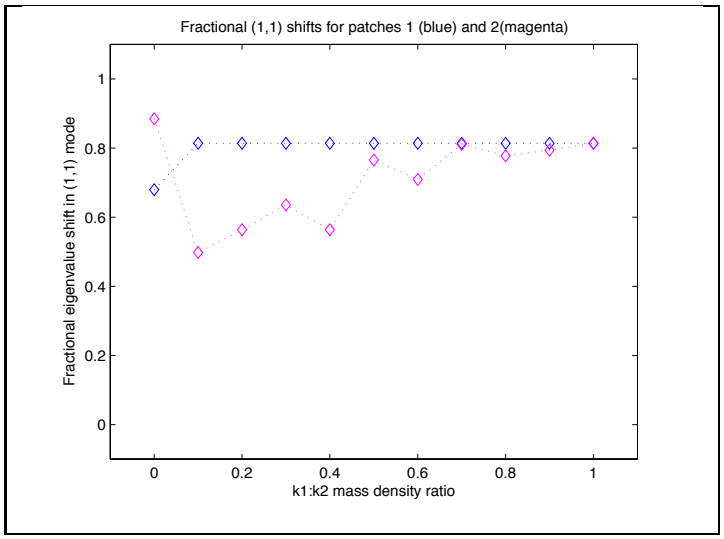


Figure 5.38: Fractional signal strength in (1,1) mode, for max-min of normalized signal shifts in the (1, 1) and (2,1) modes.

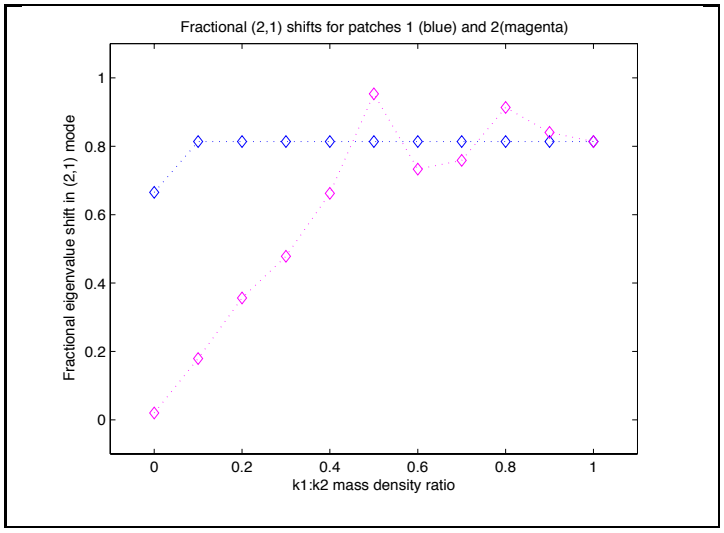


Figure 5.39: Fractional signal strength in (2,1) mode, for max-min of normalized signal shifts in the (1, 1) and (2,1) modes.

Maximizing the sum of dynamically custom-weighted signal shifts in the (1,1) and (2,1) modes

We have so far considered only two weighting schemes for our signal shifts - raw, unweighted shifts (i.e. all shifts weighted by a value of 1), and shifts weighted by their associated, unperturbed eigenvalues (normalization). However, we are not constrained to these options; we can introduce arbitrary weighting assignments for any of our basic objective functions. We present one example here.

Such assignments might reflect confidence (or lack thereof) in, for example, the accuracy of readout of a particular mode, or in the binding strength or specificity of a particular patch. Assigning custom weights therefore permits us to choose an objective function (and therefore a set of patch placements) which is more precisely suited to an individual situation. From differences between the non-normalized and normalized version of previous objective functions, we have seen that the weighting applied to a function can change its behavior. Now we will consider a case that differs from the previous ones in three ways: first, the weighting assignment depends on the relative sizes of the signal shifts. That is, the largest

signal shift is assigned a particular weight, whether that signal shift arises from patch 1 in mode (2,1), patch 2 in mode (1,1), etc. Therefore, as we increase the density of patch 1, changing the size order of the individual signal shifts, we are also changing their weights. Second, our weights differ not only between modes, but between patches (i.e. two patches in the same mode are weighted differently). Finally, we have large disparities between some of the weights, where previously such disparities were limited to the relative disparities in unperturbed values of the eigenmodes. In this objective function we consider the (1,1) and (2,1) modes, and look at two patches of added mass. The purpose of this objective is to maximize the weighted sum of all four eigenvalue shifts, one arising from each patch in each mode. However, the weighting depends on the values of the shifts, so this a max-sum problem with max-min characteristics. The weights are: 1,3,5, and 991 respectively for the biggest to the smallest shifts. As is usual, we see that the objective function value increases with increasing mass (see Fig. 5.40). However, this increase is actually superlinear, despite the fact that (as we shall see subsequently) our patch positions do not change with changing mass ratios. This is due to our dynamic weighting scheme - while the patch positions do not change with changing mass ratios, the weights assigned to those patches do. This shows the significance of understanding subtleties in the effects of the weighting schemes we use. While the function value increases superlinearly with the mass ratio $k_1 : k_2$, it increases sublinearly with the total added mass, as we can see in Fig. 5.41. This is because the patch associated with the maximally weighted signal shift contribution (in this case, patch 2) is remaining on an optimal position on the membrane. As it is contributing the largest signal shift, its larger contribution is weighted by 991 (and its smaller by either 3 or 5, depending on the $k_1 : k_2$ ratio). When we add more mass to the system, we are adding it to patch 1, whose contributions are multipled by only 1 and either 3 or 5.

In Figures 5.42 and 5.43, we see that both patch 1 and patch 2 remain fixed in their positions regardless of the masses (ignoring as usual the case of $k1 = 0$, in which the position of patch 1 is irrelevant). So despite the subtleties of the effects of the weighting scheme on the output of the objective function, it appears that in practical terms it is immaterial - the

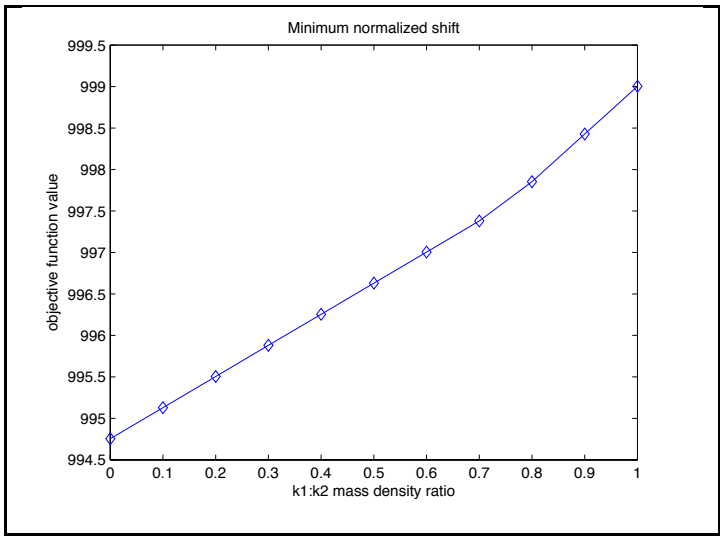


Figure 5.40: Function values plotted against $k_1 : k_2$, sum of dynamically weighted signal shifts in the (1,1) and (2,1) modes.

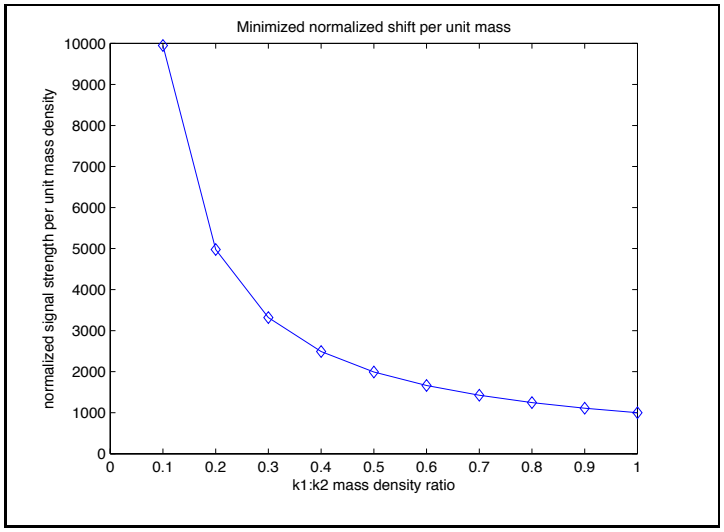


Figure 5.41: Function per unit mass, sum of dynamically weighted signal shifts in the (1,1) and (2,1) modes.

