

SUNFEST

www.seas.upenn.edu/~sunfest



2008

SUMMER UNDERGRADUATE FELLOWSHIPS IN SENSOR TECHNOLOGIES



TECHNICAL REPORT

TR-CST12SEP08

*Center for Sensor Technologies
University of Pennsylvania
Philadelphia, PA 19104*

Acknowledgement

This material is based upon work supported by the National Science Foundation under Grant No. 0754741.

Disclaimer

Any opinions, findings, and conclusions or recommendations expressed in this material are those of the author(s) and do not necessarily reflect the views of the National Science Foundation.



SUNFEST 2008



SUNFEST 2008

SUMMER UNDERGRADUATE FELLOWSHIP IN SENSOR TECHNOLOGIES

Sponsored by the National Science Foundation (Award no. EEC-0754741)

PREFACE

This report is the result of eleven undergraduate students' research efforts during the summer of 2008. From May 28th through August 09, 2008. Eleven students from Penn and other colleges participated in the SUNFEST program, which is organized by the Center for Sensor Technologies of the School of Engineering and Applied Science at the University of Pennsylvania. This unique "Summer Experience for Undergraduates in Sensor Technologies" program was initiated in 1986 and has grown considerably in size. It is now recognized as one of the most successful summer programs for undergraduates in the country. I would like to express my sincere gratitude to the National Science Foundation for their continued support for this REU Site.

The purpose of the SUNFEST program is to provide bright, motivated undergraduate students with the opportunity to become involved in active research projects under the supervision of a faculty member and his graduate student(s). The general area of research concentrates on sensor technologies and includes projects such as materials and technology for sensors, microstructures, smart imagers, bio-sensors and robotics. By providing the students with hands-on experience and integrating them with a larger research group where they can work together with other students, the program intends to guide them in their career choices. By exposing the students to the world of research, we hope they will be more inclined to go on for advanced degrees in science and engineering.

The students participated in a variety of hands-on workshops in order to give them the tools to do first-rate research or enhance their communication skills. These included "Ethics in Science and Engineering", "Use of Electronic Databases", "Applying to Graduate School" and "Writing Technical Reports". Students also had plenty of opportunity for social interactions among themselves or with faculty and graduate student advisors.

This booklet contains reports from this year's projects, the quality of which testifies to the high level of research and commitment by these students and their supervisors. I would like to express my sincere thanks to the students for their enthusiastic participation; the help of the faculty members, graduate students and support staff is very much appreciated. I would also like to thank Ms. Delores Magobet, Sherri Butler, Sid Deliwala, and the rest of the ESE staff for their invaluable help in making this program run smoothly.

Jan Van der Spiegel, Director
Center for Sensor Technologies

FINAL REPORT
2008 SUMMER UNDERGRADUATE FELLOWSHIP
IN SENSOR TECHNOLOGIES
Sponsored by the National Science Foundation

<http://www.ese.upenn.edu/~sunfest/pastProjects/Projects08.html>

TABLE OF CONTENTS

LIST OF STUDENT PARTICIPANTS	iv-x
ENABLING FEEDBACK FORCE CONTROL FOR COOPERATIVE TOWING ROBOTS <i>Clarence E. Agbi (Electrical Engineering) – Yale University</i> <i>Advisors: Dr. Vijay Kumar and Jim Keller</i>	1 – 14
GROOVENET3: A SCALABLE TRAFFIC SIMULATOR FOR CONGESTION PROBING AND PREDICTION <i>Uchenna Kevin Anyanwu (Computer Engineering) – California State University at San Jose</i> <i>Advisor: Rahul Mangharam</i>	15 - 36
EVALUATING AN INTERLEUKIN-1β INJECTION TO INDUCE DEGENERATION IN THE RAT LUMBAR INTERVERTEBRAL DISC <i>Alta Berger (BME) – George Washington University</i> <i>Advisors: Amy Orlansky, MS, and Dawn Elliot, PhD</i>	37 - 52
DYE-SENSITIZED ZNO FIBERS FROM ELECTROSPINNING AND PHOTOVOLTAIC CELLS <i>Ramon Luis Figueroa-Diaz (Chemistry) – University of Puerto Rico at Cayey</i> <i>Advisor: Jorge Santiago-Aviles</i>	53 – 68
NEONUR: A FEEDING DEVICE FOR PREMATURE NEONATAL NURSING <i>David E. Joffe (Electrical and Computer Engineering) – Carnegie Mellon University</i> <i>Advisor: Dr. Jay N. Zemel</i>	69 - 80
GROWTH OF CARBON NANOTUBES VIA CHEMICAL VAPOR DEPOSITION <i>Alexei Matyushov (Physics) – Arizona State University</i> <i>Advisors: Zhengtang Luo, A.T. Charlie Johnson</i>	81 - 95
DEVELOPMENT OF A MICRO PCR REACTOR FOR LAB-ON-A-CHIP DEVICES <i>Erika Martinez Nieves (Art and Science Department) - University of Puerto Rico</i> <i>Advisor: Haim H. Bau</i>	96 - 110
OPTIMIZING LEGGED LOCOMOTION USING TUNABLE LEG <i>Kamruzzaman Rony (Electrical Engineering) – Stony Brook University</i> <i>Advisors: Dr. Dan Koditschek and Kevin Galloway</i>	111 - 138

TALKING TO ROBOTS: SPEECH-DIRECTED MOTION PLANNING <i>Anil Venkatesh (Electrical Engineering) – University of Pennsylvania</i> <i>Advisor: Dr. George J. Pappas</i>	139 - 153
INVESTIGATION OF MSC DIFFERENTIATION ON ELECTROSPUN NANOFIBROUS SCAFFOLDS <i>Emily Wible (Bioengineering) – University of Pennsylvania</i> <i>Advisor: Robert L. Mauck and Nandan L. Nerurkar</i>	154 - 163
COMPACT ATTITUDE SENSOR SYSTEM USING SR-UKF <i>Christopher Baldassano (Electrical Engineering) – Princeton University</i> <i>Advisor: Dr. Mark Yim</i>	164 - 195

**LIST OF STUDENT PARTICIPANTS
IN THE SUNFEST PROGRAM SINCE 1986**

Summer, 2008

Clarence Agbi	Yale University
Uchenna Anyanwu	San Jose State University
Christopher Baldassano	Princeton University
Alta Berger	George Washington University
Ramon Luis Figueroa	University of Puerto Rico
David Joffe	Carnegie Mellon University
Erika Martinez	University of Puerto Rico
Alexei Matyushov	Arizona State University
Kamruzzaman Rony	Stony Brook University
Anil Venkatesh	University of Pennsylvania
Emily Wible	University of Pennsylvania

Summer, 2007

Mulutsga Bereketab	Virginia Polytechnic Institute Sonia
A. Bhaskar	Princeton University
Patrick Duggan	Providence College
Nataliya Kilevskaya	University of Florida
Ryan Li	Case Western Reserve University
Viktor L. Orekhov	Tennessee Technological University
Andrew Potter	Brown University
Pamela Tsing	University of Pennsylvania
Victor Uriarte	Florida International University
Adriane Wotawa-Bergen	University at Buffalo
Arelys Rosado Gomez	University of Puerto Rico

Summer, 2006

Sam Burden	University of Washington
Jose M. Castillo Colon	University of Puerto Rico
Alexsandra Fridshtand	Lehigh University
Shakera Guess	Lincoln University
Journee Isip	Columbia University
Nathan Lazarus	University of Pennsylvania
Armand O'Donnell	University of Pennsylvania
William Peeples	Lincoln University
Raúl Pérez Martínez	University of Puerto Rico
Helen Schwerdt	Johns Hopkins University
Xiaoning Yuan	Duke University

Summer, 2005

Robert Callan	University of Pennsylvania
David Cohen	University of Pennsylvania
Louie Huang	University of Pennsylvania
Roman Geykhman	University of Pennsylvania
An Nguyen	University of Pennsylvania
Olga Paley	University of California at Berkeley
Miguel Perez Tolentino	University of Puerto Rico
Ebenge Usip	University of Southern California
Adam Wang	University of Texas, Austin
Kejia Wu	University of Pennsylvania

Summer, 2004

Benjamin Bau	Massachusetts Institute of
Technology	University of Pennsylvania
Alexander H. Chang	University of Pennsylvania
Seth Charlip-Blumlein	University of Pennsylvania
Ling Dong	University of Rochester
David Jamison	Johns Hopkins University
Dominique Low	University of Pennsylvania
Emmanuel U. Onyegam	University of Texas, Dallas
J. Miguel Ortigosa	Florida Atlantic University
William Rivera	University of Puerto Rico, Mayaguez
Matthew Saucedo	Texas A&M University, Kingsville
Olivia Tsai	Carnegie Mellon University

Summer, 2003

Emily Blem	Swarthmore College
Brian Corwin	University of Pennsylvania
Vinayak Deshpande	University of Virginia
Nicole DiLello	Princeton University
Jennifer Geinzer	University of Pittsburgh
Jonathan Goulet	University of Pennsylvania
Mpitulo Kala-Lufulwabo	University of Pittsburgh
Emery Ku	Swarthmore College
Greg Kuperman	University of Pennsylvania
Linda Lamptey	University of Pennsylvania
Prasheel Lillaney	University of Pennsylvania
Enrique Rojas	University of Pennsylvania

Summer, 2002

Christopher Bremer	Colorado School of Mines
Aslan Ettehadien	Morgan State University
April Harper	Hampton University
Catherine Lachance	University of Pennsylvania
Adrian Lau	University of Pennsylvania
Cynthia Moreno	University of Miami
Yao Hua Ooi	University of Pennsylvania
Amber Sallerson	University of Maryland/Baltimore
Jiong Shen	University of California-Berkeley
Kamela Watson	Cornell University
John Zelena	Wilkes University

Summer, 2001

Gregory Barlow	North Carolina State University
Yale Chang	University of Pennsylvania
Luo Chen	University of Rochester
Karla Conn	University of Kentucky
Charisma Edwards	Clark Atlanta University
EunSik Kim	University of Pennsylvania
Mary Kutteruf	Bryn Mawr College
Vito Sabella	University of Pennsylvania
William Sacks	Williams College
Santiago Serrano	Drexel University
Kiran Thadani	University of Pennsylvania
Dorci Lee Torres	University of Puerto Rico

Summer, 2000

Lauren Berryman	University of Pennsylvania
Salme DeAnna Burns	University of Pennsylvania
Frederick Diaz	University of Pennsylvania (AMPS)
Hector Dimas	University of Pennsylvania (AMPS)
Xiomara Feliciano	University of Turabo (Puerto Rico)
Jason Gillman	University of Pennsylvania
Tamara Knutsen	Harvard University
Heather Marandola	Swarthmore College
Charlotte Martinez	University of Pennsylvania
Julie Neiling	University of Evansville, Indiana
Shiva Portonova	University of Pennsylvania

Summer, 1999

David Auerbach	Swarthmore
Darnel Degand	University of Pennsylvania
Hector E. Dimas	University of Pennsylvania
Ian Gelfand	University of Pennsylvania
Jason Gillman	University of Pennsylvania
Jolymar Gonzalez	University of Puerto Rico
Kapil Kedia	University of Pennsylvania
Patrick Lu	Princeton University
Catherine Reynoso	Hampton University
Philip Schwartz	University of Pennsylvania

Summer, 1998

Tarem Ozair Ahmed	Middlebury University
Jeffrey Berman	University of Pennsylvania
Alexis Diaz	Turabo University, Puerto Rico
Clara E. Dimas	University of Pennsylvania
David Friedman	University of Pennsylvania
Christin Lundgren	Bucknell University
Heather Anne Lynch	Villanova University
Sancho Pinto	University of Pennsylvania
Andrew Utada	Emory University
Edain (Eddie) Velazquez	University of Pennsylvania

Summer, 1997

Francis Chew	University of Pennsylvania
Gavin Haentjens	University of Pennsylvania
Ali Hussain	University of Pennsylvania
Timothy Moulton	University of Pennsylvania
Joseph Murray	Oklahoma University
O'Neil Palmer	University of Pennsylvania
Kelum Pinnaduwege	University of Pennsylvania
John Rieffel	Swarthmore College
Juan Carlos Saez	University of Puerto Rico, Cayey

Summer, 1996

Rachel Branson	Lincoln University
Corinne Bright	Swarthmore College
Alison Davis	Harvard University
Rachel Green	Lincoln University
George Koch	University of Pennsylvania
Sandro Molina	University of Puerto Rico-Cayey
Brian Tyrrell	University of Pennsylvania
Joshua Vatsky	University of Pennsylvania
Eric Ward	Lincoln University

Summer, 1995

Maya Lynne Avent	Lincoln University
Tyson S. Clark	Utah State University
Ryan Peter Di Sabella	University of Pittsburgh
Oswaldo L. Figueroa	University of Puerto Rico-Humacao
Colleen P. Halfpenny	Georgetown University
Brandeis Marquette	Johns Hopkins
Andreas Olofsson	University of Pennsylvania
Benjamin A. Santos	University of Puerto Rico-Mayaguez
Kwame Ulmer	Lincoln University

Summer, 1994

Alyssa Apsel	Swarthmore College
Everton Gibson	Temple University
Jennifer Healy-McKinney	Widener University
Peter Jacobs	Swarthmore College
Sang Yoon Lee	University of Pennsylvania
Paul Longo	University of Pennsylvania
Laura Sivitz	Bryn Mawr College
Zachary Walton	Harvard University

Summer, 1993

Adam Cole	Swarthmore College
James Collins	University of Pennsylvania
Brandon Collings	Hamilton University
Alex Garcia	University of Puerto Rico
Todd Kerner	Haverford College
Naomi Takahashi	University of Pennsylvania
Christopher Rothey	University of Pennsylvania
Michael Thompson	University of Pennsylvania
Kara Ko	University of Pennsylvania

David Williams
Vassil Shtonov

Cornell University
University of Pennsylvania

Summer, 1992

James Collins
Tabbatha Dobbins
Robert G. Hathaway
Jason Kinner
Brenelly Lozada
P. Mark Montana
Dominic Napolitano
Marie Rocelie Santiago

University of Pennsylvania
Lincoln University
University of Pennsylvania
University of Pennsylvania
University of Puerto Rico
University of Pennsylvania
University of Pennsylvania
Cayey University College

Summer, 1991

Gwendolyn Baretto
Jaimie Castro
James Collins
Philip Chen
Sanath Fernando
Zaven Kalayjian
Patrick Montana
Mahesh Prakriya
Sean Slepner
Min Xiao

Swarthmore College
University of Puerto Rico
University of Pennsylvania
Temple University
University of Pennsylvania
University of Pennsylvania

Summer, 1990

Angel Diaz
David Feenan
Jacques Ip Yam
Zaven Kalayjian
Jill Kawalec
Karl Kennedy
Jinsoo Kim
Colleen McCloskey
Faisal Mian
Elizabeth Penadés

University of Puerto Rico
University of Pennsylvania
University of Pennsylvania
University of Pennsylvania
University of Pennsylvania
Geneva
University of Pennsylvania
Temple University
University of Pennsylvania
University of Pennsylvania

Summer, 1989

Peter Kinget	Katholiek University of Leuven
Chris Gerdes	University of Pennsylvania
Zuhair Khan	University of Pennsylvania
Reuven Meth	Temple
Steven Powell	University of Pennsylvania
Aldo Salzberg	University of Puerto Rico
Ari M. Solow	University of Maryland
Arel Weisberg	University of Pennsylvania
Jane Xin	University of Pennsylvania

Summer, 1988

Lixin Cao	University of Pennsylvania
Adnan Choudhury	University of Pennsylvania
D. Alicea-Rosario	University of Puerto Rico
Chris Donham	University of Pennsylvania
Angela Lee	University of Pennsylvania
Donald Smith	Geneva
Tracey Wolfsdorf	Northwestern University
Chai Wah Wu	Lehigh University
Lisa Jones	University of Pennsylvania

Summer, 1987

Salman Ahsan	University of Pennsylvania
Joseph Dao	University of Pennsylvania
Frank DiMeo	University of Pennsylvania
Brian Fletcher	University of Pennsylvania
Marc Loinaz	University of Pennsylvania
Rudy Rivera	University of Puerto Rico
Wolfram Urbanek	University of Pennsylvania
Philip Avelino	University of Pennsylvania
Lisa Jones	University of Pennsylvania

Summer, 1986

Lisa Yost	University of Pennsylvania
Greg Kreider	University of Pennsylvania
Mark Helsel	University of Pennsylvania

ENABLING FEEDBACK FORCE CONTROL FOR COOPERATIVE TOWING ROBOTS

NSF Summer Undergraduate Fellowship in Sensor Technologies
Clarence E. Agbi (Electrical Engineering) – Yale University
Advisors: Dr. Vijay Kumar and Jim Keller

ABSTRACT

Present research looks at the manipulation of a payload using multiple cooperative robots to drag it with cables to an assigned position within a given degree of accuracy. This project will increase the autonomy of simple tasks such as towing and will improve the efficiency of such systems. Moreover, these results will be applied towards improving cooperative technology and incorporated into the research of Jonathan Fink, et al [1] at the GRASP Lab. The project endeavors to accurately measure cable tensions between the payload and multiple cooperative robots in real time, transmitting this information to the user. We use cost-effective load cell sensors to measure tensile forces, outputting a change in frequency in proportion to the change in tension on the cell. Next, these signals are routed into the I/O pins of the BASIC Stamp 2 which wirelessly communicate this data to another BASIC Stamp 2 using an RF transmitter and receiver at 433.92 MHz. Our preliminary results show that we can wirelessly transmit information to a user, thus enabling crucial feedback to this system. This feedback would allow the robots to correct their position and velocity to maintain tension, allowing for less positional error in navigating the payload.

1. INTRODUCTION

As an emerging field, robotics takes an interdisciplinary approach towards efficient automation. Systems such as the robotic arm integrate mechatronics and artificial intelligence to maximize efficiency of simple tasks at minimum costs. Present cutting-edge research attempts this feat using cooperative robotic manipulation, allowing for multiple robotic systems to manipulate or move the same object. This combines the social theory of cooperation practiced in nature by animals (e.g. birds, bees, humans) with a mechanized task or process.

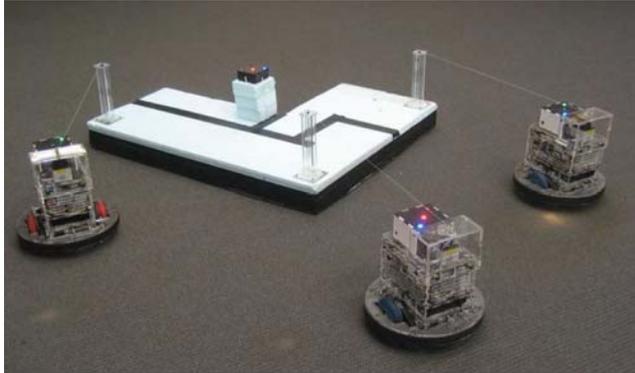


Figure 1: Robots cooperatively tow a payload using cables. This system employs no feedback of cable tension, causing some cables to go slack in the process.

While cooperative robotics is not a new idea, it has only recently been applied to tasks such as towing, a process that currently requires human input. The lack of studies in this important manipulation process has led to our research in cooperative towing [1]. We are investigating the use of multiple robots to maximize the accuracy and efficiency of towing large objects while minimizing positional error and human input. Specifically, we experiment with feedback control to improve our current system of cooperative towing.

This system, as shown in Figure 1, currently uses three automated ‘tugboats’ or robots to pull a heavier payload using a thin cable. As the robots tow the payload, they exert tensile forces on the attached cables causing the payload to move against opposing frictional forces. Sometimes, however, some of the cables go slack, causing the payload to move independently of the robots and increasing positional error of the payload. Thus, feedback control becomes very important in driving down this error and increasing efficiency. Our current stage of research seeks to implement feedback of the cable tensions into the system to monitor slackness, and to force robots to correct the cable tension. Consequently, the cables experience slack less frequently which means increased towing efficiency.

This paper is organized as follows: Section 2 discusses the background and basis for this formative work in cooperative towing, using prior research in the robotics and automation field. Sections 3 and 4 address the design aspects of the feedback system, focusing on the load cell and the BASIC Stamp respectively. Section 5 illustrates preliminary experimental results of the feedback system, and uses data analysis to discuss the consequences of these results. Section 6 details recommendations for future work, and Section 7 acknowledges those entities that made this research possible. Finally, Section 8 lists references on which this research is based.

2. DEVELOPMENT OF COOPERATIVE TOWING

2.1 Cooperative Robotics

Cooperative robotics uses multiple robots, working in coordination through sensory perception or explicit communication [2], to accomplish a set task. Thus, there are several advantages to cooperative robotics over using a single robot to accomplish the same task [2]. Inherent to cooperative robots is the ability to be flexible and adapt to different situations. Whereas some systems rely on one leader and many followers, cooperation suggests a decentralized system where work is equally shared among all systems or robots [2, 3]. Likewise, it also means that the failure of one robot does not result in the disruption of the entire system, since the remaining functional robots can adjust and adapt to the change in environment. The assumption is that cooperative robots are knowledgeable of their position and movement in real time as well as their neighbors' position and movement in real time [2].

2.2 Applying Cooperative Robotics to Simple Tasks

Prior research in the area looked at the use of robots with arms or casters to carry the object, [3] eliminating the problem of slack cable. An improved version of this research was later proposed where no force sensors were used, but rather looked at error estimation [4]. Other studies looked at fostering a decentralized system by switching the roles of leader among the robots and examining the way the robots communicate with each other [2]. Later research ventured into cooperative towing using cables, though the authors acknowledged the shortcomings in control and suggested feedback as a future improvement [1]. Based on these findings, we propose to implement a feedback system using microcontrollers and force sensors to measure the tension and enhance the performance (dynamic response and accuracy) of the system.

3. LOAD CELL TECHNOLOGY

An integral component of our research is to measure the tensile forces between the payload and the towing robots. Good force measurements require very precise load cells or force sensors that provide consistent readings in spite of slight changes in condition such as temperature. Additionally, we require miniature sensors that will not add weight to the payload and can measure maximum forces of 10lbs as determined by the dry friction between the payload and the floor. Consequently, only a handful of load cells or force sensors could be considered for this project.

3.1 Strain Gauge Load Cells

Most load cells use precise strain gauges to measure strain, which is directly proportional to force. However, strain gauges are very sensitive pieces of equipment and expensive to manufacture. To cut manufacturing and packaging costs, we endeavored to produce a cost-effective strain gauge load cell. This proved to be difficult, though, since factors such as sensitivity to environmental conditions, amplification of small strains, and bonding gauges to appropriate metals could not be easily controlled.

3.2 Capacitive Load Cells

In our quest for better force sensors, we look to the iLoad Mini produced by LoadStar and recently adapted by Parallax, Inc. Shown in Figure 2, this load cell uses breakthrough capacitive technology for load sensing, which harnesses changes in capacitance to measure loads quickly and accurately [5]. The iLoad Mini is the smallest load cell made by LoadStar with a diameter of 1.25 inches. Under applied loads, it outputs a 5V Transistor-Transistor Logic (TTL) square wave whose frequency is proportional to compression or tensile forces associated to the load.



Figure 2: The iLoad Mini

LoadStar also claims 2% full scale accuracy for hysteresis, linearity, and repeatability as well as environmental compensation from 5°C to 40°C [5]. Using this device, we could minimize errors due to hysteresis and thermal conditions while still being cost effective and precise in measurement. Lastly, the iLoad Mini is relatively inexpensive compared to other strain gauge load cells from competing companies. It features a starting price of \$99.

3.3 Implementing the Capacitive Load Cells

Next, we integrate the load cells into the overall design of the feedback loop. Each load cell is governed by the set of equations below, relating the changes in frequency output to the applied load. They operate at 5V DC and require a bit-operated input control pin, CTRL. When CTRL is set high (1), the frequency pin outputs C_x which refers to the load-related frequency. Otherwise, a low input (0) to CTRL will cause the frequency pin to output C_{ref} , a reference frequency used for environmental compensation. Because all three load cells do not behave exactly the same, they have different constants (K , q_A , q_B , q_C) to calibrate for minor differences. Table 1 show these different constant values provided by Parallax for each sensor as identified in the governing equations.

Equations 1-3, below, describe the set of equations that relate applied loads to frequencies C_x and C_{ref} . The first equation uses C_{ref} , C_x , and K to produce a corrected capacitance, C_c , accounting for environmental conditions. This allows for a more accurate reading in case of environmental changes, slight or otherwise. The quadratic form of Equation 2 describes the non-linear behavior of C_c as it relates to the load. The negative sign calibrates tension measurements from negative to positive. Finally, Equation 3 calibrates all weights in relation to a zero load or tare weight. This produces a precise net weight that should correspond with the actual weight.

$$C_c = 5E6 - [10(C_x - K \cdot C_{ref})] \quad (1)$$

$$\text{Load} = -(qA \cdot C_c^2 + qB \cdot C_c + qC) \quad (2)$$

$$\text{Weight} = \text{Load}_{x\text{ lbs}} - \text{Load}_{0\text{ lbs}} \quad (3)$$

Table 1: These are calibrated constants for each sensor as it applies to the equations above, and the reference frequency (Cref) used for environmental compensation as well as the frequency proportional to the load (Cx).

Label	Sensor 1	Sensor 2	Sensor 3
Serial Number	M081700881	M081700883	M081700887
K	8.254556E-01	5.291063E-01	6.676898E-01
qA	-1.627213E-12	-1.355136E-12	-1.308366E-12
qB	3.275433E-05	2.780857E-05	2.794689E-05
qC	-1.184411E+02	-9.107356E+01	-9.726224E+01
Cref (Hz)	228,540	239,340	235,670
Cx, 0lbs (Hz)	214,120	218,450	217,920

4 PARALLAX BASIC STAMP



Figure 3: The BASIC Stamp 2 connected to the Parallax 433 MHz RF Receiver

Next, we devise a feedback system to read, process, and send the data collected from the cable tensions to a user interface, using the BASIC Stamp 2, manufactured by Parallax. The BASIC Stamp 2 offers compatibility with other Parallax devices such as the wireless transmitter and receiver, reliable support/documentation, and a simple programming language, PBASIC 2.5.

4.1 Wireless Communication

To maintain the autonomy of the payload, we use wireless communication to send the tension data to the user PC, which controls the tugboats/robots. This wireless communication consists of the Parallax 433 MHz RF Transmitter and Receiver shown in Figure 3, which can easily be accessed and programmed by a BASIC Stamp 2. These modules use Linx RF chips, operating on 5V DC and high baud rates of 12k-19.2k. It also has a communication range of a couple hundred feet which is sufficient for this particular setup.

Lastly, in keeping with the cost-effective nature of this project, the RF transmitter and receiver are relatively inexpensive.

Each RF module is easily integrated, connecting to Pin 15 of the BASIC Stamp 2 and using two very simple commands in PBASIC 2.5: SEROUT (to send information) and SERIN (to receive information). The transmitter pulses at 2.4ms to sync with the receiver, first sending characters to denote the start of transmission and then sending data one byte at a time.

4.2 Shortcomings of the Basic Stamp 2, Revision J

4.2.1 Limited Processing and Memory Constraints

One shortcoming is that the BASIC Stamp 2 is a single-threaded microprocessor, meaning it can execute only one instruction at a time. As a result, we cannot interrupt the main program to count the frequencies of three different load cells simultaneously. Therefore, frequencies are counted one at a time for 100 milliseconds (ms), meaning it would take a minimum of 300 ms to just count frequencies on all three load cells. Moreover, the instruction set is executed at 4000 instructions per second, which is approximately 1 line of code every 250 microseconds (us). Assuming an additional 120 lines of code executed at 4000 instructions/sec, the entire program would run a total of 330 ms, looping about 3 times per second.

For the purposes of our experiment, updating the sensors three times per second is still an adequate sample rate for the measurements to be considered current and up-to-date. While counting less than 100ms would increase the amount of time the sensors are updated, the information would likely be inaccurate because the counted frequencies would return less significant digits. Alternatively, counting for more time would allow for very precise but out-of-date measurements. Our current setting of 100ms, therefore, is optimal for this experiment. This also means minimal coding is crucial to faster update rates.

Moreover, the BASIC Stamp 2 has limited memory which holds a maximum of 26 one-byte variables. This is a constraint that is difficult to navigate around because we use word-sized variables (equivalent to 2 bytes) to maintain 5 significant digits. Limited to only 13 word variables, our code cuts corners by using many constants and reusing word-sized variables.

4.2.2 Aliasing in the I/O pins

Another major shortcoming, specific to earlier revisions of the BASIC Stamp 2, is the inability to accurately count frequencies higher than 120 kHz even though the Stamp runs at a clock speed of 20 MHz. This presents a problem because the load cells output a maximum frequency (Cx and Cref) almost twice that of 120 kHz. One solution around the problem is to use IC prescalers to divide down the load cell frequencies enough to be counted. However, space and power constraints on both the proto-board and the prescalers make it very difficult to integrate.

Consequently, we test the effects of reading frequencies higher than 120 kHz through the I/O pins using a function generator, outputting a 5V DC square wave at 50% duty cycle with varying frequencies. Table 2 shows the results of this test. From 0 – 123 kHz, the output frequency follows the actual input frequency as Parallax claims. From 124 – 126 kHz, the output frequency fluctuates a lot since the input is too fast to be accurately counted. From 126 – 250 kHz, Equation 4 models the actual frequency while at frequencies higher than 250 kHz, Equation 5 models the actual frequency, where N is the quotient of Output Freq/250 kHz.

$$\text{Output Freq (kHz)} - 250 \text{ kHz} = \text{Actual Freq (kHz)} \quad (4)$$

$$\text{Output Freq (kHz)} + N * 250 \text{ kHz} = \text{Actual Freq (kHz)} \quad (5)$$

This phenomenon is known as aliasing which is an error where the input is faster than the twice the sampling rate. According to Parallax, the BASIC Stamp 2 (Revision J) can see transitions in

a pulse width of 4.16 micro-seconds or 2 transitions (one period) in a pulse width of 8.32 micro-seconds. A period width of 8.32 micro-seconds is approximately 120 kHz. However, the load cells operate in the frequency regime of 180-230 kHz, which means that they have smaller periodic widths; consequently, the BASIC stamp will only catch some transitions and not others. From the results in Table 2, though, we can accurately model the operating frequencies of the load cell using Equation 4, accounting for the aliasing. This is reflected in the transmitter source code found in Appendix A.

Actual Freq. (kHz)	Output Freq. (kHz)	Actual Freq. (kHz)	Output Freq. (kHz)
100.2	100.04	129.2	121.04
110.2	110.01	130.2	120.02
120.2	120.02	140.2	110.02
121.2	121.00	150.1	100.11
122.2	122.03	200.1	50.17
123.2	123.02	240.1	10.23
124.3	123.90-124.00	250.3	.14
125.2	123.04-124.27	260.4	10.04
126.2	123.90-124.05	300.5	50.04
127.2	123.01	340.5	90.07
128.2	120.01	373.5	123.02

Table 2: These are the results of inputting a 5VDC square wave at 50% duty cycle into the I/O pins of the BASIC Stamp 2.

5 EXPERIMENTAL RESULTS AND CONCLUSIONS

Preliminary results show wireless communication and readings from the load cell under different weights. Towers were also built to encase the load cells and to ensure proper tension alignment from the load cell to the tugboat. This is illustrated with the side, top, and bottom view of the tower in Figure 4. The tower structure, however, introduced another element of error into the feedback system caused by the dry friction between the cable and the hole at the top of the tower (see Figure 4). As a result, simulated tensions on the load cells using metric weights provide readings with fluctuating errors. For weights greater than or equal to about 500 grams, the load cell provides accurate readings with an error of about 10 percent. Meanwhile, weights less than 500 grams produce inaccurate results with errors greater than 10 percent. Some of the error is due to rounding errors in the program, found in Appendices A and B, since the BASIC Stamp does not process floating numbers. However, a big part of the error is due to the friction between the cable and the hole at the top of the tower; thus, reducing friction becomes an important design challenge.

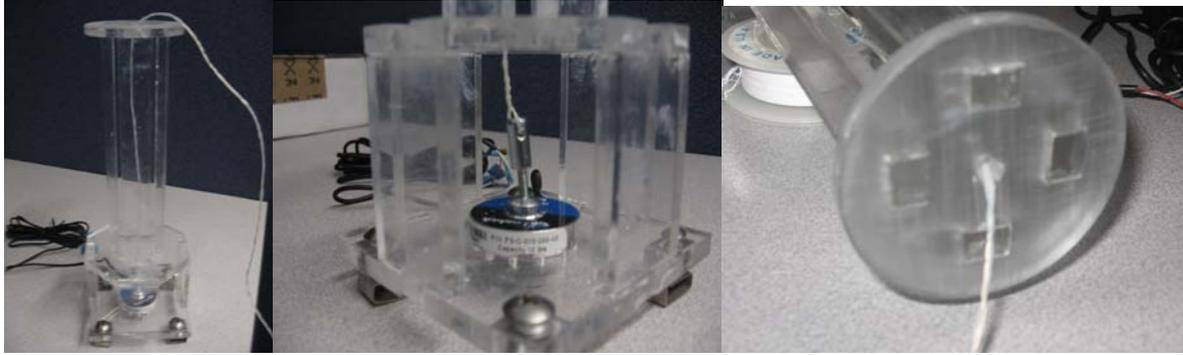


Figure 4: (Left) This tower was constructed to encase the load cell, and replaced existing towers on the payload. (Middle) The bottom view of the tower displays the iLoad Mini connected to the cable-threaded screw, allowing us to measure tension. (Right) The top view of the tower shows the cable coming through the small hole which is a major source of friction the heavier the load becomes.

Using an angler drill-bit, we scraped the sides of the hole and create a cone-shaped hole. We also twisted multiple cables together to increase the load capacity of the cable and to ensure that the Simulations show that this change in design helped, but did not eliminate a lot of the frictional errors. Likewise, adjusting the program code to account for these forces would require coding different error conditions to compensate for the friction. This would increase the program size, and cause the system to update less frequently.

6 FUTURE WORK

To avoid issues surrounding the BASIC Stamp 2, future works should use the BASIC Stamp 2p24 to replace earlier revisions of BASIC Stamp 2, such as revision j. This particular Stamp can easily be replaced in our setup, and has an increased program execution rate of 12,000 instructions per second and can count up to 300 kHz. The downside is that it draws significantly more current and power, and is more expensive than our current stamp.

Further improvements on our design might include testing different load cells such as the expensive strain gauges. These load cells might be less error prone and will eliminate the issue of aliasing and sampling because they output voltages proportional to the load. The only concern with this design would be to read the different voltage inputs. In order to accomplish this task, the voltages would have to be put through external Analog to Digital Converters (ADCs). Depending on space and power constraints, there might need to be a 3 to 1 multiplexer (MUX) with the output attached to an ADC. As a result, there is less space and power taken up.

Friction should also be taken into consideration in future designs. It comprised of a majority of the error in the load cell readings, even though we smoothed much of the surfaces in contact with the cable. Future designs should consider further both the hardware and software aspects of the problem, perhaps calibrating sensor constants to take into account frictional forces on the cable.

Lastly, future work should look into the actual integration of this feedback system into the overall control system for the towing of the robots. There needs to be an interface between the output of the BASIC Stamp 2 and the input for the towing robots, so that the robots can

automatically correct their speed and velocity based on the tension data. This crucial implementation would enhance the overall research in cooperative towing.

7 ACKNOWLEDGMENTS

I would like to thank Professor Vijay Kumar and Jim Keller who have afforded me the opportunity to participate in this research. I would also like to thank Soonkyum Kim, with whom I worked closely for the construction and completion of this project, as well as the GRASP Laboratory for allowing me to use their space and equipment. Lastly, I would like to acknowledge and thank Dr. Jan Van der Spiegel and the NSF-funded SUNFEST Program for their research support.

8 REFERENCES

- [1] Fink, Jonathan; Cheng, P.; Kumar, V., "Cooperative Towing with Multiple Robots." In ASME International Design Engineering Technical Conferences & Computers and Information in Engineering Conference.
- [2] Chaimowicz, L.; Sugar, T.; Kumar, V.; Campos, M.F.M., "An architecture for tightly coupled multi-robot cooperation," *Robotics and Automation, 2001. Proceedings 2001 ICRA. IEEE International Conference*, vol.3, no., pp. 2992-2997 vol.3, 2001
- [3] Kume, Y.; Hirata, Y.; Zhi-Dong Wang; Kosuge, K., "Decentralized control of multiple mobile manipulators handling a single object in coordination," *Intelligent Robots and System, 2002. IEEE/RSJ International Conference*, vol.3, no., pp. 2758-2763 vol.3, 2002
- [4] Kume, Y.; Hirata, Y.; Kosuge, K., "Coordinated motion control of multiple mobile manipulators handling a single object without using force/torque sensors," *Intelligent Robots and Systems, 2007. IROS 2007. IEEE/RSJ International Conference*, vol., no., pp.4077-4082, Oct. 29 2007-Nov. 2 2007
- [5] LoadStar Sensors Inc., iLoad Mini Load Cell, <http://loadstarsensors.com/iloadmini.html>

APPENDIX A: Source Code for Transmitter

```
' Final Transmitter.bs2
' {$STAMP BS2}
' {$PBASIC 2.5}
,
'Final Transmitter with iLoad Mini Test Code (BS2)
'This will track the tensions taken in from the iLoad Mini
,
'Sensor 1: M081700881
'Sensor 2: M081700883
'Sensor 3: M081700887

CTRL1 PIN 6 'Sensor 1
FREQ1 PIN 7
CTRL2 PIN 8 'Sensor 2
FREQ2 PIN 9
CTRL3 PIN 10 'Sensor 3
FREQ3 PIN 11

SEND PIN 15

qAC VAR Word
qBC VAR Word
LoadC VAR Word
Cc VAR Word
Cref VAR Word
Cx VAR Word
CrefF VAR Word
CxF VAR Word
Tare1 VAR Word
Tare2 VAR Word
Tare3 VAR Word
Weight VAR Word

K1 CON 54097 '.8254556*65536 = 54097
K2 CON 34676 '.5291063*65536 = 34676
K3 CON 43758 '.6676898*65536 = 43758

qA1 CON 10664 '.1627213*65536 = 10664
qA2 CON 8881 '.1355136*65536 = 8881
qA3 CON 8575 '.1308366*65536 = 8575

qB1 CON 21466 '.3275433*65536 = 21466
```

```

qB2  CON  18225 '.2780857*65536 = 18225
qB3  CON  18315 '.2794689*65536 = 18315

qC1  CON  11844 'Decimal 118.44
qC2  CON  9107  'Decimal 91.07
qC3  CON  9726  'Decimal 97.26

```

Main:

```

Tare1 = 0
Tare2 = 0
Tare3 = 0

```

```

GOSUB ReadingSensor1
Tare1 = LoadC-3
GOSUB ReadingSensor2
Tare2 = LoadC-3
GOSUB ReadingSensor3
Tare3 = LoadC-3

```

DO

```

GOSUB ReadingSensor1
GOSUB ReadingSensor2
GOSUB ReadingSensor3
LOOP

```

END

```
' (A*65536)**B = A*B
```

ReadingSensor1:

```

HIGH CTRL1
PAUSE 5
COUNT FREQ1, 100, CxF
PAUSE 5

```

LOW CTRL1

```

PAUSE 5
COUNT FREQ1, 100, CrefF
PAUSE 5

```

```

Cx = 25000 - CxF      'Correct the frequency since we can't read above 120 kHz
Cref = 25000 - CrefF

```

```
Cc = (50000 - Cx + (Cref**K1))      'Cc = X,XXX,X00
```

```
qAC = (Cc**qA1)**((Cc/100)*66)
qBC = Cc**qB1
```

```
LoadC = qAC - qBC + qC1      'LoadC = 000X.XX
Weight = LoadC - Tare1
```

```
DEBUG SDEC LoadC, CR
DEBUG SDEC Weight, CR
```

```
PULSOUT SEND, 1200          'Sync pulse for the receiver
SEROUT SEND, 16468, ["ABC!", Weight.HIGHBYTE, Weight.LOWBYTE]
```

```
RETURN
```

```
ReadingSensor2:
```

```
HIGH CTRL2
PAUSE 5
COUNT FREQ2, 100, CxF
PAUSE 5
```

```
LOW CTRL2
PAUSE 5
COUNT FREQ2, 100, CrefF
PAUSE 5
```

```
Cx = 25000 - CxF      'Correct the frequency since we can't read above 120 kHz
Cref = 25000 - CrefF
```

```
Cc = (50000 - Cx + (Cref**K2))      'Cc = X,XXX,X00
qAC = (Cc**qA2)**((Cc/100)*66)
qBC = Cc**qB2
```

```
LoadC = qAC - qBC + qC2      'LoadC is negative for tension so all signs are switched!
Weight = LoadC-Tare2
```

```
PULSOUT SEND, 1200          'Sync pulse for the receiver
SEROUT SEND, 16468, ["DEF!", Weight.HIGHBYTE, Weight.LOWBYTE]
```

```
RETURN
```

```
ReadingSensor3:
```

```
HIGH CTRL3
PAUSE 5
COUNT FREQ3, 100, CxF
PAUSE 5
```

LOW CTRL3

PAUSE 5

COUNT FREQ3, 100, CrefF

PAUSE 5

Cx = 25000 - CxF 'Correct the frequency since we can't read above 120 kHz

Cref = 25000 - CrefF

' DEBUG ? Cx, CR

' DEBUG ? Cref, CR

Cc = (50000 - Cx + (Cref**K3)) 'Cc = X,XXX,X00

qAC = (Cc**qA3)**((Cc/100)*66)

qBC = Cc**qB3

LoadC = qAC - qBC + qC3 'LoadC is negative for tension so all signs are switched!

Weight = LoadC-Tare3

PULSOUT SEND, 1200 'Sync pulse for the receiver

SEROUT SEND, 16468, ["GHI!", Weight.HIGHBYTE, Weight.LOWBYTE]

RETURN

APPENDIX B: Source Code for Receiver

'Final Receiver.bs2

'{\$\$STAMP BS2}

'{\$PBASIC 2.5}

,

'Final Receiver with iLoad Mini Test Code (BS2)

'This will track the tensions taken in from the iLoad Mini

,

'Sensor 1: M081700881

'Sensor 2: M081700883

'Sensor 3: M081700887

RECEIVE PIN 15

Load1 VAR Word

Load2 VAR Word

Load3 VAR Word

Main:

DO

SERIN RECEIVE, 16468, [WAIT("ABC!"), Load1.HIGHBYTE, Load1.LOWBYTE]

SERIN RECEIVE, 16468, [WAIT("DEF!"), Load2.HIGHBYTE, Load2.LOWBYTE]

SERIN RECEIVE, 16468, [WAIT("GHI!"), Load3.HIGHBYTE, Load3.LOWBYTE]

DEBUG "Sensor 1: ", DEC Load1 DIG 4, DEC Load1 DIG 3, DEC Load1 DIG 2, ".", DEC
Load1 DIG 1, DEC Load1 DIG 0, " lbs", CR

DEBUG "Sensor 2: ", DEC Load2 DIG 4, DEC Load2 DIG 3, DEC Load2 DIG 2, ".", DEC
Load2 DIG 1, DEC Load2 DIG 0, " lbs", CR

DEBUG "Sensor 3: ", DEC Load3 DIG 4, DEC Load3 DIG 3, DEC Load3 DIG 2, ".", DEC
Load3 DIG 1, DEC Load3 DIG 0, " lbs", CR

LOOP

END

GROOVENET3: A SCALABLE TRAFFIC SIMULATOR FOR CONGESTION PROBING AND PREDICTION

NSF Summer Undergraduate Fellowship in Sensor Technologies
Uchenna Kevin Anyanwu (Computer Engineering) – California State University at San Jose
Advisor: Rahul Mangharam

Abstract

Traffic congestion is a huge problem for developed countries such as the United States. Severe traffic congestion that can slow traffic up to several miles is common for everyday drivers in United States. Before the problem can be solved, government agencies and private agencies need reliable data to help understand traffic congestion. The goal of this research is to enable GrooveNet, a vehicle-to-vehicle simulation program, to record and analyze large amounts of historical traffic data, quickly and efficiently, and provide a playback capability on historical data. Methods used to develop this robust capability were proposed by Kanul, et al, at Carnegie Mellon University. Using historical traffic data, GrooveNet displays congestion over time and proposes a less congested route for a simulated vehicle. By gathering historical data, the upcoming version of GrooveNet will have a large knowledge repository that will ultimately give way to sophisticated machine learning and prediction algorithms that allow GrooveNet to navigate vehicles from origin to destination, safely and quickly, thus decreasing traffic congestion.

1. Introduction

This research proposes a vehicle-to-vehicle simulator called GrooveNet which will not only retrieve historic information generated by vehicles, but also, in real-time, produce information on the current state of traffic, using efficient and fast real-time congestion probing techniques.

GrooveNet is a vehicle-to-vehicle hybrid (allows communication between real and virtual vehicles) simulation program. The graphic below shows how two vehicles propagate information between one another using assistance from virtual vehicles. No matter how far two real vehicles are apart from each other, GrooveNet will find the fastest path that will allow two real vehicles to communicate (*Figure 1*). GrooveNet allows vehicles to communicate time-critical information, such as air-bag deployment, vehicle malfunction, and more, as means to promote safety and awareness for drivers within a bounded, topological region. GrooveNet is an open-source research project spearheaded by Dr. Rahul Mangharam. GrooveNet currently simulates hundreds of vehicles on any street map in the US. Each vehicle can be equipped with a model. For example, the Random Walk Model allows a particular vehicle to approach an intersection and randomly choose which way to turn at the intersection.

