Analyte Sensitive Ultrasound Contrast Agents Based On Molecularly Imprinted Nanogel Sensors

David Troïani

Ph. D. Dissertation

Department of Chemistry

McGill University

Montréal, Québec, Canada

August 11, 2011

A dissertation submitted to McGill University in partial fulfillment of the requirements of the degree of Doctor of Philosophy

Abstract

Recently developed ultrasound contrast agents provide traditional ultrasound techniques with highly localized contrast within samples depending on contrast agent concentration. Contrast agents resonate at characteristic frequencies, allowing background signals in backscatter collected from samples to be easily removed through filtration, leaving only resonance from contrast agents remaining. The goal of this thesis was to develop analyte sensitive contrast agents based on molecularly imprinted poly(N-isopropylacrylamide) (pNIPA) nanogel polymers. Molecularly imprinted pNIPA was synthesized in presence of target analyte theophylline. Ultrasonic analysis of pNIPA behavior in the presence of varying theophylline concentration revealed amplitude changes at various frequencies. Analysis of chemically similar caffeine demonstrated ultrasonic changes at different frequencies.

Solutions containing increasing amounts of theophylline in the 8.4 to 167 μM range with 1% by weight molecularly imprinted pNIPA in water were analysed ultrasonically. Concentration models displayed very high linearity (r^2 coefficient exceeding 0.99). Additional concentration models were constructed in a matrix of solutions containing both the imprinted analyte theophylline, and interferant caffeine. Regression models for the two analytes demonstrated good linearity in the micromolar range (r^2 of 0.98 for theophylline, 0.87 for caffeine) using different subsets of frequencies for each analyte.

A tighter binding arrangement between analyte and pNIPA was achieved through synthesis of molecularly imprinted pNIPA in the collapsed phase. This increased analyte sensitivity and linear range to nanomolar concentrations. Quantification assays were carried out on a dopamine oxidation product, (5-6-dihydroxyindole, DHI), from 16.7 to 163 nM. High linearity was obtained (r^2

correlation coefficient exceeding 0.99). The experiment was repeated in a presence of albumin, a biologically relevant interferant, with good agreement between actual and estimated concentration.

Multi-analyte quantification was improved by combining two differently imprinted pNIPA nanogels to form multiplexed nanogels. Simultaneous quantification assays were carried out for theophylline (8.4 to 49 uM) and DHI (48.8 to 176 nM). Good linearity between estimated and actual concentrations were obtained (r^2 of 0.99 for DHI, 0.96 for theophylline).

Determination of a larger analyte, tobacco mosaic virus (TMV), was also carried out. Concentration models in the 9 to 140 ppb range showed excellent linearity (correlation coefficients exceeding 0.99). The process was repeated in presence of another virus, tomato bushy stunt virus (TBSV), acting as an interferant. Similar linearity was obtained.

Critical points of the ultrasound quantification system based on molecularly imprinted nanogels are summarized in the Conclusion chapter. Improvements focusing on obtaining stronger ultrasonic signals in aforementioned analyses are discussed in the Future Works section.

Résumé

Le développement des agents de contraste ultrasonique ont produit des agents qui fournissent du contraste très élevé et localisé dans l'échantillon, selon la concentration des agents. Ces agents résonnent avec des fréquences particulières, et facilitent la soustraction du bruit par filtration, qui laissent seulement le résonance des agents. Le but de cette thèse est de développer des agents de contraste capable de quantifier des analytes en utilisant des polymères poly(N-isopropylacrylamide) (pNIPA) imprimés avec des molécules cible comme point de départ. La synthèse des polymères pNIPA était faite en présence de la molécule théophylline. L'analyse ultrasonique de pNIPA en présence de differentes concentration de théophylline conduit à des changements d'amplitudes à plusieurs fréquences. L'analyse de pNIPA avec de la caféine a produit des changements à d'autres fréquences.

Des solutions avec théophylline (8.4 à 167 μM) et 1% massique de pNIPA imprimé avec théophylline ont été analysées avec le système ultrasonique. Les modèles de concentration que cette analyse a produit ont un très haut niveau de linéarité (r^2 plus que 0.99). Plusieurs modèles de concentration ont été construit avec une matrice de solution qui contenait théophylline et caféine. Les modèles de régression pour les deux analytes ont démontrés de la linéarité dans les concentrations micromolar (r^2 de 0.98 pour théophylline, 0.87 pour caféine), utilisant différentes fréquences pour chaque analyte.

Une affinité plus grande entre l'analyte et pNIPA a été gràce à la synthèse avec du pNIPA comprimé. Cela a élevé la sensibilité aux concentrations nanomolaires. Des quantifications d'un produit d'oxidation de la dopamine (5-6-dihydroxyindole, DHI) de 16.7 à 163 nM ont été faites. Les résultats ont démontrés une très bonne linéarité

(r^2 plus que 0.99). L'expérience a été reproduite avec de l'albumine dans chaque solution, encore avec des bons résultats.

La quantification de plusieurs analytes simultanément a été améliorée avec la combinaison de deux nanogels pNIPA imprimés avec les différentes analytes. Les analyses de concentration ont été menées en parallèle pour la théophylline (8.4 à 49 uM) et le DHI (48.8 à 176 nM). Les concentrations estimées étaient en accord avec les concentrations actuelles (r^2 de 0.99 pour DHI, 0.96 pour théophylline).

Un analyte plus large, le virus tabac mosaïque (TMV), a été detecté avec le système ultrasonique. Des modèles de concentration de 9 à 140 ppb ont démontrés un très haut niveau de linéarité (r^2 plus que 0.99). Le processus a été répeté avec un autre virus, tomate bushy stunt (TBSV), ajouté aux solutions de TMV avec des résultats semblables.

Les aspects critiques du système de quantification ultrasonique sont récapitulés dans le chapitre de conclusion. Des améliorations pour obtenir des signaux plus fort dans les analyses ultrasoniques sont discutés dans la section futur.

Table of Contents

List of Figures

List of Tables

List of Commonly Used Symbols and Abbreviations

Z	Acoustic Impedance
ΔF	Free Energy (per unit volume)
Φ	Polymer Volume Fraction
T	Temperature
T_c	Critical Phase Transition Threshold Temperature
y	Dependant Variable (concentration)
x_n	Independant Variable (amplitude at select frequencies)
b_n	Slope Coefficient (corresponding to associated frequencies)
f_0	Contrast Agent Resonance Frequency
a_e	Equilibrium Particle Radius
ρ_L	Density of Surrounding Liquid
p_0	Atmospheric Pressure
G_s	Shell Shear Modulus
d_{Se}	Shell Thickness
K	Bulk Modulus
pNIPA	poly(N-isopropylacrylamide)
HPC	Hydroxypropylcellulose
MIP	Molecularly Imprinted Polymer
DHI	5,1-dihydroxyindole
TMV	Tobacco Mosaic Virus
TBSV	Tomato Bushy Stunt Virus
TSM	Thickness Shear Mode
QCM	Quartz Crystal Microbalance

Original Contributions to Knowledge

1. Development of a novel analyte quantification system based on changes in ultrasonic properties at multiple frequencies of molecularly imprinted pNIPA nanogel in the presence of varying analyte concentrations.

2. Development of a multiplexed, analyte sensitive nanogel contrast agent able to simultaneously quantify multiple analytes from a mixture of two differently imprinted nanogel contrast agents.

3. Development of novel ultrasound analyte quantification system to quantify supramolecular structures such as viruses, and to simultaneously quantify several molecular analytes using a mixture of differently imprinted pNIPA nanogels.

Contribution of Authors

The following list provides details regarding papers in this dissertation and contributions of each author. Dr. David H. Burns was primary reviewer and editor for each paper, and dissertation supervisor for David Troïani.

Chapter 3

David Troïani, Jonathan R. Dion, and David H. Burns, "Ultrasonic Quantitation using Smart Biosensors", *Talanta, Volume 83, Issue 5, February 2011, Pages 1371-1375.*

David Troïani developed the synthesis and imprinting procedure for the high affinity pNIPA nanogel biosensors, prepared pNIPA nanogel biosensors, and constructed the sample used for ultrasound data acquisition of pNIPA biosensors. David Troïani also performed all data acquisition and analysis related to pNIPA nanogel biosensors. Jonathan R. Dion developed the synthesis and imprinting procedure for the low affinity HPC nanogel biosensors, prepared HPC nanogel biosensors and constructed the sample cell used for ultrasound data acquisition of HPC biosensors. Jonathan R. Dion also performed all data acquisition and analysis related to HPC nanogel biosensors. David Troïani and Jonathan R. Dion developed the theoretical model relating changes in ultrasonic frequency profile to nanogel physical properties. The paper was written and prepared for publication by David Troïani and Jonathan R. Dion, and edited by David H. Burns.

Chapter 4

David Troïani and David H. Burns, "Quantification of Dopamine Oxidation Product using Smart Ultrasound Nanogel Contrast Agents", *to be submitted.*

David Troïani developed the synthesis and imprinting procedure for, and prepared the 5,1-dihydroxyindole (DHI) imprinted pNIPA nanogel contrast agents.

David Troïani also constructed the sample cell for ultrasonic data acquisition and performed all data acquisition and processing. The paper was written and prepared for publication by David Troïani and edited by David H. Burns.

Chapter 5

David Troïani and David H. Burns, "Simultaneous Ultrasonic Analyte Quantification using Multiplexed Nanogel Contrast Agents", *to be submitted.*

David Troïani developed the synthesis and imprinting procedures for the 5,1-dihydroxyindole (DHI) imprinted pNIPA nanogel, and the theophylline imprinted pNIPA nanogel. David Troïani also prepared both types of imprinted pNIPA nanogel, and performed all data acquisition and processing. The paper was written and prepared for publication by David Troïani and edited by David H. Burns.

Chapter 6

David Troïani and David H. Burns, "Tobacco Mosaic Virus Quantification Using Ultrasound and Virus Sensitive Nanogels", *to be submitted.*

David Troïani developed the synthesis and imprinting procedures for, and prepared the tobacco mosaic virus (TMV) sensitive pNIPA nanogels. David Troïani performed all data acquisition and processing. The paper was written and prepared for publication by David Troïani and edited by David H. Burns.

Copyright Waiver

A copyright waiver allowing the inclusion of the previously published manuscript titled "Ultrasonic Quantification Using Smart Nanogel Sensors" in this dissertation was electronically obtained through the Copyright Clearance Center Inc., representing Elsevier Limited. The copyright waiver is detailed below.

This is a License Agreement between David Troiani ("You") and Elsevier ("Elsevier") provided by Copyright Clearance Center ("CCC"). The license consists of your order details, the terms and conditions provided by Elsevier, and the payment terms and conditions.

Supplier:	Elsevier Limited The Boulevard, Langford Lane, Kidlington Oxford, OX5 1GB, UK Registered Company Number 1982084
License Number:	2680730980081
License Date:	Jun 02, 2011
Licensed Content Publisher:	Elsevier
Licensed Content Publication:	Talanta
Licensed Content Title:	Ultrasonic Quantification Using Smart Nanogel Sensors
Licensed Content Authors:	David Troïani, Jonathan R. Dion, David H. Burns
Licensed Content Date:	15 February 2011
Licensed Content Volume:	Number 83

Licensed Content Issue:	Number 5
Number of Pages:	5, Start Page 1371, End Page 1375
Type of Use:	Reuse in a Thesis/Dissertation
Portion:	Full Article
Format:	Both Print and Electronic

Are you the author of this Elsevier article? Yes

Will you be translating? No

Order reference number:

Title of your thesis/dissertation:	Analyte Sensitive Ultrasound Contrast Agents Based On Molecularly Imprinted Nanogel Sensors
Expected Completion Date:	Dec 2011
Estimated Size (Number of Pages):	250

Acknowledgements

I thank the following members of the David Burns lab group for their advice, encouragement and great senses of humor (in no particular order): Dirk Bandilla, Jonathan Dion, Fabiano Pandozzi, Francis Esmonde-White, Lucy Botros, Shing Kwok, Kristin Power and our italian exchange student Alberto Bonomi. Their positivity and humorous catch-phrases made the laboratory a pleasant environment to work in.

I also thank the chemistry department's machining division for their high standards for precision and accuracy: Bill Bastian, Fred Kluck, and Jean-Phillipe Guay. In addition, I thank the chemistry department's graduate student coordinator Chantal Marotte for her patience and treating each graduate student as a long-time friend, the department chair's secretary Sandra Aerssen for her hard work, and genuinely being interested in the well being of graduate students, and department secretaries Fay Nurse and Alison McCaffrey, for their hard work and friendly nature.

Lastly, I thank my laboratory supervisor Dr. David Burns, whose creative thinking, understanding, level headedness, and uplifting meetings greatly aided this thesis.

Chapter 1.

Research Objectives and Dissertation Overview

1.1 Research Objectives

Molecular quantification in point-of-care settings can be achieved through a variety of well established instruments and methods. Instruments use different approaches to quantify biologically relevant analytes, some relying on separation mechanisms (lateral flow chromatography, HPLC), others on absorption and emission (fluorescence, NIR absorbance). Detection of a single target analyte in ideal circumstances often generates appreciable detection limits with relatively high signal. Challenges arise when analysing multi-analyte samples that contain overlapping analyte signatures, which is often the case in biological samples. Another issue stems from instruments that use optical measurement. Low penetration depth of optical sources results in difficulties when analysing samples with multiple boundaries between analyte and detector. Although these issues can sometimes be overcome using separation techniques, centrifuging or data processing techniques, addition of such procedures extends analysis time and adds layers of unwanted complexity. Analysis of samples exhibiting aforementioned issues might more easily be done acoustically, using amplitude of characteristic frequencies in analyte signatures to determine concentration.

The advantages of acoustic techniques over traditional instrument methods come in two forms. Acoustic signals provide added penetration depth with lower signal loss across boundaries, making multiple sample layers less of an issue. The second advantage offers the capability to quantify analytes having similar optical signatures. However, molecules by themselves do not respond uniquely to acoustic perturbation for quantification purposes. Additional mechanisms are required to accurately quantify analytes and separate relevant acoustics from background signals. A prime example of acoustic signal separation is the use of ultrasound contrast agents. When ultrasonically perturbed, these contrast agents resonate at a sharp frequency and can be easily filtered from background signals, providing increased signal, edge detection and resolution. The resonance in contrast agents is similar to current analytical sensors using ultrasound detection. One example of ultrasonic quantification is the thickness shear mode (TSM) resonator. TSMs are vibrating quartz crystals which shift in resonance frequency according to analyte concentration in gaseous or liquid media. These systems are discussed in greater depth in Chapter 2.

Compared to optical methods, quantitative ultrasound is a field that is still in an early stage of development. Improvements in analyte sensitivity of current acoustic analyte quantification mechanisms would provide greater applicability. Acoustic quantification could provide improved alternatives to optical measurements, especially in biological fields.

The goal of this research was to provide a novel approach to acoustic analyte quantification. This approach is based on using ultrasound with analyte sensitive contrast agents composed of molecularly imprinted nanogels. The main objective was to develop this technique to quantify analytes by recording ultrasonic changes in molecularly imprinted nanogels in presence of varying analyte concentration. Clinically relevant analytes with biological interferents were selected for quantification assays. Various areas of this novel ultrasonic quantification system were modified and advanced to provide optimal analyte analysis under several conditions.

1.2 Dissertation Overview and Layout

The dissertation begins with an introductory look at bioanalyte quantification methods for point-of-care analysis of the respiratory system in Chapter 2. While a thesis of this scope can be related to many point-of-care systems, this thesis focuses on respiration. The respiratory system is essential for all life and in the human body is closely interrelated with other vital areas, including circulatory and cardiovascular systems. Plants also have a complex respiratory system, essential for life and continued growth. Several quantitative instruments and methods for bioanalytes required in proper respiratory function in both the human body and plants are discussed. Current ultrasound methods that quantify respiratory bioanalytes are also examined in detail, giving perspective on the field of quantitative ultrasonics.

The thesis continues with the development of nanogel contrast agents and characterization of ultrasonic properties, discussed in Chapter 3. Established synthesis procedures were modified in order to produce nano-sized hydrogels or nanogels, more suited for this application, in contrast to semi-solid nanogels prominently found in literature. Sample cells were designed and constructed to facilitate ultrasonic characterization, with data processing techniques programmed to compute ultrasound frequency profiles.

The next major challenge in this thesis dealt with devising a mechanism to impart changes in ultrasonic profile of nanogels in presence of analyte, explained in Chapter 3. It had already been shown that docking of analyte into molecularly imprinted pockets causes measurement changes in nanogel properties. These changes could also alter ultrasonic profile of nanogels. Existing molecular imprinting procedures for nanogels were adapted and imprinted nanogels were synthesized. Synthesis lead to soluble molecularly imprinted poly(N-isopropylacrylamide) copolymers, or simply pNIPA nanogels. Analyte quantification assays were carried out in the micromolar range on theophylline, an analyte used in treatment of severe respiratory illnesses.

Synthesis of the imprinted pNIPA nanogel in the condensed phase was explored in Chapter 4 to achieve tighter binding to analyte for greater sensitivity. The analyte 5,6-dihydroxyindole (DHI), an oxidation product of dopamine, was analysed in this chapter. The imprinting and synthesis processes of DHI imprinted pNIPA was adapted from procedures established in previous chapters.

The ultrasonic analyte quantification technique developed in previous chapters was expanded to include simultaneous, multi-analyte analysis in Chapter 5. Two differently imprinted and differently sized nanogels were combined to form a multiplexed nanogel, suitable for simultaneous multi-analyte quantification. Assays of multiplexed nanogel imprinted for theophylline and DHI were prepared, and ultrasonically examined using aforementioned techniques. Quantification of both analytes was conducted in the nano to micromolar range, with the unique ultrasonic signature of each analyte used to construct separate concentration models for both analytes.

Extending the ultrasonic quantification system to supramolecular analytes was achieved in Chapter 6. The imprinting process was modified to accomodate tobacco mosaic virus (TMV), a much larger analyte, with quantification assays conducted in the parts-per-billion range. Detection limits obtained for TMV were comparable to much lengthier and more involved ELISA (enzyme-linked immunoassay) techniques.

Results from quantification assays for the aforementioned analytes are found in the conclusion section of Chapter 7. A summary of the theory behind ultrasonic analyte quantification is also present in this section, as well as the overall effect of altering quantification parameters including synthesis phase, particle size, percent solution. Chapter 7.2 discusses areas of interest regarding the continuation of this thesis. Several topics focused around increasing signal strength during quantification are covered here. This section also includes avenues for synthesizing gas filled imprinted pNIPA contrast agent microbubbles, as opposed to contrast agent nanogel particles prominently found in this thesis.

Chapter 2.

Introduction to Respiratory Bioprocess Monitoring Instruments and Methods

2.1 Respiratory System Bioprocess Monitoring

2.1.1 Respiratory System in the Human Body

Respiration is essential to life, and a healthy respiratory system is not only required for normal bodily function, it is closely related with many other vital areas. Many cardiovascular diseases manifest symptoms that hamper respiration, resulting in troubled breathing, excessive coughing, inefficient reoxygenation of blood and water buildup in lung tissue. Conversely, respiratory illnesses such as asthma often directly affect cardiovascular health, raising probability of cardiovascular problems. A simplified caricature of the respiratory system, and the affect some therapeutic compounds and afflictions can have on respiration is shown in Figure 2.1.

Oxygenation of hemoglobin in blood through alveolar regions of the lungs represents the respiratory system at its most basic level, with carbon dioxide and other metabolic wastes exchanged as a result and removed from circulation. Oxygenated hemoglobin releases oxygen to other areas of the body, providing energy through various metabolic pathways necessary to sustain life. Once oxyhemoglobin reaches an energy deficient area of the body, the oxygen it transports is used to create stored energy via a system known as aerobic respiration. Organelles (organized structures found within a living cell) known as mitochondria use the aerobic respiratory system in tandem with oxygen released by oxyhemoglobin to form many adenosine triphosphate (ATP) molecules, which is the body's main source of stored energy.

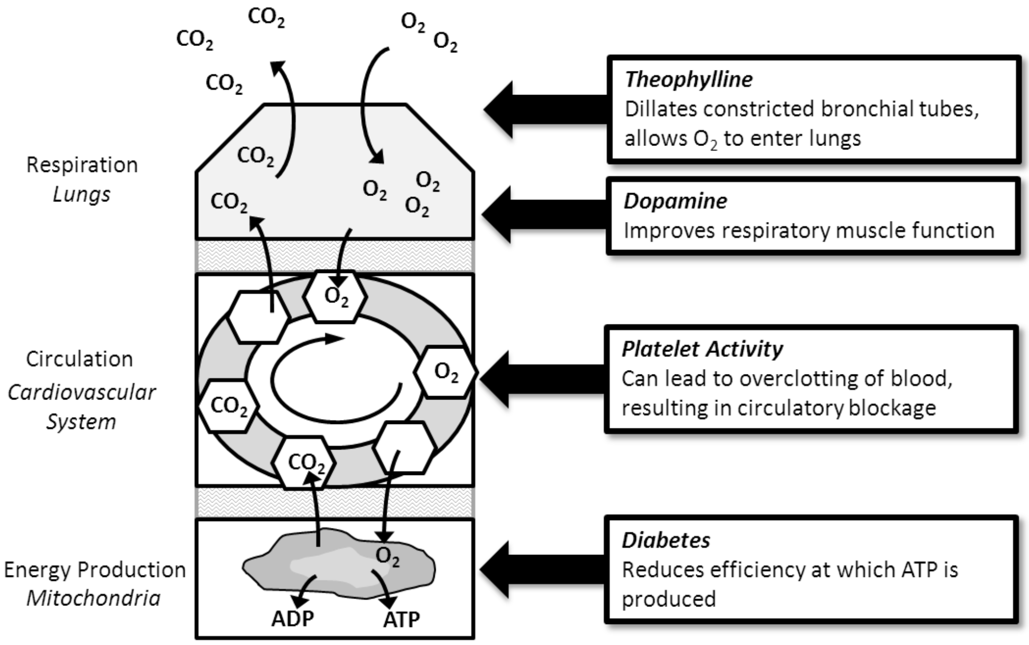
CO2
CO2
CO2
O2
O2
Respiration
Lungs
CO2
CO2
O2
O2
O2
Circulation
Cardiovascular
System
O2
O2
CO2
CO2
Energy Production
Mitochondria
O2
ADP
ATP
Theophylline
Dillates constricted bronchial tubes, allows O2 to enter lungs
Dopamine
Improves respiratory muscle function
Platelet Activity
Can lead to overclotting of blood, resulting in circulatory blockage
Diabetes
Reduces efficiency at which ATP is produced

Figure 2.1. A simplified picture of the respiratory system. The affect some therapeutic compounds, such as theophylline and dopamine, on respiration is described beside the appropriate area. Platelet activity and diabetes affect circulation and energy production respectively, and are also noted in the figure.

Carbon dioxide (CO_2) is a common byproduct of this system, produced as metabolic waste of respiration. Newly deoxygenated hemoglobin, or deoxyhemoglobin, is used as one of the body's methods of CO_2 removal from the bloodstream. Deoxyhemoglobin transports CO_2 back to the lungs to be exhaled. Since CO_2 is more easily dissolved in deoxyhemoglobin, it does not compete with oxygen to bind with the iron ion. However, certain molecules, such as carbon monoxide (CO), do in fact compete with oxygen to bind with the same heme group producing fatal results, as CO cannot be used to produce energy. Since all bodily systems depend, in some shape or form, on oxygen received through oxyhemoglobin, proper respiratory function is crucial in maintaining normal organ function and is essential for disease prevention.

Several afflictions, such as increased platelet activity and diabetes explained in Figure 2.1, can have drastic negative affects on respiration. As a result, accurate monitoring of respiratory bioprocesses through key bioanalytes found in blood critical to gauge the health of many bodily systems.

Clinical applications of respiratory monitoring requires high standards to be effective. Rapid and simple data acquisition, quick result turn-around time (less than ten minutes), maximum portability and accuracy similar or superseding much larger instruments are some challenges facing applications in bioprocess monitoring.[1,2,3] Reproducibility and low manufacturing cost also highlight other areas of significant importance, especially for disposable and single-use quantitative testing kits. Such disposable devices are often considered complete and independent instruments, with self-contained sample acquisition or sample flowthrough, dry reagent (if required), detector, and readout. Challenge regarding samples primarily deals with varying levels of dilution and complex matrices, such as whole blood. These particular challenges demand analytical systems with high tolerance for matrix effects. Precision demands for these devices is quite high, and can be considered similar to requiring multiple full-sized instruments to produce identical readout for one sample, a feat not easily accomplished.

Bioprocess monitoring in whole blood is most useful for diagnosing patients. In many cases, accurate quantification is crucial for point-of-care analysis. Erroneous results or false positives can lead to misdiagnosis of symptoms, resulting in treatment for the wrong condition or no treatment at all. In a similar fashion, plant respiration is also critical, with healthy plants producing higher crop yields. Detailed explanation into plant respiration in the following section stresses the importance of respiratory bioprocess monitoring to maintain high crop yields.

2.1.2 Plant Respiratory System

Respiratory bioprocess monitoring can be applied to plant respiration in similar fashion than for the body's respiratory system. Monitoring key analytes that affect to plant respiration can provide insight into overall crop health, and help diagnose disease and infections that reduce crop yield. The respiratory system of most plants is equal in complexity to the human system, however functions in a completely different manner.

Respiration in plants is achieved through photosynthesis, which at the simplest level is formation of organics and oxygen using carbon dioxide, water, glucose and light as reagents. Although various species of plants process these reagents differently, certain characteristics remain constant. One common feature is that photosynthesis will start with the absorption of light energy through chloroplasts, organelles found in plant cells. These organelles contain chlorophyll which are necessary to obtain energy through sunlight, and incidentally also give foiliage its green color. This energy is used in creating complex organic molecules from carbon dioxide and water absorbed through stomata, tiny pores in plant leaves. The Calvin cycle, a set a reaction, is employed to product organics in the form of $(CH_2O)_n$, the most common being a glucose compound in the form of $C_6H_{12}O_6$. Glucose is used to produce ATP through cellular respiration, and supplies most basic energy needs of plant cells, often is enough to last through nighttime. Excess glucose produced by the Calvin cycle is converted into a polysaccharide amylose (starch), and amylopectin. These products fulfill other plant needs, such as building additional cell walls for growth and increased surface area for greater absorption of natural resources. An intriguing aspect of photosynthesis is it allows plants to create their own sustenance directly from naturally occuring sources, equivalent to having human beings biologically synthesize their own food from air and water. However, as with the human body, disease and infection can reduce a plants ability to breath, reducing absorption of natural resources and consequently plant lifespan.

Examples of threats are given in the the nicotiana species of plants, which are cultivated for a variety of uses including as a natural pesticide and for hepatoprotection (preventing damage to the liver).[4,5] Nicotiana species are susceptible to a commonly occuring infection that can drastically reduces crop yield. The infection is caused by tobacco mosaic virus (TMV) and produces various symptoms, the primary of which is a "mosaic" yellow and brown discoloration of tobacco plant leaves. TMV hampers a plant's respiration by affecting chlorophyll production, causing plants to become chlorotic. Infection will eventually lead to stunted growth, as plants infected with TMV cannot produce enough chlorophyll to absorb natural resources required to grow. Similar to with the human body, monitoring key analytes in nicotiana plants, such as chlorophyll or TMV itself, can be used to diagnose infection.[6] The need to develop methods to quantify TMV and other respiratory analytes resulted in a variety of analytical techniques.

Instrument methods for detection of respiratory bioanalytes are covered in the next section. Methods based on separation chiefly centered around HPLC are investigated first followed by optical and electrochemical techniques, which can be used independantly or coupled with separation methods. Current acoustic methods for analyte quantification are discussed at the end of this chapter.

2.2 High Performance Liquid Chromatography

Matrix effects from the multitude of components found in whole blood often prevents molecular quantification in several instruments. High performance liquid chromatography (HPLC) on serum extracted from whole blood provides a reliable clinical tool for simultaneous bioanalyte point-of-care quantification. Although not a real time system, HPLC can simultaneously quantify several key bioanalytes such as theophylline, 3-methylxanthine, 1-methyluric acid, and caffeine in serum, with detection limits in the low to mid ng/mL range, depending on the method.[7] Typical separations can take upwards of 15-30 minutes depending on analysis conditions, not including 1-2 hours required to first process blood into serum, a requirement for the HPLC methods at hand.

2.2.1 Theophylline Detection Using HPLC

Theophylline is a bronchodilator and respiratory stimulant, effectively opening constricted airway passages in the body. It is used in treatments for acute and chronic asthma, and is also employed in treating congestive heart failure and acute pulmonary edema alongside other therapeutic drugs. Due to varied differences in uptake and elimination of theophylline between patients receiving similar dosage, constant serum monitoring is required for therapy, in order to maintain theophylline concentrations in the narrow therapeutic window between 10 and 20 μg/mL.[8] Toxicity occurs at concentrations over 25 μg/mL, manifesting conditions from nausea and vomiting to agitation, tachycardia (abnormal rapid heartbeat), tremors and seizures.[9] Maintaining theophylline concentrations in the narrow therapeutic window is crucial for successful therapy.

Typically, a 100 μL serum aliquot is required for successful HPLC analysis of theophylline. Preparation steps include extraction of serum from whole blood, along with preparing internal standard solution. This HPLC technique provides theophylline detection limits around 25 ng/mL, and variability for same day and different days analyses is generally low, demonstrated as roughly 5%.[7] This is key for clinical settings that experience sample backlog, where insufficient staff or instrumentation is available to analyse samples as they are acquired. Regression statistics show excellent linearity relating peak ratios to serum concentrations, often with correlation coefficients (r^2) of 0.99 and higher. Post analysis assay recovery was shown to be dependant on analyte concentrations, ranging from 67% to 97% recovery. Lastly, interference from other therapeutic compounds normally administered in combination with theophylline is negligible.[7] For example, interferants studied in one particular assay were composed of six different compounds with almost no negative impact on theophylline quantification.

HPLC based methods can provide satisfactory results for theophylline analysis, however still lack some fundamental requirements for true point-of-care analysis. The most critical of these is sample analysis time. Although 15-30 minutes does not constitute a long period of time, when analysing hundreds of samples daily, every minute spent acquiring results must be as efficient as possible. Polymer-based HPLC separation provides superior performance in analysis time, and provides additional advantages over traditional HPLC separation columns.

2.2.2 Theophylline Detection Using Polymer-Based HPLC

Faster HPLC methods based on cross-linked polymers molecularly imprinted for theophylline have been recently developed. Heavily cross-linked poly(methacrylic acid-co-ethylene dimethacrylate) molecularly imprinted polymer (MIP) incorporated in a separatory column provides a highly selective, chemically stable and physically robust stationary phase.[10] While processing of whole blood to serum is still required, selective determination of theophylline from 20 µL of serum can be completed in less than 6 minutes. Synthesizing the MIP stationary phase is done by allowing organization of theophylline template with functional monomers, followed by polymerization, and finally Sohxet extraction of theophylline template. The resulting

MIP contains monomers spatially arranged around cavities conforming to theophylline's molecular shape. Under optimal separating conditions, the MIP stationary phase can behave as solid phase sorbent to retain and extract theophylline. Elution of theophylline from HPLC columns containing the MIP stationary phase is achieved by injecting 20 μL of a solvent with the right polarity to disrupt hydrogen bonding between theophylline and MIP, with theophylline extraction efficiency at 89%. Detection limits of 120 ng/mL were obtained with this technique using a 270 nm direct absorption UV detector. As with traditional HPLC, a calibration curve with over four orders of magnitude (0.25-1000 μg/mL) returned high linearity (r^2 exceeding 0.99), making this method many times more selective than other HPLC theophylline quantitation assays.

Advantages of polymer-based HPLC methods for theophylline quantification are increased robustness towards matrices, and increased selectivity towards theophylline, and linear range spanning four orders of magnitude. Although HPLC methods demonstrate excellent performance in measuring theophylline and other respiratory analytes under a variety of conditions, staff training requirements make it a non ideal solution for point-of-care analysis. Optical instrumentation has less intense training requirements, since these instruments are generally simpler to operate.

2.3 Optical Measurements

Chemiluminescence, color reflectance, and optical absorption encompass various forms of optical measurement for bioanalyte quantification. Since optical measurements can have overlapping analyte signatures and interfering matrix effects, devices relying on optics usually involve some form of reagent specifically designed for the target analyte. Complexes with unique optical signatures are formed between reagent and target analyte and are more easily discerned from background signals.

One example of an analysis done with this kind of reagent is quantification of D-dimer, an important bioanalyte for cardiac health, named after the two crosslinked D fragments of fibrinogen protein it contains.

2.3.1 D-Dimer Detection Using Color Reflectance

Coronary artery disease is one of the world's major causes of death. Fatal acute cardiac events still occur with terrifying frequency despite various advances made in therapeutic treatments. In addition, less critical heart diseases that do not pose immediate threat can gradually affect the respiratory system in a negative manner. For example, mitral valve prolapse is a heart condition whereby the mitral valve bicusp does not seal tightly after opening. A valve operating in this fashion causes some of the pumped blood to return through the valve instead of reaching its intented destination. Over long periods, this can cause water buildup in the lungs and reduce oxygenation efficiency.

Quantification of myocardial infarction markers provides insight into cardiac status and disease diagnosis for triage patients with chest pain, and in many cases, rapid results can prevent patient mortality. These cases cannot afford methods in which blood plasma is processed from whole blood, since processing of this nature is very lengthy. As such, many health care professions have increasingly looked for support from advances in rapid quantitative myocardial infarction assays for heart disease prevention, spurring high levels of interest focused on reducing diagnosis time.[11]

D-dimer is excellent for monitoring *in vivo* formation of fibrin. Alongside ultrasonography, it is useful for ruling out diagnosis of cardiovascular disease such as deep venous thrombosis or pulmonary embolism, both of which yield vein or artery blockage in critical areas.[12,13,14,15] Crucial for effective preventative measures or treatment of deep venous thrombosis and pulmonary embolism is time between sampling and results, which in these cases should be limited to a maximum of half an hour.

A rapid point-of-care testing assay for D-dimer, developed by Roche Diagnostics (Manheim, Germany), quantitatively determines D-dimer concentration in heparinized whole blood. The test is measured with a corresponding reader, illustrated in Figure 2.2.[16] D-dimer assays use two antibodies with D-dimer recognition for quantification blood samples. A complex is formed when D-dimer comes into contact with the reagents, the resulting complex containing D-dimer sandwiched inbetween the two antibodies. Once the complex is taken into the detection zone via a third biotinylated antibody, a fourth gold-labelled antibody produces reddish color which is recorded using a camera inside the reader.[17]

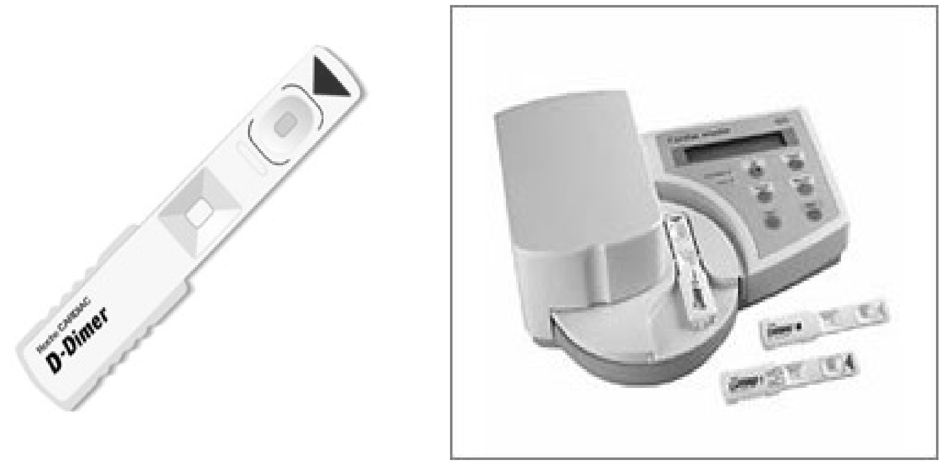

Figure 2.2. CARDIAC D-Dimer (left) with associated CARDIAC Reader (right), developed by Roche Diagnostics (Manheim, Germany). The device quantifies D-dimer using 150 µL whole blood. Quantification is determined when D-dimer in whole blood yields an antibody complex when interacting with gold labeled antibodies in the detection zone, producing a reddish color. Color reflectance is measured using a camera in the CARDIAC Reader.

Investigation into analytical precision, accuracy and quantitative results of the D-dimer assay was also conducted.[18] Concentration ranges of D-dimer spanned 0.1 mg/L to 4 mg/L, with results appearing in 10 minutes. Assays were compared to control solutions, with precision determined to be 7 to 13% compared to controls, and different day variation calculated as 10 to 13%. Specificity of the D-dimer quantification assay was assessed by spiking samples with interfering substances such as hemoglobin, biotin and over a dozen other relevant compounds. Analytical deviations of D-dimer assays were less than 10% in all cases. In addition, altering the amount of sample by adding or subtracting 10 μL blood cause result deviations no more than 18%.[18] This method demonstrates high level of robustness towards matrix and interferants.

Monitoring platelet activity, which can determine blood clot formation, is another useful method for determining cardiac health. This method is used during cardiac surgery to anticipate blood clotting issues, due to the capability of obtaining very rapid results.

2.3.2 Platelet Activity Using Optical Absorbance

Monitoring platelet activity is a common occurrence during coronary surgery. Platelet glycoprotein IIb/IIIa (GP IIa/IIIb) receptors mediate the formation of thrombus platelet, which induces clotting of blood and can produce fatal consequences during surgery if not monitored properly. Antagonists of GP IIa/IIIb are often given to prevent complications stemming from thrombic platelet formation during percutaneous coronary surgery (i.e. accessing inner organs and tissue through needle puncture instead of cutting epidermal layers), while certain conditions require chronic treatment. Some GP IIa/IIIb antagonist agents are administered orally, however with considerable variation in antagonist uptake between individuals.[19]

Elevated levels of GP IIa/IIIb antagonist for long periods of time can cause hemorrhaging, whereas insufficient amounts will not prevent thrombic platelet formation and can lead to serious complications.[20,21] Rapid quantitative methods would increase treatment efficiency by allowing immediate dosage adjustment based on individual response to antithrombotic effect of GP IIa/IIIb antagonist agents.

A fast, automated, cartridge-based method was developed for quantification of GP IIa/IIIb receptors representative of platelet ability to agglutinate (i.e. clumping together) fibrinogen-coated beads. This procedure requires undiluted whole blood inserted into a cartridge. The rapid platelet-function assay (RPFA) device measures agglutination through absorbance with a light source and optical sensors. Mixing of sample is controlled through the use of magnets in conjunction with steel spheres.[19] On-board microprocessor takes 16 measurements per second over a period of 70 seconds to determine rate and extent of agglutination, which is quantified based on increase in IR transmission through the sample. The device then calculates a best fit model, and displays quantitative results in 3 minutes, with minimal input from the user. Fibrinogen-coated beads required for assays are manufactured by coating polystyrene with fibrinogen using passive adsorption, then dying the result with infrared (IR) dye to enhance signals of bead agglutination. Day to day variations studied by measuring an individual's blood five times over a period of 5 months yielded less than 4% variation. Those following chronic oral GP IIb/IIIa antagonist therapy would benefit most from this rapid quantification method due to the narrow therapeutic window and steep dose-response relationship.[19]

There are many other diseases and conditions that affect respiration, with diabetes being one of the major afflictions. Diabetes affects mitochondia's ability to generate ATP (energy), affecting every bodily system.[22,23] Control of diabetes helps reduce impact on respiratory and overall health.

2.3.3 Diabetes Monitoring Using Color Reflectance

Glycosylated hemoglobin is formed when a glucose molecule binds to hemoglobin, occuring without the aid of an enzyme. The binding of glucose and hemoglobin first forms an unstable compound that undergoes further rearrangement to form stable hemoglobin A1c (HbA1c). This binding is very strong, and remains throughout the life cycle of a red blood cell. HbA1c is not known to serve a useful purpose in the body, however it is representative of protein glycenation throughout the body, and therefore useful for monitoring diabetes. If the blood level of glucose is not properly controlled, as in patients with diabetes, hyperglycemia can occur, reducing the efficiency at which ATP is synthesized by mitochondria. In addition to reduced mitochondrial respiration, myocardial infarction, stroke, retinopathy, and microalbuminuria are just a few cardiovascular diseases that are closely related to diabetes.[24] Accurate monitoring is therefore highly recommended to avoid decreased respiratory and cardiac health. Recent studies support the idea that regular measurements of HbA1c with immediate feedback can improve diabetes control, allowing those suffering from this illness to better adjust glucose intake and insulin application.[25,26] If diabetes is improperly treated, several complications can arise such as vision impairment, diabetic nephropathy (i.e. damage to kidneys), in addition to aforementioned cardiovascular diseases.

Since close monitoring is required at all times, monitoring HbA1c with portable instrumentation is ideal. Research done produced a single-use hand-held device that quantifies HbA1c and total hemoglobin from diluted drops of whole blood using time-delay color reflectance.[27] The device, known as DRx HbA1c®, requires only 10 µL of whole blood, available using a finger-stick. It requires no calibration, no manipulation other than diluting the sample with provided diluent, and insertion of sample. Reagent present in the device provides the blue (for HbA1c) and red-brown (for total hemoglobin) colors for reflectance when exposed to diluted blood. On-board microprocessor calculates the assay results and readout is done using an LCD display. Quantitative results appear in 8 minutes, and the results are nearly as accurate as with laboratory solutions such as the Bio-Rad DiaSTAT™ (Bio-Rad Laboratories, California, USA). Figure 2.3 demonstrates the size difference between the two instruments. [28,29] Repeatability was determined as roughly 10%. Alternative methods are almost exclusively based on sending samples to laboratory, which provide equivalent results, however involves a time delay (from hours to days).

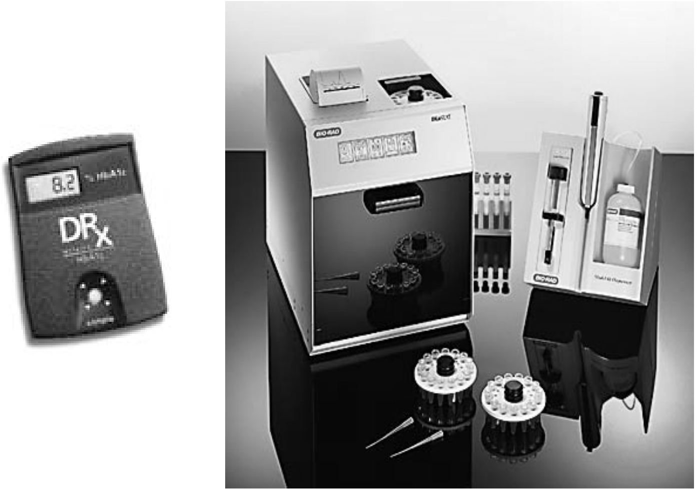

Figure 2.3. Comparison of two hemoglobin A1c quantification instruments. The DRx HbA1c®, shown on the left, is a hand-held portable device. The Bio-Rad DiaSTAT™, on the right, is a much larger laboratory instrument. Both instruments quantify with similar results (within 1%).

Optical methods are currently the most effective in accurate control of diabetes. Another example of respiratory disease is respiratory degredation. Bioanalyte monitoring of fluorescent coenzymes are very sensitive bioindicators of respiratory degredation.[30] Time resolved fluorescence techniques are typically easy to use, require little staff training, and provide excellent detection limits.

2.4 Time Resolved Fluorescence

Fluorescence based methods allow potential for very high specificity and excellent detection limits, provided target analytes can either fluoresce or be made to fluoresce. Time resolved fluorescence bears many similarities to straight fluorescence in that light of a certain wavelength is used to excite target molecules, which then emit light of their own at the same wavelength (resonance fluorescence) or different wavelength (non-resonance fluorescence) compared to the excitation source. The differences between fluorescence and the time resolved variety stem from the type of measurement. Time resolved fluorescence measures fluorescence lifetime after samples have been exposed to a flash of light, as opposed to traditional steady state fluorescence measurements. Optical data is recorded when background fluorescence has decreased, leaving only analyte fluorescence remaining. Matrix and background elimination of this manner produces very impressive sensitivity. Since few molecules fluoresce on their own, use of reagent to tag or form a stable fluorescing complex is often a requirement when dealing with respiratory or cardiovascular bioanalytes.

2.4.1 Myocardial Monitoring Using Time Resolved Fluorescence

Chronic cardiac disease sufferers would benefit from quick quantitative measurements, as results could be ready during each patient-physician encounter. This greatly reduces time required for patient care, as separate visits for drawing blood samples and obtaining results would no longer be required.

Immunoassays based on time-resolved fluorescence have resulted in rapid quantitative determination of myocardial infarction markers in whole blood. One such concept, based on a stable europium chelate, delivers results in 15 to 20 minutes using whole blood.[31] The europium chelate based reagent operates as an all-in-one universal immunoassay for three biochemical markers, cardiac troponin I (cTnI), MB creatine kinase (CK-MB), and myoglobin. Dynamic range for calibration conducted with the europium chelate reagent extended over at least 3 orders in magnitude for CK-MB, 4 orders for cTnI and myoglobin, with detection limits in the sub ng/mL range for each biochemical marker. Result agreement between assays from whole blood and blood plasma were very high, as well as having inter-assay precision ranging of roughly 10%. These results compare favorably to commercially available testing kits suitable for laboratory environments. Correlation was high (r^2 of 0.94) when comparing the cTnI quantitative results to a commercial equivalent.[31,32,33]

Primary benefits of this time resolved fluorescence method lies in rapid, quantitative results performed on whole blood with one universal dry reagent. Moreover, this method compares favorably to more complex laboratory counterparts. This portable time resolved technique is highly suited for non-laboratory environments such as satellite laboratories in larger complexes, or small medical facilities either administering primary or specialized care.[31]

There are instances however when respiratory bioanalytes have overlapping signatures with background or other analytes. An example of a respiratory bioanalyte that would be difficult to measure optically is dopamine. Samples containing dopamine typically require highly selective electrochemical measurement means, which rely on measuring electrical signal through a sample.