only consideration determining the placement of the patches is the relationship between the number of optimal positions and the number of patches. However, there is a subtlety here as well. We have previously seen that normalization can change the response surface of the effective mode. We now point out that some weighting schemes can change the number of anti-nodes in an effective mode, i.e. the number of optimal positions on a response surface. While that was not the case for this function, it is a consideration for future weighting schemes, and could be used to screen such schemes for suitability in objective functions. Finally, we note that the use of different weights for each contribution means that the response surface in question can differ within the same objective function, depending not only on which patch we are considering, but on the relative masses of the patches. This means that the potential for complex behavior increases even further. For this reason, we do not explore this type of objective function in depth here, but instead assign these functions to our future work section, discussed in

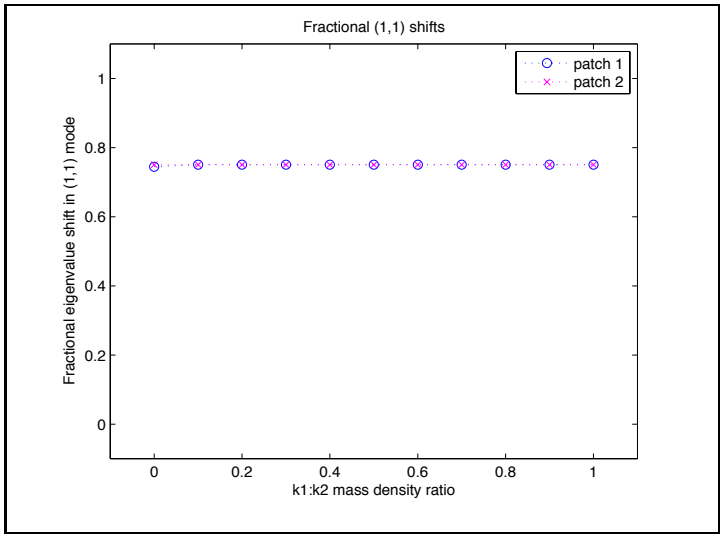


Figure 5.42: Fractional signal strength in (1,1) mode, sum of dynamically weighted signal shifts in the (1,1) and (2,1) modes.

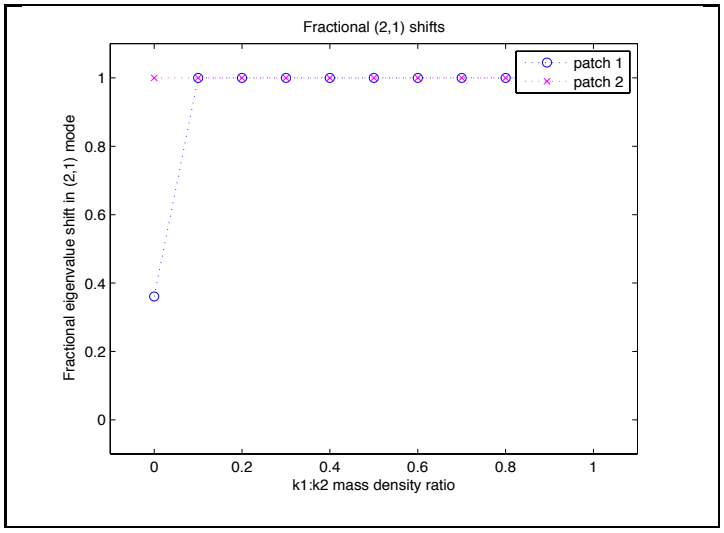


Figure 5.43: Fractional signal strength in (2,1) mode, sum of dynamically weighted signal shifts in the (1,1) and (2,1) modes.

5.2.3 Signal overlap

The final objective function we consider is that minimizing signal overlap resulting from two patches of added mass, in only one mode. This function is designed to allow discrimination between three cases (patch 1 shows binding, patch 2 shows binding, both patches show binding) despite an underdetermined system. This function is meaningful only when we assign ranges instead of individual values to the added mass factors k_1 and k_2. We therefore fix $k_2 = 1$ and vary k_1 from 0 to 1 as before, but in addition we define range parameters Δk_1 and Δk_2, so that for a given k_1, the possible range of bound mass is $[k_1, k_1 + \Delta k_1]$ (similarly for k_2). The addition of these two parameters increases the number of interesting test cases for this function; we present here the case for $\Delta k_1 = \Delta k_2 = 0.1$, i.e. for relatively small ranges of bound masses.

The formulation of this function minimizes not only the overlap between the induced signal shift ranges, but also the gaps between ranges, ensuring that we do not guarantee small overlaps only by making the total signal shifts small and far apart. Because we penalize gaps between shift ranges, we expect that the performance of our objective function will be

poor for small k and Δk values, as even centered on the anti-node these will result in small signal shift ranges. Therefore, for small enough k and Δk, we are ensured positive gaps [5]. So we expect to see higher function values for small k_1, although this does not reflect the quality of the patch positions. In constrast, we should see very good function values for larger k_1, while possibly observing decreases in the quality of the patch positions. Indeed, in Fig. 5.44 this is precisely what we see: the objective function's minimal value is non-zero but decreasing for $k_1 = 0, 0.1, 0.2$. Once $k_1 = 0.3$, the signal shift range induced by the patch is large enough to prevent any gaps between ranges, and the function value goes to zero, where it remains for further increases of k_1.

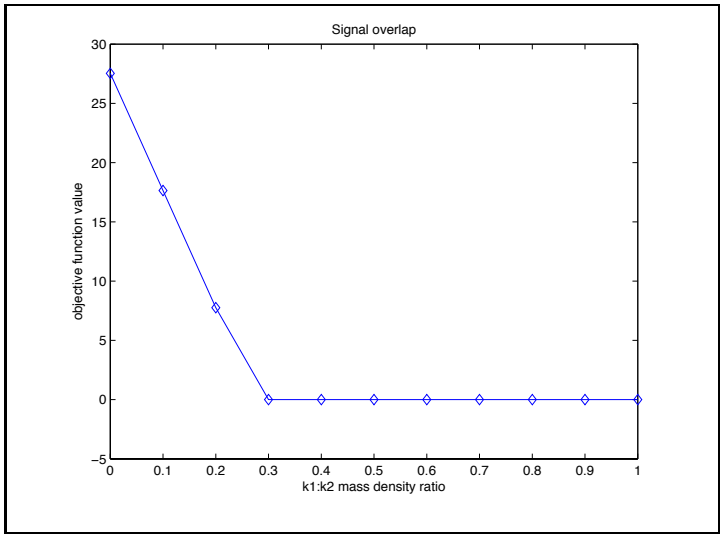


Figure 5.44: Function values plotted against $k_1 : k_2$, minimizing signal overlap in the (1,1) mode.

When we consider the quality of the positions chosen by the objective function, we note again competition between the patches for the single optimal position in the (1,1) mode, as seen in Fig. 5.45. When k_1 is small, patch 2 occupies the anti-node, and patch 1 is

[5] It is worth noting that small gaps between the ranges may be desirable in order to aid in discriminating between cases and reducing the confusing effects of noise. The objective function can be altered to prefer a fixed gap size instead of a gap of zero.

relegated to a relatively poor position, its distance away dictated by the minimal patch separation constraint. As k_1 increases, so does the size of the signal shift range induced by patch 1. To avoid overlaps (not only between the range induced by patch 1 and the range induced by patch 2, but between those ranges and that produced by summing the two of them), the size of the range induced by patch 2 must decrease, and so it must more to a less desirably patch position. As $k_1 -> k_2$, we see again that we are approaching a solution that is symmetric in the signal shift space.

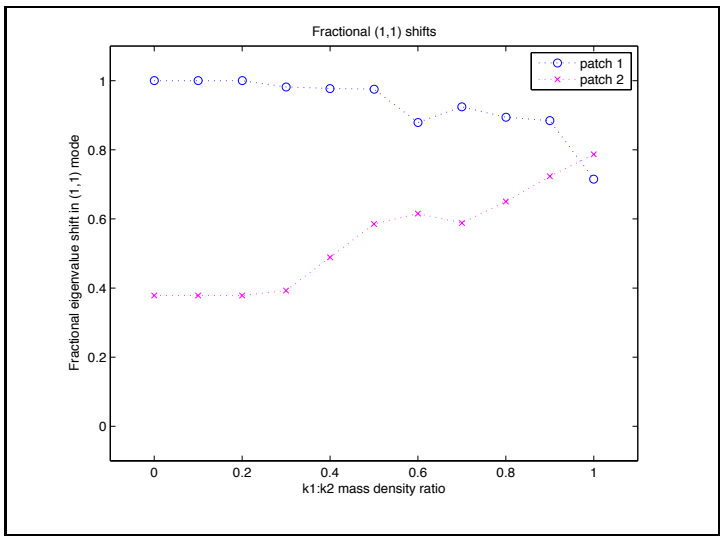


Figure 5.45: Fractional signal strength in the (1,1) mode, minimizing signal overlap in the (1,1) mode.