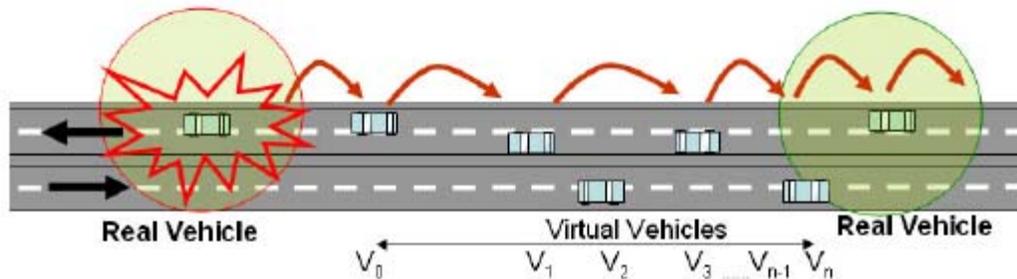


Figure 5

In 2007, Kanual Shah, Raj Rajkumar, Ph.D, and Rahul Mangharam, Ph.D, presented several models that will enable vehicles in GrooveNet to probe traffic conditions and make reasonable decisions about which path to take in order to minimize distance traveled and time to reach a destination. Kanual, et al, proposed two types of models, which characterizes the behavior of each vehicle. These modes are Active congestion probing and Passive congestion probing. Active congestion probing includes passive congestion control, which enables vehicles to send and receive traffic information from a central server, and active congestion control, which enables vehicles to send and receive traffic information from each other. Passive congestion probing enables each vehicle to maintain data about paths taken and combine that data with transportation data to create trend chart or historical information.

This project has several goals. The first is to understand current traffic generation, recording and playback capabilities of GrooveNet 2.0. Second, develop a mechanism for recording historic speed and traffic density data. Third, develop a mechanism for recording traffic accidents. Fourth, define an efficient binary format to record and playback historic traffic data. Fifth, integrate the above within GrooveNet 3.0. Lastly, use historic traffic data to predict traffic congestion and determine alternate less-congested routes in GrooveNet 3.0. The overall goal is to demonstrate traffic congestion development under normal conditions and under traffic incidence in the simulator, GrooveNet.

2. Background

Traffic congestion is a major problem in fast, developing countries. In the United States, there is a need to minimize traffic congestion and increase safety for all drivers. According to U.S. Mobile Report congestion is getting worse in America's 437 urban areas [5]. Unfortunately, there is no one treatment for traffic congestion.

2.1 Vehicle to Vehicle Simulator

Unlike traffic simulators that handle traffic flow using a queuing perspective, GrooveNet was developed to enable real-time communication between real vehicles, simulated vehicles and between real and simulated vehicles. Incorporating GPS and Wireless networks allows GrooveNet vehicles to obtain a holistic view of traffic flow and make intelligent path predictions based on that current state of traffic. The following figure shows two different vehicles communicating traffic information. Vehicle status from a real vehicle can be broadcast to other vehicles (real and simulated) within a wide range. (Figure 2).

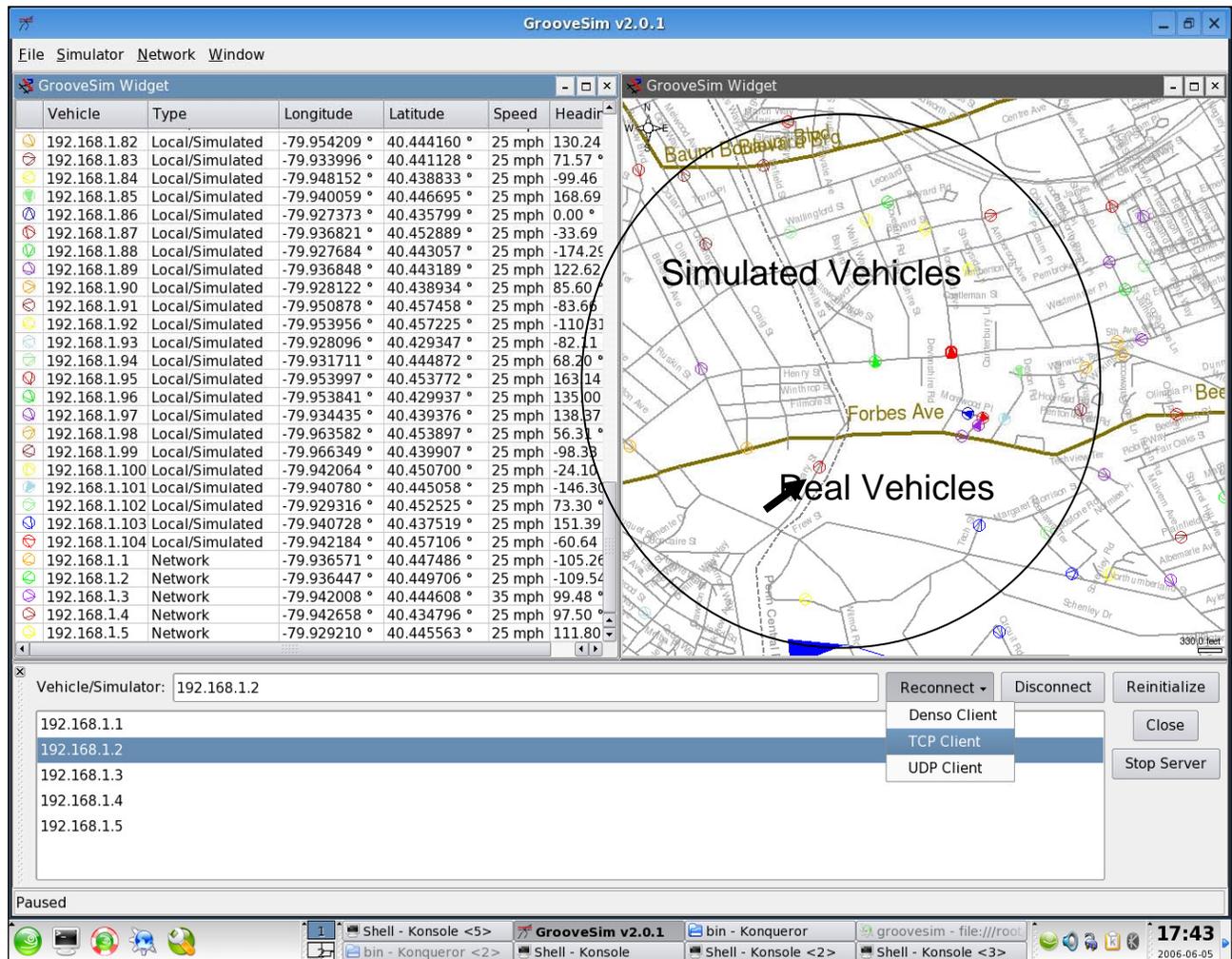


Figure 2

2.2 Active congestion probing

Active congestion probing is important to vehicles that need to determine the optimal time and distance to a location in real-time. This type of congestion probing provides every vehicle the ability to communicate and work together in order to minimize traffic congestion. The gain in this technique is that accurate data about traffic congestion can be achieved.

2.3 Passive congestion probing

Passive congestion probing combines route information and historical transportation data, located in a database, from each vehicle to produce a trend chart that can be used by vehicles to make intelligent decisions.

2.4 Hybrid congestion probing

The Hybrid congestion approach combines active and passive congestion approaches. First, the idea is to obtain active congestion information, and then, if there isn't sufficient active congestion information, GrooveNet will use the historic/trend data.

2.5 Traffic Models

Various models have been created to understand traffic flow using historic data. These models are classified as Macroscopic, Microscopic, and Mesoscopic models. The Macroscopic model is similar to water flowing through a pipe. Microscopic model characterizes individual vehicles with detailed behavior. Mesoscopic model defines individual vehicles with aggregate behavior. One macroscopic model of interest is “The Lighthill, Whithman and Richards model” (Figure 2). The model can be written as two forms follows:

$$\frac{n(x)\partial C(x,t)}{\partial t} + \frac{\partial q(x,t)}{\partial x} = 0 \quad \frac{\partial K}{\partial t} + \frac{\partial}{\partial x} Q_r(K, x) = 0$$

Figure 6

This is expressed as the sum of partial derivatives, where $C(x,t)$ means traffic density, $n(x)$ is the number of lanes at position x , and $q(x,t)$ is the traffic flow of vehicles per hour at location x at time t . “The Lighthill, Whithman and Richards model” is called the Law of Conservation of vehicles in traffic.

3 Current GrooveNet HistoricalData and Algorithms

GrooveNet has several data structures for extracting, saving, and printing historical data, but is not integrated with the overall system. In order to enable GrooveNet to record and play back historical data, the primary goal is to reconstruct and complete the data structures and functions implemented by Kanul.

3.4 HistoricalData Struct

```
typedef struct HistoricalData {
    int RecordId , SegmentSpeed;
    int speedL[48], speedR[48];
}HistoricalData;
```

The HistoricalData data structure was implemented to organize variables in memory that pertains to historical data. The importance of the above implementation is to serve as storage for historical information, which will be written to a file. The variables are as follows: first, two integer arrays of forty eight segment (lane) speeds; second, an integer for RecordId, which is an identification for a road on the map.

HistoricalInfo Class

```
Class HistoricalInfo{
public:
    //Constructor and Destructor functions
    int Load ( ...);
    inline bool IsLoaded ( ... ) const
    void Save( ... );
    void Setspeed( ... );
    int Extractinfo( ... );
    int GetRecIndex( ... );
    void LoadHistoricalInfo (...);
private:
    bool m_Loaded;
    std::vector<HistoricalData> recSpeed;
}
```

The above HistoricalInfo class is the current baseline for analyzing and displaying traffic data in GrooveNet. The goal of this class is to give the software an efficient mechanism for handling historical information. The Load function, working together with ExtractInfo and isLoaded function, is used to load historical information—a text file in binary format—from the hard drive to memory. The Save function, which uses Setspeed, GetRecIndex, and GetIndex, writes traffic speed data to a file in binary efficient format. The private members are m_Loaded, which allows GrooveNet to know if the system is safe for processing. The private dynamic array of HistoricalData structures, which is an STL (standard template library) implementation of a dynamic array, is used to store structures of historical datum in memory, which will be written to a file on the hard drive.

4. Revised GrooveNet HistoricalData and Algorithms

Several revisions need to be made in Kanual's implementation of the HistoricalData class, structures and accompanying algorithms.

First, the HistoricalData structure declaration needed to be changed in order to reflect a simple database that contains essential traffic information. SpeedL and speedR variables were clearly not feasible because writing at most 80 speed information to a file in disk would hoard a lot of disk space. Lieu of using SpeedL and SpeedR, the best approach was to obtain average speed of each segment. RecordId variable was retained. Additional variables were added to the HistoricalData struct. The revised structure is as follows:

```
typedef struct HistoricalData {
    int RecordId;
    int SpeedDirection;
    int SegmentSpeed;
    int NumVehicles;
}HistoricalData;
```

Second, the HisticalInfo class structure needed to be revamped. This class is important to efficiently getting historical data from GrooveNet to a file and reading from it. When redesigning the class, it was important to keep the overall functionality as is. The first change was to delete all unwanted functions such as Setspeed, Getspeed, and GetRecIndex. The second change was to add important functions such as HistoricData_Read, Historic_Count, h_count.

```
class HistoricalInfo
{
public :
    HistoricalInfo();
    ~HistoricalInfo();
    int Load(...);
    inline bool IsLoaded() const
    {
        return m_bLoaded;
    }
    void Save(...);
    int Extractinfo(...);
    int HistoricData_Count(... );
    void HistoricData_Read(...);
    void LoadHistoricalinfo();
    std::list<unsigned int> returnListRec();
private:
    bool m_bLoaded;
    std::vector<HistoricalData> recSpeed;
    std::list<unsigned int> records;
    int h_count;
```

```

    QString h_strFilename;
};

```

The most important function of the HistoricalInfo class structure is the Save and LoadHistoricInfo function. The Save function allows GrooveNet to save segment identification numbers that have vehicles moving less than 80 percent of the road speed every 5 seconds of the simulation time. In order to develop this algorithm, a function to capture data every five seconds needed to be implemented. Within that function, a call to the Save must be made. The Save function iterates through all segments, finding all vehicles in the same segment, taking the average speed of all vehicles in the segment, and saving all average speed segment ids that are less than 80 percent of the road speed. All data is saved on to multiple master files and an index file. The master file has all historical data, separated by a delimiter, which divides each historical data entry.

The user programmer must notice that each row is 5 seconds of recording time. When played by the simulator, each row will be displayed as 1 second.

5. Playback Feature

A playback feature requires several buttons to be implemented into the current user interface. Qt graphics library was utilized to develop the following buttons: PLAY, FASTFORWARD, REWIND, PAUSE, and STOP (figure 3). These buttons are used to control the Map Visual, which displays the data by the proposed HistoricalInfo Class. The following screen shot displays the new user interface of GrooveNet:

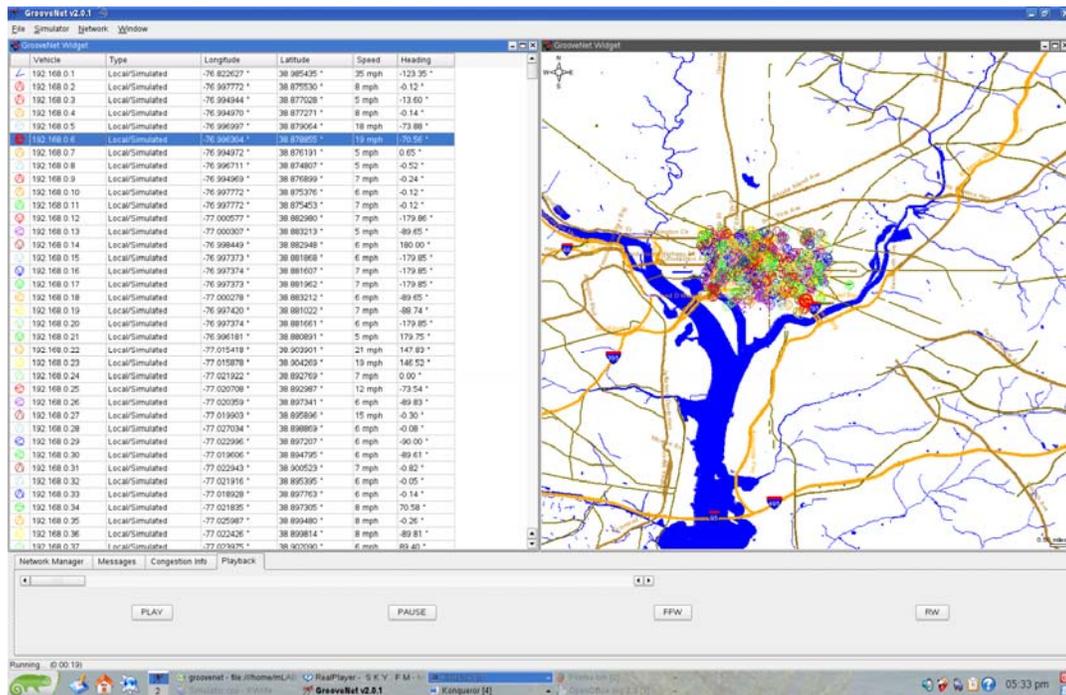


Figure 3

A QPlayback data structure is derived from QWidget class in order to create the above features. The constructor class, which creates the play, pause, fast-forward, and rewind buttons, requires that we define signals and slots. Signals—play, pause, rewind, stop, fast-forward (not displayed)—activate messages that are sent to the MapWidget, which has a slot implementation for receiving messages.

5.1 Play Function

The play function executes an algorithm that analyzes both an index and binary files. As described above, the index file contains the size in bytes that should be read by GrooveNet in order to obtain a data set. Each line represents one second of the recording. When this data has been read into memory, GrooveNet can open the specific binary data file, then reads one data set and displays the traffic information on the screen.

5.2 Fast-forward Function

The fast-forward function executes the same functionality as the play function instead it runs much faster. The programmer can adjust the speed by changing the variable global variable #defines in the header file of QPlayback.

5.3 Stop Function

The stop function resets the current position to the beginning of the historical file

5.4 Pause Function

The stop function holds the current position of the play button

6. Discussion and Conclusion

GrooveNet facilitates understanding dynamics of traffic congestion and alleviating traffic prevention. GrooveNet has the ability to simulate a wide array of traffic congestion scenarios on variety of maps. Enabling GrooveNet to record and play back historical data allows the user to experience the dynamics of traffic build-up over time and space. When using the Traffic recording functionality, GrooveNet compiles data as it is being produced and sends that data into a repository or data base, on the hard drive. At any time in the future, the user is able to either analyze the raw data dump or re-load it to GrooveNet for playback purposes. The user is able to playback the file as if it were a video and to view progression of simulated traffic overtime.

7. Recommendation

An index file method can be used to implement the rewind button in the playback functionality. The index is basically a catalog of how to find specific information in a huge data file. The index method can be applied to GrooveNet by writing a set of algorithms that writes to a separate file,

printing the size in bytes of each data record in the historical file. This file is read in memory by GrooveNet, allowing it to traverse backward through the historical file.

The next step is to incorporate the playback feature in the parallel version of GrooveNet. This can be done easily using the proposed algorithms; however, there is always room for optimization for the new version. Also, since the parallel version of GrooveNet uses less processor power, GrooveNet can spend more time processing greater chunks of data for a smoother playback capability.

8. Acknowledgements

I would like to thank my advisors, Dr. Rahul Mangharam and Dr. Jan Van der Spiegel of the University of Pennsylvania, for their support and advice. I would also like to thank all Miroslav Pajic and Danny Lustig of the University of Pennsylvania for their patience and close collaboration with this project. I would also like to thank the National Science Foundation for providing deserving undergraduates, like myself, a chance to be a part of cutting edge research at the University of Pennsylvania.

9. References

- [1] K. Shah, R. Rajkumar, and R. Mangharam, "Real-time traffic congestion probing Using Vehicle to Vehicle Networks," M.S. thesis, Carnegie Mellon University, Pennsylvania, PA. December 2007.
- [2] R. Mangharam, D. S. Weller, D. D. Stancil, R. Rajkumar, J. S. Parikh, GrooveSim: A Topography-Accurate Simulator for Geographic Routing in Vehicular Networks *Proceedings of Second ACM International Workshop on Vehicular Ad hoc Networks (Mobicom/VANET 2005)*. Cologne, Germany. September 2005.
- [3] R. Mangharam, D. S. Weller, R. Rajkumar, Priyantha Mudalige and Fan Bai, "GrooveNet: A Hybrid Simulator for Vehicle-to-Vehicle Networks," presented at *Second International Workshop on Vehicle-to-Vehicle Communications (V2VCOM)*, San Jose, USA. July 2006.
- [4] D. Schrank and T. Lomax, "The 2007 URBAN Mobility Report," September 2007. [Online]. Available: http://financecommission.dot.gov/Documents/Background%20Documents/mobility_report_2007_wappx.pdf [Accessed: July 12, 2008]
- [5] J.P.Lebacque, J.B. Lesort, F.Giorgi. "Introducing Buses into First-Order Macroscopic Traffic Flow Model," *Transportation Research Record: Journal of the Transportation Research Board*, vol.1644/1998, pp. 70- 79, January 2007.

10. APPENDIX: SOURCE CODE

```
#ifndef _QPLAYBACK_H
#define _QPLAYBACK_H

#include <qwidget.h>
#include <qmutex.h>
#include <qlayout.h>
#include <qlistbox.h>
#include <qlabel.h>
#include <qlineedit.h>
#include <qtextedit.h>
#include <qpushbutton.h>
#include <qscrollbar.h>
#include "MapDB.h"
#include "ModelMgr.h"
#include "MapVisual.h"
#include <qfile.h>

class QLabel;
class QLineEdit;
class QPushButton;
class QPopupMenu;
class QListBox;
class QScrollBar;

class QPlayback : public QWidget
{
    Q_OBJECT
public:
    QPlayback(QWidget * parent = NULL, const char * name = 0, Qt::WFlags f = 0);
    virtual ~QPlayback();
    virtual void SetPlaybackInitialized(bool m_bLoaded);
protected slots:
    void signalPlay();
    void SET_TIMER_PLAY();
    void SET_TIMER_STOP();
    void SET_TIMER_FAST();
    void SET_TIMER_PAUSE();
    void SET_TIMER_SCROLLER();
    void moveScroller();
    void signalMoved(int);
protected:
    QHBoxLayout * p_layout;
    QPushButton * p_play;
    QPushButton * p_fforward;
```

```

QPushButton * p_rewind;
QPushButton * p_pause;
QPushButton * p_stop;
QWidget * p_listClients;
QScrollBar * p_scroll;
bool stop;
MapVisual * ptrModelV;
Model* ptrModel;
QString strFilename;
QTextStream reader;
QTimer * play_time;
int scroll_value;
};

#endif
#include <qlayout.h>
#include <qlabel.h>
#include <qlineedit.h>
#include <qpushbutton.h>
#include <qpopupmenu.h>
#include <qlistbox.h>
#include <qwidget.h>
#include <qtimer.h>
#include <qdialog.h>

#include "QPlayback.h"
#include "HistoricalInfo.h"
#include "MainWindow.h"
#include "QMapWidget.h"
#include "Visualizer.h"
#include "Simulator.h"

#define FFW 8

QPlayback::QPlayback(QWidget * parent, const char * name, Qt::WFlags f)
: QWidget(parent, name, f)
{

QHBoxLayout * button_p_layout;

button_p_layout = new QHBoxLayout(this,8,8,"layout"); //declare layout for buttons

p_play = new QPushButton("PLAY", this);
p_play->setEnabled(false); //disable the button

connect(p_play, SIGNAL(clicked()), this, SLOT(signalPlay()));

```

```

p_stop = new QPushButton("STOP", this);
p_stop->setEnabled(false);

p_fforward = new QPushButton("FFW", this);
p_fforward->setEnabled(false);

p_rewind = new QPushButton("RW", this);
p_rewind->setEnabled(false);

p_pause = new QPushButton("PAUSE", this);
p_pause->setEnabled(false);

p_scroll = new QScrollBar(Qt::Horizontal, this);
p_scroll->setEnabled(false);
p_scroll->resize(1000, p_scroll->height()); //resize the scroll bar

play_time = new QTimer(this);

button_p_layout->addWidget(p_play, 0, Qt::AlignCenter); //add the widget to the layout
button_p_layout->addWidget(p_pause, 0, Qt::AlignCenter);
button_p_layout->addWidget(p_fforward, 0, Qt::AlignCenter);
button_p_layout->addWidget(p_rewind, 0, Qt::AlignCenter);
button_p_layout->addWidget(p_stop,0, Qt::AlignCenter);

connect(p_play, SIGNAL(clicked()), this, SLOT(SET_TIMER_PLAY()));
connect(p_stop, SIGNAL(clicked()), this, SLOT(SET_TIMER_STOP()));
connect(p_fforward, SIGNAL(clicked()), this, SLOT(SET_TIMER_FAST()));
connect(p_pause, SIGNAL(clicked()), this, SLOT(SET_TIMER_PAUSE()));
connect(p_play, SIGNAL(clicked()), this, SLOT(SET_TIMER_SCROLLER()));
connect(play_time, SIGNAL(timeout()), this, SLOT(signalPlay()));
connect(play_time, SIGNAL(timeout()), this, SLOT(moveScroller()));
//connects a timer to the play slot; emits timeout signals every 1 second

p_scroll->setRange(0,51);
p_scroll->setSteps(1, 0);

connect(p_scroll, SIGNAL(sliderMoved(int)), this, SLOT(signalMoved(int)));
}
void QPlayback:: signalMoved(int temp)
{
    //get the current value of the slider when press
    if ( scroll_value < p_scroll->value( ) )
    {
        signalPlay(); //read forward in the data file
    }
    else

```

```

        {
            //read backward in the data file
        }
    }
void QPlayback:: SetPlaybackInitialized(bool m_bLoaded)
{
    if(m_bLoaded)
    {
        p_play->setEnabled(true);
        p_fforward->setEnabled(true);
        p_rewind->setEnabled(true);
        p_pause->setEnabled(true);
        p_scroll->setEnabled(true);
        p_stop->setEnabled(true);
    }
    else
    {
        p_play->setEnabled(false);
        p_fforward->setEnabled(false);
        p_rewind->setEnabled(false);
        p_pause->setEnabled(false);
        p_scroll->setEnabled(false);
        p_stop->setEnabled(false);
    }
    stop = false;
    g_pMapDB->Path4=true;
    g_pSimulator->m_ModelMgr.GetModel("MapVisual0", ptrModel);
    if(ptrModel)
    {
        ptrModelIV = (MapVisual *)ptrModel;
    }
    printf("CONSTRUCTOR CALLED \n");
}
QPlayback::~~QPlayback()
{
    delete p_pause;
    delete p_play;
    delete p_rewind;
    delete p_fforward;
    delete p_scroll;
    delete p_layout;
    p_pause = p_play = p_rewind = p_fforward = NULL;
}
void QPlayback:: SET_TIMER_PLAY()
{

```

```

        play_time->start(1000, FALSE);
    }
void QPlayback:: SET_TIMER_SCROLLER()
{
    play_time->start(1000, FALSE);
}
void QPlayback:: SET_TIMER_PAUSE()
{
    play_time->stop();
    scroll_value=p_scroll->value();
    printf("The value of the slider is: %i\n", scroll_value);
}
void QPlayback:: SET_TIMER_FAST()
{
    int msec_time;
    msec_time= 1000/FFW;
    play_time->start(msec_time , FALSE);
}

void QPlayback:: SET_TIMER_STOP()
{
    play_time->stop();
    g_pMainWindow->m_pHist->file->reset();
    p_scroll->setValue(0);
}
void QPlayback:: signalPlay()
{
    g_pMainWindow->m_pHist->HistoricData_Read(); //read new data
    ptrModelV->recenterVal();
}
void QPlayback:: moveScroller()
{
    //find the the current position
    //add one to the current position
    //store the new poision
    p_scroll->setValue(p_scroll->value() + 1);
}

#ifndef _HISTORICALINFO_H
#define _HISTORICALINFO_H

#include "CarModel.h"
#include "MapDB.h"
#include "CarRegistry.h"
#include "StreetSpeedModel.h"

```

```

#include "QMapWidget.h"
#include "MapVisual.h"
#include "Visualizer.h"
#include "QPlayback.h"

#include <qfile.h>
#include <qtextstream.h>

#define PARAM_MODEL "MODEL"
#define PARAM_TYPE "TYPE"
#define PARAM_DEPENDS "DEPENDS"

typedef struct HistoricalData {
    int CongId;
    int RecordId;
    int SpeedDirection;
    int SegmentSpeed;
    int NumVehicles;
}HistoricalData;

class HistoricalInfo
{
public :
    HistoricalInfo();
    ~HistoricalInfo();

    int Load(const QString & strFilename);
    int Getspeed(int RecID, int direction, struct timeval CurTime);
    int Extractinfo(const QString & strLine, std::vector<HistoricalData > & recInfo);
    int GetIndex(struct timeval CurTime);
    int HistoricData_Count(QTextStream * reader );
    int GetRecIndex(int RecID);
    void Save(QTextStream & writer, double & timeStamp);
    void Setspeed(int RecID, int direction, int speed, struct timeval CurTime);
    void HistoricData_Read();
    void LoadHistoricalinfo();
    void ClearList();
    void returnListRec(std::list<unsigned int> &, std::list<unsigned int> &,
std::list<unsigned int> &, std::list<unsigned int> &);

    std::vector<HistoricalData> recSpeed;
    std::list<unsigned int> records1;
    std::list<unsigned int> records2;
    std::list<unsigned int> records3;

```

```

std::list<unsigned int> records4;

QFile * file;
QString h_strFilename;
QTextStream * reader;

bool m_bLoaded;
int countStars;
int h_count;
};
#endif

#include "HistoricalInfo.h"
#include "MainWindow.h"

#include <qfile.h>
#include <qtextstream.h>
#include <qiodevice.h>
#define MAXSIZE 60

HistoricalInfo::HistoricalInfo()
{
    m_bLoaded = false;
    countStars = 0;
}

HistoricalInfo::~HistoricalInfo()
{
    recSpeed.clear();    //might not be needed
}
int HistoricalInfo::Load(const QString & strFilename)
{
    file = new QFile(strFilename);    //create new file, attached to a file name

    reader = new QTextStream(file);    //point reader to the file

    if (!file->open(IO_ReadOnly | IO_Translate))    //test that file is opens successfully
    {
        errno = ENOENT;    //exit if not opened correctly
        return 0;
    }

    reader->setDevice(file); //set device

```

```

        if (g_pMainWindow != NULL && g_pMainWindow->m_pLblStatus != NULL) //test
main window pointer
            g_pMainWindow->m_pLblStatus->setText("Loading Historical Info...");

        if(!m_bLoaded)        //if playback tab is not loaded
        {
            g_pMainWindow->m_pPlayback = new QPlayback(g_pMainWindow-
>tabsContainer, "playback");
            //create new playback tab
            g_pMainWindow->tabsWidget->addTab(g_pMainWindow->m_pPlayback,
QString("Playback"));
            //add the playback table to the window
            g_pMainWindow->m_pPlayback->SetPlaybackInitialized(true);
            //enable playback tab
            m_bLoaded = true;    //set playback tab loaded
        }
        return 1;
    }
}
void HistoricalInfo::HistoricData_Read()
{
    QString line;
    while (!(line = reader->readLine()).isNull()) //read lines from the file until there is no
more data
    {
        if (line.isEmpty())    //if line is empty then go to the next line
            continue;
        if(line == "*")        //if line is a star(delimiter) then ...
        {
            //we have found a whole data set then break
            break;
        }
        if(!Extractinfo(line, recSpeed)) //we can parse the line
            printf("error occured while extracting historical data\n");    //or error out!
    }
    LoadHistoricalinfo(); //load segment data into a list
}
int HistoricalInfo::HistoricData_Count(QTextStream * reader)
{
    QString line;
    int h_count = 0;
    while (!(line = reader->readLine()).isNull())
    {
        if (line.isEmpty())
            continue; //skip the next if, go to while
        if(line == "*")
        {
            ++h_count; //track the number of historical data sets in file
        }
    }
}

```

```

        }
    }
    if(h_count>0)
        return h_count;
    else return 0;
}
int HistoricalInfo::Extractinfo(const QString & strLine, std::vector<HistoricalData > & recInfo)
{
    //parse one line of input and dump it into respective element in to a vector of historical
data
    HistoricalData temp;
    QStringList list = QStringList::split(" ",strLine);
    temp.CongId = list[0].toInt();
    temp.RecordId = list[1].toInt();
    temp.SegmentSpeed = list[2].toInt();
    temp.SpeedDirection = list[3].toInt();
    //printf("%d, %d \n",temp.CongId, temp.RecordId,temp.SegmentSpeed,
temp.SpeedDirection); debugging
    recInfo.push_back(temp); //save data into datastructure
    return 1; //return sucess
}
void HistoricalInfo::Save(QTextStream & writer, double & fTime)
{
    std::map<in_addr_t, CarModel* >::iterator iterCar; //declare iterators for maps and a list
    std::map<in_addr_t, CarModel *>::iterator iterCar1;
    std::list<int>::iterator iterSegment;

    std::map<in_addr_t, CarModel *> * pCarRegistry = g_pCarRegistry->acquireLock();
//not sure if needed

    std::list<int> segment; //
    std::list<int> recSizes;
    std::list<int>::iterator iterSizes;

    MapRecord * psRec;
    StreetSpeedModel speedLim;
    short roadSpeedLimit = 0.0;
    double fSeconds(0);
    float mSec(0);
    mSec=modf(fTime, &fSeconds);
    int iSeconds = (int)fSeconds;

    for (iterCar = pCarRegistry->begin(); iterCar != pCarRegistry->end(); ++iterCar)
    {
        unsigned int record = iterCar->second->GetCurrentRecord();

```

```

bool foundSeg = false;
//search list of used segments
for(iterSegment= segment.begin() ; iterSegment != segment.end());
++iterSegment)
{
    if( *iterSegment == record)
    {
        foundSeg=true;    //segment found in list
    }
}

if(!foundSeg) //segment not in list
{

    segment.push_back(record); //save segment in list

    unsigned int totalspeed = 0;

    unsigned int count = 0;

    short roadAverage = 0;

    bool SpeedL1 =0, SpeedL = 0;

    short direction1 = 0;

    short direction = iterCar->second->GetCurrentDirection();

    int vecSpeed = iterCar->second->GetCurrentSpeed();

    roadSpeedLimit = GetRoadSpeed(record);

    if(direction>= -4500 && direction < 13500)
    {
        SpeedL = true;
    }
    if(direction< -4500 || direction >= 13500)
    {
        SpeedL = false;
    }
    for (iterCar1 = pCarRegistry->begin(); iterCar1 != pCarRegistry->end(); +
++iterCar1)
    {
        if ( iterCar1->second->GetCurrentRecord() == record ) //found a
vehicle                                     on the same segment
    {

```

```

        direction1 = iterCar1->second->GetCurrentDirection();
        if((direction1 >= -4500 && direction1 < 13500) &&
SpeedL)
        {
            totalspeed = totalspeed + iterCar1->second-
                >GetCurrentSpeed();
            ++count;
        }
        if((direction1 < -4500 || direction1 >= 13500) && !SpeedL)
        {
            totalspeed = totalspeed + iterCar1->second-
                >GetCurrentSpeed();
            ++count;
        }
    }
    roadAverage = totalspeed/count;

    if (roadAverage >= .70*roadSpeedLimit && roadAverage <
        0.80*roadSpeedLimit) //record only vehicles that are traveling
below 80% of
        the speed limit
    {
        writer<<1<<" "<<record<< " "<< roadAverage <<"
        "<<direction<<"
                "<<count<<" "<<iSeconds<<"\n";
    }

    if (roadAverage >= .60*roadSpeedLimit && roadAverage <
        0.70*roadSpeedLimit) //record only vehicles that are
traveling below
        80% of the speed limit
    {
        writer<<2<<" "<<record<< " "<< roadAverage <<"
        "<<direction<<"
                "<<count<<" "<<iSeconds<<"\n";
    }

    if (roadAverage >= .50*roadSpeedLimit && roadAverage <
        0.60*roadSpeedLimit) //record only vehicles that are
traveling below
        80% of the speed limit
    {
        writer<<3<<" "<<record<< " "<< roadAverage <<"
        "<<direction<<"
                "<<count<<" "<<iSeconds<<"\n";
    }

    if (roadAverage < 0.50*roadSpeedLimit) //record only vehicles that are
traveling below 80% of the speed limit
    {

```

```

writer<<4<<" "<<record<<" "<<roadAverage <<"
" <<direction<<" " <<count<<" "<<iSeconds<<"\n";
    }

    }
}
writer<<"*\n";
++countStars;
//find how many bytes have been read so far
//recSizes.push_front(0); //save this value in a list
//write the list to an index file
if(countStars == MAXSIZE) //we have captured 1 minute of data
{
    //
    writer.unsetDevice(); //detach writer from data file and close the data file
    //detach writer from index file
}
}
void HistoricalInfo::LoadHistoricalinfo()
{
    for (int n = 0; n < recSpeed.size() ; n++)
    {
        switch (recSpeed[n].CongId)
        {
            case 1: //70..80
                records1.push_back(recSpeed[n].RecordId); //copy every record
                element from recSpeed into a list
                break;
            case 2: //60..80
                records2.push_back(recSpeed[n].RecordId); //copy every record
                element from recSpeed into a list
                break;
            case 3: // ..
                records3.push_back(recSpeed[n].RecordId); //copy every record
                element from recSpeed into a list
                break;
            case 4: //..
                records4.push_back(recSpeed[n].RecordId); //copy every record
                element from recSpeed into a list
                break;
        }
    }
}

```

```
}  
void HistoricalInfo :: returnListRec(std::list<unsigned int> & recordsa, std::list<unsigned int>  
&recordsb, std::list<unsigned int> &recordsc, std::list<unsigned int> &recordsd)  
{  
    recordsa = records1;  
    recordsb = records2;  
    recordsc = records3;  
    recordsd = records4;  
}  
void HistoricalInfo::ClearList()  
{  
    records1.clear();  
    records2.clear();  
    records3.clear();  
    records4.clear();  
    recSpeed.clear();  
}
```

EVALUATING AN INTERLEUKIN-1 β INJECTION TO INDUCE DEGENERATION IN THE RAT LUMBAR INTERVERTEBRAL DISC

NSF Summer Undergraduate Fellowship in Sensor Technologies
Alta Berger (BME) – George Washington University
Advisors: Amy Orlansky, MS, and Dawn Elliot, PhD

ABSTRACT

Early signs of degenerate intervertebral discs include a decrease in the glycosaminoglycan (GAG) content within the nucleus pulposus and the annulus fibrosus, which has been linked to an alteration in the mechanics of the disc. However, the mechanisms for this reduction in GAG are currently unknown. An animal model that demonstrates disc degeneration would be useful not only for understanding the process of degeneration, but also for exploring methods of treatment or regeneration. Specifically, this study aimed to develop an *in-vivo* murine model resembling early intervertebral disc degeneration by introducing Interleukin-1 β (IL-1) into the lumbar spine. IL-1 was injected into the nucleus pulposus of the lumbar intervertebral discs of rats, and these discs were evaluated using mechanical testing, biochemical assays, and histology at one and four weeks post-injection. A control group and a sham injection of PBS were used to evaluate the significance of the IL-1 injection. At one week, a reduction was seen in nucleus GAG content in the IL-1 discs. At four weeks, glycosaminoglycan restoration was seen in the IL-1 β treated discs, suggesting the possibility of recovery of the disc.

1. INTRODUCTION

Intervertebral disc degeneration is currently recognized as a major cause of debilitating lower back pain, affecting millions of people worldwide. Despite the prevalence of this health concern, little is known about the cause of this degeneration. To date, treatments exist to cope with degenerated discs (such as disc removal and spinal fusion), but no methods have been developed to stop or reverse the process of degeneration. With age, biological alterations in the intervertebral disc (IVD) occur which lead to a decreased ability to absorb force. Degeneration begins in the nucleus pulposus (NP) of the disc [1], and therefore an animal model that mimics early NP alterations would be useful to understand progression and to develop treatments. Previous studies have shown that a loss of glycosaminoglycan (GAG) content in the intervertebral discs correlates to a decrease in neutral zone stiffness, and to an increase in neutral zone displacement and range of motion [2]. The decrease of GAG content in the nucleus pulposus significantly affects the mechanical function of the intervertebral disc and therefore may contribute to degeneration of the disc.

Although a decrease in GAG content has been linked to disc degeneration, the reason for this reduction still remains unclear. Interleukin-1 (IL-1), a naturally occurring cytokine, has been shown to play a role in cartilage degradation [3]. Specifically, it has been associated with the destruction of aggrecan. A connection between IL-1 and the decline of GAG would suggest the possibility of inhibitors capable of preventing or reversing degeneration and thereby alleviating lower back pain.

The objective of this study was to develop an *in-vivo* model in the rat lumbar spine mimicking disc degeneration possibly brought on by interleukin-1. Specifically, GAG content and mechanical properties were analyzed. This data helped to describe the relationship between IL-1 and disc degeneration.

2. BACKGROUND

2.1 *The Intervertebral Disc: Composition*

In the spine, the soft tissue structures located between the bony vertebral bodies are known as intervertebral discs. These discs are made up of two components: the nucleus pulposus and the annulus fibrosus (AF). The nucleus pulposus, located at the center of the disc, has a gelatinous consistency. Surrounding the NP is the annulus fibrosus, which consists of highly organized fibers. Figure 1 demonstrates this organization of the IVD. The disc has visco-elastic properties, with the NP responsible for the viscous behavior and the AF responsible for the elastic response.

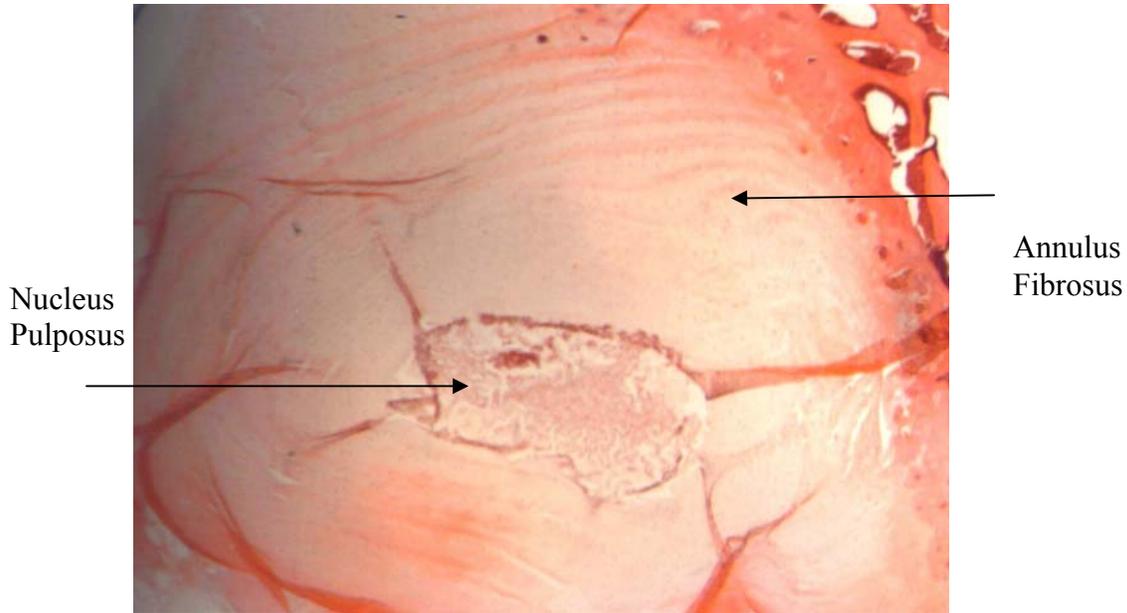


Figure 1. An intervertebral disc.

2.2 *The Intervertebral Disc: Degeneration*

The clinical relevance of disc degeneration is paramount; at least 70% of the population will suffer from lower back pain during their lives [1]. Degeneration typically begins in the NP, with alterations in the types and proportions of aggrecans and collagens. Specifically, there is a shift from the production of type II collagen to type I collagen. Also, large aggregating proteoglycans are broken down, which results in a reduction of GAG content in the NP. These changes reduce the water content of the NP, which causes the NP to lose some of its gelatinous consistency. With age, the fibers of the AF become disorganized and are vulnerable to tears. As the NP loses its ability to dissipate force, the AF becomes responsible for absorbing a greater amount of force. With degeneration come a loss of mechanical function and a reduced ability to withstand load in the disc.

2.3 *Glycosaminoglycan*

Glycosaminoglycans are long, unbranched polysaccharides with a high affinity for water. A reduction in the GAG content of the nucleus pulposus is an early marker of intervertebral disc degeneration. A decrease in GAG content correlates to a loss of water content in the NP, which will lower the osmotic pressure of the disc [5]. This results in a decreased ability to withstand load, shown by a decrease in neutral zone stiffness and an increase in both range of motion and in neutral zone displacement.

2.4 Chondroitinase-ABC

Aggrecan, which is the major proteoglycan in the IVD, contains chondroitin sulfate side chains [5]. Chondroitinase-ABC (ChABC) is an enzyme that digests chondroitin sulfate isomers, and therefore has previously been used to induce degeneration in the disc [11]. Injecting discs with ChABC significantly reduced GAG content of the NP, which corresponded to reduced mechanical ability [5, 13]. While ChABC aptly stimulates the changes associated with disc degeneration, it is not naturally occurring in animals and therefore is not a possible cause of degeneration.

2.5 Interleukin-1

Le Maitre and coworkers showed that both degenerate and non-degenerate IVDs produce interleukin-1 [4]. The presence of IL-1 has been linked to the presence of matrix degrading enzymes as well as a decrease in the production of proteoglycans and type II collagen. An *in-vivo* model of reduced GAG content due to IL-1 could be used for future studies of treatment involving inhibiting IL-1 in the disc.

2.6 Histology

Numerous histological grading systems have been developed to classify the degree of degradation of intervertebral discs. These scoring systems often consider the annulus fibrosus, the border between the AF and NP, the matrix of the NP, and the cellularity of the NP [9, 10, 11, 12]. With degeneration, the fibers of the annulus become disorganized, and the border between the AF and the NP becomes less clearly defined. In addition to this, the NP becomes more populated with chondrocytes while the abundance of notochordal cells decreases with degeneration.

3. MATERIALS AND METHODS

3.1 Study Design

This study included ten skeletally mature Sprague Dawley rats (age 7-9 months). Animals were assigned to either a 1-week or 4-week time point. Each group had rats assigned to receive a 1 μ L injection of rh-IL-1 β (n=4), a one μ L injection of 1X phosphate-buffered saline (PBS) (n=4), or no injection (n=2). Within each animal, the L3L4, L4L5, and L5L6 discs were injected.

3.2 Surgery

The rats were operated on using aseptic technique. Via inhalation, the rats were anesthetized and placed in a supine position. The lumbar spine was reached using an anterior approach. A

custom made 33-gauge needle attached to a microsyringe was inserted anteriorly to a controlled depth of 2 mm for the injections. This specific depth positioned the needle tip approximately in the center of the nucleus pulposus. The abdominal wall of the rat was closed and the rats were monitored for 45 minutes while they recovered. After this time period, the rats were returned to normal housing. Animals were euthanized at 1 week or 4 weeks after surgery and motion segments were harvested. The motion segments were then stored until testing in PBS at -20°C .

3.3 Mechanical Testing

The motion segments were thawed for one hour in room temperature PBS. The mechanical testing consisted of axial compression-tension cyclic testing followed by a compressive creep test. In order to ensure tissue hydration, testing was done in a 37°C PBS bath. The motion segments were gripped using customized microvises attached to an Instron 5542 testing system (Instron, Canton, MA). Compression-tension testing involved 20 cycles of loading from 4.5 N compression to 3 N tension at a frequency of 0.1 Hz. Following the cyclic loading was a ramp from 0 to 4.5 N compression and a compressive creep test with an applied load of 4.5 N for 45 minutes. After the testing samples were rehydrated in a room temperature PBS bath for 1 hour and then returned to -20°C for storage until biochemical analysis.