2.5 Electrochemical Measurements

Quantification of bioanalytes done by electrochemical means are achieved through measuring potential or current across a sample. Electrodes are typically employed in this method, and samples are often subject to separation mechanisms, usually lateral flow chromatography, prior to electrochemical measurements. Matrix effects and reproducibility are in most cases a larger issue than with other methods, since accuracy of quantification is directly related to efficiency of target bioanalyte separation from sample matrix. Nevertheless, many medical devices designed for clinical settings and home use ultimately end with some kind of electrochemical measurement to quantify target bioanalytes.

2.5.1 Dopamine Detection Using Electrochemical Analysis

Dopamine is a catecholamine neurotransmitter found in the brain, which affect a wide variety of bodily functions. This neurotransmitter is most commonly associated with Parkinson's Disease, in which action of dopamine in neurons of the substantia nigra (a section found in the midbrain) is insufficient.[34,35,36] Dopamine can also be used to regulate respiration, improve respiratory muscle function, and is associated with cardiovascular function.[37, 38, 39] Lastly, clinical settings can make use of dopamine to improve renal function in patients suffering from congestive heart failure.[40] Dopamine is clearly an analyte that interacts with many bodily systems, and thus requires good quantification methods.

Currently, few techniques exist for direct dopamine quantification, with the majority of methods functioning off-line and lacking real time result output. Sample extraction and preparation for these techniques is often lengthy, very invasive, and involves multiple complex steps. As such, current dopamine biomonitoring techniques are far from ideal and acoustic methods discussed later have potential to be far less invasive. One example of a current electrochemical detection method is coupled with HPLC. This technique is able to quantify dopamine along with five other bioanalytes simultaneously.[41] Unfortunately, this method requires brain tissue samples for analytes. Samples weighinh 1 mg were used for this type of analysis, and internal standardized calibration curves demonstrated linearity in the 0.1 to 50 ng dopamine in the presence of the other bioanalytes in 10 ng amounts. Correlation coefficient (r^2) for the calibration exceeded 0.99, with coefficients of variation as low as roughly 4%, while dopamine detection limits were calculated as 20 to 60 pg, depending on sample.[41] Although these methods illustrate solid performance, requisite sampling directly from the brain severely limits general applicability in clinical settings.

A highly selective electrochemical method of dopamine quantification was more recently developed based on glassy carbon electrodes modified with melanin-type polymer.[42] Modification of such electrodes using the polymer provides analyte selectivity through molecular recognition. Ascorbic acid was added to solutions to enhance the dopamine signal by reduction of electrochemically generated dopaminequinone. Constructing dopamine calibration curves using this technique yielded high sensitivity and good linearity in the concentration range spanning 7 to 45 nM, with correlation coefficient (r) exceeding 0.99. Dopamine detection limits were calculated as 5.0 nM, while inter-electrode reproducibility was reported as less than 8.0% RSD, depending on experiment conditions.[42]

Quantification of dopamine and bioanalytes *in vivo* is not easily accomplished. Acoustic analysis has the potential for simultaneous multianalyte analysis complex matrices with minimal invasiveness. Chief among current acoustic quantification methods are thickness shear mode resonators.

2.6 Thickness Shear Mode Resonators

Quantitative ultrasound is a fairly new field, have only come to fruition in the last few decades. Primary interest in this field has been focused on thickness shear mode (TSM) resonators. TSM resonators are based on the concept of resonance frequency shifts based on changes in mass of a piezoelectric device. Most TSM resonators are constructed using two electrodes, with a piezoelectric quartz wafer inserted in between. When RF potential is applied to the electrodes, the piezoelectric wafer resonates at its native resonance frequency, typically operating between 1 and 10 MHz. TSMs constructed using quartz are often referred to as quatrz crystal microbalances (QCM). The piezoelectric device resonance frequency can be determined by the following relationship,

$$f_0 = \frac{1}{2\pi \sqrt{L_m C_m}}, \qquad \text{Eq. 2.1}$$

where f_0 is the wafer resonance frequency, L_m is the mechanical inductance, and C_m is the mechanical capacitance of the device. Molecules in direct contact with the quartz wafer alter resonance frequency and magnitude by modifying the mechanical inductance and capacitance, occuring due to the damping effect of the molecules. Amount of resonance frequency shift and magnitude attenuation is proportional to density. Surface area of the wafer also influences the resonance frequency since surface area is proportional to capacitance through the following relationship,

$$C_m = \varepsilon_r \varepsilon_0 \frac{A}{d}, \qquad \text{Eq. 2.2}$$

where ε_r is the dielectric constant of the material between the capacitor plates (wafer), ε_0 is the eletric constant (8.854×10^{-12} F m^{-1}), A is the area of overlap between plates and material, and d is the separation between plates.

TSM resonators are most often used to determine liquid properties such as density and viscosity. Applications of TSM resonators cover a wide variety of different fields, from jet fuel monitoring to detecting changes in viscosity and pH. Moreover, a respiratory monitoring system was recently developed based on a porphyrin modified QCM.[43] Smooth and textured surface TSM resonators can be manufacturer for specific environments, increasing applicability.[44] Due to the increased number of molecules found in liquids, sensitivity cannot match TSM resonators operating in gaseous media. TSM resonators are also very susceptible to molecular fouling (i.e. saturation of resonating media with molecules) which prevents additional molecules from shifting resonance frequency. The cause of increased molecular fouling stems from having a limited wafer surface area to interact with surrounding molecules.

Analyte sensitivity is achieved by coating TSM resonator quartz wafers with a molecularly sensitive layer. Only certain molecules will adhere to the wafer surface, improving selection of which molecules effect resonance frequency and magnitude. Analyte sensitive TSM resonators are often referred as "electronic noses", and are used throughout the food industry as flavor sensors, in addition to fragrance sensors in cosmetics.[45] Analyte sensitive QCM sensors have also been developed to detect other bioanalytes, such as viruses. Several methods were developed incorporating QCMs in biosensors for herpes, hepatitus B, and tobacco mosaic virus (TMV).[46, 47, 48] In these applications, QCMs were coated with peptide nucleic acids (for herpes), antibodies (for hepatitus B), and molecularly imprinted polymer (for TMV). Sensitivities for these methods vary wildly, and are very dependant on the strength of interaction between analyte and wafer surface coating. The best scenario demonstrates sensitivity approaching a single herpes virus particle with peptide nucleic acid coated QCM, which is very impressive.[46] A detection limit of low pg/L was achieved for hepatitus B,[47] and TMV detection resulted in a limit of 100 to 1000 ppb.[48] Although these coated piezoelectric devices cannot exclusively interact with analyte in question, it is clear the recognition surface coating increases the overall likelyhood of interactions being more frequently with target analyte. Recognition coatings can hence make frequency shifts more respresentative of target analyte presence. Despite the advantages of analyte sensitive TSM resonators, they are still susceptible to a number of negative factors which can present significant technological challenges.

Since TSMs are based on particle interaction with the quartz wafer, any environmental or ambient condition affecting molecular motion will also affect TSM resonator efficiency. Wafer irregularities originating from manufacturing processes can also broaden native frequencies, impairing detection of small frequency shifts.

Acoustic methods capable of simultaneous multi-analyte analyses not limited by molecular fouling need to be developed for more reliable acoustic analytical concentration analysis.

2.7 Ultrasound Contrast Agents

Tangential work related to TSMs lead to the recent development of ultrasound contrast agents. Contrast enhanced ultrasound provides a straight-forward method to distinguish between acoustic signal of contrast agents and background acoustics. The contrast agents are composed of gas-filled polymeric microbubbles, and share some qualities with TSMs in that they exhibit sharp resonance frequencies when perturbed by ultrasound. The spherical shape of contrast agents allows them to resonate at specific frequencies based on modulus, microbubble size, and shell thickness.[49] When contrast agents delivered into a target sample are perturbed with ultrasound, the agents will resonate at their characteristic frequencies. Ultrasound backscatter from the entire sample is collected, unwanted background frequencies are filtered and leaving only contrast agent resonance remaining. This technique, illustrated in Figure 2.4, improves echogenicity, edge detection, and contrast in imaging applications over other ultrasound methods.

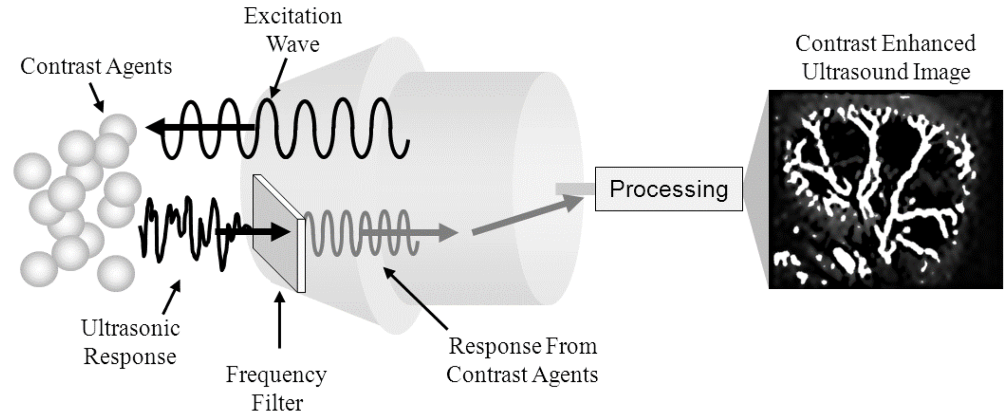

Figure 2.4. Contrast enhanced ultrasound from specific contrast agent resonance frequency. A perturbation wave is used to excite contrast agents, and the resulting back scatter is band filtered for the specific contrast agent resonance frequency. Sharper edges and higher contrast result in better visualization of blood vessels, such as in the rabbit kidney shown above.

Contrast agents are applied to visualize ventricle and atrium volume in echocardiograms, kidney stones, and blood vessels of specific organs when other ultrasound techniques cannot provide enough detail. Table 2.1 shows a selection of current contrast agents used in medical fields. Due to their small size and nature, many such contrast agents have native frequencies well above the 1 to 10 MHz range, and are therefore probed by looking at sub harmonics.[50] Analysing sub harmonics in this manner allows contrast agents to be manufactured in smaller size (at the micrometer level) where they can traverse through samples easily.

Although gas filled microbubbles are ideally suited as contrast agents since they typically resonate at a very narrow frequency, ultrasound contrast agents are not limited to being composed of microbubbles filled with a particular gas. Any micron or sub-micron compound that, when ultrasonically perturbed, resonantes at a characteristic frequency or frequencies can serve as an ultrasound contrast agent. For example, SonoRx is a contrast agent manufactured by ImaRx LLC composed of a modified cellulose structure and does not contain an interior gas (see Table 2.1).

Name	Manufacturer	Shell Composition	Gas	Microbubble Size
Albunex	Mallinckrodt	Albumin	Air	1-15 um
Levovist	Schering AG	Galactose	Air	< 10 um
MRX 115	Definity	Phospholipid	Octafluoropropane	< 10 um
Optison	Molecular Biosystems	Albumin	Octafluoropentane	< 10 um
Sonazoid	GE Healthcare	Surfactant Membrane	Fluorocarbon	3.2 um
Imavist	IMCOR Pharma	Surfactant Membrane	Perfluorohexane	< 10 um
Sonovue	ImaRx LLC	Phospholipid	Sulfur hexafluoride	11 um
AI 700	Acusphere	PLG copolymer	Perfluorocarbon	N / A
SonoRx	ImaRx LLC	Simethicone-coated cellulose (no gas)		N / A

Table 2.1. Common ultrasound contrast agents, major manufacturers, contrast agent composition and sizes (current as of June 2011).

Nanogel polymers have unique ultrasonic characteristics that can be manipulated by altering environmental parameters, making nanogels attractive as potential ultrasound contrast agents. These characteristics can be visualized when analysing pNIPA nanogels at a range of temperatures.

When probing pNIPA nanogels with a single ultrasound frequency transducer, sharp ultrasonic attenuation at the point of phase transition is observed as temperature is increased passed the critical threshold temperature (T_c) of 33.6°C.[51] Attenuation at the phase transition threshold investigated can be represented by data similar to Figure 2.5.

Adding molecular imprinting to the synthesis process allows pNIPA nanogel to interact more strongly with target molecules compared to others. This process has already been established in literature.[52] Once polymerization is complete,

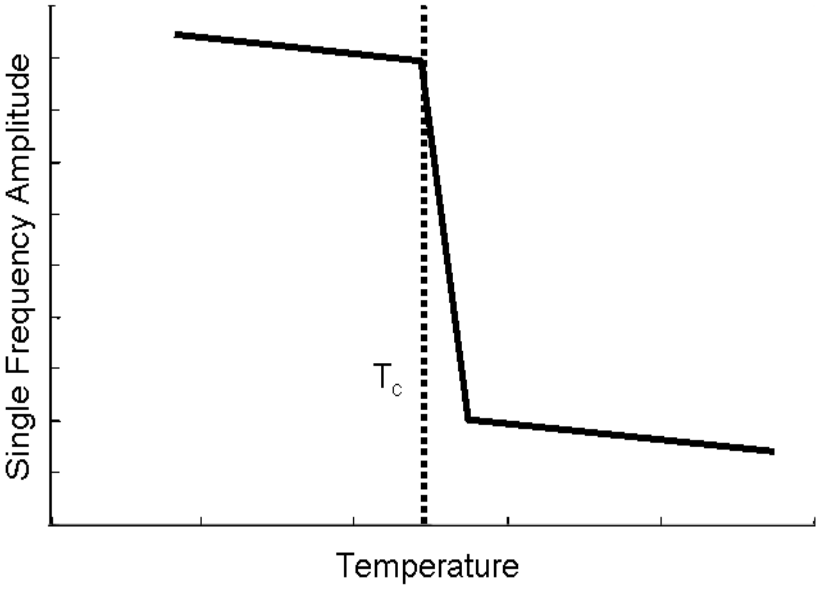

Figure 2.5. A generated example of typical ultrasonic behavior of pNIPA nanogels undergoing phase transition as temperature is increased past critical threshold T_c. Monitoring a single ultrasonic frequency reveals sharp attenuation, with an associated sharp increase in turbidity, at the phase transition threshold between swollen and collapsed state. Additionally, change in ultrasonic properties and turbidity for pNIPA a result of undergoing phase transition is completely reversible.

molecularly imprinted "pockets" selective to the chosen analyte are formed. Presence of analyte in imprinted polymer networks can then have subtle, but measureable, effects on physical and ultrasonic properties of the nanogel. The combination of ultrasound, pNIPA nanogels and molecular recognition provides a generalized method of synthesizing analyte sensitive contrast agents. This method combines minimally invasive properties of ultrasound with molecular imprinting for a selective quantification method suitable to become a rapid point-of-care system.

Point-of-care in clinical applications generally has stringent requirements. Previous work in this field has shown that systems must be capable of delivering results reproducable within 5 to 10% RSD in the nano to micromolar range. Likewise, analysis should be rapid (under 5 minutes), without being prone to molecular fouling. Maximum portability and minimal staff training outline other desired properties. The contrast agent approach to molecular quantification described in this thesis provides a method adhering to all aforementioned requirements. The follow chapters demonstrate how this ultrasound system based on analyte sensitive nanogels is an excellent point-of-care method.

Chapter 3.

Ultrasonic Quantification Using Smart Biosensors

This chapter discusses the development of analyte sensitive nanogel contrast agents. Characterization of the nanogel contrast agents is illustrated in the first section, describing ultrasonic properties associated with nanogel phase transition. These contrast agents were then combined with a modified A-mode ultrasonic detection system, which is detailed in later sections of this chapter. Finally, ultrasonic quantification of a target analyte in solution using analyte sensitive nanogel contrast agents, the goal of this thesis, is demonstrated in a published paper presented towards the end of the chapter.

3.1 Characterization of Nanogel Contrast Agents

Ultrasonic properties of pNIPA were investigated to determine the extent of ultrasonic activity produced when altering parameters, and whether or not parametric changes could be measured ultrasonically. Literature has demonstrated a larger, reversible change in ultrasonic signature of pNIPA under a range of conditions, through analysis of changes at one ultrasonic frequency.[51] Probing and monitoring ultrasonic amplitude at this frequency using a simple single tranducer system revealed sharp attenuation as temperature was increased passed the pNIPA critical phase transition threshold. The sharpest attenuation was noted at the critical phase transition temperature, pinpointing the exact temperature of pNIPA network collapse. We reproduced this experiment with a similar ultrasonic detection system using a 1.9 MHz narrowband transducer to both emit negative impulses and record backscatter. The pNIPA nanogel was synthesized as described in the above work.[51] Monitoring ultrasonic amplitude at 1.9 MHz as pNIPA temperature was increased illustrated the same kind of attenuation seen in aforementioned literature, as shown in Figure 3.1. These results demonstrated the ability to impart changes in pNIPA ultrasonic profile through adjustment of temperature. Other research has shown that altering several parameters, such as hydrostatic pressure, ionic strength and hydrophobic

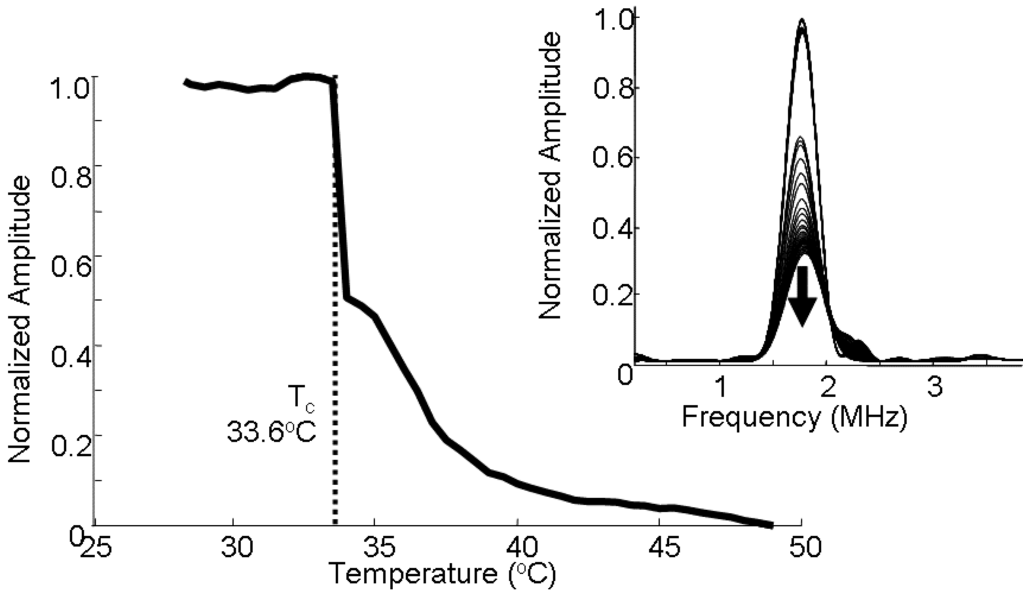

Figure 3.1. Ultrasonic attenuation at 1.9 MHz in ppNIPA nanogels undergoing phase transition. This transition was triggered by raising sample temperature above critical threshold temperature T_c. Data was acquired using a 1.9 MHz narrow band transducer (Advanced Technology Labs Inc., Pennsylvania, U.S.) to produce negative impulses and receive backscatter. A Panametrics 500PR Pulser/Receiver (Panametrics Inc., Massachusetts, U.S.) was used to generate the electrical impulses driving the transducers. An SDS 200 oscilloscope (SoftDSP Co., Seoul, Korea) sampling at 12.5 MHz using 9 bit A/D conversion converted data to digital format.

interactions can produce the same ultrasonic changes in pNIPA and other nanogels.[53, 54] However ultrasonic changes of this nature can come from numerous sources, hence amplitude at only one frequency provides insufficient data for analyte quantification. If additional frequencies could be monitored, trends correlating to presence of analyte could be verified at multiple frequencies and multilinear regression might be employed to calculate concentration. We therefore improved on the original experiment by adding a 5 MHz wide band ultrasonic receiver, facing the 1.9 MHz narrow band emitter, and measuring transmission of the ultrasonic spectrum between 1 and 10 MHz. Details regarding this modified A-mode ultrasonic detection system and its components are available in Appendix B.

Probing pNIPA nanogels with narrow frequency impulses and recording ultrasonic transmission with wide band receiver allowed monitoring of the entire ultrasonic spectrum as opposed to only one frequency. Temperature was inceased, as previously done, to induce phase transition in pNIPA. This resulted in ultrasonic spectra exhibiting attenuation and amplification simultaneously at different frequencies is shown in Figure 3.2. Additional data obtained using this system allows amplitude trends correlated with increase in concentration to be identified much more easily in potential quantification experiments. The foundation of this thesis was developing a method through which ultrasonic profile changes in pNIPA can be imparted based on analyte concentration.

The most common and cost-effective approach for analyte selectivity in many fields of chemistry is often molecular imprinting. In many applications, polymers synthesized in the presence of the target analyte, with subsequent analyte removal, produces imprinted pockets which conform to target analyte molecules. As previously mentioned, work on molecular imprinting of pNIPA derived polymers has been

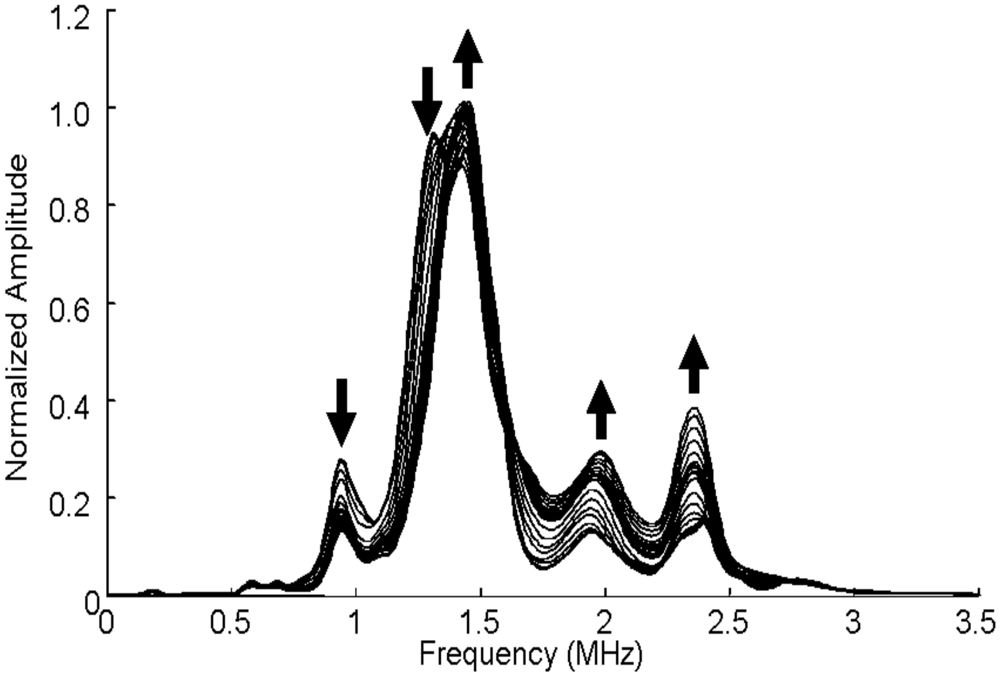

Figure 3.2. Ultrasonic attenuation and amplification (denoted by arrows) in the 0.5 to 3 MHz range in ppNIPA nanogels undergoing phase transition. This transition was triggered by raising sample temperature above critical threshold temperature T_c (33.6 °C). Data was acquired using a 1.9 MHz narrow band transducer (Advanced Technology Labs Inc., Pennsylvania, U.S.) to produce ultrasonic signal and a 10 MHz wide band transducer (Optel Inc., Wrocław, Poland). A Panametrics 500PR Pulser/Receiver (Panametrics Inc., Massachusetts, U.S.) and a SDS 200 oscilloscope (SoftDSP Co., Seoul, Korea) sampling at 12.5 MHz using 9 bit A/D conversion were used to generate impulses and convert data to digital format.

covered in the literature, and sythesis procedures are straight forward.[64] Additionally, research into the impact of analyte presence on molecularly imprinted nanogels has demonstrated measureable shifts in nanogel physical properties. These physical property changes can potentially lead to changes in ultrasonic profile, similar to the effect of phase transition on nanogel network, revealing a paradigm through which analyte presence can alter nanogel ultrasonic properties. [53, 54] Incorporation of molecular imprinting to develop analyte sensitive pNIPA contrast agents is covered in the next section, and analyte quantification assays in the micromolar range are also illustrated and discussed.

3.2 Investigating Analyte Ultrasonic Signature

Measuring ultrasonic profile changes that occur when molecularly imprinted pNIPA nanogels are exposed to a particular analyte was the next approach toward developing analyte sensitive contrast agents. Solutions of pNIPA nanogel were synthesized in the presence of theophylline, a known bronchodilator, which was selected as a clinically relevant target analyte.[55] The synthesis procedure is described in section 3.9.2 of this chapter. The theophylline template was subsequently removed from the synthesized polymer using successive separatory extractions, leaving only the molecularly imprinted pNIPA remaining. Ultrasonic data was collected from solutions containing solutions containing high and low concentrations of theophylline and a chemically similar molecule, caffeine. Analyte specific ultrasonic signatures were determine by the difference between high and low concentration spectra for each analyte. Frequency changes unique to each analyte were noted when comparing ultrasonic signatures of the two analytes. These results are displayed in Figure 3.3, with darkened areas representing frequency changes between high and low concentrations for each analyte. The regions of greatest difference between theophylline and caffeine exist in the 1.5 to 2.0 MHz region, with negative amplitude changes attributable to theophylline, and positive changes belonging mostly to caffeine.

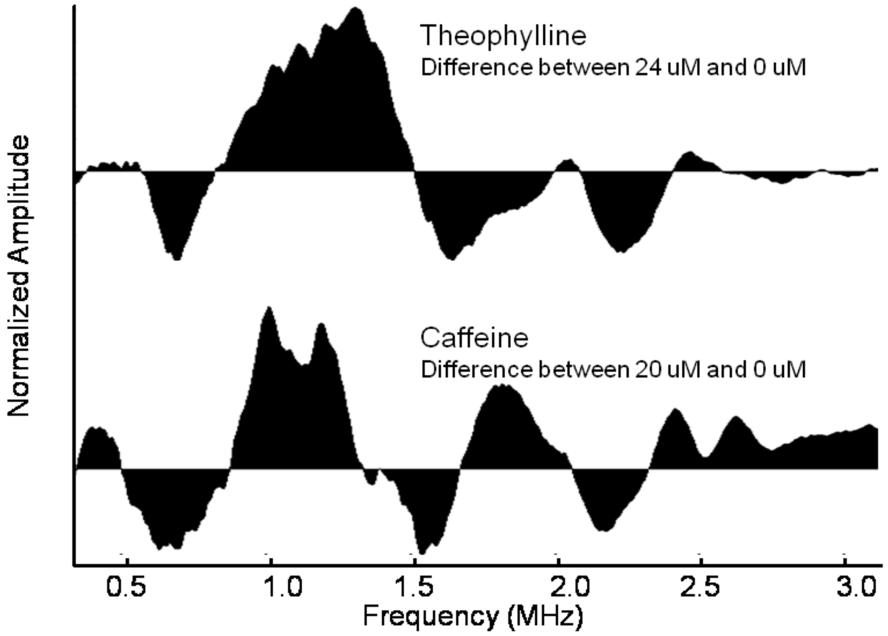

Figure 3.3. Difference spectra comparing low and high concentrations of theophylline and caffeine. Ultrasonic frequency spectra of solutions containing high concentration of analyte were subtracted from spectra with low analyte concentration. The resulting difference spectra demonstrates frequency changes attributed to concentration changes for each analyte.

For example, selecting frequencies specific to concentration changes of theophylline in the 1.5 to 2.0 MHz region allows construction of concentration models selective for theophylline in presence of caffeine.

The results shown in Figure 3.3 also illustrated that minute, localized effects of many analyte molecules docking into imprinted binding sites leads to measureable ultrasonic profile alterations. Having identified unique ultrasonic profile changes of pNIPA related to analyte concentration, a concentration model was constructed for theophylline. The regression model would then determine if any linear relationship is present between ultrasonic profile change and analyte concentration.

3.3 Manuscript

Ultrasonic Quantification using Smart Biosensors

Authors:

David Troïani, Jonathan R. Dion, and David H. Burns

3.4 Abstract

Analyte quantification in samples with extensive matrix effects can be challenging using conventional analytical techniques. Ultrasound has been shown to easily penetrate samples that can be difficult to measure optically or electrochemically, though it provides little chemical information. Recent ultrasound contrast agents provide highly localized contrast within a sample based on concentration. We have developed a general approach for creating smart biosensors based on molecularly imprinted nanogel polymers that recognize and bind a target analyte, changing ultrasonic properties with analyte concentration. Multilinear analyte calibration in nanogel solutions provided quantification of the chosen analyte, theophylline, from 8.4 µM to 6.1 mM with a high degree of linearity (correlation coefficient exceeding 0.99). Simultaneous quantification of both theophylline and of an interfering species, caffeine, was also carried out, providing an avenue for simultaneous analyte analysis with one smart biosensor that can be dispersed and remotely detected.

3.5 Keywords

Ultrasound, nanogel, molecular imprinting, biosensor, analyte recognition

3.6 Introduction

3.6.1 Background

Detection and quantification of analytes in biomedical applications is typically done by optical spectroscopies or electrochemical techniques. However, many samples containing multiple analytes with overlapping chemical signatures often require chromatographic separation or extensive sample pre-cleaning.[56, 57, 58] We have developed a general system for creating smart biosensors based on molecularly imprinted nanogel polymers. The biosensor exhibits a characteristic ultrasound frequency profile that is dependent on the stiffness and size of the polymer. Target recognition and binding cause changes to these physical parameters, and changes to the ultrasonic frequency profile are measured. We have applied this technique to the analytical determination of theophylline, a therapeutic agent for respiratory diseases. Therapeutic concentrations of theophylline are typically in the 55 to 110 μM range.[59] Caffeine, a structurally similar molecule, was also investigated by this method as a competitive agent. Using this technique, a high degree of linearity for individual and simultaneous quantification of both chemical species is demonstrated.

Hydroxypropyl cellulose (HPC) and N-isopropylacrylamide (pNIPA) nanogels undergo a reversible, volume phase transition between swollen and condensed states in solution. In the swollen state, nanogel solutions are clear, owing to a large and diffuse structure. The condensed state is characterized by an increase in turbidity as the nanogel microspheres contract, becoming stiffer and expelling water from the structure. As a result of these physical changes, ultrasonic properties of nanogels are highly dependent on the phase of the gel. Nanogel phase transitions can be prompted by external forces such as fluctuations in ionic strength, temperature, or hydrostatic pressure. When the temperature of the nanogel solution is elevated pasta the critical threshold temperature (T_c), a broad change to the ultrasound spectrum will be seen. Figure 3.4 shows the ultrasound frequency profiles of the HPC and pNIPA nanogels. The mean of the spectra has been subtracted to more clearly illustrate the changes in the frequency power spectrum when the nanogels undergo a temperature-induced phase transition. It can be seen that frequency exchanges occur over the entire spectrum, including both in-phase and out-of-phase changes.

To promote analyte specificity in the nanogel biosensors, molecularly-selective binding sites can be created by allowing self-assembly of the nanogel monomers and analyte prior to polymerization.[60, 61] Upon docking of the analyte into an imprinted pocket, the nanogel will undergo a change in physical properties and in ultrasonic response. Molecular sensitivity of the binding sites is dependent on the affinity of imprinted nanogels for an analyte. This affinity for the template is based

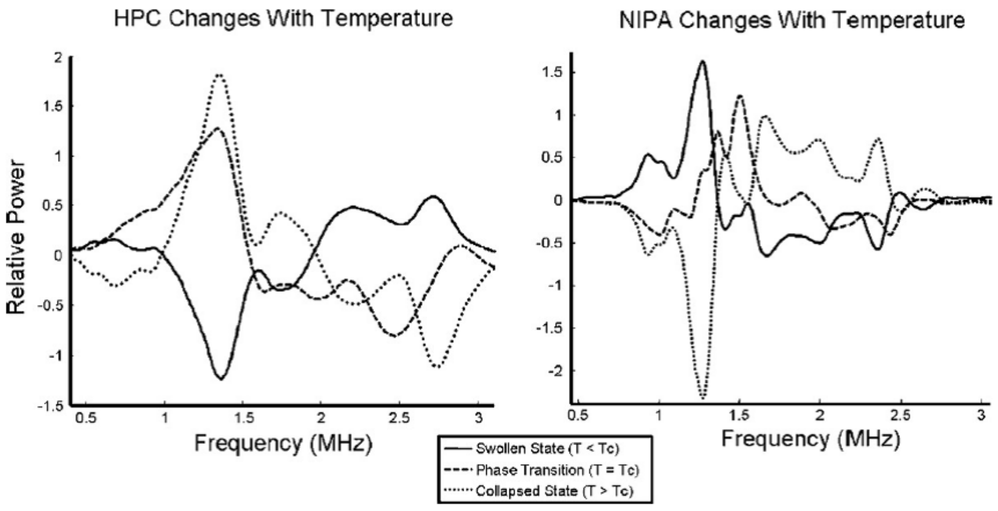

Figure 3.4. Frequency profiles of pNIPA and HPC nanogels undergoing phase transition as a result of increasing temperature. Mean ultrasound spectral profile has been subtracted to emphasize the changes before ($T < T_c$) (solid line), during ($T = T_c$) (dashed line), and after ($T > T_c$) phase transition relative to critical threshold temperature T_c.

on hydrogen bonding between the polymer network and the template molecule, as well as steric factors due to the shape of the molecularly imprinted pocket. By adjusting the specific chemistries of the nanogels, both low and high affinity biosensors can be created. With high molecular weight HPC, affinity towards the analyte is based on large, loose hydrogen bonding pockets once crosslinking is complete.[62] In contrast, high affinity binding is achieved in pNIPA nanogels that are copolymerized with methacrylic acid (MAA), which forms strong hydrogen bonds with the template.[63] Molecularly imprinted pNIPA nanogels have been shown to exhibit turbidity changes with sub-mM concentrations of theophylline.[64]

We present a unique method for analyte quantification using imprinted nanogels in concert with ultrasonic detection. Molecularly imprinting nanogels with analyte template can provide high degrees of analyte specificity. Non-invasive ultrasonic detection of nanogels provides potential for rapid, efficient detection of analytes based on contrast enhanced backscatter at select frequencies.

3.6.2 Principle

A conceptual model can be developed to illustrate the changes to the ultrasonic frequency profile of nanogel polymers when an analyte binds to the molecularly imprinted pockets. Considering nanogels to be deformable microspheres in solution with given diameter and modulus, changes in nanogel microsphere physical properties at equilibrium can alter ultrasonic profile. Each microsphere resonates at a given frequency when perturbed by ultrasound pressure waves. Resonant frequency shifts occur as a result of altering the size and stiffness of the microsphere. Physical properties of molecularly imprinted nanogels change with docking of an analyte. In particular, nanogel microspheres have been shown to both increase in stiffness and decrease in size as analyte molecules occupy the binding sites.[65, 66] With ultrasonic compression and rarefaction, polymer beads oscillate isotropically at specific resonance frequencies dependant on these physical properties.[67] A model, while not ideal for network nanogel beads in solution, approximates behavior in ultrasonic fields by assuming nanogels to be gas-filled microspheres with a thin polymer shell. This thin shell approximation is straightforward to model. Though exact comparisons would not be expected from the shell model for spherical nanogel networks, the model illustrates relative trends of the resonant frequency with changing nanogel physical properties. Nanogel resonance frequency for thin-shelled microspheres in solution can be derived from the resonance frequency of gas-filled contrast agent microbubbles as,

$$f_0 = \frac{1}{2\pi \quad a_e}\sqrt{\frac{1}{\rho_L}\left(3kp_0 + 12G_s\frac{d_{Se}}{a_e}\right)}, \qquad \text{Eq. 3.2}$$

where a_e is the equilibrium radius of the microbubble, ρ_L is the density of the surrounding liquid, k is the polytropic exponent of the gas, p_0 is atmospheric pressure, G_s is the shell shear modulus, and d_{Se} is the shell thickness.[68] Bulk modulus of an adiabatic gas, K_g, can be written as,

$$K_g = kp_0, \qquad \text{Eq. 3.3}$$

while the bulk modulus of a polymeric microbubble, K_p, receives additional contributions from the outer shell,

$$K_p = K_g + 4G_s\frac{d_{Se}}{a_e}. \qquad \text{Eq. 3.4}$$

For a deformable nanogel microbubble with a very thin, shell and no interior gas, the bulk modulus contribution from adiabatic gas can be negated (K_g = 0), and using equation 3.4, equation 3.2 can be rewritten as,

$$f_0 = \frac{1}{2\pi \quad a_e}\sqrt{\frac{3K_p}{\rho_L}}, \qquad \text{Eq. 3.5}$$

which resembles the Minnaert equation for microbubbles. As the modulus (or stiffness) increases, the resonance frequency of the nanogel would increase. In contrast, an increase in nanogel radius would result in the nanogel resonance frequency decreasing.

An imprinted nanogel will have a specific resonance frequency based on the size and stiffness. When the template molecule binds to the molecularly imprinted pocket, both the size and stiffness of the gel are expected to change, leading to a more complicated relationship of the resonance frequency. Though a structurally similar molecule with lower affinity may also interact with the molecularly imprinted pocket, the decreased hydrogen bonding would result in different physical changes through the nanogel network. In a nanogel with multiple pockets, the relationship between the modulus and radius leads to specific frequency profiles for nanogels dependent on both the concentration and affinity of the nanogel analytes bound. In addition, nonlinear propagation of pulsed ultrasound in aqueous samples at high acoustic pressures broadens the frequency content.[69] Therefore, multiple resonance frequencies could be probed with one ultrasonic pulse. The concentration of analyte absorbed into the nanogel pockets was estimated based on measurements of multiple frequencies within the ultrasound pulse propagating through the sample cell.

3.7 Materials and Methods

3.7.1 Synthesis of Theophylline Imprinted HPC Polymer

Low affinity molecularly imprinted HPC was prepared by 0.5 g of HPC powder (100,000 MW) and 0.1 g of theophylline to 49.4 g of dH_2O and stirring for 3 days to form a homogeneous 1 wt % solution of HPC. A total of 40 μL of divinylsulfone (DVS) and sodium chloride to a concentration of 1 mM were added.[70] After 3 hours of stirring, 5 drops of 1 M sodium hydroxide were added to the solution to raise the pH to 12. The cross-linking reaction was allowed to continue for 5 hours. The cross-linked polymers were then dialyzed against dH_2O for 3 days to remove the theophylline and any free DVS. The extraction of theophylline was confirmed spectroscopically at 271 nm. All chemicals were purchased from Sigma-Aldrich (Ontario, Canada). The mean HPC nanogel diameter was determined to be 105 nm (0.39 PDI and batch-to-batch variability of 20%) by dynamic light scattering using a Brookhaven Instruments ZetaPALS particle size analyser.

3.7.2 Synthesis of Theophylline Imprinted pNIPA Polymer

High affinity molecularly imprinted pNIPA was prepared by adding 1.0 g of pNIPA monomer, 0.08 g of N,N'-methylene-bis-acrylamide (MBA), 0.08 g methacrylic acid (MAA), and 0.18 g theophylline to 99 mL of distilled water (dH_2O) to form a homogeneous 1 wt % pNIPA solution with stirring over 4 hours to ensure complete dissolution.[71] Oxygen in the solution was purged with nitrogen gas during this time. We then added 15 mg of ammonium persulfate to initiate the polymerization and 60µL of tetramethylethylenediamine as an accelerator. The solution was left to polymerize for 4 hours with gentle stirring. Once the imprinted nanogel was formed, the theophylline was removed by successive methylene chloride extractions. The extraction of theophylline was confirmed spectroscopically at 271 nm. All chemicals were purchased from Sigma-Aldrich (Ontario, Canada). The mean pNIPA nanogel diameter was determined to be 396 nm (0.39 PDI and batch-to-batch variability of 5%) by dynamic light scattering using a Brookhaven Instruments ZetaPALS particle size analyser.

3.7.3 Nanogel Solutions

HPC and pNIPA nanogels were used to estimate theophylline concentrations based on the ultrasonic response. Solutions containing 1% imprinted nanogel by weight in water were made and increasing amounts of analyte were added to each solution. In the low affinity-binding imprinting HPC, concentrations of theophylline ranged between 0.2 to 6.1 mM. In order to reflect the higher affinity binding in pNIPA, concentrations of theophylline ranged between 8.4 to 167 μM theophylline.

To assess the selectivity of the molecularly imprinted nanogels, caffeine was used as an interfering species while calibrating for theophylline, due to the chemical structures differing by one methyl group. Matrices of 25 HPC and 30 pNIPA 1% by weight solutions were prepared, wherein the concentrations of both theophylline and caffeine were varied. For pNIPA, concentrations of caffeine ranged from 4.1 to 21 μM over six solutions and concentrations of theophylline spanned 8.4 to 24 μM over five solutions per caffeine concentration. Likewise, for HPC, the caffeine and theophylline concentrations ranged from 0 to 9.1 mM and 0.2 to 6.1 mM, respectively.

3.7.4 Instrumentation

Ultrasound transducers for sending and receiving pulses were affixed to the sample reservoir as shown in Figure 3.5. The sample cells used were a 1.8 cm plexiglas cuvette for the HPC solutions and a 3.8 cm aluminum cell for the pNIPA solutions. A 1.9 MHz narrow-band transducer (Advanced Technology Labs Inc., Pennsylvania, U.S.) generated ultrasound pulses and a 10 MHz wideband transducer (Optel Inc., Wrocław, Poland) received the ultrasonic signal. A Panametrics 500PR Pulser/Receiver (Panametrics Inc., Massachusetts, U.S.) as the pulse generator for the

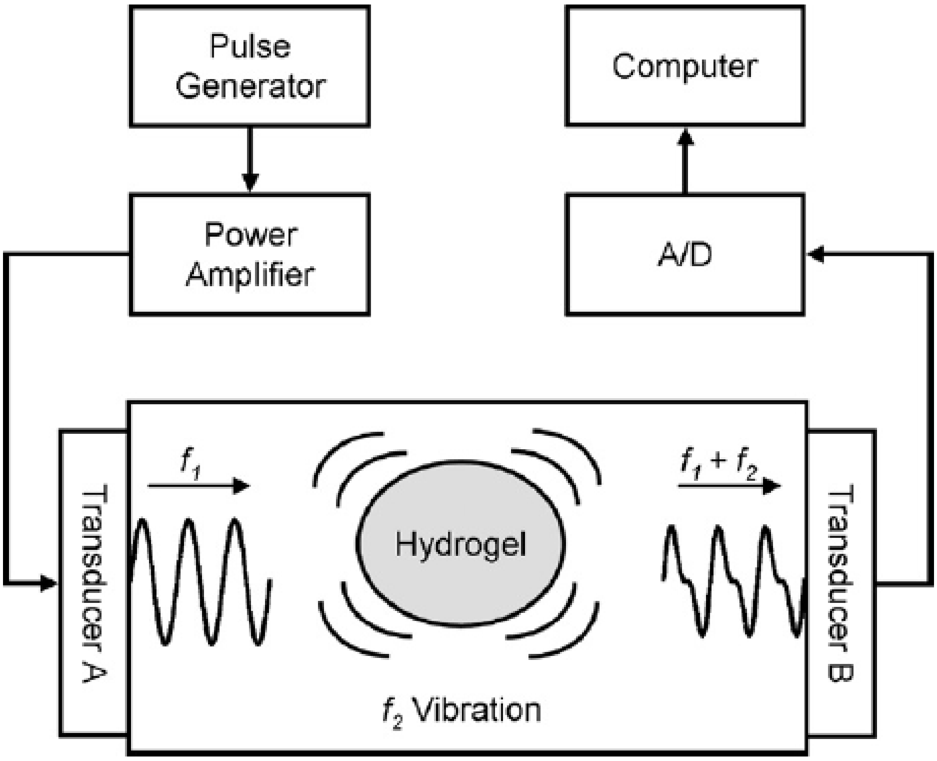

Figure 3.5. Schematic for ultrasound data acquisition. A pulse generator and power amplifier are used to drive an ultrasound transducer. Transducer A generates pulses at frequency f_1, which travel through the sample cell and interact with the nanogel sensors. Transducer B on the parallel end of the sample cell records the ultrasound pulse that includes frequency f_2 cause by the resonating polymer. A computer controlled oscilloscope digitized the signal for further analysis.

transducers. The Panametrics 500PR generated 10 ns 250 V negative impulses to drive the 1.9 MHz narrow-band transducer. A SDS 200 oscilloscope (SoftDSP Co., Seoul, Korea) sampling at 12.5 MHz using 9 bit A/D conversion, and Handyscope HS3 (TiePie Engineering, Sneek, Netherlands) sampling at 50MHz using 12 bit A/D conversion, collected the ultrasonic data from the 10 MHz wideband transducer for the pNIPA and HPC, respectively.

3.7.5 Data Processing

Ultrasound waves propagating through the sample cell were measured by the receiving transducer and digitized by the computer controlled oscilloscope. Data for each sample were acquired over 3 minutes, for a total number of 4000 waveforms. Each sample was analysed in triplicate. Likewise, the complete quantification analysis was also repeated three times. These waveforms were averaged to increase the signal to noise ratio. A fast Fourier transform algorithm was then applied to the averaged data to allow processing in the frequency domain. Variability due to instrumental and temperature changes was minimized by total area normalization, and random noise fluctuations were removed using a boxcar smoothing function. Frequency spectra were divided into independent calibration and test sets. The test set consisted of a series of samples with varying theophylline levels at one caffeine concentration that was not found in the calibration data set.