Chapter 6

Conclusions and future directions

6.1 Conclusions

The development of sensitive, accurate, and rapid sensors for detecting biological and chemical sensors has importance for many fields, including medicine and medical research, agricultural and food science, environmental monitoring and research, and anti-terrorism efforts. The performance and design of a biochemical sensor is enhanced by accurate device modeling and optimization, and multiplexing (the use of a single sensor to detect multiple substances simultaneously) expands the capability of such a device. For these reasons, we have chosen to pursue a comprehensive modeling and optimization program for a membrane-based acoustic wave MEMS biochemical sensor.

Sensors operate under a wide range of modalities. They may detect targets in gas, liquid, or solid samples. Target analytes may be detected via mechnical, electrical, chemical, or biological mechanisms, and detection may be signalled by changes in quantities such as fluoresence, color, capacitance, or mass. Our sensor uses target-specific binding agents to bind target analytes to the surface of a membrane resonator; the resulting change in mass is detected by monitoring the frequency response of the membrane.

Because we are using a membrane as our resonator, we can take advantage of the large surface area to determine where to bind our targets; the ability to control the distribu-

tion of mass on the surface permits us to both increase the sensitivity of our device; and to detect multiple targets simultaneously, allowing a single sensor to operate as as sensor array. The first step in taking advantage of these properties is to develop an accurate model of the membrane's response to different mass distributions. We begin with a classical second-order partial differential equation describing the the position function for a membrane undergoing out-of-plane vibrations, and verify that this is the appropriate model for our case (as opposed to, for example, a plate model or a varying-tension model). For a rectangular membrane with constant density and tension, the solution is known, allowing us to analytically calculate the frequency of the component eigenmodes. For uneven mass (and therefore uneven density) distributions, we use COMSOL, a software finite element method (FEM) package, to simulate the membrane's eigenmodes and eigenvalues. While the finite element simulations produce accurate numerical approximations of response of the membrane's frequency characteristics, they do not provide us with information about gradients, and in addition, each simulation has a high computational cost. This means that these simulations are not well-suited to the next step in our program, which is optimization. We therefore have used matrix perturbation analysis to develop an accurate reduced-order model to describe our sensor's behavior under distributed mass loading.

To perform the perturbation analysis, we extend the original model to include a small additional mass term, expressed as a small factor ϵmultiplied by the original mass term. We then consider a formal power series in terms of ϵ. The leading term in this series is the analytic solution to the unperturbed case, while later coefficients of the ϵ terms describe how the new solution deviates from that of the unperturbed case. In our case, we use a first-order perturbation for simplicity, and verified that the expected additional deviation resulting from higher-order terms is negligible. In order to prevent degenerate cases, we use Fredholm's Alternative Theorem, which imposes an additional restriction on our membrane geometry, requiring that membranes be non-square.

Having developed an accurate reduced-order model for our membrane's response to added mass functions, we can proceed to consider optimal mass distributions under different condi-

tions. We consider several different types of objectives, including maximizing sensitivity to one target, maximizing average or minimum sensitivity when several targets are considered, and ensuring discrimination between signals arising from multiple targets in underdetermined systems. We formulate constraints, beginning with those that make our expression for eigenvalue shifts, derived from our reduced-order model, analytically integrable. We add further constraints arising from the sensor characteristics, expected uses, and restrictions on patch functionalization.

We use two different optimization methods, depending on the specific optimization problem at hand. The linear proxy method we developed reduces the optimization search space from from two dimensions to one, and works rapidly and reliably. However, it is applicable in only a small number of cases. For the other cases, we use genetic algorithms to arrive at our solutions. Using a Matlab toolbox called the Genetic Algorithm Toolbox, we develop a program which treats each potential solution as a chromosome, able to undergo mutations. The fitness of each chromosome is evaluated at each step. Higher fitness chromosomes are more likely to be selected for mutation or to be included directly in the next generation of potential solutions. We iterate this process to converge to a set of maximally fit chromosomes.

We apply these optimization methods to our suite of objective functions, and use the results to characterize the behavior of our sensor and to examine the effectiveness of different objective functions under different conditions, such as the number and choice of operating modes, the number of functionalized patches, the number of target analytes, and the relative masses and concentrations of those analytes. This is a very broad space, and we often see tradeoffs in multiple elements, such as between targets and between modes, operating simultaneously.

The results of this research are

- An accurate reduced-order model of the membrane response to non-uniform mass distribution, applicable not only to our device's parameters, but to a wide range of alternative designs. This means that we can use the model not only to optimize our

device's performance, but to develop optimal device designs for different conditions; this option is discussed further in Sec. 6.2.

- A suite of objective functions and attendant constraints, designed to optimize sensor performance with respect to several different goals, and under varying conditions.
- A set of optimization methods tailored to the sensor characteristics, objective functions and model characteristics, and verified to work in test cases.
- A description of our sensor's behavior as we vary parameters of the mass distribution, selected modes, number of target analytes, and specific optimization goals. This description allows us to understand complex tradeoffs in the optimization search space, and to evaluate which operating modes, objective functions, and sensor target combinations result in improved sensor performance.

6.2 Future Directions

We envision four main components of the extension of this research. From the most specific and immediate to the least:

- First, we wish to refine the reduced-order model to capture more of the relevant physical operating conditions of our sensor, including factors such as viscous damping, and non-uniform tension within the membrane itself. The more precise and general our model is (while retaining the simplicity of the reduced-order model), the more we can use it to optimize operation under varying conditions.
- Second, so far our optimization has focused on implementation, i.e., how to best use the sensor we have; we have not yet considered optimization of the device design. Developing metrics for evaluating designs will allow us to close an optimization loop, and let our understanding of the device's operation determine the design details of the sensor itself. For example, this will allow us to choose optimal values for the shape , size and thickness of the membrane.

- Regardless of the design or model we choose, we know that many of the device characteristics will be subject to uncertainty. We need to consider how these uncertainties in model inputs cause uncertainty in the output. We have a special interest in ensuring that our system is robust and that our optimization results are meaningful in the face of the expected uncertainties. Our third branch of future research, therefore, is to perform uncertainty and sensitivity analyses.

- Finally, we wish to optimize sensor performance under a wider range of conditions. We have developed a suite of objective functions for the optimization of the performance of a single sensor, but we have not yet fully explored this space. In addition, we have considered the optimization of an array of sensors. Adding discrete choices of which sensor to use for each functionalized region will mean working with Mixed-Integer Non-Linear Programs (MINLPs) instead of the NLPs we have so far studied. For this portion of the research, we will again need to carefully formulate appropriate objective functions and constraints, and to choose and implement the proper optimization methods for this new set of problems.

6.2.1 Modeling under different physical conditions

The baseline model we use for determining the displacement of a membrane undergoing out-of-plane vibrations is formulated without any damping coefficients. However, we expect to operate our sensor in a fluid (either gas or liquid), and therefore need to incorporate viscous damping in the model. While this is straightforward for the uniform mass distribution case, or when using computational methods to calculate the frequency response, incorporating this effect into a reduced order model requires a new matrix perturbation analysis performed on the damped model.

In addition to damping, we must consider the fact that due to the specific mesh design of our sensor, we may see non-homogeneities or isotropic behavior in the membrane tension. To determine if this is the case, we will perform a detailed finite element simulation of

membrane vibration, this time including the micro-scale structural elements comprising the mesh. If we do indeed find that the tension is non-uniform, we will have to characterize this non-uniformity, and either mitigate it via changes in the mesh design, or include it in our reduced order model. There is no guarantee that we would be able to analytically develop a reduced order model that accurately models differential mass loading, viscous damping, and non-uniform tension. If we cannot, our options are to develop an empirical model from a large number of simulations, or to rely on numerical simulations as the inner loop of our optimization problem.

Other modeling options that would be dictacted by the physical properties of our sensor include altering the boundary conditions for the membrane or switching from a membrane to a plate, or composite plate and membrane, model.

Because our optimization framework is independent of our model details, we can develop and evaluate multiple models without rederiving the optimization procedures.

6.2.2 Uncertainty and sensitivity analyses

In this thesis, we have assumed that we have accurate values for all our device parameters. In the real world, this may not be the case; many of these parameters will be subject to uncertainty. An experimental uncertainty assessment is thus called for to quantify these uncertainties. In lieu of a full set of physical experiments, an extensive literature search could give us uncertainty estimates for some parameters, derived from prior physical experiments on related structural elements, functionalization methods, measurement systems, etc. Some experiments characterizing relevant aspect of our membrane have already been performed by other group members [5], and this information will be a valuable starting point.