Data from the final cycle of compression-tension were analyzed using a trilinear fit model [2, 6, 7, 8] to obtain values for compressive, tensile, and neutral zone stiffness along with cyclic range of motion and neutral zone displacement. The final cycle was analyzed to ensure that the sample had gone through adequate preconditioning, eliminating the effects of super hydration that could have occurred from the free swelling in PBS during the treatment protocol [2, 6, 7, 8].

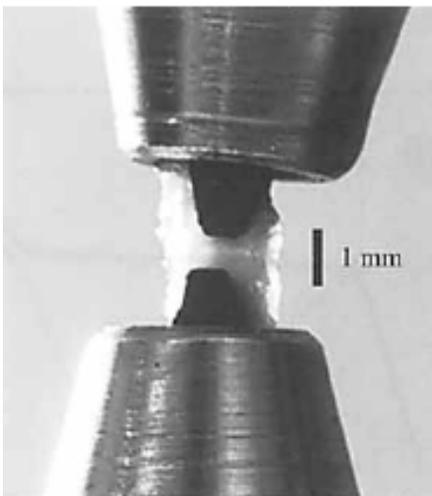


Figure 2: Customized microvises gripping bone-disc-bone motion segment. [8]

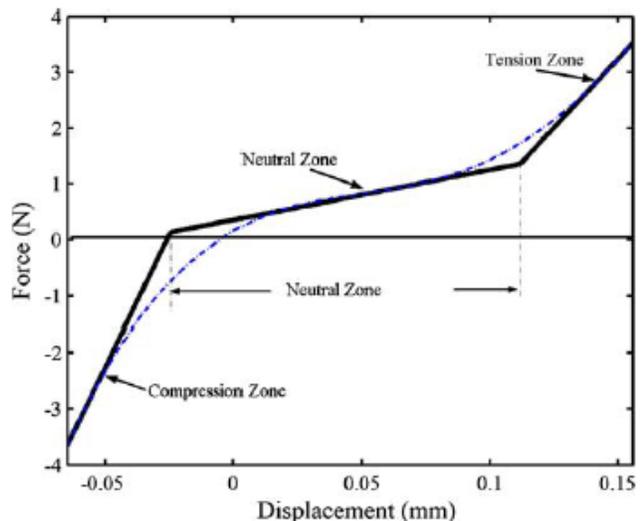


Figure 3: Representative force displacement curve showing the three loading zones: tension, compression, and the neutral zone. [8]

3.4 Biochemistry

Samples were analyzed for glycosaminoglycan content using the 1-9 dimethylmethylene blue (DMMB) binding assay. The discs were isolated from the vertebral body using a scalpel immediately upon removal from -20°C. The separated discs were put onto a freezing stage for 2 minutes to prevent the loss of nucleus pulposus as well as to maintain a frozen state [2]. A hollow 1.5-mm-diameter dermal biopsy punch (Miltex Instrument Comp., Bethpage, NY) was used to remove the entire nucleus pulposus from the isolated disc. A hollow 3.6 mm X 2.5 mm elliptical biopsy punch was used to isolate the inner annulus fibrosus, and the remaining disc tissue was considered to be the outer annulus fibrosus. The wet weights of the tissues were measured and the samples were then dehydrated at 65°C for 24 hours. Dry weight was then obtained and water content was calculated as $\%H_2O = (WW - DW) / WW$. This was followed by tissue digestion using a 5 mg/mL proteinase K solution at 65°C for 18 hours. Finally, the spectrophotometer was used to determine the GAG content based on the color reaction with the DMMB assay.

3.5 Histology

After fixation and decalcification the discs were embedded in paraffin for histological analysis. Axial sections of the disc were taken at a thickness of 7 μ m. Slides were then stained with either alcian blue and picosirius red or hematoxylin and eosin and were analyzed for disc organization and cellularity. The specific stains used, picosirius red and alcian blue, stain red for collagen and blue for glycosaminoglycan.

4. RESULTS

4.1 Disc Mechanics

A significant portion of the one week mechanical data was compromised due to difficulties encountered during mechanical testing. The 4 week data showed that the IL-1 injected discs had a smaller neutral zone length and range of motion and greater neutral zone stiffness as compared to the PBS injected discs.

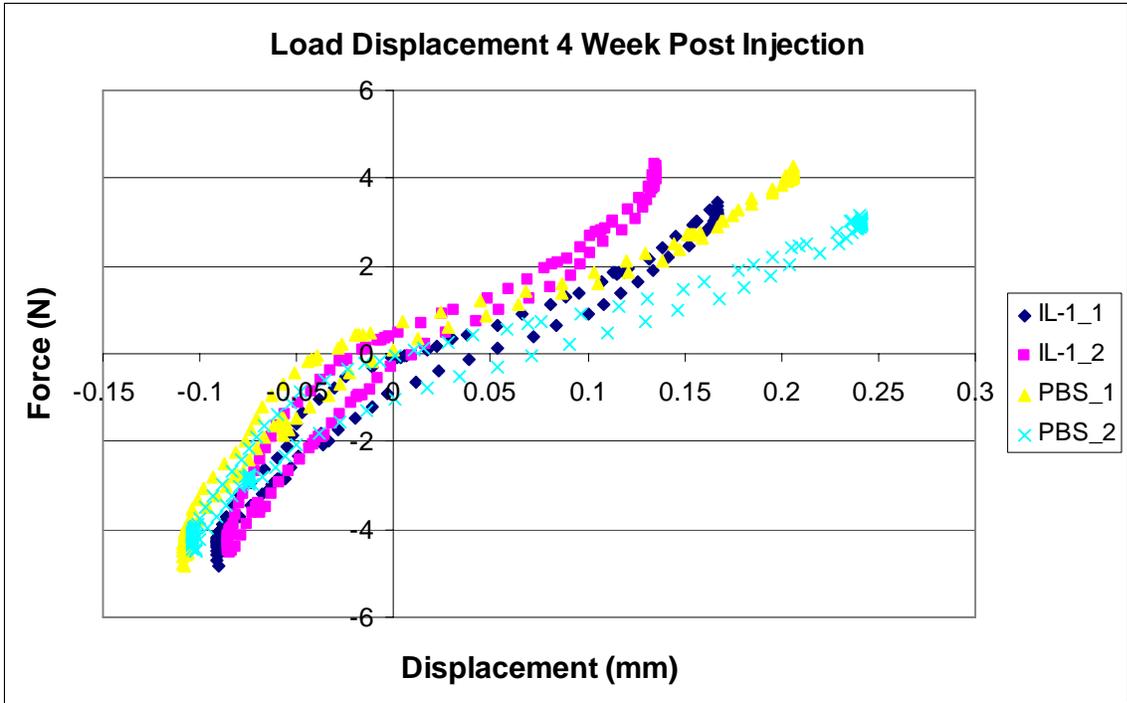
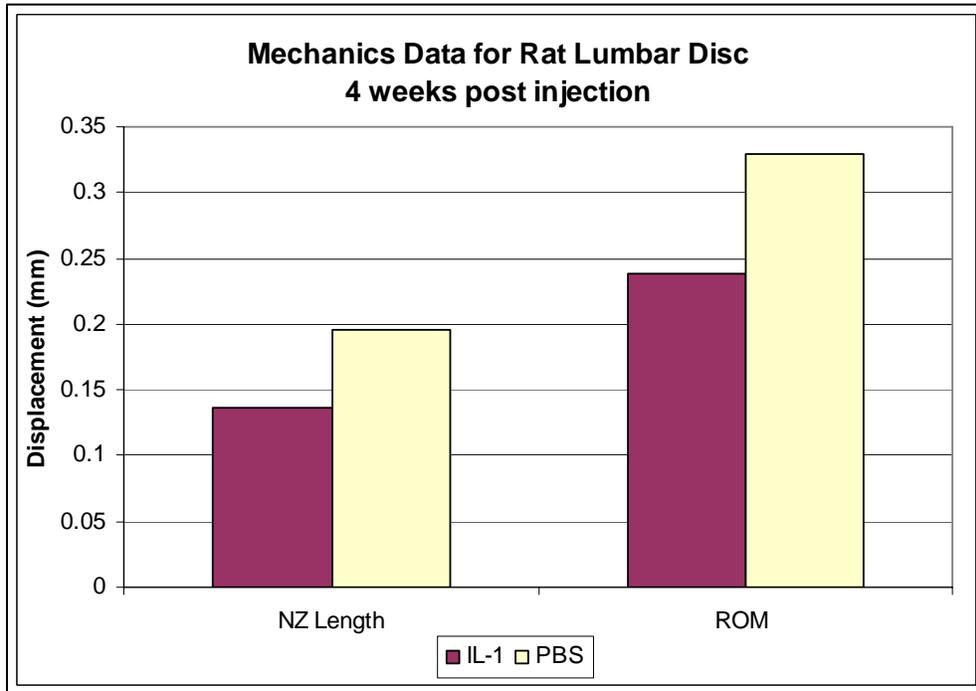


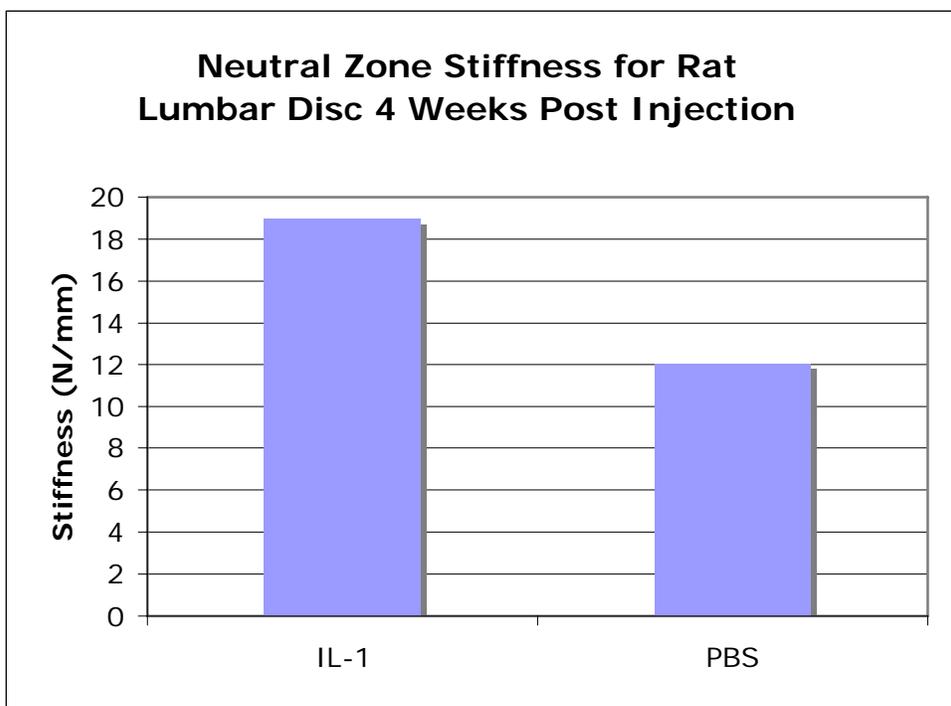
Figure 4: Load displacement curves from mechanical testing

<i>Rat Intervertebral Disc</i>	<i>Neutral Zone Length (mm)</i>	<i>Neutral Zone Stiffness (N/mm)</i>	<i>Range of Motion (mm)</i>
IL-1_1 L3L4	0.161	18.56	0.258
IL-1_2 L3L4	0.112	19.33	0.220
IL-1 Average	0.1365	18.94	0.239
PBS_1 L3L4	0.179	12.40	0.315
PBS_2 L3L4	0.213	11.62	0.345
PBS Average	0.196	12.01	0.330

Table 1: 4 week mechanical data.

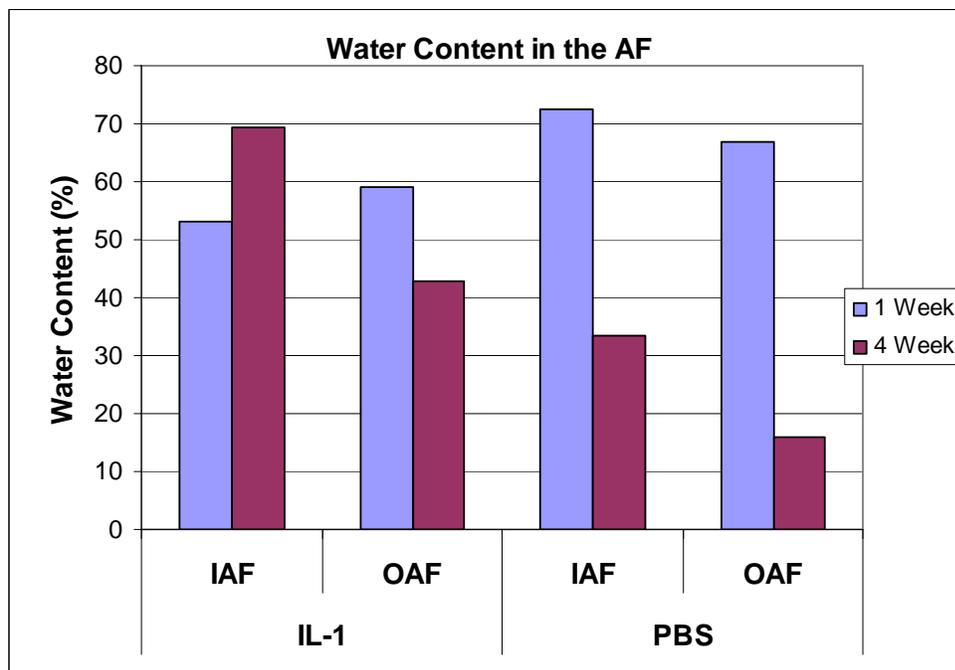
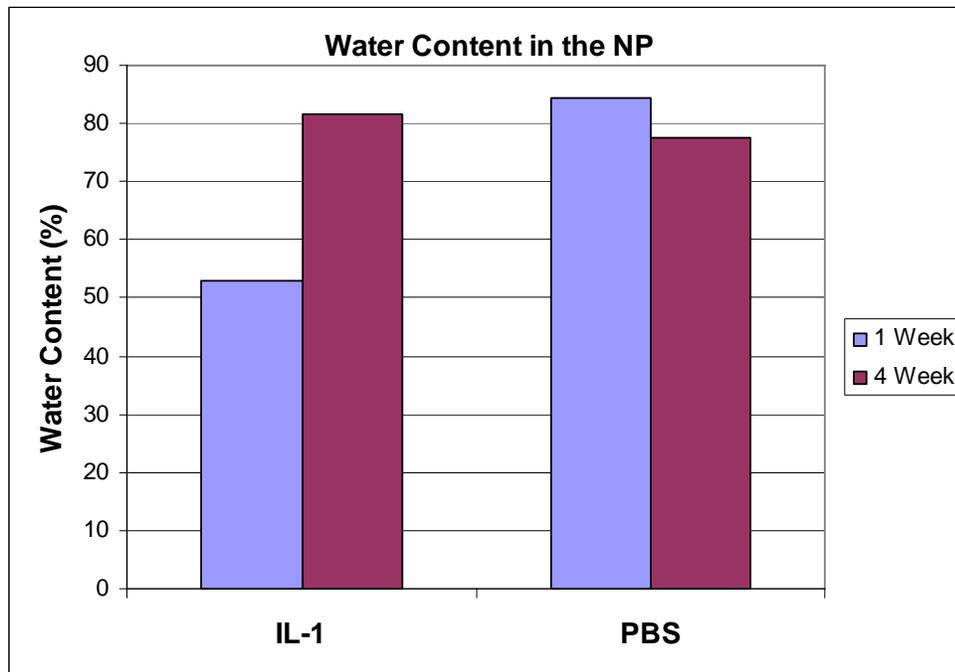


*Figure 5:
Mechanics
results for 4
week rats.*



4.2 Disc Biochemistry

After 1 week, there was a significant drop in the water content in the NP of the IL-1 injected discs as compared to the PBS injected discs. In comparing GAG content, there was an 8.63% reduction in the IL-1 NP versus the PBS NP. At four weeks, the IL-1 discs had higher water content and a 51.23% increase in GAG content compared to the PBS discs.



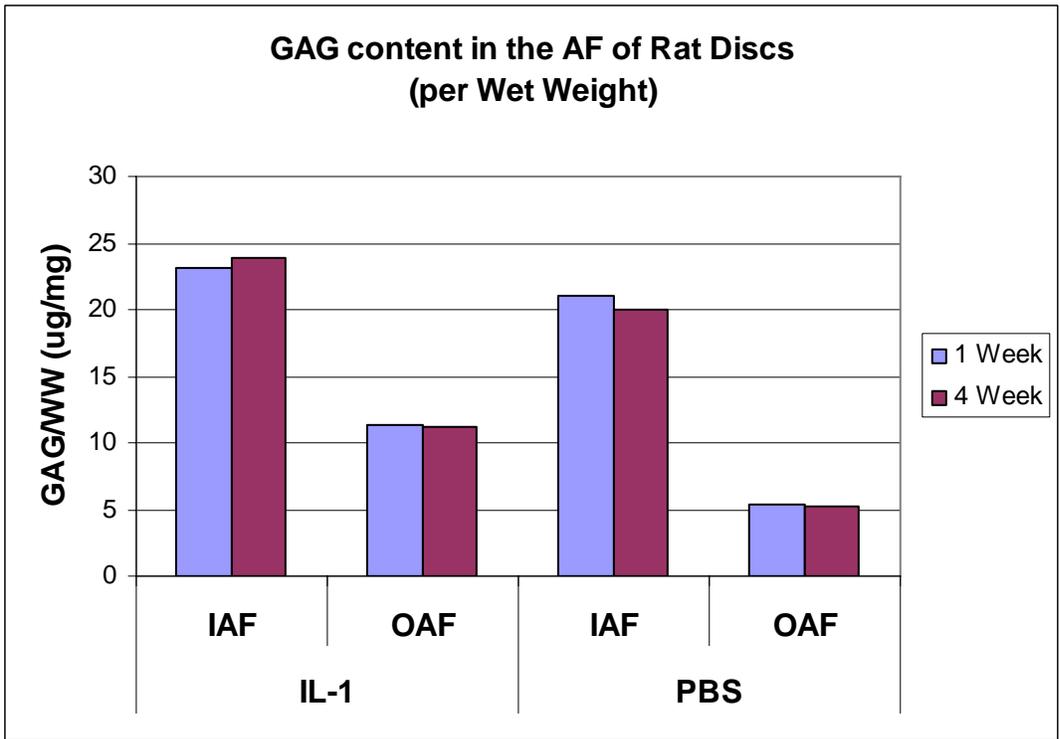
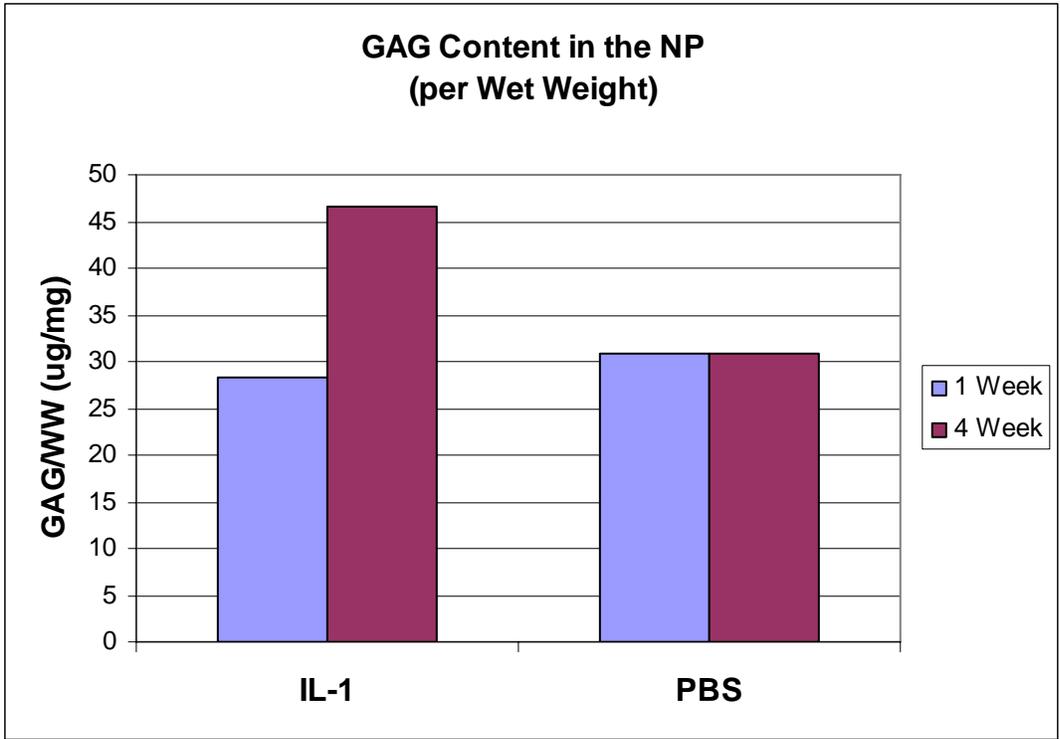


Figure 6: Biochemical Data: Water and GAG content 1 and 4 weeks post injection.

4.3 Disc Histology

The following slides were stained with alcian blue and picosirius red. The nucleus pulposus, composed mostly of glycosaminoglycan, stained a bright blue while the annulus fibrosus, which is comprised of collagen, stained red. The slides were analyzed for signs of degradation.

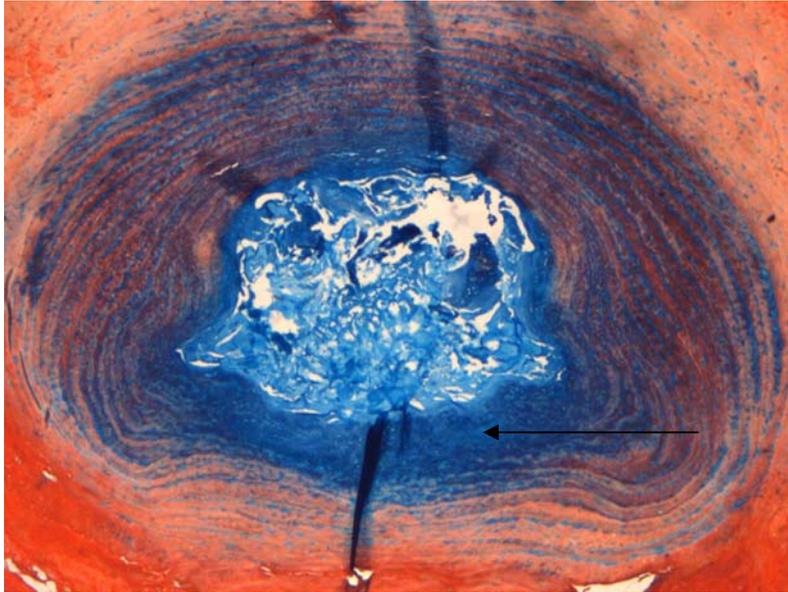


Figure 7: IL-1 injected, 1 week. The black arrow indicates the decreased ability to distinguish the NP-AF border that comes with degeneration.

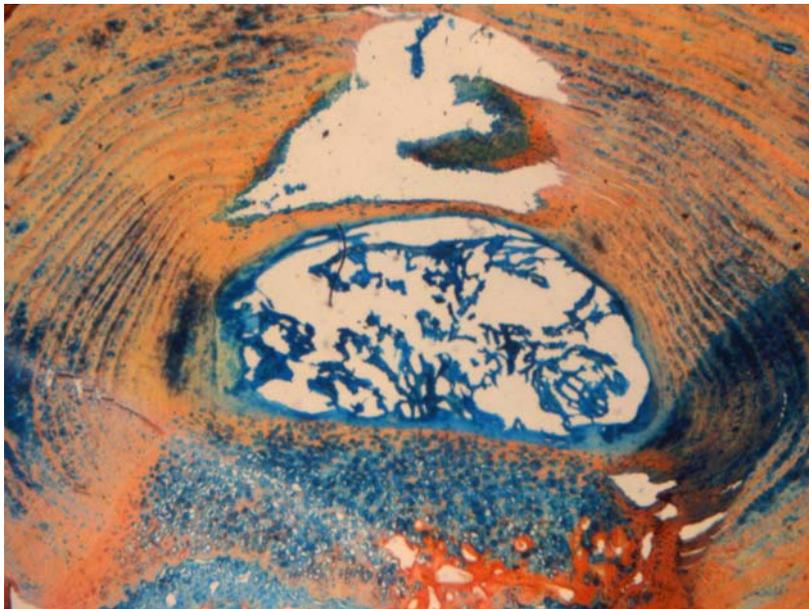


Figure 8: PBS injected, 1 week. The white area seen is due to tearing which occurred while sectioning the disc for histology and is not a result of degeneration.

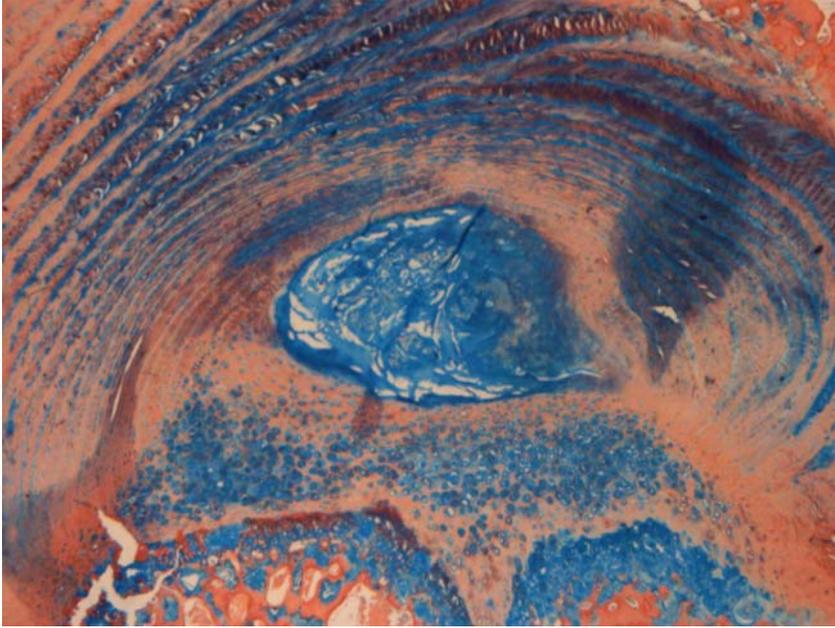


Figure 9: No injection, 1 week. The spotted area at the bottom of the disc is the endplate.

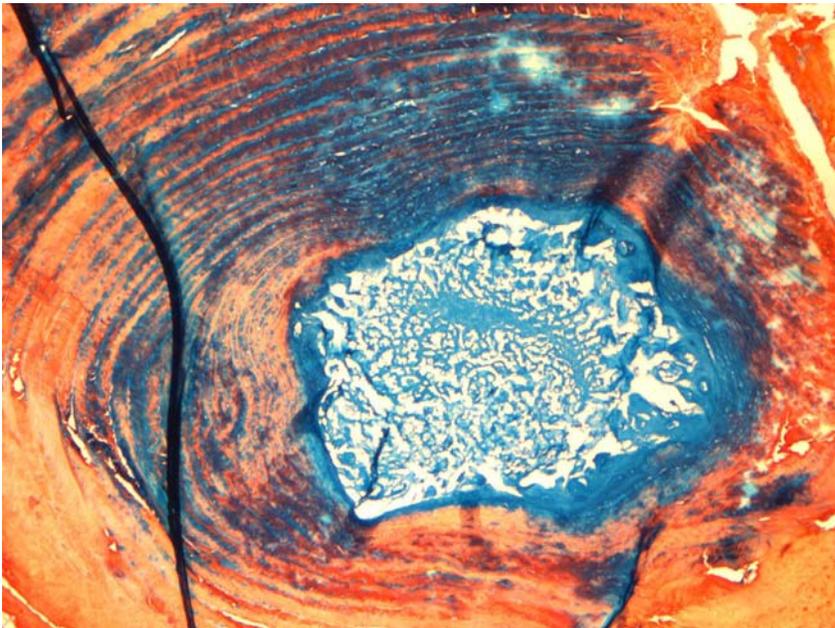
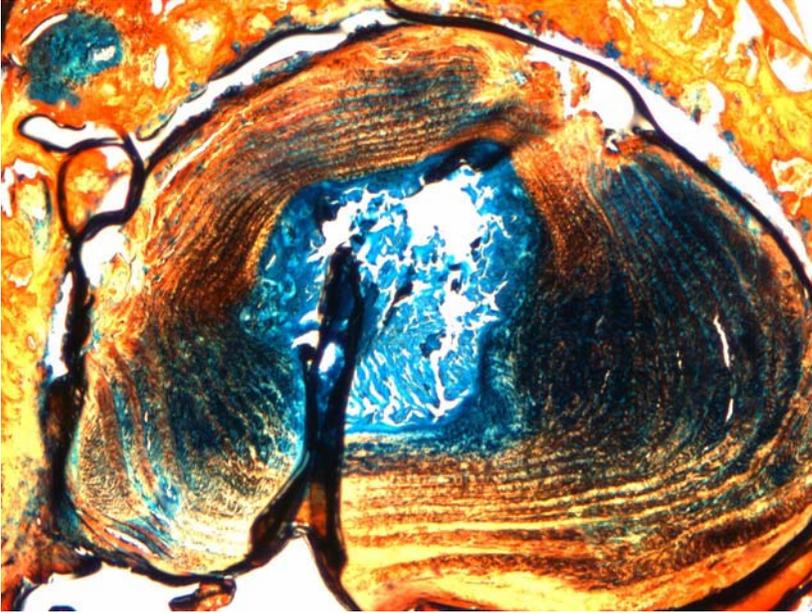
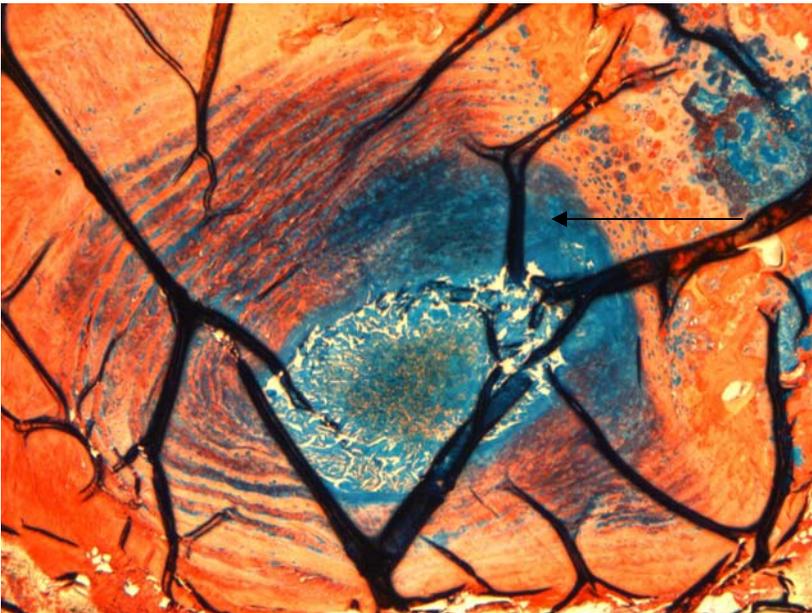


Figure 10: PBS injected, 4 week.



*Figure 11:
No injection,
4 week.*



*Figure 12: IL-1
injected, 4 week.
The dark lines seen
are folds in the
tissue due to the
process of creating
the slides. The black
arrow is used to
highlight the
decreased ability to
determine the border
between the NP and
the AF.*

5. DISCUSSION

The 8.63% change in GAG content seen in the IL-1 discs as compared to the PBS discs supported the hypothesis that this cytokine would affect the composition of the NP. As predicted, the water content was connected to the GAG content in the discs. While the mechanical data from the 1 week group was lost due to problems with testing, previous studies have shown a correlation between GAG content and mechanical function [2, 13]. Therefore, one would expect that the IL-1 injected discs would show an increase in the range of motion and

neutral zone length as well as a decrease in the neutral zone stiffness. Histology from the 1 week discs indicate signs of degeneration in the IL-1 injected discs. Specifically, the IL-1 disc exhibits a blue area that does not appear to be part of the nucleus, but is also not considered annulus (indicated by the black arrow in figure 7). This corresponds to the decreased ability to distinguish the border between the NP and the AF seen in degenerate discs. In addition to this, while the fibers of the AF in the control and PBS injected discs are organized around the NP, the AF of the IL-1 disc contains portions with a serpentine pattern. This disorganization is characteristic of degenerate discs.

The drastic increase in GAG content seen in the 4 week IL-1 injected discs suggests the possibility of recovery from the matrix degrading effects of the cytokine. One possible explanation of this increase is that the presence of IL-1 may be causing an increase in IL-1 receptor antagonist, which would allow it to further regulate IL-1. If IL-1 acts as a foreign substance when it is injected, cells may produce antibodies against it in order to prevent further attack. This increase in antibodies, or IL-1 RA, could be responsible for the increase in GAG by inhibiting the effect of IL-1 on the proteoglycans. This reaction could be tested by checking for the presence of IL-1 and IL-1 RA in the discs. For future studies, it will be important to distinguish between the presence of IL-1 and the activity of IL-1. It is possible that, although the IL-1 remained in the discs 4 weeks post injection, the cytokine was no longer active after this period of time. Because the IL-1 did not produce a lasting effect on the discs, a different method needs to be developed to ensure that the discs are experiencing a constant process of degeneration. This could be achieved by using beads that will release a certain amount of the cytokine over time.

The mechanical data corresponds to the biochemical results; an increase in GAG content in the IL-1 discs led to an increase in neutral zone stiffness and a decrease in range of motion and neutral zone length compared to the PBS discs. In the histology from the week 4 discs, the border between the NP and AF was somewhat difficult to establish in the IL-1 injected disc, shown by the arrow in figure 12. The 4 week PBS injected disc also contains an area where it is hard to distinguish between the NP and AF. All three of the discs show organized fibers in the AF.

6. CONCLUSION

There was a decrease in the levels of GAG seen in the 1 week IL-1 injected rats, supporting the idea that IL-1 plays a role in disc degeneration. While the 4 week mechanical data corresponded to the biochemical data, with n=2 the need for a larger study is great in order to draw significant conclusions. The recovery of GAG levels in the IL-1 injected discs of the 4 week rats indicates that a method needs to be developed that can release the IL-1 into the NP over time. Although histology remains a somewhat subjective area, certain markers of degeneration can be identified. These include the decreased ability to distinguish the border between the NP and the AF as well as the disorganization of the AF.

7. RECOMMENDATIONS

A necessary continuation for this study would be to include a greater number of subjects. In addition to this, a method needs to be developed to ensure that the IL-1 produces a lasting effect on the discs. A possible way to achieve this would be to encapsulate the IL-1 in such a way that it would be released periodically. In addition, the activity of IL-1 should be monitored to distinguish between remaining in the disc and staying active in the disc.

8. ACKNOWLEDGEMENTS

I would like to thank my advisors, Amy Orlansky and Dawn Elliot, for their advice and helpfulness. The process used in this study was developed through the McKay lab and is detailed in previous studies [2, 6, 7, 8, 13]. I would also like to thank the National Science Foundation for providing continued support to the SUNFEST program, which has provided me with a great experience and exposure to interesting research.

REFERENCES

- [1] A.J. Freemont, A. Watkins, C. Le Maitre, M. Jeziorska, J.A. Hoyland. (2002) Current Understanding of Cellular and Molecular Events in Intervertebral Disc Degeneration: Implications for Therapy. *Journal of Pathology* [Online]. 196, 374-379. Available: 10.1002/path.1050
- [2] John I. Boxberger, Sounok Sen, Chandra S. Yerramalli, Dawn M. Elliot. (2006, Sep.) Nucleus Pulposus Glycosaminoglycan Content Is Correlated with Axial Mechanics in Rat Lumbar Motion Segments. *Journal of Orthopaedic Research* [Online]. 24, 1906-15. Available: 10.1002/jor.20221
- [3] Jacques Claire, Marjolaine Gosset, Francis Berenbaum, Cem Gabay. (2006) The Role of IL-1 and IL-1 RA in Joint Inflammation and Cartilage Degradation. *Interleukins* [Online]. 74,371-403. Available: 10.1016/S0083-6729(06)74016-X
- [4] Christine Lyn Le Maitre, Anthony J. Freemont, Judith Alison Hoyland. (2005, Apr.) The Role of Interleukin-1 in the Pathogenesis of Human Intervertebral Disc Degeneration. *Arthritis Research & Therapy* [Online]. 7, R732-R745. Available: 10.1186/ar1732
- [5] Alessandra Colombini, Giovanni Lombardia, Massimiliano Marco Corsi, Giuseppe Banfi. (2008) Pathophysiology of the Human Intervertebral Disc. *The International Journal of Biochemistry & Cell Biology* [Online]. 40, 837-842. Available: 10.1016/j.biocel.2007.12.011
- [6] C.S. Yerramalli, A.I. Chou, G.J. Miller, S.B. Nicoll, K.R. Chin, D.M. Elliot. (2007) The Effect of Nucleus Pulposus Crosslinking and Glycosaminoglycan Degradation on Disc

Mechanical Function. *Biomechanics and Modeling in Mechanobiology*. [Online]. 6, 13-20. Available: 10.1007/s10237-006-0043-0

- [7] Wade Johannessen, Jordan M. Cloyd, Grace D. O'Connell, Edward J. Vresilovic, Dawn M. Elliot. (2006, Apr.) Trans-endplate Nucleotomy Increases Deformation and Creep Response in Axial Loading. *Annals of Biomedical Engineering* [Online]. 34, 687-696. Available: 10.1007/s10439-005-9070-8
- [8] Joseph J. Sarver and Dawn M. Elliot. (2005) Mechanical Differences Between Lumbar and Tail Discs in the Mouse. *Journal of Orthopaedic Research* [Online]. 23, 150-155. Available: 10.1016/j.orthres.2004.04.010
- [9] Koichi Masuda, Yoichi Aota, Carol Muehleman, Yoshiyuki Imai, Masahiko Okuma, Eugene J. Thonar, Gunnar B. Andersson, Howard S. An. (2004) A Novel Rabbit Model of Mild, Reproducible Disc Degeneration by an Annulus Needle Puncture: Correlation Between the Degree of Disc Injury and Radiological and Histological Appearances of Disc Degeneration. *Spine* [Online]. 30, 5-14.
- [10] T. Wang, L. Zhang, C. Huang, A. G. Cheng, G. T. Dang. (2004) Relationship Between Osteopenia and Lumbar Intervertebral Disc Degeneration in Ovariectomized Rats. *Calcified Tissue International* [Online]. 75, 205-213. Available: 10.1007/s00223-004-0240-8
- [11] Roel J. Hoogendoorn, Paul I. Wuisman, Theo H. Smit, Vincent E. Everts, Marco N. Helder. (2007) Experimental Intervertebral Disc Degeneration Induced by Chondroitinase ABC in the Goat. *Spine* [Online]. 32, 1816-1825.
- [12] Andreas G. Nerlich, Erwin D. Schleicher, Norbert Boos. (1997, Dec.) 1997 Volvo Award Winner in Basic Science Studies: Immunohistologic Markers for Age-Related Changes of Human Lumbar Intervertebral Discs. *Spine* [Online]. 22, 2781-2795.
- [13] John I. Boxberger, Joshua D. Auerbach, Sounok Sen, Dawn M. Elliott. (2008) An *In Vivo* Model of Reduced Nucleus Pulposus Glycosaminoglycan Content in the Rat Lumbar Intervertebral Disc. *Spine* [Online]. 33, 146-154.

DYE-SENSITIZED ZNO FIBERS FROM ELECTROSPINNING AND PHOTOVOLTAIC CELLS

NSF Summer Undergraduate Fellowship in Sensor Technologies
Ramon Luis Figueroa-Diaz (Chemistry) – University of Puerto Rico at Cayey
Advisor: Jorge Santiago-Aviles

Abstract

Dye-Sensitized Solar Cells (DSC) have been studied because of the ability to transform solar into electrical energy at low cost. The system that DCS uses is a photo-electrochemical system based on a semiconductor sandwiched between a photosensitized anode and cathode, both immersed in an electrolyte. First it was important to develop a semiconductor for the DSC. Zinc oxide is a good semiconductor, and its synthesis is neither difficult nor expensive. Zinc oxide was produce using Poly(ethylene oxide), zinc acetate, water and anhydrous ether. Then we used the electrospinning technique with the solution to obtain fibers. These fibers were heated at 650°C for 6 hours. For the characterization of the fibers we used Optical Microscope, Scanning Electron Microscope (SEM), and Raman Spectrometry. With the optical microscope we conclude that with the solution fibers could be made. By the shape of the fibers in the pictures taken with the SEM we could say that the fibers look like zinc oxide fibers. Finally with the Raman Spectrometry analyses we conclude that the fibers were compose by zinc oxide.

1. Introduction

With the high cost of petroleum, new sources of energy must be developed in order to reduce the cost of living. With this in mind the scientific community has created and developed new power sources, which would help to reduce costs. An example of these sources is the Dye Sensitized Solar Cells (DSC). Recently, DSCs have come to the attention of the science community because is a way of generating energy without petroleum. The only problem with solar cells is that the production is expensive. A solar cell is a device for converting energy from the sun into electricity [1]. Modern DSCs are based on the application of semiconductor physics. They are basically just P-N junction photodiodes with a very large light-sensitive area. The photovoltaic effect, which causes the cell to convert light directly into electrical energy, occurs in the three energy-conversion layers [2]. This solar cell is expensive and is not accessible to everyone. This is why solar cells have to be modified to reduce the cost and increase efficiency. Because of the properties of the DSC, new materials could be used to produce them. It has been demonstrated that zinc oxide is a good semi-conductor, which could be used for DSC [3,4].

Zinc oxide micro and nanoscopic fibers can be obtained using the electrospinning technique, which is not expensive. The electrospinning technique has been recently adopted by materials scientist and electrical engineers as a convenient technique for the inexpensive production of nano-fibers. In the electrospinning process a high voltage is used to create an electrically charged jet of polymer solution or melt, which dries or solidifies to leave a polymer fiber. One electrode is placed into the spinning solution or melt and the other attached to a collector. Electric field is applied to the end of a capillary tube that contains the polymer fluid held by its surface tension. This induces a charge transfer to the surface of the liquid [5]. Also to obtain crystalline, solid zinc oxide, the material obtained in the electrospinning process must be oxidized. Heating the solution at a high temperature to decompose the precursors and finally oxidizing the zinc, this usually results in a complete oxidation.

In DSC, the semiconductor must have a particular orientation because the electric current path must be continuous. The electrospinning technique is random; it doesn't have any deposition order, that is why the fibers must be deposited in a oriented form. There are different types of electrospinning setups (e.g. vertical electrospinning setup, side electrospinning setup, aligned fiber electrospinning setup). The best way to deposit zinc oxide is using the aligned fiber setup, this will make easier the heating and characterization process.

2. Background

2.1 Dye Sensitized Solar Cells (DSC)

Dye-sensitized solar cells (DSC) are a relatively new class of low-cost solar cell. These cells are based on a photoelectrochemical system (Figure 1). A photoelectrochemical system is based on a semiconductor placed between a photosensitized anode and cathode, both submerged in an electrolyte. These cells are very promising because they are made with low-cost materials.

DSC cells effectively separate the two functions of the silicon used in traditional solar cells. In traditional solar cells the silicon gives the source of photoelectrons, and provide the potential

barrier to separate the charges and create a current. In DSC, the semiconductor is used only to separate the charges, the photoelectrons are provided from a photosensitive dye. Although the charges are not provided completely by the semiconductor, it provide by the combination of it and an electrolyte [6].

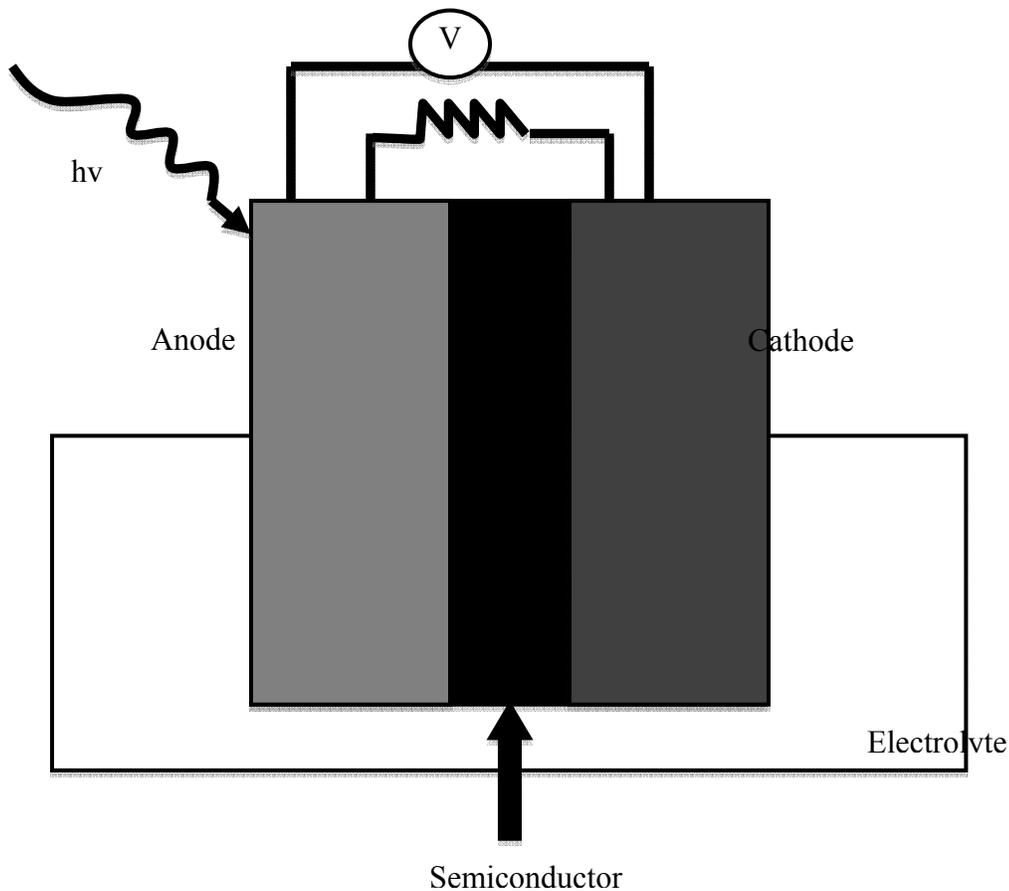


Figure 1

2.1.1 Principle

Solar cells are based generally in converting light into energy by converting this light into heat or by the photovoltaic effect. DSC are based on the photovoltaic effect, which occurs when light falling on a two-layer semiconductor material produces a potential difference, or voltage, between the two layers. The voltage produced in DSC is capable of conduct a current through an external electrical circuit that can be used to power up electrical devices [7].