Stagewise multilinear regression (SMLR) was used to estimate the concentration of theophylline based on the magnitude of the ultrasound frequencies measured for a given sample. The algorithm determined the regression of the magnitude at each frequency with the analyte concentrations in the calibration set in order to determine the highest correlation. The residual values are then calculated and the process is repeated iteratively with the subset of frequencies not yet included in the model. Based on the linear combination of this subset of ultrasonic frequency intensities, the data are described in the form

$$Y = \mathrm{b}_0 + \mathrm{b}_1 X_1 + \mathrm{b}_2 X_2 + \ldots + \mathrm{b}_\mathrm{n} X_\mathrm{n} \qquad \text{Eq. 3.6}$$

where Y is the dependant variable, {X} are independent variables, and {b} are weighting coefficients. This multilinear model was subsequently used to estimate the concentration of the analyte in the independent test set samples. To avoid overfitting of data by the model, parsimony was determined by F-test ($\alpha = 0.05$) between the standard errors of each model so that the addition of additional parameters would not significantly impact the SMLR model and estimation.[72] All data analysis was done in Matlab (The MathWorks Inc., 2008a, Massachusetts, U.S.).

3.8 Results and Discussion

To examine the relationship of the ultrasound signal with analyte binding in the nanogels, we have quantified the target analyte sensitivity in distilled water. In both HPC and pNIPA nanogels, the concentration of theophylline was estimated with three frequencies in the derived model. In the low affinity binding HPC, concentrations of theophylline were determined in the millimolar range. A standard error of estimate (SEE) of 0.1 mM with a correlation coefficient (r^2) exceeding 0.99 was obtained in the 0.1 to 6.1 mM range. With theophylline binding, the mean particle size of the HPC nanogel increased by 14%. The higher affinity binding of pNIPA permitted using a micromolar theophylline concentration range, with a SEE of 2.6 µM and an r^2 exceeding 0.99 in the 8.4 to 167 µM range. Unlike the HPC sensor, binding to theophylline resulted in a 16% decrease in overall sensor diameter. The decrease in size suggests that the stronger hydrogen bonding between pNIPA and theophylline results in a tighter induced fit in the molecularly imprinted binding pocket. The greater structural changes would cause larger changes to the resonance frequency of the pNIPA nanogel, increasing the sensitivity. Methods of storing pNIPA nanogel while still retaining quantification capability at a later time were investigated next.

Two storage methods of imprinted pNIPA nanogels and their effect on analyte quantification were analysed. Theophylline imprinted pNIPA nanogels were synthesized, and concentration models constructed for the 8.8 to 44 µM range using the same synthesis procedures and instrumentation. The remaining pNIPA was separated into two samples: one stored at room temperature, the other lyophilyzed and stored in a chemical freezer. Both samples were left in their respective storage facilities for 6 months. After this time period, the lyophilyzed sample was redissolved in equal amounts of water it originally contained. Two sets of calibration solutions containing 8.8 to 44 μM theophylline were created, and SMLR concentration models were constructed for each set. The results are displayed in Figure 3.6, with regression statistics and selected frequencies in table 3.1. Similar frequencies in all three cases were selected for concentration models, and high degrees of linearity were obtained. A few minor differences exist when comparing the three datasets, such as slightly different standard errors, and a differently selected tertiary frequency in one case. Nevertheless, the results show that for a period of 6 months, lyophilyzation had no significant negative impact on the quantification ability of pNIPA nanogels compared to the original concentration model, and model determined from sample stored at room temperature. For storage time periods greater than 6 months, lyophilyzation is highly recommended as some samples stored at room temperature for more than one year tended to precipitate out of solution.

Storing pNIPA for up to six months is deemed appropriate, either by lyophilyzation or air-tight storage at room temperature. The standard errors of quantification assays conducted with both stored samples were higher than quantification prior to storage, which was expected as both ambient conditions and conditions of pNIPA samples undoubtedly changed during storage. Although longer time periods weren't explored, lyophilyzed should be able to retain the same quantification ability longer than six months. This assumption is based on concentration models constructed for the lyophylized sample that were achieved using the exact frequencies of the pre-storage sample, as shown in table 3.1. The standard

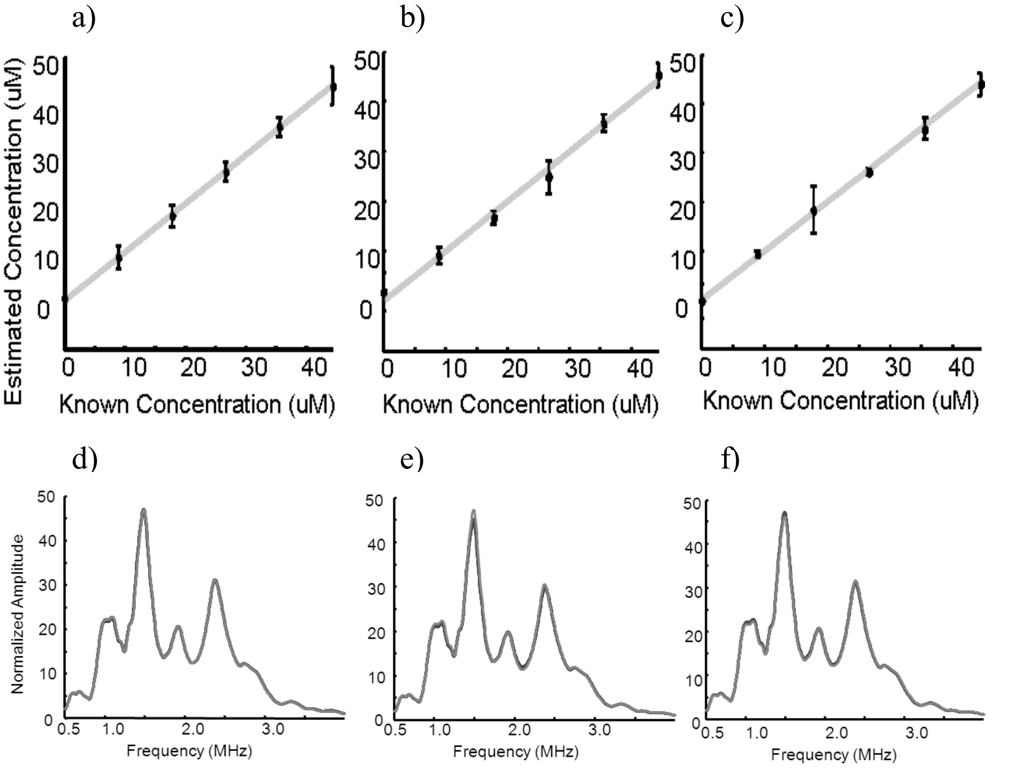

Figure 3.6. Comparison of estimated and known theophylline concentration for three concentration models constructed using SMLR. Amplitude changes at 3 frequencies were employed. Imprinted nanogels made from pNIPA used for ultrasonic theophylline quantification for each assay were treated as follows: a) represents quantification before storage, b) represents quantification after 6 month storage at room temperature, and c) represents quantification after lyophilyzation of nanogel and subsequent 6 month in a chemical freezer. Ultrasonic spectra for the three concentration models are shown in d) before storage, e) 6 months at room temperature, and f) lyophilyzed and stored 6 months in a chemical freezer. Correlation coefficients (r^2) for all three regression models were determined to be 0.99. Leave-one-out cross validation was used to assess model robustness. Details regression statistics are shown in table 3.1.

	a) Pre storage	b) Stored 6 months, room temperature	c) Stored 6 months, lyophilyzed, freezer
Frequencies selected for SMLR model	1.4, 2.4 and 2.1 MHz	1.4, 2.4 and 2.0 MHz	1.4, 2.4 and 2.1 MHz
Correlation coefficient (r^2)	0.99	0.99	0.99
Standard Error	2.1 μM	2.6 μM	2.5 μM

Table 3.1. Multilinear regression statistics for data plots in Figure 3.6. Frequency selection to construct slope coefficients for multilinear regression equations to quantify theophylline are all identical, with the exception of the tertiary frequency selected for regression in b). Although for a six month period both storage methods yield similar regression statistics, lyophilyzation of imprinted pNIPA nanogels results in slightly lower standard error, limit of detection, and offers the possibility of storage for much longer periods.

error and limit of detection associated with the lyophylized sample was also lower than the sample stored at room temperature, illustrating that lyophilyzation of pNIPA nanogel samples for future quantification is the choice method of storage resulting.

An interfering species was used in order to assess the specificity of the molecularly imprinted nanogels. A commonly used pair of chemicals to determine sensitivity and specificity of imprinted nanogels is theophylline and caffeine, which are molecularly-similar xanthine derivatives.[64, 73] Presence of an additional N-methyl group on caffeine decreases hydrogen bonding potential and increases steric hindrance for molecularly imprinted binding sites resulting in a lower binding constant. Solutions were prepared in which theophylline and caffeine of comparable concentrations were independently varied. Caffeine concentrations ranged from 0 to 9.1 mM in HPC and from 0 to 21.0 μM in pNIPA. Likewise, concentrations of theophylline were ranged from 0.2 to 6.1 mM and from 8.4 to 24.0 μM for HPC and pNIPA, respectively.

To estimate the concentration of theophylline independently of caffeine, separate calibration models using SMLR were created for HPC and pNIPA sensors. Models were tested on an independent evaluation set consisting of one concentration of caffeine excluded from the calibration data. For low affinity HPC, the use of six frequencies, 1.2, 2.2, 1.7, 1.1, 2.8, and 1.6 MHz, was determined to be the most parsimonious for the quantification of theophylline and resulted in a SEE of 0.6 mM with a corresponding r^2 of 0.95. Figure 3.7(a) illustrates the typical frequency spectrum of HPC nanogels, as well as the frequencies used by the multilinear model for theophylline estimation. The concentrations of theophylline estimated by the SMLR model are plotted against the known values in Figure 3.7(b), showing linearity of the full range with minimal bias. The most parsimonious estimation of theophylline in high affinity pNIPA nanogels was obtained with five frequencies, 1.8, 2.0, 1.7, 2.2, and 1.3 MHz. A SEE of 1.5 μM with a corresponding r^2 of 0.98 was obtained. Figure 3.7(c) illustrates the typical frequency spectrum of pNIPA nanogels, as well as the frequencies used for theophylline estimation. The concentrations of theophylline estimated by the SMLR model are plotted against the known values in Figure 3.7(d), showing linearity of the full range with minimal bias. The difference in selected frequencies and correlation coefficients for multilinear quantification in both nanogels can be partly explained by the different analyte binding strengths and levels of interferant competition. Two nearly identical molecules competing for the same localized hydrogen bonding sites can alter with the nanogel resonant frequencies. The low affinity HPC nanogel required an addition frequency component to quantify

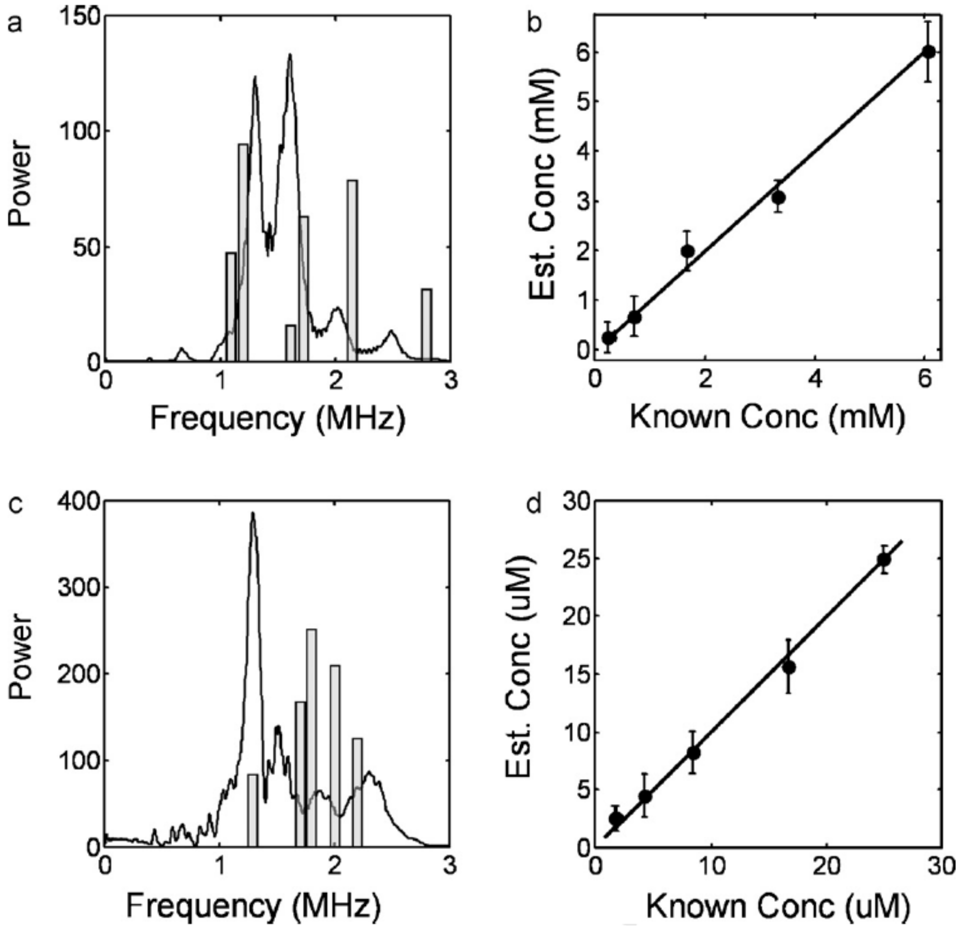

Figure 3.7. Theophylline quantification results in presence of caffeine. a) ultrasonic frequencies selected to determine analyte concentrations in HPC, b) estimates of theophylline concentrations with HPC sensors using a multilinear model plotted against known values, c) ultrasonic frequencies selected to determine analyte concentrations in pNIPA, d) estimates of theophylline concentrations with pNIPA sensors using a multilinear model plotted against known values.

theophylline with caffeine due to the greater impact an interferant has on a less specific analyte binding environment. Results show that calibration was possible for both nanogels independent of caffeine interference with an overall linear quantification spanning the micromolar to millimolar range.

To determine if simultaneous analyte quantification was possible using the ultrasound signal, concentration of caffeine in the above theophylline/caffeine mixtures was estimated. Chemical similarity between the two xanthines allows both of the compounds to access the binding sites, albeit with separate binding constants. The decreased interaction between caffeine and the molecularly imprinted pocket as compared to theophylline should induce distinctly different physical changes in the nanogel. As with theophylline, sensitivity of HPC to caffeine was lower than in pNIPA, and for this analyte, proved to be too low for quantification (r^2<0.6). The mean particle size increased by 23% with caffeine binding, compared to 14% with theophylline. It is possible this greater increase in particle size decreased the resonance frequency of the HPC nanogels so that the bandwidth of the ultrasonic transducers was not optimal for the determination of caffeine-bound sensors.

Simultaneous caffeine quantification was possible with the pNIPA sensor using five frequencies resulting in a SEE of 3.3 μM and an r^2 of 0.87. The mean particle size for caffeine binding with pNIPA showed only a 5% decrease, as compared to the 16% decrease with the theophylline molecule. This suggests that while the hydrogen bonding may still have formed strong associations, the presence of the additional methyl group likely resulted in a steric strain on the pocket, preventing certain conformational changes that result when the template molecule occupies the pocket. Lower correlation coefficients were expected when quantifying caffeine in this scenario, as the nanogels were molecularly imprinted with a theophylline template which is structurally different than caffeine. Results are consistent with the proposed model and provide a mechanism for simultaneous quantification of multiple analytes with one nanogel sensor. With tuning of the imprinted nanogel using different binding mechanisms such as, hydrophobicity and π stacking, high sensitivity and specificity should be possible for several analytes.

3.9 Conclusion

We have successfully quantified theophylline in solutions ranging between 8.4 μM and 6.1 mM using two molecularly imprinted nanogels. This allows physiological monitoring of theophylline within the therapeutic range (55 μM to 110 μM), as well as higher concentrations which are toxic.[74] The model for contrast agent resonant frequency suggests that the change in the ultrasonic signal with increased analyte concentration is likely due to changes in nanogel physical properties, notably modulus and radius. Furthermore, it was also shown that this molecular quantification is possible in a matrix containing an interfering species with a nearly identical chemical structure. Measurement of multiple frequencies allowed quantification of an imprinted analyte in a matrix containing an interfering analyte with a nearly identical chemical structure. Molecular imprinting of pNIPA nanogels provides a greater analyte selectivity, which is attributed to the presence of the MAA binding group. Likewise, increased specificity due to stronger binding results in a nanogel more adaptable to analyte molecules and analogous species. These findings are strongly indicative that the molecular imprinting process provided the selectivity required to implement a detection system using ultrasound. It is also possible to quantify the interfering species by looking at different frequencies of the ultrasonic response. This opens up several possibilities for simultaneous detection of multiple compounds with a single templated molecularly imprinted nanogel. Overall, ultrasonically detected smart nanogel biosensors appear very promising for a variety of environmental, industrial and clinical applications.

3.10 Acknowledgements

This work was supported in part by the National Science and Engineering Research Council of Canada and by the Fonds Québécois de la Recherche sur la Nature et les Technologies.

Chapter 4.

Quantification of Dopamine Oxidation Product Using Nanogel Contrast Agents

In the previous chapter, two molecularly imprinted nanogels were investigated with different analyte binding strengths. The HPC nanogel was shown to have a lower analyte binding affinity, whereas the pNIPA nanogel had higher binding affinity. Both nanogels were synthesized in the swollen state with analyte present. An interesting avenue to further develop analyte binding sensitivity in molecularly imprinted nanogels would be to carry out pNIPA imprinting in the collapsed phase, leading to a tighter binding arrangement between nanogel and analyte. This could lead to greater signal differences between presence and absence of analyte.

This chapter investigates ultrasonic analyte quantification using molecularly imprinted pNIPA nanogels synthesized in the collapsed state. Dopamine was selected as the analyte for this quantification assay. The therapeutic monitoring range for dopamine is several orders of magnitude lower than theophylline quantified in the previous chapter. The therapeutic range for dopamine lines in the 13 to 130 nM range, compared to the low micromolar therapeutic range for theophylline.[75] Changes in nanogel solution composition were required for sub micromolar quantification of dopamine. Chief among these changes was to reduce the percentage of nanogel by weight in each sample to allow for signals indicative of sub micromolar concentration changes.

The linear dynamic range of analyte sensitive pNIPA nanogels can be controlled through percent loading of nanogel in samples. The ultrasound signal components emerging from the number of pNIPA nanogel particles containing docked analyte at any one time. If pNIPA is added to samples in high loading percentage, signals from a few analyte molecules docked into imprinted pNIPA particles will not cause significant change in overall signal, since the majority of signal will stem from pNIPA particles without docked analyte. This scenario leads to a dynamic range in the higher analyte concentrations, such as micromolar amounts. By lowering the weight percentage of pNIPA in each sample, lower concentrations of analyte will have a more visible impact on overall signal. In this scenario, signal from analyte molecules docking into pNIPA particles would be greater relative to pNIPA particles without docked analyte. This leads to more favorable conditions for mid nanomolar analyte concentration ranges. In order to build concentration models for dopamine in the mid nanomolar range, the weight percent loading of pNIPA in each sample was reduced from 1.0 % to 0.8%.

Dopamine itself also provided an interesting challenge as it is readily oxidized, which can occur spontaneously or catalyzed by enzymes.[76] Current research for dopamine quantification includes the use of boron-doped microelectrodes which must be in direct contact with any dopamine containing sample.[77] The goal of this chapter was to determine if dopamine could be quantified through a dopamine oxidation product, 5-6-dihydroxyindole (DHI), using minimally invasive analyte sensitive ultrasound contrast agents. An additional aspect of interest was accuracy of analyte sensitive agents in the presence of a albumin, a biological interferant found in many biofluids and other clinical samples.

4.1 Manuscript

Quantification of Dopamine Oxidation Product using Smart Ultrasound Sensors from Nanogel Contrast Agents

Authors:

David Troïani and David H. Burns

4.2 Abstract

Quantification of analyte with optical instrumentation in environments containing matrix effects and spectral overlap presents multiple challenges. Ultrasound has been shown to easily penetrate samples which are complex to measure using optical wavelengths, however no molecular information is obtained. We have developed a general approach for an ultrasonic quantification system that uses smart biosensors based on molecularly imprinted nanogel polymers. Multilinear analyte quantification in aqueous nanogel solution led to the construction of a concentration model for dopamine through an oxidation product of dopamine (5-6-dihydroxyindole, DHI) from 16.7 to 163 nM. High degrees of linearity (correlation coefficient exceeding 0.99) were obtained. Additional assays were conducted in the presence of biologically relevant interfering species in the 16.6 to 150 nM range. Results provide correlation coefficient of 0.96 with the interfering species present in solution.

4.3 Keywords

ultrasound, nanogel, molecular imprinting, n-ispropylacrylamide, dopamine, 5-6-dihydroxyindole, biosensor, analyte recognition

4.4 Introduction

4.4.1 Background

Optical instruments are widely used to conduct analyses in various fields of chemistry and biochemistry. Many samples analysed in this manner can have matrix effects and overlapping components, making it difficult to quantify analytes properly. Ultrasound can easily penetrate samples that are challenging to measure optically, however little to no molecular information is obtained. Recently developed contrast agents improve the capabilities of ultrasound and enhance ultrasound echogenicity.[78] Contrast agents are composed of spherical gas-filled microbubbles that resonate at characteristic frequencies. Nanogels, such as N-ispropylacrylamide co-polymers (pNIPA), could be used as ultrasound contrast agents due to ultrasonically sensitive properties. These polymers undergo reversible volume phase transition between swollen and collapsed states in solution. In the swollen state, nanogels exist as a diffuse network of interconnected chains. In the collapsed state, nanogels take the form of compressed microparticles in solution, with increased turbidity and stiffness compared to swollen state. Changes in ultrasonic properties accompanies transition from swollen to collapsed state.[51] Transition can be triggered by environmental factors such as temperature, ionic strength, hydrostatic pressure, or hydrogen.[79, 80] When increasing temperature past critical threshold T_c (33.6 °C),[51] broad frequency changes are noted (Figure 4.1). Frequency changes appear over the entire spectrum, in both amplification and attenuation of signal. We have developed a general approach for ultrasonic molecular quantification using selective analyte sensors created from molecularly imprinted nanogel polymers that selectively bind to specific analyte molecules. Analyte recognition alters the equilibrium size and modulus of nanogel sensor particles, inducing specific changes in ultrasonic frequency spectrum.

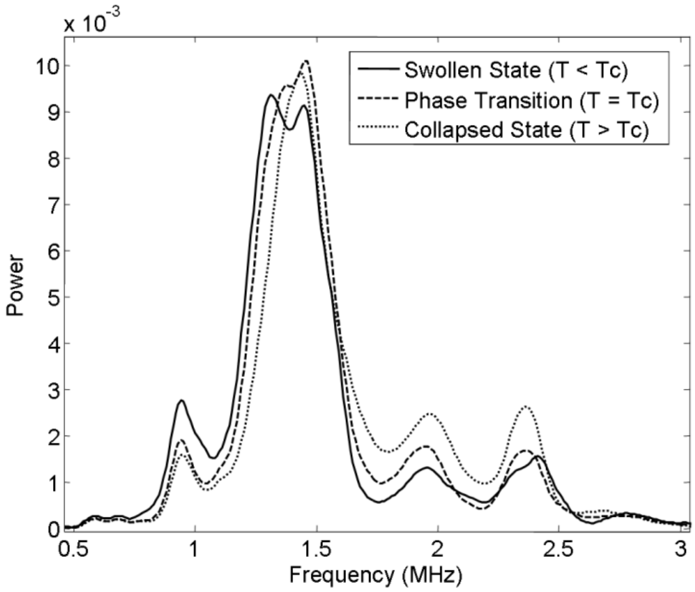

Figure 4.1. Multiple frequency interaction of pNIPA nanogels undergoing phase transition as a result of increasing temperature. Ultrasound spectra are shown before (T < T_c), during (T = T_c, 33.6 ^{o}C), and after (T >T_c) phase transition relative to critical threshold temperature T_c.

Analyte sensitive nanogels can be developed through specific molecular recognition. Methods for molecularly imprinting nanogels are established in the literature.[81, 82] pNIPA nanogels are synthesized by first allowing self assembly of pNIPA monomers around analyte molecules prior to polymerization, creating molecularly imprinted pockets (Figure 4.2). Analyte docking into binding sites momentarily alters physical properties, such as local rigidity, size, and ultrasonic properties of the nanogel network. While an analyte molecule remains in a binding site for only a short time, the effect of many analyte molecules interacting

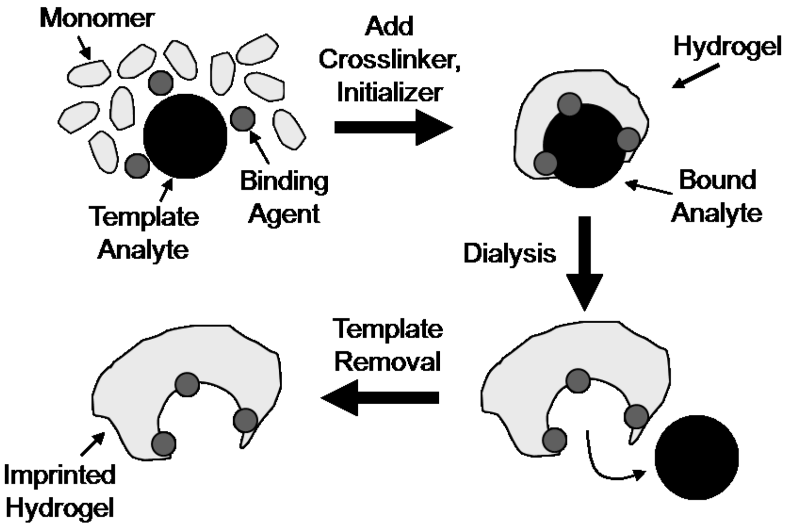

Figure 4.2. Nanogel monomers self assembling around template analyte. Addition of crosslinker and initializer begins the polymerization process around template analyte. Template is removed through dialysis, leaving only molecularly imprinted nanogel particles remaining.

with multiple sites over numerous nanogel particles leads to minute, but measureable ultrasonic profile changes. Quantification is accomplished using recorded ultrasonic data and multilinear regression (MLR) algorithms written in the Matlab programming language. Analyte concentration models are then constructed using characteristic changes in ultrasound response at multiple frequencies. In this paper, we demonstrate quantification of dopamine through an oxidation product of dopamine (5-6-dihydroxyindole, DHI). Dopamine is rarely analysed directly in samples as oxidation can occur spontaneously, or catalyed by enzymes, *in vivo*.[83] Analysis of dopamine through DHI has been established and provides an alternative to dopamine quantification in biologically relevant samples.[84, 85] Multiple DHI quantification experiments were carried out in solution with good linearity in the nanomolar range, and additional trials were conducted in the presence of an interfering species.

4.4.2 Theory

A model can be developed to illustrate the effect of analyte binding on an ultrasonic spectrum by considering nanogel particles to be deformable microbubbles in solution. Since microbubbles have specific resonant frequencies when perturbed by ultrasound, resonant frequency shifts can be observed when altering microbubble size and/or stiffness, much like plucking on guitar strings of different thickness and length produces different sounds. Imprinting nanogels for particular analytes can change nanogel physical properties, such as size and stiffness, depending on the amount of analyte present. Docking of an analyte into molecularly imprinted binding sites is done through hydrogen bonding with the binding agent MAA incorporated into a nanogel polymer network. Research has shown that molecularly imprinted nanogels increase in stiffness and decrease in particle size as analyte molecules occupy binding sites, with a greater number of contributions attributed to more analytes docked in binding sites.[86, 87] This model is an approximation of nanogel interaction with ultrasound in solution, assuming nanogel particles to be hollow microspheres with very thin shell thicknesses. Hollow thin-shelled microbubbles are relatively straight forward to model and approximate nanogel behavior well enough to visualize interaction between ultrasound and nanogel with localized changes in stiffness due to nanogels' molecularly sensitive shell.

Ultrasound waves cause isotropic oscillation in nanogels at their resonant frequencies from repeated compression and rarefaction. If nanogel particles are deformable microbubbles with resonant frequency f_0, specific particle size at equilibrium a_e, and bulk modulus K_p (proportional to overall stiffness), a relationship between these parameters can be expressed in the following form,

$$f_0 = \frac{1}{2\pi \quad a_e}\sqrt{\frac{3K_p}{\rho_L}} \qquad \text{Eq. 4.1}$$

known as the Minnaert equation, where ρ_L is the density of the surrounding liquid. As the bulk modulus of a microbubble increases, proportional to an increase in overall stiffness, microbubble resonant frequency shifts to higher frequencies. Increasing microbubble equilibrium radius results in a resonant frequency shift to lower frequencies. Imprinted nanogels will have a defined resonant frequency in the absence of analyte, illustrated by point A in Figure 4.3. When an analyte interacts with a binding site, localized stiffness and nanogel particle size can be affected, leading to a shift in resonant frequency (point B, Figure 4.3). Complex interrelationships between the two non-linear parameters leads to specific frequency profiles for

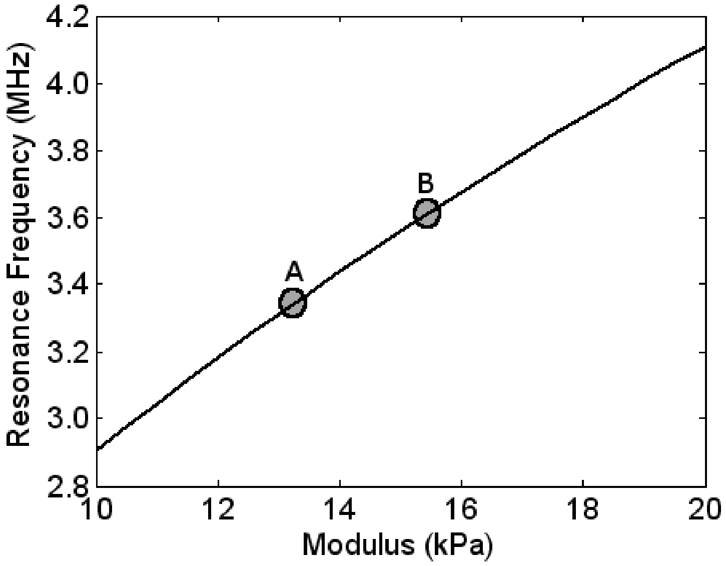

Figure 4.3. Resonant frequency of a sub-micron sized microsphere as a function of modulus (with fixed radius) based on Minnaert equation. A given microsphere will have a resonant frequency as illustrated at point A. Binding of analyte into a molecularly imprinted pocket can result in a modulus increase and therefore shift resonant frequency to point B.

nanogels dependant on analyte concentration, the analyte of interest, and nanogel molecular composition. Nonlinear propagation of pulsed ultrasound in aqueous media at high acoustic pressures is known to broaden ultrasonic frequency profiles,[88] therefore multiple resonance frequencies could be probed with one pulse. From measurements of multiple frequencies within the first transmission pulse through the sample cell, we calculated the concentration of analyte absorbed into the nanogel pockets.

4.5 Materials and Methods

4.5.1 Synthesis of DHI Imprinted Nanogel

N-isopropylacrylamide (pNIPA, 500 mg), methylene-bis-acrylamide (MBA, 50mg), methacrylic acid (MAA, 50 uL) and dopamine hydrochloride (50 mg) were added to 60 mL previously boiled milli-Q water. The solution was deoxygenated with nitrogen and stirred overnight. Azo-bis-isobutyronitrile (AIBN, 20 mg) initiator was added to the mixture, and the solution was deoxygenated with nitrogen and stirred for 4 hours at 65^{o}C. Upon cooling, the resulting polymer solution was heated to 36^{o}C and phase transition was confirmed visually by an increase in turbidity. The solution was dialyzed to remove DHI template. Extraction of DHI was confirmed spectroscopically at 220 nm.[89] Mean particle size was verified to be 164 nm by dynamic light scattering using a Malvern Instruments Zetasizer Nano ZS (Malvern Instruments, USA). All chemicals were purchased from Sigma-Aldrich (Ontario, Canada).

4.5.2 Nanogel Solutions

pNIPA nanogel solutions were used to estimate DHI concentration based on ultrasonic response. Six solutions containing 0.8 % by weight pNIPA nanogel in deionized water were made. DHI was added to pNIPA solutions from 16.7 to 163 nM.

To assess the selectivity of the imprinted nanogel, a second set of solutions were prepared in the 16.6 to 150 nM range with bovine serum albumin (BSA) used as an interfering agent. BSA (25 mg/mL) was added to each nanogel solution similar to concentrations found *in-vivo*.[90]

4.5.3 Instrumentation

A sample cell was constructed with two ultrasound transducers affixed to each end. The sample pathlength was 1 cm. Clinical ultrasound gel was applied for coupling. A 1.9 MHz narrow-band transducer (Advanced Technology Labs Inc., Pennsylvania, U.S.) generated ultrasound pulses and a 5 MHz wideband transducer (Russel NDE Systems, Alberta, Canada) received the ultrasonic signal. Electrical signals were generated with a Panametrics 500PR Pulser/Receiver (Panametrics Inc.,

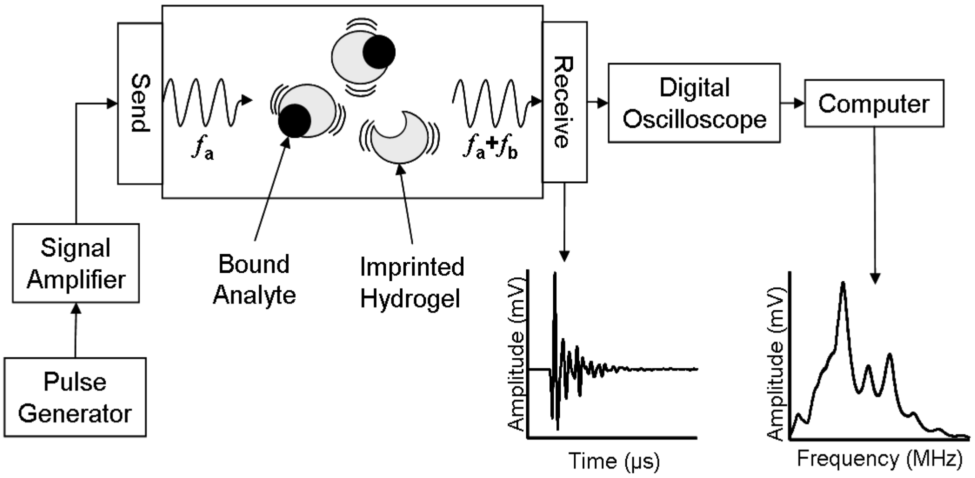

Figure 4.4. Simplified view of ultrasound instrument setup. A pulse generator produced 2 μs long pulses at a rate of 500 Hz. For each pulse, the 1.9 MHz transducer generated 3 cycles of ultrasound. Pulses traveled through the sample cell and interacted with the nanogel microspheres. A 5 MHz wideband transducer on the other end of the sample cell detected the ultrasound transmission.

Massachusetts, U.S.), serving as the pulse generator for the transducers. Signals were digitized with a Handyscope HS3 (TiePie Engineering, Sneek, Netherlands) sampling at 50MHz using 12 bit A/D conversion. The complete system is shown in Figure 4.4.

4.5.4 Data Processing

Ultrasound data received from the 5 MHz wideband transducer were processed to develop frequency spectra for each trial. Discrepancies in phase offset were corrected prior to frequency spectra determination via Fourier transform. Each sample was analysed in triplicate. In addition, the entire quantification assay was also repeated three times. Frequency spectra from replicate measurements were seperated into calibration sets used to develop concentration models, and independant validation test sets. Validity of the concentration model was assesed by using the model to calculate concentrations of solutions in the independant test set. Stagewise multilinear regression (SMLR) was used to determine the linear combination of select frequencies that best describe the data in the form of

$$y = b_0 + b_1x_1 + b_2x_2 + ... + b_nx_n \qquad \text{Eq. 4.2}$$

where y is the dependant variable, {x} are independent variables, and {b} are the coefficients determined. Model parsimony was determined using F-tests (α=0.05) carried out with the standard errors of each developed model held against each subsequent model containing an additional {b} coefficient.[72] All software used to determine concentration models was written in Matlab (The MathWorks Inc., Massachusetts, U.S.). A flowchart of experiment and processing procedures is illustrated in Figure 4.5.

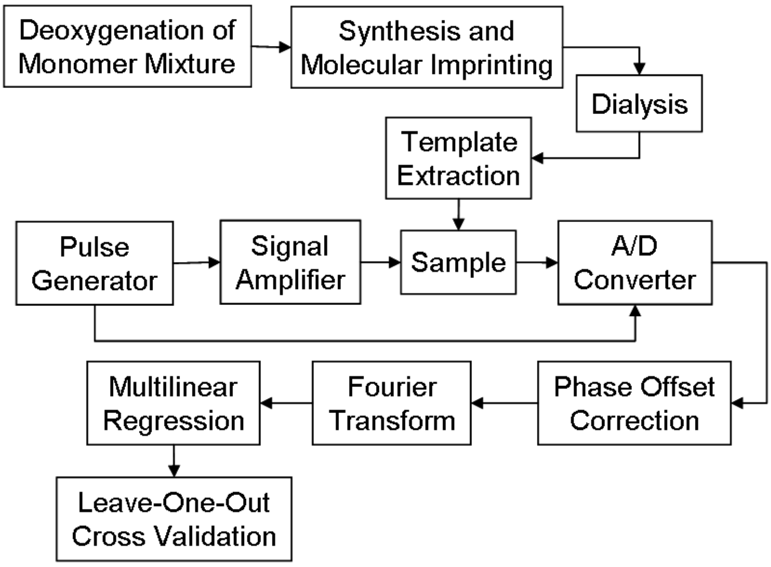

Figure 4.5. Flowchart of experiment and processing procedures.

4.6 Results and Discussion

The relationship between ultrasonic response and physical properties of the nanogel network is key to quantifying analytes using the ultrasound system. Nanogels undergoing phase transition will alter microparticle radius, stiffness and structure, which are responsible for the observed attenuation and amplification in signal shown in Figure 4.2. In molecularly imprinted nanogels, docking of analyte molecules into molecular binding sites can produce minute nanogel network changes comparable to phase transition, however on a much smaller scale. The local nanogel network surrounding the bound analyte stiffens, which can cause a corresponding change in ultrasound. Although a phase transition event due to all imprinted binding sites being simultaneously occupied is unlikely, minute ultrasound signal changes are large enough to quantify analyte presence using multilinear regression analysis.

We have investigated the relationship between ultrasonic response and analyte concentration in pNIPA nanogels using solutions free of interfering species. Dopamine stock was left to oxidize, and the resulting DHI was added to imprinted nanogel solutions resulting in DHI concentrations of 16.7 to 163 nM. Imprinted pNIPA nanogel was added to each solution, and DHI concentration was estimated using multilinear regression models based on amplitudes at three ultrasonic frequencies, specifically 0.5, 1.2 and 2.0 MHz in order of significance. Concentration models derived from these frequencies show high linearity in the nanomolar range, with a correlation coefficient (r^2) of 0.99, with 7.8 nM standard error of estimate (SEE). Figure 4.6 a) shows the profile of the selected ultrasonic frequencies and b) illustrates the calibration curve of the constructed model. Determining the specificity of the analyte imprinting process in pNIPA required addition of a commonly used biological interfering species, bovine serum albumin (BSA), during quantification. Imprinting specificity was investigated by adding BSA to each solution in physiological concentrations[90] prior to ultrasonic quantification of

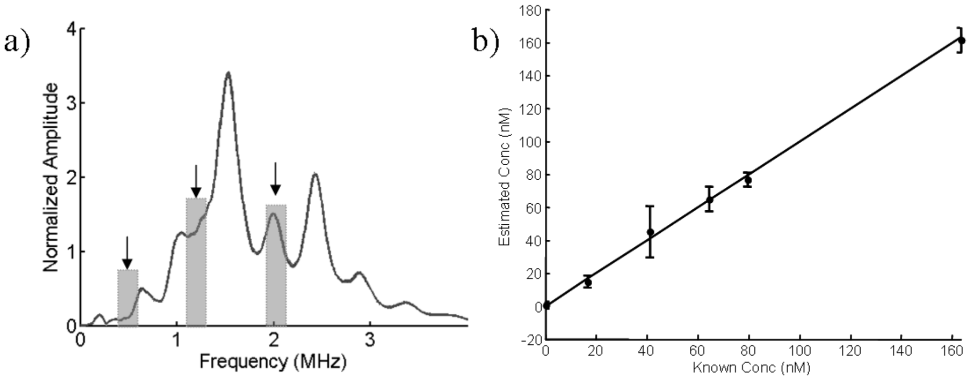

Figure 4.6. DHI Quantification results using ultrasound and molecularly imprinted pNIPA nanogel in an intereference free matrix. Figure a) refers to frequency profile and SMLR frequency selection for DHI concentration model construction. The model is illustrated in b), in the 16.7 to 163 nM range.

DHI. Concentrations of DHI in this assay ranged from 16.6 to 150 nM. Results from quantification assays in presence of BSA are visually very different than previously conducted matrix free assays. The most prominent feature of adding a biological interferent of such large mass in the ultrasonic spectrum is overall dampening of the area surrounding the excitation frequency (1.9 MHz), as seen in Figure 4.7 a). As a consequence of this dampening, SMLR peaks correlating amplitude with analyte concentration were selected in the outskirts of the spectrum, most notably at roughly the harmonics (0.7 and 3.2 MHz) of the excitation frequency. Although the peak-to-peak ratio for large peaks in both profiles is similar, smaller details concentrated near the center of the profile are de-emphasized by the dampening effect of BSA interference. The outer areas of the profile have minimal signal dampening from BSA, still emphasizing quantitative information regarding DHI concentration. The concentration model obtained a correlation coefficient of 0.96 and a corresponding standard error of 15.8 nM, illustrated in Figure 4.7 b).

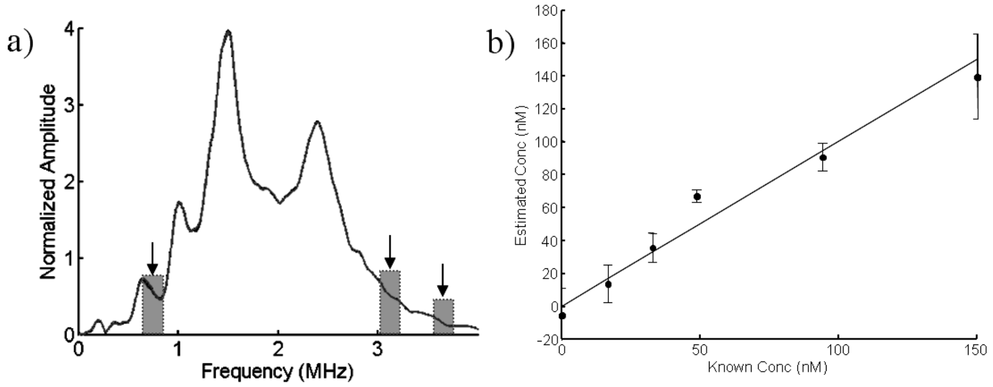

Figure 4.7. DHI Quantification results using ultrasound and molecularly imprinted pNIPA nanogel with BSA added in 25 mg/mL amounts. Figure a) refers to frequency profile and SMLR frequency selection for DHI concentration model construction. The model is illustrated in b), in the 16.6 to 150 nM range.

4.7 Conclusion

Using molecularly imprinted pNIPA nanogels, we have successfully quantified DHI in solutions ranging between 16.7 and 163 nM. This range overlaps with the therapeutic range for dopamine, 13 to 130 nM,[91] and is therefore satisfactory for the therapeutic monitoring of dopamine through DHI. Change in ultrasonic signal with increased analyte concentration is probably due to changes of interdependent nanogel physical properties such as modulus and particle radius. We also showed that DHI quantification is possible in 16.6 to 150 nM range with the added presence of a biologically relevant interfering species in physiological quantities. These results provide support for the implementation of an analyte detection system using ultrasound and molecularly imprinted pNIPA nanogels in clinically relevant settings through analysis of multiple frequency interactions in ultrasonic profiles. Overall, ultrasonically detected nanogel biosensors appear promising for a variety of clinical and environmental applications.

4.8 Acknowledgement

This work was supported in part by the National Science and Engineering Research Council of Canada and by the Fonds Québécois de la Recherche sur la Nature et les Technologies.

Chapter 5.

Simultaneous Ultrasonic Analyte Quantification Using Multiplexed Nanogels

The primary objective in this chapter involved combining two imprinted nanogels, each sensitive toward a different analyte. Two analyte concentration ranges differing by several orders of magnitude were investigated simultaneously using different percent loading of imprinted nanogel for each analyte. Quantification of single molecule analytes has been shown to provide concentration models with high degrees of linearity in both matrix-free solutions, and in the presence of various interferons. Simultanous multi-analyte quantification could reduce analysis time and cost, while increasing sample throughput. Previous quantification trials have shown that each analyte and respectively imprinted nanogel utilize a unique set of frequencies. Combining two differently imprinted nanogels to create multiplexed nanogels should provide simultaneous sensitivity to multiple analytes. Determining the frequency set demonstrating highest sensitivity to each analyte's concentration allowed construction of concentration models for multi-analyte analysis in the same solution.

5.1 Manuscript

Simultaneous Ultrasonic Analyte Quantitation Using Multiplexed Nanogel Contrast Agents

Authors:

David Troïani and David H. Burns

5.2 Abstract

Simultaneous quantification of multiple analytes can provide challenges using optical methods, particularly in environments with matrix effects and overlap of analyte signatures. Ultrasound easily penetrates samples that present significant challenges with optical monitoring. However no molecular information is obtained through ultrasound use. We have developed a unique method for analyte quantification using ultrasonically responsive contrast agents based on analyte sensitive nanogel polymers. Simultaneous multilinear analyte quantification of theophylline and 5-6-dihydroxyindole (DHI) resulted in a concentration regression model in the 8.4 to 49 uM range for theophylline, 48.8 to 176 nM range for DHI, using two differently sized imprinted nanogels. Good agreement between estimated and actual concentrations were obtained when analyzing unknown solutions (r^2 of 0.99 for DHI, 0.96 for theophylline) containing both analytes.