In addition to a characterization of the uncertainties, we need to know how sensitive our sensor response is to variations in parameter values. We therefore propose a sensitivity analysis to determine how robust our model is to uncertainties in input variables and parameters. This will allow is to do several things. First, we can identify inputs which lead to very large or very small uncertainties in outputs. We may be able to eliminate some

of the latter inputs, simplifying the model, while we will need to focus on ways to reduce uncertainty in the former to improve the robustness of our model. A clearer understanding of the relationship between inputs and outputs, even where the resulting uncertainties in the output are neither very large nor very small, will allow us to more easily optimize our sensor design for specific sets of conditions. The uncertainty analysis coupled with the sensitivity analysis will allow us to evaluate our confidence in the model and to see where improvements can best be made.

We propose using a combination of two methods of sensitivity analysis: calculating partial derivatives to elucidate relationships between specific pairs of inputs and outputs; and variance-based sensitivity analysis, which treats the model as a black-box [61].

Because we have a relatively simple analytical model, we are able to take partial derivatives of the relevant outputs (e.g. eigenvalues) with respect to inputs (e.g. $x-$ or $y-$position of patch, density) without relying on numerical methods. This will give us a large set of functions expressing sensitivity to the different possible inputs, and show us which ones are individually significant. However, this method only considers a single input variable at a time, and therefore does not account for interactions between variables. To measure these effects, we turn to variance-based sensitivity methods.

Variance-based methods take a probabilistic approach by modeling both inputs and outputs as probability distributions. Variance decomposition is then employed to attribute fractions of the variance to different inputs, and, critically, to interactions between inputs. The decomposition results in first-order sensitivity indices, one for each input i, which measure the fractional contribution to the total variance arising from varying i. This contribution is the effect of varying only i, but it is averaged over variations in other inputs to avoid measuring the sensitivity to i at only one point. This averaging is what gives this mehod its global applicability. Higher-order indices are used to measure the contributions from the interactions between multiple input variables.

Again the fact that our model is a simple analytical one is advantageous; we expect that some of the sensitivity indices can be calculated analytically. If this is not the case, the

indices may be estimated using Monte Carlo methods to sample a large sequence of input values.

6.2.3 Membrane design optimization

The optimization described in this thesis assumed a particular membrane design with fixed characteristics. However, we have some control over many of these parameters, such as membrane dimensions, density, tension, and material properties. In addition, the use of MEMS technology means that we can control the distribution of materials within the membrane itself, building in inhomogeneities to enhance specific sensor properties or mode behaviors. In order to optimize the design of the membrane, we need first to develop metrics to evaluate different designs; we again propose considering characteristics such as sensitivity, robustness, etc., and also expect to expand this list to include things like applicability to a broad range of sensing target characteristics or, on the other end of the spectrum, optimization for very specific sensing problems.

The use of MEMS technology means that we are not restricted to a uniform density across the membrane, but instead can specify the distribution of mass within the membrane. This capability allows us to change the characteristics of individual modes, including mode shapes and eigenvalues, and raises the possibility of optimizing specific mode characteristics to improve sensor performance.

6.2.4 Additional objective functions for the single-membrane case

With one brief exception, we considered only uniformly weight signal shifts, or normalized signal shifts. We wish to consider a broad range of weight assignments for our signal shifts. Such assignments might reflect confidence (or lack thereof) in, for example, the accuracy of readout of a particular mode, or in the binding strength or specificity of a particular patch. Assigning custom weights therefore permits us to choose an objective function (and therefore a set of patch placements) which is more precisely suited to an individual situation. Further, we can apply these weight assignments not only to max-sum objective functions, but to any

type we choose, opening the door to increasingly specialized optimization functions.

The majority of our objective functions showed sensitivity in their results to the ratio of bound masses on the surface; this means that in cases where we may expect a broad range of masses, it is unclear which ratio to use to determine patch positioning. We wish to develop objective functions which will take the distribution of possible mass ratios into to account when determining where to place patches. These objective functions will naturally vary depending on whether we are hoping to determine the quantities bound, or to determine simply whether specific targets are present or absent.

Depending on the application, a false positive (or a false negative) may be a very bad outcome or a neutral one. Objective functions which take this, and related information, into account are therefore highly desirable. For example, functions could minimize the false positive rate; or maximize sensitivity, specificity, accuracy, or positive predictive value. [1] Working within the framework of the receiver operating characteristic and related performance measures will allow us to "tune" our sensor to create tests which have the desired statistical classification properties.

6.2.5 Multiple membrane and multiple chip optimization

We have considered here the optimization of a single membrane with distributed masses. Given the design of the sensor chip and the need for sensor arrays, there are two natural extensions to this work: the optimization of a single chip with multiple membranes, and the optimization of an array consisting of multiple chips. This research again requires the development of new objective functions and constraints. For multiple membrane optimization, we will consider the characteristics of the problem formulation and the single membrane case, as well as new issues which arise from adding additional membranes. For example, we will need to quantify changes in sensitivity or accuracy which may come about via the distribution of functionalized regions over multiple membranes, and re-evaluate constraints

[1] As such objective functions would depend on estimates of confidence in the validity of the signal shifts resulting from binding, custom weight assignments are necessary for their development.

on the placement of regions when more than one membrane is available for functionalization. The addition of discrete choices to the set of continuous choices used before, will mean that we will need to model our optimization problem as an MINLP, necessitating new optimization methods.

For multiple chip optimization, where each chip will use a separate portion of our sample, we have to consider additional factors such as unevenly mixed samples, contamination, and chip-to-chip variability.

The increase in information means that we will be able to use Bayesian statistical methods to evaluate or improve the accuracy of our output. These methods will allow us to filter noise arising from background binding and to reduce the adverse effects of individual membrane failures.