2.2 Zinc Oxide

Zinc Oxide is an amorphous white or yellowish powder that is insoluble in water and in organic solvents but soluble in acid and alkali. Zinc oxide particles may be spherical, acicular or

nodular depending on the developing process. The particle shape is important for getting good physical properties. The zinc oxide UV Spectrum shows that absorbs virtually all ultra violet light radiation at wavelengths below 360 nm and provides binders outstanding protection, this is important for the characterization of zinc oxide [8]. Also zinc oxide is known to be a good semiconductor.

Zinc oxide is a versatile material; it has medical, sensors and electrical applications. The most studied application is electrical. The most common of the electrical applications are in laser diodes and light emitting diodes since it has exciton and biexciton energies of 60 meV and 15 meV, respectively [9].

2.2.1 Precursors

Zinc oxide can be obtained from many different processes, but the most common with the electrospinning process are organo-metallics, and then by the application of heat. To perform the electrospinning technique a solution has to be made. This solution has a mixture of zinc acetate in water and Poly(ethylene oxide) in anhydrous ether. After that the fibers are electro-spun and finally oxidized [10].

2.2.2 Characterization

In order to know if the synthesis of zinc oxide is completed, a characterization step is needed. Because zinc oxide is a well-studied material its characterization is simpler. The most common way to characterize zinc oxide is by IR and by Raman spectroscopy. In IR Spectroscopy, zinc oxide shows three peaks; one nearly the $1,100\text{ cm}^{-1}$ region, one nearly the 900 cm^{-1} region, and one nearly the 800 cm^{-1} region. In Raman Spectroscopy, zinc oxide shows two peaks; one intense peak at 440 cm^{-1} , and one less intense peak at 335 cm^{-1} [11]. These peaks are related to the relative motion of the atoms in the material, when excited by the light source.

2.2.2.1 Raman Spectroscopy

Raman spectroscopy is a technique used in solid state physics and chemistry to study vibrational, rotational, and other low-frequency modes. This technique relies on inelastic scattering usually from a laser in the visible, near infrared, or near ultraviolet light spectrum. The laser light interacts with the phonons or other excitations in the system, resulting in the energy of the laser photons being shifted in an up or down direction. The shift in energy gives information about the phonon modes in the system not like IR spectroscopy. IR spectroscopy yields similar, but complementary information is obtained (i.e., functional groups, orientation in space and substitution pattern).

2.3 Electrospinning Technique

The electrospinning technique is the process that uses a high voltage to create an electrically charged jet of polymer solution or melt, which dries or solidifies to leave a polymer fiber. One electrode is placed into the spinning solution or melt and another one is attached to a collector.

An electric field is applied to a needle that has the solution. When there is a pendant droplet, the electric field induces a charge at the droplet and a jet forms. This jet travels to the collector. In the way, the solvent evaporates and when the solution hits the collector it solidify (Figure 2). This solid is known as a polymer fiber, and is really small. In order to see these fibers a microscope is needed. The most common microscope is the Scanning Electron Microscope.

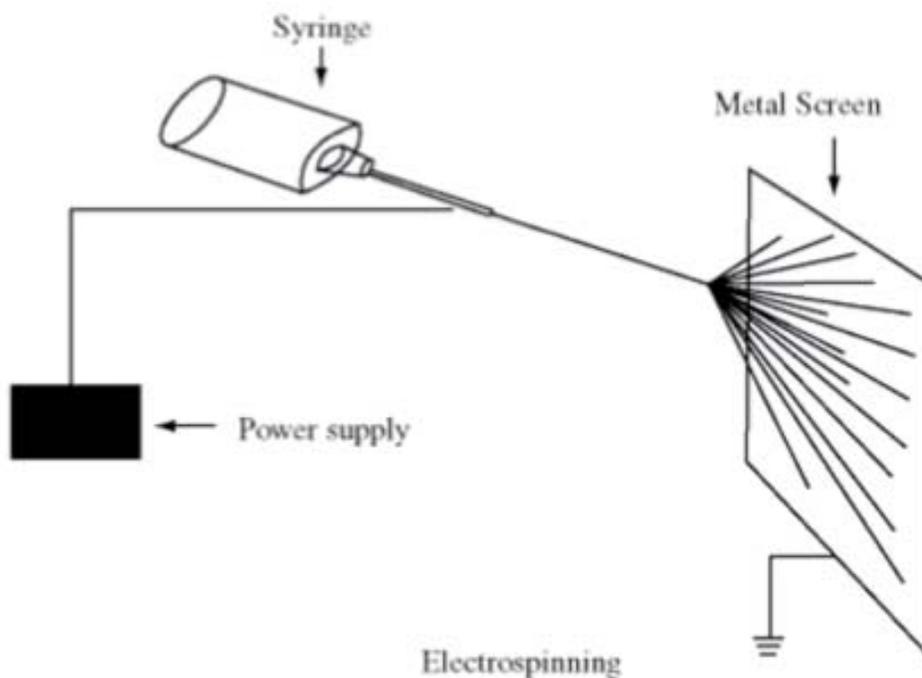


Figure 2.

2.3.1 Scanning Electron Microscope (SEM)

The scanning electron microscope (SEM) is a type of electron microscope that images the sample surface by scanning it with a high-energy beam of electrons in a raster scan pattern. The electrons interact with the atoms that make up the sample producing signals that contain information about the sample's surface (i.e. topography, composition) and other properties such as electrical conductivity.

3. Materials and Experiment

3.1 Overview

A methodology was developed to make zinc oxide fibers, due some problems the methodology was changed. A zinc acetate/Poly(ethylene oxide) solution was made and used for the electrospinning process. The fibers were heated to complete the oxidation process. Then they were characterized with a Raman Spectrometer. Finally the fibers were imaged in a Scanning Electron Microscope.

3.2 Experiment

In order to obtain zinc oxide, a precursor solution needs to be made. To produce the Zinc Oxide fibers 1.25g of Zinc Acetate will be mixed and stirred with 2.5mL of DI water. The composition will be a mixture of 20% (by weight) Poly(Vinyl Alcohol) (PVA) and water, which was heated and mixed overnight. Then the electro-spinning was performed to get nano-fibers in different orientations for the study, which is the best method in order to obtain stoichiometry values close to the desired ones.

Due to problems discussed later on this paper the methodology was changed.

3.3 Changes

The changes were basically changing the solvent and the polymer. First a solution of 19% by weigh of zinc acetate in water was made. Then a solution of 13% by weight of Poly(ethylene oxide) (900,000 MW) in ether anhydrous was added to the first solution. This final solution was mixed over night.

When the solution was ready, it was placed in a syringe. This syringe was positioned in the aligned fibers electrospinning setup. In order to study the fibers, different voltage and different solution rates were used. The fibers were collected in silicon plates.

After the electrospinning process the fibers collected were heated from 400°C to 700°C with an increasing rate of 240°C per hour for 6 hours. For the characterization, the fibers were studied using an Optical Microscope, a Scanning Electron Microscope and Raman Spectrometry.

4. Results

The solution in part 3.2 was made successfully. When the electrospinning process was tried no fibers were obtained. The concentration was changed, the rate was changed and the voltage was changed, but no fibers were obtained. Then the materials were changed.

The solution in part 3.3 was successfully made, and then the electrospinning process was performed. After the electrospinning process, fibers were obtained. Different voltages and different rates were used in this process. The fibers obtained were very good fibers, although no alignment was seen. For this last matter, different distances between the needle of the syringe and the collector were varied, but no alignment was obtained. After that pictures were taken using a Scanning Electron Microscope (Figure 3, 4, 5).

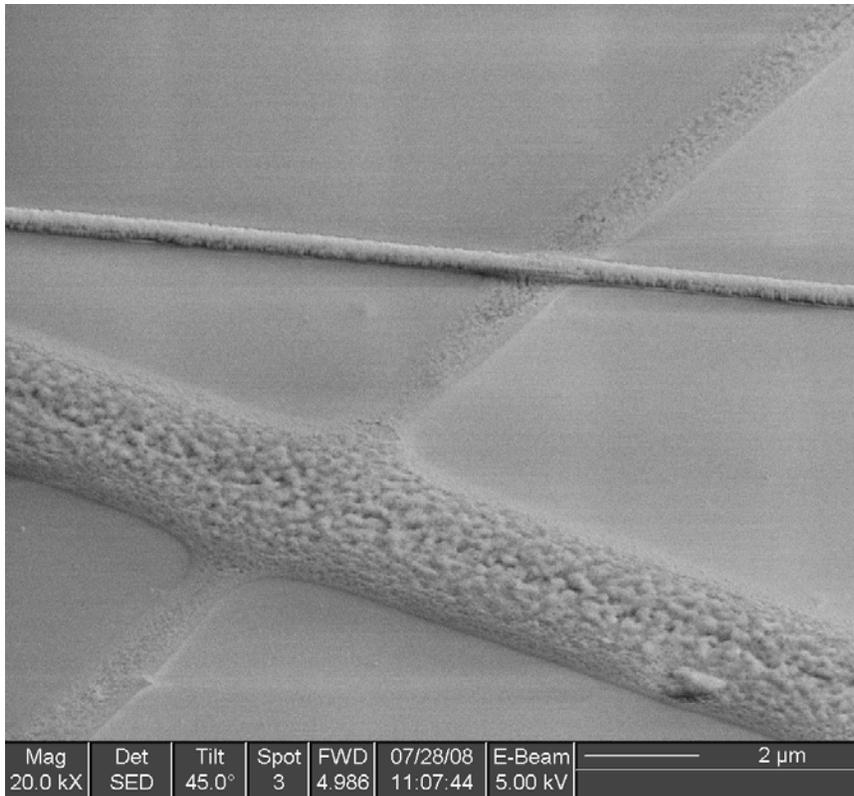


Figure 3

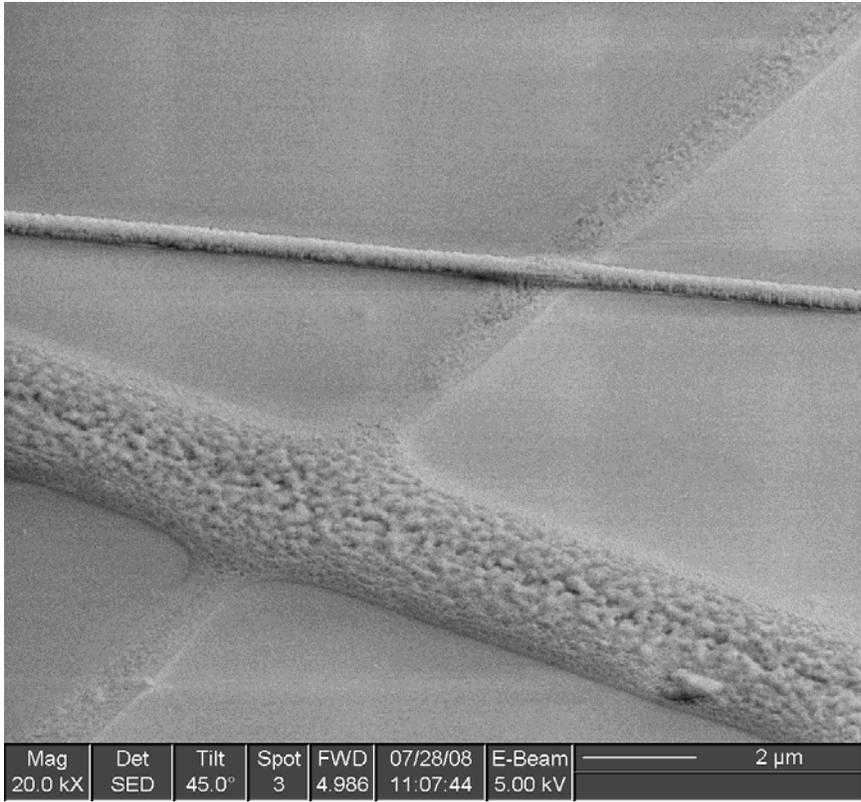


Figure 4

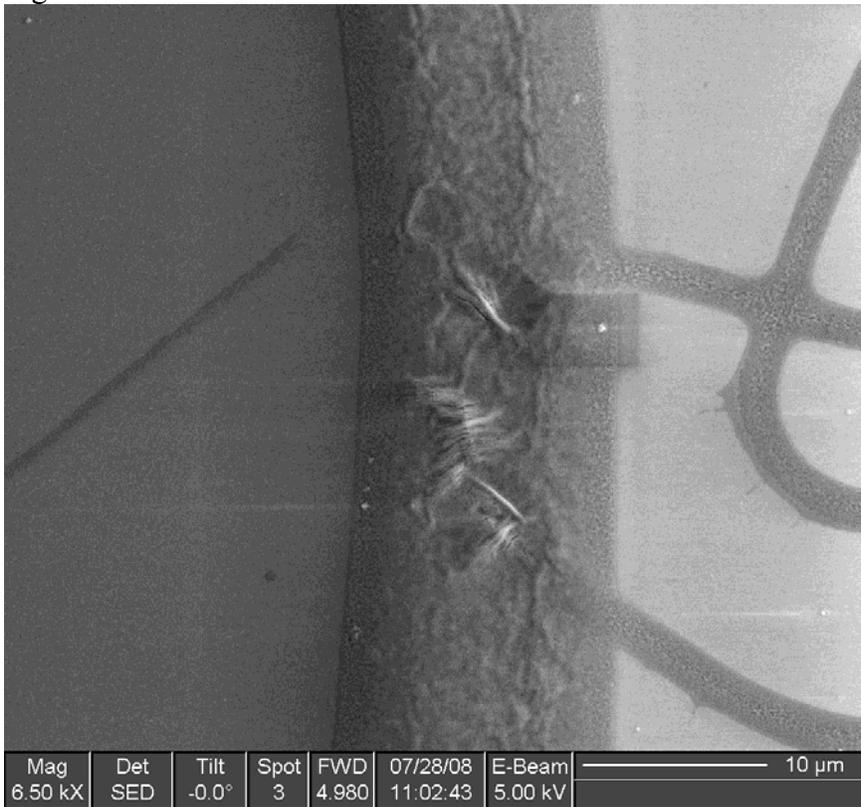


Figure 5

The fibers were heated at 650°C for 6 hours, after that pictures were taken using an Optical Microscope (Figures 6, 7, 8, 9) and using a Scanning Electron Microscope (Figures 10, 11, 12, 13).



Figure 6

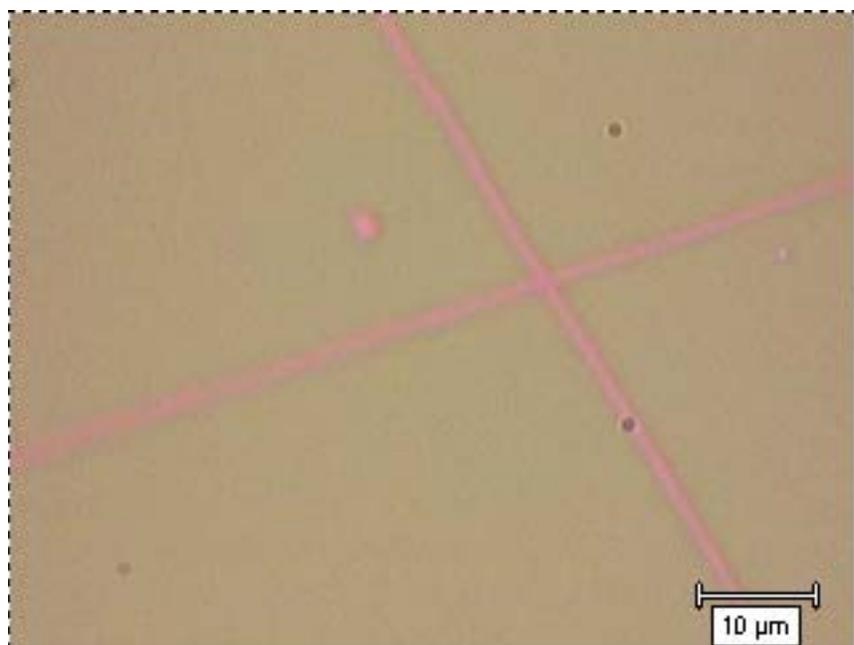


Figure 7

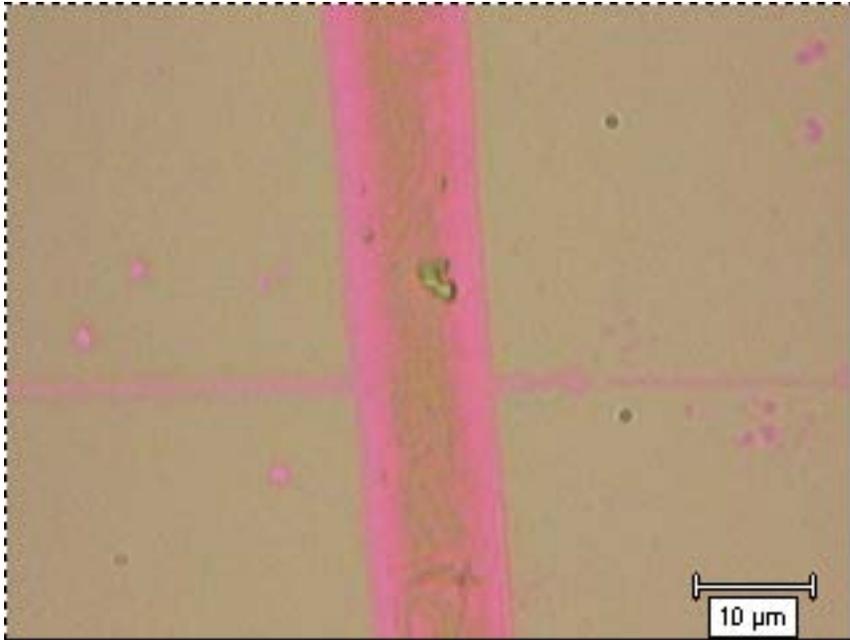


Figure 8

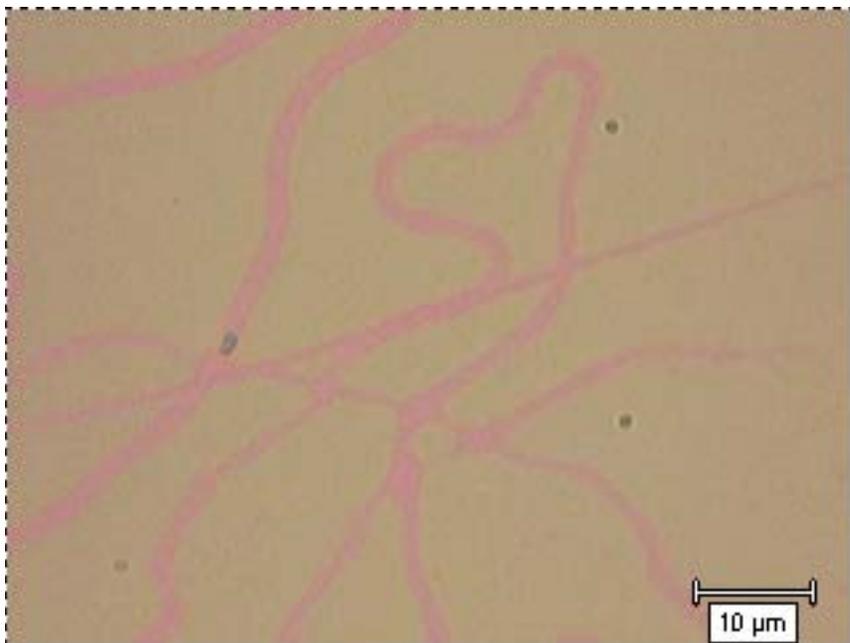
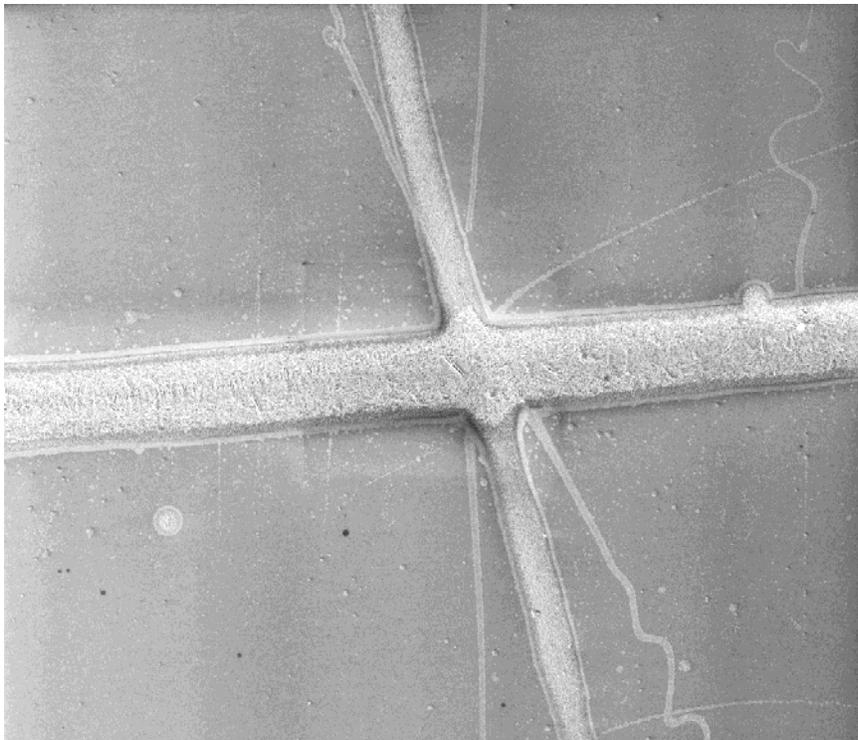


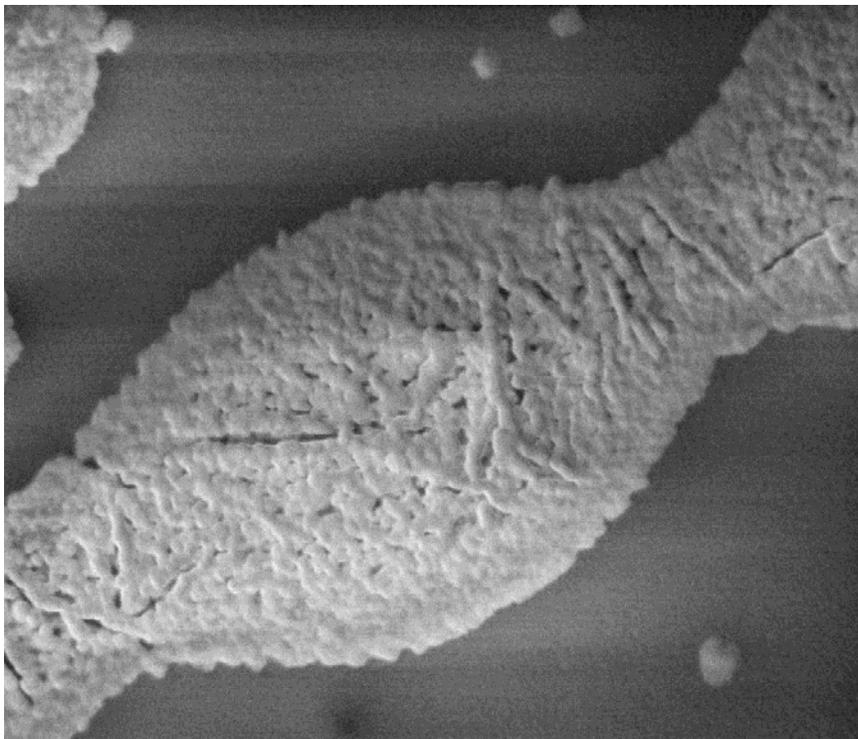
Figure 9

By the pictures obtained with the optical microscope, we can tell that after heating the fibers there were fibers and that is possible that they are zinc oxide fibers.



Mag	Det	Tilt	Spot	FWD	08/01/08	E-Beam	20 μ m
1.50 kX	SED	0.0°	3	4.937	10:29:08	5.00 kV	

Figure 10



Mag	Det	Tilt	Spot	FWD	08/01/08	E-Beam	1 μ m
65.0 kX	SED	0.0°	3	5.024	10:19:09	5.00 kV	

Figure 11

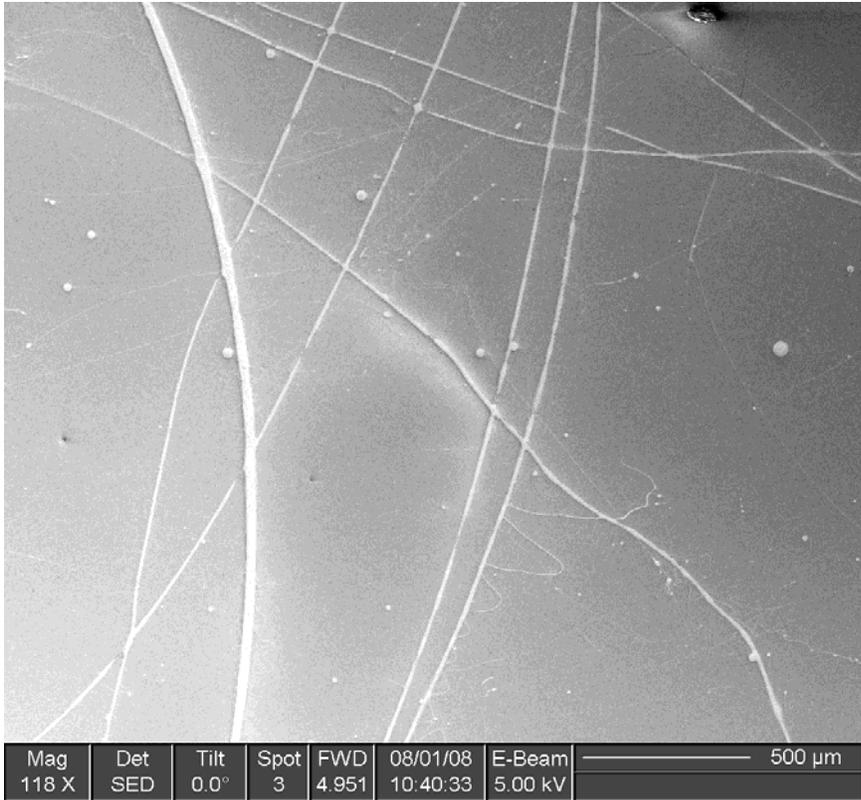


Figure 12

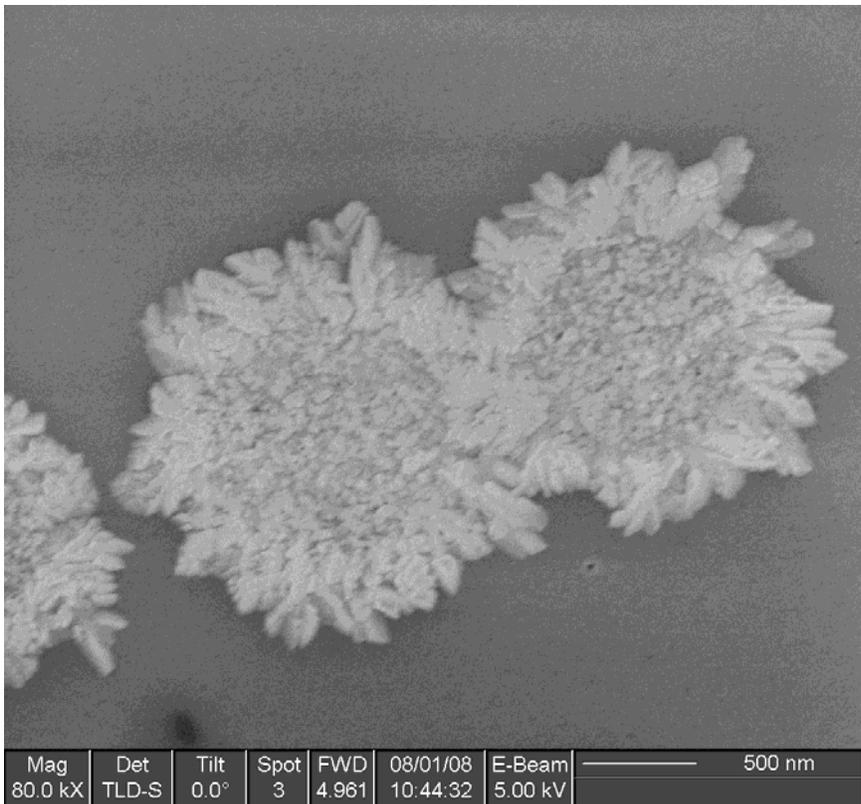


Figure 13

By the look on the fibers, compared to the ones studied by Punniamoorthy Ravirajan, et al, the fibers that were obtained are indeed zinc oxide fibers. In figure 13, the fibers look like crystals; this is the shape of zinc oxide.

Finally, we used the Raman Spectrometer to see the fibers spectrum and to be certain that what we had was indeed zinc oxide (Figure 14).

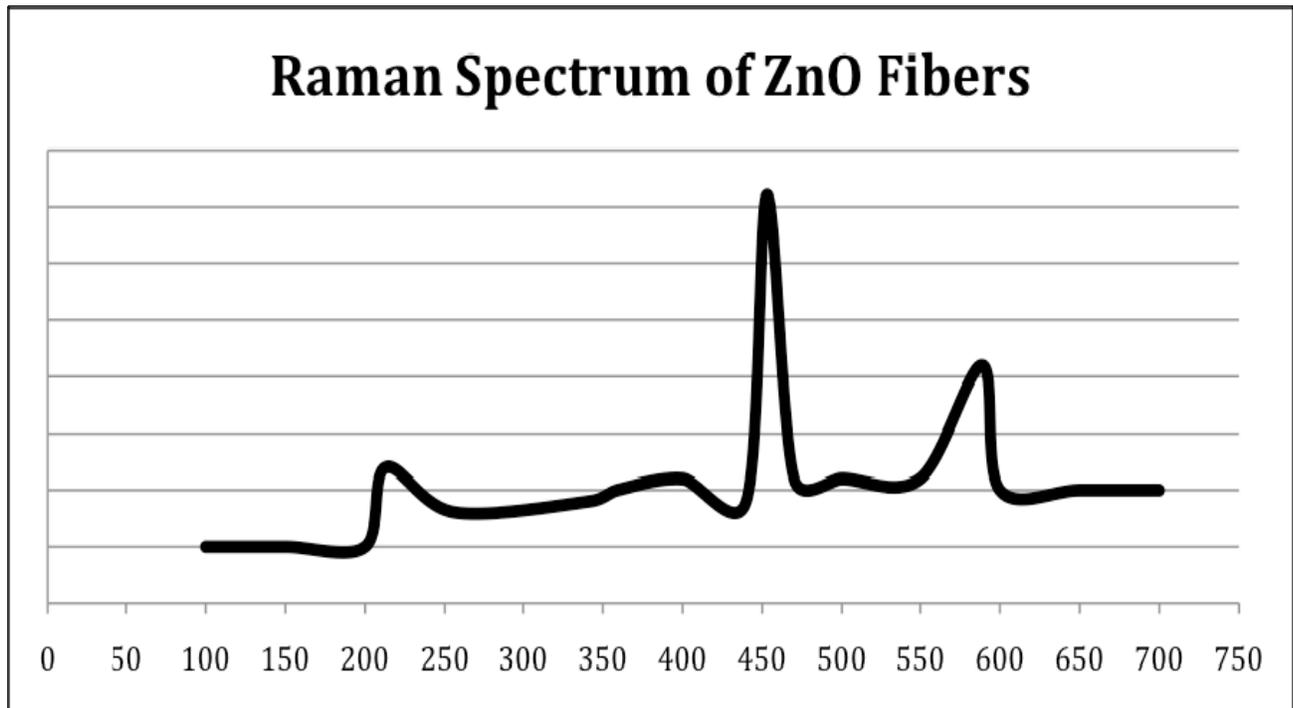


Figure 14

The spectrum tells us that there are three important signals. The first one is a weak signal at 213 cm^{-1} ; the second signal is a strong peak at 453 cm^{-1} ; and a third one a medium peak at 589 cm^{-1} . Comparing these values with the ones in the study by Periasamy Viswanathamurthi, et al, means that our values are close to the ones in the study. With this in mind we can assume that our fibers are composed by zinc oxide.

5. DISCUSSION AND CONCLUSIONS

The methodology that was developed the first time did not produce any results that could be used for the Dye Sensitized Solar Cells (DSC). By changing the concentration of the mixture, we were able to get better results but still no fibers in the collector. The reason for this is that the solvent used was water. Water has a boiling point at 100°C , which is high and it doesn't have the time to vaporize before it gets to the collector. This doesn't let the fiber to spin and what you get is a lot of dots in your collector.

By changing the concentration of the solution we didn't had any good results, the only thing that seem to change was when the voltage was changed. This was because higher voltage allows the jet to get charged, but he distance was too short and the jet didn't get completely charged.

After the change of the solution methodology, the electrospinning process was performed. In the electrospinning process fibers were obtain in a silicon collector. It seems that by changing the solvent to a more volatile one fibers can be developed. This is because the solvent evaporates before it gets to the collector and this allows the jet to solidify and finally create a fiber. This was check using the Scanning Electron Microscope. With the microscope pictures of the fibers were taken and the fibers that were obtained were not aligned. By this we can tell that the methodology developed to generate fibers was successful only to generate fibers but not to obtain them aligned.

Following the heating of the fibers, the pictures taken with the optical microscope. With this pictures we can only tell that after the heating there were still fibers, and that the heat does not affect the way that the fibers are arranged when they are collected. After the pictures we analyze the fibers using the Raman Spectrometer. There was a problem because the fibers were deposited on silicon plates and silicon has two signals that overlap the zinc oxide signals. We had to run some background spectrums of silicon in order to subtract them from the fiber's spectrums. With the results we can conclude that the composition of the fibers was indeed zinc oxide.

We can conclude that the first methodology developed for zinc oxide fibers do not work to generate zinc oxide fibers. After the changing of the methodology we can conclude that the methodology works to generate zinc oxide fibers but these fibers are not aligned.

6. Recommendations

This part of the project is just the beginning; there are many things to do with the work that has been done. First, a methodology must be developed in order to get aligned fibers. This methodology must consist in changing the voltage applied to the syringe used for electrospinning and changing the distance between the collector and the syringe.

Another recommendation is that if this project is going to be reproduce, would be better to use a collector with heat capacity different than silicon. Silicon is active in most of the characterization methodologies.

7. Acknowledgments

I would like to thank Dr. Jorge Santiago-Aviles for giving me the opportunity to work with him and for all the support that he gave me since the beginning of the program. I also want to thank Preethi Gopu, the graduate student that I worked with for all the help she gave me through the program. I'm also thankful with the LRSM Building for letting me use the necessary equipment to conduct my research. Finally, I want to thank the National Science Foundation, the University of Pennsylvania, Dr. Jan Van der Spiegel, and the Microsoft Corporation for their dedication and contribution with the REU programs all over the years.

8. References

- [1] S. R. Scully, M. D. McGehee.(2006, Aug). Effects of optical interference and energy transfer on exciton diffusion length measurements in organic semiconductors. *J. Phys.* [Online]. 100(3), pp. 034907-1 – 034907-5. Available: <http://gcep.stanford.edu/learn/extpublications.html>
- [2] W. J. E. Beek, *et al.* (2005, Oct.). Hybrid solar cells using a zinc oxide precursor and a conjugated polymer. *Adv. Funct. Mater.* [Online]. 15(10). pp. 1703-1707.
- [3] P. Viswanathamurthi, *et al.* (2003, Dec.). The photoluminescence properties of zinc oxide nanofibres prepared by electrospinning. *Nanotechnology.* [Online].15(2004). pp. 320-323.
- [4] G. Li, *et al.* (2005, Nov). High-efficiency solution processable polymer photovoltaic cells by self-organization of polymer blends. *Nat. Mater.* [Online]. 4(2005). pp. 864-868. Available: <http://www.nature.com/naturematerials/>
- [5] K. Molnár, E. Košťáková, L. Mészáros. (2008, Jun.). Electrospinning of PVA/carbon nanotube composite nanofibers: the effect of processing parameters. *Mater. Sci. Forum.* [Online]. 589(2008). pp. 221-226.
- [6] P. Ravirajan, *et al.* (2006, Feb.). Hybrid Polymer/Zinc Oxide Photovoltaic Devices with Vertically Oriented ZnO Nanorods and an Amphiphilic Molecular Interface Layer. *J. Phys. Chem.* [Online]. 110(2006) pp. 7635-7639.
- [7] H. Oba, *et al.* (1986, Nov.). Organic dye materials for optical recording media. *Appl. Optics.* [Online]. 25(22). pp. 4023-4026.
- [8] Y. Ni, *et al.* (2007, Mar.). Preparation, characterization and property study of zinc oxide nanoparticles via a simple solution-combusting method. *Nanotechnology.* [Online]. 18(2007). pp. 1-6.
- [9] H. Wu, W. Pan. (2005, Jun.). Preparation of Zinc Oxide Nanofibers by Electrospinning. *J. Am. Ceram. So.* [Online]. 89(2). pp. 699-701.
- [10] P. W. M. Blom, *et al.* (2007, May). Device Physics of Polymer: Fullerene Bulk

Heterojunction Solar Cells. *Adv. Mater.* [Online]. 19(2007). pp. 1551-1566.

[11] M. Rajalakshmi, A. K. Arora. (1999, Nov). Optical phonon confinement in zinc oxide nanoparticles. *J. Appl. Phys.* [Online]. 87(5). pp. 2445-2448.

NEONUR: A FEEDING DEVICE FOR PREMATURE NEONATAL NURSING

NSF Summer Undergraduate Fellowship in Sensor Technologies
David E. Joffe (Electrical and Computer Engineering) – Carnegie Mellon University
Advisor: Dr. Jay N. Zemel

ABSTRACT

Premature infants lack the ability to communicate verbally; therefore doctors who care for them need tools to measure their health, particularly their ability to nourish themselves adequately. These measurements need to be obtained in a variety of locations, such as intensive care nurseries and in homes. The bulky size and tethered nature of current devices measuring infant feeding make such monitoring too difficult. Nurses doing research at the Children's Hospital of Philadelphia (CHOP) have requested the University of Pennsylvania School of Engineering to develop a device that will meet the needs of CHOP nursing researchers. During the 2007-2008 academic year, University of Pennsylvania Seniors Leslie Chen and Preeti Rajendran have worked on developing a prototype for a new system that has progressed to the troubleshooting phase. In particular, problems with the on board FLASH memory chip persist. This summer, functionality of data acquisition, data amplification, and Enhanced Universal Synchronous Asynchronous Receiver Transmitter (EUSART) communication between the microcontroller and a personal computer have been tested, modified and are now confirmed to be functioning properly. There is still a need for more testing before a prototype will be ready for further testing and optimization.

1. INTRODUCTION

1.1. Motivation

In the United States in 2006 over half a million babies were born prematurely. [1] Researchers in the Department of Nursing at the Children's Hospital of Philadelphia (CHOP) are studying the development of the feeding instinct and sucking abilities of ill and premature neonates. To monitor the babies' sucking ability, researchers have been using a device that measures the flow of milk through a bottle to the infant. Measurement of infant sucking behavior is important in determining infant health, as it is a critical function leading to growth and development. [2]

The researchers at CHOP have been using a device that requires the complicated assembly of many parts and time-consuming cleaning after each use. In addition, the device that is currently in use consists of a large, expensive processing unit that must be connected by a wire to a feeding apparatus. This impairs the ability of researchers to maneuver and orient the device as needed, so measurements are difficult to obtain in the real settings in which these infants will need to nourish themselves adequately. Furthermore, the poor maneuverability and the complex process of preparing the current device for each use introduce significant human error into the data.

A prototype NeoNur device has been developed up to the debugging stage. The electrical circuit consists of a pressure sensor, a PIC microcontroller, an instrumentation amplifier, and a FLASH memory chip, along with a battery, resistors, capacitors, pins to program the PIC microcontroller, and pins to communicate with a computer. Using this system, the device is designed to collect inputs from a pressure sensor, process them, and save them to the FLASH memory chip. Upon request, the device would then retrieve that information and send it to a computer. According to calculations made by a University of Pennsylvania senior design team for the 2007-2008 academic year, the device is designed to cost a maximum of \$170 per unit, while the equipment that is currently in use costs almost \$5000.00.

1.2. Goals

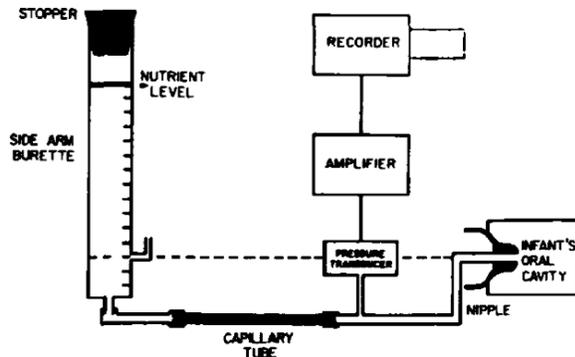
This past academic year, a senior design team at University of Pennsylvania worked on the NeoNur device and has developed software and hardware to the troubleshooting phase. The goals for the summer of 2008 were:

1. Identifying which parts of the design are functioning properly
2. Determining the sources of error in hardware and software
3. Implementing a simple solution to correct these problems.

2. BACKGROUND

2.1. Previous Devices

In the early 1960's researchers at the University of Pennsylvania School of Medicine developed an apparatus to measure infant feeding behavior. Using the device, researchers found that healthy babies produced pressures between 0 and -300mmHg compared to atmospheric pressure. [3] The device used (Figure 1) is composed of a burette with a stopper in the top filled with formula, milk, or other nutritious fluid, that is connected by a capillary tube to a pressure transducer and a nipple, from which the infant sucks the fluid. The pressure transducer is then connected to an amplifier that sends information to a recorder.



While this device successfully measured feeding, the process was unnatural, as the infant had to lie on his or her back with the nipple in its mouth instead of being held in a more upright position as occurs while feeding from a bottle or from a breast.

Figure 7: Kron apparatus for measuring newborn sucking behavior

Reuben E. Kron and Mitchell Litt further developed the idea of making a device to measure feeding in the early 1990's. The device that resulted is the apparatus that is currently being used by researchers at CHOP. It was an improvement over the previously used device in that it made feeding much more natural, as the neonates could be held while being fed, instead of having to lie on their backs. However, the wires still got in the way. Figure 2 demonstrates that the device currently in use at CHOP has a processing unit as well as a bottle with the sensory electronics contained in the upper portion. It is large and bulky, the bottle is tethered to the processing unit, and the system as a whole is not easily transportable.



(a)



(c)



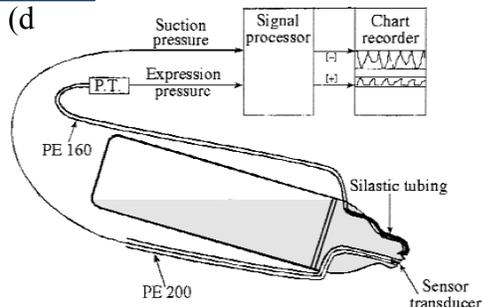
(b)



(d)

Figure 2: The bulky processing unit (a), the 8 tools used to assemble and disassemble the device (b), and the 14 parts of the bottle (c,d), of the current device being used at CHOP

It requires eight tools to disassemble the device for cleaning in



between uses, and the bottle breaks up into many small pieces. Disassembling, cleaning and reassembling the current device takes approximately 35 minutes and it is a complex and confusing process that has introduced error into the calculations and measurements taken by the device.

A team from Houston, Texas developed another apparatus in 2001 that is based on the same principles as the previous devices (Figure 3). A nipple is connected to a pressure transducer via a capillary tube, which then outputs data to a signal processor connected to a chart recorder. [4]

Figure 3: Device developed in Texas

None of the previous devices have managed to implement a wireless system or made the apparatus user-friendly. Additionally, none of the other devices have the capability to integrate other vital signs into the measurements collected, showing the interaction between other body functions, such as breathing, and neonatal feeding.

2.2. Scientific Principles Behind Design

The NeoNur design is based on the Hagen-Poiseuille equation, which states that the volumetric flow rate of a liquid through a pipe is proportional to the pressure differential at either end of that pipe. Since volumetric flow is constant in a continuous pipe, with atmospheric pressure on one end of the tube, the pressure at any point in the tube with respect to atmospheric pressure is proportional to the pressure created within the feeding infant's mouth with respect to atmospheric pressure. Since the infants' ability to create negative pressure is necessary for normal feeding, measuring pressure in the milk is desirable.