5.3 Keywords

ultrasound, nanogel, molecular imprinting, simultaneous quantification, n-ispropylacrylamide, theophylline, 5-6-dihydroxyindole, biosensor, analyte recognition

5.4 Introduction

5.4.1 Background

Analyte quantification in point-of-care settings can benifit a great deal from incorporation of phase transition nanogels. Wide variety of potential applications such as shape-memory gels,[92] drug delivery systems,[93] and molecular separation,[94] incorporates nanogels such as hydroxypropylcellulose (HPC) and poly(N-isopropylacrylamide) (pNIPA). In addition, nanogels are straight-forward and cost-effectiveness to synthesize.

At equilibrium, nanogels exist in solution as loose, swollen chain networks that freely allow solvent molecules to pass through. In this state, solvent-chain interactions keep the network swollen and flexible. When specific enviromental parameters are altered passed a critical threshold point, nanogel chain networks collapse on themselves forming tightly bound particles. In this collapsed phase, inter-chain attraction dominates over solvent-chain interactions, breaking hydrogen bonding with solvent. Several environmental parameters can produce this phenomenon, not limited to temperature, hydrostatic pressure, hydrogen bonding, ionic strength and hydrophobic interactions.[80, 95] Change from swollen to collapsed state, is associated with dramatic changes in nanogel properties in addition to collapse of polymer network. In collapsed state, nanogel stiffness and solution turbidity increase, while particle size decreases. Physical property differences between the two states allow synthesis of aforementioned shape memory gels, where co-polymers of pNIPA nanogel are synthesized with a certain conformation in the collapsed state.[92] Transition to the swollen state causes the nanogel to lose its conformation, however shape memeory is retained, and can be reversibly accessed by phase transition. Another aspect of phase transition revolves around molecules trapped in the collapsed nanogel network due to decreased particle size. Potential drug delivery systems depend on particle size difference between collapsed and swollen states to deliver encapsulated drugs to localized target locations.[93] Said drugs are then released by stimulating phase transition back to swollen state using a controllable environmental parameters.

Recent work has shown that ultrasonic properties are altered with nanogel state.[51, 96, 97] The relationship between ultrasonic attenuation and network state of pNIPA nanogels was investigated, specifically that phase transition to collapsed state induces ultrasonic attenuation at a single frequency. Results demonstrated a sharp increase in turbidity and an associated sharp, single frequency attenutation in ultrasonic amplitude past the critical phase transition temperature (33.6°C).[51] Extending spectrum of recorded frequencies utilizing a wideband ultrasonic receiver produces the ability to probe phase transition behavior at multiple frequencies. We observed ultrasonic attenuation and amplification in multiple frequencies during pNIPA nanogel phase transition, illustrating different frequency profiles for collapsed and swollen states. Results suggest that unique ultrasonic properties of pNIPA nanogel make it suitable for use as an ultrasound contrast agent. Coupled with molecular imprinting of pNIPA nanogels thoroughly explored in literature,[98, 99] the potential exists for nanogel contrast agents sensitive to a target analyte..

Contrast agents are spherical, gas-filled microbubbles that resonate at specific frequencies, unique to the type of contrast agent.[100] Probing contrast agents with ultrasound provides backscatter which is band-pass filtered to remove unwanted frequencies. Contrast agent resonance can hence be isolated, providing clear, highly resolved imaging. Ultrasound contrast agents are currently used extensively in modern medical sonography to improve imaging capabilities and enhancing echogenicity of ultrasonic imaging techniques. This technique is employed chiefly in cardiology, gynecology and other related fields where imaging of blood vessels and organ cavities are required.[101, 102, 103] However, despite the greatly improved contrast and resolution over traditional ultrasound imaging, no molecular information is obtained through ultrasonograms.

Combining molecular imprinting and pNIPA's characteristic ultrasonic properties, we have previously developed a novel analyte quantification system that relies on changes in ultrasonic profile of imprinted pNIPA contrast agents, which depend on presence of analyte, to determine analyte concentration in solution. Ultrasonic quantification has numerous potential applications, stemming from fast, minimally-invasive determination of analytes. Concentration models were constructed with multilinear regression using attenuation and amplification at select frequencies unique to each imprinted analyte. Two analyte molecules have been quantified using this method: theophylline, a bronchodilator used in treating breathing disorders,[104] and 5-6-dihydroxyindole (DHI). The latter compound is an oxidation product of dopamine, a neurotransmitter linked with depression and certain mental illnesses,[105, 106] and has been analysed to quantify dopamine in literature.[107, 108] Both theophylline and DHI have been quantified in matrix-free environment and in presence of interfering species (caffeine in theophylline, bovine serum albumin in DHI) with good linearity ($r^2 > 0.99$) in constructed concentration models.

Ultrasonic quantification of multiple analytes in one sample provides greater potential for applications in which analytes cannot be analysed separately. Simultaneous quantification is achieved using combination of two differently imprinted nanogels with different ultrasonic properties used to quantify multiple analytes (see Figure 5.1), referred to as a multiplexed nanogel. Following similar principle to our previous work, which involved amplitude changes in select frequencies in imprinted pNIPA related to presence of analyte, a system of differently sized multiplexed nanogels can be assembled to quantify two distinct analyte molecules in solution simultaneously. Interaction between the two imprinted pNIPA contrast agent and their respective analytes results in different, unique frequency interactions. With unique, size dependant sets of frequencies correlated to analyte concentrations, multilinear regression analysis can be used to focus to construct

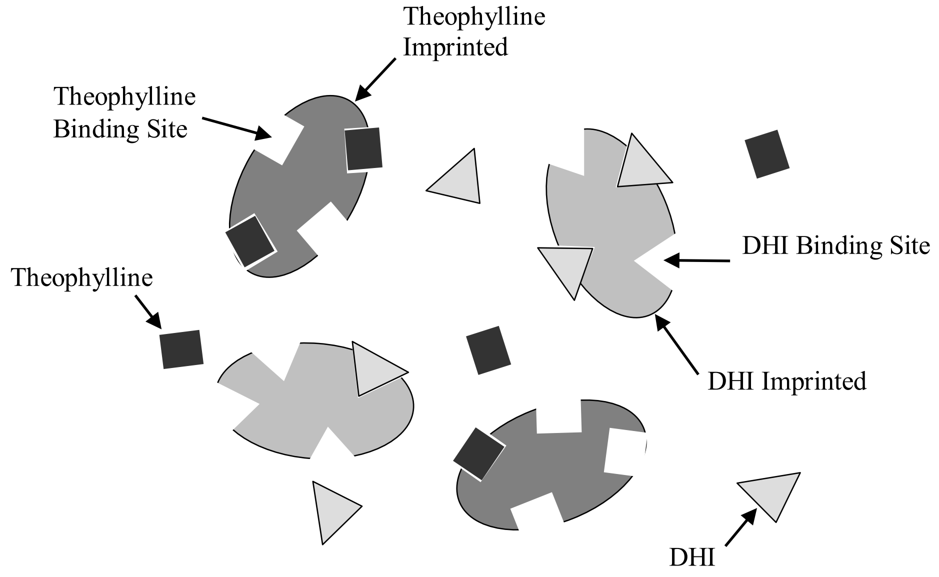

Figure 5.1. Visualization of multiplexed molecularly imprinted nanogels. Two differently imprinted nanogels are combined in solution for simultaneous analyte quantitation.

concentration models for individual analytes. A simplified version of this system is shown in Figure 5.2. A base ultrasound pulse, $F_{o,}$ is promoted through the sample cell containing pNIPA contrast agents imprinted with, for example, analyte A. Signal F_o interacts with the imprinted pNIPA contrast agent, and interactions cause a change in ultrasonic signal depending on presence of A analyte molecules docked in imprinted pNIPA particles. Particles with docked analyte A would add signal component f_A to base signal F_o, while particles without analyte contribute a different component, f_{Ao}, as shown in Figure 5.2 a). Properties and amplitude of signal f_A can be used with multilinear analysis to determine how much analyte is present in solution. A similar scenario can be derived for a two component system with a multiplexed nanogel, where f_A and f_B represent signal components added from having analytes A and B

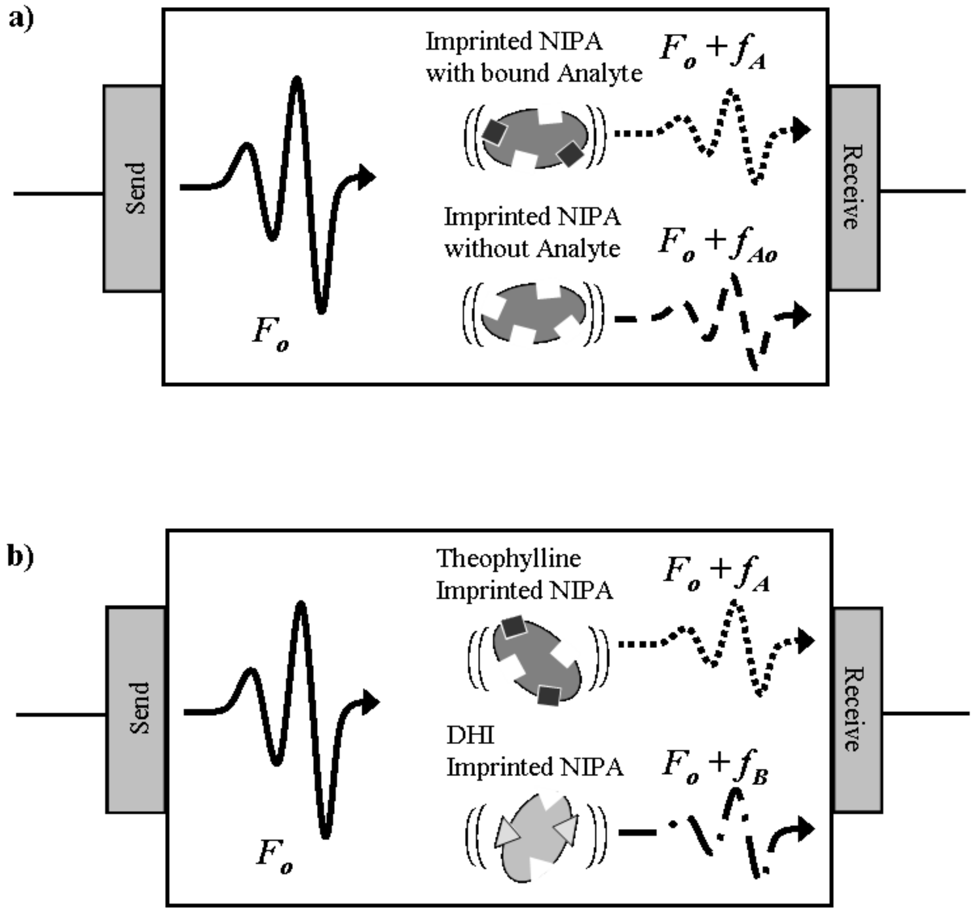
a)
Send
Imprinted NIPA with bound Analyte
Fo + fA
Imprinted NIPA without Analyte
Fo + fAo
Fo
Receive
b)
Send
Theophylline Imprinted NIPA
Fo + fA
DHI Imprinted NIPA
Fo + fB
Fo
Receive

Figure 5.2. Ultrasonic quantitation using multiplexed nanogel contrast agents. Various frequency components are added to base pulse signal F_o by molecularly imprinted nanogel. Figure a) refers to differences between components added by pNIPA particles containing bound analyte (f_A component added to signal) and without analyte (f_{Ao} component added to signal) analyte. Figure b) illustrates different frequency components added by differently imprinted pNIPA with bound analyte (f_A referring to component added by theophylline imprinted pNIPA, f_B referring to DHI imprinted pNIPA).

docked into respectively imprinted pNIPA contrast agents, as illustrated in Figure 5.2 b). Signals f_{Ao} and f_{Bo} represent components added from imprinted pNIPA contrast agents without any docked analyte. Composition and amplitude of signals f_A and f_B are used to construct multilinear concentration estimates of analytes A and B.

5.4.2 Theory

Previous work in this area focused on ultrasonic response of pNIPA nanogels using a narrowband ultrasound transducer operating in reflective mode (pulse-echo configuration).[51] Single frequency attenuation resulted from analysing pNIPA undergoing phase transition from swollen to collapsed state. In order to characterize pNIPA's ultrasonic properties further, use of a wideband ultrasound receiver was employed by our group. Analysis of pNIPA phase transition was achieved by sending a narrow frequency ultrasound pulse into the pNIPA sample, and receiving data with a wideband receiver opposite the sample cell (transmissive configuration). Interaction between pulse and pNIPA, as well as non-linear propagation of high pressure pulsed ultrasound,[109] leads to non-linear broadening of original ultrasound signal. Performing Fourier analysis on broadened signal illustrates multiple frequencies are affected by pNIPA phase transition, with both amplification and attenuation present. Broad frequency spectra of ultrasonic interactions allows visualization of multiple frequency changes present in ultrasonic profile of pNIPA. This provides an avenue for multilinear regression analysis in an environment with controlled parameters, such as a system where presence of analyte can alter frequency profile. Molecularly imprinting pNIPA nanogels is therefore a key aspect of this system, as it provides a paradigm for frequency attenuation and amplification related to analyte concentration.

Imprinting pNIPA particles with a target analyte can alter pNIPA frequency profile by causing changes in physical properties depending on presence or absence of analyte. Docking of analyte into molecularly imprinted pockets can be seen as stiffening localized areas in pNIPA particles, and decreasing particle size, as docked analyte molecules will force the pNIPA network to adopt a momentary conformation. Work done in this area has shown that increase in stiffness and decrease in particle size results from analyte molecules occupy binding sites in molecularly imprinted nanogels.[110,111] Assuming imprinted pNIPA particles in solution take the form of deformable microbubbles, localized physical changes can be related to alterations in pNIPA frequency profile. If specific particle size at equilibrium a_{e}, and bulk modulus K_p (proportional to overall stiffness), are considered as chief contributing parameters to pNIPA particles resonant frequency f_0, a relationship can be expressed in the following form,

$$f_0 = \frac{1}{2\pi \quad a_e}\sqrt{\frac{3K_p}{\rho_L}}, \qquad \text{Eq. 5.1}$$

known as the Minnaert equation, where ρ_L is the density of the surrounding liquid. Consider an imprinted pNIPA microbubble surrounded by a few analyte molecules. The microbubble will experience altered resonance frequency due to different physical properties caused by docking of the analyte, with greater amounts of analyte providing larger contributions to resonance frequency shift. Complex, non-linear interrelationships between the particle size and bulk modulus non leads to specific frequency profiles for imprinted pNIPA microbubbles dependant on analyte concentration, analyte of interest, and microbubble molecular composition.

5.5 Materials and Methods

5.5.1 Synthesis of Theophylline Imprinted Nanogel

Molecularly imprinted pNIPA was prepared by adding 1.0 g of pNIPA monomer, 0.08 g of N,N'-methylene-bis-acrylamide (MBA), 0.08 g methacrylic acid (MAA), and 0.18 g theophylline to 99 mL of distilled water (dH_2O) to form a homogeneous 1 wt. % pNIPA solution with stirring over 4 h to ensure complete dissolution.[112] Oxygen in the solution was purged with nitrogen gas during this time. We then added 15 mg of ammonium persulfate to initiate the polymerization and 60μL of tetramethylethylenediamine as an accelerator. The solution was left to polymerize for 4 hours with gentle stirring. Once the imprinted nanogel was formed, the theophylline was removed by successive methylene chloride extractions. The extraction of theophylline was confirmed spectroscopically at 271 nm. All chemicals were purchased from Sigma-Aldrich (Ontario, Canada). Dynamic light scattering was used to characterize pNIPA size distribution. Mean nanogel particle size was attributed to 220 nm. A flowchart outlining synthesis is shown in Figure 5.3 a).

5.5.2 Synthesis of DHI Imprinted Nanogel

N-isopropylacrylamide (pNIPA, 500 mg), methylene-bis-acrylamide (MBA, 50mg), methacrylic acid (MAA, 50 uL) and dopamine hydrochloride (50 mg) were

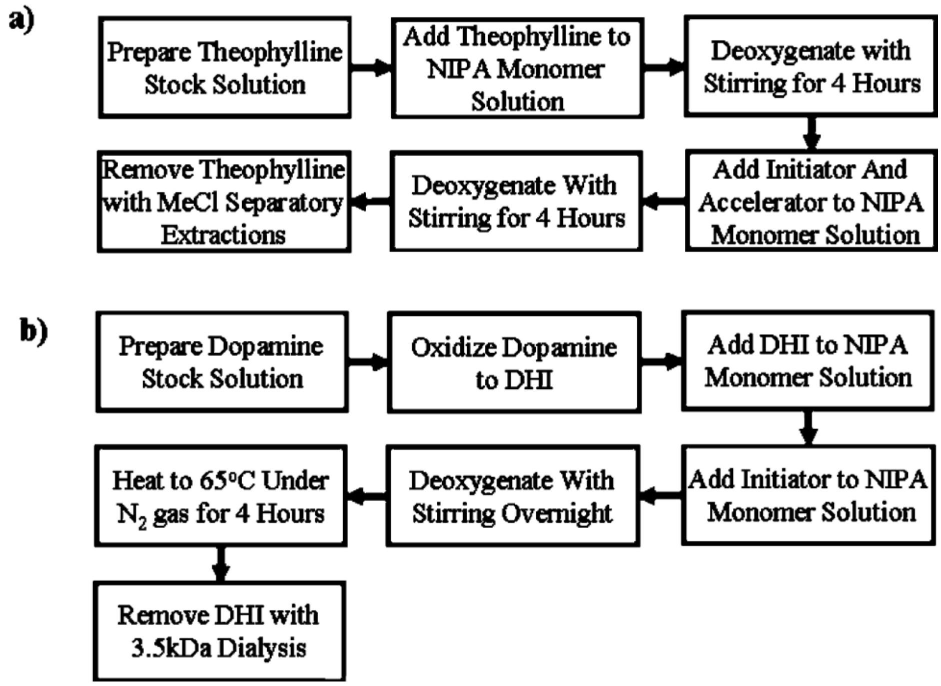
a)
Prepare Theophylline Stock Solution
Add Theophylline to NIPA Monomer Solution
Deoxygenate with Stirring for 4 Hours
Add Initiator And Accelerator to NIPA Monomer Solution
Deoxygenate With Stirring for 4 Hours
Remove Theophylline with MeCl Separatory Extractions
b)
Prepare Dopamine Stock Solution
Oxidize Dopamine to DHI
Add DHI to NIPA Monomer Solution
Add Initiator to NIPA Monomer Solution
Deoxygenate With Stirring Overnight
Heat to 65°C Under N$_2$ gas for 4 Hours
Remove DHI with 3.5kDa Dialysis

Figure 5.3. Flowchart for molecular imprinting procedure of ppNIPA nanogels. Figure denotes, a) theophylline imprinted pNIPA nanogel, and b) DHI imprinted pNIPA nanogel.

added to 60 mL previously boiled milli-Q water to produce a 0.9 wt. % pNIPA solution. The solution was deoxygenated with nitrogen and stirred overnight. Azo-bis-isobutyronitrile (AIBN, 20 mg) initiator was added to the mixture, and the solution was deoxygenated with nitrogen and stirred for 4 hours at 65^{o}C. Upon cooling, the resulting polymer solution was heated to 36^{o}C and phase transition was confirmed visually by an increase in turbidity. The solution was dialyzed to remove DHI template. Extraction of DHI was confirmed spectroscopically at 220 nm.[113] DLS verified mean particle size to be 164 nm. All chemicals were purchased from Sigma-Aldrich (Ontario, Canada). An outline of this procedure is shown in Figure 5.3 b).

5.5.3 Nanogel Solutions

Nanogel solutions were used to estimate theophylline and DHI concentration based on ultrasonic response. A matrix of twenty-five solutions containing 0.8 % by weight theophylline imprinted nanogel and 0.8 % by weight DHI imprinted nanogel in deionized water were made. Theophylline was added to nanogel solutions spanning 8.4 to 49 uM concentration range in one matrix dimension, while DHI was added in 48.8 to 176 nM amounts in the other dimension. A set of replicate scans was withheld from calibration, and used as an unknown set.

5.5.4 Instrumentation

A custom sample cell was constructed from thin plexiglass with one ultrasound transducer affixed to each end. Clinical ultrasound gel was applied for coupling. A 1.9 MHz narrow-band transducer (Advanced Technology Labs Inc., Pennsylvania, U.S.) generated ultrasound pulses and a 5 MHz wideband transducer (Russel NDE Systems, Alberta, Canada) received the ultrasonic signal to maximize

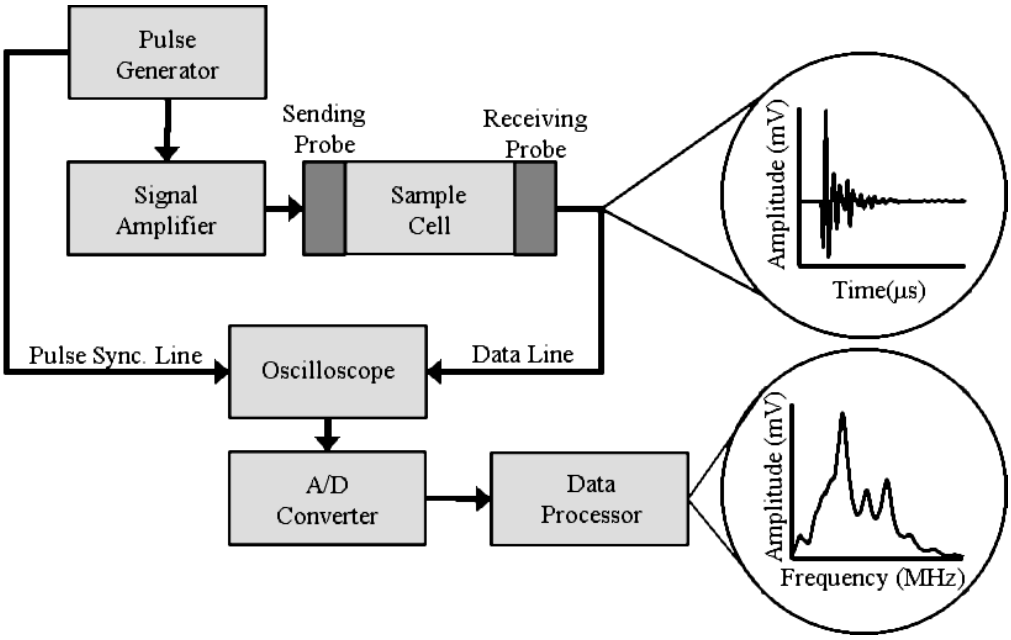

Figure 5.4. Ultrasonic quantification system instrument setup. Impulses of 2 μs were generated to drive the 1.9 MHz transducer, which sent ultrasound through the sample cell to interact with imprinted nanogel. A 5 MHz wideband transducer on the other end of the sample cell detected the ultrasound transmission, which was digitized by the oscilloscope, and processed on computer.

enharmonic signal acquisition. Electrical signals were generated with a Panametrics 500PR Pulser/Receiver (Panametrics Inc., Massachusetts, U.S.), serving as the pulse generator for the transducers. Signals were digitized with a Handyscope HS3 (TiePie Engineering, Sneek, Netherlands) sampling at 50MHz using 12 bit A/D conversion. The complete system is shown in Figure 5.4.

5.5.5 Data Processing

Ultrasonic data from all solutions were digitized and the corresponding frequency spectra were determined using Fourier analysis. Solutions were separated into calibration sets (used to construct concentration models) and unknowns (used for model validation). Each sample was analysed in triplicate. Likewise, the entire quantification analysis was also repeated three times. Stagewise multilinear regression (SMLR) was used to construct concentration models using a linear combination of amplitudes at select frequencies to determine concentration in the form of,

$$y = b_0 + b_1x_1 + b_2x_2 + ... + b_nx_n \qquad \text{Eq. 5.2}$$

where y is the dependant variable, {x} are independent variables, and {b} are the coefficients determined. SMLR is an iterative process that cycled through acquired spectra and determined a trend with highest linearity corresponding to concentration increase, yielding a slope coefficient. The determined trend was subtracted from all spectra (to avoid selecting frequencies closely centered together) and the process was repeated until the desired number of coefficients is obtained. Theophylline and DHI concentration was then calculated through the linear sum of products between slope coefficients and corresponding frequency amplitudes, as shown in equation 5.2. Data overfitting was avoided by selecting the fewest possible number of slope coefficients, to obtain the most parsimonious model. Parsimony was determined using F-tests (α=0.05) carried out with the standard errors of each developed model against each subsequent model containing an additional {b} coefficient.[72] All software was written in Matlab (The MathWorks Inc., Massachusetts, U.S.).

5.6 Results and Discussion

Relationship between ultrasonic profile and physical properties of imprinted nanogel contrast agents is key to our system of ultrasound analyte quantification. Phase can be induced by ultrasonic perturbation[114], hydrostatic pressure, ionic strength, and hydrophobic interactions.[80, 95] Since amplification and attenuation in multiple frequencies are observed when nanogels undergo phase transition, any modification of nanogel physical properties will also cause multiple alterations in ultrasound profile under similar circumstances. Imprinting nanogels with target analyte allows both particle size and stiffness to undergo change when imprinted nanogels are subject to target analytes.[110, 111] Consequently, different ultrasonic profiles can result when imprinted nanogel particles are probed with ultrasound in presence of target analyte as opposed to imprinted nanogel particles in absence of analyte. Greater analyte concentrations will result in increased number of analyte docking into binding sites, further altering physical properties of imprinted nanogel. Therefore degree of ultrasonic profile change due to analyte can be used to determine analyte concentration in solution.

Differently imprinted nanogel contrast agents can also have different interactions with respective analytes due to imprinting method, particular analyte, nanogel composition, and size distribution. This allows multiplexed nanogels to simultaneously determine multiple analyte concentrations depending on different frequencies selected for constructing MLR concentration models. In order to establish unique frequency changes each analyte imparts on ultrasonic profile in presence of multiplexed nanogel, difference spectra were calculated through subtraction of spectra containing high and low concentrations of theophylline and DHI. The difference spectra of these two analytes are shown in Figure 5.5. Increasing theophylline concentration produces a general increase in frequency amplitude, with the exception of the large negative change between 1.0 and 2.0 MHz. DHI appears to have an inverse affect on frequency amplitude, demonstrating a decrease in frequency amplitude except for a large increase in the 1.0 to 2.0 MHz region. The difference spectra in this figure were normalized with respect to area. The results in Figure 5.5 suggest that simultaneous analyte quantification of DHI and theophylline might be possible, since each analyte in presence of multiplexed nanogel has a unique ultrasonic profile.

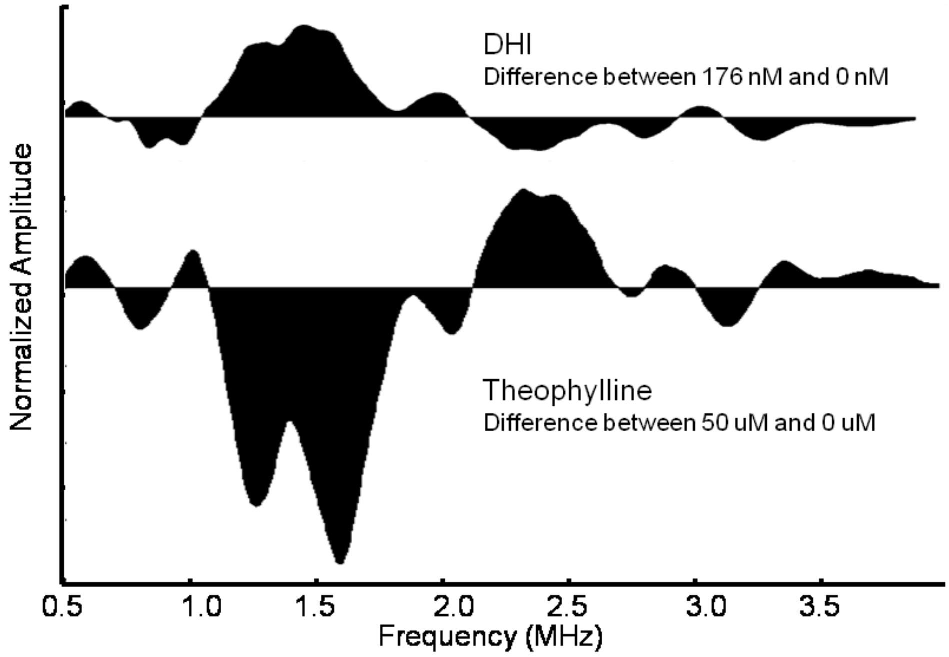

Figure 5.5. Difference spectra of comparing low and high concentrations of DHI and theophylline. The difference spectra were calculated by subtracting ultrasonic spectra of solutions containing low analyte concentration from high concentration spectra. Unique frequency changes attributed to DHI and theohylline are shown across the spectra by darkened areas.

The previously mentioned matrix of solutions containing DHI in 48.8 to 176 nM amounts, and theophylline spanning 8.4 to 49 uM were probed with our modified A-mode ultrasound instrumentation to construct concentration models for each analyte. Concentration models were constructed with MLR using amplitudes at 4 frequencies for DHI, and 6 for theophylline (see Figure 5.6 a)). MLR constructed concentration models were validated against a test set, comprimised of replicate scans not included in the calibration set. High linearity between model and test set was achieve for DHI

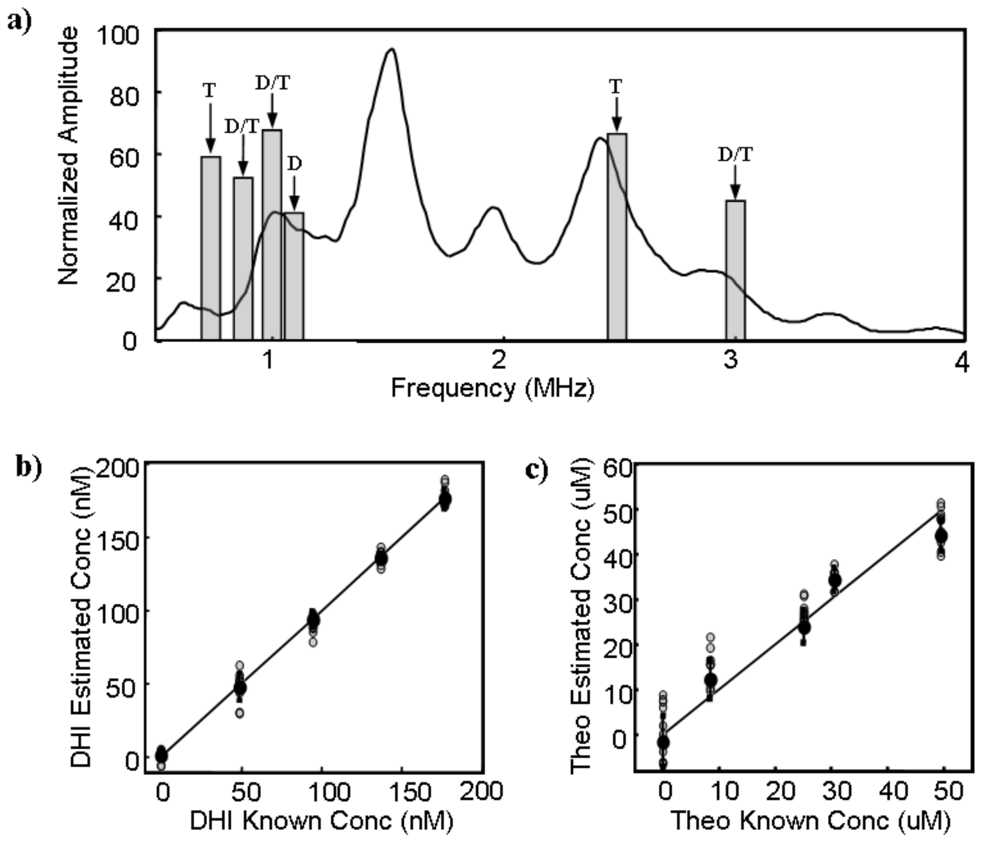

Figure 5.6. Results of simultaneous analyte quantitation using molecularly imprinted multiplexed nanogels. Figure a) represents frequencies selected to construct concentration models for DHI (D) and theophylline (T). Figures b) and c) refer to concentration estimates using constructed models for DHI and theophylline, respectively. Calibration sets are represented by the black circles with error bars, and validation test sets are denoted by light gray circles.

(r^2 = 0.99), and good agreement was obtained for theophylline concentration model and test set (r^2 = 0.96), with DHI shown in Figure 5.6 b), theophylline shown in Figure 5.6 c). The Standard Error of the Estimate for both concentration models were determined to be 7.9 nM for DHI, and 5.3 μM for theophylline. Both concentration ranges overlap with therapeutic and physiological concentration ranges of dopamine (DHI precursor) and theophylline.[115, 116] Results indicate several ultrasound frequency sets were common in quantifying both DHI and theophylline, notably 0.8, 1.0 and 3.0 MHz. while a few were unique to each analyte, 1.1 MHz for DHI, 0.7 and 2.5 MHz for theophylline. Similar frequencies selected to construct concentration models can be explained from using imprinted pNIPA in both cases, and similar molecular sizes of both DHI and theophylline. However, different initiators and polymerization conditions were required when synthesizing nanogels imprinted for DHI, due to unsuccessful attempts at synthesis using the proceedure outlined for theophylline imprinting. Consequently, Particle sizes for both nanogels are subsequently different, owing to DHI imprinted pNIPA nanogel being synthesized in the collapsed state, at a temperature above T_c, as required by the initiator used. This can result in certain unique frequencies selected for quantification of individual analytes, as shown in Figure 5.6 a). It is, however, interesting to note that no frequencies were selected in the region of greatest change with respect to analyte concentration, namely the 1.0 to 2.0 MHz region, as illustrated in Figure 5.6. Although this region illustrated large profile changes for both analytes, it is possible these do not necessarily correlate linearly or non-linearly with increasing analyte concentration.

The concentration ranges analysed for both analytes also differed by orders of magnitude, requiring different percent loadings. The percent loading of DHI imprinted pNIPA was on par with solutions made in chapter 4, and were hence appropriate for the nM concentration range. The percent loading of theophylline imprinted pNIPA was considerably lower than that used for the μM in chapter 3, which explains the lower degree of linearity obtained for the μM theophylline concentration model. Higher concentration estimate agreement with DHI quantification can also be ascribed to stronger analyte-nanogel interaction due to synthesis of nanogel in collapsed state. Tightly knit analyte binding sites can result from pNIPA particles synthesized in collapsed state surrounding analytes, as opposed to binding sites created in loose pNIPA network at equilibrium during theophylline imprinted pNIPA polymerization.

5.7 Conclusion

Using multiplexed molecularly imprinted pNIPA nanogels, we have successfully quantified DHI and theophylline simultaneously in concentrations ranging containing DHI in 48.8 to 176 nM amounts, and theophylline spanning 8.4 to 49 uM. These ranges overlap with the therapeutic range for dopamine, 13 to 130 nM,[116] and for theophylline, can be extended into the physiological range of 50 to 100 uM.[115] These results provide additional support for the implementation of a multi-analyte detection system with simultaneous quantification capabilities using ultrasound and molecularly imprinted pNIPA nanogels. Overall, combination of several differently-imprinted nanogel contrast agents shows promise for rapid, simultaneous multi-analyte concentration analysis that can be applied to clinical settings.

5.8 Acknowledgements

This work was supported in part by the National Science and Engineering Research Council of Canada and by the Fonds Québécois de la Recherche sur la Nature et les Technologies.

Chapter 6.

Tobacco Mosaic Virus Quantification Using Ultrasound and Virus Sensitive Nanogels

As shown in previous chapters, ultrasonic quantification of molecular analytes was achieved through the use of molecularly imprinted nanogel polymers, with detection limits in the low to mid nM range. The key motivation in this chapter was to attempt quantification of analytes with increased mass in order to extend the range and application of ultrasound quantification using analyte sensitive contrast agents. Macromolecular structures, such as viruses, are very relevant as analytes in biomedical fields. Viral detection is typically done using fluorescence or enzyme linked immunosorbent assay (ELISA). Both of of these methods involve many preparation steps including fluorescent tagging of analytes, preparation of antibody reaction (and secondary antibody if required), incubation, and so on. These preparation steps require many hours and specialized equipment, and in many cases, incubation must be done overnight. Finally, color results of ELISA test wells can require up to 30 minutes incubation to confidently conclude presence or absence of virus. Rapid, low cost methods of quantitation utilizing ultrasound could reduce the need for multiple preparation steps, and increase output of results while reducing analysis time. Tobacco mosaic virus (TMV), an RNA virus infectious to tobacco plants but harmless to humans, is commonly used in clinical and macromolecular research and was chosen as the analyte. TMV structure is composed of four α helices that loop into a tube structure, with RNA located in a center channel protected by coat protein, with a total molecular weight of about 18,000.[117] Due to the size of the TMV and small particle diameter of the previously pNIPA synthesized contrast agents, macromolecular structures could not be imprinted in the same fashion as molecules. Instead, polymerization takes place in the presence of template macromolecule in order to develop pNIPA contrast agents that could interact with and briefly adhere to the macromolecular analyte. As such, additional motivation behind this research was to investigate ultrasonic profile changes of pNIPA contrast agents in presence of TMV. Size difference between pNIPA contrast agents and TMV meant alterations in ultrasonic properties could no longer be attributable to localized changes in contrast agent stiffness and physical properties. Instead, mass changes of several pNIPA particles adhering to TMV could be the cause of alterations in ultrasonic profile, much like larger masses on a single mass-spring system change oscillation frequencies of the spring. Through dynamic light scattering, the average number of pNIPA contrast agents adhering to TMV can be determined. This could provide support for the mass changing theory behind ultrasonic changes recorded in solutions containing TMV sensitive pNIPA contrast agents and various concentration levels of TMV itself. Ascertaining specificity of TMV sensitive contrast agents was also a goal of this research. Tomato bushy stunt virus (TBSV) is spherical in shape, and was used to determine whether or not multiple viruses can be detected simultaneously. TBSV also served as an interferant for TMV quantification.

6.1 Manuscript

Tobacco Mosaic Virus Quantification Using Ultrasound and Virus Sensitive Nanogels

Authors:

David Troïani and David H. Burns

6.2 Abstract

Analyte quantification in biomedicine often involves multi-step processes from sample acquisition to analysis using traditional instrumentation. We have developed a general approach for an ultrasonic quantification system that uses smart biosensors based on imprinted poly-N-isopropylacrylamide (pNIPA) nanogels, eliminating many of the steps involved with traditional instruments. Multilinear analyte quantification in aqueous nanogel solution led to the construction of a concentration model for tobacco mosaic virus (TMV) from 9 to 140 ppb resulting in correlation coefficient of 0.99. Additional trials were prepared and conducted in the presence of an interfering species, tomato bushy stunt virus (TBSV). Regression analysis of TMV in presence of TBSV interferant illustrated good linearity ($r^2 = 0.97$), while analysis of TBSV in the same scenario provided a much lower correlation coefficient and was only semiquantitative.

6.3 Keywords

ultrasound, nanogel, molecular imprinting, n-ispropylacrylamide, tobacco mosaic virus, biosensor, analyte recognition, tomato bushy stunt virus

6.4 Introduction

6.4.1 Background

Rapid analyte quantification methods are of key importance in many fields of chemistry and biomedicine. Analysis of pathogens such as viruses and bacteria, usually involve many steps including sample extraction, which can involve uncomfortable biopsies, amplification or concentration of target analyte, reverse transcription often involving enzymes, and finally analyte quantification.[118,119] Virus and bacteria analysis is typically done via fluorescence using fluorescing proteins often developed through enzymatic assays. Prime examples of this method include detection of hepatitis C virus and Escherichia coli (E. coli) bacterium. Hepatitis C virus detection utilized fluorescent enzyme immunoassay and E. coli analysis was carried out using fluorescein-labeled antibodies.[120, 121] General assessment for presence of TMV is typically done using ELISA tests, which require many hours of sample preparation and incubation including overnight incubation necessary in certain cases. This lengthy period between sampling and results can be inconvinient and costly, especially for time sensitive results. TMV ELISA tests are generally capable of identifing concentrations above 4 ppb, with typical ELISA tests having an RSD of roughly 5%.[122] Development of fast, low cost methods of bioanalyte analysis using alternative technologies such as ultrasound, could reduce the number of steps needed for analysis.

Ultrasound has long been the method of choice for penetrating scattering media, such as blood and tissue. Acoustic waves are non-destructive and penetrate scattering media effectively, making ultrasound the ideal choice for non-invasive *in vivo* imaging. Methods utilizing ultrasound imaging are common in clinical settings, composed mostly of fetal analysis and cardiovascular sonography. Recent interest in ultrasound has focused on thickness shear mode (TSM) resonators, also known as quartz microbalances, as a modality for ultrasound analyte quantification. Quartz crystal oscillators with known specific resonance frequencies are immersed in either gas or liquid media. Resonance is altered according to amount of analyte adsorbed onto the surface, with greater accumulation of analyte resulting in increased crystal mass, which changes resonance frequency. TSM resonators have already been applied to analysis of a wide variety of proteins, biomarkers, and antibiotics.[123] Another application involves using arrays of TSM resonators as "electronic noses", with focus towards recognizing multiple analytes.[124,125] However, TSM resonator arrays are also subject to limitations, centered mostly around interfacial effects from molecular motion, surface irregularities, and fouling of unwanted analytes on resonator surfaces.[126]

Ultrasound contrast agents have also gathered interest in biomedical fields. Contrast agents are composed of membrane encapsulated spherical gas-filled microbubbles that resonate at characteristic frequencies, improving the imaging capabilities of ultrasound and enhance ultrasound echogenicity.[127] Unfortunately, lack of molecular information and analyte quantification are still present. Our group has developed a unique approach to analyte quantification by complementing low cost, non-invasive ultrasound with molecularly sensitive contrast agents made from poly-N-isopropylacrylamide (pNIPA) nanogels.

Nanogels are polymers that have distinct properties in solution. At equilibrium, nanogel particles exist as a loose, swollen interconnection of polymer chains forming a network which allows solvent molecules to freely travel through. Upon altering certain environmental parameters past a threshold value, the loose network quickly collapses on itself. In this state, intra-network chain interactions dominate over attraction between network chains and solvent. This transition has been characterized in detail.[128, 129, 130] The prime characteristic of interest is change in ultrasonic properties that accompanies physical property changes of pNIPA nanogel as it undergoes spinodal decomposition. Sharp ultrasonic attenuation at a single frequency is associated with phase transition.[51] Results indicate that unique ultrasonic properties of pNIPA nanogel make it suitable for use as an ultrasound contrast agent. What makes nanogels attractive as ultrasound contrast agents is the ability to synthesize molecularly sensitive nanogels, which are well established in literature.[131,

[132] By allowing self-assembly of the nanogel monomers and analyte prior to initiating polymerization, molecularly sensitive binding sites can be created.[80, 133] Upon docking of the analyte into an imprinted pocket, the nanogel will undergo a change in physical properties and in ultrasonic response. Research into molecularly imprinted nanogels have demonstrated increased stiffness and decrease in particle size results from analyte molecules docking with binding sites in molecularly imprinted nanogels.[134, 135]

However, bioanalytes in general take the shape of supramolecular structures such as viruses. The ultrasound analyte quantification system described here relies on imprinted nanogel contrast agents that recognize viruses. Tobacco mosaic virus (TMV), our test analyte of interest, is a rod-shaped RNA virus composed of 4 α helices that loop into a 300 nm long, 18 nm wide structure, with RNA located in a center channel protected by coat protein. TMV normally infects tobacco plants, visible through rotting and yellow-brown discoloration, and is responsible for damaging crop yields, though consumption of infected crops has been shown to have no effect on humans.

Ultrasonic quantification of TMV was carried out using the modified A-mode ultrasound system and TMV sensitive nanogels. When surrounded by TMV virus, the nanogel interacts differently with ultrasound, providing additional frequency components. A simplified illustration of this effect is demonstrated in Figure 6.1. Greater number of viruses present in solution can lead to greater number of pNIPA-viruses interactions, causing a larger change in ultrasonic profile. Results were obtained from multiple TMV quantification experiments centered on TMV sensitive pNIPA nanogels and a model was developed for analyte specific contrast agent microbubbles.

Additional quantification assays were conducted in the presence of TBSV, a

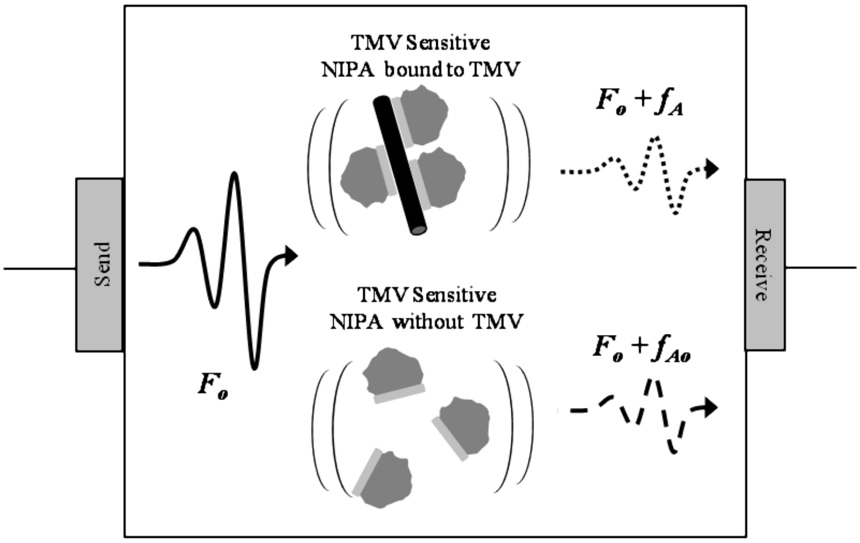

Figure 6.1. Ultrasonic quantification using TMV sensitive nanogel contrast agents. Various frequency components are added to base pulse signal F_o by pNIPA nanogel in presence and absence of TMV. Signal f_A represents components added from pNIPA binding to TMV. Signal f_{Ao} represents components added from pNIPA nanogel in absence of TMV.

virus of comparable size to TMV. Unlike TMV, TBSV exists in a spherical form with a diameter of 35 nm. TBSV was chosen as an interferant to investigate how the pNIPA nanogels would behave when bound to each virus, ultimately determining how specific the nanogel sensitivity was towards the target TMV. Although an ideal scenario would solely include interactions between pNIPA and TMV irrespective of sample matrix, in real systems pNIPA may interact with both viruses simultaneously. However due to specificity of imprinted pNIPA toward TMV, as well as the different size and mass of each virus type, interactions between pNIPA and TMV can yield different ultrasonic frequency profile changes than interactions with TBSV, allowing quantification of TMV through multi-linear regression.