Appendix A

Code examples and notes

A.1 Mathematica code

We performed shift measure and linear proxy calculations, and produced images of mode shapes, response surfaces, and effective modes using Mathematica. The following code calculates shift measures and produces associated contour plots for various modes and effective modes using our standard simulation parameters.

```
delx=10;
dely=10;
a=150;
b=100;
k=1;
n1=1;
n2=2;
n3=3;
m1=1;
m2=2;
m3=3;
```

```
d=20;
t=12;

smoneone[x_,y_] = k*Integrate[(Sin[Pi*n1*x/a])^2, {x,x-0.5*delx, x+0.5*delx}]*
Integrate[(Sin[Pi*m1*y/b])^2, {y, y-0.5*dely, y+ 0.5*dely}];
smonetwo[x_,y_] =k*Integrate[(Sin[Pi*n1*x/a])^2, {x,x-0.5*delx, x+0.5*delx}]*
Integrate[(Sin[Pi*m2*y/b])^2, {y, y-0.5*dely, y+ 0.5*dely}];
smtwotwo[x_,y_] =k*Integrate[(Sin[Pi*n2*x/a])^2, {x,x-0.5*delx, x+0.5*delx}]*
Integrate[(Sin[Pi*m2*y/b])^2, {y, y-0.5*dely, y+ 0.5*dely}];
smtwoone[x_,y_] =k*Integrate[(Sin[Pi*n2*x/a])^2, {x,x-0.5*delx, x+0.5*delx}]*
Integrate[(Sin[Pi*m1*y/b])^2, {y, y-0.5*dely, y+ 0.5*dely}];
smonethree[x_,y_] =k*Integrate[(Sin[Pi*n1*x/a])^2, {x,x-0.5*delx, x+0.5*delx}]*
Integrate[(Sin[Pi*m3*y/b])^2, {y, y-0.5*dely, y+ 0.5*dely}];
smtwothree[x_,y_] =k*Integrate[(Sin[Pi*n2*x/a])^2, {x,x-0.5*delx, x+0.5*delx}]*
Integrate[(Sin[Pi*m3*y/b])^2, {y, y-0.5*dely, y+ 0.5*dely}];
smthreeone[x_,y_] =k*Integrate[(Sin[Pi*n3*x/a])^2, {x,x-0.5*delx, x+0.5*delx}]*
Integrate[(Sin[Pi*m1*y/b])^2, {y, y-0.5*dely, y+ 0.5*dely}];
smthreetwo[x_,y_] =k*Integrate[(Sin[Pi*n3*x/a])^2, {x,x-0.5*delx, x+0.5*delx}]*
Integrate[(Sin[Pi*m2*y/b])^2, {y, y-0.5*dely, y+ 0.5*dely}];

cplotsmonetwo=ContourPlot[smonetwo[x,y],{x,0,a}, {y,0,b}, Axes->True,
 AxesLabel->{\[Mu]m,\[Mu]m}, ContourShading->False,
 AspectRatio->Automatic, ContourStyle->Automatic,
 ColorOutput->Automatic, ColorFunction->Hue, Contours ->
 20,BaseStyle->{FontFamily->"Times", FontSize ->t}, Frame -> None]

plotsmonetwo=Plot3D[smonetwo[x,y],{x,0,a}, {y,0,b}, Axes->True,
 AxesLabel->{\[Mu]m,\[Mu]m}, AspectRatio->Automatic,
```

```
 ColorOutput->Automatic, ColorFunction->Hue,
 BaseStyle->{FontFamily->"Times", FontSize ->t}, Ticks->{Automatic,
 Automatic, None}]

lineplotsmonetwo=Plot[smonetwo[75,y], {y,0,b}, AxesLabel->{"\[Mu]m",
 ""}, PlotLabel->"eigenvalue shift function for (2,1) mode along the
 line y=75", Ticks -> {Automatic, None}]
 lineplotsmoneone=Plot[smoneone[75,y], {y,0,b/2},
 AxesLabel->{"\[Mu]m", "\[Lambda] shift"}, Ticks->{Automatic, None},
 PlotLabel->"eigenvalue shift function for (1,1) mode along the proxy
 line x=50"] lineplot2smoneone=Plot[smoneone[75,y], {y,0,b/2},
 AxesLabel->{"\[Mu]m", "\[Lambda] shift"}, Ticks->{Automatic, None},
 PlotStyle->{Thick, Blue}]

normplot3dcombo123 =
 Plot3D[smoneone[x, y]/smoneone[75, 50] +
   smtwoone[x, y]/smtwoone[37.5, 50] +
   smthreeone[x, y]/smthreeone[75, 50], {x, 0, a}, {y, 0, b},
  Mesh -> None, Boxed -> False, Axes -> True,
  AxesLabel -> {\[Mu]m, \[Mu]m}, AspectRatio -> Automatic,
  ColorOutput -> Automatic, ColorFunction -> Hue,
  BaseStyle -> {FontFamily -> "Times", FontSize -> t}]

plot3dcombo123 =
 Plot3D[smoneone[x, y] + smtwoone[x, y] + smthreeone[x, y], {x, 0,
   a}, {y, 0, b}, Axes -> True, AxesLabel -> {\[Mu]m, \[Mu]m},
  Mesh -> None, Boxed -> False, AspectRatio -> Automatic,
  ColorOutput -> Automatic, ColorFunction -> Hue,
```

```
  BaseStyle -> {FontFamily -> "Times", FontSize -> t, Mesh -> None}]


plot3dratio123 =
 Plot3D[(smoneone[x, y] + smtwoone[x, y] +
     smthreeone[x, y])/ (smoneone[x, y]/smoneone[75, 50] +
     smtwoone[x, y]/smtwoone[37.5, 50] +
     smthreeone[x, y]/smthreeone[75, 50]), {x, 0, a}, {y, 0, b},
  Axes -> True, AxesLabel -> {\[Mu]m, \[Mu]m}, Mesh -> None,
  Boxed -> False, AspectRatio -> Automatic, ColorOutput -> Automatic,
  ColorFunction -> Hue,
  BaseStyle -> {FontFamily -> "Times", FontSize -> t, Mesh -> None}]
```

An example of linear proxy code:

```
Needs["PlotLegends`"]
contourPlotRule = ({EdgeForm[],
     r_?(MemberQ[{RGBColor, Hue, CMYKColor, GrayLevel}, Head[#]] &),
     i___} :> {EdgeForm[r], r, i});


delx = 10;
dely = 10;
a = 150;
b = 100;


k1 = 0;
k2 = 1;
n1 = 1;
n2 = 2;
```

```
n3 = 3;
m1 = 1;
m2 = 2;
m3 = 3;
d = 10;

smoneone[x_, y_, k_] =
  k*Integrate[(Sin[Pi*n1*x/a])^2, {x, x - 0.5*delx, x + 0.5*delx}]*
   Integrate[(Sin[Pi*m1*y/b])^2, {y, y - 0.5*dely, y + 0.5*dely}];
smonetwo[x_, y_, k_] =
  k*Integrate[(Sin[Pi*n1*x/a])^2, {x, x - 0.5*delx, x + 0.5*delx}]*
   Integrate[(Sin[Pi*m2*y/b])^2, {y, y - 0.5*dely, y + 0.5*dely}];
smtwotwo[x_, y_, k_] =
  k*Integrate[(Sin[Pi*n2*x/a])^2, {x, x - 0.5*delx, x + 0.5*delx}]*
   Integrate[(Sin[Pi*m2*y/b])^2, {y, y - 0.5*dely, y + 0.5*dely}];
smtwoone[x_, y_, k_] =
  k*Integrate[(Sin[Pi*n2*x/a])^2, {x, x - 0.5*delx, x + 0.5*delx}]*
   Integrate[(Sin[Pi*m1*y/b])^2, {y, y - 0.5*dely, y + 0.5*dely}];
smonethree[x_, y_, k_] =
  k*Integrate[(Sin[Pi*n1*x/a])^2, {x, x - 0.5*delx, x + 0.5*delx}]*
   Integrate[(Sin[Pi*m3*y/b])^2, {y, y - 0.5*dely, y + 0.5*dely}];
smtwothree[x_, y_, k_] =
  k*Integrate[(Sin[Pi*n2*x/a])^2, {x, x - 0.5*delx, x + 0.5*delx}]*
   Integrate[(Sin[Pi*m3*y/b])^2, {y, y - 0.5*dely, y + 0.5*dely}];
smthreeone[x_, y_, k_] =
  k*Integrate[(Sin[Pi*n3*x/a])^2, {x, x - 0.5*delx, x + 0.5*delx}]*
   Integrate[(Sin[Pi*m1*y/b])^2, {y, y - 0.5*dely, y + 0.5*dely}];
smthreetwo[x_, y_, k_] =
```

```
  k*Integrate[(Sin[Pi*n3*x/a])^2, {x, x - 0.5*delx, x + 0.5*delx}]*
   Integrate[(Sin[Pi*m2*y/b])^2, {y, y - 0.5*dely, y + 0.5*dely}];

t = 12;

lponeone[x_, k_] = smoneone[x, 50, k]

PP = Plot[lponeone[x, 1], {x, 0, 150}]

ContourPlot[smoneone[x, y, 1], {x, 0, 150}, {y, 0, 100},
 AspectRatio -> Automatic, ContourShading -> None]

out = Maximize[{lponeone[x1, 1] + lponeone[x2, 1], x1 < a/2, x2 < a/2,
    x1 + x2 + delx + d <= a}, {x1, x2}]

For[k1 = -0.1, k1 < 1.1, k1 = k1 + 0.1;
 out = Maximize[{lponeone[x1, k1] + lponeone[x2, k2], 0 < x1 <= a/2,
    0 < x2 <= a/2, x1 + x2 + delx + d <= a}, {x1, x2}], Print[out]]

llp1 = ListLinePlot[{{0, 98.8198}, {0.1, 107.2}, {0.2,
    115.804}, {0.3, 124.59}, {0.4, 133.525}, {0.5, 142.584}, {0.6,
    151.746}, {0.7, 160.994}, {0.8, 170.314}, {0.9, 179.685}, {1.0,
    189.128}}, PlotMarkers -> {"x", 14},
  PlotLabel -> "sum of eigenvalue shifts from 2 patches",
  AxesLabel -> {"k1:k2 ratio", ""}, RotateLabel -> True,
  PlotRange -> {{-0.1, 1.1}, {90, 190}}]
```

A.2 Matlab code

The bulk of our optimization was performed using Matlab and the GA toolbox. We define baseline functions such as the signal shifts induced by patches in various modes, e.g.:

```
% Signal11.m
%
% This function calculates the normalized signal in the (1,1) mode
% for a dx by dy patch on a membrane that is a by b units in size.
%
% Syntax:z = Signal11(x,y)
%
% Input parameters: x & y : the (x,y) position of the bottom left
% corner of the patch
%%
% Output parameter: z: the signal
%

function z = Signal11(x,y)

% Membrane and patch size characteristics
a=150;
b=100;
dx=10;
dy=10;

intx=2.*pi.*dx + a.*sin((2.*pi.*x)./a) - a.*sin((2.*pi.*(x+dx)./a));
inty=2.*pi.*dy + b.*sin((2.*pi.*y)./b) - b.*sin((2.*pi.*(y+dy)./b));
```

```
z=(1./(16.*pi.^2)).*intx.*inty;
```