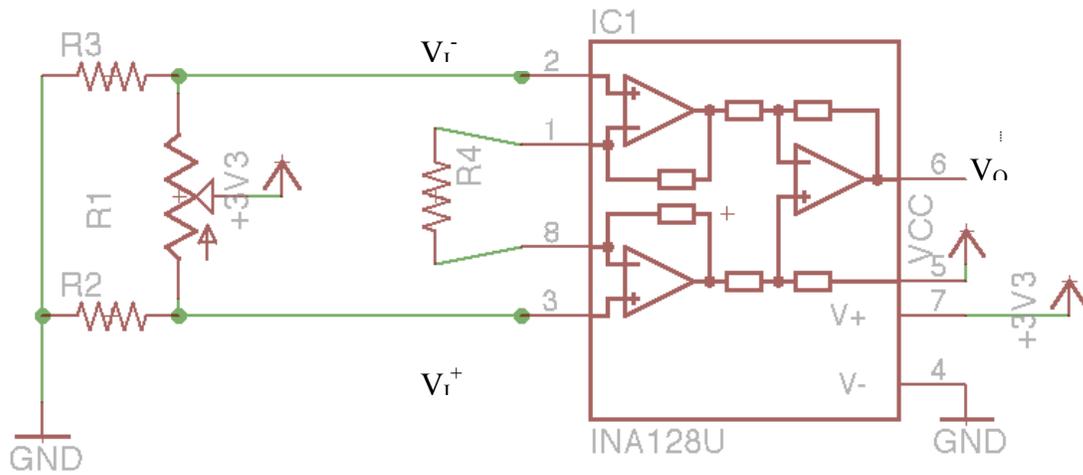
The pressure transducer uses piezoresistors arranged in a Wheatstone bridge. When a voltage is placed across two opposite corners of the Wheatstone bridge, the voltage difference across the other two corners is proportional to the pressure deforming the piezoresistive elements.

3. TESTING THE PRESSURE TRANSDUCER

Qualitative testing showed that the pressure transducer was functional. Because there were no devices available to create a preset or known level of pressure, one node of the pressure sensor was connected to ground, and the opposite node was connected to a power source set to 3V. Sucking on the tube connected to pressure sensor resulted in the voltage across the other two nodes increasing above zero volts; blowing through the same tube caused the voltage to drop below zero.

4. TESTING THE INSTRUMENTATION AMPLIFIER

Multiple tests of the instrumental amplifier led to the conclusion that the instrumental amplifier is functional. In all tests performed specifically on the instrumental amplifier, the circuit



depicted in Figure 4 was used. The amplifier used the

Figure 4: The Circuit test the amplifier. $R_1=92.5k\Omega$, $R_2=R_3=98.7k\Omega$, $R_4=R_G$.

INA128U by Burr-Brown Products from Texas Instruments. Signal amplification can be controlled by selecting a resistor value, R_G , that goes across pins 1 and 8 of the chip.

In testing, researchers found that babies produce no more than 300 mmHg of negative pressure [3]. Because the pressure sensor being used is rated at a sensitivity of $5\mu V/mmHg/V$, and because the pressure sensor will be powered by 3V, the maximum voltage coming from the pressure sensor would be 4.5mV. To insure a margin of safety, a maximum voltage of up to 5mV was assumed. The input range of the ADC of the microcontroller is 3.2mV to 3.0 V. A gain between 200 and 300 would result in a maximum voltage between 2.15V and 2.75V.

4.1. INA128 Test 1

The first test of the amplifier used the R_G value from the circuit from the beginning of the debugging phase of the project. This R_G value was $997\Omega \approx 1k\Omega$; however, as the data in Table 1 and Figure 5 show, the gain is much lower than expected, at 40 to 50 in the range of operation.

Table 1

V_I (mV)	V_O (V)	$V_O - V_{REF}$ (V)	G
1,440	2.31	1.06	0.736
959	2.31	1.06	1.11
514	2.30	1.05	2.04



249	2.30	1.05	4.22
99.4	2.30	1.05	10.6
50.6	2.30	1.05	20.75
24.8 [†]	2.30	1.05	42.3
10.2 [†]	1.69	0.44	43.1
4.91 ^{*†}	1.46	0.21	43.7
2.46 ^{*†}	1.36	0.11	44.7
1.01 ^{*†}	1.30	0.05	47.5
0.500 ^{*†}	1.27	0.02	40
0.00 ^{*†}	1.25	0.00	N/A
-0.498 ^{*‡}	1.04	-0.21	422
-1.01 ^{*†}	1.21	-0.04	39.6
-2.53 ^{*†}	1.14	-0.11	43.5
-5.10 ^{*†}	1.03	-0.22	43.1
-10.1 [†]	0.819	-0.431	42.7
-25.2 [†]	0.176	-1.07	42.6
-50.1	0.0802	-1.1698	23.3
-100.	0.0830	-1.167	11.7
-250.	0.0897	-1.1603	4.64
-501	0.0960	-1.154	2.30
-1000	0.105	-1.146	1.15
-1440	0.104	-1.146	0.796

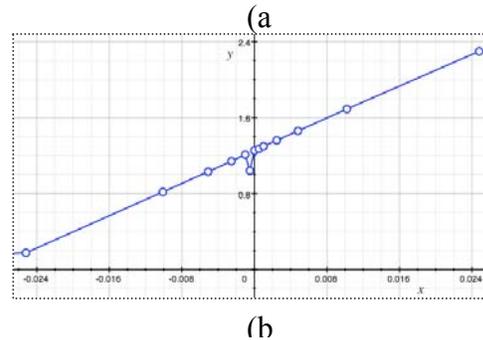


Figure 5: Graph of data from Table 1 (a), and the linear region therein (b).

- * In range of operation
- † In linear region
- ‡ Outlier

The average gain within the range of operation was 43.0. As a result, the choice of R_G is clearly incorrect. The calculation that R_G ought to be around 1 k Ω was made based on a previous circuit that used the INA122 differential amplifier, and had not been updated when the INA128 instrumentation amplifier was implemented. As a result, the gain equation used had been:

$$G = 5 + (200 \text{ k}\Omega / R_G)$$

but gain equation for the instrumentation amplifier is:

$$G = 1 + (50 \text{ k}\Omega / R_G)$$

As a result, the 1 k Ω resistor was replaced with one with a value of 270 Ω .

4.2. INA128 Test 2

For the second Test of the instrumentation amplifier, a 270 Ω resistor was used for R_G , in order to yield a gain of about 190. The results are shown in Table 2 and in Figure 6.

While the gain increased significantly, testing had been done on a breadboard, and the circuit needed to be implemented using surface-mount resistors. Since there were no 270 Ω surface mount resistors available, testing needed to confirm that the amplifier functioned with a 240 Ω resistor soldered onto a test board.

Table 2

V_i (mV)	V_O (V)	$V_O - V_{Ref}$ (mV)	G
-1,150	0.947	-308	0.268
-959	0.977	-278	0.290

-494	1.05	-202	0.409
-252	1.11	-140	0.556
-100	1.19	-688	0.688
-49.9	1.22	-358	0.717
-25.1	0.0801	-1,170	46.6
-9.82	0.0798	-1,170	119
-5.08 [†]	0.559	-695	136
-2.47 [†]	0.919	-336	136
-1.00 [†]	1,12	-131	131
-0.550 [†]	1.18	-70.5	128
0.003 [†]	1.26	3.49	1160*
0.490 [†]	1.32	69.8	142
1.05 [†]	1.40	146	139
2.50 [†]	1.60	346	138
5.07 [†]	1.95	701	138
11.0	2.30	1,050	95.5
25.1	2.30	1,050	41.8
49.0	2.30	1,050	21.4
100.	2.30	1,050	10.5
248	2.30	1,050	4.23
508	2.31	1,060	2.07
905	2.31	1,060	1.16
1,160	2.31	1,060	0.914

* In range of operation

[†] In linear region

[‡] Outlier

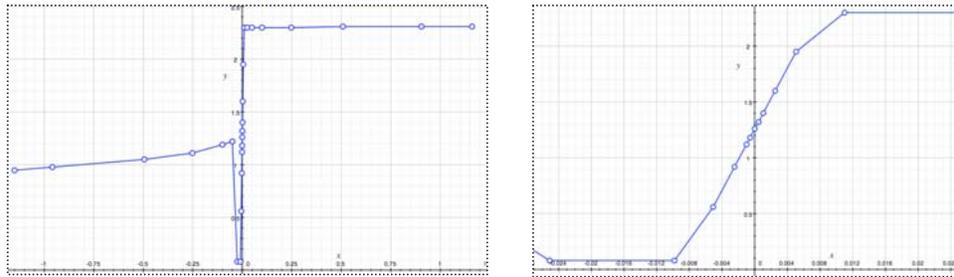


Figure 6: Graph of V_O from Table 2. All points (top) and relevant area (bottom)

4.3. INA128 Test 3

The third test of the instrumentation amplifier used the test board and sockets. Plugging only instrumentation amplifier into the test board yielded the results in Table 3 and Figure 7.

Table 3

V_I (mV)	V_O (mV)	$V_O - V_{Ref}$ (mV)	G
-104	88.0	-1,157	11.1

-11.6	82.3	-1,162.7	100
-1.48*	1,020	-225	152
0	1,245	0	N/A
1.09*	1,400	155	142
2.34*	1,590	345	147
4.60*	1,940	695	151
10.2	2,300	1,055	103
98.9	2,300	1,055	10.7

*Range of Operation

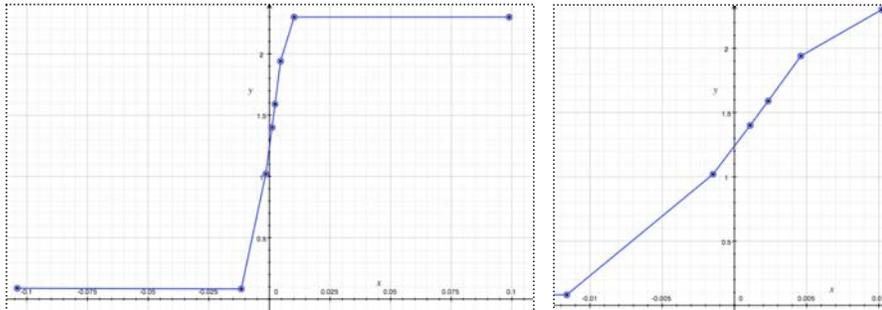


Figure 7: The full result (left) and the range of operation (right) from the data in Table 3

4.4. INA128 Test 4

In order to check consistent performance using different chips, 3 different INA128U chips were tested side-by-side. The results in Figure 8 show that all three chips followed almost the exact same path. The amplification and exact values that the test yielded were appropriate for the needs of the circuit.

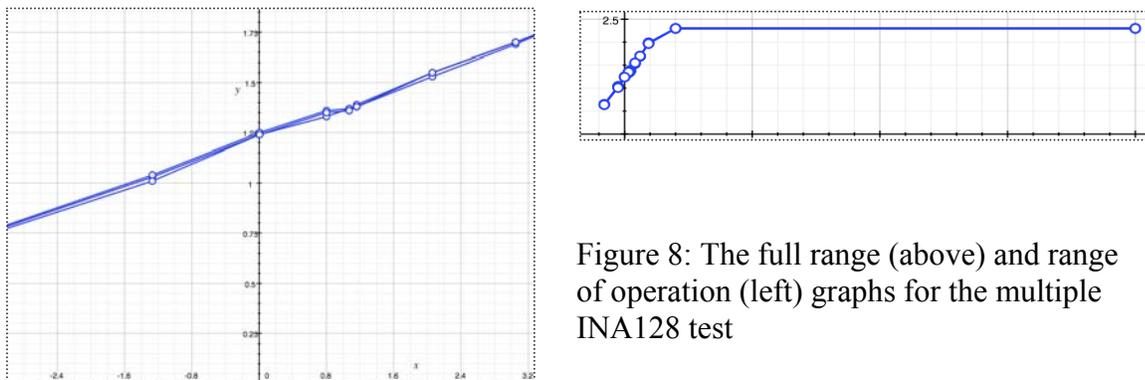


Figure 8: The full range (above) and range of operation (left) graphs for the multiple INA128 test

5. TESTING THE MICROCONTROLLER

Three aspects of the microcontroller had to be tested: the ADC input, the EUSART asynchronous communication with an external computer, and serial communication with a FLASH memory chip.

5.1. Testing the ADC

The first part of the microcontroller tested was the analog to digital converter. A sinusoidal signal was sent into the ADC port of the microcontroller. The microcontroller was programmed to take the digital conversion of the signal and to output it serially to one of its pins. Figure 9 shows the analog to digital converter functioning properly.

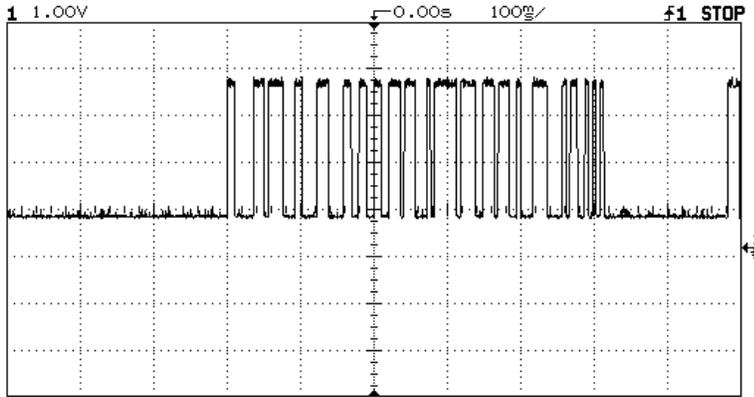


Figure 9: Output from the ADC test

5.2. Testing the EUSART

The second part of the microcontroller that was tested was the EUSART communication between the computer and the microcontroller. Initial testing (Figure 10) showed that the computer successfully sent information to the microcontroller; however, the microcontroller was not able to store that information. In order to solve this problem, a header was used. The header

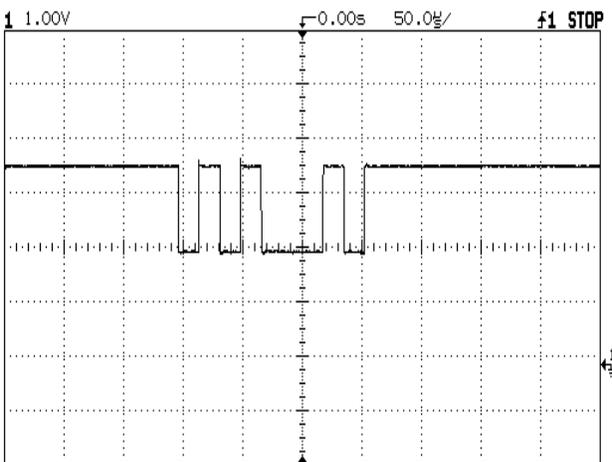


Figure 10: Transmission of an ASCII 'E'.
The signal reads 0/1010 0010/1 =
start/0x45/stop. 0x45 = 'E', and is sent LSB

allowed the computer to display the internal workings of the microcontroller. With the help of the header it was found that there were problems with the wiring of the test board. The circuit was designed such that the microcontroller and the computer were connected output-to-output and input-to-input, so that neither was receiving information that the other was transmitting. After soldering extra pins and inverting the EUSART cable, the problems with wiring were solved.

Appropriate changes were made to the schematic of the circuit so that future circuits will be functional without soldering in extra wires. Thus, the EUSART connection with the computer was shown to be functional.

5.3. Testing the SSP

The last element in the circuit that was tested was SSP communication between the microcontroller and the FLASH memory chip. SSP communication from the microcontroller to the FLASH appeared to be successful; however, using a header will be necessary to confirm that information is properly received and stored. The functionality of the header requires it to be powered by the computer, which is only able to output 5V, a voltage that would damage the FLASH memory chip. In order to use the FLASH and the header at the same time, a voltage converter needs to be assembled to protect the FLASH and to raise the output voltage of the FLASH to 5V. Development of such a voltage converter is underway, and will be completed soon. The serial communication is functional; however, the accuracy of the information sent to and from the FLASH is not yet confirmed.

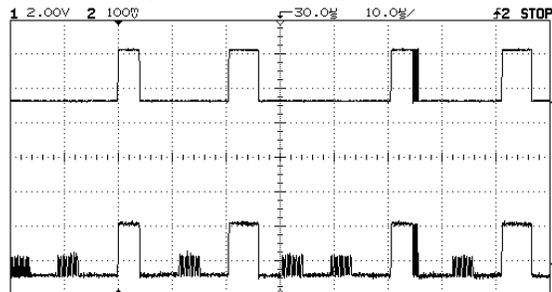


Figure 11: When the chip select signal (top) is high, the signal to the FLASH (bottom) stays high, but when the chip select signal is low, data is transferred to the FLASH

6. CONCLUSIONS

The entire circuit has not yet been tested, but finishing testing is well within sight. It will still be necessary to calibrate the device, and ensure that the pressure readings are accurate. One major reason to do so is because the INA128 testing was not done using the microcontroller. Reading outputs through the microcontroller is a better method for testing the amplifier because it provides a true measurement of what is being recorded. Additionally, testing while using the actual program that will ultimately run in the device needs to be completed. Figuring out how to progress through the routine of feeding the infants is necessary. Ultimately, the feeding device is on its way to completion.

7. NOMENCLATURE

ADC	Analog-to Digital Converter
EUSART	Enhanced Universal Synchronous/Asynchronous Receiver/Transmitter
LSB	Least Significant Bit
MSB	Most Significant Bit
SSP	Synchronous Serial Port

8. HARDWARE

Microcontroller: PIC16F690 SOIC by Microchip

Instrumentation Amplifier: INA128U by Burr-Brown of Texas Instruments

FLASH: S25FL016A by Spansion

Pressure Sensor: Various

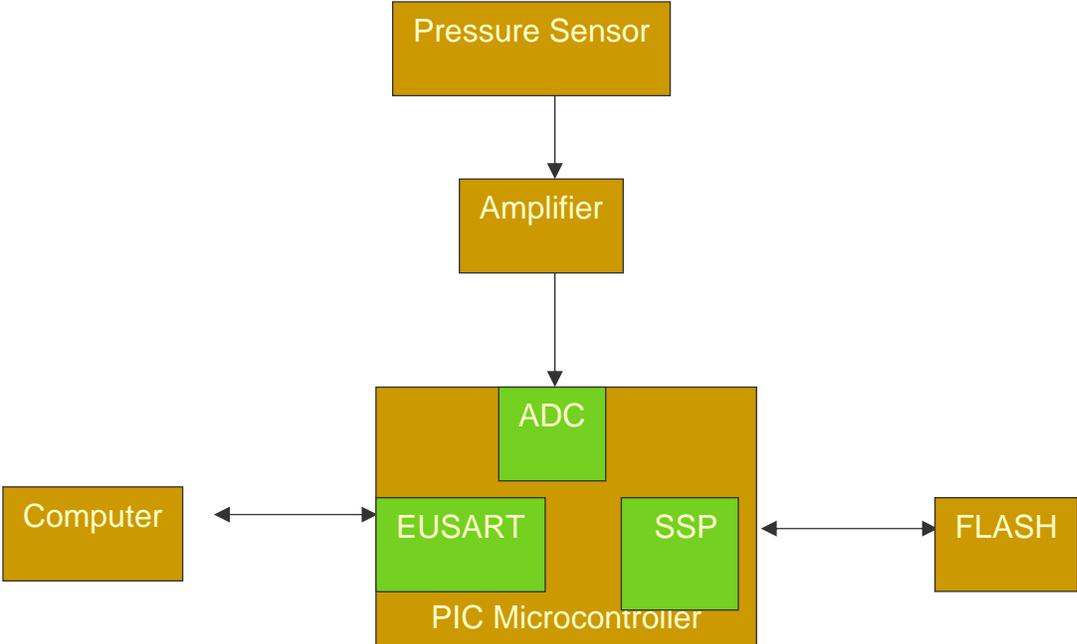
9. ACKNOWLEDGMENTS

I would like to thank Dr. Jay Zemel for being a helpful and supportive advisor, answering numerous questions, explaining the concepts involved in designing the device. Dr. Zemel provided guidance through the obstacles I faced doing my research, and made my work possible. I would also like to thank Sanket Doshi greatly; working in the lab with me, he was an excellent go-to-guy every day. I would not have been able to accomplish what I have without his help. I would also like to thank Leslie Chen and Preeti Rajendran for laying the groundwork and providing resources for my work this summer. Also, thank you to the RCA laboratory for allowing me to use their resources. Thanks are also due to Dr. Jan Van der Spiegel, and everyone else that organized SUNFEST 2008, for creating such an incredible opportunity and an outstanding program. Finally I would like to thank the National Science Foundation for funding the SUNFEST program, and for giving young engineering and science students, like myself, opportunities to conduct research as undergraduates.

10. REFERENCES

- [1] J.A. Martin and H. Kung. (2008, Apr. 4). Annual Summary of Vital Statistics: 2006. *PEDIATRICS* [Online]. 121(4) pp. 788-801. Available: <http://pediatrics.aappublications.org/cgi/content/full/121/4/788>
- [2] A.F. Bell, R. Lucas, and R.C. White-Traut. (2008). Concept clarification of neonatal neurobehavioural organization. *Journal of Advanced Nursing* [Online]. 61(5), 570-581. Available: <http://www3.interscience.wiley.com/cgi-bin/fulltext/119403755/HTMLSTART>
- [3] R.E. Kron, M. Stein, and K.E. Goddard. (1963, Mar.-Apr.). A Method of Measuring Sucking Behavior of Newborn Infants. *Psychosomatic Medicine* [Online]. 25(2), 181-191. Available: <http://www.psychosomaticmedicine.org/cgi/reprint/25/2/181>
- [4] Lau, C., Kusnierczyk, I. (2001, Jan.). Quantitative Evaluation of Infant's Nonnutritive and Nutritive Sucking. *Dysphagia* [Online]. 16(1), 58-67. Available: <http://www.springerlink.com/content/mgk91hnmnfuj66yx/>

11. APPENDIX Schematic of Information Flow Through Hardware



GROWTH OF CARBON NANOTUBES VIA CHEMICAL VAPOR DEPOSITION

NSF Summer Undergraduate Fellowship in Sensor Technologies
Alexei Matyushov (Physics) – Arizona State University
Advisors: Zhengtang Luo, A.T. Charlie Johnson

ABSTRACT

Carbon nanotubes (CNTs) are an unusual, tubular form of carbon, composed of a lattice of carbon hexagons rolled into a tube of nanometer-scale diameter. After the finding of CNTs in 1991 by Iijima on the cathode of an arc-discharge instrument, nanotechnology and nanoscience expanded tremendously. The subsequent discoveries of carbon nanotubes' unique properties fueled research into all aspects of CNTs, from fundamental physics, to synthesis techniques, to technological applications. However, synthesis remains a challenge in almost every aspect, including control of length, diameter, and orientation. Also elusive is control of resulting nanotubes' electric properties, which directly depend on variations in molecular structure. The other major challenge is production on a large scale, also currently far from accomplishable. Thus, we are concerned with optimizing methods to produce more consistent and higher yields. Specifically, we used chemical vapor deposition (CVD) with a focus on maximizing CNT length, alignment, and density of tubes per area. CNTs of lengths around 1mm have been achieved on silicon wafer chips, and two parameters – method of catalyst application, and catalyst concentration – have been optimized to achieve CNTs of this length. Additionally, moderate alignment of CNTs was achieved on ST-cut quartz chips and an apparent correlation between catalyst particle size and nanotube growth was found.

Introduction

Carbon nanotubes were first discovered in 1952 by Radushkevich and Lukyanovich.[1] However, only after their “rediscovery” in 1991 by Iijima in the wake of the 1985 finding of fullerenes, did carbon nanotubes gain widespread attention and burgeon into the intensely researched area that these materials are today. [2] CNTs have been found to possess a number of extraordinary properties, offering new capabilities and performance beyond the possibilities of heretofore known materials. These discoveries have led researchers to imagine technological applications such as nanoscale robots and medical devices, nanoscale electric circuits, and perhaps even such revolutionary technologies as the science fiction concept of a space elevator. [4] But before such things can materialize, nanotechnology still needs to progress a long way. The great obstacle now is that current CNT synthesis is poorly controlled and is not possible on a large scale. CNTs synthesized in labs are often more or less randomly oriented, variously straight or curled, and with varying lengths and diameters. This results in uncontrolled electronic properties (among other effects), with approximately two thirds of CNTs being semiconducting and approximately one third, metallic. And the quantities produced are small, which keeps the price of nanotubes exceedingly high: \$2,000/g for high quality single-walled CNTs. [2] Hence, synthesis techniques must be optimized to produce CNTs controllably and cheaply on a large scale.

The three most important methods of synthesis currently used are arc discharge, laser ablation, and chemical vapor deposition (CVD) [2]. Of these, CVD shows the most promise for achieving the goal of mass production; despite that, however, the nanotechnology field is still young enough that there are no comprehensive results elucidating the relationship between key parameters (e.g. temperature, catalyst, carbon source) and the resulting nanotube properties (e.g. length, diameter, morphology). [2]

Hence, we focused on optimizing CVD parameters with the goal of growing single-walled carbon nanotubes with maximum length, density, and favorable alignment and geometry (i.e. straight nanotubes aligned in the same direction). More specifically, we mostly used a technique involving ultralow feeding gas flow, which makes possible the growth of significantly longer (>5mm) SWNTs than other methods. [5] Not only would achieving these goals contribute to advancing the field of nanotechnology, but it would also provide useful tools for fundamental science of carbon nanotubes. Such long tubes are easier to manipulate, yet they retain the single-molecule characteristics of shorter nanotubes. Thus, research on CNTs can proceed more rapidly by using such subjects, which are easier to study.

Our experiments can be divided into three major parts: we used an ultra-low gas flow CVD process with the catalyst, Iron(III) nitrate, nonahydrate, and silicon wafer chips as the substrate; the “standard” CVD process with the same catalyst, but with ST-cut quartz chips as the substrate; and again the ultra-low gas flow CVD process with silicon wafer chips, but with a catalyst mixture consisting of Ruthenium Chloride, Iron(III) nitrate, nonahydrate, and 11nm-diameter alumina nanoparticles. In all cases, the carbon source was methane gas. We used a liquid catalyst mixture with concentration of catalyst often on the order of 50mg/L but with some variation which usually was no more than a factor of 1/10. The method of fluidized catalyst application was also varied between using a syringe and a spin-coater. The results were analyzed qualitatively, as we looked for the presence and quality of CNTs on the wafer chips with a scanning electron microscope, and also quantitatively with the help of the image analysis software, Image J.

Background

I. Carbon Nanotube Structure

One basic principle of CNTs is the distinction between multiwalled (MWNTs) and single-walled (SWNTs) nanotubes. MWNTs are composed of concentric tubes of graphene (a single layer of carbon arranged in the same crystalline lattice pattern as graphite), with inter-layer spacing of 3.4\AA , like in graphite.

These can have many shells and range in diameter from a few to several

hundred nanometers. [3] The first CNTs discovered were multiwalled. In 1993, the first single walled nanotubes were found (like in Figure 1). These nanotubes have the structure of a single rolled up graphene sheet and have much smaller and more uniform diameter than MWNTs, typically around 1 nanometer. [3] Although CNTs are not formed via the roll up of a graphene sheet, it is nevertheless a useful visualization in understanding nanotube geometry. The degree of “twist” resulting from where the hypothetically rolled up graphene sheet overlaps to form the nanotube determines the chirality of a CNT. Chirality marks the difference in CNTs between zigzag, armchair, and chiral types. (See Figure 1.) Quantitatively, chirality is measured by the chiral -- or wrapping -- vector, C_h , as shown in Figure 2 connecting the points A and A', which would trace the nanotube's circumference if the graphene were rolled up. This way, the nanotube geometry can be fully defined by integers n and m in equation 1, where a_1 and a_2 are unit vectors (which are also shown in Fig. 2).

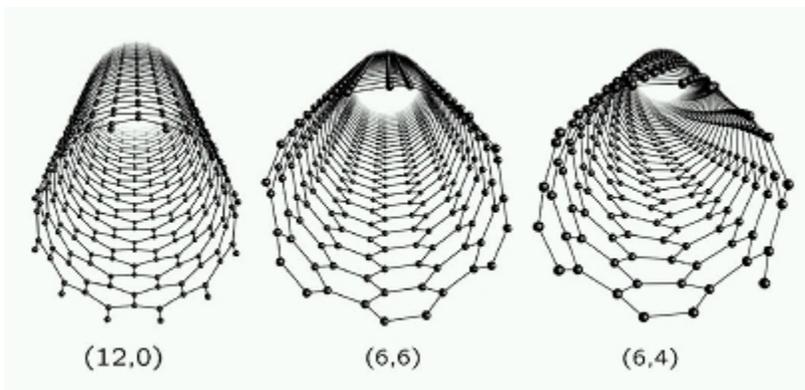


Figure 1: Single-walled carbon nanotubes. (12,0) is a zigzag CNT, (6,6) is an armchair type, and (6,4) is chiral. [3]

$$\mathbf{C}_h = n\mathbf{a}_1 + m\mathbf{a}_2 \quad (1)$$

Thus, the chirality of a CNT is specified simply by (n,m) , and the magnitude of \mathbf{C}_h gives the circumference. In Figure 2, the example of a $(5,3)$ nanotube is shown. Corresponding to the (n,m) chirality is the chiral angle (θ in Fig. 2), which can range from 0° to up to 30° due to the hexagonal geometry of the graphene sheet. All SWNTs of the configuration $(n,0)$ ($\theta = 0^\circ$) are zigzag tubes, whereas configurations (n,n) ($\theta = 30^\circ$) are armchair tubes. They are named such due to the zigzag and armchair patterns, which can be seen along each tube's circumference in Figure 1 and along the dashed lines in Figure 3. All remaining nanotubes, like the $(6,4)$ one in Fig. 1, are referred to as chiral. The chirality is also the determinant of CNT electronic properties. In addition to equation 1, the following qualifiers are needed:

$$|n - m| = 3q, q \neq 0 \quad (2)$$

$$|n - m| = 3q \pm 1, q \neq 0 \quad (3)$$

If equation 2 is satisfied, the CNT is metallic; a CNT fitting equation 3 is semiconducting.

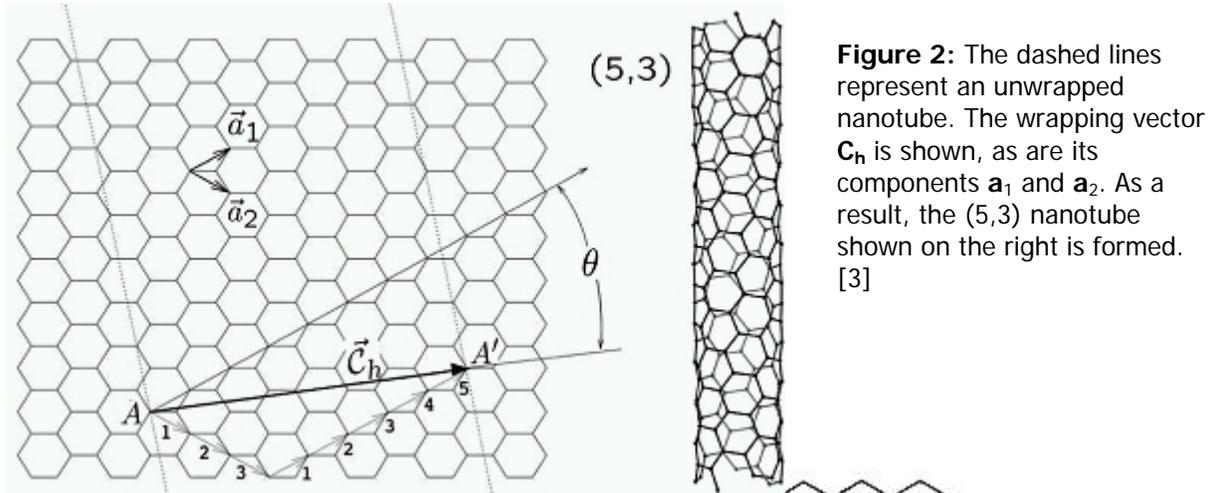


Figure 2: The dashed lines represent an unwrapped nanotube. The wrapping vector \mathbf{C}_h is shown, as are its components \mathbf{a}_1 and \mathbf{a}_2 . As a result, the $(5,3)$ nanotube shown on the right is formed. [3]

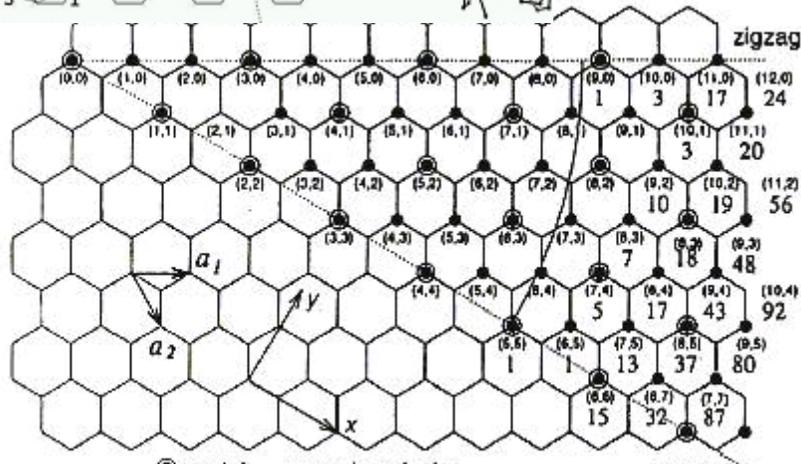


Figure 3: A map of possible (n,m) combinations and the resulting metallic or semiconducting properties. [8]

II. Carbon Nanotube Properties

With this structure come extraordinary properties. The sp^2 carbon bonds in the hexagon lattice grant CNTs superb strength in the axial direction. Compared to steel, the Young's modulus for single-walled carbon nanotubes can be more than 8 times greater, as high as 1,700 GPa, and the tensile strength is over 50 times that of steel, at 75 GPa. At the same time, the hollow CNTs have a low density: 1.3 to 1.4 $g\cdot cm^3$ [6] compared to typical steel's approximate 7.8 $g\cdot cm^3$. This makes CNTs even more impressive as potential structural materials, resulting in a specific strength of up to 48,000 $kN\cdot m\cdot kg^{-1}$. By comparison, carbon steel's specific strength is 154 $kN\cdot m\cdot kg^{-1}$.

The electronic properties are also striking. As already mentioned, CNTs can be both metallic and semiconducting, which means they can be employed as both the transistors and the current-carrying wires in new, nanoscale circuits. In both capacities, they would exceed the performance of currently used materials such as copper and silicon. As current carriers, metallic nanotubes can sustain current densities at least 1,000 times as great as those sustained by copper. Current densities on the order of $10^9 A/cm^2$ are routinely achieved in our group, whereas copper wires burn out at around $10^6 A/cm^2$. [6] As semiconductors, CNTs could offer great flexibility. The band gap of nanotubes varies from as high as that of silicon to zero, like that of a metal, based on diameter and chirality. Thus, the most appropriate semiconductor for a given purpose could be had just by optimizing nanotubes to the correct geometry. Furthermore, it has been shown that integrating semiconducting nanotubes into circuit elements, like field-effect transistors (FETs), can yield devices with electronic behavior which is quite similar to that of typical, silicon transistors. Like their silicon counterparts, such nanotube FETs can experience changes in conductivity by a factor of one million or more. However, the nanotube cross-sectional dimensions of approximately one nanometer allow such FETs to switch at much lower power than the silicon FETs. It is predicted that a nanoscale switch could run at one terahertz or more. [6]

For technological applications, especially in electronics, the prospects of carbon nanotubes are further enhanced by excellent thermal properties. CNTs are stable at up to 2800°C in vacuum and 750°C in air, comparable to the 600°C to 1,000°C melting range of metal wires in microchips. [6] More impressively, CNTs also display some of the best thermal conductivity. Single-walled carbon nanotubes exhibit a heat-transmission rate ranging from 1750 to 5800 $W\cdot m^{-1}\cdot K^{-1}$, and multiwalled carbon nanotubes show a rate of 3000 $W\cdot m^{-1}\cdot K^{-1}$. [2] This is similar to, or better than, the best quality diamond's heat transfer rate of 3320 $W\cdot m^{-1}\cdot K^{-1}$, and up to a factor of 15 higher than the 385 $W\cdot m^{-1}\cdot K^{-1}$ of copper, one of the better heat conductors commonly used in current electronics. Thus, carbon nanotubes could support much denser (i.e. faster) circuits than the present edge of microprocessor technology.

Materials and Methods

Chemical vapor deposition (CVD) was used to grow carbon nanotubes. The carbon source was methane gas delivered through a flow meter system and a furnace in which the substrate(s) for the CNTs were located. The substrates used were mostly silicon wafer chips (SiO_2/Si), but ST-cut quartz chips (SiO_2) were also experimented with. Chips were made, typically around 1-2 cm^2 in size, by cutting and snapping wafers using a diamond cutter. These chips were placed inside a quartz tube which was in turn placed into the furnace and tightly connected and sealed to the gas flow tubes. In addition to methane, the system was also infused with argon and hydrogen.

To generate CNTs on a substrate via CVD, a catalyst is needed. For this role, we (almost always) used the catalyst, Iron(III) nitrate nonahydrate, mixed in isopropanol. The starting point was to place 10mg of the catalyst crystals into a 50mL centrifuge tube and to fill that to 50mL with isopropanol. The next step was not very standardized. The objective was to mix the catalyst and isopropanol until no more solid particles could be seen by the eye, and hopefully at this point the catalyst particles in the mixture would be broken down to sizes approximately matching typical nanotube diameter sizes so that nanotubes could grow on those particles. However, we are not certain what the best method to achieve this is, and there is no strict procedure. Most of the time, the centrifuge tube was simply shaken by hand for about 5 minutes until particles could no longer be detected and only liquid was visible. This mixture was then diluted in several variations, since the concentration was one of the parameters varied and tested in this project. Some of the concentrations that were tried were $1/4^{\text{th}}$, $1/5^{\text{th}}$, $1/10^{\text{th}}$, $1/25^{\text{th}}$ and $1/125^{\text{th}}$ of the initial catalyst and isopropanol mixture.

Our group makes use of several CVD procedures designed for growing various types of CNTs. Broadly speaking, two such procedures were used in this work. One of these, the most commonly used procedure in our group, typically produces dense growth of short CNTs, covering much of the silicon wafer chip. This involves using the concentration of 50mg/L of catalyst ($1/4^{\text{th}}$ of the initial catalyst mixture) and spin-coating 15 drops of it onto a silicon wafer chip spinning at 3,000 rpm. Following that is a fairly short and simple process using the CVD furnace. This process will be referred to as the “standard” CVD method. While this process was used occasionally, a different, more specialized, process was used for most of this research work. This second method is longer, more complex, and is designed to produce much longer CNTs. Most importantly, the difference is that the methane flow rate is on the order of 1,000 times lower than that employed in the standard process, along with some other differences in argon and hydrogen gas flows, the temperature used, and time needed. This method will be referred to as the ultra-low gas flow method.

Attempts were made to optimize the latter procedure further to produce higher quality CNTs than previously attained. To this end, the concentration of catalyst and the method of catalyst application onto the wafer chip were varied. The two main ways in which catalyst was applied to the substrate were using a syringe, and spin-coating. When using a syringe, the catalyst was applied directly to the chip by brushing the tip of the needle against it and gently squeezing out the liquid mixture. This introduced the parameter of where on the chip to apply the catalyst mixture. Logically, it was typically applied to the polished surface of the chip but only on one side, which would then be the side closest to the incoming gas flow. That way, the nanotubes would hopefully grow in the general direction of the gas flow across the surface to the other side of the chip. The amount of catalyst discharged, making either a broad streak or very

thin, was also varied. Another variation tried was to apply the liquid only to the thin edge of the chip, not touching the chip's surface at all.

The other major method of catalyst application, spin-coating, was used to apply catalyst over the entire chip. This method introduced many other variable parameters. In addition to the catalyst concentration (as always), other parameters were the spin speed (rpm), the duration of spinning, and the number of drops of catalyst added.

The ready chips were then inserted and pushed one by one into position in a quartz tube. It was made certain to place the chips closer to one end of the tube, since experimental wisdom among our group held that the temperature was more accurate in one half of the furnace than the other. After completion of the CVD process, the last step was observation of the results using the scanning electron microscope (SEM). Most of the results were analyzed qualitatively only: whether CNTs could be found on the chip or not, and if so, the typical length was estimated roughly and compared qualitatively the degree of alignment and nanotube density to previous results.

Other experiments: a.) ST-cut quartz substrate and b.) catalyst composed of Ruthenium Chloride/Iron (III) nitrate, nonahydrate/alumina nanoparticles

In addition to using the above procedures with silicon chips, two other major variations were done: using ST-cut quartz as the substrate while using the standard CVD method, and using a different catalyst mixture with the ultra-low gas flow CVD process. With the quartz chips, in addition to qualitative analysis, the SEM images were analyzed quantitatively using the image analysis software, ImageJ, in order to quantify catalyst particle size and CNT distributions. This was to investigate whether the data supports the hypothesis of a relationship between catalyst particle size and the likelihood of CNT growth from those particles.

Lastly, a different catalyst was tried. This was a mixture of Ruthenium Chloride, Iron (III) nitrate, nonahydrate (same as before), and alumina nanoparticles of an average diameter of 11nm. The ultra-low gas flow process was used (with some increases in gas flow rates), and both spin-coating and syringe application was performed with different samples.

Results and Discussion

I. Ultra-low gas flow using Iron(III) nitrate, nonahydrate catalyst on SiO₂/Si chips

The method of growing long nanotubes on SiO₂/Si chips using the Iron(III) nitrate, nonahydrate catalyst and the ultra-low gas flow procedure proved to be very problematic. The process was highly reproducible at first, yielding carbon nanotubes at every attempt. This success rate lasted until about early July. Then, from that point until the end of this research project (early August), it consistently failed. Almost without exception, no nanotubes were produced in all later attempts, despite careful repetitions of previously successful combinations of catalyst concentration and application method.

Thus, during the former period of the project, progress was steadily made towards optimizing parameters such as catalyst concentration. Consequently, an increase in nanotube lengths was observed, as can be compared in Figures 4 and 5. Figure 4 shows a result of an early experiment. Figure 5 displays a CNT from the last successful trial, which also produced the

longest CNTs found using this method. The particular CNT shown in Fig. 5 was the longest one observed on that sample, and it measured 660 microns. (Although the length of the CNT in Figure 4 was not measured, it is evident that it is much shorter. It fit entirely within one screen, whereas the CNT in Figure 5 required a composition of 5 screenshots from the SEM, and both images were taken at similar magnifications.) This sample was achieved using 5 drops of catalyst, at a concentration of 8mg/L, spin-coated onto a silicon chip at 6,000 rpm.

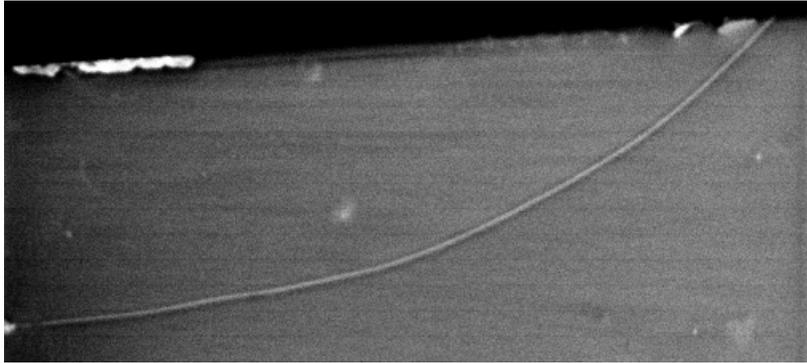
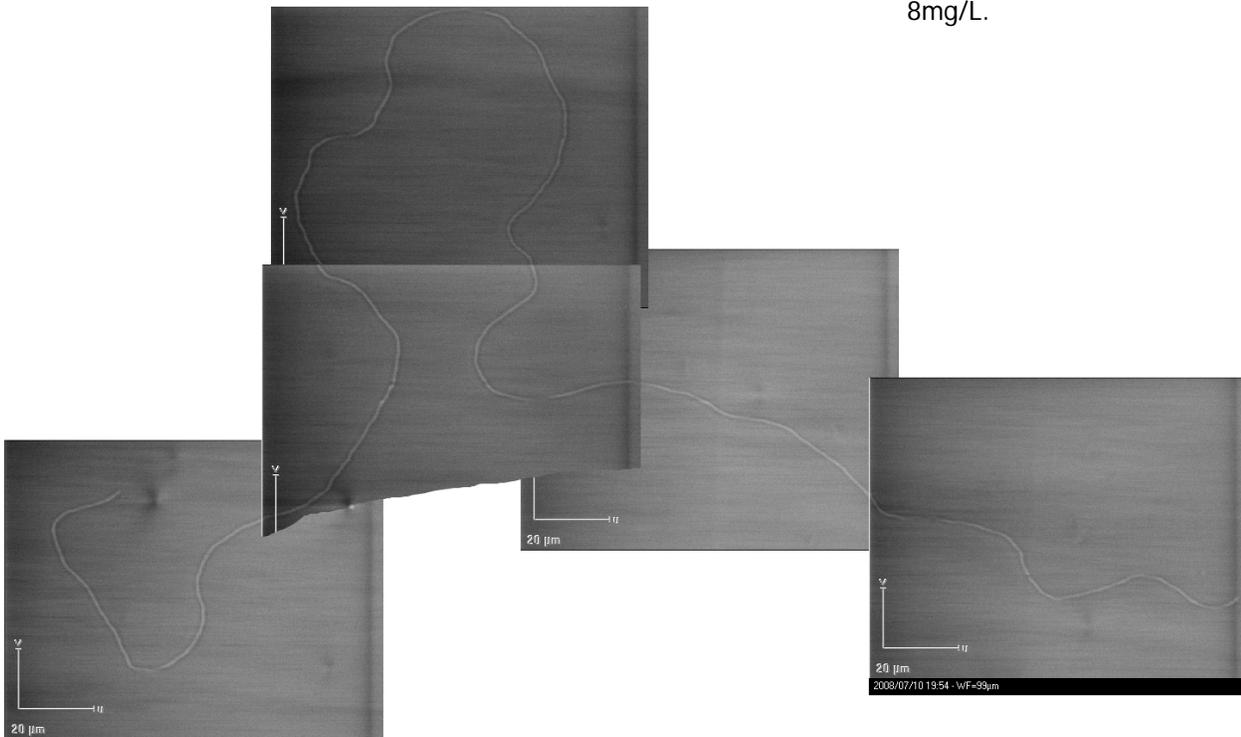


Figure 4: (Left) An early result growing nanotubes with the ultra-low gas flow process on silicon

Figure 5: (Below) The longest CNT achieved using ultra-low gas flow CVD process on silicon chips. Length measures 660 μm . Catalyst was applied via spin-coating at a concentration of 8mg/L.