6.4.2 Theory

If nanogels are considered as deformable microbubbles, a model can be developed by examining the interaction between ultrasound and nanogel properties at equilibrium and in presence of analyte. Nanogel microbubbles have a native resonance frequency, and resonate when perturbed by ultrasonic waves based on nanogel physical properties. By synthesizing nanogels with chemistry that promotes nanogel-virus interaction, nanogel particles can be made sensitive to presence of viral analyte. Much like a single spring-mass system, nanogel microbubbles momentarily aglomerated around individual virus particles will have a greater size and mass than nanogel alone, altering ultrasonic resonance frequency profile of the resulting larger particle. Since nanogel-virus interaction are only momentary, it is unlikely that nanogel particles stay bound to virus for longer than a short time period. Nevertheless, many such interactions, each creating minute changes in ultrasonic frequency spectra produce enough change for quantification. Not all nanogel-virus interactions will have the same duration, or the same resulting shape, size or mass, thus interrelationship of multiple non-linear amplitude changes at multiple frequencies are necessary to construct valid concentration models. This model, although not ideal for ultrasound-nanogel interaction, approximates behavior through the notion of changing particle size and mass in presence of viral analute affecting resonance frequency profile of nanogels in solution.

6.5 Materials and Methods

6.5.1 Synthesis of TMV Selective Nanogel

N-isopropylacrylamide (pNIPA, 500 mg), methylene-bis-acrylamide (MBA, 50mg), methacrylic acid (MAA, 50 uL), tetramethylethylenediamine (TMED, 60 μL) and 100 μL of TMV solution (15 mg dissolved in 2 mL milli-Q water) were added to

60 mL milli-Q water. The solution was deoxygenated with nitrogen and stirred for 4 hours. Ammonium persulfate (APS, 10 mg) initiator was added to the mixture, and the solution was deoxygenated with nitrogen and stirred overnight. The resulting polymer solution was heated to 65^{o}C for 10 minutes and precipitated TMV was removed via filtration. Successful polymerization was determined by visually confirming the increase in turbidity of a phase transition as a result of increasing temperature passed T_c, 33.6^{o}C. Particle sizing of pNIPA was determined by dynamic light scattering to be 130 nm in diameter using a Malvern Instruments Zetasizer Nano ZS (Malvern Instruments, USA) equipped with a 659 nm laser. Diameter of particles in solutions containing pNIPA and TMV was also done, and showed increasing particle size (398 nm, 464 nm and 552 nm) with increased TMV concentration (10 ppb, 50 ppb and 90 ppb respectively). Solutions containing only TMV resulted in particles of 351 nm diameter. Particle sizing of the pNIPA nanogels in differing concentrations of TBSV (10 ppb, 50 ppb, 90 ppb) yielded sizes between 100 and 160 nm, with a less specific increase in particle size related TBSV concentrations. The measurements of solutions containing TBSV had overall significantly smaller particle size than TMV solutions, due to the lower interaction between TBSV and pNIPA. This was expected, as the pNIPA was synthesized to be sensitive to TMV. All chemicals were purchased from Sigma-Aldrich (Ontario, Canada). TMV and TBSV were purchased from Agdia Inc. (Indiana, USA).

6.5.2 Nanogel Solutions

Nanogel solutions containing pNIPA were used to estimate TMV concentration based on ultrasonic profile changes. Six solutions containing 0.8 % by weight pNIPA nanogel in deionized water were made. TMV was added to pNIPA solutions in concentration amounts of 9 to 140 ppb. Using the approximate TMV molecular weight of 18,000,[117] this translates to roughly 0.5 nM to 7.8 nM. In order to assess the specificity towards TMV, a second set of samples were prepared. The second set consisted of a matrix of 25 solutions with TMV serially increasing from 9 ppb to 92 ppb (0.5 to 5.2 nM) at each TBSV concentration (from 18 to 74 ppb, or 1.0 to 4.1 nM).

6.5.3 Instrumentation

A 1 cm path length sample cuvent was constructed from plexiglass, which was housed in a sample cell containing one ultrasound transducer affixed to each end. Clinical ultrasound gel was applied for coupling. A 1.9 MHz narrow-band transducer (Advanced Technology Labs Inc., Pennsylvania, U.S.) generated ultrasound pulses and a 5 MHz wideband transducer (Russel NDE Systems, Alberta, Canada) received the ultrasonic signal. Electrical signals were generated with a Panametrics 500PR Pulser/Receiver (Panametrics Inc., Massachusetts, U.S.), serving as the pulse generator for the transducers. Signals were digitized with a Handyscope HS3 (TiePie Engineering, Sneek, Netherlands) sampling at 50MHz using 12 bit A/D conversion.

6.5.4 Data Processing

Acoustic data received from the 5 MHz wideband transducer were transformed using a Fast Fourier Transform (FFT) algorithm to frequency spectra for each solution. Each sample was analysed in triplicate. In addition, each entire quantification assay was also repeated three times. All spectra were acquired one after another in sequence, and were seperated into calibration sets used in developing concentration models, and test sets. Validity of the concentration model was determined using the model to calculate concentrations in the test set. Stagewise multilinear regression (SMLR) was used to determine the linear combination of select frequencies that best describe the data in the form of

$$y = b_0 + b_1x_1 + b_2x_2 + \ldots + b_nx_n \qquad \text{Eq. 6.1}$$

where y is the dependant variable, {x} are independent variables, and {b} are the coefficients determined. Parsimony was determined using an F-test (α=0.05) where the standard errors of each developed model against each subsequent model containing an additional {b} coefficient.[72] All software was written in Matlab (The MathWorks Inc., Massachusetts, USA).

6.6 Results and Discussion

Interrelationships between physical properties of the nanogel network and nanogel ultrasonic profile are vital to the ultrasonic analyte quantification system. Interaction between analyte virus and nanogel produce resonance frequency shifts related to alterations in size and stiffness of the nanogel itself, causing a corresponding change in ultrasonic profile. As a greater number of virus analytes are introduced into the system, signal from pNIPA bound to virus dominates over unbound pNIPA, decreasing signal contribution from unbound pNIPA. Analyte concentration is then determined using the increase in signal from bound pNIPA related to the amount of virus analyte present and multilinear regression.

The relationship between ultrasonic response and TMV concentration in TMV sensitive pNIPA nanogels using solutions free of interfering species was first investigated. TMV was added to pNIPA solutions in concentrations of 9 to 140 ppb (0.5 to 7.8 nM). Viruses concentrations were estimated using multilinear regression models based on amplitudes at three ultrasonic frequencies, specifically 2.1, 1.0 and 2.3 MHz in order of significance to the model. Concentration models constructed from these frequencies resulted in a correlation coefficient (r^2) of 0.99 after leave-one-out cross validation, with 8 ppb (0.4 nM) standard error of estimate (see Figure 6.2). The high linearity and low standard error of this model demonstrates the excellent quantification capability of the ultrasound system.

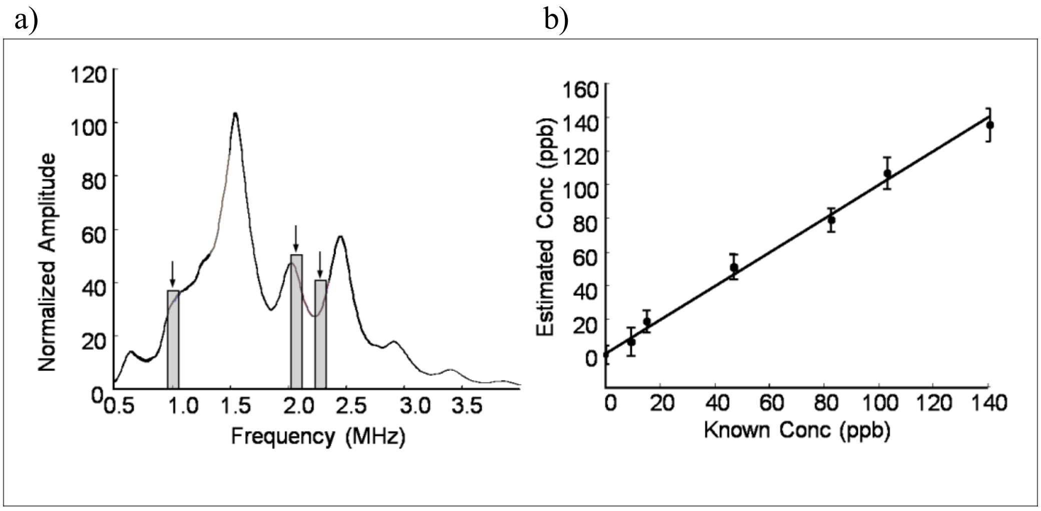

Figure 6.2. Quantification results using ultrasound and analyte sensitive pNIPA nanogel. Figure a) represents frequencies selected to construct concentration models for TMV. Figure b) refers to concentration estimates using constructed models for TMV. Error bars indicate standard error after leave-one-out cross validation.

The selectivity of imprinted pNIPA towards TMV was demonstrated when the multilinear concentration regression model developed for TMV was applied to ultrasonic spectra of solutions containing TBSV in the 18 to 74 ppb (1.0 to 4.1 nM) range. In this case, the expected result would be a lack of fit between estimated and actual TBSV concentrations, if the pNIPA is selective towards TMV. Results indicated that absolutely no linear fit could be established when applying the TMV concentration model to ultrasonic spectra of TBSV solutions. The correlation between estimated and actual TBSV concentrations was non existant (r^2 of 0.05), with a very high standard error of 11000 ppb. This error is more than 4 orders of magnitude higher than the standard error associated with the concentration model constructed for TBSV, and supports the selectivity of imprinted pNIPA nanogels towards TMV.

Analyte specificity was also investigated by serially adding TBSV in 18 to 74 ppb (1.0 to 4.1 nM) amounts to solutions containing 9 to 92 ppb (0.5 to 5.1 nM) TMV. The outcome was a matrix of 25 solutions with TMV serially increasing from 9 ppb to 92 ppb at each TBSV concentration (from 18 to 74 ppb). Investigation of ultrasonic profile change with respect to concentration increases of TMV and TBSV resulted in difference spectra showing multiple frequency amplitude changes. The difference spectra were calculated by subtracting ultrasonic profiles of high concentration TMV and TBSV from their low concentration variants, and are shown in Figure 6.3. The most notable observation when comparing the two

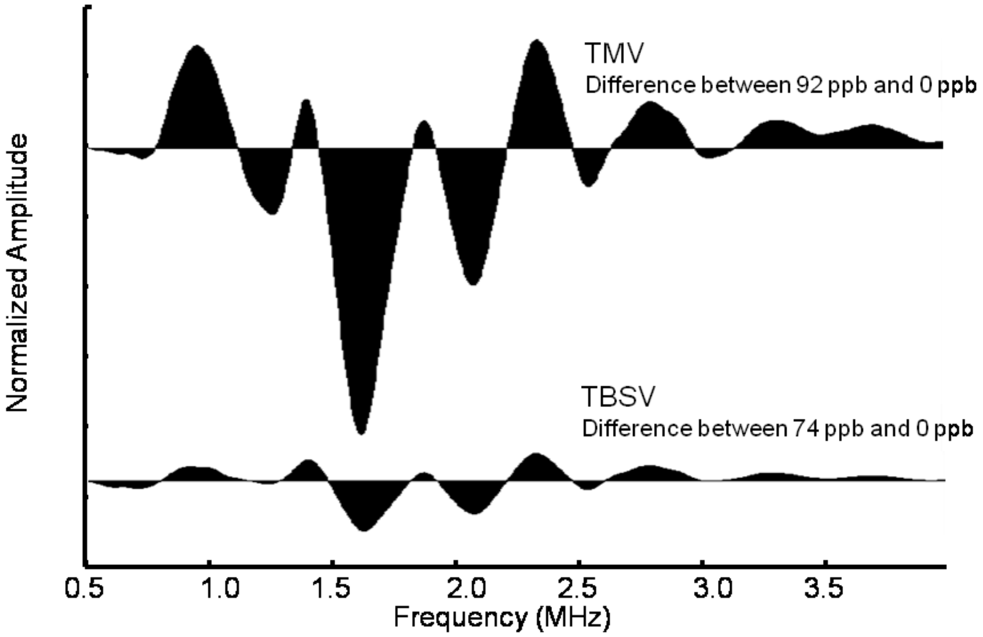

Figure 6.3. Difference spectra of comparing low and high concentrations of TMV and TBSV. Ultrasonic frequency spectra of solutions containing high concentration of analyte were subtracted from spectra with low analyte concentration. The resulting difference spectra demonstrates frequency changes attributed to concentration changes for each analyte.

differencespectra is the magnitudeof the TMV profile change versus TBSV. Comparing the ratio of peak heights from each spectra, particularly in the 0.5 to 1.5 MHz area also shows considerable variation. Upon closer inspection, the two difference spectra share many common components despite contrasting magnitudes and peak-to-peak ratios (TBSV having ~25% the magnitude of TMV. This can be attriubted to the imprinting process, which was targetted towards TMV, resulting in this virus providing a greater change in ultrasonic profile than the TBSV interferant. The observed change in signal can be loosely indentified as being the result of changing mass in a single mass-spring system. Any analyte which can adhere to pNIPA particles will demonstrate similar mass-induced changes in ultrasonic profile. However, target analytes will contribute much larger profile changes in presence of imprinted pNIPA particles due to selectivity of said particles towards the target, as seen in Figure 6.3. Another variation, although much more subtle, between the two difference spectra is illustrated by the slight frequency shift of major components. This is most easily seen in the large negative component at 1.6 MHz, however it is also present to a lesser degree elsewhere in the spectra. A frequency shift of this nature is expected, as increasing the mass in our pseudo mass-spring nanogel system would alter the ultrasonic profile through a shift in the resonant frequency.

Quantification of TMV in presence of TBSV was carried out next. Construction of concentration models for TMV with TBSV interferant was accomplished with a multilinear concentration model constructed from ultrasonic amplitudes at 1.3, 3.9, 2.0, 0.6 and 3.2 MHz, as in Figure 6.4 a). This resulted in a model with a correlation coefficient of 0.97 with corresponding standard error of 8 ppb (0.4 nM), illustrated in Figure 6.4 b). TMV quantification is still possible with TBSV present, however the degree of linearity in the concentration model is somewhat less than quantification assays under an interference free matrix. These results are very encouraging since although the linearity decreased slightly, the standard error of TMV quantified in presence of TBSV did not change significantly from TMV in an interferance free matrix. Comparison of the signal components from the two viruses in Figure 6.3 was now utilized to determine feasibility of simultaneously quantifying TBSV with TMV using TMV sensitive pNIPA nanogel. Models constructed to quantify the TBSV in the presence of TMV resulted in lower model linearity (r^2 coefficient of 0.85) and significantly higher standard error of .

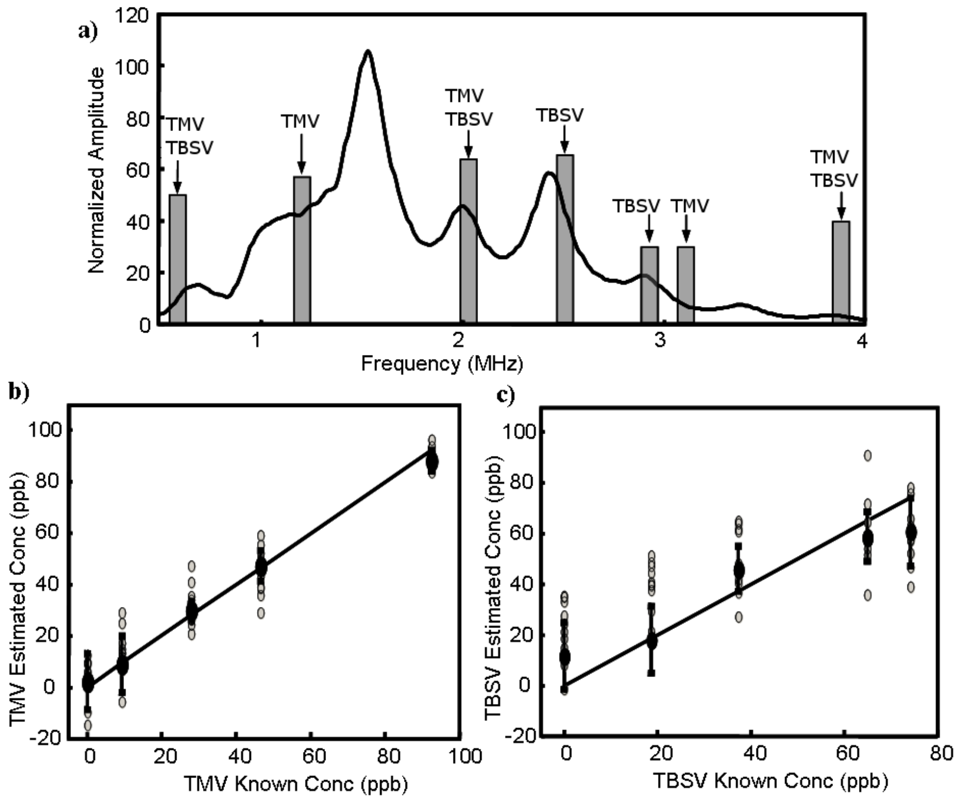

Figure 6.4. Quantification results using ultrasound and TMV sensitive pNIPA nanogel. Figure a) represents frequencies selected to construct concentration models for TMV and TBSV. Figures b) and c) refer to concentration estimates using constructed models for TMV and TBSV, respectively. Calibration sets are represented by the black circles with error bars, and validation test sets are denoted by light gray circles.

estimate SEE of 15 ppb (0.8 nM). Frequencies selected for TBSV quantification were 2.6, 3.0, 0.5, 2.2 and 3.9 MHz in order of weighting, as illustrated in Figure 6.4 a). The results of the multilinear regression model for TBSV are shown in Figure 6.4 c), and indicates lower sensitivity toward TBSV in quantification assays. Different frequencies selected for constructing concentration models for TMV and TBSV are partially explained by different viral sizes, and particles sizes of pNIPA nanogel when interacting with the two viruses. Higher frequencies selected for TBSV quantification can imply a smaller resulting particle size when interacting with pNIPA nanogel. Lower correlation between actual and estimated concentrations also denote lower specificity towards TBSV, resulting in the previously noted smaller particle sizes due to lack of interaction between pNIPA and TBSV. With tuning of the nanogel synthesis process for different viral recognition, high sensitivity and specificity should be possible for several analytes, however simultaneous analyte quantification would require a different approach.

6.7 Conclusion

Using pNIPA nanogels synthesized with viral recognition, TMV in aqueous solution has been successfully quantified in concentrations ranging between 9 and 140 ppb (0.5 and 7.8 nM). Change in ultrasonic signal is specific to increase in TMV concentration compared to TBSV. Different ultrasonic components resulting from probing solutions containing TBSV and TMV imprinted pNIPA can be used to quantify TBSV, however not with the same precision as when the nanogel is imprinted for the virus. It is nevertheless interesting that quantification of both viruses is possible to some degree using one imprinted pNIPA nanogel, which provides the possibility of simultaneously screening for multiple analytes with different types of imprinted pNIPA nanogel. These results provide support for the implementation of an analyte detection system using ultrasound and analyte sensitive pNIPA nanogels in clinically relevant settings through analysis of multiple frequency interactions in ultrasonic profiles. Overall, ultrasonically detected nanogel biosensors appear promising for a variety of clinical and environmental applications.

6.8 Acknowledgements

This work was supported in part by the National Science and Engineering Research Council of Canada and by the Fonds Québécois de la Recherche sur la Nature et les Technologies.

Chapter 7.

Conclusion and Future Work

Quantification of molecular and supramolecular analytes is a necessity in many fields from different disciplines including medicine, biology and chemistry, as well as numerous subfields. Most quantification is done using spectroscopic techniques based on optical measurements. However these techniques can present difficulties with complex matrices and overlapping spectral signatures, which are commmon in multi-analyte environments, as well as many biological samples. Ultrasonic analyte quantification is a field that has largely been unexplored. Based on molecularly selective contrast agents made from imprinted nanogels, this technique allows tailoring for specific analytes in potentially complex matrices. Spectral overlap becomes less of an issue, as high specificity toward single analytes can be achieve through the molecular imprinting process. Ultrasound can penetrate much deeper into samples than optical techniques, and ultrasonics are also minimally invasive in many cases. Fine tuning nanogel particle size and ultrasound frequency range can be done for environments where more ultrasonic penetration and backscatter is required.

7.1 Conclusion

Analyte quantitation using ultrasound coupled with molecularly senstive contrast agents based on pNIPA nanogels has been shown to quantify analytes in the nM range with excellent correlation between estimated and actual concentration. Quantified analytes include theophylline, caffeine and dopamine (through DHI) with excellent linearity between actual concentrations and estimates. Concentration models constructed using ultrasonic profile differences and SMLR have demonstrated effective analyte quantification in interferance-free solutions, as well as in the presence of molecular interferons ranging from closely structured molecules to serum albumin. Ultrasonic profile changes occuring from localized stiffness (modulus), size and other physical properties were shown to be related to analyte docking in molecularly imprinted binding sites, dependant on analyte concentration in solution. Alterations in nanogel contrast agent stiffness and size were related to resonant frequency shifts via the Minnaert equation. Combination of two differently imprinted nanogels to form a multiplexed nanogel solution was demonstrated to quantify multiple analytes in solution simultaneously with similar correlation coefficients. Storage of pNIPA nanogel contrast agents through lyophylization for several months was determined to have minimal consequences on ability to quantify molecular analytes.

Quantifying supramolecular structures has produced concentration models able to quantify virus structures in the ppb range. Viral interferons added to solution illustrated a slight decrease in quantification efficiency, however good correlation was still obtained. Theory behind ultrasonic changes was discussed to be closely related to a difference in mass in single mass-spring systems.

This method has great potential as a rapid point-of-care technique for quantifying multiple analytes in the nano to micromolar range. The ease of use and minimal invasiveness of the ultrasound system described in this thesis provides a wide variety of potential applications in clinical and biological fields.

7.2 Future Work

7.2.1 Ultrasonic Analyte Image Mapping

A key area to continue this thesis would be in the direction of combining ultrasonic analyte quantification with contrast image mapping of an imprinted nanogel sample in water. In preliminary trials of this experiment, a nanogel sample containing on pNIPA nanogel was placed in an air tight pastic container and submersed in a water bath. Two ultrasound transducers were attached to either ends of the water bath, with tips submersed in water. The receiving wideband transducer was set to a fixed position, while the narrowband transducer generating ultrasound pulses was mounted on a servo motor driven by a BASIC Stamp II (Parallax Inc., Rocklin CA, USA), and scanned across the water bath in an arched pathway (see Appendix C). A set number of positions between beginning and end of the arc were pre-selected, and ultrasound was pulsed into the water bath at each position in sequence. Data recorded using the fixed wideband receiving transducer was processed into spectrograms illustrating how the ultrasound signal changed with time. Preliminary results show differences in ultrasonic profile several frequencies as the scanning transducer passed over the nanogel sample along the arched pathway. The spectrograms indicate where the nanogel was located in the water bath, with multiple frequencies being affected. This provided support that analyte quantification inside the submersed imprinted nanogel sample might be possible, using similar processing techniques to analyte quantification done in this thesis. Analyte quantification in this experiment would represent analyte image mapping inside an environment where the target is not in direct contact with ultrasound transducers. This type of enviroment would require multiple media boundaries to be crossed, analogous to accessing a particular organ in the human body.

7.2.2 Analyte Sensitive Nanogel Fundamental Frequencies

Investigating ultrasonic analyte quantification at higher frequencies provides an interesting avenue to obtain stronger ultrasonic signal. It has already been shown that sub-harmonic backscatter from contrast agents probed in the ultrasonic range can be used for characterization.[50] Recording fundamental harmonics in higher frequency ranges would provide greater ultrasound signal changes in presence of analyte, however at the cost of decreased penetration depth.

A system designed in this fashion would have to generate ultrasound with much greater energy, since attenuation across media is significantly higher with increased ultrasound frequency. A general rule of thumb for the relationship between penetration depth and ultrasonic frequency can be expressed as follows,

$$\text{Depth} = \frac{1}{\text{Frequency} \cdot \alpha}, \qquad \text{Eq. 7.1}$$

where α represents an attenuation coefficient specific to the medium the ultrasound is travelling through. This equation illustrates that the penetration depth is quickly reduced when increasing ultrasonic frequency. Thin sample cells designed with microfluidics would have to used in order to record signal at acceptable levels. The advantage of probing nanogel fundamental frequencies lies in a greater amount of acoustic data indicative of analyte concentration, which could translate into much lower detection limits than demonstrated in this thesis.

7.2.3 Increased Concentration Sensitivity

The results of experiments conducted in Chapter 4 demonstrated lower detection limits than results in Chapter 3, providing analyte sensitivity in the nM range. A lower percent loading of imprinted pNIPA per sample solution can be attributed to the nM detection limits. Having fewer imprinted pNIPA particles signifies a greater overall difference in signal between pNIPA particles containing docked analyte and particles without docked analyte. For example, a solution containing a high percent loading of imprinted pNIPA particles would not show a significant signal change if one analyte molecule docked into one of the many pNIPA particles. Investigating detection limits in the sub nM range by further lowering the percent loading of imprinted pNIPA per sample would therefore be a fruitful avenue to pursue in the thesis. Challenges in sub nM quantification might involve the reduced overall pNIPA ultrasound signal obtained from drastically lower percent loadings. This limitation could be overcome through amplification of pNIPA signal extracted using blank correction or background correction techniques. A stable environment including precise ambient acoustic, temperature and humidity control would have to be used in order to have adequate blank or background spectra.

7.2.4 pNIPA Microbubble Synthesis

Synthesizing pNIPA microbubbles would be another approach to obtain stronger acoustic signal during quantification. Aforementioned synthesis procedures resulted in imprinted pNIPA particles with approximate spherical shape. An approximation of the Minnaert equation of a gas filled microbubble was applied to the imprinted pNIPA particles to support quantitative ultrasonic results. Synthesizing microbubbles would provide an accurate mathematical model to be developed, allowing quantitative resonance frequency shifts to be calculated and possible even predicted in presence of analyte. Amplitude of acoustic data recorded during quantification would increase due to sharper and more intense resonance frequencies gas-filled microbubbles have over polymer particles. Fine tuning of imprinted pNIPA microbubble resonance frequency would also be possible using the mathematical model and pNIPA microbubble physical properties.

Overall, with these added features the ultrasound quantification technique could measure new constituents with the potential of *in vivo* analyte mapping. This technique can result in a reliable, easy to use device for rapid clinical assessments.

Appendix A: Modified A-Mode Ultrasound Instrumentation, Response and Simulation

This section details characterisation of ultrasonic instrumentation used to acquire acoustic data in all experiments presented in this thesis. Instrumentation included a sample cell with two ultrasound transducers affixed to opposite faces, while ultrasound gel applied between transducer and cell ensured good surface contact. Transducers selected for application had characteristic frequencies of 1.9 MHz with narrow distribtion, and 5 MHz with wide frequency distribution. The narrow frequency band 1.9 MHz transducer (Advanced Technology Labs Inc., Pennsylvania, U.S.) converted electrical pulses generated by a Panametrics 500PR Pulser/Receiver (Panametrics Inc., Massachusetts, U.S.) into acoustic signals,

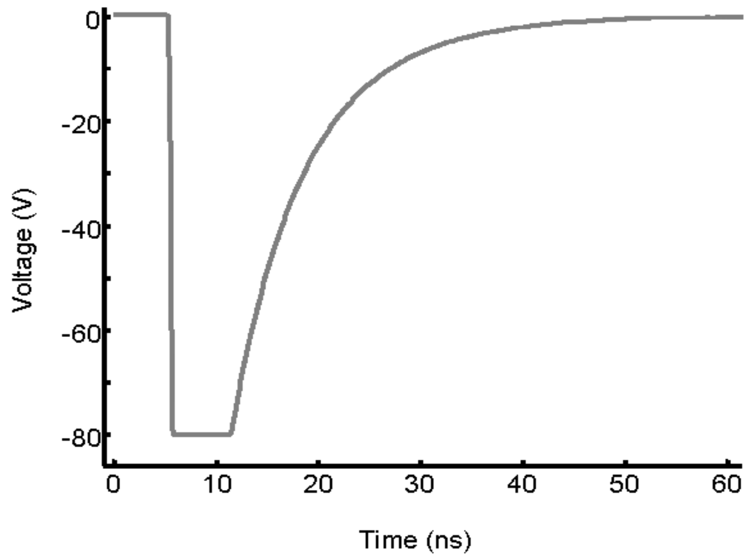

Figure B.1. Negative impulse produced by the Panametrics amplifier. A 10 ns 250 volt negative impulse is generated by the amplifier with adjustable impulse generation rate between 500 Hz and 5 kHz to drive ultrasound transducers. The effect on transducers is much like hitting a drum, the drum membrane (piezoelectric device) oscillates and exponentially decays to zero after being hit by the drum stick (negative impulse).

which were directed into the sample cell. Once ultrasound penetrated the cell and interacted with sample, the 5 MHz wide band transducer (Russel NDE Systems, Alberta, Canada) recorded acoustic signals on the opposite face.

Instrument response from various components was analysed independantly. Negative 10 ns impulses generated with -250 volt amplitude by the Panametrics amplifier were sent to the narrow band 1.9 MHz to produce ultrasound signals, as illustrated in Figure B.1. Equipment able to detect such a large negative voltage was not available, which explains the sharp cut off at -80 volts, however technical manuals for the amplifier specified the -250 volt peak voltage of the negative impulses. The 1.9 MHz transducer operating in reflectance mode driven by the Panametrics amplifier produced spectra illustrated in Figure B.2. Response from the entire

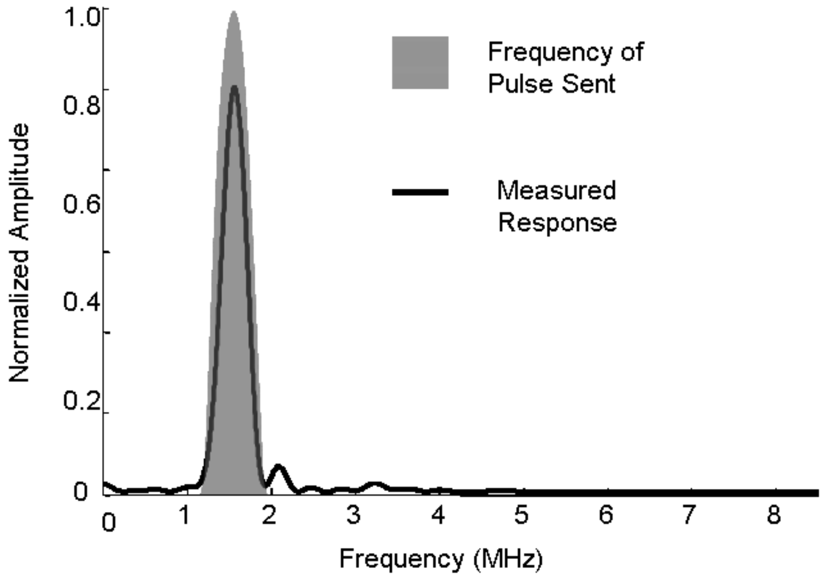

Figure B.2. Frequency response of the 1.9 MHz ultrasound transducer operating in reflectance. Frequency distribution of the transducer is shown in gray, and measured response is shown in black. Ultrasound was generated through this transducer from negative impulses produced by the Panametrics amplifier, shown in Figure A.1.

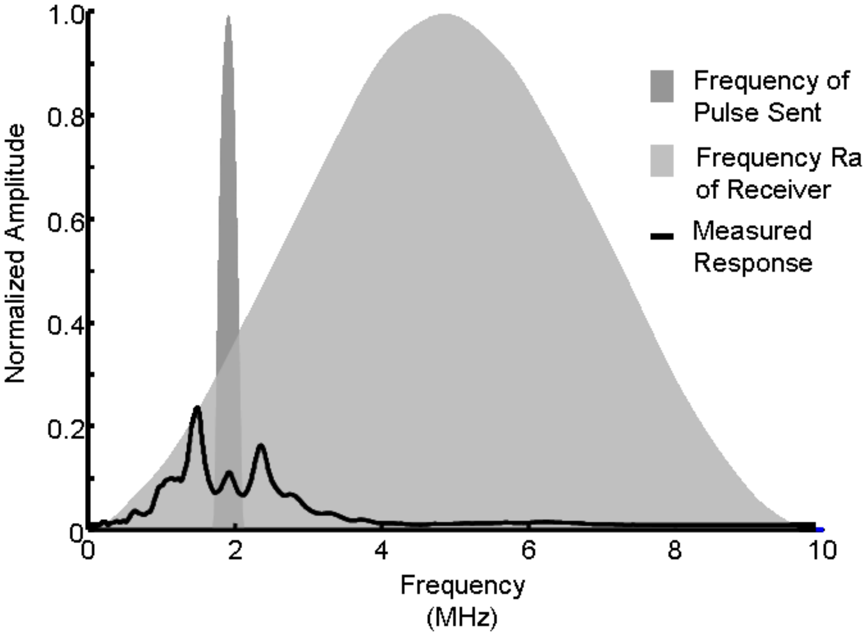

Figure B.3. Response of entire ultrasound system. Frequency response of the 1.9 MHz transducer sending pulses through the sample cell is shown in dark gray. Response of the 5 MHz wideband transducer is illustrated in light gray, according to manufacturerer's technical specifications. Total measured response of entire system including the Panametrics amplifier, transducers, and oscilloscope is denoted by the black line. Frequency response offsets of the two transducers were selected in order to maximize nonlinear broadening and enharmonic responses, since amplitude changes at multiple frequencies are used for analyte calibration.

ultrasound system is identified in Figure B.3. Bandwidth offsets of the two transducers provided maximal sensitivity to nonlinear broadening and enharmonic signals. Reception of these signals was crucial for ultrasonic quantification, as data recorded at multiple enharmonic frequencies of spectra were used to construct concentration models for analytes in question.

Typical data acquired using instrumentation for ultrasonic quantification is demonstrated in Figure B.4. As illustrated in the Figure, several pulses are recorded, however only the first pulse is processed. The remaining echoes caused by acoustic signals travelling back and forth through the sample cell are discarded, since moving through the cell multiple times distorts them beyond use. The first recorded pulse is then transformed into frequency spectra, such as figures A.1, A.2 and A.3, by performing Fourier transformation. Frequency spectra received presents many more peaks than the original ultrasound signal sent by the 1.9 MHz transducer. Two

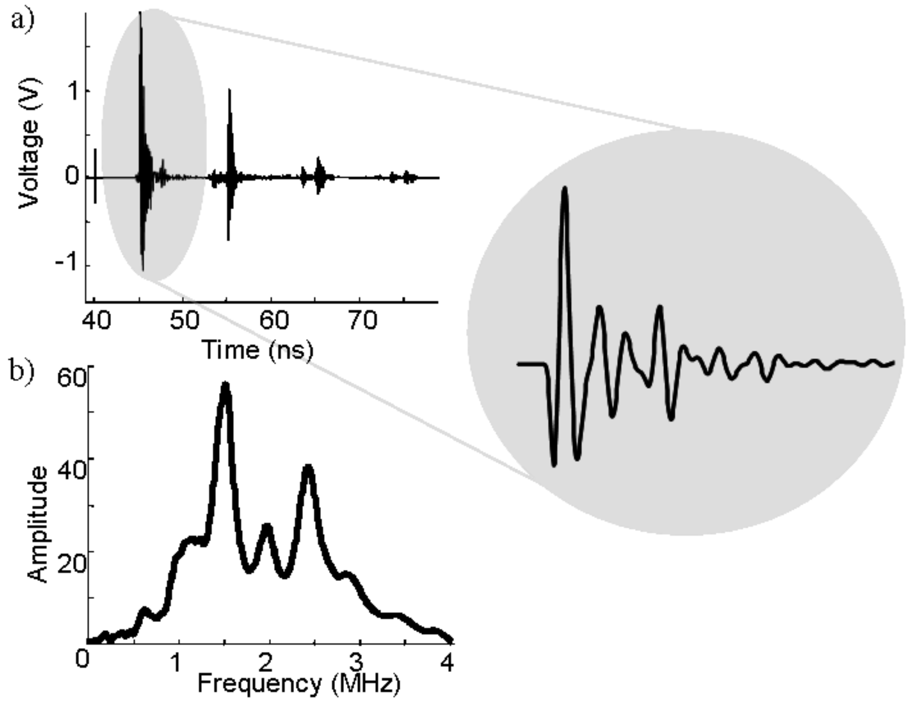

Figure B.4. Typical ultrasound data acquired during analyte quantification. Figure a) demonstrates ultrasound pulses that were transmitted through the sample cell and recorded with the 5 MHz wideband transducer. The first pulse transmitted through the cell is enlarged on the right. Subsequent pulses, each at lower amplitude than the previous, are also observed, occuring due to ultrasound reflections between cell walls. These were discarded due to being distorted beyond use. The Fourier spectrum of the first pulse is shown in b).

primary factors can be identified as sources for signal broadening. Although only first transmissions were used to construct frequency spectra, subtle echoes were created within plexiglass panes of the sample cell, adding a few components to the signal. These components occurred in a very small time frame, and were impossible to separate in the first transmission. The other source of signal broadening comes from wavefront distortion of ultrasound traversing multiple media boundaries at high acoustic pressures.[136] In acoustic signals travelling at high pressures with a planar wavefront, different parts of the wave travel at different speeds. Figure B.5 a)

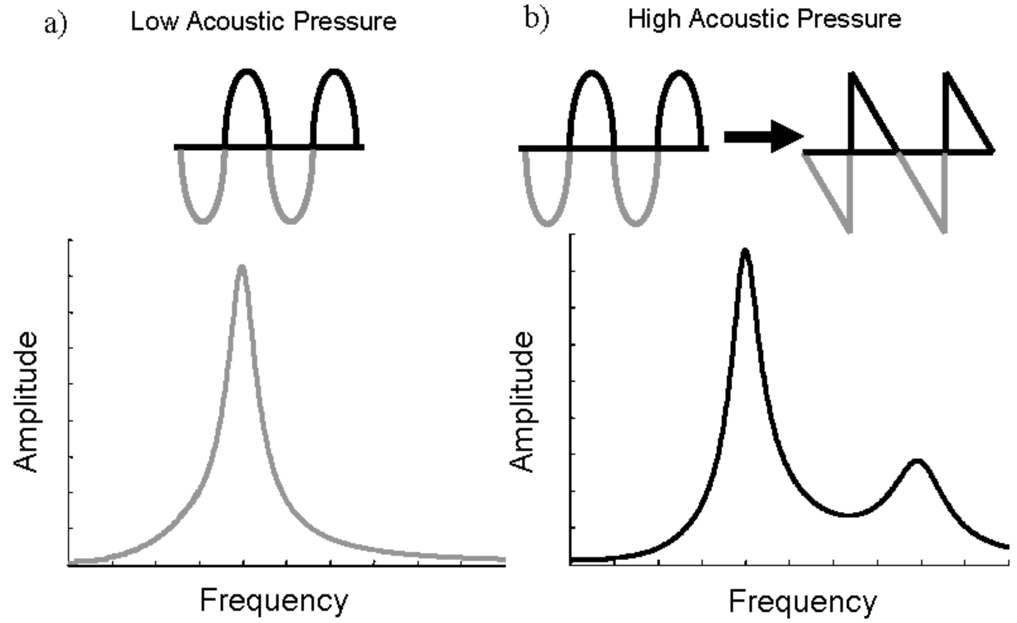

Figure B.5. Depiction of the effect of high acoustic pressure on sinusoidal waves. Figure a) illustrates behavior typical at low acoustic pressures of ultrasonic waves. A Fourier transformation is performed on the wave yielding a narrow frequency distribution. Figure b) demonstrates the effect of high acoustic pressures. The lower compressed portion of the wave travels faster than the upper rarefaction half, causing sawtooth-like distortion. A broader frequency distribution, and appearance of additional harmonic and enharmonic components ensues.

denotes a low pressure sinusoidal acoustic wave, and the corresponding Fourier spectrum. The effects of elevated acoustic pressure on a wave are illustrated in Figure B.5 b), demonstrating the compression half-cycle of the wave travelling faster than the rarefaction half-cycle, creating sawtooth-like distortion. The effect of sawtooth-like distortion on a wave, even in very subtle amounts, leads to significant signal broadening, appearance of enharmonic signals, and harmonic peaks, as illustrated in the Fourier spectrum of Figure B.5 b). These results are expected, since a perfect sawtooth wave can be broken down into a linear sum of an infinite amount of sinusoidal waves, and therefore any sinusoidal wave with sawtooth character will rapidly increase in complexity.

While the recorded ultasound signal is difficult to model, a simulation of the previously described factors can be applied to a mathematically generated wave. An approximation of the ultrasound signal travelling under ideal conditions at high acoustic pressures can then be visualized. A negative impulse can be constructed taking the form,

$$- \ \mathrm{in}(a) \cdot e^{-} \ , \qquad \text{Eq A.1}$$

where a and b are arbitrary parameters. When plotted on axes, a decaying impulse of this nature resembles the wave depicted in Figure B.6 a). The corresponding frequency spectrum was determined via Fourier transformation is shown in Figure B.6 b). We can apply the following operator function on the generated wave with a planar wavefront travelling in water,

$$p_i^{k+1} = p_i^k \left\{ \frac{1}{1 - \left(\frac{\beta \ \ \Delta z}{\Delta t \cdot \rho_o \cdot c_o{}^3} \right) \left(p_{i+1}^k - p_i^k \right)} \right\}, \qquad \text{Eq A.2}$$

where p_k^i is the planar wave function at spatial step k, time interval i, β is an arbitrary parameter equal to 3.5 for water at 20°C, Δt is the time interval, c_o is the speed of sound in the medium, Δz is the propagation step length, ρ_o is the density of the medium.[136] The operator function constructs the new shape of the wave at spatial step k+1 and time interval i, for each time interval specified. Selecting a small time and

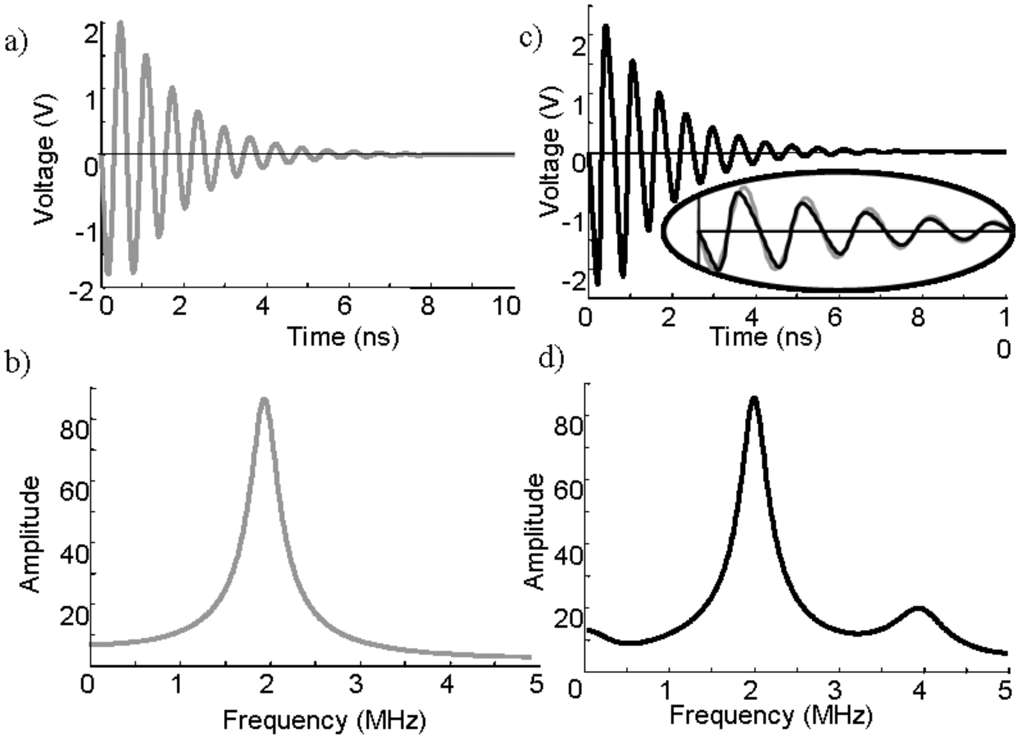

Figure B.6. Effect of high pressure acoustic distortion on a mathematically generated ideal decaying ultrasound pulse. Figure a) demonstrates a decaying pulse under low pressure conditions, with b) showing the associated Fourier spectrum illustrating one primary peak. Figure c) illustrates the effect of high pressure distortion on the decaying wave. Slight sawtooth-like distortion is noted, which is best visualized in the zoomed in portion of c) comparing the low and high pressure pulses. The associated Fourier spectrum in d) shows the appearance of a small harmonic peak, and increase in enharmonic signals as a result of the distortion.

spatial window demonstrates the sawtooth-like distortion present at elevated acoustic pressures, and is shown in Figure B.6 c), along with the frequency spectrum of the distorted wave denoted in d). As a result of the distortion, signal broadening takes place, denoted by a harmonic peak appearing at 4 MHz, and enharmonic frequencies showing general increase in amplitude between 2 and 4 MHz, as well as below 2 MHz. Band broadening in these results is very similar to the broadening seen in ultrasonic data recorded in this thesis, and is further emphasized by SMLR techniques preferentially selecting enharmonic frequencies to construct regression models.