The baseline functions are called by objective functions.

```
% OBJJANE5.M  (OBJective function - Jane Valentine #1)
%
% This function maximizes the minimum eigenvalue shift in
% the (1,1) mode for two patches.
%
%
% Syntax:  ObjVal = objfun1(Phen)
%
% Input parameters:
%    Chrom     - Matrix containing the chromosomes of the current
%                population. Each row corresponds to one individual's
%                string representation.
%    Phen      - is the real-value representation of the binary
%                string representation used in Chrom.  Phen is
%                recovered from Chrom via Phen = bs2rv(Chrom, FieldD).
%
% Output parameters:
%    ObjVal    - Column vector containing the objective values of the
%                individuals in the current population.
%
%

function ObjVal = objjane5(Phen, k1, k2);
```

```
[Nind, Nvar] = size(Phen);

if Nvar ~= 4
  error('size of matrix Chrom not correct for function evaluation')
end

if Nind ~= 1
    error('size of matrix Chrom not correct for function evaluation')
end

% Take columns from Phen and assign as variables

x1 = Phen(:,1);
x2 = Phen(:,2);
y1 = Phen(:,3);
y2 = Phen(:,4);

p1 = 100;

% (x1,y1) represents the bottom left of patch 1, while (x2,y2) is the
% bottom left of patch 2.
%
% This function represents the minimum of two signals
% from the (1,1) mode: we compare the
% signal arising in the (1,1) mode for patch 1 to the signal arising in
% the (1,1) mode for patch 2.  We negate the
% minimum as it is to be used in a minimization routine, and we
% want to maximize it.
```

```
%
% M is a penalty function to enforce minimum separation distance
% between two patches.

M=0;

if sqrt((x1-x2).^2 + (y1-y2).^2) < 20;
M=1000;
else
M=0;
end

s1=(k1.*Signal11(x1,y1));
s2=(k2.*Signal11(x2,y2));
s=[s1,s2];

ObjVal = -1.*min(s) + M;

output=ObjVal;
```

Objective functions are called by a code written to run a genetic algorithm program; many of the functions in the following piece of code are part of the GA toolbox.

```
% SGAJANE.M  (Genetic Algorithm run on given objective function and
% parameters.)
%
%
% This function implements the Simple Genetic Algorithm described
```

```
% in the examples section of the GA Toolbox manual, and is adapted
% to use objective functions written by Jane Valentine
%

% Author:     Andrew Chipperfield
% History:    23-Mar-94     file created
%             12-Oct-05     file adapted by Jane Valentine
%             17-Sep-07     further adaptations by J. Valentine
%      18-Feb-08    further adaptations by J. Valentine

function output = sgajane(objective, k1, k2)

NIND = 50;           % Number of individuals per subpopulation
MAXGEN = 150;        % maximum Number of generations
GGAP = .9;           % Generation gap, how many new individuals are created
NVAR = 4;            % Number of decision variables
PRECI = 25;          % Precision of binary representation

% Declare global variables
   global FieldD Chrom FitnV SelCh

% Build field descriptor
   FieldD = [rep([PRECI],[1, NVAR]); 45 45 20 20; 95 95 ...
             70 70; rep([1; 0; 1 ;1], [1, NVAR])];

% Initialise population
   Chrom = crtbp(NIND, NVAR*PRECI);
```

```
% Reset counters
   Best = NaN*ones(MAXGEN,1);% best in current population
   gen = 0;% generational counter
   lc = 0;                       % last gen. at which obj. val. changed

% Evaluate initial population, and store in matrix with population values

% ObjV=[feval(objective,bs2rv(Chrom,FieldD),k1,k2), bs2rv(Chrom, FieldD)];

switch objective
 case {1}
  disp('Objective function 1')
  ObjV = [objjane1(bs2rv(Chrom,FieldD),k1,k2), bs2rv(Chrom, FieldD)];
 case {2}
  disp('Objective function 2')
  ObjV = [objjane2(bs2rv(Chrom,FieldD),k1,k2), bs2rv(Chrom, FieldD)];
 case{3}
  disp('Objective function 3')
  ObjV = [objjane3(bs2rv(Chrom,FieldD),k1,k2), bs2rv(Chrom, FieldD)];
 end

% Track best individual and display convergence
   Best(gen+1) = min(ObjV(:,1));
   plot(Best,'ro'); xlabel('generation'); ylabel('f(x)');
   text(0.5,0.95,['Best = ', num2str(Best(gen+1))],'Units','normalized');
   drawnow;

% Generational loop
```

```
  while gen < MAXGEN,

   % Assign fitness-value to entire population
      FitnV = ranking(ObjV(:,1));

   % Select individuals for breeding
      SelCh = select('sus', Chrom, FitnV, GGAP);

   % Recombine selected individuals (crossover)
      SelCh = recombin('xovsp',SelCh,0.7);

   % Perform mutation on offspring
      SelCh = mut(SelCh);

   % Evaluate offspring, call objective function

%   ObjVSel=feval(objective,bs2rv(SelCh,FieldD),k1,k2);

switch objective
case {1}
 ObjVSel = objjane1(bs2rv(SelCh,FieldD),k1,k2);
case {2}
 ObjVSel = objjane2(bs2rv(SelCh,FieldD),k1,k2);
case {3}
 ObjVSel = objjane3(bs2rv(SelCh,FieldD),k1,k2);
end

   % Reinsert offspring into current population
```

```
        [Chrom ObjV(:,1)]=reins(Chrom,SelCh,1,1,ObjV(:,1),ObjVSel);

    % Increment generational counter
        gen = gen+1;

    % Update display and record current best individual
        Best(gen+1) = min(ObjV(:,1));
        plot(Best,'ro'); xlabel('generation'); ylabel('f(x)');
        text(0.5,0.95,['Best = ', num2str(Best(gen+1))],'Units','normalized');
        drawnow;
    % Update last change
        if Best(gen+1)~= Best(gen);
          lc=gen+1;
        else
          lc=lc;
        end
   end

% Export output so I can graph the results.
% output=

% End of GA
```

Finally, to run over many k values, we loop the genetic algorithm program.

```
% repeatedly run sgajane (my genetic algorithm m-file) on objective
% function 5 (maximize the minimum of two normalized eigenvalue shifts
% in the (1,1) mode) while varying the k1:k2 ratio (i.e. the
```

```
% alpha:beta: ratio)

% File created 04/16/08
% by Jane E. Valentine

%create empty matrix to hold results - we will fill it up as we go.

varyphen=[]

% run GA program on objective function
for k1=0:0.1:1.0
  %run sgajane on objective function 5 with k1 varying, k2=1
  phen=sgajane(5, k1, 1);

  %extract 10 best results
  phen2=phen(1:10,:);

  %extract single best result, and append best result to varyphen
  varyphen = [varyphen;phen2(1,:)]

end

%assign names to columns
x1=varyphen(:,2);
x2=varyphen(:,3);
y1=varyphen(:,4);
y2=varyphen(:,5);
val=varyphen(:,1);
```

```
%indexing by k1 value
kcol=[0:0.1:1.0];

%plotting raw results
figure
plot(kcol,-val,'bd-')
xlabel('k1:k2 mass density ratio')
ylabel('objective function value')
title(['Minimum eigenvalue shift'])

%plot results normalized by mass of smaller patch
kcol2=transpose(kcol);

figure
plot(kcol,-val./kcol2,'bd-')
xlabel('k1:k2 mass density ratio')
ylabel('normalized signal strength per unit mass density')
title(['Minimum eigenvalue shift per unit mass'])

% plot tradeoffs between patches: (1,1) mode
mode11_1=zeros(11,3);
mode11_1(:,2)=x1;
mode11_1(:,3)=y1;
for i=1:11
mode11_1(i,1)=Signal11(x1(i),y1(i))/Signal11(70,45);
end
```

```
mode11_2=zeros(11,3);
mode11_2(:,2)=x2;
mode11_2(:,3)=y2;
for i=1:11
mode11_2(i,1)=Signal11(x2(i),y2(i))/Signal11(70,45);
end

figure
plot(kcol,mode11_1(:,1),'bo:',kcol,mode11_2(:,1),'mx:')
xlabel('k1:k2 mass density ratio')
ylabel('Fractional eigenvalue shift in (1,1) mode')
legend('patch 1','patch 2')
axis([-0.10 1.10 -0.10 1.10])
title(['Fractional (1,1) shifts'])
```

Bibliography

[1] F. Aberl, H. Wolf, C. Koblinger, S. Drost, P. Woias, and S. Koch. HIV serology using piezoelectric immunosensors. *Sensors and Actuators B*, 18-19:271–275, 1994.

[2] J.D. Adams, G. Parrott, C. Bauer, T. Sant, L. Manning, M. Jones, B. Rogers, D. McGorkle, and T.L. Ferrell. Nanowatt chemical vapor detection with a self-sensing, piezoelectric microcantilever array. *Applied Physics Letters*, 83(16):3428–3430, 2003.

[3] A.M. Attallah, M.M. Abdel-Aziz, A.M. El-Sayed, and A.A. Tabll. Detection of serum p53 protein in patients with different gastrointestinal cancers. *Cancer Detection and Prevention*, 27(127-131), 2003.