The chip from which the image in Figure 5 was taken contained quite a number of long nanotubes similar to that one. (They were not counted; the results were analyzed mostly qualitatively.) They were, however, fairly sparse. It appeared that the length of nanotubes produced after spin-coating the chip and using the ultra-low gas flow process was inversely correlated to the density of nanotubes. That is, the longest nanotubes (like in Fig. 5) seemed to be present on chips with the fewest and most sparsely spaced nanotubes. If that is the case, it is an unfortunate trade-off: CNT length is gained at the expense of nanotube density per area, which is inconsistent with the goals of this project of maximizing both length and density simultaneously.

The third objective during this project, nanotube alignment, was not met using the ultra-low gas flow process with Iron(III) nitrate, nanohydrate catalyst. The CNT in Figure 5 is not at all straight, and even though it can be said that it is crudely aligned in the left-to-right direction, this direction was not aligned relative to the gas flow direction. In fact, the gas flow direction on that particular sample was in the up-and-down direction. Similarly, other CNTs observed on that sample and all other samples had random alignment, instead of preferring a particular direction. In addition, they were often curvy or coiled similar to the CNT in Figure 5 or even more so. Hence, alignment was uncontrolled.

In the latter period of the project, after early July, the ultra-low gas flow process with Iron(III) nitrate, nanohydrate catalyst, and SiO₂/Si chips as substrates, consistently failed to produce CNTs. Instead, results such as in Figure 6 could often be observed. These appear to be the catalyst particles. If so, then it seems that lack of nanotube growth is due to these particles being too big. Theories hold that when nanotubes grow from catalyst nanoparticles, the CNTs match the diameter of the particles. But in these images, the particles are on the order of a few to several microns in diameter, far exceeding the diameter range of CNTs, which is on the order of nanometers. The crucial question then is, why was the ultra-low gas flow process successful and highly repeatable in the earlier period but then abruptly stopped yielding CNTs after early July. No answer to that has been determined in the course of this work, and figuring out this puzzle would certainly be very helpful.

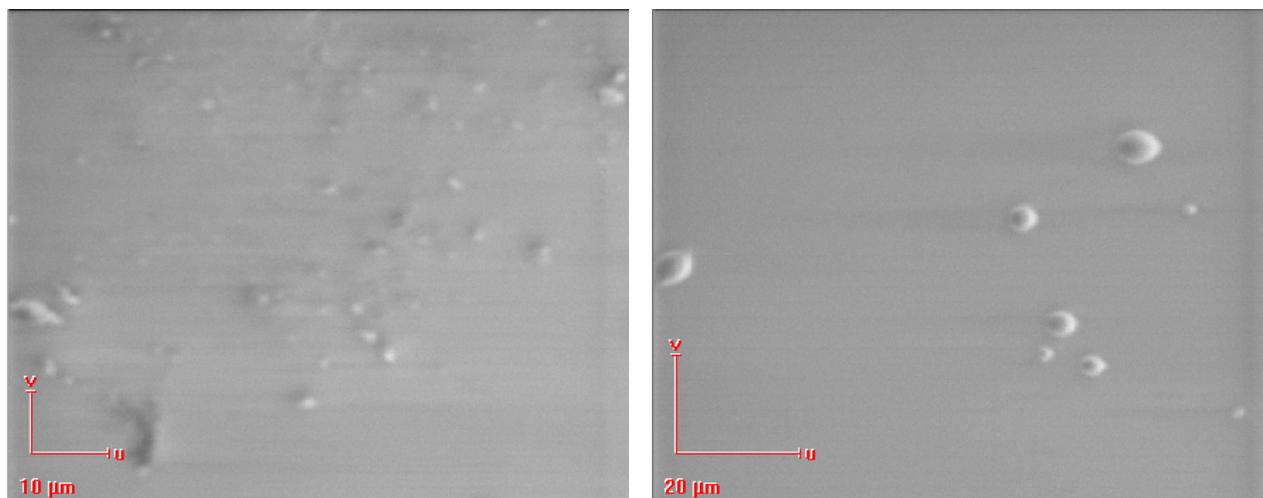


Figure 6: Unsuccessful CNT growth attempts using ultra-low gas flow with silicon chips. Oversized catalyst particles are visible but no CNTs.

II. CNT growth on ST-cut quartz

CNTs were produced successfully using the “standard” growth method (using 50mg/L of catalyst concentration and spin-coating at 3,000 rpm with 15 drops) on ST-cut quartz chips. The nanotube alignment was inconsistent. Some areas of the quartz chip had very straight nanotubes which were aligned in a particular direction, but other areas featured non-straight nanotubes with no discernible alignment (compare Figures 7 and 8). The other notable aspect of these results is that even where well-aligned and straight nanotubes were found, they were fairly sparse. These results are inferior in terms of both alignment and density to those reported by Yuan et al. who also used ST-cut quartz as the substrate (see Figure 9). [7] This raises another question of why these results are so different despite using the same substrate. Likely it is due to the fact that different catalysts were used (Yuan et al. did not use iron).

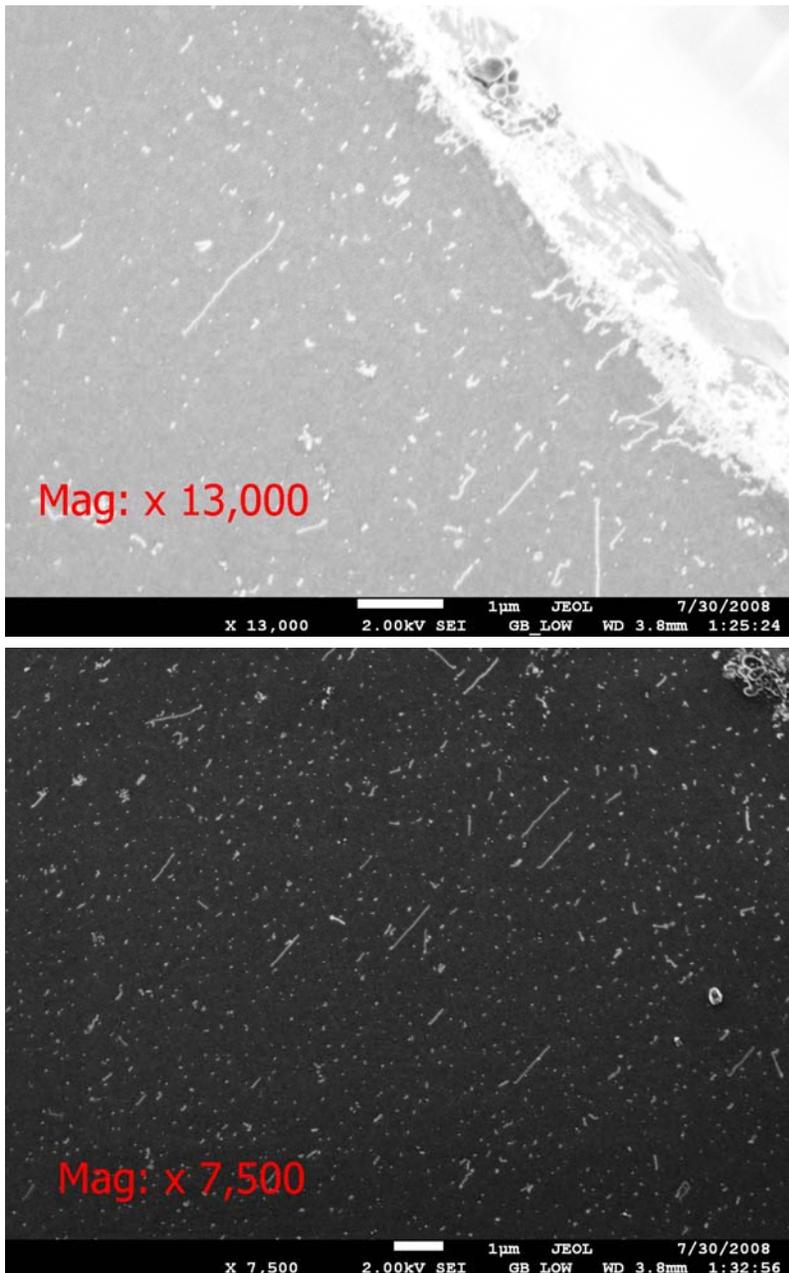


Figure 7: ST-cut quartz substrate. Two examples of images where straight nanotubes aligned in a specific direction can be observed.

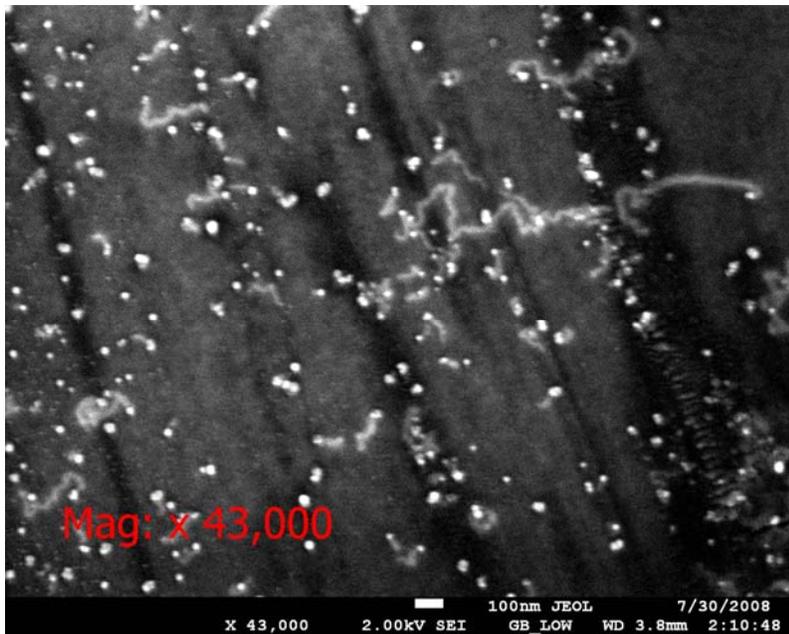


Figure 8: Non-straight CNTs on ST-cut quartz showing little or no directional alignment

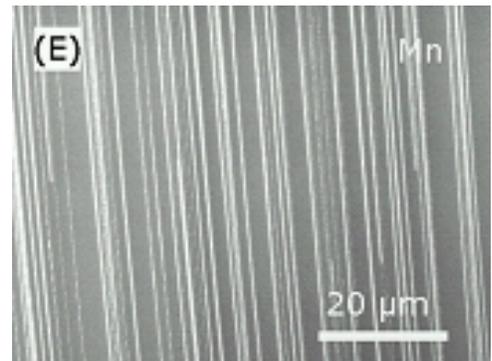


Figure 9: CNTs produced by Yuan et al. on ST-cut quartz using Mn catalyst

Digital analysis results using the software, ImageJ, are shown in Figures 10 and 11. The mean area of the catalyst particles analyzed was 927 nm^2 . Among this set, the subset of particles from which nanotubes grew had a mean area of 705 nm^2 . The mean of nanotube-containing particles is lower than the overall mean particle area, and the size distribution of nanotube-containing particles is also narrower than the overall distribution. This data appears to confirm that there is a relationship between catalyst particle size and the likelihood of CNT growth from a particle because nanotubes grew on particles mostly within the range of 150 to 1050 nm^2 . It also fits with the lack of CNTs shown in Figure 6, since the nanoparticles there have diameters on the order of microns and are therefore clearly much larger than the size range in Figure 11.

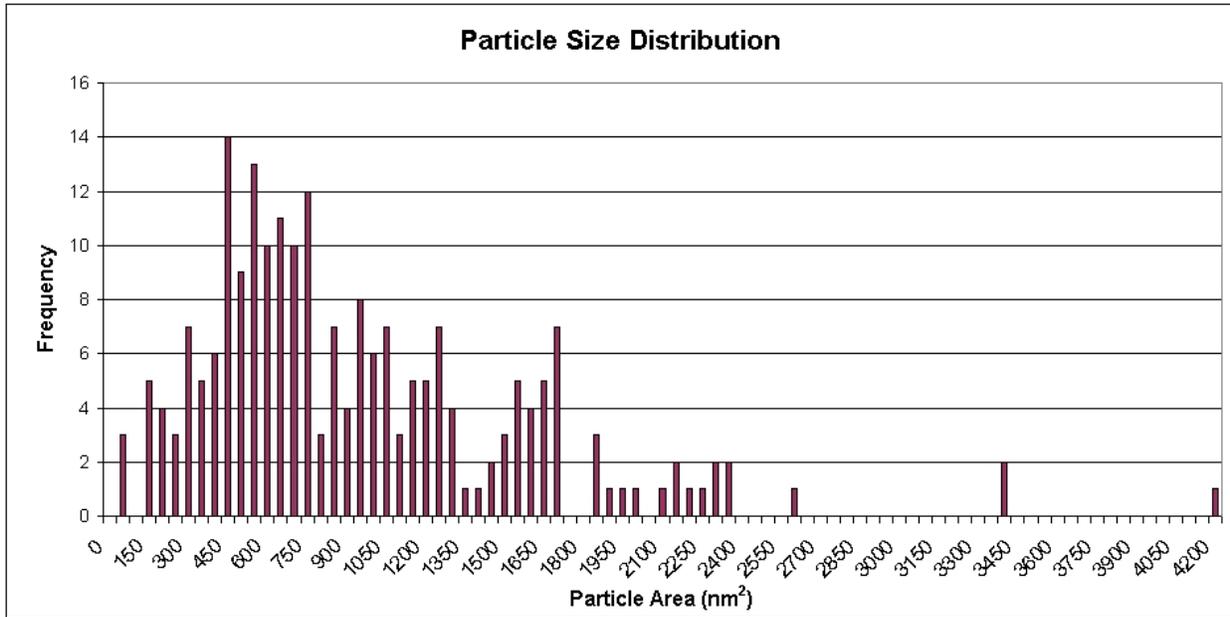


Figure 10: Size distribution of catalyst nanoparticles observed on ST-cut quartz substrate.

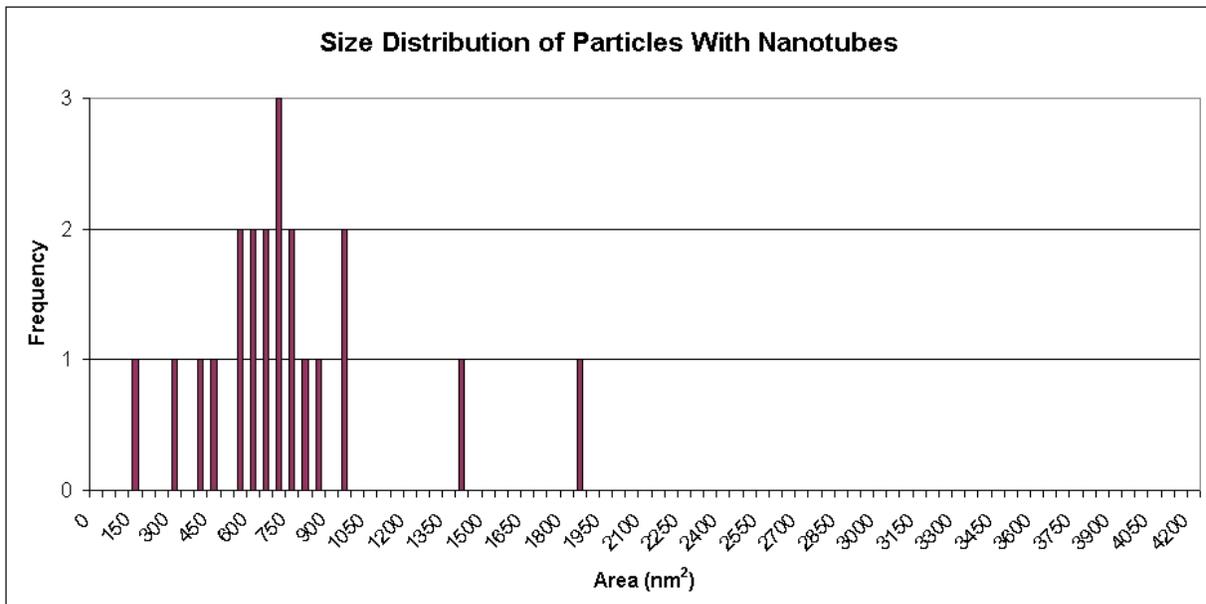
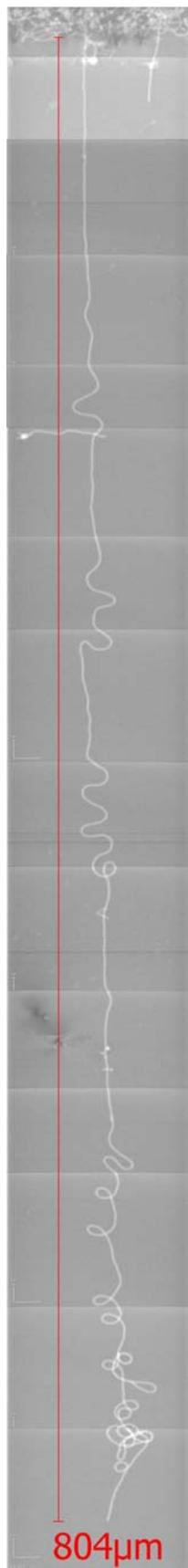


Figure 11: Size distribution of catalyst nanoparticles from which CNTs grew, as observed on ST-cut quartz substrate.

III. Ultra-low gas flow using Ruthenium Chloride and Iron(III) nitrate, nonahydrate, catalysts and 11nm alumina nanoparticles on SiO₂/Si chips



The last experiment, which employed the ultra-low gas flow method and used a liquefied mixture of Ruthenium Chloride, Iron(III) nitrate, nonahydrate, and 11nm-diameter alumina nanoparticles as catalysts on silicon chips, yielded the best CNTs. The most important goal was length, and the nanotubes produced by this method exceeded previous results in this respect. The longest CNT found is shown in Figure 12, which from end to end stretches for at least 804 microns (the upper end is ill-defined, so measurement is only approximate). If it were uncoiled into a straight line, it would certainly be longer than 1mm. The sample containing this nanotube was made by applying the liquid catalyst mixture by swiping the syringe across one edge of the chip to produce a thin streak of the mixture along the edge. That edge was then positioned so that it would be the first one over which gas flowed within the quartz tube. Long nanotubes such as this one could be found at the border of the catalyst zone, where the clean area of the chip began. Within the catalyst zone, there were catalyst particles and nanotubes densely packed and interwoven, so the nanotubes had no room to attain long length.

The CNTs found on this sample also showed alignment, the second goal of this project. The nanotube in Figure 12 follows a rather straight trajectory in its overall shape (disregarding the meanders and coils along the way), and this direction was parallel with the direction of gas flow. The same traits were also observed in many other long CNTs found within the transition zone between the catalyst-covered area and the uncovered silicon chip area, although there were also many exceptions. This marked the first time that significant alignment was observed in CNTs produced by the ultra-low gas flow method.

Figure 12: The longest CNT achieved during this project: 804 microns long. Ultra-low gas flow method was employed using a catalyst mixture consisting of Ruthenium Chloride, Iron(III) nitrate, nonahydrate, and alumina nanoparticles of 11nm diameter placed onto silicon chips.

Conclusions

This project attempted to grow carbon nanotubes as long as possible, as aligned as possible, and as dense as possible. The best results were achieved using the Ruthenium Chloride, Iron(III) nitrate, nanohydrate, and alumina nanoparticle catalyst mixture under the ultra-low gas flow process. This combination of catalyst and CVD process yielded long nanotubes, the longest of which was over 800 microns, and many of the nanotubes exhibited alignment parallel to the gas flow. However, it remains a mystery why the ultra-low gas flow process was so problematic with the Iron(III), nanohydrate recipe that it consistently yielded CNTs in the early half of the research project and then abruptly stopped yielding any CNTs in the latter half. The other important result to investigate further is the shortcoming of the results on the ST-cut quartz compared to those reported by others – whether it truly has to do with the difference in catalysts or not. The last major point drawn from this study is the data supporting the prevailing theory that catalyst particles need to be of a certain size to prompt nanotube growth. Thus, it appears that researchers need to continue seeking ways to maximize the presence of appropriately sized particles.

Acknowledgements

I thank my mentors, Zhengtang Luo and Charlie Johnson for working with me and contributing to my project. I am grateful to NSF for funding this REU opportunity and to University of Pennsylvania for hosting it. I am also grateful to Jan Van der Spiegel for administering and assisting with the REU program.

References

- [1] Guest Editorial, "Who Should Be Given the Credit for the Discovery of Carbon Nanotubes?," *CARBON*, vol. 44, 2006, pp. 1621-1623.
- [2] C. See and A. Harris, "A Review of Carbon Nanotube Synthesis Via Fluidized-Bed Chemical Vapor Deposition," *Ind. Eng. Chem. Res.*, vol. 46, 2007, pp. 997-1012.
- [3] J. Charlier, X. Blase, and S. Roche, "Electronic and Transport Properties of Nanotubes," *Reviews of Modern Physics*, vol. 79, May. 2007, pp. 677-711.
- [4] E. Drexler, "Nanotechnology," *AccessScience@McGraw-Hill*;
<http://www.accessscience.com>.
- [5] Z. Jin et al., "Ultralow Feeding Gas Flow Guiding Growth of Large-Scale Horizontally Aligned Single-Walled Carbon Nanotube Arrays," *Nano Letters*, vol. 7, 2007, pp. 2073 - 2079.
- [6] P.J.F. Harris, "Carbon Nanotubes," *AccessScience@McGraw-Hill*;
<http://www.accessscience.com>.
- [7] D. Yuan et al., "Horizontally Aligned Single-Walled Carbon Nanotube on Quartz from a Large Variety of Metal Catalysts," *Nano Letters*, vol. xx, Jun. 2008.
- [8] M. Daenen et al., "The Wondrous World of Carbon Nanotubes," *Eindhoven University of Technology*, Feb. 2003.

DEVELOPMENT OF A MICRO PCR REACTOR FOR LAB-ON-A-CHIP DEVICES

NSF Summer Undergraduate Fellowship in Sensor Technologies
Erika Martinez Nieves (Art and Science Department) - University of Puerto Rico
Advisor: Haim H. Bau

ABSTRACT

The process of performing a Polymerase Chain Reaction (PCR) on a lab-on-a-chip (LOC) device is difficult to achieve. When treating a patient, a timely and accurate result is needed, but obtaining those results is challenging when using a conventional PCR machine. However, if the ability to detect molecules such as DNA during the PCR process happening inside the LOC device was accomplished, then the desired results would be obtained. One potential solution is the use of Real Time-Polymerase Chain Reaction (RT-PCR). Using SYBR Green as the principle dye for the RT-PCR process, this paper discusses the use of acrylic chips and RT-PCR tubes, exposed to different experimental parameters, being analyzed with the RT-PCR technique. According to our studies, better signals can be obtained from a RT-PCR melting curve with the use of FTA membranes than with the use of other membranes such as alumina and Porex® membranes. Good results were obtained from samples taken from chips that have experienced a PCR process with and without FTA membranes. Also, the acrylic plastic chip did not show signs of leakage or any kind of plastic damage while using normal PCR program settings. More importantly, RT-PCR tubes and chips were exposed to the different wash and incubation steps that would be performed on a LOC. Through the RT-PCR machine, good melting curves results were obtained from the vials but for the chips, further research is needed.

INTRODUCTION

In developing countries, a diseased patient may not receive proper medical care because of the unavailability of specialized equipment and lack of skilled personnel. [1,3] Countries with advanced technologies face other challenges, for example, late diagnosis of a disease like OSCC (Oral Squamous Cell Carcinoma) which leads to head and neck cancer. [2] A microfluidic, miniaturized laboratory system (Lab-on-a-chip or LOC) is being developed to address these challenges. LOC is one of many POC (Point-of-care) devices which accurately and rapidly analyze small volumes of samples, without the need for skilled professionals. [1]

Polymerase Chain Reaction (PCR) is an important part for the analysis of a clinical sample [4,5] and needs to take place on the POC device. Lab-on-a-chip devices have been shown to have promise. However, the difficulty of realizing such analysis in the field, with only a self contained, miniaturized chip has not been successfully resolved. [6]

2. BACKGROUND

2.1 PCR and LOC

LOC (Lab-on-a-chip) devices are composed of multiple microchannels and chambers where different stages of DNA analysis take place. [6] The PCR process can take place in one chamber or in multiple chambers. [5] PCR consists of the amplification of DNA by the creation of copies through a series of cycles where multiple reactions occur. Many diagnostic, forensic, agricultural and biologically-related applications utilize PCR technology. One application involves detecting DNA molecules from a diversity of pathogens within a solution [4], an application commonly found in a lab-on-chip device. For a complete analysis of a patient's disease, DNA must be traced and analyzed after it has been amplified through the PCR process; depending on the DNA result, a person can have a clear diagnosis of whether or not they have a disease. A simple LOC system is shown below in Figure 1.

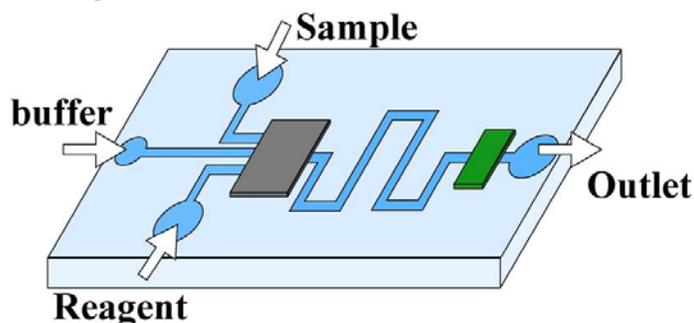


Figure 1: LOC system (Nanospad, 2007).

2.2 Polymerase Chain Reaction (PCR)

2.2.1 History

In 1971, Kleppe and his research group were the first ones to introduce DNA amplification into the science field; yet, this theoretical concept was ignored by most biologists due to inaccessibility of specialized instruments. In 1985, Saiky and colleagues gave the first demonstration of a PCR process. In 1988, the same group described a thermostable polymerase which was the key component of the amplification process and the model used for PCR machines of today. [4]

Chunsun Zhang and colleagues [5] report that around 30 years ago, an attempt to fabricate PCR microfluidics devices was demonstrated by a research group from Stanford University. In 1993, Northrup and colleagues described the first silicon-based stationary PCR chip. Zhang also mentions that in 1998, stationary chamber PCR microfluidics was introduced. Since then, many have shown great interest in the development of these systems to be used in, for example, LOC devices.

2.2.2 PCR: How does it work?

The DNA amplification process consists of three stages:

✓ Denaturation: The reaction mixture is heated to 94-98°C for 20-30 seconds; this causes the hydrogen bonds between the strands of a DNA sample to break, producing single strands of DNA.

✓ Annealing: Temperature is reduced to 50-65°C for 20-40 seconds; the primers (single strands of synthetically synthesized DNA [7]) form hydrogen bonds for the attachment with their matching sequences meaning they will pair with their corresponded bases: adenine (A) with thymine (T), and cytosine (C) with guanine (G). [8]

✓ Extension: Temperature is raised to 72°C for 30-50 seconds; here, the primers unite with the single-stranded DNA and thus, new DNA is created; the temperature may change depending on the primer used.

The process described above comprises a cycle and is illustrated below in Figure 2. Such cycles can be repeated 25-30 times to obtain millions of copies of the DNA under study. For a successful amplification cycle, the sample must be maintained accurately at the desired temperature. Also, if the time between temperature changes is small, the possibilities of having any undesired product, such as primer-dimers, are reduced. Primer-dimers are present when one primer combines with another primer instead of to the single-stranded DNA.

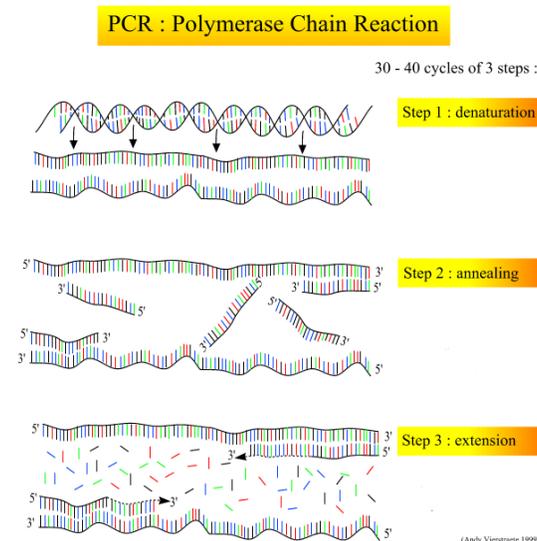


Figure 2: The PCR steps (Vierstrate, 1999).

Amplified DNA needs to be measured so it can be analyzed. PCR dyes are some of the tools used for the identification of DNA. Two principle types of dyes exist:

✓ DNA fluorescent-binding:

The dye shows low fluorescence in the unbound state. But when it binds to a double stranded (ds) DNA molecule, fluorescence increases, thus indicating the presence of the molecule. The signal from the dye is proportional to the quantity of nucleic acid. SYBR Green and Ethidium bromide are common dyes used for this purpose. In research, this kind of dye is used for direct analysis and quantification of DNA. [5,7,11,12]

✓ Labeled nucleic acid:

The dye contains fluorophore-coupled elements which are materials composed of fluorescent dye molecules, or the dye itself [12]. Either of these elements interacts with the DNA, allowing quantification of the nucleic acid. This method uses the phenomenon of Fluorescent Resonance Energy Transfer (FRET) where at least two fluorophores, called the donor and acceptor, must be near each other on a DNA strand. The donor fluorophore is excited by an external light source and transmits light to the acceptor fluorophore, exciting the acceptor. When both molecules are at the same level of excitation, the emitted energy (light) of the donor travels around its space at different distances and orientations.[12] Changes in the fluorescent intensity of a DNA molecule resulting from such variations are what permits the tracking of the genetic material throughout the amplification. This phenomenon permits the dye to be very specific, targeting a portion of the dsDNA. A common dye used for this technique is TaqMAN. [11]

2.2.3 Difference between RT-PCR and conventional PCR

A PCR machine must have a thermal engine. The engine's function, as its name indicates, is to produce heat which is utilized during the amplification process. This process may be analyzed during its development depending on the PCR platform used. There are two types of platforms:

✓ Conventional PCR machines:

The PCR process proceeds in a tube inside the machine. After the cycling process is complete, the amplified DNA is analyzed by agarose gel electrophoresis. Agarose is a purified, gelatinous substance obtained from algae. [7] Gel electrophoresis is a technique used for the separation of DNA in a gel matrix depending on the size of the DNA molecule. The negatively charged DNA molecule moves through the gel via an electric field produced by an external power source. [7] The most common dye used for DNA tracking is Ethidium Bromide (EtBr). The dye reacts with the DNA in the gel and fluoresces under ultraviolet light, creating different band sizes that represent different DNA sizes as shown in Figure 3. Yet, EtBr is a known carcinogen, so safer dyes such as SYBR Green can be used.

The time required for the PCR process (2-3 hours) plus the extra time required for gel analysis makes this approach impractical for field applications. More importantly, this analysis is not efficacious since the initial status of the nucleic acid or the number of copies obtained is unknown. [9]

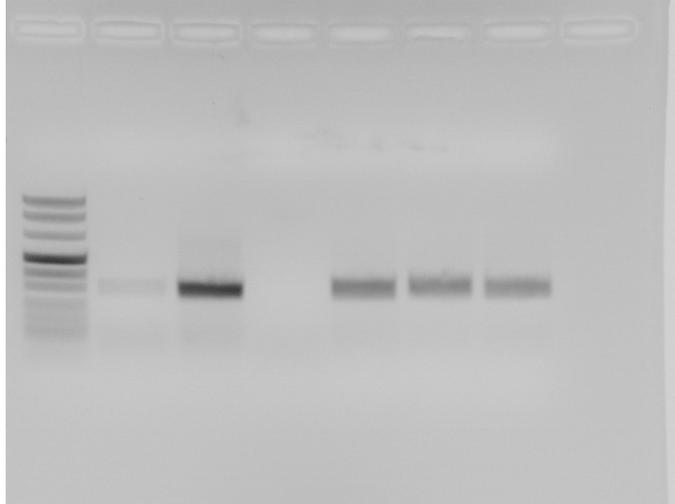


Figure 3: Electrophoresis gel from membrane test (more details in Sec.:4.0)

✓ RT-PCR (Real Time-Polymerase Chain Reaction):

The PCR progress proceeds inside the RT-PCR machine, and DNA amplification is monitored by a fluorometer. The machine emits light which interacts with the DNA copies. The light emitted by the copies is then detected by the optic component of the machine and analyzed. This way, the instrument measures the amount of fluorescence emitted by a dye bound to DNA molecules during the amplification. [10]

The use of gel electrophoresis is not needed in RT-PCR, thereby reducing the time the analysis takes. Also, the real time dye measurement gives the platform knowledge of the status of the initial DNA during the PCR process. The quantity of copies produced, and the detection of any mutation or pathogen present during the process are some of the results the machine provides. Researchers have developed miniaturized versions of such technology. [5] A disadvantage of this platform is that the light received by the optic instrument from the sample can be limited, thus reducing the sensitivity of detection.



Figure 4: Conventional PCR (left) and RT-PCR machines (right)

3. PCR CHIP: DESIGN, FABRICATION AND TESTING

To simulate a chamber from an LOC device, a chip with one chamber was created. The chamber was designed using SOLIDWORKS software (as seen in Figure 5) and fabricated with a UNIVERSAL LASER laser cutter. The chip was composed of three layers which were bonded with a MMA/methanol solution or double-sided tape. The simple 25~50 μl (various chips were used in the experiments) chamber was located in the middle layer where it contained two microchannels for liquid flow. The plastic used for the fabrication of the chip was acrylic. Another option would be the use of polycarbonate, but light transmission through acrylic is higher than the former which makes it a better option for microscope analysis and for RT-PCR analysis [13, 14].

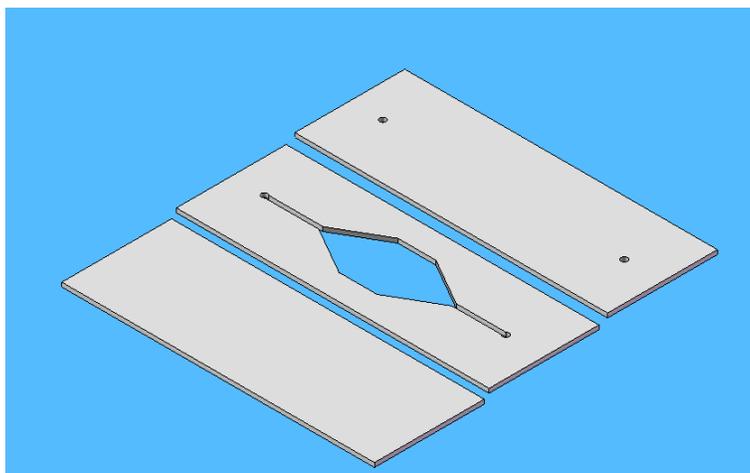


Figure 5: Top, middle, and bottom of PCR acrylic chip assembly

The PCR chip must resist high temperatures during the amplification process inside the PCR machine and must not lose any liquid to be efficient.

For this, some chips were loaded with *Water, Deionized Ultra Filtered (D.I.U.F.)* and then the inlet and outlet holes were sealed with aluminum-foil tape. Others were sealed with a piece of acrylic plastic (6 by 4 mm), which was bonded using MMA/methanol solution. The chips were placed on a hot plate for 10 minutes with a constant temperature of 95-100 $^{\circ}\text{C}$. After the 10 minutes passed, the results were that neither the foil-tape chip nor the solvent-bonded (MMA/methanol solution) chip showed leakage. Yet, after several tests, it was decided that the better solution to prevent leaks was to use the aluminum foil tape.

The next step was to test the PCR chips in the RT-PCR machine. An acrylic chip containing a thermocouple was placed inside the machine to measure the chip's temperature (see Figure 6), and compare it to the temperature of the PCR's cavity as indicated by the RT-PCR program. After several runs where the lid and plate temperatures were also measured, it was found that the chip was not at the desired temperature. Nevertheless, when the chip was tested in a conventional PCR machine (*Techné TC-912*), there was no kind of damage to the PCR chip or liquid loss. During the experiments, a piece of rubber was placed on top of the chip, and later the lid was closed tightly. The rubber would make the chips have a good contact with the machine's block, thus creating a good heat transfer between the block and the chip. The pressure made by the

rubber did not affect the amplification process occurring inside the chip (this will be discussed in Sec.: 4.0)



Figure 6: Thermometer (R); thermocouple (L). With the thermocouple attached to a PCR chip, the temperature inside the RT-PCR machine was measured.

In other tests, the PCR chips were loaded with water, sealed and placed inside the RT-PCR machine. The RT-PCR program was set to perform a melting curve analysis in which the sample, after the PCR process is completed, is heated and the change in fluorescence is detected from the resulting single-stranded DNA [15]. In this case, there should be no signal since there was not any DNA present inside the chips. Yet, the signal obtained was not clear (this will be discussed in Sec. 5.0).

It was concluded that for further studies of the DNA amplification process in the chip to be done, a normal PCR machine must be used. After the process is completed, a melting curve analysis, and microscope and electrophoresis gel analysis can be performed to confirm that the PCR process was successful.

4. PCR CHIP AND RT-PCR

4.1 Membranes

In a PCR process, the membrane's function is to recover DNA molecules that have been liberated after, for example, a lysis process has taken place. However, it is not well known if the membrane has any chemicals that could interfere with data obtained by the RT-PCR machine and thus, affect the resulting melting curve.

In the next tests, several white PCR vials were loaded with different membranes to see if a signal could be obtained from inside the vials (white vials must be used for better results).

If clear vials are used, the light coming from the optical instrument would trespass them and no information could be gained. Different membranes were used such as Flinders Technology Associates (FTA) membrane [16], alumina membrane and Porex® membrane. Also, the use of a shredded FTA membrane was examined.

The FTA membrane is a cellulose membrane that contains different kinds of lyophilized chemicals and proteins for the lysis process of microorganisms and the capturing of their DNA. [16] Alumina has strong atomic bonding which makes it a widely used material in engineering ceramics. [17] However, when this material is combined with a salt solution and microorganism,

a lysis process occurs. [18] During the reaction, the interaction between the DNA molecules and the alumina surface gets stronger, ending in the attachment of the DNA molecules to the alumina. The Porex membrane is a porous, polymeric membrane which works as a filter that separates molecules depending on their size and the membrane's overall porosity. Two possibilities for molecules to pass through the porous structure is by temperature gradient or concentration. [19] Some polymers used to fabricate the membrane are polytetrafluoroethylene, polyurethane and polyacrylate. [20]

First, FTA and alumina membranes were tested. Some tubes could contain one or two FTA membranes or alumina membranes; others could contain FTA membranes not washed or washed once or twice with *Whatman FTA purification reagent* buffer and TE (10 mM Tris, 1mM E, Ph=7.5) buffer. Each vial contained 4 μl of *template DNA 10 ng/ μl* and *lambda primer mixture-A 20 pmol/ μl* ; 25 μl of *SYBR Green Super Mix* and *Cellgro, molecular grade water* for a total volume of 50 μl . These were placed in the RT-PCR machine and with them, two positive controls (vials with no membranes). A PCR program for lambda DNA (found in the Appendix) was used and after 35 cycles, a melting curve was obtained as shown in Figure 7. Later, the results were confirmed with an electrophoresis gel analysis and fluorescent microscope analysis.

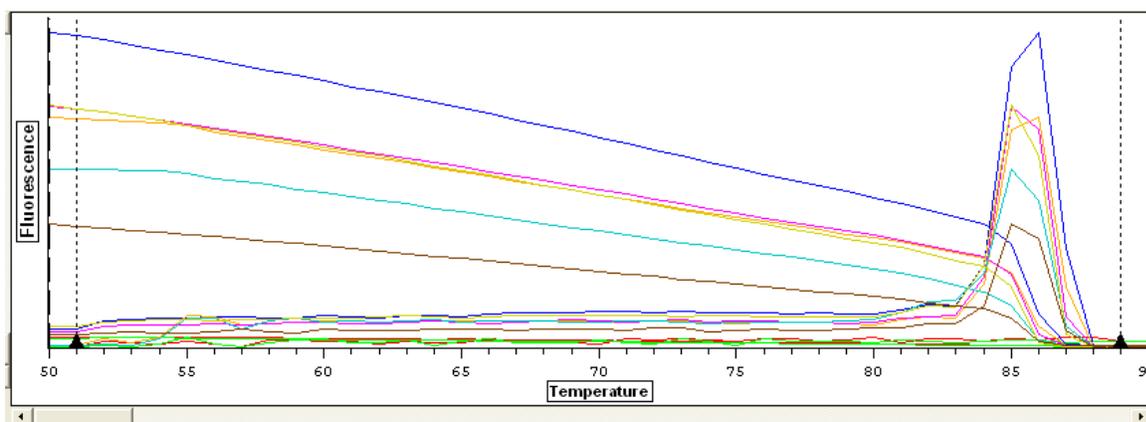


Figure 7: Melting curve from membranes within tubes test.

In chips, to verify that amplification had occurred, a melting curve analysis was performed. PCR chips were loaded with 60 μl of the following solution: 90 μl of *SYBR Green Super Mix*; 10 μl *template DNA 10 ng/ μl* ; 3 μl of *lambda primer mixture-A 20 pmol/ μl* + 77 μl of water. Each chip was placed in a normal PCR machine and to ensure good heat transfer, a piece of rubber was placed on top of them. A lambda DNA program was run (35 cycles) and, then a microscope and electrophoresis gel study was completed.

After several tests, normal FTA membranes as well as shredded FTA membranes were placed inside the chips.

✓ *If cells were used:*

Unwashed membranes were placed inside the chambers and, then diluted cells with a concentration of 10^6 cells/ml filled the chip. After twenty minutes, the remaining cells were removed and 80 μl of PCR solution (160 μl of *SYBR Green Super Mix*; 4.0 μl of *B-cereus cells primers*; 2.0 μl of *TAQ-DNA Polymerase-Native*; 150 μl of water) was added to each *50 μl chamber. The chips were placed in the conventional PCR machine and then, a lambda DNA

program was run. Later on, a melting curve and an electrophoresis gel and microscope analysis were completed.

✓ *If lambda template DNA was used:*

Membranes were washed with FTA buffer and TE buffer and, then placed inside the chips. Each chamber was loaded with 25 μ l of DNA mix (3.5 μ l of *lambda primer mixture-A 20 pmol*; 10 μ l *lambda template DNA* + 75 μ l of water) and 25 μ l of *SYBR Green Super Mix*. After they were placed in the normal PCR, a lambda DNA program was run and afterwards, microscope and electrophoresis gel analyses were made. A fluorescent microscope image of the FTA membrane inside the chip after experiencing the PCR process is shown in Figure 8.

*Different chip sizes were used.

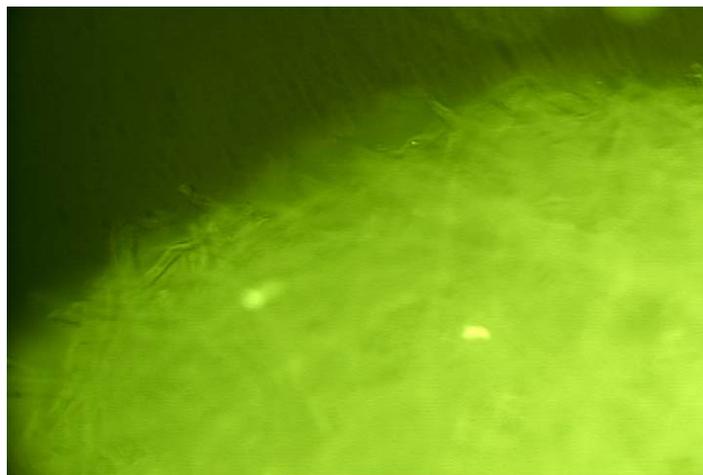


Figure 8: Fluorescing FTA membrane inside a PCR chip.

4.2 Solidification process

Next, membranes were tested in a paraffin environment with the idea of completing lysis, washing, and PCR in a single chamber. This simplified process would make the design more efficient since no extra chamber would be needed. Before studying this idea, one test was done to ensure that paraffin did not interfere with the amplification process. Three PCR vials containing solidified PCR reagents (*GE Healthcare PuReTaq™Ready-To-Go™PCR bead*; *SYBR Green Super Mix*; *lambda primer mixture-A 20 pmol/ μ l*) were loaded with different components. One vial was a positive control (solidified PCR reagents only); one vial had paraffin, and last vial had reagents plus one FTA membrane. A melting curve was obtained and gel electrophoresis was run.