MLR_HS3_DT.m
(Matlab program for loading, parsing and processing ultrasound data, excluding SMLR calculation)

```
function [freq_sel_index, act_conc, est_conc, see, rsqrd,
est_conc_cv, see_cv, rsqrd_cv] = MLR_HS3_DT(num_freqs, loocv_switch,
plot_switch, smooth_win, normalizeFFT_switch, blankcorr_switch,
phasecorr_switch, ftest_switch, meancenter_switch, outlier_switch,
scan_span)
% function MLR_HS3_DT.m
% McGill University
% David Troïani
% Feb 02, 2009
% MatLab 7.0+
%
% This script loads all *.mat ultrasound data files, calculates the
FFTs and performs Stagewise MLR
% on the data to extract trends for concentration
%
% num_freqs = number of frequencies to select for MLR
% loocv_switch = 1 enables Leave-One-Out Cross Validation, 0 disables
LOOCV
% plot_switch = 1 enables plots, 0 disables plots
% smooth_win = odd number, size of the smoothing window, 0 for no
smoothing
% normalizeFFT_switch = 1 normalizes sol and water FFTs according to
respective total power,
%                     2 normalizes sol FFTs and scales water FFTs
using index of highest power in sol,
%                     3 normalizes sol FFTS and scales water FFTs
using a ratio of highest powers from both spectra,
%                     0 no normalization
%                     (settings 2 and 3 are only useful if
blankcorr_switch is 1)
% blankcorr_switch = 1 blank corrects using water spectra, 0 for no
blank correction
```

```
% phasecorr_switch = 1 corrects for phase (shifts pulse according to
max peak),
%                    2 uses the entire pulse (and reflections) for
FFTs,
%                    0 for no phase correction (uses TimeStart and
TimeEnd inside the function)
% ftest_switch = 1 to calculate ftest for parsimony in MLR, 0
disables ftest calculation
% meancenter_switch = 1 to mean-center the FFT data, 0 disables mean-
centering
% outlier_switch = 1 to remove outliers that are more than
"std_outlier_factor" number of standard deviations from the mean
%                  0 for no outlier removal
% scan_span = array of 2 values specifying which scan number range to
use
%             0 to use the default values written in the function

if nargin <= 3,
    smooth_win = 0;
    normalizeFFT_switch = 0;
    blankcorr_switch = 0;
    phasecorr_switch = 0;
    ftest_switch = 0;
    meancenter_switch = 0;
    outlier_switch = 0;
    scan_span = 0;
end;

ConcLabels = ['1';'2';'3';'4';'5';'6']; % Text labels for the .mat
files
%Concs = [0 50 150 200 250 550] /1000/1000; %L analyte added
Concs = [0 50 125 200 250 550] /1000/1000; %L analyte added
ScanLabels = ['_Scan1';'_Scan2';'_Scan3';'_Scan4';'_Scan5'];% Text
labels for the .mat files
Units = 'nM';

Concs_MLR = zeros(size(ConcLabels, 1)*size(ScanLabels, 1), 1);
MolarMass = 153.1784; %g/mol of analyte
SampleVol = 4 / 1000; %L volume of each sample
StockVol = 100 / 1000; %L volume of the analyte stock solution
StockConc = ((0.0207 / 1000 / MolarMass) / StockVol); %mol/L conc of
analyte stock solution
Concs = Concs * StockConc ./ (SampleVol + Concs); %mol/L conc of
analyte in samples
Concs = Concs * 1000 * 1000 * 1000; % nM
act_conc = Concs;

std_outlier_factor = 1.4; % Multiply the standard dev of the spectra
means by this for optional outlier removal (default is 1.15)
TimeDataSize = 5000; % Size of the time domain data array
TimeStart = 460; % Starting index to take FFT
TimeEnd = 1100; % Ending index to take FFT
if scan_span == 0,
    MaxTimeScanNumber = 3600;
    MinTimeScanNumber = 1;
else,
    MinTimeScanNumber = scan_span(1);
    MaxTimeScanNumber = scan_span(2);
end
FFTPadNum = 10000; % Padding size of FFT
```

```
SamplingFreqMHz = 50; % Frequency of aquisition (for SDS200
oscilloscope, 12.5 MHz, for HS3, 50 MHz)
FFTFreqAxis = [0:SamplingFreqMHz./2/((FFTPadNum./2)-
1):SamplingFreqMHz./2]; % Frequency axis for plotting the FFTs

TimeData = zeros(TimeDataSize, size(ConcLabels, 1)*size(ScanLabels,
1)); % pre-allocate array for time data
TimeDataWater = zeros(TimeDataSize, size(ConcLabels,
1)*size(ScanLabels, 1)); % pre-allocate array for time data
FFTData = zeros(round(FFTPadNum./2), size(ConcLabels,
1)*size(ScanLabels, 1)); % pre-allocate array for FFT data
temp_FFTData = zeros(round(FFTPadNum./2), 1); % pre-allocate array
for temporary FFT data
temp_TimeData = zeros(TimeDataSize, MaxTimeScanNumber-
MinTimeScanNumber);

DataIndex = 0; % This index will count how many time data arrays
there will be (at each temperature, then each temperature per conc,
at all the concs)

disp('Loading data and processing...');

for i = 1:size(ConcLabels, 1),
    for j = 1:size(ScanLabels, 1),

        DataIndex = DataIndex + 1;

        eval(['load ', ConcLabels(i,:), ScanLabels(j,:), '.mat']);
        for k = MinTimeScanNumber:MaxTimeScanNumber,
            temp_TimeData(:,k-MinTimeScanNumber+1) = tpd(k).Data;
        end;
        temp_TimeDataMean = mean(temp_TimeData, 2); % Average all the
scans into 1
        TimeData(:, DataIndex) = temp_TimeDataMean;
        temp_TimeDataMean = temp_TimeDataMean -
mean(temp_TimeDataMean(1:TimeStart)); % Voltage Offset correction

        if phasecorr_switch == 1,
            [not_used, MaxIndex] = max(temp_TimeDataMean);
            temp_TimeDataMean = temp_TimeDataMean(MaxIndex-
50:MaxIndex+575); % Phase correction
            temp_FFTData = abs(fft(temp_TimeDataMean, FFTPadNum));
        elseif phasecorr_switch == 2,
            temp_FFTData = abs(fft(temp_TimeDataMean, FFTPadNum));
        else,
            temp_FFTData =
abs(fft(temp_TimeDataMean(TimeStart:TimeEnd), FFTPadNum));
        end

        temp_FFTData =
temp_FFTData(1:round(length(temp_FFTData)./2));

        if smooth_win ~= 0,
            temp_FFTData = smooth(temp_FFTData, smooth_win);
        end;
        if normalizeFFT_switch >= 1,
            temp_FFTData = temp_FFTData./sum(temp_FFTData);
        end;
        FFTData(:, DataIndex) = temp_FFTData;

        clear tpd;

        if blankcorr_switch == 1,

            % Water blank correction
            eval(['load ', ConcLabels(i,:), ScanLabels(j,:),
'_Water.mat']);
            for k = MinTimeScanNumber:MaxTimeScanNumber,
                temp_TimeData(:,k-MinTimeScanNumber+1) = tpd(k).Data;
            end;
            temp_TimeDataMean = mean(temp_TimeData, 2); % Average all
the scans into 1
            TimeDataWater(:, DataIndex) = temp_TimeDataMean;
            temp_TimeDataMean = temp_TimeDataMean -
mean(temp_TimeDataMean(1:TimeStart)); % Voltage Offset correction

            if phasecorr_switch == 1,
                [not_used, MaxIndex] = max(temp_TimeDataMean);
                temp_TimeDataMean = temp_TimeDataMean(MaxIndex-
50:MaxIndex+575); % Phase correction
                temp_FFTData = abs(fft(temp_TimeDataMean,
FFTPadNum));
            elseif phasecorr_switch == 2,
                temp_FFTData = abs(fft(temp_TimeDataMean,
FFTPadNum));
            else,
                temp_FFTData =
abs(fft(temp_TimeDataMean(TimeStart:TimeEnd), FFTPadNum));
            end

            temp_FFTData =
temp_FFTData(1:round(length(temp_FFTData)./2));

            if smooth_win ~= 0,
                temp_FFTData = smooth(temp_FFTData, smooth_win);
            end;
            if normalizeFFT_switch >= 1 & normalizeFFT_switch <= 3,
                if normalizeFFT_switch == 1,
                    temp_FFTData = temp_FFTData./sum(temp_FFTData);
                elseif normalizeFFT_switch == 2,
                    [not_used, MaxIndex] = max(FFTData(:,DataIndex));
                    temp_FFTData = temp_FFTData .* (FFTData(MaxIndex,
DataIndex) ./ temp_FFTData(MaxIndex));
                elseif normalizeFFT_switch == 3,
                    [not_used, MaxIndex] = max(FFTData(:,DataIndex));
                    [not_used, MaxIndex2] = max(temp_FFTData);
                    temp_FFTData = temp_FFTData .* (FFTData(MaxIndex,
DataIndex) ./ temp_FFTData(MaxIndex2));
                end;
            end;

           FFTData(:, DataIndex) = FFTData(:, DataIndex) -
temp_FFTData;

           clear tpd;

       end;
```

```
    end
end

disp('Done.');

for i = 1:DataIndex,
    Concs_MLR(i) = Concs(ceil(i./(size(ScanLabels, 1))));
end;

if meancenter_switch == 1,
    mcn_fftdata = FFTData'; % Transposes the data
    [mcn_fftdata,mx] = mncn(mcn_fftdata); % Mean Center subtraction
    FFTData = mcn_fftdata';
end

disp('Calculating regression...');

[r,m,see,freq_sel_index,est_conc,rsqrd]=stgmlr(Concs_MLR,
FFTData(round(0.02*(FFTPadNum./2)):round(0.16.*(FFTPadNum./2)), :),
num_freqs);
if loocv_switch == 1,
    [coef, est_conc_cv, coefall, rsqrd_cv,
see_cv]=jonloo2(FFTData(round(0.02.*(FFTPadNum./2)):round(0.16.*(FFTP
adNum./2)), :), Concs_MLR', freq_sel_index, 0);
else,
    est_conc_cv = 0;
    see_cv = 0;
    rsqrd_cv=0;
end;
freq_sel_index = freq_sel_index + round(0.02.*(FFTPadNum./2));

indextoremove = [];
old_Concs_MLR = Concs_MLR;
old_est_conc_cv = est_conc_cv;
old_est_conc = est_conc;

if outlier_switch == 1,
    temp_meanConcArray =
zeros(length(old_est_conc)./size(ScanLabels,1),1);
    temp_meanstd_req =
zeros(length(old_est_conc)./size(ScanLabels,1),1);
    hk2 = 1;

    if loocv_switch == 1,
        for gaga = 1:size(ScanLabels,1):length(est_conc_cv);
            temp_meanConcArray(hk2) =
mean(est_conc_cv(gaga:gaga+size(ScanLabels,1)-1));
            temp_meanstd_req(hk2) =
std(est_conc_cv(gaga:gaga+size(ScanLabels,1)-1)) .*
std_outlier_factor;
            hk2 = hk2+1;
        end
    else,
        for gaga = 1:size(ScanLabels,1):length(est_conc);
            temp_meanConcArray(hk2) =
mean(est_conc(gaga:gaga+size(ScanLabels,1)-1));
            temp_meanstd_req(hk2) =
std(est_conc(gaga:gaga+size(ScanLabels,1)-1)) .* std_outlier_factor;
```

```
            hk2 = hk2+1;
        end
    end

    kraken = 0;
    if loocv_switch == 1,
        for xyz2=1:DataIndex,
           kraken = ceil(xyz2./size(ScanLabels,1));
           if (est_conc_cv(xyz2) >
temp_meanConcArray(kraken)+temp_meanstd_req(kraken)) ||
(est_conc_cv(xyz2) < temp_meanConcArray(kraken)-
temp_meanstd_req(kraken))
                indextoremove = [indextoremove, xyz2];
           end
        end;
    else
        for xyz2=1:DataIndex,
           kraken = ceil(xyz2./size(ScanLabels,1));
           if (est_conc(xyz2) >
temp_meanConcArray(kraken)+temp_meanstd_req(kraken)) ||
(est_conc(xyz2) < temp_meanConcArray(kraken)-
temp_meanstd_req(kraken))
                indextoremove = [indextoremove, xyz2];
           end
        end;
    end
    FFTData(:, indextoremove) = [];
    Concs_MLR(indextoremove) = [];

    [r,m,see,freq_sel_index,est_conc,rsqrd]=stgmlr(Concs_MLR,
FFTData(round(0.02*(FFTPadNum./2)):round(0.16.*(FFTPadNum./2)), :),
num_freqs);
    if loocv_switch == 1,
        [coef, est_conc_cv, coefall, rsqrd_cv,
see_cv]=jonloo2(FFTData(round(0.02.*(FFTPadNum./2)):round(0.16.*(FFTP
adNum./2)), :), Concs_MLR', freq_sel_index, 0);
    else,
        est_conc_cv = 0;
        see_cv = 0;
        rsqrd_cv=0;
    end;
    freq_sel_index = freq_sel_index + round(0.02.*(FFTPadNum./2));
end

xyz2 = 1;
kkx = 1;
ErrorBar_mean = zeros(length(act_conc), 1);
ErrorBar_lower = zeros(length(act_conc), 1);
ErrorBar_upper = zeros(length(act_conc), 1);

if loocv_switch == 1,
    while max(xyz2) < length(Concs_MLR)
        [xyz2] = find(Concs_MLR == Concs_MLR(xyz2));
        ErrorBar_mean(kkx) = mean(est_conc_cv(xyz2));
        ErrorBar_upper(kkx) = std(est_conc_cv(xyz2));
        ErrorBar_lower(kkx) = std(est_conc_cv(xyz2));
        xyz2 = max(xyz2)+1;
        kkx = kkx+1;
    end
else
```

```
    while max(xyz2) < length(Concs_MLR)
        [xyz2] = find(Concs_MLR == Concs_MLR(xyz2));
        ErrorBar_mean(kkx) = mean(est_conc(xyz2));
        ErrorBar_upper(kkx) = std(est_conc(xyz2));
        ErrorBar_lower(kkx) = std(est_conc(xyz2));
        xyz2 = max(xyz2)+1;
        kkx = kkx+1;
    end
end;

disp('Done.');

if ftest_switch == 1,
    see_array = zeros(1, num_freqs-1);
    f_array = zeros(1, num_freqs-1);
    fcrit_array = zeros(1, num_freqs-1);
    disp('Performing F-tests...');
    for i = 2:num_freqs,
        [not_used, not_used2, see1, freq_sel_index, est_conc,
rsqrd1]=stgmlr(Concs_MLR,
FFTData(round(0.02.*(FFTPadNum./2)):round(0.16.*(FFTPadNum./2)), :),
i-1);
        %see1=std(Concs_MLR - est_conc); % Standard error calculation
        if loocv_switch == 1,
            [not_used, est_conc_cv, not_used3, rsqrd_cv1,
see1]=jonloo2(FFTData(round(0.02.*(FFTPadNum./2)):round(0.16.*(FFTPad
Num./2)), :), Concs_MLR', freq_sel_index, 0);
            %see1=std(Concs_MLR - est_conc_cv'); % Standard error
calculation
        end;

        [not_used, not_used2, see2, freq_sel_index, est_conc,
rsqrd2]=stgmlr(Concs_MLR,
FFTData(round(0.02.*(FFTPadNum./2)):round(0.16.*(FFTPadNum./2)), :),
i);
        %see2=std(Concs_MLR - est_conc); % Standard error calculation
        if loocv_switch == 1,
            [not_used, est_conc_cv, not_used3, rsqrd_cv2,
see2]=jonloo2(FFTData(round(0.02.*(FFTPadNum./2)):round(0.16.*(FFTPad
Num./2)), :), Concs_MLR', freq_sel_index, 0);
            %see2=std(Concs_MLR - est_conc_cv'); % Standard error
calculation
        end;

        freq_sel_index = freq_sel_index +
round(0.02.*(FFTPadNum./2));

        see_array(i-1) = see2;

        N = size(ConcLabels, 1)*size(ScanLabels, 1);
        p = i;

        f = see1.^2 / see2.^2;

        f_array(i-1) = f;

        fcrit = ftest(0.90, N-(p-1), N-p, 2);

        fcrit_array(i-1) = fcrit;
        if loocv_switch == 1,
            disp([num2str(i),' factors: F = ', num2str(f), '    Fcrit
= ', num2str(fcrit), '   seeLOOCV = ', num2str(see2), Units, '
r2LOOCV = ', num2str(rsqrd_cv2)]);
        else,
            disp([num2str(i),' factors: F = ', num2str(f), '    Fcrit
= ', num2str(fcrit), '   see = ', num2str(see2), Units, '    r2 = ',
num2str(rsqrd2)]);
        end
    end;
    disp('Done.');

    if plot_switch == 1,
        Figure;
        plot([2:num_freqs], see_array);
        xlabel('Number of Frequency
Components','FontName','Arial','FontSize',16);
        ylabel(['Standard Error of Estimate ', '(', Units,
')'],'FontName','Arial','FontSize',16);
        set(gca,'FontName','Arial');
        set(gca,'FontSize',16);
        set(gca,'LineWidth',1.5);
        set(gcf,'Color',[1 1 1]);
        Figure;
        plot([2:num_freqs], f_array, '-b', [2:num_freqs],
fcrit_array, '-r');
        legend(strvcat('F','Fcrit'));
        xlabel('Number of Frequency
Components','FontName','Arial','FontSize',16);
        ylabel('F-Test result','FontName','Arial','FontSize',16);
        title('Comparing one model to the
next','FontName','Arial','FontSize',16);
        set(gca,'FontName','Arial');
        set(gca,'FontSize',16);
        set(gca,'LineWidth',1.5);
        set(gcf,'Color',[1 1 1]);

        for i = 2:num_freqs,
            f_array(i-1) = see_array(i-1).^2 ./ see_array(num_freqs-
1).^2;
            fcrit_array(i-1) = ftest(0.95, N-i, N-num_freqs, 2);
        end;

        Figure;
        plot([2:num_freqs], f_array, '-b', [2:num_freqs],
fcrit_array, '-r');
        legend(strvcat('F','Fcrit'));
        xlabel('Number of Frequency
Components','FontName','Arial','FontSize',16);
        ylabel('F-Test result','FontName','Arial','FontSize',16);
        title('Comparing the best model to
each','FontName','Arial','FontSize',16);
        set(gca,'FontName','Arial');
        set(gca,'FontSize',16);
        set(gca,'LineWidth',1.5);
        set(gcf,'Color',[1 1 1]);
    end;
 end
```

```
disp('Regression Results...');

disp('Frequencies Selected for MultiLinear Regression...');
disp([num2str(FFTFreqAxis(freq_sel_index)), ' MHz']);
disp('Concentration Range...');
disp([num2str(act_conc), ' ', Units]);
if outlier_switch == 1,
    disp([num2str(length(old_Concs_MLR)), ' Spectra, ',
num2str(length(old_est_conc)-length(Concs_MLR)), ' Removed as
Outliers, ', num2str(length(Concs_MLR)), ' Spectra Used']);
    disp(['StdDev Outlier Factor: ', num2str(std_outlier_factor)]);
else
    disp([num2str(length(Concs_MLR)), ' Spectra']);
end

Sbk_index = find(Concs_MLR == 0);
if isempty(Sbk_index), Sbk_index = [1 2]; end;
if loocv_switch == 1,
    Sbk = std(est_conc_cv(Sbk_index));
else,
    Sbk = std(est_conc(Sbk_index));
end
disp(['Limit of Detection... ', num2str(3*Sbk), ' ', Units]);

disp(['r^2  : ', num2str(rsqrd), '    see  : ', num2str(see),' ',
Units]);

if loocv_switch == 1,
    disp(['r^2LOOCV: ', num2str(rsqrd_cv), '    seeLOOCV: ',
num2str(see_cv),' ', Units]);
end;

if plot_switch == 1,
    disp('Plotting results...');

    Figure;
    hold on;
    plot(TimeData);
    xlabel('Sample Data Point','FontName','Arial','FontSize',16);
    ylabel('Volts','FontName','Arial','FontSize',16);
    set(gca,'FontName','Arial');
    set(gca,'FontSize',16);
    set(gca,'LineWidth',1.5);
    set(gcf,'Color',[1 1 1]);

    if blankcorr_switch == 1,
        Figure;
        hold on;
        plot(TimeDataWater);
        xlabel('Water Data Point','FontName','Arial','FontSize',16);
        ylabel('Volts','FontName','Arial','FontSize',16);
        set(gca,'FontName','Arial');
        set(gca,'FontSize',16);
        set(gca,'LineWidth',1.5);
        set(gcf,'Color',[1 1 1]);
    end

    Figure;
    hold on;
    plot(FFTFreqAxis, FFTData, 'LineWidth',2);
    xlim([0 FFTFreqAxis(round(0.16.*(FFTPadNum./2)))]);
    xlabel('Frequency (MHz)','FontName','Arial','FontSize',16);
    if normalizeFFT_switch == 1,
        ylabel('Normalized
Amplitude','FontName','Arial','FontSize',16);
    else,
        ylabel('Amplitude','FontName','Arial','FontSize',16);
    end;
    set(gca,'FontName','Arial');
    set(gca,'FontSize',16);
    set(gca,'LineWidth',1.5);
    set(gcf,'Color',[1 1 1]);

    if outlier_switch == 1,
        Figure;
        hold on;
        plot(act_conc, act_conc, 'r-','LineWidth',2);
        if loocv_switch == 1,
            plot(old_Concs_MLR, old_est_conc_cv, 'bo','LineWidth',
2);
        else,
            plot(old_Concs_MLR, old_est_conc, 'bo','LineWidth', 2);
        end;
        xlabel(['Actual Concentration
(',Units,')'],'FontName','Arial','FontSize',16);
        ylabel(['Estimated Concentration
(',Units,')'],'FontName','Arial','FontSize',16);
        set(gca,'FontName','Arial');
        set(gca,'FontSize',16);
        set(gca,'LineWidth',1.5);
        set(gcf,'Color',[1 1 1]);
        xlim([0 act_conc(length(act_conc))]);
    end;

    Figure;
    hold on;
    plot(act_conc, act_conc, 'k-','LineWidth',1);
    errorbar(act_conc, ErrorBar_mean, ErrorBar_lower, ErrorBar_upper,
'.k', 'MarkerSize', 20, 'LineWidth',1);
    xlabel(['Known Conc
(',Units,')'],'FontName','Arial','FontSize',16);
    ylabel(['Estimated Conc
(',Units,')'],'FontName','Arial','FontSize',16);
    set(gca,'FontName','Arial');
    set(gca,'FontSize',16);
    set(gca,'LineWidth',1.5);
    set(gcf,'Color',[1 1 1]);
    xlim([0 act_conc(length(act_conc))]);

    Figure;
    hold on;
    plot(act_conc, act_conc, 'r-','LineWidth',2);
    if loocv_switch == 1,
        plot(Concs_MLR, est_conc_cv, 'bo','LineWidth', 2);
    else,
        plot(Concs_MLR, est_conc, 'bo','LineWidth', 2);
    end;
    xlabel(['Actual Concentration
(',Units,')'],'FontName','Arial','FontSize',16);
    ylabel(['Estimated Concentration
(',Units,')'],'FontName','Arial','FontSize',16);
```

```
    set(gca,'FontName','Arial');
    set(gca,'FontSize',16);
    set(gca,'LineWidth',1.5);
    set(gcf,'Color',[1 1 1]);
    curaxis = axis;
    if loocv_switch == 1,
        text(curaxis(2).*0.5, curaxis(4).*0.1, strvcat(['r^2 LOOCV = 
', num2str(rsqrd_cv, 2)],['see LOOCV = ', num2str(see_cv, 2), ' 
',Units]),'FontName','Arial','FontSize',16 );
    else,
        text(curaxis(2).*0.5, curaxis(4).*0.1, strvcat(['r^2 = ', 
num2str(rsqrd, 2)],['see = ', num2str(see, 2), ' 
',Units]),'FontName','Arial','FontSize',16 );
    end
    xlim([0 act_conc(length(act_conc))]);
    disp('Done.');

%     reps = [1:1:size(FFTData,2)];
%     Figure;
% 
surf(FFTFreqAxis(round(0.02.*(FFTPadNum./2)):round(0.16.*(FFTPadNum./
2))), reps', 
FFTData(round(0.02.*(FFTPadNum./2)):round(0.16.*(FFTPadNum./2)), 
:)');
%     shading interp;
%     ylabel('Spectrum Number','FontName','Arial','FontSize',16);
%     xlabel('Frequency (MHz)','FontName','Arial','FontSize',16);
%     if normalizeFFT_switch == 1,
%         zlabel('Normalized 
Amplitude','FontName','Arial','FontSize',16);
%     else,
%         zlabel('Amplitude','FontName','Arial','FontSize',16);
%     end;
%     set(gca,'FontName','Arial');
%     set(gca,'FontSize',16);
%     set(gca,'LineWidth',1.5);
%     set(gcf,'Color',[1 1 1]);
end;

return;
```

Appendix B: Current Phase Transition Nanogel Applications

What makes nanogels attractive as contrast agents is the ultrasonic profile change associated with polymer network collapse, known as volume phase transition. At equilibrium, nanogels exist in loose, swollen networks (known as the swollen phase) hydrogen bonding between interconnected polymer chains and solvent allows solvent molecules to freely pass through the polymer network. When certain environmental parameters including temperature, hydrostatic pressure, ionic strength and hydrophobic interactions are increased passed a threshold point, the polymer network undergoes a conformation change and shrinks in size.[79, 80] In the collapsed phase, the nanogel exists as small network particles The process from swollen to collapsed phase is characterized by intra network forces dominating over solvent-chain hydrogen bonding. This process, spinodal decomposition, is qualitatively described by the Flory-Huggins theory, where the free energy per unit volume of polymer interactive with solvent can be shown as,

$$\Delta F = \Delta F_{Mix} + \Delta F_{Elast} + \Delta F_{Ion}, \qquad \text{Eq. B.1}$$

where,

$$\Delta F_{Mix} = k_B T \left[\chi \Phi (1 - \Phi) + (1 - \Phi) \ln (1 - \Phi) \right] \text{,Eq. B.2a}$$

$$\Delta F_{Elast} = \frac{3 k_B T \Phi}{2 N_c} \left[\left(\frac{\Phi_0}{\Phi} \right)^{2/3} - 1 - \ln \left(\frac{\Phi_0}{\Phi} \right) \right], \qquad \text{Eq. B.2b}$$

$$\Delta F_{Ion} = 0, \qquad \text{Eq. B.2c}$$

χ in the Flory-Huggins interaction parameter, k_B is the Boltzman constant, T is temperature, N_c is the number of monomers per chain, Φ is polymer volume fraction, Φ_0 is polymer volume fraction in the reference state, and free energy contribution from ionic forces (ΔF_{Ion}) is considered inconsequential.[137,138] The phase transition critical point occurs when the contribution from the free energy of mixing and free energy of elasticity are equal and cancel each other out.

Unique properties of phase transition nanogels lend them to interesting uses. Poly(n-isopropylacrylamide) (pNIPA) is among the most commonly used nanogel, due to a low phase transition temperature of 33.6°C,[139] and bulk modulus of approximately 14 kPa making the nanogel deformable.[140] Many diverse applications utilize pNIPA copolymers for varied uses. One application demonstrates the synthesis of pNIPA copolymer shape memory gels that can reversibly transform into a predefined shape after phase transition.[141] In the collapsed phase, these memory gels occupy simple linear shapes. When environmental conditions are shifted to induce phase transition, pNIPA sections of the copolymer deform in a predefined manner, while remaining sections of the copolymer either enlarge linearly or remain unchanged. This unique construction allows memory gels to take on a variety of complex shapes, such as pentagons and swirls. Another application utilizes the collapsing properties of nanogels based on pNIPA for *in vivo* drug delivery systems.[142] Synthesis of pNIPA based nanogels is carried out and subsequently added to a solution containing a particular drug compound. Phase transition is induced to trap drug molecules inside collapsed nanogel particles, which are then delivered to target areas (such as organs in the human body). Drug molecules can then be released through phase transition back to swollen phase. Copolymers with analyte selective end groups can increase sensitivity of drug delivery polymers to target areas, raising efficacy of the delivery system. Ultrasonic profiles of pNIPA nanogels depend on environmental parameters, and can reversibly change as the nanogel transitions from one state into another. This particular property makes pNIPA nanogels suitable for use as contrast agents.

Appendix C: Ultrasonic Analyte Image Mapping

Combining ultrasonic analyte quantification with ultrasound's proven imaging capabilities provides an avenue for potential sample analyte image mapping. Applications of analyte image mapping in medical fields could provide rapid *in vivo* analyte quantification and localization in one rapid, minimally invasive process. Ultrasound analyte image mapping would be akin to ultrasonic fetal imaging (see Figure 2.3), except colored areas would signify presence of a particular analyte, instead of tissue layers or bone. Localized frequency shifts in ultrasonic profile of nanogel administered or injected into samples would reveal concentration densities of particular analytes with spatial resolution.

Preliminary assays done in this thesis involved spatially identifying a sealed tubular sac containing either water, molecularly imprinted pNIPA nanogel, or molecularly imprinted pNIPA nanogel with 100 μM theophylline, set within a water bath, based on frequency profile changes. A modified version of ultrasonic instrumentation and programs in previous experiments was used for these assays. A 1.9 MHz ultrasound transducer (Advanced Technology Labs Inc., Pennsylvania, U.S.) generating pulses was fixed to one side of the water bath, in direct contact with liquid. This transducer was driven by a Panametrics 500PR Pulser/Receiver (Panametrics Inc., Massachusetts, U.S.), generating 10 ns 250 V negative impulses. A 5 MHz wideband transducer (Advanced Technology Labs Inc., Pennsylvania, U.S.) was mounted on a servo motor to receive ultrasonic backscatter, also in direct contact with liquid. The position of the 5 MHz transducer was controlled by a HiTEC HS-5735MG digital quarter scale servo motor (HiTEC RCD Inc., California, U.S.) operated by a Basic STAMP II (Parallax Inc., Rocklin CA, USA). The Basic STAMP II was programmed to receive motor positioning data via serial port connection from Matlab, and direct the servo motor to the correct position. Acoustic data was acquired and digitized with a SDS 200 oscilloscope (SoftDSP Co., Seoul, Korea) sampling at 12.5 MHz using 9 bit A/D conversion. The instrument setup is shown in Figure D.1.

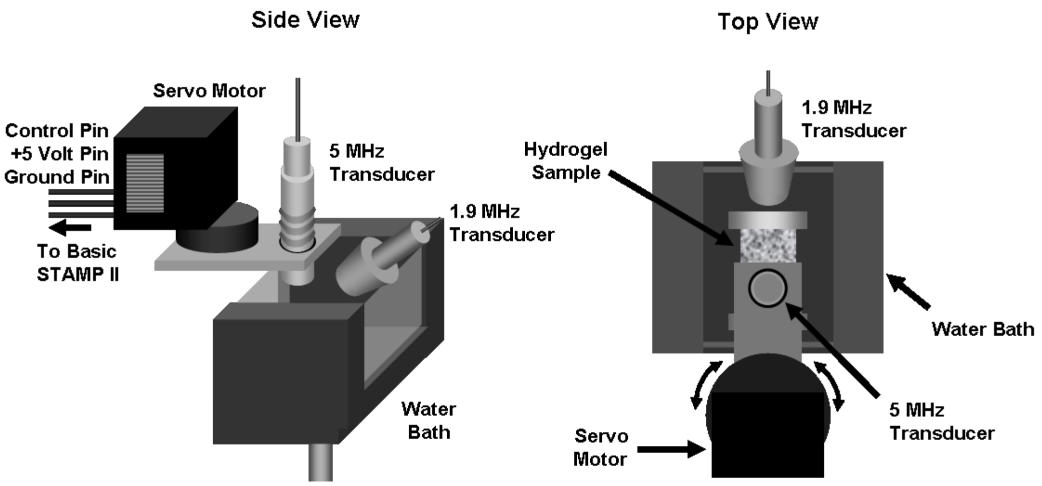

Figure D.1. Depiction of ultrasound instrument setup for ultrasound nanogel imaging experiments. A 1.9 MHz transducers was fixed to one side of the water bath, while the 5 MHz wideband transducer was mounted on a servo motor operated by a Basic STAMP II for variable positioning. Both transducers were in direct contact with liquid to maximize signal reception. A sealed imprinted pNIPA nanogel sample was placed at the bottom of the water bath for detection.

The program coded in Matlab instructed the Basic STAMP II to position the servo motor at the beginning end of its arc. The Matlab program then acquired an ultrasound scan, and adjusted the motor position a slight distance down the arc. The process was repeated, until a total of 56 scans were acquired over the scanning pathway. Ultrasonic data acquired at each of the 56 motor positions was processed into one spectrogram in Matlab. Several spectrograms were obtained, illustrating amplification and attenuation at several frequencies when the scanning transducer was above the imprinted pNIPA nanogel sample in the water bath. The spectrogram results are reported in figures D.2 a) and b). A data plot of amplitude at a notable frequency (5.4 MHz) related to transducer position is shown in Figure D.2 c). In each

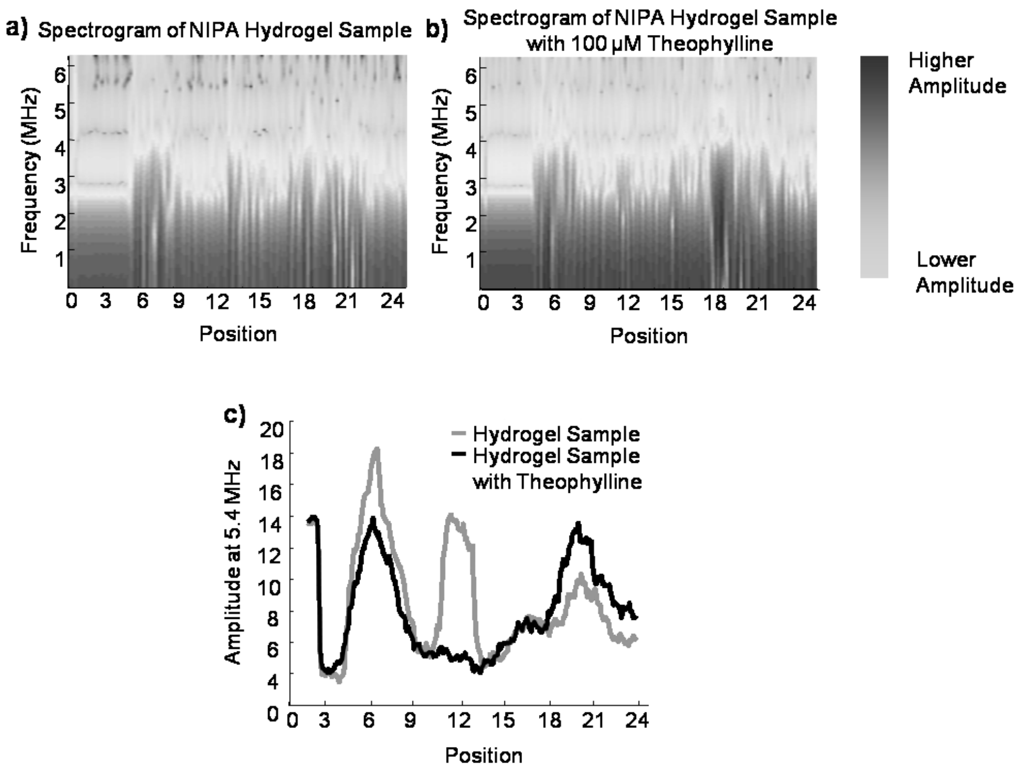

Figure D.2. Spectrograms assembled from ultrasonic data taken of a sealed sample in a water bath. The sealed sample was placed between transducer positions 6 and 24. Differences in amplitude at various frequencies between spectrograms between positions 6 and 24 are visible, notably between positions 12 and 18, differentiating samples containing: a) imprinted pNIPA nanogel, b) imprinted ppNIPA nanogel with 100 μM theophylline. Figure c) illustrates a data plot of ultrasound amplitude relative to position at 5.4 MHz. High amplitude backscatter at positions 6 and 24 were mostly caused by ultrasonic reflection from the edges of the tubular sealed sample.

spectrogram, the sealed sample was placed between transducer positions 6 and 24. Sample signal was most prominent when the transducer was directly above the tubular sac, notable between transducer positions 12 and 18, allowing ultrasound to properly penetrate the sample, interact with imprinted nanogel, and reflect back to the recording transducer with a different ultrasound profile. Amplitude differences at several frequencies are visible in this range, with those between positions 12 and 18 most indicative of sample type. Each sample had characteristic frequency profiles, with nanogel samples demonstrating general attenuation compared to the water sample. Due to the lack of color reproduction in Figure D.2., some detail is less apparent, however differences between spectrograms is still visible. Figure B.2 c) more easily demonstrates frequency amplitude differences between the three samples at two characteristic frequencies. Addition of theophylline to the imprinted nanogel sample provided further ultrasonic attenuation between positions 12 and 18 at the specific frequency shown in Figure D.2 c). Edges of the tubular plastic sac containing samples reflected ultrasound to a greater degree, illustrated by a general increase in ultrasonic amplitude across near transducer positions 6 and 24, where the transducer hovered over the edges of the sample.

Overall, these results demonstrate that different samples sealed in a container with multiple media boundaries separating transducer and sample yielded different ultrasonic profiles. Spatial identification along scanning pathway was also possible by examining attenuation and amplification of ultrasound at the various transducer positions. Imprinted nanogel samples with and without target analyte also provided different frequency characteristics, presenting the possibility of spatial resolution with simultaneous analyte quantification.

BS_SerialServoControl_v3.bs2
(control code for Basic STAMP II)

```
' {$STAMP BS2}
' {$PBASIC 2.5}

' BASIC Stamp Servo Motor Test Program
' Version 3.0
' David Troïani
' March 24, 2005

' This program waits for a decimal pulse width value sent through the
' serial port by MatLab, and changes the position of the HiTEC HS
' 5735MG digital quarter scale standard servo motor accordingly.

Servo_pin   CON 12      'I/O pin that is connected to servo
Pulse_width VAR Word    'Hold the pulse widths sent to the motor

Pulse_width = 0

Main:
SERIN 16, 84, [DEC Pulse_width] 'SERIN Pin, Baudrate, variable
                                 'Pin = 16, the BASIC Stamp's serial
input pin
                                 'Baudrate = 396, which is the code
for 2400
                                 'Baudrate = 188, which is the code
for 4800
                                 'Baudrate =  84, which is the code
for 9600

IF Pulse_width<370 OR Pulse_width>1100 THEN Main '370 to 1100 is the
valid range

PULSOUT Servo_pin, Pulse_width
GOTO Main
```