[4] M.K. Baller, H.P. Lang, J. Fritz, C. Gerber, J.K. Gimzewsk, U. Drechsler, H. Rothuizen, M. Despont, P. Vettiger, F.M. Battiston, J.P. Ramseyer, P. Fornaro, E. Meyer, and H.J. Guntherodt. A cantilever array-based artifical nose. *Ultramicroscopy*, 82(1-4):1–9, 2000.

[5] M.J. Bartkovsky. *Development of an Acoustic-Wave Biosensing Device*. PhD thesis, Department of Chemical Engineering, Carnegie Mellon University, Pittsburgh, PA USA, August 2006. Advisors: Todd M. Przybycien and Steinar Hauan.

[6] M.J. Bartkovsky, S. Hauan, and T.M. Przybycien. Photochemical modification of a MEMS membrane device for use as a novel gravimetric based biosensor. Presented at the AIChE annual meeting, Austin TX USA, Advances in Biosensors I (paper 36g, November 2004.

[7] M.J. Bartkovsky, T.M. Przybycien, J.J. Neumann, and Hauan. S. A first generation MEMS membrane based biosensor. Presented at the AIChE annual meeting, paper 385e, November 2003.

[8] L.T. Biegler, I.E. Grossmann, and A.W. Westerberg. *Systematic Methods of Chemical Process Design*. Prentice Hall, Upper Saddle River, NJ 07458, 1997.

[9] M.A. Bos and J.M. Kleijn. Determination of the orientation distribution of adsorbed fluorophores using TIRF. I Theory. *Biophysical Journal*, 68:2566–2572, 1995.

[10] M.A. Bos and J.M. Kleijn. Determination of the orientation distribution of adsorbed fluorophores using TIRF. II Measurements on porphyrin and cytochrome c. *Biophysical Journal*, 68:2573–2579, 1995.

[11] C.L. Britton, Jr., R.L. Jones, P.I. Oden, Z. Hu, R.J. Warmack, S.F. Smith, W.L. Bryan, and J.M. Rochelle. Multiple-input microcantilever sensors. *Ultramicroscopy*, 82(1-4):17–21, 2000.

[12] S. Caser and Hilhorst. Random sequential adsorption of hard discs and squares: Exact bounds for the covering fraction. *Journal of Physics A: Math. Gen.*, 28:3887–3900, 1995.

[13] J.Y. Chang and J.A. Wickert. Response of modulated doublet modes to traveling wave excitation. *Journal of Sound and Vibration*, 242:69–83, 2001.

[14] A.J. Chipperfield, P.J. Fleming, H Pohlheim, and C.M. Fonseca. *Genetic Algorithm Toolbox for use with Matlab*. Department of Automatic Control and Systems Engineering, University of Sheffield, U.K., 1.2 edition, January 1995.

[15] A.J. Clark, L.A. Whitehead, C.A. Haynes, and A. Kotlicki. Novel resonant-frequency sensor to detect the kinetics of protein adsorption. *Review of Scientific Instruments*, 73(12):4339–4346, 2002.

[16] M.A. Cooper. Label-free screening of bio-molecular interactions. *Anal. Bioanal. Chem.*, 337:834–842, 2003.

[17] R. Courant and D. Hilbert. *Methods of Mathematical Physics*, volume VI. John Wiley & Sons, 1953.

[18] V.S.J. Craig and M. Plunkett. Determination of coupled solvent mass in quartz crystal microbalance measurements using deuterated solvents. *Journal of Colloid and Interface Science*, 262:126–129, 2003.

[19] A. D'Amico and C. Di Natale. A contribution on some basic definitions of sensors properties. *IEEE Sensors Journal*, 1(3), 2001.

[20] J.L. Dang, K. Heroux, J. Kearney, A. Arasteh, M. Gostomski, and P.A. Emanuel. Bacillus spore inactivation method affect detection assays. *Applied and Environmental Microbiology*, 67(8):3665–3670, August 2001.

[21] J. Daniels and N. Pourmand. Label-free impedance biosensors: Opportunities and challanges. *Electroanalysis*, 19(12):1239–1257, 2007.

[22] M.J. Dennison and A.P.F. Turner. Biosensors for environmental monitoring. *Biotech. Adv.*, 13:1–12, 1995.

[23] P. D'Orazio. Biosensors in clinical chemistry. *Clinica Chimica Acta*, 334:41–69, 2003.

[24] I. Dufour and L. Fadel. Resonant microcantilever type chemical sensors: analytical modeling in view of optimization. *Sensors and Actuators B*, 91(3):353–361, 2003.

[25] A. Graham and M. Moo-Young. Biosensors: Recent Trends. *Biotech Advs*, 3:209–218, 1985.

[26] J. Grate and J. Klusty. Surface acoustic wave vapor sensors based on resonator devices. *Analytical Chemistry*, 63(17):1719–1727, September 1991.

[27] J. Grate, S. Martin, and R. White. Acoustic wave microsensors: Part I. *Analytical Chemistry*, 65(21):940–948, Nov 1993.

[28] J. Grate, S. Martin, and R. White. Acoustic wave microsensors: Part II. *Analytical Chemistry*, 65(22):987–996, November 1993.

[29] J.W. Grate. Acoustic wave microsensor arrays for vapor sensing. *Chemical Reviews*, 100(7):2627–2648, 2000.

[30] J.W. Grate, S.W. Wenzel, and R.M. White. Flexural plate wave devices for chemical analysis. *Analytical Chemistry*, 63(15):1552–1561, 1991.

[31] G.L. Harding, J. Du, P.R. Dencher, D. Barnett, and E. Howe. Love wave acoustic immunosensor operating in liquid. *Sensors and Actuators A*, 1997.

[32] S. Hauan, T.M. Przybycien, K.J. Gabriel, J.J. Neumann, and M.J. Bartkovsky. A MEMS based biosensor. U.S. Patent Application, 2003. Filed Nov 6th; Serial No. 10/702.709.

[33] A. Hillier and M. Ward. Scanning electrochemical mass sensitivity mapping of the quartz crystal microbalance in liquid media. *Analytical Chemistry*, 64(21), November 1992.

[34] B. Ilic, D. Czaplewski, M. Zalalutdinov, and G. Craighead. Single cell detection with micromechanical oscillators. *Journal of Vacuum Science Technology B*, 19(6):2825–2828, Nov/Dec 2001.

[35] The Mathworks Inc. FEMLAB - Finite Element Modelling LABoratory. http://www.femlab.com.

[36] The Mathworks Inc. MATLAB – MATrix LABoratory. http://www.mathworks.com.

[37] J. Janata and R. Huber, editors. *Solid State Chemical Sensors*. Academic Press, Inc, 1985.

[38] J. Janin, S. Miller, and C. Chothia. Surface, subunit interfaces and interior of oligomeric proteins. *J. Mol. Biol.*, 204(1):155–164, 1988.

[39] M. Keusgen. Biosensors: new approaches in drug discovery. *Naturwissenschaften*, 89:433–444, 2002.

[40] R.S. Laugeson. Eigenvalues of the laplacian on inhomogeneous membranes. *Am. J. Math.*, 120(2):305–344, 1998.

[41] P. Leonard, S. Hearty, J. Brennan, L. Dunne, J. Quinn, T. Chakraborty, and R. O'Kennedy. Advances in biosensors for detection of pathogens in food and water. *Enzyme and Microbial Technology*, 32:3–13, 2003.

[42] P. B. Luppa, L. J. Sokoll, and D. W. Chan. Immunosensors - principles and applications to clinical chemistry. *Clinica Chimica Acta*, 314(1-26), 2001.

[43] N. Maluf. *An Introduction to Microelectromechanical Systems Engineering.* Artech House, Inc., Boston, MA., 2000.

[44] A.W. Martinez, S.T. Phillips, E. Carrilho, S.W. Thomas III, H. Sindi, and G.M. Whitesides. Simple telemedicine for developing regions: camera phones and paper-based microfluidic devices for real-time, off-site diagnosis. *Anal. Chem*, 80(10):3699–3707, 2008.

[45] A.W. Martinez, S.T. Phillips, and G.M. Whitesides. Three-dimensional microfluidic devices fabricated in layered paper and tape. *Proceedings of the National Academy of Sciences*, 105(50):19606, 2008.

[46] C.M. McCarthy. Recovery of a density from the eigenvalues of a nonhomogeneous membrane. In *Proceedings of Inverse Problems in Engineerings: Theory and Practice*, 1999.

[47] R. McKendry, J. Zhang, Y. Arntz, T. Strunz, M. Hegner, H.P. Lang, M.K. Baller, U. Certa, E. Meyer, H. Guntherodt, and C. Gerber. Multiple label-free biodetection and quantitative DNA-binding assays on a nanomechanical cantilever array. *PNAS*, 99(15):9783–9788, 2002.