After this test, various PCR vials were loaded and left to dry overnight with 20 μ l of the following solution: 8 *GE Healthcare PuReTaq™Ready-To-Go™PCR bead*; 108 μ l of *SYBR Green Super Mix*; 4.0 μ l of *B-Cereus cell primers* if *B-Cereus* cells are used and 3.5 μ l of *lambda primer mixture-A 20 pmol/ μ l* if *template DNA 10 ng/ μ l* is used; 3.0 μ l *Invitrogen TAQ-DNA Polymerase, Native* (this reagent is optional). In the morning, 1/3 of a *Ampliwax® PCR Gem 50* was loaded to each tube and heated to 75 °C for 50 seconds.

✓ If FTA or shredded FTA membranes and cells were used:

Unwashed membranes and diluted cells with a concentration of 10^8 cells/ml were added. After twenty minutes, the remaining cells were discarded; the membranes were washed once or twice with 25 μ l of FTA buffer and TE buffer; finally, 40 μ l of water were added to each vial. PCR *B-cereus* cells program was run on the RT-PCR machine and, later an electrophoresis gel was performed (program found in the Appendix section).

✓ If alumina or the porous membranes and cells were used:

Unwashed membranes and 50 μ l of lysis buffer with cells (200 μ l of *Buffer AL Lysis buffer 12 ml* + 20 μ l of *QIAGEN Proteinase K 10 ml*; 590 μ l of water; 40 μ l of diluted cells with a concentration of 10^8 cells/ml) were added to each vial. After twenty minutes, the remaining solution was removed; the membranes were washed once or twice with 200mM NaCl and water (sometimes, the alumina membrane was washed with TE buffer). Finally, 40 μ l of PCR reagents (100 μ l of *SYBR Green Super Mix*; 3.5 μ l of *B-cereus cells primer*; 1.0 μ l of *TAQ-DNA Polymerase-Native*; 102 μ l of water) were added. A PCR *B-cereus* cells program was run on the RT-PCR machine and, later an electrophoresis gel was performed.

✓ If lambda template DNA was used:

Any membrane used was washed first; then it was loaded into the vial with a solution of *template DNA 10 ng/ μ l* and water; a lambda DNA program was run on the RT-PCR machine and later, an electrophoresis gel was performed.

The dry storage process on a PCR chip was also tested. A number of PCR chips were loaded and left to dry overnight with 6 to 8 μ l of PCR solution (4 *GE Healthcare PuReTaq™Ready-To-Go™PCR bead*; 4.8 μ l of *B-cereus cells primers* or 2.0 μ l of *lambda primer mixture-A 20 pmol/ μ l*; *Biomatrix™ tubes*; 20 μ l of *SYBR Green Super Mix*). In the morning, small amounts of wax were added on top of the PCR reagents and heated up to 75° C (the PCR reagents must be completely covered). Unwashed FTA and shredded FTA membranes with diluted cells (conc. 10^8 cells/ml) were added to the chambers. Twenty minutes later, the remaining cells were removed and left to dry for one to two hours. The membranes were washed once with FTA buffer and TE buffer and, half an hour later, they were loaded with water. After placing a piece of rubber on top of the chips and securing the lid, a PCR *B-cereus* cells program was run on the normal PCR and later, a melting curve and an electrophoresis gel and microscope analysis were performed. A chip following the PCR process is shown in Figure 9.

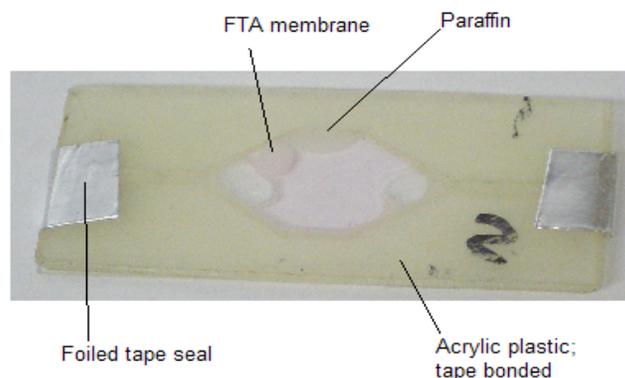


Figure 9: Chip after PCR process

To melt the wax, the heat from the RT-PCR and PCR machines was used. For the vials, the RT-PCR was programmed to maintain a constant temperature of 75° C. For the chips, the same program was used with the only difference being the application of a metal plate into the RT-PCR block. The chips were placed on top of the plate until the paraffin melted. The same process could be done on a normal PCR machine.

5. DISCUSSION AND CONCLUSIONS

Results showed that the foil tape seal protection on the inlet and outlet holes was as good as the acrylic solvent bonded seal. However, the foil tape is easier to handle and faster to apply to the chip.

The material can resist the temperature changes of the PCR process, thus no loss of sample is seen. Also, the acrylic plastic shows no signs of stress after the PCR process unless too much pressure is applied, in which case the material expands.

One possibility for the difference in temperatures readings between the thermocouple and the RT-PCR software could be the finite distance that exists between the lid of the machine and the chip. For a successful PCR process, there must be a good heat transfer between the chip and the plates of the machine.

Some reasons for the unclear melting curve obtained from the chip on the RT-PCR machine (as showed in figure 10) could be the depth of the PCR chip, the unsuitable background were the chip is placed (black background) and again, the distance between the chip and the lid of the machine. If a good signal is to be acquired, there must be a minimum amount of time were the light coming from the optical instrument can interact with the sample's molecules so the sufficient information can be received. If the depth is reduced, the volume is reduced, thus the time must increase if any valuable data is hoped to be collected. Also, a suitable background is needed because of the machine's sensitive optical instrument.

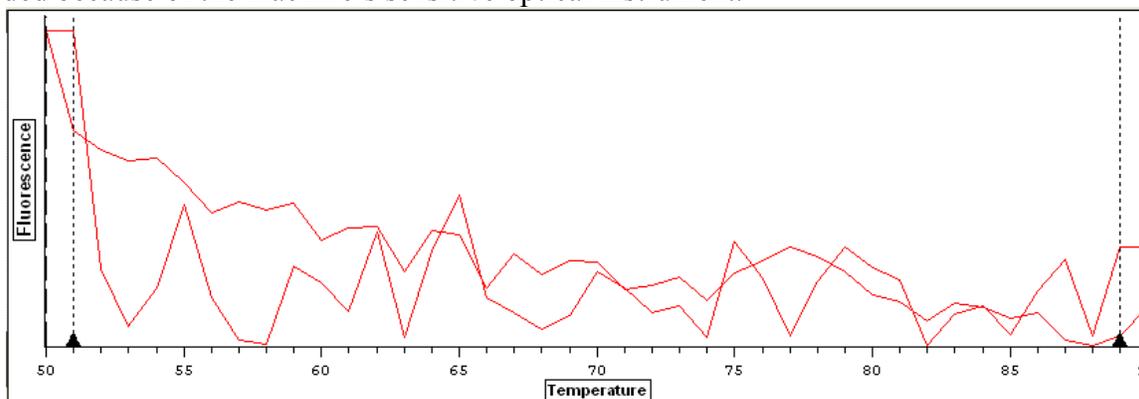


Figure 10: Unclear melting curve obtained from a chip.

Paraffin is a promising encapsulation method that can be used for the PCR process and as part of the LOC system. Studies have shown that paraffin can conserve the PCR reagent during the washing of any membrane used in the experiments. The lysis and purification process of cells is not affected by the paraffin. Results from melting curve analysis also show that paraffin does not affect the resulting DNA copies. Some melting curve results obtained from chips demonstrate

that DNA amplification can take place inside the chips. Amplification can be confirmed with microscope and electrophoresis results. In addition, in past experiments from colleagues, the solidification of PCR reagents and the use of wax in the chip have been shown to work. However, the results from experiments made in summer 2008 are not consistent.

According to our studies, FTA membranes work best on tubes and on chips in the solidification process. Excellent results have been obtained from melting curve analysis and from microscope and electrophoresis gel analyses. From melting curve results, fluorescence coming from a sample containing an FTA membrane can be as much as 50% stronger than a signal coming from a sample containing an alumina membrane.

However, samples containing FTA membrane are 25% to 30% or even less strong than samples containing the porous membrane or shredded FTA membranes. These results bring high hopes for the porous membrane to be used on the chips and thus, the LOC system. The use of the shredded FTA membrane will be a good choice too. The studies demonstrate that shredding does not cause the FTA membrane to lose its properties. Being shredded gives it the advantage of covering more surface area, thus “capturing” more DNA.

Nevertheless, the alumina membrane worked best in tubes where no wax was added. Melting curve results show that signals coming from samples containing alumina membranes could be as strong or stronger than signals coming from samples containing FTA membranes. These results show that the alumina membrane can work in conditions where no other materials, like wax, can interfere with it. Because alumina is a brittle material it is possible that during the PCR process, in tubes where wax was used, the alumina cracked and interfered with the RT-PCR optical analysis.

In chips where no paraffin was placed, results show that the FTA membrane is not affected by the pressure the chip experiences in the PCR machine. Melting curve results show strong bands from the chip’s sample. This can be confirmed with the strong fluorescence seen under the microscope and the strong, clear bands obtained from the electrophoresis gel.

Yet, our studies show that the FTA membrane can fluoresce on its own. This detail must be kept in mind when a melting curve or other type of analysis is made based on the RT-PCR results. Also, another detail is that washing the membranes once or twice does not significantly impact the results.

6. RECOMMENDATIONS

DNA amplification inside a chip inside an RT-PCR machine is challenging. The tape bonded method is not recommended for RT-PCR analysis because of the machine’s sensitive optical instruments. The tape’s fluorescence is stronger than the dye normally used, thus the information gathered by the machine would not be correct. Another kind of bonding method must be used. In addition, since acrylic can auto fluoresce, a good signal might not be obtained from the RT-PCR. Another matter that must be resolved is the heat transfer problem. This fact brings another problem which is the depth of the chip as compared to the depth of the PCR vials. The RT-PCR machine was designed to realize vial analysis and not chip analysis.

The solidification process using paraffin in a chip must be further investigated if good results are to be obtained from the RT-PCR device. The Porex membrane and the shredded FTA membrane must also be further investigated.

7. ACKNOWLEDGEMENTS

I would like to thank my advisor, Dr. Haim Bau of the University of Pennsylvania, for the opportunity to work on this project. I would also like to thank Dr. Michael Mauk, Jason Thompson and all the other researchers in Towne 216 for their support and advice during the summer. I would also like to thank Dr. Jan Van der Spiegel and those who sponsor the SUNFEST program for giving students like me the opportunity to do research in their area of interest.

8. REFERENCES

1. Paul Yager, Thayne Edwards, Elain Fu, Kristen Helton, Kjell Nelson, Milton R. Tam, Bernhard H. Weigl, "Microfluidic diagnostic technologies for global public health". *Nature*. pp. 412-418, (2006, July). Available: doi:10.1038/nature05064
2. Barry L. Ziober, Michael G. Mauk, Erica M. Falls, Zongyuan Chen, Amy F. Ziober, Haim H. Bau. "Lab-on-a-chip for oral cancer screening and diagnosis". *Head and Neck-Journal*, vol.30, pp. 111-121
3. Curtis D. Chin, Vincent Linder, Samuel K Sia. "Lab-on-a-chip for global health: Past studies and future opportunities". vol.7, 2007, pp 41-57
4. N.A. Saunders, "An introduction to Real-Time PCR", www.horizonpress.com/pcr/
5. Chunsun Zhang, Jinliang Xu, Wenli Ma, Wenling Zheng. "PCR microfluidic devices for DNA amplification". *Biotechnology Advances*, vol:24, May/June, 2006, pp 243-284
6. Andrea E. Guber, Mathias Hecke, Dirk Herrmann, Alban Muslija, Volker Saile, Lutz Eichorn, Thomas Gietzelt, Werner Hoffmann, Peter C. Hauser, Jatisai Tanyanyiwa, Adreas Gerlach, Norbertr Gottshlich, Günther Knebel. August 2004 "Microfluidic lab-on-a-chip systems based on polymers—fabrication and application". *Chemical Engineer Journal*. vol:101. pp. 447-453
7. Unknown author, "Agarose Gel Electrophoresis of DNA", <http://www.vivo.colostate.edu/hbooks/genetics/biotech/gels/agardna.html>
8. Unknown author, "The Polymerase Chain Reaction", <http://www.sumanasinc.com/webcontent/animations/content/pcr.html>
9. Unknown author, "Real-Time PCR Vs. Traditional PCR", Applied Biosystems, pp 15.
10. J. M. J. Logan and K. J. Edwards, "An overview of PCR platform", www.horizonpress.com/pcr/
11. Unknown author, "Real-Time PCR Application Guide", Bio-Rad Inc., CA, pp 3-25
12. M. A. Lee, D. J. Squirrell, D. L. Leslie, T. Brown Homogeneous, "Fluorescent Chemistries for Real-Time PCR", www.horizonpress.com/pcr/
13. Unknown author, "Acrylic PMMA (Polymethyl-Methacrylate) Specifications", http://www.boedeker.com/acryl_p.htm
14. Unknown author, "Light Transmission for Elvex Safety Lenses" <http://www.elvex.com/facts10.htm>

15. Unknown author, "Melting Curve Technology", <https://www.roche-applied-science.com/servlet/RCConfigureUser?URL=StoreFramesetView&storeId=10202&catalogId=10202&langId=-1&countryId=us>
16. Hugo Moscoso, Sthephan G. Thayer, Charles L. Hofacre, Stanley H Kleven. "Inactivation, storage, and PCR detection of Mycoplasma on FTA® filter paper". American Association of Avian Pathologists, Kennett Square, PA., 2004, vol. 48, n°4, pp. 841-850
17. Unknown author, "Aluminum Oxide, Al₂O₃", <http://accuratus.com/alumox.html>
18. Joon-ho Kim, Yoon-kyoung Cho, Geun-bae Lim, Soo-hwan Jeong, "Method for isolating nucleic acid using alumina", U.S. Patent 20050136463, june 25, 2005
19. Unknown author, "Polymeric membranes". <http://www.solvaymembranes.com/infocus/whataremembranes/polymericmembranes/0,,473-2-0,00.htm>
20. Brzezinki S., Malinowska G., Nowak T., Schmidt H., Marcinkowska D., Kaleta A. "Structure and properties of microporous polywrthane membranes designed for textile-polymeric composite systems". (2005, January-December). *FIBRES & TEXTILES IN EASTERN EUROPE*. vol.13, issue: 6, pp. 53-58.
Available:http://apps.isiknowledge.com/full_record.do?product=UA&search_mode=GeneralSearch&qid=1&SID=2CmFjNj8khP8LI6C3fN&page=1&doc=1&colname=WOS

9. APPENDIX

A. PCR B-cereus program specifications in RT-PCR:

Cycles: 35

-Lid temperature: Constant, 99° C

-Starting temperature: 95° C for 3:00 min.

-Cycle:

95° C for 30 sec.,

55° C for 30 sec.,

68° C for 40 sec.

-Plate read after each cycle

-Melting Curve: 50° C to 90° C; when the temperature raises 1° C, there is a plate read which takes 10 sec.

B. PCR lambda DNA program specifications in RT-PCR:

Cycles: 35

-Lid temperature: Constant, 99° C

-Starting temperature: 94° C for 1:00 min.

-Cycle:

94° C for 30 sec.,

60° C for 30 sec.,

72° C for 1:00 sec.

-Plate read after each cycle

-Melting Curve: 50° C to 90° C; when the temperature raises 1° C, there is a plate read which takes 10 sec.

C. PCR B-cereus program specification in a normal PCR:

Cycles: 30

-Lid temperature: off

-Starting temperature: 94.5° C for 3:00 min.

-Cycle:

94.5° C for 15 sec.,

54° C for 15 sec.,

68° C for 30 sec.

Holding temperature: 10° C forever

-Melting Curve: 50° C to 90° C; when the temperature raises 1° C, there is a plate read which takes 10 sec.

**The melting curve for the chip's sample is the same as that used in the other programs.

OPTIMIZING LEGGED LOCOMOTION USING TUNABLE LEG STIFFNESS

NSF Summer Undergraduate Fellowship in Sensor Technologies
Kamruzzaman Rony (Electrical Engineering) – Stony Brook University
Advisors: Dr. Dan Koditschek and Kevin Galloway

ABSTRACT

Running efficiently and successfully over unstructured terrain is an important and necessary characteristic for modern legged robots. Research suggests that a necessary step toward achieving this goal is to design legged robots with variable leg stiffness capabilities which can be controlled by the robot autonomously. The specific problem addressed in this research is to develop a wireless (infrared) tunable leg stiffness module that will change leg stiffness when commanded by the robot. In this research, a legged robot with six C-shape legs called RHex is used. The development of the tunable leg stiffness module had two phases. The first phase focused on the design of an infrared communication module that would allow each leg to communicate with the robot body. The second phase involved the integration of a motor and rotary sensor to accurately control stiffness of the leg based commands from robot. In this paper, we discuss the development of this novel tunable robot leg and highlight its advantages and limitations for optimizing RHex-like legged locomotion platforms.

1. INTRODUCTION

Animals or biological systems are capable of moving or running in dynamic fashion over realistic terrain (that varies in geometry, with rises and dips, and in dynamic properties, such as ground stiffness or damping) by varying the stiffness of their limbs in real times to adapt to the changes in the environment [1]. According to the researchers, adjustable leg stiffness is necessary to close the performance gap between robots and animals [1]. Therefore, designing a robot that can autonomously traverse a variety of terrain types requires dynamic stiffness control in its legs.

RHex, a six-legged robot, is one of the most successful autonomous running robots to date. It is the first autonomous dynamic legged locomotion system to passively exchange spring energy through natural body dynamics [2]. It is also the fastest autonomous legged robot capable of operating on rough terrain [3]. Its leg design resembles that of a cockroach and can be explained using the SLIP (Spring Loaded Inverted Pendulum) model. This model treats an animal as a point mass on a linear spring and has been demonstrated to accurately model the center of mass motion of running animals, as well as the ground reaction forces associated with their gaits.

The legs of RHex have a C-shaped profile and have been constructed from materials ranging from carbon fiber to nitinol. These legs have a particular stiffness and consequently a particular natural frequency of vibration. Tuning the gait parameters to match the natural frequency of this spring-mass system allows RHex to achieve very dynamic and efficient gaits; however, changes in speed, payload, and terrain can adversely affect the natural frequency of this spring-mass system. Therefore, tunable leg stiffness is required for RHex so that it can tune its leg stiffness to run effectively and efficiently. Figure 1 shows a RHex with tunable-stiffness legs.

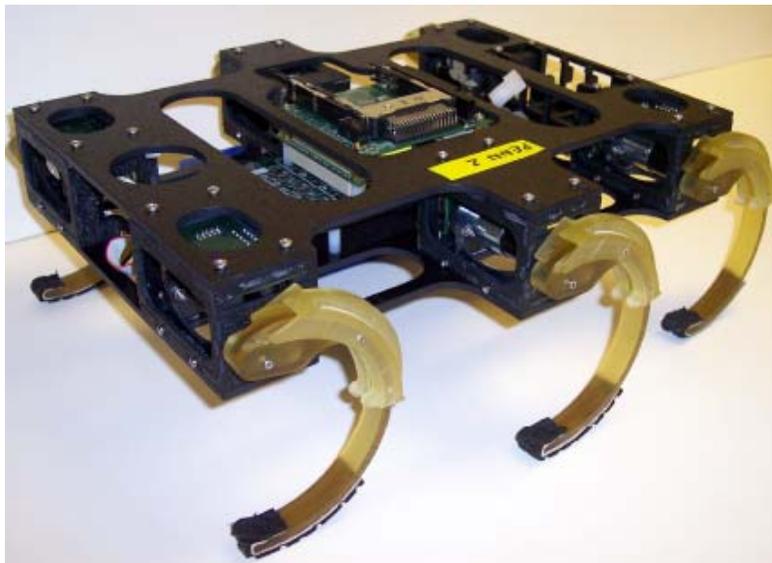


Figure 1: Tunable stiffness legs adapted to RHex

The specific problem addressed in this research is to develop a wireless tunable leg stiffness module that will change leg stiffness when commanded by the robot. Developing a leg stiffness module consists of two parts. One part is to design a communication module so that the body and the leg can communicate with each other. To address the issue of communication between the robot and the leg, we

have selected an infrared device (IrDA transceiver) that comes with a transmitter and receiver. Both the leg and the body have IrDA transceivers, IrDA encoders/decoders and microcontrollers. The IrDA device is inexpensive, robust and is used widely in industry. Wired connection was not a good option for this problem because the legs rotate continuously and slip-rings are too expensive and unsuitable for dirty environments. The second part is to control a motor that changes a slider's position based on the data provided by the body so that the leg can achieve a particular stiffness (Figure 2).

There are different types of stiffness control. The method we are using for our research is known as the structural stiffness control. To realize this method, a slider (approximately one third of the length of the leg) is placed on each leg of the RHex. The stiffness of the leg can be changed by moving a motor-controlled slider over the leg. Whenever the change in leg stiffness is required (for example, when RHex traverses a different terrain) the body will send a signal to the leg commanding the leg to move the slider to a certain position to achieve the optimized leg stiffness for locomotion.

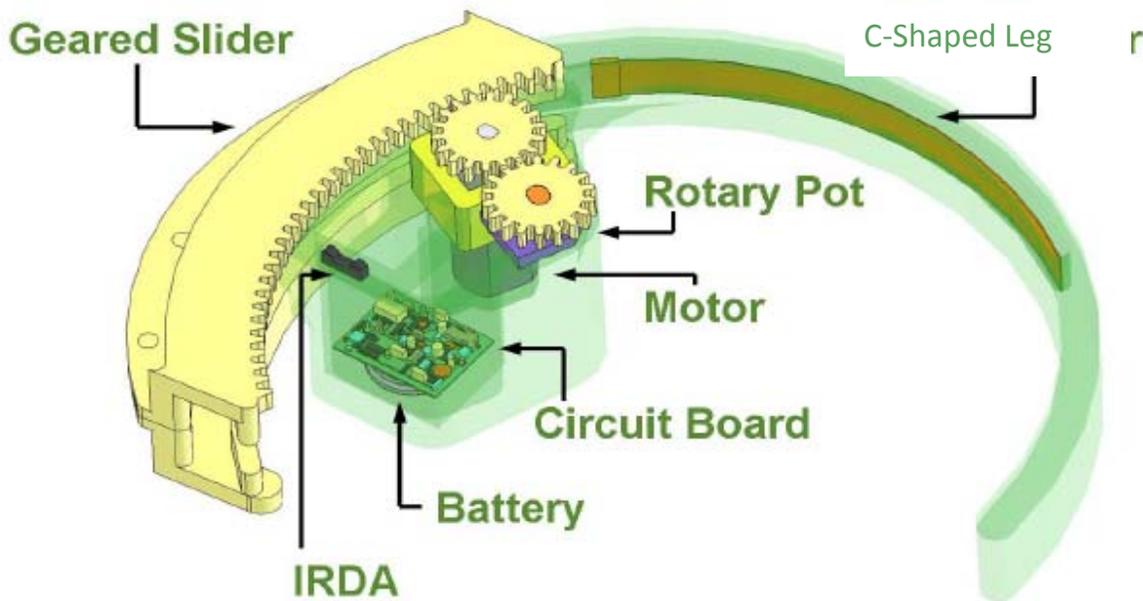


Figure 2: A single C-shape leg with embedded structural stiffness control

Figure 2 shows a single C-shaped leg that has structural stiffness control embedded in it. When the leg receives a signal from the body through IR transceiver (IrDA), then the circuit encodes the signal and rotates the motor to move the geared slider to a certain location to achieve a particular natural frequency. The potentiometer (rotary pot) is used to track the position of the geared slider on the leg basically to minimize the error of sliding.

2. BACKGROUND

The dynamic legged locomotion in animals is very complex; therefore, it is very difficult to implement it in robots. One practical and feasible way for implementing dynamic legged locomotion in robots is to use a simple mathematical model that almost exactly resembles the legged locomotion of animals. The Spring- Loaded Inverted Pendulum (SLIP) model for animal running is one such model that has successfully approximated the sagittal plane dynamics and ground reaction forces of animals ranging from cockroaches to kangaroos [4]. This model has been applied to RHex's forward motion.

Spring- Loaded Inverted Pendulum (SLIP) model:

The SLIP model is a reasonable approximation describing the center of mass (CoM) motion of an animal in a running gait, regardless of the number of legs, the size of the animal, or the running gait employed [4, 5, 6, 7]. This model treats an animal or robot as a point mass on a single mass-less linear spring. Figure 3 shows a forward motion in the SLIP model.

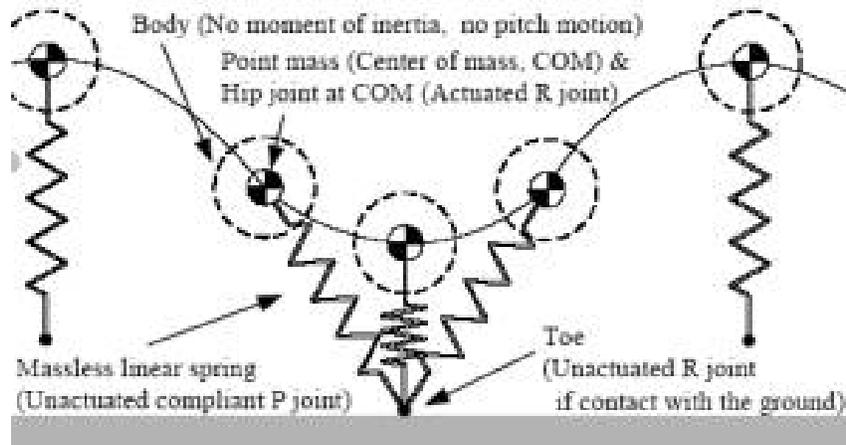


Figure 3: Forward motion using SLIP model of animal running

In this figure, the forward motion of the center of mass of an animal is modeled by the spring-mass system in two independent stages: the flight stage and the stance stage. During the flight stage, the motion is dictated by the effect of gravity on the mass. The stance phase occurs from the moment the spring-leg comes in contact with the ground at some angle, through its compression, rotation and decompression, and until it leaves the ground at its full length. During the stance phase, the opposing forces responsible for the spring's compression and eventual decompression are gravity and the spring force. The effect of gravity is directly dependent on the mass, while the spring force is directly proportional to the compression of the spring. The specific trajectory followed by the SLIP model is dependent on the angle of incidence of the spring with the vertical upon touchdown and the vector of the landing velocity of the center of mass [4].

It may seem that the model in figure 3 is only applicable to a monopod. However, this model accurately describes bipedal, quadruped, and hexapod dynamics with the introduction of Sutherland's virtual leg [8]. Virtual leg describes the idea that legs whose motion is synchronized can be modeled as a

single leg. In a hexapod, the three legs comprising a single tripod are synchronized, resulting in two virtual legs. Similarly, quadrupeds can be treated as bipeds. Since the two virtual legs are never both in stance, we are justified in treating them as we would those of a bounding monopod [9]. Given that RHex is largely biologically inspired, the SLIP model of motion is a valid candidate to capture the dynamics of its center of mass, with the stiffness of the three legs of a tripod determining the spring constant of the single linear spring that represents the virtual leg in the model.

3. Description:

The main objective of the tunable leg stiffness module is to help the RHex to change its leg stiffness autonomously so that when it goes to a different terrain or needs to change its speed or running gaits, it can achieve the most suitable stiffness to perform its locomotion optimally. Therefore, the role of our autonomous stiffness control module is to transmit the data (which is received from the main control system and basically is the level of stiffness that is best suited for the new terrain or new speed) to the leg of RHex as infrared signals, and changing the level of stiffness of the leg depending on the data received from the main control of the robot. Therefore, it is obvious from the hardware point of view that our whole module is divided into two sub-modules. One sub-module remains in the body of the RHex, and the other remains in the leg. The following figure shows the set up of our tunable leg stiffness control module:

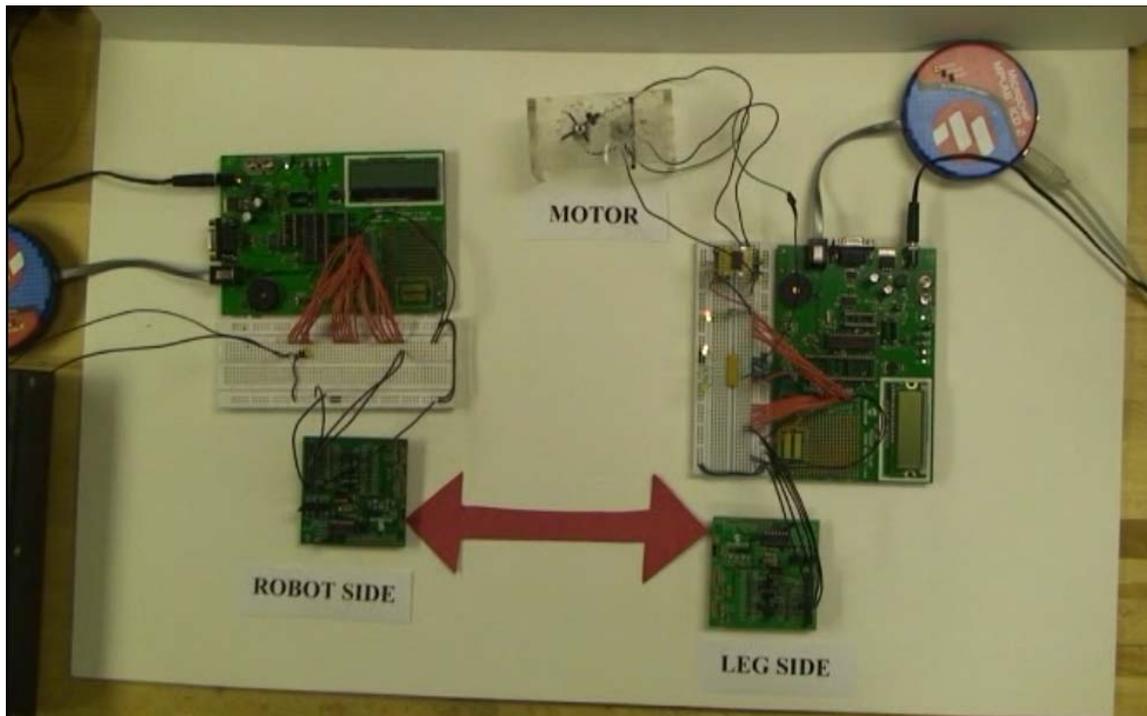


Figure 4: Picture of the whole circuit setup

In the above figure, the left part is the body sub-module and the right part is the leg sub-module. The small green boards at the bottom are the infrared communication units. The transparent device, on the bottom right corner with two gears on top, is actually the testing equipment that

imitates the function of a leg with slider. The motor and the potentiometer are attached with this device and connected to our main circuitry.

The tunable leg stiffness module that we have developed consists of both the hardware and the software. The functions and details of both the hardware and the software parts and their components are briefly described in the following sections.

3.1 Hardware:

The following figure shows a basic block diagram of the tunable leg stiffness hardware module:

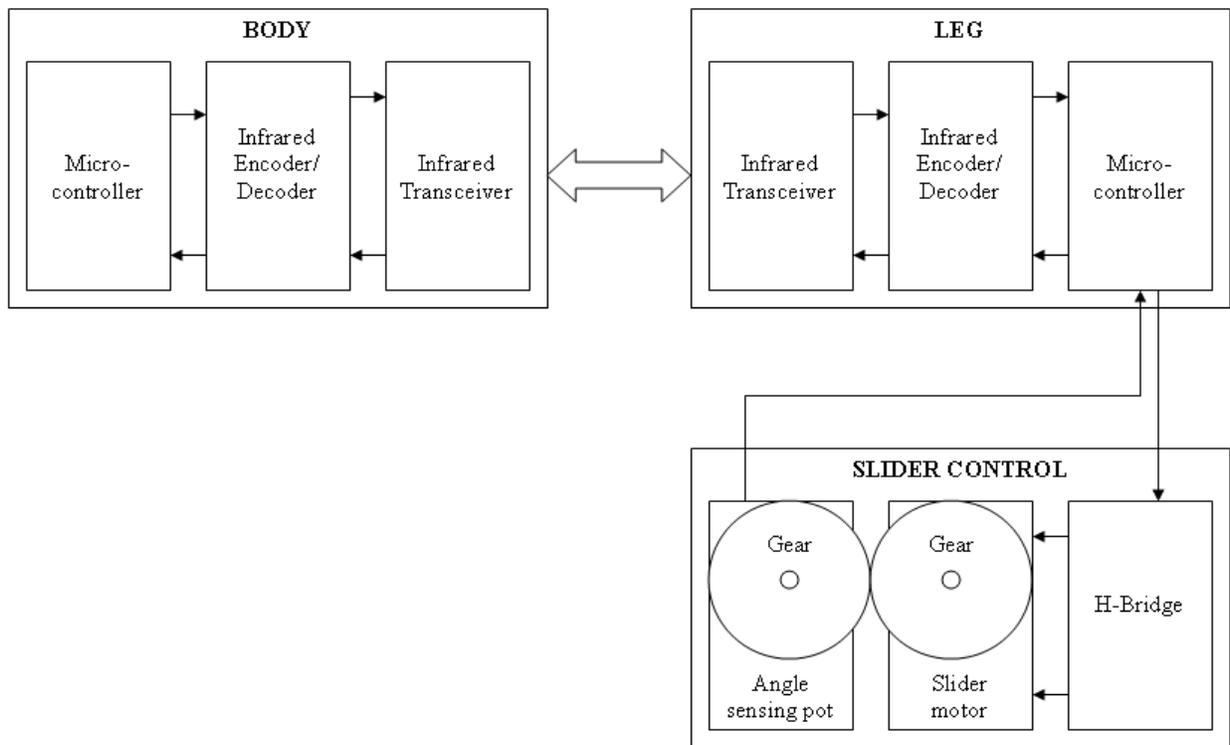


Figure 5: Block diagram of the hardware of the tunable leg stiffness module.

In this figure, we see that both the body and the leg sub-modules have their own microcontroller, IrDA encoder/decoder and infrared transceiver. The last two components help the microcontrollers to communicate in between themselves using infrared signal. The leg sub-modules have some additional components. They are the H-bridge, the motor and the potentiometer. These components are required to place the slider to a certain position on the leg to achieve particular leg stiffness.

The functions of the different components of our module are described here. The microcontroller in the body part gets the data from the main microcontroller of the RHex through wired connection. Then the body-microcontroller sends the data to the infrared encoder/decoder, which after encoding sends the encoded data to the infrared transceiver. The Infrared transceiver transmits infrared signals which are received by the infrared transceiver in the leg. The infrared encoder/decoder connected to the infrared transceiver in the leg part decodes the infrared signal and send the data to the microcontroller connected to it. Depending on the received data, the microcontroller controls the H-bridge which actually controls the motor's rotation and the direction of rotation. The motor gear is in contact with another gear which is tied to an angle sensing potentiometer. Therefore, when the motor rotates, the voltage at the variable node of the potentiometer also changes. By measuring the voltage of the potentiometer, we can tell the position of the motor or the slider on the leg. Since the voltage of the potentiometer is analog, it indicates infinite number of position. However, for our application we need to deal with definite number of positions. For this reason we have used an analog-to-digital converter (ADC). The microcontroller we are using has a 10 bit ADC as its peripheral and we have used that. Beside the 10 bit ADC, we have also used other two peripherals of the microcontroller. They are the universal synchronous asynchronous receiver transmitter (USART) and the pulse width modulation (PWM) peripherals. We have used USART for infrared communication and PWM to control the driving power (or the speed of rotation) of the motor.

The microcontrollers that we have used for both sub-modules are the same and they are PIC18F2680. The model numbers for infrared encoder/decoders, transceivers, H-bridge and potentiometer are respectively MCP2120, TFDU-4300, HIP4020 and SV01L. Brief descriptions of these components along with their mentionable features are described in the following sections.

3.1.1 PIC18F2680 Microcontroller Unit (MCU):

PIC18F2680 is a 28 pin enhanced flash microcontrollers with ECAN™ technology, 10-Bit A/D converter and nano-watt Technology. Its maximum DC operating frequency is 40 MHz. Its main features are as follows:

- 65536 bytes of program memory.
- 3328 bytes of data memory.
- 1024 bytes of data EEPROM memory.
- Total 19 interrupt sources
- Three 8-bits I/O ports – port A, B and C.
- Four timers.
- One capture/compare/PWM module.
- 10 different oscillator modes with one 8 MHz internal oscillator.
- MSSP and Enhanced USART for serial communication.
- 10 bit A/D converter with 8 input channels.
- Programmable High/Low voltage detects option.
- 75 instructions with general instruction set enabled and 83 instructions with extended instruction set enabled.

These are the main features that are used for the development of our module. However, this MCU offers many other important features that can be utilized to develop a high-end embedded system. The pin configuration of this MCU is shown in Figure 2.

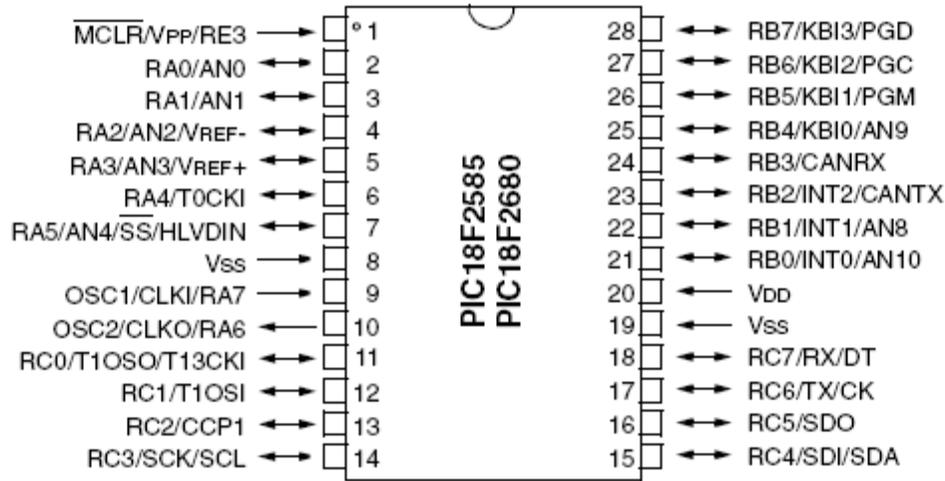


Figure 6: Pin diagram of PIC18F2680 PDIP

In our design, we are using internal 8 MHz oscillator for both the body and the leg sub modules. For the MCU in the body sub-module the pins that we have used are pin 2, 3, 4, 5, 8, 17, 18, 19 and 20. Among them, pin 20 is connected to VDD (+5V), pin 8 and 19 are connected to GND. Pin 2, 3, 4, 5 are connected to switch and are the input to the module. Among these four pins, pin 2, 3, 4 determines the position of the slider on the leg and pin 5 determines the direction of rotation of the pin. Since we are using 3 pins to determine the position of the slider, so the slider can be placed to $2^3 = 8$ different position, which is good enough for our present research. If more positions are needed then we can use more pins to indicate the position. Pin 17 (TX) and 18 (RX) are used for the serial data transmission using USART to implement infrared communication scheme. These two pins are connected respectively to pin 12 (TX) and 11 (RX) of the infrared encoder/decoder of the body sub-module.

On the other hand, for the MCU in the leg sub-module the pins that we have used are pin 2, 3, 4, 5, 8, 13, 17, 18, 19 and 20. Among these pins, pin 20 is connected to VDD (+5V), pin 8 and 19 is connected to GND. Pin 17 (TX) and 18 (RX) are used to implement the infrared communication scheme and they are respectively connected to pin 12 (TX) and 11 (RX) of the infrared encoder/decoder of the leg sub-module. Pin 13 (CCP1) and pin 3 are respectively connected to pin 10 and pin 9 of the H-bridge. Pin 13 provides the pulse width modulated signal to the H-bridge and eventually to the motor. Pin 3 controls the direction of rotation of the motors. The data at pin 3 is the same data that is entered at pin 5 of the body module. Pin 4 (Vref-) and 5 (Vref+) are connected to the GND and VDD pins of the angle sensing potentiometer respectively. Pin 2 is configured as an analog input for the 10 bit A/D converter and is connected to the variable voltage pin of the potentiometer.

The following table describes the position of the slider depending on the inputs (pin 2, 3, 4 and 5) at the MCU of body sub-module.

Table 1: Position of the slider depending on the input at the body sub-module

INPUT			POSITION (Angle – only for testing)	
PIN 2	PIN 3	PIN 4	PIN 5 = 0 (Clockwise)	PIN 5 = 1 (Anticlockwise)
0	0	0	1 (0°)	1 (0°)
0	0	1	2 (45°)	2 (45°)
0	1	0	3 (90°)	3 (90°)
0	1	1	4 (135°)	4 (135°)
1	0	0	5 (180°)	5 (180°)
1	0	1	6 (225°)	6 (225°)
1	1	0	7 (270°)	7 (270°)
1	1	1	8 (315°)	8 (315°)

The following sections provide brief overviews and configurations of different peripherals of the MCU that are used in our design. These peripherals are the USART module, PWM module and 10 bit A/D Converter module.

3.1.1.1 USART module:

The Universal Synchronous Asynchronous Receiver Transmitter (USART) module is one of the two serial I/O modules of the PIC18F2680 MCU. USART is also known as a Serial Communications Interface or SCI. In our design, we have used the USART in asynchronous mode to transmit/receive data serially in between the MCU and the infrared encoder/decoder.

The pins of the USART are multiplexed with PORTC. In order to configure RC6/TX/CK and RC7/RX/DT as a USART:

- bit SPEN (RCSTA<7>) must be set (= 1)
- bit TRISC<7> must be set (= 1)
- bit TRISC<6> must be cleared (= 0) for Asynchronous mode.

The operation of the Enhanced USART module is controlled through three registers. They are:

- Transmit Status and Control (TXSTA)
- Receive Status and Control (RCSTA)
- Baud Rate Control (BAUDCON)

The details of these registers are given in appendix A.

3.1.1.2 PWM module:

The PWM module of PIC18F2680 MCU produces up to a 10-bit resolution PWM output signal. This signal is output through pin 13 (RC2/CCP1). Since the CCP1 pin is multiplexed with a PORTC data latch, the appropriate TRIS bit must be cleared to make the CCP1 pin an output.

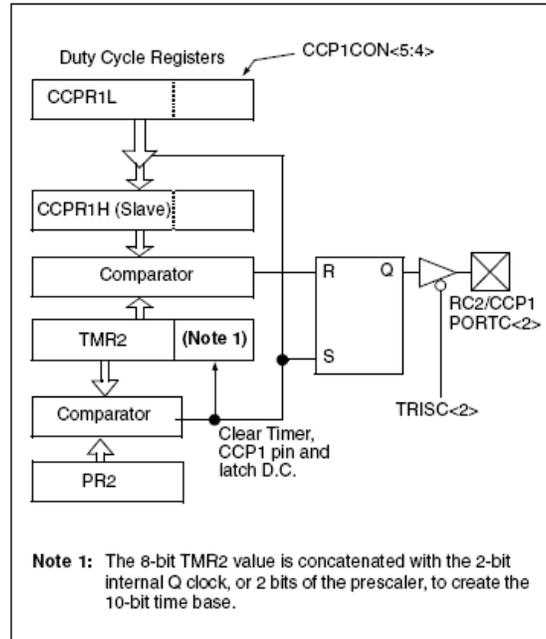


Figure 7: Simplified PWM block diagram

Figure 7 shows a simplified block diagram of the CCP1 module in PWM mode. A step-by-step procedure on how to set up the CCP1 module for PWM operation is given below:

1. Set the PWM period by writing to the PR2 register.
2. Set the PWM duty cycle by writing to the CCPR1L register and CCP1CON<5:4> bits.
3. Make the CCP1 pin an output by clearing the appropriate TRIS bit.
4. Set the TMR2 prescale value, then enable Timer2 by writing to T2CON.
5. Configure the CCP1 module for PWM operation.

To calculate the value for the PWM period and duty cycle we have used the following two formulas respectively.

$$(1) \text{ PWM Period} = [(PR2) + 1] \cdot 4 \cdot \text{TOSC} \cdot (\text{TMR2 Prescale Value})$$

$$(2) \text{ PWM Duty Cycle} = (\text{ECCPR1L:ECCP1CON<5:4>}) \cdot \text{TOSC} \cdot (\text{TMR2 Prescale Value})$$

The registers that we have used for configuring the PWM are CCPR1L, CCP1CON<5:4>, TRISC and T2CON. Details of these registers are provided in Appendix A.

3.1.1.3 10 bit A/D Converter module:

The Analog-to-Digital (A/D) converter module has 8 inputs and it allows the conversion of an analog input signal to a corresponding 10-bit digital number. This module has five registers:

- A/D Result High Register (ADRESH)
- A/D Result Low Register (ADRESL)
- A/D Control Register 0 (ADCON0)
- A/D Control Register 1 (ADCON1)
- A/D Control Register 2 (ADCON2)

The ADCON0 register controls the operation of the A/D module. The ADCON1 register configures the functions of the port pins. The ADCON2 register configures the A/D clock source, programmed acquisition time and justification. Details of these registers are provided in Appendix A. The following steps should be followed to perform an A/D conversion:

1. Configure the A/D module:
 - Configure analog pins, voltage reference and digital I/O (ADCON1)
 - Select A/D input channel (ADCON0)
 - Select A/D acquisition time (ADCON2)
 - Select A/D conversion clock (ADCON2)
 - Turn on A/D module (ADCON0)
2. Wait the required acquisition time (if required).
3. Start conversion:
 - Set GO/DONE bit (ADCON0 register)
4. Wait for A/D conversion to complete, by either:
 - Polling for the GO/DONE bit to be cleared or
 - Waiting for the A/D interrupt
5. Read A/D Result registers (ADRESH:ADRESL);
6. For next conversion, go to step 1 or step 2, as required.