sdsinterf1_968.m
(Matlab program controlling ultrasound scanner and servo motor positioning data)

```
% Script File: sdsinterf.m
%
% Version 1.968
% December 28, 2004
%
% David Troïani
% McGill University
% Department of Chemistry
%
% This script file sets up parameters for data acquisition with an
% SDS200 oscilloscope manufactured by SoftDSP (www.softdsp.com).
% Once parameters have been set, they are saved into a text file
% "sdsparams.txt", and the script calls the external program
% "sdsgetdat2.exe" to acquire data using these parameters.
%
% The external program saves the acquired data into a text file
% "sdsdat.txt", which is then loaded and processed by this script.
%
% If the text file "sdsparams.txt" already exists from previous
% data acquisitions, these parameters will be automatically loaded
% when starting the script.
%
% This program also controls the position of a HiTEC HS-5735MG
% digital quarter scale standard servo motor, which holds an
% ultrasound transducer. The transducer feeds data to the
% oscilloscope, which relays the data to MatLab.
% Note: A properly programmed BASIC Stamp II is required for this
function

% Version History:
% 0.1a - Incomplete program
% 0.2a - Rewrote parameter saving to use the SDS200 number codes
instead of the text codes (i.e. 3 instead of "BUFFER_SIZE_10K")
% 0.3a - Added support for selection the acquisition channel
% 0.4a - Calculated the sampling frequencies **sdsgetdat2.exe now
works properly***
% 0.8b - First functionning program, the motor scans and the
oscilloscope takes data
% 0.9b - Fixed many bugs (motor not initializing, missed position
pulses, serial conflicts)
% 0.92 - First stable version of scanning motor and data acquisition
% 0.95 - Added support for smoothing, mean centering the data,
loading data, saving data
% 1.0  - Implemented plotting functions, tested data acquisition, all
looks good so far
% 1.1  - Added.... stuff.... fixed.... things... zzzzzz
% 1.50 - TONS of stuff fixed, calculate voltage average in MATLab
script instead of C++ program (less chance of round off errors)
% 1.54 - Added support for solution temperature range when scanning
with the 'o' option
% 1.55 - Added support for temperature range plotting
% 1.56 - Made it compatible with Jonjon (i.e. added support for doing
one scan only)
% 1.57 - A few bugs fixed, some things tweaked
% 1.58 - Hajelleyluyia Praise the Lord sound added when data
acquisition finished on all modes
% 1.59 - C++ program now deals with DOUBLE integers (64-bit
precision) so it does the average calculation again
% 1.60 - Added full temperature acquisition support through the
serial port using a multimeter
% 1.61 - Removed the annoying "close all figures" commands
% 1.62 - Just for fun, increased maximum number of averages to
200,000 although that would take about 2 hours...
% 1.63 - Fixed a bug where the program would produce an error if you
decide to (V)iew data without actually having any data
% 1.64 - Added a line that tells you if any data has been loaded into
memory and whether or not the data has been saved
% 1.65 - Added an abort option (CTRL-C or CTRL-BREAK) for the
temperature range scans, plus a few bugs fixed
% 1.68 - Changed the temperature cut off for option 'o', it is now
set to 28 degrees celcius
```

```
% 1.69 - Added support for changing the voltage range on individual
channels (trig chan should be always set to 5V though)
% 1.70 - Added support for turning off the "Data Acquisition
Finished" Hallelujah sound
% 1.71 - Added lots more comments in the code
% 1.72 - Added the MHz equivalents to the sampling rate menu
% 1.73 - Fixa da goud spelink mestaks!
% 1.75 - Added support for changing the delay time
% 1.77 - The plots on option 'v' now display the .mat filename as the
title (and replaces '_' with spaces in the title if any)
% 1.79 - New feature, data stripping, to remove parts of the data you
don't need (only for temperature dependent scans)
% 1.81 - Fixed some bugs related to the data stripping feature
% 1.83 - Crush the souls of more bugs, CRUSHED THEM MWAHAHAHA!!!
% 1.84 - Added an option to display the current temperature from the
Multi-Meter
% 1.85 - Added more comments in the code
% 1.86 - Added support to change the Channel Coupling on both
channels
% 1.88 - Fixed a bug with sample rate calculations
% 1.89 - Added a feature to name the FTs. This way the 'fftarray'
variables won't overwrite themselves when loading multiple files
% 1.90 - Added code that saves the raw FTs as well
% 1.91 - Updated the Servo Motor scanning code to v1.91
% 1.92 - Fixed a longstanding fft frequency axis bug
% 1.93 - Fixed a longstanding timestep calculation bug
% 1.94 - GRAAAH! Ok, for some reason, when ETS Mode is ON, the SDS200
takes 10,000 points, otherwise it's 5,000, corrected for that
% 1.95 - The first 100 points when ETSMode is ON are garbage (all are
same negative value), so corrected for that as well
% 1.96 - One minor bug that only saved the last scan for option 'm'
is now fixed
% 1.966 - Changed the FFT calculations so it does abs(FFT.^2) and not
abs(FFT).^2
% 1.968 - Fixed frequency axis again, so that max FT freq is the
Nyquist freq (i.e. sample freq / 2)

% POSSIBLE FUTURE IMPROVEMENTS:
% - Use fancy Windows menus
% - Pass the parameters to the C++ program by command-line instead of
through a text file
% - Return the data from C++ to MATLab by passing printing formatted
text to the prompt instead of a text file
% - Eventually use fancy DLL MATLab compiler stuff that Francis told
me about to pass the data as an array

clear;


load handel; % Loads the "Hallelujah" sound clip that comes with
MATLab (for when your data acquistion is complete :)
program_loop = 1; % This makes the program loop until the user
decides to quit (program_loop = 0)

try % Try to open "sdsparams.txt" if it exists
    fid = fopen('sdsparams.txt','rt'); % Open the file to 'r'ead
't'ext.

    bufferSize = fgetl(fid); % The fgetl statement reads a complete
line as a string in a text file
```

```
    bufferSize = str2num(bufferSize); % str2num converts strings to
numbers

    sampleRate = fgetl(fid);
    sampleRate = str2num(sampleRate);

    ETSMode = fgetl(fid);
    ETSMode = str2num(ETSMode);

    numAverage = fgetl(fid);
    numAverage = str2num(numAverage);

    voltRange1 = fgetl(fid);
    voltRange1 = str2num(voltRange1);

    voltRange2 = fgetl(fid);
    voltRange2 = str2num(voltRange2);

    acqChan = fgetl(fid);
    acqChan = str2num(acqChan);

    trigChan = fgetl(fid);
    trigChan = str2num(trigChan);

    delayTime = fgetl(fid);
    delayTime = str2num(delayTime);

    meanCenter = fgetl(fid);

    smoothData = fgetl(fid);

    smoothWindow = fgetl(fid);
    smoothWindow = str2num(smoothWindow);

    timeStep = fgetl(fid);
    timeStep = str2num(timeStep);

    Ch1Coup = fgetl(fid);
    Ch1Coup = str2num(Ch1Coup);

    Ch2Coup = fgetl(fid);
    Ch2Coup = str2num(Ch2Coup);

    fclose(fid); % Closes the file

catch
    % Initialize default parameters if no "sdsparams.txt" file is
found
    bufferSize = 3;
    sampleRate = 25;
    ETSMode = 0;
    numAverage = 1;
    voltRange1 = 7;
    voltRange2 = 9;
    acqChan = 1;
    trigChan = 2;
    delayTime = 0;
    meanCenter = 'Yes';
    smoothData = 'Yes';
    smoothWindow = 51;
    timeStep = ((10*400*10^-3/5000)/10^-3); % Calculates the "time
between samples" for the oscilloscope according to a formula
                                             % in the SDK
documentation (page 16)
    Ch1Coup = 0;
    Ch2Coup = 0;

end; % End of try/catch block

% Initializes the data variables. This is required since this program
uses loops that will crash if they cannot find the variables
newfftname = [];
filename = [];
data_volt = [];
data_time = [];
pulse_position = [];
temperatureRange = [];
fftarray = [];
fftarray_raw = [];
fftfreq = [];
dataSaved = 0; % This is to warn the user that they should save their
data once data acquistion is complete
PraiseTheLord = 'No'; % If this is Yes, then a Hallelujah sound will
play when data acquisition is complete

while(program_loop) % Loop while program_loop is 1 (or "true")

    clc; % Clears the screen

    % This function converts all the numerical parameters to use
nicer names so the user can see what parameters are selected
    [bufferNice, sampleNice, ETSNice, voltNice1, voltNice2,
Ch1CoupNice, Ch2CoupNice] = sdsmakenice(bufferSize, sampleRate,
ETSMode, voltRange1, voltRange2, Ch1Coup, Ch2Coup);

    % Display fancy menu
    disp('==================================================');
    disp('SDS 200 Data Acquisition Interface');
    disp('==================================================');
    disp('Version: 1.968');
    disp('December 28, 2004');
    disp(' ');
    disp('David Troïani');
    disp('McGill University');
    disp('Department of Chemistry');
    disp('--------------------------------------------------');
    disp('Current Parameters');
    disp('--------------------------------------------------');
    disp(['1.  Buffer Size                 : ', bufferNice]);
    disp(['2.  Sampling Rate               : ', sampleNice, ' (ETS
Mode ',ETSNice,')']);
    disp(['       Time Between Data Points : ', num2str(timeStep), '
millisecond(s)']);
    disp(['       Sampling Frequency       : ', num2str(
(1/(timeStep/10^3))/10^6 ), ' MHz']);
    disp(['       Max FT Frequency         : ', num2str(
(1/(timeStep/10^3))/10^6/2 ), ' MHz']);
```

```
    disp(['3.  Number of Averages Per Scan : ',
num2str(numAverage)]);
    disp(['4.  Channel 1 Voltage/Coupling  : ', voltNice1, ' - ',
Ch1CoupNice, ' Coupling']);
    disp(['5.  Channel 2 Voltage/Coupling  : ', voltNice2, ' - ',
Ch2CoupNice, ' Coupling']);
    disp(['6.  Acquisition Channel         : Channel ',
num2str(acqChan)]);
    disp(['7.  Trigger Source Channel      : Channel ',
num2str(trigChan)]);
    disp(['8.  Delay Time                  : ', num2str(delayTime), '
second(s)']);
    disp(' ');
    disp(['9.  Mean Center the data?       : ', meanCenter]);
    disp(['10. Smooth the data?            : ', smoothData]);
    disp(['11. Boxcar smoothing window     : ',
num2str(smoothWindow)]);
    disp(['12. Play a sound when finished? : ', PraiseTheLord]);
    disp(' ');
    disp('J.  Acquire a single data scan at a specific temperature
and plot the results');
    disp('O.  Acquire data scans over a range of temperatures');
    disp('T.  Display current temperature');
    disp('M.  Acquire data scans over a range of transducer positions
using a servo motor and a Basic STAMP 2');
    disp('V.  View and process the data');
    disp('E.  Strip out a section of the data (only for temperature
range data scans)');
    disp('S.  Save data');
    disp('L.  Load data');
    disp('D.  Reload default parameters');
    disp('C.  Close all figures');
    disp('Q.  Quit to the command prompt');
    disp('--------------------------------------------------');
    if (isempty(data_volt)), disp('[ ] - No data loaded into
memory'); % i.e. "If data_volt is empty, show this"...
    else, disp('[X] - Data has been loaded into memory'); end; %...
"otherwise, show this"

    % The next line tells the user that the data hasn't been saved
yet
    % It only gets printed if data_volt is not empty (i.e. there IS
data) and dataSaved is equal to 0
    if (~isempty(data_volt) & (dataSaved == 0)), disp('*** Your data
has not been saved yet ***'); end;

    disp('--------------------------------------------------');
    mainmenu_ch = input('Enter your selection... ','s');
    switch(lower(mainmenu_ch)) % Goes to a certain "case" depending
on what the user entered
        % lower(mainmenu_ch) simply converts the user's selection to
lower case so that CAPITALS and small letters are treated the same

    case '1' % Change buffer size
        clc;
        disp('===================================================');
        disp('Buffer Size');
        disp('===================================================');
        disp('A Buffer Size of  10,000 bytes allows:');
        disp('         100  Hz to   100 MHz Sample Rates (ETS Mode
OFF)');
```

```
        disp('        250 MHz to 5,000 MHz Sample Rates (ETS Mode
ON)');
        disp(' ');
        disp('A Buffer Size of 500,000 bytes allows:');
        disp('        2.5 kHz to     1 MHz Sample Rates (ETS Mode
OFF)');
        disp('---------------------------------------------------');
        disp('1. 10,000 byte buffer');
        disp('2. 500,000 byte buffer - SDS200A Only');
        disp('* Note: The 500,000 byte buffer size has not been
implemented due to a lack of an SDS200A unit');
        disp('---------------------------------------------------');
        bufferSize = input('Enter your selection... ');
        switch(bufferSize)
        case 1
            bufferSize = 3;
        case 2
            bufferSize = 7;
        otherwise
            bufferSize = 3;
            disp('Invalid command. Press any key to return to the
main menu...');
            pause;
        end; % End of switch(bufferSize) block

    case '2' % Change sample rate
        clc;
        disp('===================================================');
        disp('Sample Rate');
        disp('===================================================');
        disp('1.  200 ns/division - (10K Buffer, ETS Mode ON)');
        disp('2.  400 ns/division - (10K Buffer, ETS Mode ON)');
        disp('3.    1 us/division - (10K Buffer, ETS Mode ON)');
        disp('4.    2 us/division - (10K Buffer, ETS Mode ON)');
        disp('5.    4 us/division - (10K Buffer, ETS Mode ON)');
        disp('6.   10 us/division - (10K Buffer, ETS Mode OFF)');
        disp('7.   20 us/division - (10K Buffer, ETS Mode OFF)');
        disp('8.   40 us/division - (10K Buffer, ETS Mode OFF)');
        disp('9.  100 us/division - (10K Buffer, ETS Mode OFF)');
        disp('10. 200 us/division - (10K Buffer, ETS Mode OFF)');
        disp('11. 400 us/division - (10K Buffer, ETS Mode OFF)');
        disp('12.   1 ms/division - (10K or 500K Buffer, ETS Mode
OFF)');
        disp('13.   2 ms/division - (10K or 500K Buffer, ETS Mode
OFF)');
        disp('14.   4 ms/division - (10K or 500K Buffer, ETS Mode
OFF)');
        disp('15.  10 ms/division - (10K or 500K Buffer, ETS Mode
OFF)');
        disp('16.  20 ms/division - (10K or 500K Buffer, ETS Mode
OFF)');
        disp('17.  40 ms/division - (10K or 500K Buffer, ETS Mode
OFF)');
        disp('18. 100 ms/division - (10K or 500K Buffer, ETS Mode
OFF)');
        disp('19. 200 ms/division - (10K or 500K Buffer, ETS Mode
OFF)');
        disp('20. 400 ms/division - (10K or 500K Buffer, ETS Mode
OFF)');
        disp('21.   1  s/division - (10K Buffer, ETS Mode OFF)');
        disp('22.   2  s/division - (10K Buffer, ETS Mode OFF)');
        disp('23.   4  s/division - (10K Buffer, ETS Mode OFF)');
        disp('24.  10  s/division - (10K Buffer, ETS Mode OFF)');
        disp('---------------------------------------------------');
        sampleRate = input('Enter your selection... ');
        switch(sampleRate)
        case 1
            sampleRate = 6;
            ETSMode = 1;
            timeStep = ((10*200*10^-9/9900)/10^-3); % Calculates the
time between data points in msec's
            % The formula for this calculation can be found on page
16 of the SoftScope SDK User's Guide
        case 2
            sampleRate = 7;
            ETSMode = 1;
            timeStep = ((10*400*10^-9/9900)/10^-3);
        case 3
            sampleRate = 8;
            ETSMode = 1;
            timeStep = ((10*1*10^-6/9900)/10^-3);
        case 4
            sampleRate = 9;
            ETSMode = 1;
            timeStep = ((10*2*10^-6/9900)/10^-3);
        case 5
            sampleRate = 10;
            ETSMode = 1;
            timeStep = ((10*4*10^-6/9900)/10^-3);
        case 6
            sampleRate = 11;
            ETSMode = 0;
            timeStep = ((10*10*10^-6/5000)/10^-3);
        case 7
            sampleRate = 12;
            ETSMode = 0;
            timeStep = ((10*20*10^-6/5000)/10^-3);
        case 8
            sampleRate = 13;
            ETSMode = 0;
            timeStep = ((10*40*10^-6/5000)/10^-3);
        case 9
            sampleRate = 14;
            ETSMode = 0;
            timeStep = ((10*100*10^-6/5000)/10^-3);
        case 10
            sampleRate = 15;
            ETSMode = 0;
            timeStep = ((10*200*10^-6/5000)/10^-3);
        case 11
            sampleRate = 16;
            ETSMode = 0;
            timeStep = ((10*400*10^-6/5000)/10^-3);
        case 12
            sampleRate = 17;
            ETSMode = 0;
            timeStep = ((10*1*10^-3/5000)/10^-3);
        case 13
            sampleRate = 18;
            ETSMode = 0;
            timeStep = ((10*2*10^-3/5000)/10^-3);
        case 14
```

```
            sampleRate = 19;
            ETSMode = 0;
            timeStep = ((10*4*10^-3/5000)/10^-3);
        case 15
            sampleRate = 20;
            ETSMode = 0;
            timeStep = ((10*10*10^-3/5000)/10^-3);
        case 16
            sampleRate = 21;
            ETSMode = 0;
            timeStep = ((10*20*10^-3/5000)/10^-3);
        case 17
            sampleRate = 22;
            ETSMode = 0;
            timeStep = ((10*40*10^-3/5000)/10^-3);
        case 18
            sampleRate = 23;
            ETSMode = 0;
            timeStep = ((10*100*10^-3/5000)/10^-3);
        case 19
            sampleRate = 24;
            ETSMode = 0;
            timeStep = ((10*200*10^-3/5000)/10^-3);
        case 20
            sampleRate = 25;
            ETSMode = 0;
            timeStep = ((10*400*10^-3/5000)/10^-3);
        case 21
            sampleRate = 26;
            ETSMode = 0;
            timeStep = ((10*1/5000)/10^-3);
        case 22
            sampleRate = 27;
            ETSMode = 0;
            timeStep = ((10*2/5000)/10^-3);
        case 23
            sampleRate = 28;
            ETSMode = 0;
            timeStep = ((10*4/5000)/10^-3);
        case 24
            sampleRate = 29;
            ETSMode = 0;
            timeStep = ((10*10/5000)/10^-3);
        otherwise
            sampleRate = 25;
            ETSMode = 0;
            timeStep = ((10*400*10^-3/5000)/10^-3);
            disp('Invalid command. Press any key to return to the
main menu...');
            pause;
        end; % End of switch(sampleRate) block

    case '3' % Change number of averages
        clc;
        disp('====================================================');
        disp('Number of Averages');
        disp('====================================================');
        disp('Valid number of averages are 1 to 200,000.');
        disp('----------------------------------------------------');
        numAverage = input('Enter the number of averages... ');
        if numAverage > 200000, numAverage = 200000; end; % If the
user enters a number over 200,000, change it to 200,000
        if numAverage < 1, numAverage = 1; end; % Same thing if the
user enters a number lower than 1 (which would be stupid, right? ;)

    case '4' % Change channel 1 voltage range
        clc;
        disp('=====================================================');
        disp('Channel 1 Voltage Range');
        disp('=====================================================');
        disp('1.   10 mV');
        disp('2.   20 mV');
        disp('3.   50 mV');
        disp('4.  100 mV');
        disp('5.  200 mV');
        disp('6.  500 mV');
        disp('7.    1  V');
        disp('8.    2  V');
        disp('9.    5  V');
        disp('10.  10  V');
        disp('-----------------------------------------------------');
        voltRange1 = input('Enter your selection... ');
        switch(voltRange1)
        case 1
            voltRange1 = 1;
        case 2
            voltRange1 = 2;
        case 3
            voltRange1 = 3;
        case 4
            voltRange1 = 4;
        case 5
            voltRange1 = 5;
        case 6
            voltRange1 = 6;
        case 7
            voltRange1 = 7;
        case 8
            voltRange1 = 8;
        case 9
            voltRange1 = 9;
        case 10
            voltRange1 = 10;
        otherwise
            voltRange1 = 9;
            disp('Invalid command. Press any key to continue...');
            pause;
        end; % End of switch(voltRange1) block
        disp(' ');
        Ch1Coup = input('Enter ''A'' for AC Coupling or ''D'' for DC
Coupling... ','s');
        Ch1Coup = lower(Ch1Coup);
        if strcmpi(Ch1Coup, 'd'), Ch1Coup = 1; else Ch1Coup = 0; end;

    case '5' % Change channel 2 voltage range
        clc;
        disp('=====================================================');
        disp('Channel 2 Voltage Range');
        disp('=====================================================');
        disp('1.   10 mV');
        disp('2.   20 mV');
        disp('3.   50 mV');
        disp('4.  100 mV');
        disp('5.  200 mV');
        disp('6.  500 mV');
        disp('7.    1  V');
        disp('8.    2  V');
        disp('9.    5  V');
        disp('10.  10  V');
        disp('---------------------------------------------------');
        voltRange2 = input('Enter your selection... ');
        switch(voltRange2)
        case 1
            voltRange2 = 1;
        case 2
            voltRange2 = 2;
        case 3
            voltRange2 = 3;
        case 4
            voltRange2 = 4;
        case 5
            voltRange2 = 5;
        case 6
            voltRange2 = 6;
        case 7
            voltRange2 = 7;
        case 8
            voltRange2 = 8;
        case 9
            voltRange2 = 9;
        case 10
            voltRange2 = 10;
        otherwise
            voltRange2 = 9;
            disp('Invalid command. Press any key to return to the 
main menu...');
            pause;
        end; % End of switch(voltRange2) block
        disp(' ');
        Ch2Coup = input('Enter ''A'' for AC Coupling or ''D'' for DC 
Coupling... ','s');
        Ch2Coup = lower(Ch2Coup);
        if strcmpi(Ch2Coup, 'd'), Ch2Coup = 1; else Ch2Coup = 0; end;

    case '6' % Change acquisition channel between channel 1 and 
channel 2
        if acqChan == 1, acqChan = 2; % i.e. if acqChan 1 is 
selected, change it to 2, otherwise make it 1
        else, acqChan = 1;
        end;

    case '7' % Change trigger channel between channel 1 and channel 2
        if trigChan == 1, trigChan = 2; % same as above
        else, trigChan = 1;
        end;

    case '8' % Change Delay Time
        clc;
        disp('===================================================');
        disp('Delay Time');
        disp('===================================================');
        disp(['Valid delay times are ', num2str(-5 *
((timeStep*(5000*10^-3))/10)/2 ),' to ', num2str(5 *
((timeStep*(5000*10^-3))/10)/2 ),' seconds (0 means no delay).']);
        disp('* Note: The delay time limits are five times the sample
rate "time/division" value.');
        disp('--------------------------------------------------');
        delayTime = input('Enter the delay time in seconds... ');
        if delayTime > ( 5 * ((timeStep*(5000*10^-3))/10)/2 ),
delayTime = ( 5 * ((timeStep*(5000*10^-3))/10)/2 ); end;
        if delayTime < (-5 * ((timeStep*(5000*10^-3))/10)/2 ),
delayTime = (-5 * ((timeStep*(5000*10^-3))/10)/2 ); end;

        % ((timeStep*(10000*10^-3))/10)/2
        % This formula calculates the time/division from timeStep
(basically it's the "Sample Rate timeStep" formula in reverse
including the factor of 2 correction, hence the /2 at the end)

    case '9' % Mean center the data?
        if strcmpi(meanCenter, 'Yes'), meanCenter = 'No'; % If Yes
was previously selected, change it to No
        else, meanCenter = 'Yes'; % Otherwise, make it Yes
        end;

    case '10' % Smooth the data?
        if strcmpi(smoothData, 'Yes'), smoothData = 'No'; % same as
above
        else, smoothData = 'Yes';
        end;

    case '11' % Smooth by how much?
        clc;
        disp('==================================================');
        disp('Boxcar Smoothing Window');
        disp('==================================================');
        disp('Valid numbers are odd numbers between 1 and 1,001.');
        disp('--------------------------------------------------');
        smoothWindow = input('Enter the boxcar smoothing window...
');
        if smoothWindow > 1001, smoothWindow = 1001; end;
        if smoothWindow < 1, smoothWindow = 1; end;

        % The next line checks that the number the user entered is
odd. If not, it adds 1 to the number
        if mod(smoothWindow, 2) == 0, smoothWindow = smoothWindow +
1; end;

    case '12' % Turns the "acqusition complete" sound on/off
        if strcmpi(PraiseTheLord, 'Yes'), PraiseTheLord = 'No';
        else, PraiseTheLord = 'Yes';
        end;

    case 'j' % Acquires 1 data scan and displays the results
        try % The "try" is here so that if something goes wrong,
Matlab goes to the "catch" further down instead of crashing
            fid = fopen('sdsparams.txt','wt'); % Save the parameters
to the sdsparams.txt file
            fprintf(fid, '%s\n', num2str(bufferSize));
            fprintf(fid, '%s\n', num2str(sampleRate));
            fprintf(fid, '%s\n', num2str(ETSMode));
            fprintf(fid, '%s\n', num2str(numAverage));
            fprintf(fid, '%s\n', num2str(voltRange1));
            fprintf(fid, '%s\n', num2str(voltRange2));
            fprintf(fid, '%s\n', num2str(acqChan));
            fprintf(fid, '%s\n', num2str(trigChan));
            fprintf(fid, '%s\n', num2str(delayTime));
            fprintf(fid, '%s\n', meanCenter);
            fprintf(fid, '%s\n', smoothData);
            fprintf(fid, '%s\n', num2str(smoothWindow));
            fprintf(fid, '%s\n', num2str(timeStep));
            fprintf(fid, '%s\n', num2str(Ch1Coup));
            fprintf(fid, '%s\n', num2str(Ch2Coup));
            fclose(fid);

            % Resets the variables in case they have something in
them
            newfftname = [];
            fftarray = [];
            fftarray_raw = [];
            fftfreq = [];
            data_volt = [];
            data_time = [];
            pulse_position = [];
            temperatureRange = [];

            % Get the data from the oscilloscope using the external
program (which puts the data into sdsdat.txt)
            !sdsgetdat2.exe

            pause(0.5);

            if(ETSMode==1), data_volt = dlmread('sdsdat.txt', ',',
[101 0 10000 0]); % Read the data from the sdsdat.txt text file
            else, data_volt = dlmread('sdsdat.txt', ',', [1 0 5000
0]);
            end;

            if strcmpi(PraiseTheLord, 'Yes'),
                sound(y,Fs); % Praise the Lord, plays the
Hallegjiuhah sound once the data acqusition is done
            end;

            disp('Successfully acquired the data.');
            %disp('Press any key to continue...');
            %pause;

        catch % If anything went wrong in the "try" block above...
            disp('An error occured during the acquisition process.');
            disp('Press any key to return to the command prompt...');
            pause;
            return; % Exits the program, because if MatLab tries to
do an FFT on incomplete data, bad things happen
        end % End of try/catch block

        fftarray = [];
        fftarray_raw = [];

        fftarray_raw = fft(data_volt(:,1));

        fftarray = abs( fftarray_raw.^2 ); % Calculate FT of the scan
        fftarray = fftarray(1:length(fftarray)/2); % The second half
of the FT is a mirror image of the first half, so we remove it

        if strcmpi(smoothData, 'Yes'), % If the data said 'Yes' to
smoothing the data
            fftarray = smooth(fftarray, smoothWindow); % This
smoooooooths out the data
        end;

        if strcmpi(meanCenter, 'Yes'),
            fftarray = fftarray'; % Transposes the data (required for
the Mean Centering 'mncn' function for some reason...
            [fftarray,mx] = mncn(fftarray); % Mean Center the data
            fftarray = fftarray'; % Re-transpose the data
        end;

        % **************************** FFT FREQUENCY BUG FIX
*****************************
        fftfreq = 0:  ((1/(timeStep/10^3))/10^6) /2
/(length(fftarray)-1): (1/(timeStep/10^3))/10^6/2;
        fftarray(1) = []; % The first data point of the FT is the sum
of all points, we don't need that so we remove it.
        fftfreq(1) = [];
        %
******************************************************************
*************

        % This is the old incorrect method
        %fftfreq = ( ( (1:length(fftarray)) / length(fftarray) ).^0.5
)'; % Build freq. axis according to MatLab Sunspot FFT Demo
        %fftfreq = fftfreq ./ ( timeStep ./ 10^3 ); % This converts
the frequency from cycles/ms to cycles/second (Hz, s^-1)
        %fftfreq = fftfreq ./ 10^6; % This converts from Hz to MHz

        Figure; % Brings up a new empty plot
        plot(fftfreq, fftarray); % Displays FT on a data plot
        xlabel('Frequency (MHz)', 'FontSize', 8);
        ylabel('Amplitude', 'FontSize', 8);
        title('Frequency Domain Data');
        dataSaved = 0; % Since we acquired new data, it's time to
tell the user that this data has not been saved yet
        newfftname = input('Enter a unique name for this FFT data: ',
's');
        if (isempty(newfftname)), newfftname = 'dummy'; end;
        assignin('base',newfftname,fftarray);

        newfftname_raw = ['raw_', newfftname];
        assignin('base',newfftname_raw,fftarray_raw);

        % close all;

    case 'o' % Take scans while varying the temperature

        % The following line asks the user how long to wait between
scans
        tPeriod = input('Enter the time in minutes to pause between
each scan (max 10 minutes)... ');
        if tPeriod > 10, tPeriod = 10; end;
        if tPeriod < 0, tPeriod = 0; end;
        tPeriod = tPeriod * 60; % Convert to seconds

        fftarray = [];
        fftfreq = [];
        tInterrupt_chr = [];
        data_volt = [];
        data_time = [];
        pulse_position = [];
        temperatureRange = [];
        current_temp = [];

        while isempty(tInterrupt_chr), % "While tInterrupt_chr has 
nothing in it, do this loop..."

            try
                fid = fopen('sdsparams.txt','wt'); % Save the 
parameters to the sdsparams.txt file
                fprintf(fid, '%s\n', num2str(bufferSize));
                fprintf(fid, '%s\n', num2str(sampleRate));
                fprintf(fid, '%s\n', num2str(ETSMode));
                fprintf(fid, '%s\n', num2str(numAverage));
                fprintf(fid, '%s\n', num2str(voltRange1));
                fprintf(fid, '%s\n', num2str(voltRange2));
                fprintf(fid, '%s\n', num2str(acqChan));
                fprintf(fid, '%s\n', num2str(trigChan));
                fprintf(fid, '%s\n', num2str(delayTime));
                fprintf(fid, '%s\n', meanCenter);
                fprintf(fid, '%s\n', smoothData);
                fprintf(fid, '%s\n', num2str(smoothWindow));
                fprintf(fid, '%s\n', num2str(timeStep));
                fprintf(fid, '%s\n', num2str(Ch1Coup));
                fprintf(fid, '%s\n', num2str(Ch2Coup));
                fclose(fid);

                disp('*** Press CTRL-C or CTRL-BREAK at any time to 
abort the scans ***'); % If the user does this, the program goes to 
the "catch" block (since there's a "try" above)

                current_temp = capture_temperature2; % Acquire the 
temperature of the sample cell from the Multi-Meter

                while(current_temp>28), % While the temperature is 
over 28 degrees Celcius
                    disp(' ');

                    disp('Acquiring temperature data...');
                    current_temp = capture_temperature2;

                    disp('Acquiring oscilloscope data...');
                    !sdsgetdat2.exe
                    pause(0.5);

                    if(ETSMode==1), data_volt = dlmread('sdsdat.txt', 
',', [101 0 10000 0]); % Read the data from the sdsdat.txt text file
                    else, data_volt = dlmread('sdsdat.txt', ',', [1 0 
5000 0]);
                    end;
                    [temperatureRange]=[temperatureRange,
current_temp]; % Adds the acquired temperature to the
temperatureRange array
                    disp(['Scan taken at ',
num2str(temperatureRange(length(temperatureRange))),' degrees
Celcius.']);
                    disp(['Pausing for ', num2str(tPeriod/60), '
minute(s)...']);
                    tic % Starts the MatLab cronometer
                    while(toc<tPeriod), % While the cronometer time
is less than the user entered time, pause for 1/2 a second
                        pause(0.5);
                    end;
                end;

                if strcmpi(PraiseTheLord, 'Yes'),
                    sound(y,Fs); % Praise the Lord
                end;

                tInterrupt_chr = 'r';
                disp(' ');
                disp('Successfully acquired the data.');
                disp([num2str(length(temperatureRange)), ' scans
performed.']);
                disp('Press any key to return to the main menu...');
                pause;

            catch % So if the user pressed CTRL-C or CTRL-BREAK, the
program asks if the user wants to stop or continue
                disp(' ');
                disp('An error occured or the acquisition process was
interrupted by the user.');
                tInterrupt_chr = input('Press ''R'' to return to the
main menu or Enter to continue the scans... ','s');
                if ~strcmpi(tInterrupt_chr, 'r'), tInterrupt_chr =
[]; end; % If the user doesn't press "r", tInterrupt_chr stays empty,
so the while loop above continues
            end % End of try/catch block

        end % end of while isempty(tInterrupt_chr) block
        dataSaved = 0; % Since data was just acquired, time to tell
the user to save his/her data

    case 't'
        disp('Acquiring temperature data...');
        disp([num2str(capture_temperature2), ' degrees Celsius is the
current sample cell temperature.']);
        disp(' ');
        disp('Press any key to return to the main menu...');
        pause;

    case 'm' % Take scans while changing the position of the servo
motor
        try
            fid = fopen('sdsparams.txt','wt'); % Save the parameters
to the sdsparams.txt file
            fprintf(fid, '%s\n', num2str(bufferSize));
            fprintf(fid, '%s\n', num2str(sampleRate));
            fprintf(fid, '%s\n', num2str(ETSMode));
            fprintf(fid, '%s\n', num2str(numAverage));
```

```
         fprintf(fid, '%s\n', num2str(voltRange1));
         fprintf(fid, '%s\n', num2str(voltRange2));
         fprintf(fid, '%s\n', num2str(acqChan));
         fprintf(fid, '%s\n', num2str(trigChan));
         fprintf(fid, '%s\n', num2str(delayTime));
         fprintf(fid, '%s\n', meanCenter);
         fprintf(fid, '%s\n', smoothData);
         fprintf(fid, '%s\n', num2str(smoothWindow));
         fprintf(fid, '%s\n', num2str(timeStep));
         fprintf(fid, '%s\n', num2str(Ch1Coup));
         fprintf(fid, '%s\n', num2str(Ch2Coup));
         fclose(fid);

         % Resets the variables in case they have something in
them
         newfftname = [];
         fftarray = [];
         fftarray_raw = [];
         fftfreq = [];
         data_volt = [];
         data_time = [];
         pulse_position = [];
         temperatureRange = [];

         disp('Initializing serial port...');
         ser_obj=serial('COM1','baudrate',9600);
         ser_obj.terminator = 'CR';
         fopen(ser_obj);

         pause(1); % waits until the port is initialized
         disp('Serial port initialized.');
         disp('Setting Servo Motor to initial position...');
         fprintf(ser_obj,'%d\n',625); % send initial position
pulse width to the servomotor
         pause(1); % waits until the motor is in the initial
position
         disp('Servo Motor in initial position.');

         pulse_position=(625:5:900)'; % makes an array of
positions through which the motor will go

         for
pulse_width=pulse_position(1):5:pulse_position(length(pulse_position)
), % a FOR loop that says pulse_width will go through each pulse
position

            disp(['Position ', num2str(pulse_width), ' out of ',
num2str( pulse_position(length(pulse_position)) ), '.']); % shows
progress

            fprintf(ser_obj,'%d\n',[pulse_width]); % send
position "pulse_width" to the servomotor
            pause(0.5);
            % Get the data from the oscilloscope
            !sdsgetdat2.exe
            pause(0.5);
            if(ETSMode==1), data_volt = [data_volt
dlmread('sdsdat.txt', ',', [101 0 10000 0])]; % Read the data from
the sdsdat.txt text file
```

```
                else, data_volt = [data_volt dlmread('sdsdat.txt',
',', [1 0 5000 0])];
                end;
            end;

            disp(['Total number of scans: ',
num2str(length(pulse_position))]);
            disp('Resetting Servo Motor position...');
            pause(1); % waits until the last scan is complete
            fprintf(ser_obj,'%d\n',625); % send initial position
pulse width to the servomotor
            disp('Servo Motor position reset.');
            disp('Closing serial port...');
            pause(1); % waits until the motor is in the initial
position

            fclose(ser_obj);
            disp('Serial port closed.');
            disp('Restoring system serial port control...');
            pause(1); % waits until the port is closed
            freeserial; % frees up all serial ports so that other
programs may use them
            disp('Serial port control restored.');

            if(ETSMode==1), data_time = ((linspace(0, 9900*timeStep,
9900)))'; % creates the time data (in msec's) from the sampling rate
            else, data_time = ((linspace(0, 5000*timeStep, 5000)))';
% creates the time data (in msec's) from the sampling rate
            end;

            if strcmpi(PraiseTheLord, 'Yes'),
                sound(y,Fs); % Praise the Lord
            end;

            for x = 1:size(data_volt, 2), % Take each column as being
one scan

                fftarray_raw = [fftarray_raw fft(data_volt(:,x))];

                fftamp = abs(fft( data_volt(:,x) ).^2); % Calculate
FT of each scan
                fftamp = fftamp(1:length(fftamp)/2);

                if strcmpi(smoothData, 'Yes'), % If the data said
'Yes' to smoothing the data
                    fftamp = smooth(fftamp, smoothWindow); % This
smoooooooths out the data
                end;

                fftarray = [fftarray fftamp];
            end % end of for x = 1:size... block

            if strcmpi(meanCenter, 'Yes'),
                fftarray = fftarray'; % Transposes the data
                [fftarray,mx] = mncn(fftarray); % Mean Center
subtraction
                fftarray = fftarray';
            end;
            % ***************************** FFT FREQUENCY BUG FIX
******************************
            fftfreq = 0:  ((1/(timeStep/10^3))/10^6) /2
/(length(fftarray)-1): (1/(timeStep/10^3))/10^6/2;
            fftarray(1, :) = []; % The first data point of the FT is
the sum of all points, we don't need that so we remove it.
            fftfreq(1) = [];
            %
**********************************************************************
*************

            % This is the old incorrect method
            %fftfreq = ( ( (1:length(fftamp)) / length(fftamp) ).^0.5
)'; % Build freq. axis according to MatLab Sunspot FFT Demo
            %fftfreq = fftfreq ./ ( timeStep ./ 10^3 ); % This
converts the frequency from cycles/ms to cycles/second (Hz, s^-1)
            %fftfreq = fftfreq ./ 10^6; % This converts from Hz to
MHz

            disp('Successfully acquired the data.');
            newfftname = input('Enter a unique name for this FFT
data: ', 's');
            if (isempty(newfftname)), newfftname = 'dummy'; end;
            assignin('base',newfftname,fftarray);

            newfftname_raw = ['raw_', newfftname];
            assignin('base',newfftname_raw,fftarray_raw);

        catch
            disp('An error occured during the acquisition process.');
            disp('Press any key to return to the main menu...');
            pause;
        end % End of try/catch block
        dataSaved = 0;

    case 'v' % View and process the data

        if ~(isempty(data_volt)),

            fftarray = [];

            for x = 1:size(data_volt, 2), % Take each column as being
one scan
                fftamp = abs(fft( data_volt(:,x) ).^2); % Calculate
FT of each scan
                fftamp = fftamp(1:length(fftamp)/2);

                if strcmpi(smoothData, 'Yes'), % If the data said
'Yes' to smoothing the data
                    fftamp = smooth(fftamp, smoothWindow); % This
smoooooooths out the data
                end;

                fftarray = [fftarray fftamp];
            end % end of for x = 1:size... block

            if strcmpi(meanCenter, 'Yes'),
                fftarray = fftarray'; % Transposes the data
                [fftarray,mx] = mncn(fftarray); % Mean Center
subtraction
                fftarray = fftarray';
            end;

            % **************************** FFT FREQUENCY BUG FIX
******************************
            fftfreq = 0:  ((1/(timeStep/10^3))/10^6) /2
/(length(fftarray)-1): (1/(timeStep/10^3))/10^6 /2;
            fftarray(1, :) = []; % The first data point of the FT is
the sum of all points, we don't need that so we remove it.
            fftfreq(1) = [];
            %
**********************************************************************
*************

            % The is the old incorrect method
            %fftfreq = ( ( (1:length(fftamp)) / length(fftamp) ).^0.5
)'; % Build freq. axis according to MatLab Sunspot FFT Demo
            %fftfreq = fftfreq ./ ( timeStep ./ 10^3 ); % This
converts the frequency from cycles/ms to cycles/second (Hz, s^-1)
            %fftfreq = fftfreq ./ 10^6; % This converts from Hz to
MHz

            Figure; % create a new Figure

            if isempty(data_time) & isempty(temperatureRange),
                hold on;
                for x = 1:size(data_volt, 2),
                    plot(fftfreq, fftarray(:, x)); % This loop
displays all the FFTs on the same plot
                end;
                xlabel('Frequency (MHz)', 'FontSize', 8);
                ylabel('Amplitude', 'FontSize', 8);
                title(['Frequency Domain Data - ', strrep(filename,
'_', ' ')]);
            end;

            if isempty(data_time) & ~isempty(temperatureRange),

                surf(temperatureRange, fftfreq, fftarray); % 3D
Plotting
                shading interp; % Fancy shading
                title(['Frequency Domain Data with Respect to
Temperature - ', strrep(filename, '_', ' ')]);
                ylabel('Frequency (MHz)', 'FontSize', 8);
                zlabel('Amplitude', 'FontSize', 8);
                xlabel('Temperature (^oC)', 'FontSize', 8);
                view(105,55);

                disp('Press any key to continue...');
                pause;

                Figure;
                hold on;
                for x = 1:size(data_volt, 2),
                    plot(fftfreq, fftarray(:, x)); % Displays all the
FFTs on the same plot
                end;
                xlabel('Frequency (MHz)', 'FontSize', 8);
                ylabel('Amplitude', 'FontSize', 8);
                title(['Frequency Domain Data with Respect to 
Temperature - ', strrep(filename, '_', ' ')]);

                disp('Press any key to continue...');
                pause;
                disp(' ');

                disp('Use the mouse to select the frequency peaks to 
plot on a transducer position dependant axis.');
                disp('When finished, press enter to continue...');
                peakselect = ginput; % Lets the user select frequency 
peaks from the graph
                peakselect = peakselect(:,1); % Gets rid of the y 
data, we only need the x data (frequencies)
                freqindex = []; % Initializes this array, which will 
hold the indices of the fftfreq array that are going to be plotted
                legendtext = []; % This array will hold all the 
legend information
                disp(' ');

                for x = 1:length(peakselect),
                    % Finds the frequency indices of frequencies the 
user selected and stores them into an array
                    tempvar = find(fftfreq<peakselect(x)+0.05 & 
fftfreq>peakselect(x)-0.05); % Need a 0.05 range on both sides to be 
sure to find a frequency, as if none are found MatLab complains
                    freqindex = [freqindex; tempvar(1)]; % Adds the 
index of the frequency found in the previous line to freqindex
                end;

                %close all;
                Figure;
                hold on;

                legendtext = [];
                plotfunc = ['plot(']; % This will hold the long plot 
statement that gets assembled below

                % Plot function assembly loop
                for x = 1:length(freqindex),
                    tempvar = ['fftarray(freqindex(',num2str(x),'), 
1:length(temperatureRange))']; % The specific frequency (which was 
selected above by the user) over all transducer positions

                    if strcmpi(meanCenter, 'Yes'),
                        plotfunc = [plotfunc, 'temperatureRange, 
(',tempvar,'-(min(',tempvar,'))) / (max(',tempvar,')-
min(',tempvar,'))']; % Builds the plot statement with mean centering
                    else
                        plotfunc = [plotfunc, 'temperatureRange, 
(',tempvar,' / (max(',tempvar,')))']; % Builds the plot statement 
without mean centering
                    end;

                    if x ~= length(freqindex),
                        plotfunc = [plotfunc, ',']; % Unless at the 
complete end of the plot function assembly loop, add an ',' to the 
plot statement
                    end;
                    tempvar = fftfreq(freqindex(x));
                    legendtext = [legendtext; tempvar]; % Builds the
legend text array of frequencies
                end;

                plotfunc = [plotfunc, ')']; % Puts the final bracket
on the plot statement after it is assembled

                eval(plotfunc); % Evaluates the assembled plot
statement
                legendtext = strvcat(num2str(legendtext), 'MHz');
                xlabel('Temperature (^oC)', 'FontSize', 8);
                ylabel('Amplitude', 'FontSize', 8);
                title(['Normalized Temperature Dependency of Specific
Frequency Amplitudes - ', strrep(filename, '_', ' ')]);
                legend(legendtext);
            end; % End of if isempty(data_time) block

            if isempty(temperatureRange) & ~isempty(data_time),

                surf(pulse_position, fftfreq, fftarray); % 3D
Plotting
                shading interp; % Fancy shading
                title(['Frequency Domain Data with Respect to
Transducer Position - ', strrep(filename, '_', ' ')]);
                ylabel('Frequency (MHz)', 'FontSize', 8);
                zlabel('Amplitude', 'FontSize', 8);
                xlabel('Transducer Position', 'FontSize', 8);
                view(105,55);

                disp('Press any key to continue...');
                pause;

                %close all;
                Figure;
                hold on;
                for x = 1:size(data_volt, 2),
                    plot(fftfreq, fftarray(:, x)); % Displays all the
FFTs on the same plot
                end;
                xlabel('Frequency (MHz)', 'FontSize', 8);
                ylabel('Amplitude', 'FontSize', 8);
                title(['Frequency Domain Data with Respect to
Transducer Position - ', strrep(filename, '_', ' ')]);

                disp('Press any key to continue...');
                pause;
                disp(' ');

                disp('Use the mouse to select the frequency peaks to
plot on a transducer position dependant axis.');
                disp('When finished, press enter to continue...');
                peakselect = ginput; % Lets the user select frequency
peaks from the graph
                peakselect = peakselect(:,1); % Gets rid of the y
data, we only need the x data (frequencies)
                freqindex = []; % Initializes this array, which will
hold the indices of the fftfreq array that are going to be plotted
                legendtext = []; % This array will hold all the
legend information
```

```
                disp(' ');

                for x = 1:length(peakselect),
                    % Finds the frequency indices of frequencies the
user selected and stores them into an array
                    tempvar = find(fftfreq<peakselect(x)+0.05 &
fftfreq>peakselect(x)-0.05); % Need a 0.05 range on both sides to be
sure to find a frequency, as if none are found MatLab complains
                    freqindex = [freqindex; tempvar(1)]; % Adds the
index of the frequency found in the previous line to freqindex
                end;

                %close all;
                Figure;
                hold on;

                legendtext = [];
                plotfunc = ['plot(']; % This will hold the long plot
statement that gets assembled below

                % Plot function assembly loop
                for x = 1:length(freqindex),
                    tempvar = ['fftarray(freqindex(',num2str(x),'),
1:length(pulse_position))']; % The specific frequency (which was
selected above by the user) over all transducer positions

                    if strcmpi(meanCenter, 'Yes'),
                        plotfunc = [plotfunc, 'pulse_position,
(',tempvar,'-(min(',tempvar,'))) / (max(',tempvar,')-
min(',tempvar,'))']; % Builds the plot statement with mean centering
                    else
                        plotfunc = [plotfunc, 'pulse_position,
(',tempvar,' / (max(',tempvar,')))']; % Builds the plot statement
without mean centering
                    end;

                    if x ~= length(freqindex),
                        plotfunc = [plotfunc, ',']; % Unless at the
complete end of the plot function assembly loop, add an ',' to the
plot statement
                    end;
                    tempvar = fftfreq(freqindex(x));
                    legendtext = [legendtext; tempvar]; % Builds the
legend text array of frequencies
                end;

                plotfunc = [plotfunc, ')']; % Puts the final bracket
on the plot statement after it is assembled

                eval(plotfunc); % Evaluates the assembled plot
statement
                legendtext = strvcat(num2str(legendtext), 'MHz');
                xlabel('Transducer Position', 'FontSize', 8);
                ylabel('Amplitude', 'FontSize', 8);
                title(['Normalized Position Dependency of Specific
Frequency Amplitudes - ', strrep(filename, '_', ' ')]);
                legend(legendtext);

            end; % End of if isempty(temperatureRange) block
            disp('Press any key to return to the main menu...');
            pause;
            %close all;

        else,
            disp('No data has been loaded yet.');
            disp('Press any key to return to the main menu...');
            pause;
        end; %end of if ~(isempty(data_volt)) block

    case 'e' % Strip the data
        clc;
        disp('==================================================');
        disp('Data Stripping');
        disp('==================================================');
        disp(' ');
        disp('1. Remove all data above a certain temperature');
        disp('2. Remove all data below a certain temperature');
        disp('3. Remove all data except for a certain temperature
range');
        disp(' ');
        disp('R. Return to the main menu');
        disp('--------------------------------------------------');
        datastripchr = input('Enter your selection... ','s');
        disp(' ');
        switch(lower(datastripchr))
        case '1'
            datastripchr = input('Enter the temperature in degrees
Celcius above which all data will be removed... ');
            datastripindex = find(temperatureRange>datastripchr); %
Finds all the indices with temperatures above what the user entered
            datastripindex = datastripindex(length(datastripindex));
% Takes the last index
            temperatureRange =
temperatureRange(datastripindex:length(temperatureRange)); % Cuts out
the rest of the temperatureRange array
            data_volt = data_volt(:,datastripindex:size(data_volt,
2)); % Cuts out the voltage data associated with the cut temperatures
            disp(['All data above the nearest temperature, ',
num2str(temperatureRange(1)),' degrees Celcius, was removed']);
        case '2'
            datastripchr = input('Enter the temperature in degrees
Celcius below which all data will be removed... ');
            datastripindex = find(temperatureRange<datastripchr); %
Finds all the indices with temperatures below what the user entered
            datastripindex = datastripindex(1); % Takes the last
index
            temperatureRange = temperatureRange(1:datastripindex); %
Cuts out the rest of the temperatureRange array
            data_volt = data_volt(:,1:datastripindex); % Cuts out the
voltage data associated with the cut temperatures
            disp(['All data above the nearest temperature, ',
num2str(temperatureRange(length(temperatureRange))),' degrees
Celcius, was removed']);
        case '3'
            while (size(datastripchr)~=[1 2]) % Loop while the user
DOESN'T enter a temperature range consisting of 2 numbers
                disp('Enter the temperature range in degrees Celcius
of the data you wish to keep.');
```

```
                datastripchr = input('Use the array form (square
brackets, i.e. ''[32.5 54]'', ''[43 49]'') to enter the temperature
range... ');
            end; % end of while (length(datastripchr)~=2) block

            datastripindex = find(temperatureRange>datastripchr(2));
% Like case 1
            datastripindex = datastripindex(length(datastripindex));
            temperatureRange =
temperatureRange(datastripindex:length(temperatureRange));
            data_volt = data_volt(:,datastripindex:size(data_volt,
2));

            datastripindex = find(temperatureRange<datastripchr(1));
% Like case 2
            datastripindex = datastripindex(1);
            temperatureRange = temperatureRange(1:datastripindex);
            data_volt = data_volt(:,1:datastripindex);

            disp(['All data outside of the nearest temperature range,
', num2str(temperatureRange(length(temperatureRange))),' and ',
num2str(temperatureRange(1)), ' degrees Celcius, was removed']);
        case 'r'
        otherwise
        end; % end of switch(lower(datastripchr)) block

        disp(' ');
        datastripchr = input('Would you like to save the data now
with the file name suffix ''_stripped'' (Y/N, press enter for ''Y'')?
','s');
        if ~strcmpi(datastripchr, 'n')
            save([filename '_stripped'], 'data_volt', 'data_time',
'sampleRate', 'timeStep', 'numAverage', 'voltRange1', 'voltRange2',
'bufferSize', 'ETSMode', 'acqChan', 'trigChan', 'delayTime',
'pulse_position', 'temperatureRange', 'fftarray', 'fftfreq',
'filename');
        end;

    case 's' % Save data
        clc;
        disp('==================================================');
        disp('Save Data');
        disp('==================================================');
        dir % Shows the contents of the current folder
        disp('--------------------------------------------------');
        filename = input('Enter the file name or ''R'' to return to
the main menu... ', 's');
        if ~(strcmpi(filename, 'r')), % If the user DOESN'T enter the
letter 'r'...
            try
                save(filename, 'data_volt', 'data_time',
'sampleRate', 'timeStep', 'numAverage', 'voltRange1', 'voltRange2',
'bufferSize', 'ETSMode', 'acqChan', 'trigChan', 'delayTime',
'pulse_position', 'temperatureRange', 'fftarray', 'fftfreq',
'filename', 'Ch1Coup', 'Ch2Coup', newfftname, 'newfftname',
newfftname_raw, 'newfftname_raw');
                disp('The raw data, Fourier transform data, and their
parameters were successfully saved.');
                dataSaved = 1; % Since we just saved the data, no
need to tell the user that the data needs to be saved again
```

```
                disp('Press any key to return to the main menu...');
                pause;
            catch
                disp('An error occured during the saving process.');
                disp('Press any key to return to the main menu...');
                pause;
            end % End of try/catch block
        end % End of if ~(strcmpi) block

    case 'l' % Load data
        clc;
        disp('=====================================================');
        disp('Load Data');
        disp('=====================================================');
        dir
        disp('-----------------------------------------------------');
        filename = input('Enter the file name or ''R'' to return to 
the main menu... ', 's');
        if ~(strcmpi(filename, 'r')),
            try
                %load(filename, 'data_volt', 'data_time', 
'sampleRate', 'timeStep', 'numAverage', 'voltRange1', 'voltRange2', 
'bufferSize', 'ETSMode', 'acqChan', 'trigChan', 'delayTime', 
'pulse_position', 'temperatureRange', 'fftarray', 'fftfreq', 
'filename', 'Ch1Coup', 'Ch2Coup', newfftname, 'newfftname');
                load(filename);
                disp('The raw data, Fourier transform data, and their 
parameters were successfully loaded.');
                dataSaved = 1; % We just loaded some data, so it must 
therefore be saved somewhere, so the user doesn't need to be told to 
save the data
                disp('Press any key to return to the main menu...');
                pause;
            catch
                disp('An error occured during the loading process.');
                disp('Press any key to return to the main menu...');
                pause;
            end % End of try/catch block
        end % End of if ~(strcmpi) block

    case 'd' % Reload default parameters
        bufferSize = 3;
        sampleRate = 25;
        ETSMode = 0;
        numAverage = 1;
        voltRange1 = 7;
        voltRange2 = 9;
        acqChan = 1;
        trigChan = 2;
        delayTime = 0;
        meanCenter = 'Yes';
        smoothData = 'Yes';
        smoothWindow = 51;
        timeStep = ((10*400*10^-3/5000)/10^-3);
        Ch1Coup = 0;
        Ch1Coup = 0;
        disp('The default parameters were successfully loaded.');
        disp('Press any key to return to the main menu...');
        pause;

    case 'c'
```

```
        close all;

    case 'q' % Quit the program
        program_loop = 0;
        try
            fid = fopen('sdsparams.txt','wt'); % Save the parameters
to the sdsparams.txt file
            fprintf(fid, '%s\n', num2str(bufferSize));
            fprintf(fid, '%s\n', num2str(sampleRate));
            fprintf(fid, '%s\n', num2str(ETSMode));
            fprintf(fid, '%s\n', num2str(numAverage));
            fprintf(fid, '%s\n', num2str(voltRange1));
            fprintf(fid, '%s\n', num2str(voltRange2));
            fprintf(fid, '%s\n', num2str(acqChan));
            fprintf(fid, '%s\n', num2str(trigChan));
            fprintf(fid, '%s\n', num2str(delayTime));
            fprintf(fid, '%s\n', meanCenter);
            fprintf(fid, '%s\n', smoothData);
            fprintf(fid, '%s\n', num2str(smoothWindow));
            fprintf(fid, '%s\n', num2str(timeStep));
            fprintf(fid, '%s\n', num2str(Ch1Coup));
            fprintf(fid, '%s\n', num2str(Ch2Coup));
            fclose(fid);
        catch
        end
        disp(' ');
        disp('Exiting to the command prompt...');
        disp(' ');

    otherwise
        disp('Invalid command. Press any key to return to the main
menu...');
        pause;

    end; % End of switch(lower(mainmenu_ch)) block

end; % End of while(program_loop) block

% Clears all variables except those containing oscilloscope data
clear delayTime fftamp ans current_temp ETSMode ETSNice filename
freqindex legendtext mx peakselect plotfunc tempvar x Fs acqChan
bufferNice bufferSize dataSaved fid mainmenu_ch meanCenter numAverage
program_loop sampleNice sampleRate smoothData smoothWindow timeStep
tInterrupt_chr tPeriod trigChan voltNice1 voltNice2 voltRange1
voltRange2 y PraiseTheLord datastripchr datastripindex Ch1Coup
Ch2Coup Ch1CoupNice Ch2CoupNice newfftname newfftname_raw;
```