[48] M. A. Meineke and J. D. Gezelter. Random sequential adsorption model for the differential coverage of gold (111) surfaces by two related silicon phthalocyanines. *American Chemical Society*, 105(28):6515–6519, 2001.

[49] S. Miller-Renaud, D. Dupont, and P. Dulieu. Quantification of β-Casein in milk and cheese using optical immunosensor. *Journal of Agricultural and Food Chemistry*, 52(659-664), 2004.

[50] J. Murdock. *Perturbations: Theory and Methods.* John Wiley & Sons, 1991.

[51] M. Napoli, B. Bamieh, and K. Turner. Mathematical modeling, experimental validation and observer design for a capacitively actuated microcantilever. In *Proceedings of the American Control Conference*, pages 3732–3737, 2003.

[52] J.J. Neumann and K.J. Gabriel. CMOS-MEMS Membrane for audio-frequency acoustic actuation. *Sensors and Actuators A*, 95:175–182, 2002.

[53] P.I. Oden. Gravimetric sensing of metallic despoits using an end-loaded microfabricated beam structure. *Sensors and Actuators B*, 53(3):191–196, December 1998.

[54] Y. Osada and D.E. DeRossi, editors. *Polymer Sensors and Actuators*. Springer, 2000.

[55] C.P. Quinn, V.A. Semenova, C.M. Elie, S. Romero-Steiner, C. Greene, H. Li, K. Stamey, E. Steward-Clark, and D.S. Schmidt. Specific, sensitive, and quantitative Enzyme-Linked Immunosorbent Assay for human immunoglobin G: Antibodies to anthrax toxin protective antigen. *Emerging Infectious Diseases*, 8(10):1103–1110, October 2002.

[56] J.W.S. Rayleigh. *The Theory of Sound*. Dover Publications Inc., 2nd edition, 1945.

[57] Wolfram Research. Mathematica. http://www.wolfram.com/mathematica/.

[58] K. R. Rogers. Principles of Affinity-Based Biosensors. *Molecular Biotechnology*, 14:109–129, 2000.

[59] C.A. Rowe, L.M. Tender, M.J. Feldstein, J.P. Golden, S.B. Scruggs, B.D. MacCraith, J.J. Cras, and F.S. Ligler. Array biosensor for simultaneous identification of bacterial, viral, and protein arrays. *Analytical Chemistry*, 71(17):3846–3852, September 1999.

[60] T. Sakai, K. Shinahara, A. Torimaru, H. Tanaka, Y. Shoyama, and K. Matsumoto. Sensitive detection of Glcyrrhizin and evaluation of the affinity constants by surface plasmon resonance-based immunosenor. *Analytical Sciences*, 20:279–283, February 2004.

[61] Andrea Saltelli, Marco Ratto, Terry Andres, Francesca Campolongo, Jessica Cariboni, Debora Gatelli, Michaela Saisana, and Stefano Tarantola. *Global sensitivity analysis: the primer*. Wiley-Interscience, 2008.

[62] R.F. Schmitt, J.W. Allen, J.F. Vetelino, J. Parks, and C. Zhang. Bulk acoustic wave modes in quartz for sensing measurand induced mechanical and electrical property changes. *Sensors and Actuators B*, 76(1-3):95–102, June 2001.

[63] M. Sepaniak, P. Datskos, N. Lavrik, and C. Tipple. Microcantilever transducers: A new approach in sensor technology. *Analytical Chemistry*, 74(21):568–575, 2002.

[64] J.D. Sternhagen, C.E. Wold, W.A. Kempf, M. Karlgaard, K.D. Mitzner, R.D. Mileham, and D.W. Galipeau. A novel integrated acoustic gas and temperature sensor. *IEEE Sensors J.*, 2(4):301–306, August 2002.

[65] M.W. Steward. *Antibodies: Their structure and function*. Chapman and Hall, 1984.

[66] JW Stoop, BJM Zegers, PC Sander, and RE Ballieux. Serum immunoglobulin levels in healthy children and adults. *Clinical and experimental immunology*, 4(1):101, 1969.

[67] S. Suwansa-ard, Kanatharana P., P. Asawatreratanakul, C. Limsakul, B. Wongkittisuksa, and P. Thavarungkul. Semi disposable reactor biosensors for detecting carbamate pesticides in water. *Biosensors and Bioelectronics*, 21:445–454, 2005.

[68] C.R. Taitt, G.P. Anderson, B.M. Lingerfelt, M.J. Feldstein, and F.S. Ligler. Nine-analyte detection using an array-based biosensor. *Analytical Chemistry*, 74(23):6114–6120, December 2002.

[69] T. Thundat, P.I. Oden, and R.J. Warmack. Microcantilever sensors. *Microscale Thermophyscial Engineering*, pages 185–199, 1997.

[70] S. Timoshenko, D.H. Young, and W. Weaver. *Vibration Problems in Engineering, Fourth Edition*. John Wiley & Sons, 1974.

[71] J. Tschmelak, G. Proll, and G. Gauglitz. Immunosensor for estrone with an equal limit of detection as common analytical methods. *Anal. Bioanal. Chem.*, 378:744–745, 2004.

[72] A. Turner. Biosensors - Sense and Sensitivity. *Science*, 290(5495):1315–1317, November 2000.

[73] J.E. Valentine, T.M. Przybycien, and S. Hauan. Modeling and design of a MEMS-based biosensor. Presented at the AIChE annual meeting, paper 197h, November 2003.

[74] J.E. Valentine, T.M. Przybycien, and S. Hauan. Response surface determination of a multi-target MEMS sensor. Presented at the AIChE annual meeting, paper 509f, November 2004.

[75] J.E. Valentine, T.M. Przybycien, and S. Hauan. Design of acoustic-wave biochemical sensors using MEMS. *Journal of Applied Physics*, April 2007.

[76] G. Vendantham, H.G. Sparks, E.U. Sane, S. Tzannis, and T. Przybycien. A holistic approach for protein secondary structure estimation from infrared spectra in H2O solutions. *Analytical Biochemistry*, 285:33–49, 2000.

[77] J.R. Vig and A. Ballato. Comments about the effects of nonuniform mass loading on a Quartz Crystal Microbalance. *IEEE Transactions on Ultrasonics, Ferroelectrics, and Frequency Control*, 45(5), 1998.

[78] A.W. Wang, R. Kiwan, R.M. White, and R.L. Ceriani. A silicon-based ultrasonic immunoassay for detection of breast cancer antigens. *Sensors and Actuators B*, pages 13–21, 1998.

[79] H. Wang, H. Zeng, Z. Liu, Y. Yang, T. Deng, G. Shen, and R. Yu. Immunophenotyping of acute leukemia using an integrated piezoelectric immunosensor array. *Analytical Chemistry*, 76(2203-2209), 2004.

[80] K.Y. Wee, G.Y. Kang, J. Park, J.Y. Kang, D.S. Yoon, J.H. Park, and T.S. Kim. Novel electrical detection of label-free disease marker proteins using piezoresistive self-sensing micro-cantilevers. *Biosensors and Bioelectronics*, 20:1932–1938, 2005.

[81] S. Wenzel and R. White. Analytic comparison of the sensitivities of bulk-wave, surface-wave, and flexural plate-wave ultrasonic gravimetric sensors. *Applied Physics Letters*, 54(20):1976–1978, May 1989.

[82] S. Wenzel and R. White. Flexural plate-wave gravimetric chemical sensor. *Sensors and Actuators A*, 22(1-3):700–703, June 1989.

[83] R. White. Acoustic sensors for physical, chemical and biochemical applications. *IEEE International Frequency Control Symposium*, pages 587–594, 1998.

[84] J.A. Wickert. Discussions about perturbation analysis derivation. Personal commuication, Sep-Nov 2004.

[85] D.M. Wilson and S.D. Garrod. Optimization of gas-sensitive polymer arrays using combinations of heterogeneous and homogeneous subarrays. *IEEE Sensors Journal*, 2(3):169–178, 2002.

[86] H. Wohltjen. Mechanism of operation and design considerations for surface acoustic wave device vapour sensors. *Sensors and Actuators*, 5(4):307–325, July 1984.

[87] R. Wong and H Tse, editors. *Lateral Flow Immunoassay*. Humana Press, 2009.

[88] NASA's Hubble telescope site. http://hubblesite.org.

[89] q-sense. http://www.q-sense.com. Commercial gravimetric quartz crystal microbalance system.

[90] J. Zhao, X. Zhang, C.R. Yonzon, A.J. Haes, and R.P. Van Duyne. Localized surface plasmon resonance biosensors. *Nanomedicine*, 2006.