sdsmakenice.m

```
% Function File: sdsmakenice.m
%
% Version: 1.4
% January 10, 2005
%
% David Troïani
% McGill University
% Department of Chemistry
%
% This script makes nice names for the sdsinterf.m menu out of the
numerical SDS200 codes for the parameters.
```

```
%
% Version History:
% 0.1a - Incomplete program
% 1.0  - Complete
% 1.1  - Added the Sampling Frequencies
% 1.2  - Removed Sampling Frequencies, they are now calculated in the
sdsinterf.m script
% 1.3  - Added support for Channel 2 voltage range
% 1.4  - Added support for Channel Coupling on both channels

function [bufferNice, sampleNice, ETSNice, voltNice1, voltNice2,
Ch1CoupNice, Ch2CoupNice] = sdsmakenice(bufferSize, sampleRate,
ETSMode, voltRange1, voltRange2, Ch1Coup, Ch2Coup)

switch bufferSize,
case 3
    bufferNice = '10 Kbytes';
case 7
    bufferNice = '500 Kbytes';
end % End of switch bufferSize block

switch sampleRate,
case 6
    sampleNice = '200 ns/division';
case 7
    sampleNice = '400 ns/division';
case 8
    sampleNice = '1 us/division';
case 9
    sampleNice = '2 us/division';
case 10
    sampleNice = '4 us/division';
case 11
    sampleNice = '10 us/division';
case 12
    sampleNice = '20 us/division';
case 13
    sampleNice = '40 us/division';
case 14
    sampleNice = '100 us/division';
case 15
    sampleNice = '200 us/division';
case 16
    sampleNice = '400 us/division';
case 17
    sampleNice = '1 ms/division';
case 18
    sampleNice = '2 ms/division';
case 19
    sampleNice = '4 ms/division';
case 20
    sampleNice = '10 ms/division';
case 21
    sampleNice = '20 ms/division';
case 22
    sampleNice = '40 ms/division';
case 23
    sampleNice = '100 ms/division';
case 24
    sampleNice = '200 ms/division';
case 25
```

```
    sampleNice = '400 ms/division';
case 26
    sampleNice = '1 s/division';
case 27
    sampleNice = '2 s/division';
case 28
    sampleNice = '4 s/division';
case 29
    sampleNice = '10 s/division';
end %End of switch sampleRate block

switch ETSMode,
case 0
    ETSNice = 'OFF';
case 1
    ETSNice = 'ON';
end % End of switch ETSMode block

switch voltRange1,
case 1
    voltNice1 = '10 mV';
case 2
    voltNice1 = '20 mV';
case 3
    voltNice1 = '50 mV';
case 4
    voltNice1 = '100 mV';
case 5
    voltNice1 = '200 mV';
case 6
    voltNice1 = '500 mV';
case 7
    voltNice1 = '1 V';
case 8
    voltNice1 = '2 V';
case 9
    voltNice1 = '5 V';
case 10
    voltNice1 = '10 V';
end % End of switch voltRange block

switch voltRange2,
case 1
    voltNice2 = '10 mV';
case 2
    voltNice2 = '20 mV';
case 3
    voltNice2 = '50 mV';
case 4
    voltNice2 = '100 mV';
case 5
    voltNice2 = '200 mV';
case 6
    voltNice2 = '500 mV';
case 7
    voltNice2 = '1 V';
case 8
    voltNice2 = '2 V';
case 9
    voltNice2 = '5 V';
case 10
```

```
    voltNice2 = '10 V';
end % End of switch voltRange block

switch Ch1Coup,
case 0
    Ch1CoupNice = 'AC';
case 1
    Ch1CoupNice = 'DC';
end % End of switch Ch1CoupNice

switch Ch2Coup,
case 0
    Ch2CoupNice = 'AC';
case 1
    Ch2CoupNice = 'DC';
end % End of switch Ch2CoupNice

return;
```

sdsgetdat2.cpp

```
/*
sdsgetdat2.cpp

Version: 1.5
April 21, 2005

David Troïani
McGill University
Department of Chemistry

This file reads parameters for the SoftDSP SDS 200 oscilloscope from
a text file called
"sdsparams.txt". It applies the parameters to the oscilloscope, then
acquires data. The
data is stored in another text file called "sdsdat.txt".

This program is not meant to be used on it's own, it is designed to
accompany the
SDS Acquisition Interface script in MATLab (sdsinterf1_968.m).

Version History:

0.1b - Incomplete program, crashes when trying to acquire the data
(reason unknown)
1.0  - Program fixed by SoftDSP. They mainly added missing breaks in
the switch/cases
1.1  - Added support for changing the channel 2 voltage range
1.2  - Added support for setting the delay time
1.3  - Added support for changing the Channel Coupling on both
channels
1.4  - Fixed bug that would always export CH1 data even if CH2 was
the acquisition channel
1.5  - Added Trigger Slope UP after the AUTO Trigger Mode is set
*/
```

```
#include "stdafx.h"
#include "SoftScopeAPI.h"
#include "iostream"

/* IMPORTANT NOTES:

In order to make a C++ program that works with the SoftScope SDK
files, the following must be done:
1. Make a new 'Win32 Application' thesis (not a console thesis). VC++
will create some files.
2. Use '#include "SoftScopeAPI.h"' to include the SoftScopeAPI header
file before "iostream"
3. Optional: Use '#pragma comment(lib, "SoftScope11.lib")' to include
the SoftScope library
4. Make sure that the following files are in the thesis folder, and
are included in the thesis:
      - SoftScopeAPI.h
      - SoftScope_type.h
      - SoftScope11.lib
      - SoftScope11.dll
*/

int APIENTRY WinMain(HINSTANCE hInstance,
                     HINSTANCE hPrevInstance,
                     LPSTR     lpCmdLine,
                     int       nCmdShow)
{
      char bufferReader[8] = "       "; // This is the buffer for
reading the 'sdsparams.txt' file
      int bufferSize = 0;
      int sampleRate = 0;
      int ETSMode = 0;
      int numAverage = 0;
      int voltRange1 = 0;
      int voltRange2 = 0;
      int acqChan = 0;
      int trigChan = 0;
      int Ch1Coup = 0;
      int Ch2Coup = 0;
      double delayTime = 0;
      int i = 0; // Counter
      int j = 0; // Counter
      bool dataSuccess = true; // Aborts program if this is false,
i.e. some SDS200 command fails
      PHYSICAL_DATA *ch1Data = new PHYSICAL_DATA[10000]; // Creates a
place to store the data
      PHYSICAL_DATA *ch2Data = new PHYSICAL_DATA[10000];
      PHYSICAL_DATA *ch1DataTemp = new PHYSICAL_DATA[10000]; //
Creates a temporary place to store the data
      PHYSICAL_DATA *ch2DataTemp = new PHYSICAL_DATA[10000];

      for (i = 0; i < 10000; i++) // This loop initializes the data
      {
            ch1Data[i] = 0;
            ch1DataTemp[i] = 0;
            ch2Data[i] = 0;
            ch2DataTemp[i] = 0;
      }

      FILE * sdsparams = fopen("sdsparams.txt", "r"); // Opens the
file for reading
```

```
      if (sdsparams == NULL) // If the file can't be opened...
      {
            MessageBox(NULL, "SDSPARAMS.TXT not found! (error code #
1)", NULL, MB_OK); // Error window
            exit(1); // Exits with error code 1 (in this program,
that means error reading file)
      }

    //irehyun@softdsp.com, 2005.03.15
    /* Set pointer to beginning of file: */
    fseek( sdsparams, 0L, SEEK_SET );

      fscanf(sdsparams, "%s", bufferReader); // Grabs lines of text
from "sdsparams.txt"
      bufferSize = atoi(bufferReader); // atoi converts strings to
ints

      fscanf(sdsparams, "%s", bufferReader);
      sampleRate = atoi(bufferReader);

      fscanf(sdsparams, "%s", bufferReader);
      ETSMode = atoi(bufferReader);

      fscanf(sdsparams, "%s", bufferReader);
      numAverage = atoi(bufferReader);

      fscanf(sdsparams, "%s", bufferReader);
      voltRange1 = atoi(bufferReader);

      fscanf(sdsparams, "%s", bufferReader);
      voltRange2 = atoi(bufferReader);

      fscanf(sdsparams, "%s", bufferReader);
      acqChan = atoi(bufferReader);

      fscanf(sdsparams, "%s", bufferReader);
      trigChan = atoi(bufferReader);

      fscanf(sdsparams, "%s", bufferReader);
      delayTime = atof(bufferReader);

      fscanf(sdsparams, "%s", bufferReader);
      Ch1Coup = atoi(bufferReader);

      fscanf(sdsparams, "%s", bufferReader);
      Ch2Coup = atoi(bufferReader);

      fclose(sdsparams); // close the file after reading

      dataSuccess = sdInitialize(); // Initializes the SDS200
(required)

      if (dataSuccess == false) // If the data initialization failed
      {
            MessageBox(NULL, "SDS200 not found! (error code # 3)",
NULL, MB_OK); // Error window
            exit(3); // Exits with error code 3 (in this program,
that means SDS200 not found)
      }
```

```
   sdSetChannelOnOff(_CH1, true);
   sdSetChannelOnOff(_CH2, true);
   sdSetChannelOffsetVoltage(_CH1, 0);
   sdSetChannelOffsetVoltage(_CH2, 0);
   sdSetTriggerMode(_AUTO); // Automatic Trigger Mode
  sdSetTriggerSlope(_TRIGGER_SLOPE_UP); // Trigger on Slope Up

   switch(Ch1Coup)
   {
   case 0:
         sdSetChannelCoupling(_CH1, _AC); // Sets the Channel
Coupling
    //irehyun@softdsp.com, 2005.03.15
    break;
   case 1:
         sdSetChannelCoupling(_CH1, _DC);
    //irehyun@softdsp.com, 2005.03.15
    break;
   }

   switch(Ch2Coup)
   {
   case 0:
         sdSetChannelCoupling(_CH2, _AC); // Sets the Channel
Coupling
    //irehyun@softdsp.com, 2005.03.15
    break;
   case 1:
         sdSetChannelCoupling(_CH2, _DC);
    //irehyun@softdsp.com, 2005.03.15
    break;
   }

   switch(trigChan)
   {
   case 1:
         sdSetTriggerSource(_TRIGGER_SOURCE_CH1); // Sets the
Trigger Source channel
    //irehyun@softdsp.com, 2005.03.15
    break;
   case 2:
         sdSetTriggerSource(_TRIGGER_SOURCE_CH2);
    //irehyun@softdsp.com, 2005.03.15
    break;
   }

   switch(bufferSize) // Apparently only for the SDS200A
   {
   case 3:
    sdSetBufferSize(_10K); // Sets the buffersize (only SDS200A
can use the 500K buffer)
   case 7:
    sdSetBufferSize(_500K);
   }

   switch(sampleRate)
   {
   case 6:
         sdSetTimeDiv(_200NS_DIV); // Sets the sample rate (x-
axis).
```

```
      //timeStep = (10*200*pow(10,-9)/10000)/pow(10,-6); //
Calculates the time between data points in msec's
    //irehyun@softdsp.com, 2005.03.15
    break;
   case 7:
      sdSetTimeDiv(_400NS_DIV);
      //timeStep = (10*400*pow(10,-9)/10000)/pow(10,-6);
    //irehyun@softdsp.com, 2005.03.15
    break;
  case 8:
      sdSetTimeDiv(_1US_DIV);
      //timeStep = (10*1*pow(10,-6)/10000)/pow(10,-6);
    //irehyun@softdsp.com, 2005.03.15
    break;
  case 9:
      sdSetTimeDiv(_2US_DIV);
      //timeStep = (10*2*pow(10,-6)/10000)/pow(10,-6);
    //irehyun@softdsp.com, 2005.03.15
    break;
   case 10:
      sdSetTimeDiv(_4US_DIV);
      //timeStep = (10*4*pow(10,-6)/10000)/pow(10,-6);
    //irehyun@softdsp.com, 2005.03.15
    break;
   case 11:
      sdSetTimeDiv(_10US_DIV);
      //timeStep = (10*10*pow(10,-6)/10000)/pow(10,-6);
    //irehyun@softdsp.com, 2005.03.15
    break;
   case 12:
      sdSetTimeDiv(_20US_DIV);
      //timeStep = (10*20*pow(10,-6)/10000)/pow(10,-6);
    //irehyun@softdsp.com, 2005.03.15
    break;
   case 13:
      sdSetTimeDiv(_40US_DIV);
      //timeStep = (10*40*pow(10,-6)/10000)/pow(10,-6);
    //irehyun@softdsp.com, 2005.03.15
    break;
   case 14:
      sdSetTimeDiv(_100US_DIV);
      //timeStep = (10*100*pow(10,-6)/10000)/pow(10,-6);
    //irehyun@softdsp.com, 2005.03.15
    break;
   case 15:
      sdSetTimeDiv(_200US_DIV);
      //timeStep = (10*200*pow(10,-6)/10000)/pow(10,-6);
    //irehyun@softdsp.com, 2005.03.15
    break;
   case 16:
      sdSetTimeDiv(_400US_DIV);
      //timeStep = (10*400*pow(10,-6)/10000)/pow(10,-6);
    //irehyun@softdsp.com, 2005.03.15
    break;
   case 17:
      sdSetTimeDiv(_1MS_DIV);
      //timeStep = (10*1*pow(10,-3)/10000)/pow(10,-6);
    //irehyun@softdsp.com, 2005.03.15
    break;
   case 18:
      sdSetTimeDiv(_2MS_DIV);
```

```
        //timeStep = (10*2*pow(10,-3)/10000)/pow(10,-6);
    //irehyun@softdsp.com, 2005.03.15
    break;
  case 19:
        sdSetTimeDiv(_4MS_DIV);
        //timeStep = (10*4*pow(10,-3)/10000)/pow(10,-6);
    //irehyun@softdsp.com, 2005.03.15
    break;
  case 20:
        sdSetTimeDiv(_10MS_DIV);
        //timeStep = (10*10*pow(10,-3)/10000)/pow(10,-6);
    //irehyun@softdsp.com, 2005.03.15
    break;
  case 21:
        sdSetTimeDiv(_20MS_DIV);
        //timeStep = (10*20*pow(10,-3)/10000)/pow(10,-6);
    //irehyun@softdsp.com, 2005.03.15
    break;
  case 22:
        sdSetTimeDiv(_40MS_DIV);
        //timeStep = (10*40*pow(10,-3)/10000)/pow(10,-6);
    //irehyun@softdsp.com, 2005.03.15
    break;
  case 23:
        sdSetTimeDiv(_100MS_DIV);
        //timeStep = (10*100*pow(10,-3)/10000)/pow(10,-6);
    //irehyun@softdsp.com, 2005.03.15
    break;
  case 24:
        sdSetTimeDiv(_200MS_DIV);
        //timeStep = (10*200*pow(10,-3)/10000)/pow(10,-6);
    //irehyun@softdsp.com, 2005.03.15
    break;
  case 25:
        sdSetTimeDiv(_400MS_DIV);
        //timeStep = (10*400*pow(10,-3)/10000)/pow(10,-6);
    //irehyun@softdsp.com, 2005.03.15
    break;
  case 26:
        sdSetTimeDiv(_1S_DIV);
        //timeStep = (10*1/10000)/pow(10,-6);
    //irehyun@softdsp.com, 2005.03.15
    break;
  case 27:
        sdSetTimeDiv(_2S_DIV);
        //timeStep = (10*2/10000)/pow(10,-6);
    //irehyun@softdsp.com, 2005.03.15
    break;
  case 28:
        sdSetTimeDiv(_4S_DIV);
        //timeStep = (10*4/10000)/pow(10,-6);
    //irehyun@softdsp.com, 2005.03.15
    break;
  case 29:
        sdSetTimeDiv(_10S_DIV);
        //timeStep = (10*10/10000)/pow(10,-6);
    //irehyun@softdsp.com, 2005.03.15
    break;
  }

switch(ETSMode)
```

```
      {
      case 1:
            sdSetETSOnOff(true); // ETS Mode is required for sampling
rates lower than ....
        //irehyun@softdsp.com, 2005.03.15
        break;
      case 0:
            sdSetETSOnOff(false);
        //irehyun@softdsp.com, 2005.03.15
        break;
      }


      switch(voltRange1)
      {
      case 1:
            sdSetChannelVoltage(_CH1, _10MV_DIV); // Sets the voltage
(y-axis) scale
        //irehyun@softdsp.com, 2005.03.15
        break;
      case 2:
            sdSetChannelVoltage(_CH1, _20MV_DIV);
        //irehyun@softdsp.com, 2005.03.15
        break;
      case 3:
            sdSetChannelVoltage(_CH1, _50MV_DIV);
        //irehyun@softdsp.com, 2005.03.15
        break;
      case 4:
            sdSetChannelVoltage(_CH1, _100MV_DIV);
        //irehyun@softdsp.com, 2005.03.15
        break;
      case 5:
            sdSetChannelVoltage(_CH1, _200MV_DIV);
        //irehyun@softdsp.com, 2005.03.15
        break;
      case 6:
            sdSetChannelVoltage(_CH1, _500MV_DIV);
        //irehyun@softdsp.com, 2005.03.15
        break;
      case 7:
            sdSetChannelVoltage(_CH1, _1V_DIV);
        //irehyun@softdsp.com, 2005.03.15
        break;
      case 8:
            sdSetChannelVoltage(_CH1, _2V_DIV);
        //irehyun@softdsp.com, 2005.03.15
        break;
      case 9:
            sdSetChannelVoltage(_CH1, _5V_DIV);
        //irehyun@softdsp.com, 2005.03.15
        break;
      case 10:
            sdSetChannelVoltage(_CH1, _10V_DIV);
        //irehyun@softdsp.com, 2005.03.15
        break;
      }

    switch(voltRange2)
      {
      case 1:
            sdSetChannelVoltage(_CH2, _10MV_DIV); // Sets the voltage 
(y-axis) scale
        //irehyun@softdsp.com, 2005.03.15
        break;
      case 2:
            sdSetChannelVoltage(_CH2, _20MV_DIV);
        //irehyun@softdsp.com, 2005.03.15
        break;
      case 3:
            sdSetChannelVoltage(_CH2, _50MV_DIV);
        //irehyun@softdsp.com, 2005.03.15
        break;
      case 4:
            sdSetChannelVoltage(_CH2, _100MV_DIV);
        //irehyun@softdsp.com, 2005.03.15
        break;
      case 5:
            sdSetChannelVoltage(_CH2, _200MV_DIV);
        //irehyun@softdsp.com, 2005.03.15
        break;
      case 6:
            sdSetChannelVoltage(_CH2, _500MV_DIV);
        //irehyun@softdsp.com, 2005.03.15
        break;
      case 7:
            sdSetChannelVoltage(_CH2, _1V_DIV);
        //irehyun@softdsp.com, 2005.03.15
        break;
      case 8:
            sdSetChannelVoltage(_CH2, _2V_DIV);
        //irehyun@softdsp.com, 2005.03.15
        break;
      case 9:
            sdSetChannelVoltage(_CH2, _5V_DIV);
        //irehyun@softdsp.com, 2005.03.15
        break;
      case 10:
            sdSetChannelVoltage(_CH2, _10V_DIV);
        //irehyun@softdsp.com, 2005.03.15
        break;
      }

      if (delayTime != 0)
      {
            sdSetDelayOnOff(true);
            sdSetDelayOffsetTime(delayTime);
      }
      else
      {
            sdSetDelayOnOff(false);
      }

      switch(acqChan)
      {
      case 1:
            for (i = 0; i < numAverage; i++)
            {
            //irehyun@softdsp.com, 2005.03.15
            //sdGetData : to grab data takes time according to the 
sampling rate.
            do
```

```
                  {
                dataSuccess = sdGetData(_CH1, ch1DataTemp,
ch2DataTemp); // Grab temporary data
            }
            while (!dataSuccess);

                  for (j = 0; j < 10000; j++)
                  {
                        ch1Data[j] = ch1Data[j] + ch1DataTemp[j]; //
Add the temporary data to the main array
                  }
            }
            for (i = 0; i < 10000; i++)
            {
                  ch1Data[i] = ch1Data[i] / numAverage; //Averages
the data in main array over 'numAverage' acquisitions
            }
        //irehyun@softdsp.com, 2005.03.15
        break;
      case 2:
            for (i = 0; i < numAverage; i++)
            {
            //irehyun@softdsp.com, 2005.03.15
            do
                  {
                dataSuccess = sdGetData(_CH2, ch1DataTemp,
ch2DataTemp);
            }
            while (!dataSuccess);

                  for (j = 0; j < 10000; j++)
                  {
                        ch2Data[j] = ch2Data[j] + ch2DataTemp[j];
                  }
            }
            for (i = 0; i < 10000; i++)
            {
                  ch2Data[i] = ch2Data[i] / numAverage;
            }
        //irehyun@softdsp.com, 2005.03.15
        break;
      }

      sdFinalize(); // Shut down the SDS200 (required)

      if (dataSuccess == false) // If the data acquisition failed
      {
            MessageBox(NULL, "Data acquisition failed! (error code #
2)", NULL, MB_OK); // Error window
            exit(2); // Exits with error code 2 (in this program,
that means data acquisition failed)
      }

      FILE * sdsdat = fopen("sdsdat.txt", "w"); // Opens a file for
writing data
      fprintf(sdsdat, "%s\n", "Voltage (V)"); // Fancy header for the
data file

      switch(acqChan)
      {
      case 1:
            for (i = 0; i < 10000; i++)
            {
                  fprintf(sdsdat, "%e\n", ch1Data[i]); // Write the
data to a file
            }
            break;
      case 2:
            for (i = 0; i < 10000; i++)
            {
                  fprintf(sdsdat, "%e\n", ch2Data[i]); // Write the
data to a file
            }
            break;
      }

      fclose(sdsdat);

    return 0; // If ya got here then everything is OK!
}
```

Bibliography

1. T. H. Scheper, J. M. Hilmer, F. Lammers, C. Müller and M. Reinecke, *J. Chrom. A*, **1996**, Vol. 725 (1), 3-12.

2. P. Harms, Y. Kostov, G. Rao, *Cur. Opin. Biotechnol.*, **2002,** Vol. 13 (2), 124-127.

3. N. C. van de Merbel1, H. Lingeman, U. A. Th. Brinkman, *J. Chrom. A.*, **1996,** Vol 725 (1), 13-27.

4. R. F. Severson, O. T. Chortyk, M. G. Stephenson, D. H. Akey, J. W. Neal Jr., G. W. Pittarelli, D. M. Jackson, V. A. Sisson, *Bioregulators for Crop Protection and Pest Control*, **1994**, Vol. 557, Ch. 9, 109-121, ACS Symposium Series.

5. S. Janakat, H. Al-Merie, *J. Ethnopharmacol.*, **2002**, Vol. 83 (1-2), 135-138.

6. S. Balachandran, C. B. Osmond, P. F. Daley, *Plant Physiol.*, **1994**, Vol. 104 (3), 1059-1065.

7. M. B. Kester, C. L. Saccar, M. L. Rocci Jr., H. C. Mansmann Jr., *J. Chrom.*, **1986**, Vol. 380, 99-108.

8. L. Hendeles, M. Weinberger, *Pharmacotherap.*, **1983**, Vol. 3 (1), 2-44.

9. M. H. Jacobs, R. M. Senior, G. Kessler, *JAMA*, **1976**, Vol. 235 (18), 1983-1986.

10. W. M. Mullet, E. P. C. Lai, *Anal. Chem.*, **1998**, Vol. 70 (17), 3636-3641.

11. C. P. Price, J. M. Hicks, *Point-of-Care Testing*, **1999**, 2nd Ed., 429-448, AACC Press, Washington.

12. E. Bernadi, P. Prandoni, A. W. Lensing, et al., *Br. Med. J.*, **1998**, Vol. 317, 1037-1040.

13. P. Brill-Edwards, A. Lee, *Thromb. Haemost.*, **1999**, Vol. 82, 688-694.

14. A. Perrier, S. Desmarais, M. J. Miron, P. de Moerloose, et al., *Lancet*, **1999**, Vol. 353, 190-195.

15. P. S. Wells, D. R. Anderson, *Haemostasis*, **1999**, Vol. 29 (Suppl. 1), 10-20.

16. http://www.diavant.de/diavant/CMSFront.html?pgid=3,1,7,0

http://www.roche-diagnostics.co.in/Content/products_services/cardiac_reader.html, last accessed on Aug 22, 2010.

17. M. Müller-Bardorff, H. Freitag, T. Scheffold, T, et al. *Circulation*, **1995**, Vol. 92, 2869-2875.

18. C. E .Dempfle, M. Schraml, I. Besenthal, et al. *Clinica Chimica Acta*, **2001**, Vol. 307, 211-218.

19. J. W. Smith, S. R. Steinhubl, A. M. Lincoff, et al. *Circulation*, **1999**, Vol. 99, 620-625.

20. C. P. Cannon, C. H. McCabe, S. Borzak, et al. *Circulation*, **1998**, Vol. 96, 76-81.

21. D. A. Vorchheimer, V. Fuster, *Circculation*, **1998**, Vol. 97, 312-314.

22. R. Rabøl, P. M. V. Højberg, T. Almdal, R. Boushel, S. B. Haugaard, S. Madsbad, F. Dela, *J. Clin. Endocrin. & Metab.*, **2009**, Vol. 94 (4), 1372-1378.

23. M. Mogensen, K. Sahlin, M. Fernström, D. Glintborg, B. F. Vind, H. Beck-Nielsen, K. Højlund, *Diabetes*, **2007**, Vol. 56 (6), 1592-1599.

24. M. W. Stolar, R. J. Chilton, *Clin. Therap.*, **2003**, Vol. 25 (Suppl. 2), B4-B31.

25. L. M. Thaler, V. G. Dunbar, D. C. Ziemer, et al. *Diabetes Care*, **1999**, Vol. 22, 1415-1421.

26. E. Cagliero, E. V. Levina, D. M. Nathan, *Diabetes Care*, **1999**, Vol. 22, 1785-1789.

27. C. R. Stivers, S. R. Baddam, A. L. Clark, et al. *Diabetes Tech. & Therap.*, **2000**, Vol. 2 (4) 517-526.

28. http://www.clpmag.com/issues/articles/2000-11_08.asp, last accessed on Aug. 21, 2010.

29. http://www.embeediagnostics.com/equip/glyco/diastat.htm, last accessed on Aug. 21, 2010.

30. H. Schneckenburger, K. Konig, *Optical Eng.*, **1992**, Vol. 31(7), 1447-1451.

31. K. Pettersson, T. Katajamäki, K. Irjala, V. Leppanen, K. Majamaa-Voltti, P. Laitinen, *Luminescence*, **2000**, Vol. 15, 399-407.

32. D. J. Newman, Y. Olabiran, D. W. Bedzyk, et al., *Clin. Chem.*, **1999**, Vol. 45, 822-828.

33. A. Katrukha, A. Bereznikowa, K. Pettersson, *Scand. J. Clin. Lab. Invest. Suppl.*, **1999**, Vol. 230, 124-127.

34. J. Jankovic, *J. Neurol. Neurosurg. Psychiatry,* **2008**, Vol. 79, 368-376.

35. K. G. Lloyd, L. Davidson, O. Hornykiewicz, *J. Pharmacol. Exp. Ther.*, **1975**, Vol. 195 (3), 453-464.

36. A. Barbeau, *Canad. Med. Ass. J.*, **1969**, Vol. 101, 59-68.

37. M. Fujii, K. Umezawa, Akiko Arata, *Neuroscience Research*, **2004**, Vol. 50 (3), 355-359.

38. P. M. Lalley, *Respir. Physiol. & Neurobiol.*, **2008**, Vol. 164 (1-2), 160-167.

39. A. Ciarka, J.-L. Vincent, P. van de Borne, *Pulmon. Pharmacol. & Therap.*, **2007**, Vol. 20 (6), 607-615.

40. L. I. Goldberg, *Pharmacol. Rev.*, **1972**, Vol. 24(1), 1-29.

41. C. D .Kilts, G. R. Breese, R. B. Mailman, *J. Chrom.*, **1981**, Vol. 225, 347-357.

42. M. D. Rubianes, G. A. Rivas, *Anal. Chim. Acta*, **2001**, Vol. 440, 99-108.

43. R. Selyanchyn , S. Korposh , S. Wakamatsu, S.-W. Lee, *Sensors*, **2011**, Vol. 11(1), 1177-1191.

44. S. J. Martin, G. C. Frye, A. J. Ricco, S. D. Senturia, *Anal. Chem.*, **1993**, Vol. 65 (20), 2910–2922.

45. R. G. Berger, *Flavours and Fragrances Chemistry*, **2007**, 1st ed., 313-361, Springer: Berlin.

46. F. L. Dickert, O. Hayden, R. Bindeus, K.-J. Mann, D. Blaas and E. Waigmann, *Anal. & Bioanal. Chem.*, **2003**, Vol. 378 (8), 1929-1934.

47. M. A. Cooper, F. N. Dultsev, T. Minson, V. P. Ostanin, C. Abell & D. Klenerman, *Nat. Biotechnol.*, **2001**,Vol. 19, 833-837.

48. C. Yaoa , T. Zhu, J. Tang, R. Wu, Q. Chen, M. Chen, B. Zhang, J. Huang, W. Fu, *Biosens. & Bioelectron.*, **2008**, Vol. 23 (6), 879-885.

49. L. Hoff , P. C. Sontum, J. M. Hovem, *J. Acoust. Soc. Am.*, **2000**, Vol. 107 (4), 2272-2280.

50. F. Forsberg, W.T. Shi, B.B. Goldberg, *Ultrasonics*, **2000**, Vol. 38, 93-98.

51. Z. Hu, *App. Phys. Let.*, Vol. 74 (15), **1999**.

52. J. Matsui, K. Akamatsu, et al., *Anal. Chem*., **2004**, Vol. 76, 1310-1315.

53. R. Nossal, *Rubber Chem. Technol.*, **1988**, Vol. 16, 255-260.

54. W. R. Seitz et al., *Anal. Chim. Acta*, **1999**, Vol. 400, 55-64.

55. K. F. Rabe, H. Magnussen, G. Dent, *Eur. Respir. J.*, **1995**, Vol. 8, 637-642.

56. A. Ojha, A. Pargal, *J. Pharm. Biomed. Anal.*, **1999**, Vol. 21, 175-178.

57. R. Panchagnula, A. Sood, N. Sharda, et al., *J. Pharm. Biomed. Anal.*, **1999**, Vol. 18, 1013-1020.

58. I. Calleja, M. J. Blanco-Príeto, N. Ruz, M., et al., *J. Chromatogr.*, **2004**, Vol. 1031, 289-294.

59. P.J. Barnes, *Amer. Jour. Respir. Crit. Care Med.*, **2003**, Vol. 167, 813-818.

60. K. Mosbach, O. Ramström, *Nat. Biotechnol.*, **1996**, Vol. 14, 163-170.

61. M. E. Byrne, V. Saliana, *Int. J. Pharm.*, **2008**, Vol. 364, 188-212.

62. R. S. Gill, M. Marquez, G. Larsen, *Micropor. Mesopor. Mater.*, **2005**, Vol. 85, 129-135.

63. G. Vlatakis, L. I. Anderson, R. Muller, K. Mosbach, *Nature*, **1993**, Vol. 361, 645-647.

64. W. Fan, W. R. Seitz, *Analyst*, **2007**, Vol. 132, 1103-1106.

65. R. Nossal, *Rubber Chem. Technol.*, **1998**, Vol. 16, 255-260.

66. W. R. Seitz et al., *Anal. Chim. Acta*, **1999**, Vol. 400, 55-64.

67. A. Strybulevych, V. Leroy, M.G. Scanlon, J.H. Page, *Soft Matter*, **2007**, Vol. 3, 1388–1394.

68. L. Hoff, P. C. Sontum, J. M. Hovem, *J. Acoust. Soc. Am.*, **2000**, Vol. 107, 2272-2280.

69. A. Kvikliene et al., *Ultrasonics*, **2004**, Vol. 42, 301-307.

70. U. Anbergen, W. Oppermann, *Polymer*s, **1990**, Vol. 31, 1854-1858.

71. M. E. Byrne, K. Park, N. A. Peppas, *Adv. Drug Deliv. Rev.*, **2002**, Vol. 54, 149-161.

72. C. E. W. Gributs, D. H. Burns, *Chemom. Intell. Lab. Sys.*, **2006**, Vol. 83, 44-53.

73. S. R. Carter, S. Rimmer, *Adv. Mat.*, **2002**, Vol. 14, 667-670.

74. I. Calleja, M.J. Blanco-Príeto, N. Ruz, M.J. Renedo, M.C. Dios-Viéitez, *J. Chromatogr.*, **2004**, Vol. 1031, 289-294.

75. I. Seri, R. Tan, J. Evans, *Pediatrics*, **2001**, Vol. 107 (5), 1070-1074.

76. A. De Iuliis, G. Arrigoni et al., *Biochim. Biophys. Acta*, **2008**, Vol. 1784, 1687–1693.

77. A. Suzuki et al., *Anal. Chem.*, **2007**, Vol. 79, 8608-8615.

78. B. B. Goldberg, J. S. Raichlen, F. Forsberg, *Ultrasound Contrast Agents: Basic Principles and Clinical Applications*, **2001**, 2nd Ed., Martin Dunitz Ltd., London.

79. T. Oya et al., *Science*, **1999**, Vol. 286, 1543-1545.

80. E. Kato, *J. Chem. Phys.*, **1997**, Vol. 106 (9), 3792-3797.

81. M. Watanabe, T. Akahoshi, Y. Tabata, D. Nakayama, *J. Am. Chem. Soc.*, **1998**, Vol. 120, 5577-5578.

82. J. Matsui, K. Akamatsu et al., *Anal. Chem.*, **2004**, Vol. 76, 1310-1315.

83. A. De Iuliis, G. Arrigoni et al., *Biochim. Biophys. Acta*, **2008**, Vol. 1784, 1687-1693.

84. F. Bischoff, A. Torres, *Clin. Chem.*, **1962**, Vol. 8 (4), 370-377.

85. C. V. Atack, *Br. J. Pharmac.*, **1973**, Vol. 48, 699-714.

86. R. Nossal, *Rubber Chem. Technol.*, **1988**, Vol. 61 (2), 255-260.

87. W. R. Seitz et al., *Anal. Chim. Acta*, **1999**, Vol. 400, 55-64.

88. A. Kvikliene et al., *Ultrasonics*, **2004**, Vol. 42 (1-9), 301-307.

89. Y. Okamoto, H. S. Lee, S. T. Attarwala, *J. Org. Chem.*, **1985**, Vol. 50, 2788-2790.

90. D. H. Sullivan et al., *Am. J. Clin. Nutr.*, **1990**, Vol. 51, 749-758.

91. I. Seri, R. Tan, J. Evans, *Pediatrics*, **2001**, Vol. 107 (5), 1070-1074.

92. Y. Li., Z. Hu., Y. Chen, *J. Appl. Polym. Sci.* **1997**, Vol. 63 1173-1178.

93. V. C. Lopez, J. Hadgraft, M. J. Snowden, *Inter. J. of Pharma.*, **2005**, Vol. 292, 137-147.

94. H. Feil, Y. H. Bae, J. Feijenb, S. W. Kim, *J. Membr. Sci.*, **1991**, Vol. 64, 283-294.

95. T. Oya, et al., *Science*, **1999**, Vol. 286, 1543-1545.

96. K. V. Durme, H. Rahier, B. V. Mele, *Macromolecules*, **2005**, Vol. 38, 10155-10163.

97. E. Juliac, T. Mitsumata, T. Taniguchi, K. Iwakura, K. Koyama, *J. Phys. Chem. B*, **2003**, Vol. 107, 5426-5431.

98. M. Watanabe, T. Akahoshi, Y. Tabata, D. Nakayama, *J. Am. Chem. Soc.*, **1998**, Vol. 120, 5577-5578.

99. J. Matsui, K. Akamatsu, et al. *Anal. Chem.*, **2004**, Vol. 76, 1310-1315.

100. B. B. Goldberg, J. S. Raichlen, F. Forsberg, *Ultrasound Contrast Agents: Basic Principles and Clinical Applications*, **2001**, 2nd ed, Martin Dunitz Ltd: London.

101. F. Katherine, P. Rachel, B. Mark, *Annu. Rev. Biomed. Eng.*, **2007**, Vol. 9, 415-447.

102. H. Madjar, *Ultrasound Obstet Gynecol*, **2000**, Vol. 16, 111-114.

103. S. L. Mulvagh, et al., *J. Am. Soc. Echocardio.*, **2000**, Vol. 13 (4), 331-342.

104. K. F. Rabe, H. Magnussen, G. Dent, *Eur. Respir. J.*, **1995**, Vol. 8, 637-642.

105. E. Dailly, F. Chenu, C. E. Renard, M. Bourin, *Fundam. & Clin. Pharmacol.*, **2004**, Vol.18, 601-607.

106. G. Sedvall, *TINS*, **1990**, Vol. 13 (7), 302-308.

107. F. Bischoff, A. Torres, *Clin. Chem.*, **1962**, Vol. 8 (4), 370-377.

108. Atack, C. V. *Br. J. Pharmac.*, **1973**, Vol. 48, 699-714.

109. A. Kvikliene, et al., *Ultrasonics*, **2004**, Vol. 42 (1-9), 301-307.

110. R. Nossal, *Rubber Chem. Technol.*, **1988**, Vol. 61 (2), 255-260.

111. W. R .Seitz, et al. *Anal. Chim. Acta*, **1999**, Vol. 400, 55-64.

112. M. E. Byrne, K. Park, N. A. Peppas, *Adv. Drug Deliv. Rev.*, **2002**, Vol. 54, 149-161.

113. Y. Okamoto, H. S. Lee, S. T. Attarwala, *J. Org. Chem.*, **1985**, Vol. 50, 2788-2790 .

114. D. Huo, T. Kobayashi, *Chemistry Letters*, **2006**, Vol. 35 (7), 776.

115. P. J. Barnes, *Amer. Jour. .Respir. & Crit. Care Med.* **2003**, Vol. 167, 813-818.

116. I. Seri, R. Tan, J. Evans, *Pediatrics*, **2001**, Vol. 107 (5), 1070-1074.

117 R. E. Franklin, K. C. Holmes, *Biochim. Biophys. Acta*, **1956**, Vol. 21, 405.

118. N. A. Terrault, et al., *J. Med. Vir.*, **1998**, Vol. 51, 217-224.

119. J. Mulder, et al., *J. Clinic. Microbiol.*, **1994**, Vol. 32, 292-300.

120. T. Tanaka, et al., *J. Hepat.*, **1995**, Vol. 23 (6), 742-745.

121. M. A. McClain, et al., *Anal. Chem.*, **2001**, Vol. 73 (21), 5334-5338.

122. S.-S. Zhang, et al., *Fresen. J. Anal. Chem.*, **1999**, Vol. 364 (8), 758-762.

123. C. Singleton, *J. Mol. Recognit.*, **2007**, Vol. 20, 154-184.

124. M. Nakamura, I. Sugimoto, *Artificial Neural Networks*, **1999**, Vol. 470, 649-654.

125. R. G. Berger, *Flavours and Fragrances Chemistry, Bioprocessing and Sustainability*, **2007**, 1st Ed., 313-360, Springer, Berlin.

126. M. Thompson, A. L. Kiplingt, W. C. Duncan-Hewitt, L. V. Rajakovic, B. A. Cavic-Vlasak, *Analyst*, **1991**, Vol. 116, 881-890.

127. B. B. Goldberg, J. S. Raichlen, F. Forsberg, *Ultrasound Contrast Agents: Basic Principles and Clinical Applications*, **2001**, 2nd ed., Martin Dunitz Ltd., London.

128. Z. Hu, R. Appel, W. Xu, T. W. Zerda, *Macromolecules*, **1998**, Vol. 31, 5071-5074.

129. Y. Li, G. Wang, Z. Hu, *Macromolecules*, **1995**, Vol. 28, 4194.

130. Z. Hu, Y. Chen, C. Wang, Y. Zheng, Y. Li, *Nature*, **1998**, Vol. 393, 97.

131. M. Watanabe, T. Akahoshi, Y. Tabata, D. Nakayama, *J. Am. Chem. Soc.*, **1998**, Vol. 120, 5577-5578.

132. J. Matsui, K. Akamatsu et al., *Anal. Chem.*, **2004**, Vol. 76, 1310-1315.

133. M. E. Byrne, K. Park, N. A. Peppas, *Adv. Drug Deliv. Rev.*, **2002**, Vol. 54, 149-161.

134. R. Nossal, *Rubber Chem. Technol.*, **1988**, Vol. 61, 255-260.

135. W. R. Seitz, et al., *Anal. Chim. Acta*, **1999**, Vol. 400, 55-64.

136. A. Kvikliene, R. Jurkonis, M. Ressner, L. Hoff, T. Jansson, B. Janerot-Sjoeberg, A. Lukosevicius, P. Ask, *Ultrasonics*, **2004**, Vol. 42(1-9), 301-307.

137. H. Huang, I. Szleifer, N. A. Peppas, *Macromolecules*, **2002**, Vol. 35, 1373-1380.

138. T. Okajima, I. Harada, K. Nishio, S. Hirotsu, *J. Chem. Phys.*, 2002, Vol. 116 (20), 068-9077.

139. S. Hirotsu, *Macromolecules*, **1990**, Vol. 23 (3), 903-905.

140. S. Hirotsu, *J. Chem. Phys.*, **1991**, Vol. 94 (5).

141. Y. Li, Z. Hu, Y. Chen, *J. Appl. Polym. Sci.*, **1997**, Vol. 63, 1173–1178.

142. J. Matsui, K. Akamatsu, et al., *Anal. Chem.*, **2004**, Vol. 76, 1310-1315.