3.1.2 MCP2120 Infrared Encoder/Decoder:

This device is a high-performance and fully-static infrared encoder/decoder. This device sits between a UART and an infrared (IR) optical transceiver. Its baud rate is user selectable to standard IrDA baud rates between 9600 baud to 115.2 kbaud. The maximum baud rate is 312.5 kbaud. The pin diagram and description of this infrared encoder/ decoder is given below:

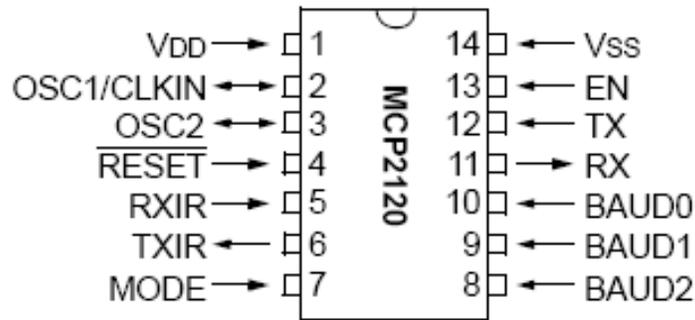


Figure 8: Pin diagram of the infrared encoder/decoder

Table 2: Pin description of the infrared encoder/decoder

Pin Name	Pin Number		Pin Type	Buffer Type	Description
	PDIP	SOIC			
VDD	1	1	—	P	Positive supply for logic and I/O pins
OSC1/CLKIN	2	2	I	CMOS	Oscillator crystal input/external clock source input
OSC2	3	3	O	—	Oscillator crystal Output
RESET	4	4	I	ST	Resets the Device
RXIR	5	5	I	ST	Asynchronous receive from infrared transceiver
TXIR	6	6	O	—	Asynchronous transmit to infrared transceiver
MODE	7	7	I	TTL	Selects the device mode (Data/Command) for Software Baud Rate operation. For more information see Section 2.4.1.2 “Software Selection” .
BAUD2	8	8	I	TTL	BAUD2:BAUD0 specify the Baud rate of the device, or if the device operates in Software Baud Rate mode. For more information see Section 2.4.1 “Baud Rate” .
BAUD1	9	9	I	TTL	
BAUD0	10	10	I	TTL	
RX	11	11	O	—	Asynchronous transmit to controller UART
TX	12	12	I	TTL	Asynchronous receive from controller UART
EN	13	13	I	—	Device Enable.
Vss	14	14	—	P	Ground reference for logic and I/O pins

Legend: TTL = TTL compatible input ST = Schmitt Trigger input with CMOS levels
I = Input O = Output
P = Power CMOS = CMOS compatible input

The data received from a standard UART is encoded (modulated), and output as electrical pulses to the IR Transceiver. The IR Transceiver also receives data which it outputs as electrical pulses. The MCP2120 decodes (demodulates) these electrical pulses and then the data is transmitted by the MCP2120 UART. This modulation and demodulation method is performed in accordance with the IrDA standard.

Table 3: Hardware baud rate selection vs. frequency

BAUD2:BAUD0	Frequency (MHz)							Bit Rate
	0.6144 ⁽¹⁾	2.000	3.6864	4.9152	7.3728	14.7456 ⁽²⁾	20.000 ⁽²⁾	
000	800	2604	4800	6400	9600	19200	26042	Fosc / 768
001	1600	5208	9600	12800	19200	38400	52083	Fosc / 384
010	3200	10417	19200	25600	38400	78600	104167	Fosc / 192
011	4800	15625	28800	38400	57600	115200	156250	Fosc / 128
100	9600	31250	57600	78600	115200	230400	312500	Fosc / 64

The baud rate for the MCP2120 can be configured either by the state of three hardware pins (BAUD2, BAUD1, and BAUD0) or through software selection. For our design, we have used hardware pins. The table above shows hardware pins configurations to select baud rates for different frequencies. For our design we have used 9600 baud rate and 8 MHz frequency. Therefore, the hardware pin configuration was BAUD2:BAUD0 = 000.

3.1.3 TFDU-4300 Infrared Transceiver:

The TFDU4300 is a low profile (2.5 mm) infrared transceiver module which is compliant to the latest IrDA® physical layer standard for fast infrared data communication, supporting IrDA speeds up to 115.2 kbit/s (SIR).

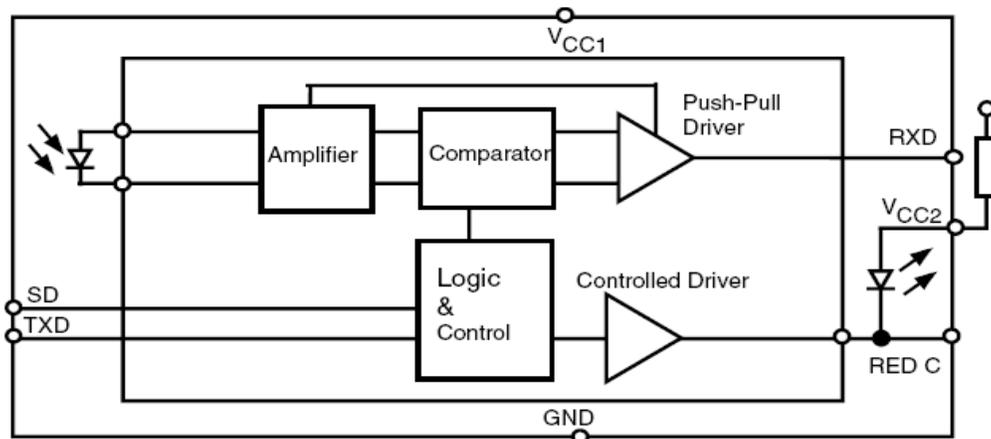


Figure 9: Functional block diagram of the infrared transceiver

Figure 9 shows the functional block diagram of the infrared transceiver which consists of a PIN photodiode, an infrared emitter (IRED), and a low-power control IC.

3.1.4 HIP4020 Half Amp Full Bridge Power Driver (H-bridge):

HIP4020 is an H-bridge that controls the direction of driving current to a motor so that a motor can rotate in both clockwise and anticlockwise directions.

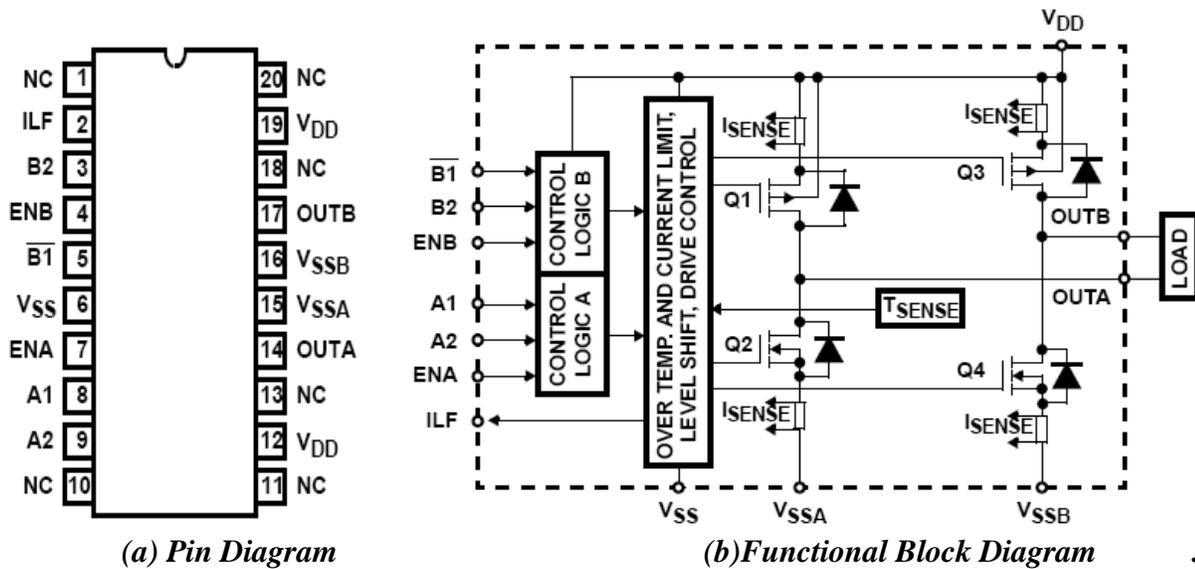


Figure 10: (a) Pin diagram and (b) Block Diagram of HIP4020

In the Functional Block Diagram of the HIP4020, the four switches and a load are arranged in an H-Configuration so that the drive voltage from terminals OUTA and OUTB can be cross-switched to change the direction of current flow in the load. This is commonly known as 4-quadrant load control. As shown in the Block Diagram, switches Q1 and Q4 are conducting or in an ON state when current flows from VDD through Q1 to the load, and then through Q4 to terminal VSSB; where load terminal OUTA is at a positive potential with respect to OUTB.

Switches Q1, Q4 are operated synchronously by the control logic. The control logic switches Q3 and Q2 to an open or OFF state when Q1 and Q4 are switched ON. To reverse the current flow in the load, the switch states are reversed where Q1 and Q4 are OFF while Q2 and Q3 are ON. Consequently, current then flows from VDD through Q3, through the load, and through Q2 to terminal VSSA, and load terminal OUTB is then at a positive potential with respect to OUTA.

3.1.5 SV01L Rotary Position Sensors (Angle sensing potentiometer):

In our design, we have used a rotary position sensor or angle sensing potentiometer (10 K resistance) to track the position of the slider on the leg. It has three pins: 1, 2 and 3. When any one of pin1 and 3 is connected to the VDD of the system, and the other is connected to the GND, then with the rotation of the gear, the voltage at pin2 varies. We connected pin1 to the VDD of our system and pin3 to the GND. Pin2 was connected directly to the analog input of the MCU which was pin 2. This variable voltage acted as the input to the A/D converter.

3.2 Software:

Both the body and the leg sub-modules have their own programs. The main task of the program of the body sub-module is to take the input from the user, encode the input data and send it to the leg sub-module serially as infrared signals. On the other hand, the task of the leg sub-module is to receive the infrared signals, decode the signals to get the input data and to control the slider motor depending on the input to change the leg stiffness to a certain level. The programs for both the body (transmitter) and the leg (receiver) sub-modules are given below:

The source code in C language for the body or transmitter part:

```
#include <p18f2680.h>
#include <stdio.h>
#include <usart.h>
#pragma config WDT = OFF

void main (void)
{

    ADCON1 = 0b00111111;
    TRISA = 0b11111111;

    OpenUSART( (USART_TX_INT_OFF & USART_RX_INT_OFF & USART_ASYNC_MODE &
    USART_EIGHT_BIT & USART_CONT_RX & USART_BRGH_LOW) , 12 );

    while (1)
    {
        WriteUSART (PORTA);
    }
}
```

The source code in C language for the leg or receiver part:

```
#include <p18f2680.h>
#include <stdio.h>
#include <usart.h>
#include <pwm.h>
#include <adc.h>
#pragma config WDT = OFF

void main (void)
{

//Declare variables
```

```

char new_sw, prev_sw;
int d_theta, m_theta;

//Configure or initialize I/O ports and peripherals

ADCON1 = 0b00001111;
TRISA = 0b00000000;

//Configure or initialize the USART
OpenUSART( (USART_TX_INT_OFF & USART_RX_INT_OFF & USART_ASYNC_MODE &
USART_EIGHT_BIT & USART_CONT_RX & USART_BRGH_LOW), 12 );

//Configure or initialize the 10 bit A/D converter
OpenADC( ADC_FOSC_32 & ADC_RIGHT_JUST & ADC_12_TAD, ADC_CH0 & ADC_INT_OFF,
14 );

//Configure the PWM module
OpenPWM(0x63);
SetDCPWM1(160);

//Infinite loop that controls the slider motor
while (1)
{
do
{
new_sw = ReadUSART();
} while (prev_sw == new_sw);

prev_sw = new_sw;
m_theta = 0b0000111111111111;
new_sw = new_sw & 0b00000111;

if (new_sw == 0b00000000 || new_sw == 0b00000100)
d_theta = 0b 0000 0000 0000 0000;
else if (new_sw == 0b00000001 || new_sw == 0b00000101)
d_theta = 0b 0000 0000 1111 1111;
else if (new_sw == 0b00000010 || new_sw == 0b00000110)
d_theta = 0b 0000 0001 1111 1111;
else
d_theta = 0b 0000 0010 1111 1111;
while (m_theta != d_theta)
{
ConvertADC();
}
}

```

```

while (BusyADC());
m_theta = ReadADC();
}
}
}

```

In the transmitter part, the first line enables digital I/O for all the pins of port A. The second line makes all the pins of port A to digital input pins. The third line calls a subroutine called 'OpenUSART()' which is defined in 'usart.h' header file. The main function of that sub-routine is to configure the usart for serial data transfer. Then we have used a while loop whose condition is always 'true' (1) that means it runs infinitely until we turn off the system. Inside the while loop we have used another subroutine called 'WriteUSART ()' which is also defined in the 'usart.h' header file. This function writes the 8-bit data that is passed as an argument to the 'TXREG' register of the USART and transmits the data as infrared signals.

On the other hand, in the transmitter part, at first the variables are declared, then the I/O ports and the peripherals (USART, PWM, A/D converter) are configured. Then the infinite while loop starts that at first receive and decode the data from the infrared signals, and then depending on the data it controls the slider motor.

For our software development and debugging we have used MPLAB IDE v8.10 and MPLAB C18 v3.20 compiler. Some pictures of the designed variable leg stiffness module during testing is given below:

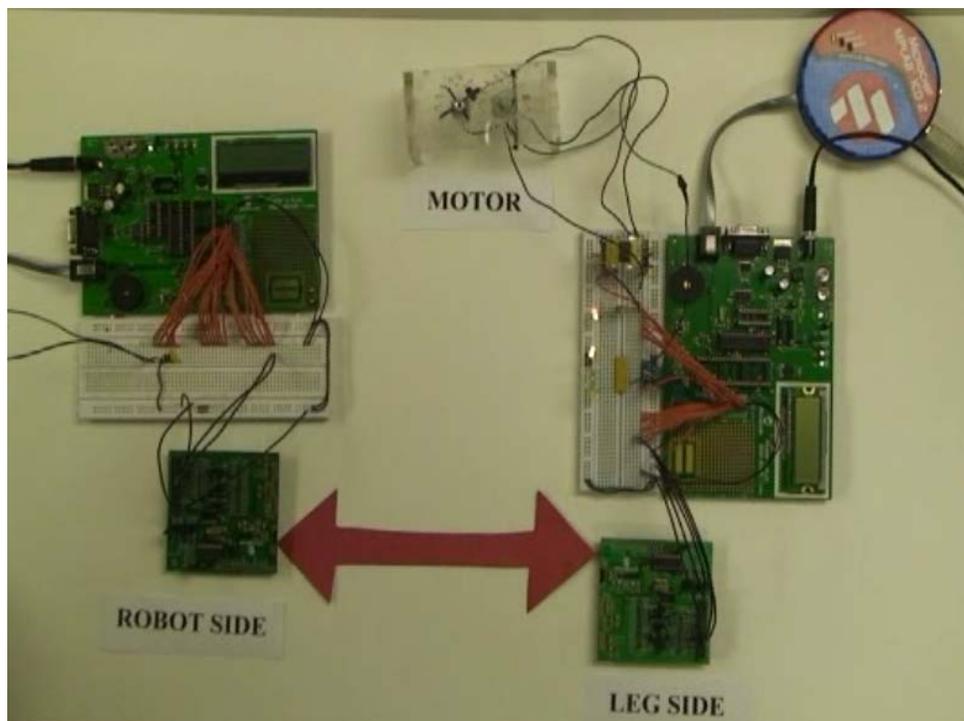


Figure 11: Variable leg stiffness module test setup

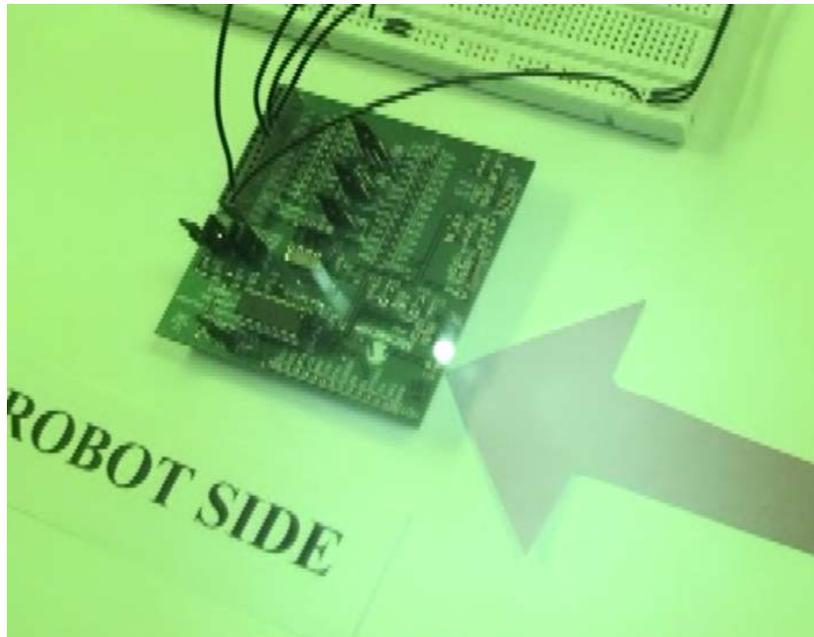


Figure 12: The infrared transmitter is transmitting data as infrared signal from the robot/body side to the leg side of the module (the video was taken in night vision mode to show the infrared signal)

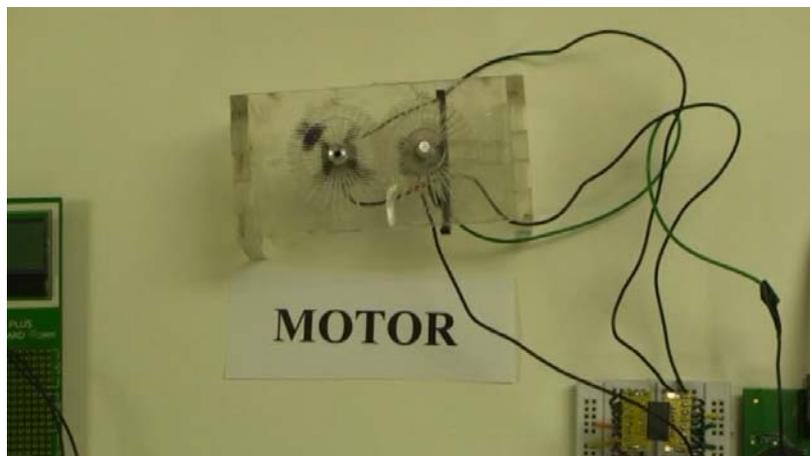


Figure 13: The slider motor (an alternative of the variable stiffness leg used only for testing) is rotating while it was being controlled by the robot/body sub-module

4. Discussion/Conclusion:

Using tunable leg stiffness control, where a robot can adjust its leg stiffness when it is required, is highly important for legged robots in order to minimize the performance gap in legged locomotion between animals and robots. Considering this fact, we have developed an electronic module

for the variable compliance legs of RHex, which allows the robot to change its legs stiffness automatically when it is commanded by the body or the main controller of the robot. This module with variable compliance leg improves the dynamic legged locomotion of RHex and allows it to move effectively and successfully in different terrains with different speeds and payloads. This research not only adds a new feature to the current dynamic legged locomotion system of RHex, it also opens the door for future research in this field.

5. Acknowledgements:

First of all, I would like to thank Dr. Koditschek for giving me the opportunity to work in the Kodlab this summer. I would also like to thank Kevin Galloway for his direction and support on the project. Bill Mather deserves a great deal of recognition for entertaining my numerous questions and requests. I am also thankful to Michael Park for his support. I would like to thank Dr. Van der Spiegel and the SUNFEST staff for their encouragement and organizational support. Special thanks go to Brianna Banford for filling my many purchasing orders. I would also like to thank the NSF REU program for their continued support of undergraduate research. Finally, I would like to thank everybody in the Kodlab and SUNFEST program for making this summer remarkable.

6. References:

1. Ferris, D., Louie, M., and Farley, C., "Running in the real world: Adjusting leg stiffness for different surfaces," *Proceedings of the Royal Society London (Series B – Biological Sciences)*, Vol. 265, Jun 1998, p 989–994. [Database: ISI Web of Knowledge]
2. Saranli, U., Buehler M., and Koditschek, D., "RHex: A simple and highly mobile hexapod robot," *International Journal of Robotics Research*, v 20, n 7, Jul 2001, p 616-31. [Database: ISI Web of Knowledge]
3. Galloway, K., Clark, J., and Koditschek, D., "Design of a multi-directional variable stiffness leg for dynamic running," *Proceedings of IMECE2007-41318*, ASME International Mechanical Engineering Congress and Exposition, Nov 2007.
4. Blickhan, R., Full, R.J., "Similarity in multilegged locomotion: Bouncing like a monopode," *Journal of Comparative Physiology*, v A, n 173, Nov 1993, p 509-517. [Database: ISI Web of Knowledge]
5. Dickinson, M., Farley, C., Full, R., Koehl, M., Kram, R., and Lehman, S. "How animals move: An integrative view," *Science*, v 288, issue 5463, Apr 2000, p100-106. [Database: ISI Web of Knowledge]
6. Blickhan, R., "The spring-mass model for running and hopping," *Journal of Biomechanics*, v 22, issue 11-12, 1989, p 1217-1227. [Database: ISI Web of Knowledge]

7. Farley, C.T. and Ferris, D.P., “Biomechanics of walking and running: Center of mass movements to muscle action,” *Exercise and Sport Sciences Reviews*, volume 26, chapter 10, 1998, p 253-285. [Database: ISI Web of Knowledge]
8. Sutherland, I.E., Ullner, M.K., “Footprints in the Asphalt,” *International Journal of Robotics Research*, v 3, n 2, 1984, p 29-36. [Database: ISI Web of Knowledge]
9. Cham, J.G., Bailey S., Clark J.E., et al., “Fast and Robust: Hexapedal Robots via Shape Deposition Manufacturing,” *International Journal of Robotics Research*, v 21, n 10-11, Oct.-Nov. 2002, p 869-82. [Database: ISI Web of Knowledge]

7. APPENDIX

Microcontroller Registers

1. USART Registers

A) TXSTA:

TXSTA: TRANSMIT STATUS AND CONTROL REGISTER

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R-1	R/W-0
CSRC	TX9	TXEN	SYNC	SEnDB	BRGH	TRMT	TX9D
bit 7							bit 0

- bit 7 **CSRC:** Clock Source Select bit
Asynchronous mode:
 Don't care.
Synchronous mode:
 1 = Master mode (clock generated internally from BRG)
 0 = Slave mode (clock from external source)
- bit 6 **TX9:** 9-bit Transmit Enable bit
 1 = Selects 9-bit transmission
 0 = Selects 8-bit transmission
- bit 5 **TXEN:** Transmit Enable bit
 1 = Transmit enabled
 0 = Transmit disabled
Note: SREN/CREN overrides TXEN in Sync mode.
- bit 4 **SYNC:** EUSART Mode Select bit
 1 = Synchronous mode
 0 = Asynchronous mode
- bit 3 **SEND:** Send Break Character bit
Asynchronous mode:
 1 = Send Sync Break on next transmission (cleared by hardware upon completion)
 0 = Sync Break transmission completed
Synchronous mode:
 Don't care.
- bit 2 **BRGH:** High Baud Rate Select bit
Asynchronous mode:
 1 = High speed
 0 = Low speed
Synchronous mode:
 Unused in this mode.
- bit 1 **TRMT:** Transmit Shift Register Status bit
 1 = TSR empty
 0 = TSR full
- bit 0 **TX9D:** 9th bit of Transmit Data
 Can be address/data bit or a parity bit.

B) RCSTA:

RCSTA: RECEIVE STATUS AND CONTROL REGISTER

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R-0	R-0	R-x
SPEN	RX9	SREN	CREN	ADDEN	FERR	OERR	RX9D
bit 7							bit 0

- bit 7 **SPEN:** Serial Port Enable bit
 1 = Serial port enabled (configures RX/DT and TX/CK pins as serial port pins)
 0 = Serial port disabled (held in Reset)
- bit 6 **RX9:** 9-bit Receive Enable bit
 1 = Selects 9-bit reception
 0 = Selects 8-bit reception
- bit 5 **SREN:** Single Receive Enable bit
Asynchronous mode:
 Don't care.
Synchronous mode – Master:
 1 = Enables single receive
 0 = Disables single receive
 This bit is cleared after reception is complete.
Synchronous mode – Slave:
 Don't care.
- bit 4 **CREN:** Continuous Receive Enable bit
Asynchronous mode:
 1 = Enables receiver
 0 = Disables receiver
Synchronous mode:
 1 = Enables continuous receive until enable bit CREN is cleared (CREN overrides SREN)
 0 = Disables continuous receive
- bit 3 **ADDEN:** Address Detect Enable bit
Asynchronous mode 9-bit (RX9 = 1):
 1 = Enables address detection, enables interrupt and loads the receive buffer when RSR<8> is set
 0 = Disables address detection, all bytes are received and ninth bit can be used as parity bit
Asynchronous mode 9-bit (RX9 = 0):
 Don't care.
- bit 2 **FERR:** Framing Error bit
 1 = Framing error (can be updated by reading RCREG register and receiving next valid byte)
 0 = No framing error
- bit 1 **OERR:** Overrun Error bit
 1 = Overrun error (can be cleared by clearing bit CREN)
 0 = No overrun error
- bit 0 **RX9D:** 9th bit of Received Data bit
 This can be address/data bit or a parity bit and must be calculated by user firmware.

C) BAUDCON:

BAUDCON: BAUD RATE CONTROL REGISTER

R/W-0	R-1	U-0	R/W-0	R/W-0	U-0	R/W-0	R/W-0
ABDOVF	RCIDL	—	SCKP	BRG16	—	WUE	ABDEN
bit 7							bit 0

- bit 7 **ABDOVF**: Auto-Baud Acquisition Rollover Status bit
1 = A BRG rollover has occurred during Auto-Baud Rate Detect mode
(must be cleared in software)
0 = No BRG rollover has occurred
- bit 6 **RCIDL**: Receive Operation Idle Status bit
1 = Receive operation is Idle
0 = Receive operation is active
- bit 5 **Unimplemented**: Read as '0'
- bit 4 **SCKP**: Synchronous Clock Polarity Select bit
Asynchronous mode:
Unused in this mode.
Synchronous mode:
1 = Idle state for clock (CK) is a high level
0 = Idle state for clock (CK) is a low level
- bit 3 **BRG16**: 16-bit Baud Rate Register Enable bit
1 = 16-bit Baud Rate Generator – SPBRGH and SPBRG
0 = 8-bit Baud Rate Generator – SPBRG only (Compatible mode), SPBRGH value ignored
- bit 2 **Unimplemented**: Read as '0'
- bit 1 **WUE**: Wake-up Enable bit
Asynchronous mode:
1 = EUSART will continue to sample the RX pin – interrupt generated on falling edge; bit
cleared in hardware on following rising edge
0 = RX pin not monitored or rising edge detected
Synchronous mode:
Unused in this mode.
- bit 0 **ABDEN**: Auto-Baud Detect Enable bit
Asynchronous mode:
1 = Enable baud rate measurement on the next character. Requires reception of a Sync field
(55h); cleared in hardware upon completion
0 = Baud rate measurement disabled or completed
Synchronous mode:
Unused in this mode.

Legend:			
R = Readable bit	W = Writable bit	U = Unimplemented bit, read as '0'	
-n = Value at POR	'1' = Bit is set	'0' = Bit is cleared	x = Bit is unknown

2. PWM module initialization and control registers

A) CCP1CON:

CCP1CON: CAPTURE/COMPARE/PWM CONTROL REGISTER

U-0	U-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
—	—	DC1B1	DC1B0	CCP1M3	CCP1M2	CCP1M1	CCP1M0
bit 7							bit 0

- bit 7-6 **Unimplemented:** Read as '0'
- bit 5-4 **DC1B1:DC1B0:** PWM Duty Cycle bit 1 and bit 0 for CCP1 Module
Capture mode:
 Unused.
Compare mode:
 Unused.
PWM mode:
 These bits are the two LSBs (bit 1 and bit 0) of the 10-bit PWM duty cycle. The eight MSBs (DC19:DC12) of the duty cycle are found in ECCPR1L.
- bit 3-0 **CCP1M3:CCP1M0:** CCP1 Module Mode Select bits
 0000 = Capture/Compare/PWM disabled (resets CCP1 module)
 0001 = Reserved
 0010 = Compare mode, toggle output on match (CCP1IF bit is set)
 0011 = Reserved
 0100 = Capture mode, every falling edge or CAN message received (time-stamp)⁽¹⁾
 0101 = Capture mode, every rising edge or CAN message received (time-stamp)⁽¹⁾
 0110 = Capture mode, every 4th rising edge or every 4th CAN message received (time-stamp)⁽¹⁾
 0111 = Capture mode, every 16th rising edge or every 16th CAN message received (time-stamp)⁽¹⁾
 1000 = Compare mode: initialize CCP1 pin low; on compare match, force CCP1 pin high (CCPIF bit is set)
 1001 = Compare mode: initialize CCP pin high; on compare match, force CCP1 pin low (CCPIF bit is set)
 1010 = Compare mode: generate software interrupt on compare match (CCPIF bit is set, CCP1 pin reflects I/O state)
 1011 = Compare mode: trigger special event, reset timer (TMR1 or TMR3, CCP1IF bit is set)
 11xx = PWM mode
- Note 1:** Selected by CANCAP (CIOCON<4>) bit; overrides the CCP1 input pin source.

B) T2CON:

T2CON: TIMER2 CONTROL REGISTER

U-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
—	T2OUTPS3	T2OUTPS2	T2OUTPS1	T2OUTPS0	TMR2ON	T2CKPS1	T2CKPS0
bit 7							bit 0

bit 7 **Unimplemented:** Read as '0'

bit 6-3 **T2OUTPS3:T2OUTPS0:** Timer2 Output Postscale Select bits
 0000 = 1:1 Postscale
 0001 = 1:2 Postscale
 •
 •
 •
 1111 = 1:16 Postscale

bit 2 **TMR2ON:** Timer2 On bit
 1 = Timer2 is on
 0 = Timer2 is off

bit 1-0 **T2CKPS1:T2CKPS0:** Timer2 Clock Prescale Select bits
 00 = Prescaler is 1
 01 = Prescaler is 4
 1x = Prescaler is 16

Legend:			
R = Readable bit	W = Writable bit	U = Unimplemented bit, read as '0'	
-n = Value at POR	'1' = Bit is set	'0' = Bit is cleared	x = Bit is unknown

3. A/D converter initialization and control registers

A) ADCON0:

ADCON0: A/D CONTROL REGISTER 0

U-0	U-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
—	—	CHS3	CHS2	CHS1	CHS0	GO/ $\overline{\text{DONE}}$	ADON
bit 7							bit 0

Legend:			
R = Readable bit	W = Writable bit	U = Unimplemented bit, read as '0'	
-n = Value at POR	'1' = Bit is set	'0' = Bit is cleared	x = Bit is unknown

- bit 7-6 **Unimplemented:** Read as '0'
- bit 5-2 **CHS3:CHS0:** Analog Channel Select bits
 - 0000 = Channel 0 (AN0)
 - 0001 = Channel 1 (AN1)
 - 0010 = Channel 2 (AN2)
 - 0011 = Channel 3 (AN3)
 - 0100 = Channel 4 (AN4)
 - 0101 = Channel 5 (AN5)^(1,2)
 - 0110 = Channel 6 (AN6)^(1,2)
 - 0111 = Channel 7 (AN7)^(1,2)
 - 1000 = Channel 8 (AN8)
 - 1001 = Channel 9 (AN9)
 - 1010 = Channel 10 (AN10)
 - 1011 = Unused
 - 1100 = Unused
 - 1101 = Unused
 - 1110 = Unused
 - 1111 = Unused

Note 1: These channels are not implemented on PIC18F2X8X devices.

2: Performing a conversion on unimplemented channels will return full-scale measurements.

- bit 1 **GO/DONE:** A/D Conversion Status bit
 When ADON = 1;
 1 = A/D conversion in progress
 0 = A/D Idle
- bit 0 **ADON:** A/D On bit
 1 = A/D converter module is enabled
 0 = A/D converter module is disabled

B) ADCON1:

ADCON1: A/D CONTROL REGISTER 1

U-0	U-0	R/W-0	R/W-0	R/W-0 ⁽¹⁾	R/W-q ⁽¹⁾	R/W-q ⁽¹⁾	R/W-q ⁽¹⁾	
—	—	VCFG1	VCFG0	PCFG3	PCFG2	PCFG1	PCFG0	
bit 7								bit 0

- bit 7-6 **Unimplemented:** Read as '0'
- bit 5 **VCFG1:** Voltage Reference Configuration bit (VREF- source)
 - 1 = VREF- (AN2)
 - 0 = AVSS
- bit 4 **VCFG0:** Voltage Reference Configuration bit (VREF+ source)
 - 1 = VREF+ (AN3)
 - 0 = AVDD

bit 3-0 **PCFG3:PCFG0: A/D Port Configuration Control bits**

PCFG3: PCFG0	AN10	AN9	AN8	AN7 ⁽²⁾	AN6 ⁽²⁾	AN5 ⁽²⁾	AN4	AN3	AN2	AN1	AN0
0000 ⁽¹⁾	A	A	A	A	A	A	A	A	A	A	A
0001	A	A	A	A	A	A	A	A	A	A	A
0010	A	A	A	A	A	A	A	A	A	A	A
0011	A	A	A	A	A	A	A	A	A	A	A
0100	A	A	A	A	A	A	A	A	A	A	A
0101	D	A	A	A	A	A	A	A	A	A	A
0110	D	D	A	A	A	A	A	A	A	A	A
0111 ⁽¹⁾	D	D	D	A	A	A	A	A	A	A	A
1000	D	D	D	D	A	A	A	A	A	A	A
1001	D	D	D	D	D	A	A	A	A	A	A
1010	D	D	D	D	D	D	A	A	A	A	A
1011	D	D	D	D	D	D	D	A	A	A	A
1100	D	D	D	D	D	D	D	D	A	A	A
1101	D	D	D	D	D	D	D	D	D	A	A
1110	D	D	D	D	D	D	D	D	D	D	A
1111	D	D	D	D	D	D	D	D	D	D	D

A = Analog input D = Digital I/O

Note 1: The POR value of the PCFG bits depends on the value of the PBADEN bit in Configuration Register 3H. When PBADEN = 1, PCFG<3:0> = 0000; when PBADEN = 0, PCFG<3:0> = 0111.

2: AN5 through AN7 are available only in PIC18F4X8X devices.

C) ADCON2:

ADCON2: A/D CONTROL REGISTER 2

R/W-0	U-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
ADFM	—	ACQT2	ACQT1	ACQT0	ADCS2	ADCS1	ADCS0
bit 7							bit 0

bit 7 **ADFM:** A/D Result Format Select bit

1 = Right justified

0 = Left justified

bit 6 **Unimplemented:** Read as '0'

bit 5-3 **ACQT2:ACQT0:** A/D Acquisition Time Select bits

111 = 20 TAD
110 = 16 TAD
101 = 12 TAD
100 = 8 TAD
011 = 6 TAD
010 = 4 TAD
001 = 2 TAD
000 = 0 TAD⁽¹⁾

bit 2-0 **ADCS2:ADCS0:** A/D Conversion Clock Select bits

111 = FRC (clock derived from A/D RC oscillator)⁽¹⁾
110 = FOSC/64
101 = FOSC/16
100 = FOSC/4
011 = FRC (clock derived from A/D RC oscillator)⁽¹⁾
010 = FOSC/32
001 = FOSC/8
000 = FOSC/2

Note 1: If the A/D FRC clock source is selected, a delay of one Tcy (instruction cycle) is added before the A/D clock starts. This allows the SLEEP instruction to be executed before starting a conversion.

Legend:

R = Readable bit

W = Writable bit

U = Unimplemented bit, read as '0'

-n = Value at POR

'1' = Bit is set

'0' = Bit is cleared

x = Bit is unknown

All these information have been obtained from the microcontroller data sheet. More details information about these registers can be found in the microcontroller datasheet (<http://ww1.microchip.com/downloads/en/DeviceDoc/39625c.pdf>).

TALKING TO ROBOTS: SPEECH-DIRECTED MOTION PLANNING

NSF Summer Undergraduate Fellowship in Sensor Technologies
Anil Venkatesh (Electrical Engineering) – University of Pennsylvania
Advisor: Dr. George J. Pappas

ABSTRACT

In order to control a robot with spoken commands, concepts from speech recognition and motion planning must be integrated. This paper provides a solution to this problem, drawing on the Spoken Language Understanding Shell system developed by Schuler, and on recent work by Kress-Gazit et al. in temporal logic motion planning. Specifically, modifications and additions to the software are described, and excerpts provided, to demonstrate how the two systems are interfaced. Furthermore, successful performance of the new system in computer simulation is exhibited, allowing the user to direct a robot in complex scenarios with a minimal number of utterances. Finally, future work is discussed in multi-robot and environment-exploration applications.

1. INTRODUCTION

The task of controlling a robot with spoken commands requires the integration of concepts from the fields of speech recognition and motion planning. This paper describes such a system which is designed with the following two objectives, both stemming from general observations on the advantages of speech direction.

Spoken input requires less effort than written input. However, the benefit vanishes if the user is required to provide extensive verbal directions. In order to justify the pursuit of a speech-directed system, the inherent advantages of this medium of input must be preserved, and hence, the number and length of necessary spoken commands minimized. Therefore, the first objective of the project is economy of spoken commands.

Spoken input is more universally accessible than written input. While a speech-directed motion planner has applications where text input does not suffice, this is for naught if the system cannot easily be adapted for diverse situations. Therefore, the second objective of the project is to reduce the amount of experience in programming required to design new scenarios for the system.

This paper details the implementation of a speech-directed motion planner. Section 2 provides background information on the two fields and describes the software packages chosen for the project. Section 3 details the process of modifying and customizing the software for this project. Description and analysis of computer simulations appear in Section 4, while Section 5 contains a discussion of the results of the project and outlines immediate improvements to be made. Finally, Section 6 addresses the viability of futures extensions to live demonstration, environment exploration, and multiple-robot scenarios.

2. BACKGROUND

2.1 The SLUSH System

The Spoken Language Understanding Shell (SLUSH) system was chosen for this project for its innovative treatment of referential semantic information. As explained in [1], SLUSH uses a single integrated phonological, syntactic, and referential semantic language model, allowing the system to narrow down its search as more information is supplied by the speaker. For example, SLUSH naturally deals with over-determined utterances like “the glass on the table in the room” without incorrectly hypothesizing a glass not contained in a room [1]. This approach is well suited to the over-determined nature of human languages.

2.2 Grammar School

Integral to the SLUSH system is its grammar file. It would not suffice to supply the system with a list of English words: any machine would soon become overwhelmed by the number of possible (though perhaps nonsensical) ways to string those words together. Instead, SLUSH uses a grammar which describes which combinations of words may occur. A typical entry could define the concept of a prepositional phrase, as follows.

```
G PP -> (IN) in NP (INPRIME)
```

Here, G stands for “grammar” and precedes every entry. PP is the user-defined name of the structure, here denoting “prepositional phrase.” The words in NP describe what a PP looks like; defining NP below as a noun phrase like “the room,” SLUSH can now recognize “in the room” as a prepositional phrase.

```
G N -> (ROOM) room
G NP -> the N
```

Meanwhile, IN is the user-defined name for the relation mapping the *context set* to the set of entities which can occur before this PP, where context set refers to the set of entities which satisfied the utterance up to this point. Its counterpart, the *source set*, is the set of entities which satisfy the current argument: for the PP “in the room,” this would be the set of rooms in the world model. Of course, INPRIME is a relation on the source set instead of the context set.

However, which entities IN and INPRIME map to is not established until lexicon entries are written for them as well. The SLUSH lexicon accompanies the grammar, describing the semantics of every relation defined in the grammar. For example, consider the entry in the lexicon for the word “room”:

```
L ROOM : (set-of-all i in (source-set) s-t ((ilk of i) is (room)))
```

Similarly to the grammar, L denotes “lexicon” and precedes every entry. ROOM is the same user-defined name for the relation, just like IN and INPRIME, which links this entry to the corresponding one in the grammar. The rest of the rule reads that ROOM denotes all elements in the source set that are of type (ilk) room. Of course, the so called base ontology file is required to define which elements in the world model satisfy this stipulation in a straightforward way; that file is included along with the grammar and lexicon in Appendix A.

Finally, SLUSH must be told how to pronounce every word it is expected to recognize. The pronunciation file is its resource for this, listing for each word any possible pronunciations and a probability of occurrence for each. The pronunciations themselves are expressed in a standard phonetic format and the file can also be found in Appendix A.

2.2 The Most Likely Sequence

Integral to the SLUSH system is the determining of a Most Likely Sequence, or MLS, to fit the spoken input. This sequence is calculated in intervals of 10 milliseconds until the entire utterance is processed, and comes in the generic form:

```
[GO, e0 . e1] STA/N_PP*; [IDENT, e0] N/alan_N; [-, -] -/-; [-, -] -/-; ; /N_SIL? 5
```

```
[GO, e0 . e1] STA/N_PP*
```

The brackets indicate that the commands “go to element e0” or “go to element e1” are hypothesized; the second component is the current *syntactic state*: in this case, the utterance so far is missing a noun (N) and an optional prepositional phrase (PP*) to be a full sentence (denoted here by STA).

```
[IDENT, e0] N/alan_N
```

The brackets denote the current hypothesis that the element e_0 is the one intended by the speaker, and not e_1 . The syntactic state in this case indicates that the noun “alan_N” is expected (i.e. alan room—the name of a room in the example).

/N_SIL? 5

The number 5 indicates that this MLS was calculated during a phonetic transition: the speaker was in the middle of a word. Meanwhile, numbers less than 5 denote various syntactic-semantic transitions between actual words. In this case, it shows that the word being pronounced needs only an N followed optionally by silence to be identified as “alan.”

Applying this method to the following MLS, we discover that it predicts the sentence “go to the alan room,” between the words “alan” and “room.” The spurious e_1 will disappear from the list of referents in the next MLS, after the word “room” has been fully registered.

[GO, e_0 . e_1]STA/N_PP*; [IDENT, e_0]N/room; [-, -]-/-; [-, -]-/-; ; /R_UW_M_SIL? 2

2.3 Linear Temporal Logic and Motion Planning

The software implementing the algorithms in [2] is well suited to this project. Named LTLMop (from Linear Temporal Logic Motion Planner), it generates an automaton to control a robot from a set of structured English instructions, which are mapped to linear temporal logic (LTL) formulas.

As explained in [2], LTL is an extension of propositional logic to a sequence of states, corresponding to samples over time, so that the value of a proposition can change between *true* and *false* for each new state. In order to take this new complexity into account, two operators are added to the set of negation (\neg) and disjunction (\vee): from [2], $\bigcirc\varphi$ expresses that φ is true in the next state, while $\varphi_1\mathcal{U}\varphi_2$ denotes that φ_1 is true in every state until φ_2 becomes true. These operators can additionally be used to define \diamond (eventually) and \square (always) [2].

Rather than have the user meticulously write out linear temporal logic formulas, LTLMop uses regular expressions to support its structured English input scheme. For example, the sentence

If you are activating Action then visit Region

is converted to $\square\diamond(\varphi \Rightarrow \psi)$ where φ is the proposition for Action and ψ is the proposition for Region. In this case, the sentence defines a *liveness condition*, read as “always eventually, φ implies ψ .” These formulas are then used to compose a controller for the robot, as detailed in [2].

3. METHODS

3.1 A New Grammar

The grammar-cum-lexicon file initially provided by SLUSH supports sentences of the form: “go to the [noun]” followed by any number of prepositional phrases; for instance, “go to the chair in the room to the left of the table.” The grammar and lexicon entry covering all of this is:

```
G Simp -> go to NP (GO)
L GO : (source-set)
```

In other words, *Simp* is the name given to simple “go to” commands, which can be applied to any entity in the world model since the lexicon entry for *GO* puts no constraints on the source set.

However, the iconic if-then command, on which any meaningful robot motion planning depends, has no implementation in the default SLUSH grammar. To rectify this situation, we must introduce new grammar and lexicon entries. For example, the construction “if you see the [noun], go to...” can be added to the original grammar in the following way (courtesy of William Schuler).

```
G Simp -> PPcond Simp
G PPcond -> if Sind (IF)
G Sind -> you see NP

L IF : (context-set)
```

Again, *PPcond* and *Sind* are the user-specified names for the if-clause and the sentence it contains, respectively. Furthermore, setting *IF* to the context set in the lexicon means in this case that its referents will be whatever entities immediately follow “if you see.” This way, the referents of the if-clause will not interfere with those in the subsequent “go to” clause. With this addition, the software now recognizes “if you see the table in the room, go to the chair to the left of the table” as well as a simple “go to” command as before.

Of course, the grammar must be further modified, since it is reasonable to expect a robot to do more than visit locations on visual triggers. The result and its expressivity are detailed in the next section.

3.2 Adapting SLUSH

Clearly, processing the MLS is the key to extracting spoken commands from SLUSH. By first filtering the phonetic transitions, the words can then be extracted by cross-referencing by the phonetic representation in the pronunciation dictionary (see Appendix A). But the true power of the SLUSH system lies in the incorporation of semantic information directly into its language model, allowing for successful interpretation of phrases like “the chair to the left of the table.” For this reason, it is also crucial to extract, for example, the referent e_0 from $[GO, e_0]$, since SLUSH may have deduced this perfectly precise command from highly imprecise language.

In order to indicate the semantics more transparently in the MLS, we adapt the form of the conditional clause as follows (# indicates removal):

```
G Simp -> go to NP (GO)
G Simp -> PPcond Simp
#G PPcond -> if Sind (IF)
#G Sind -> you see NP
G PPcond -> if you see NP (IF-SEE)
G PPcond -> if you are in NP (IF-ARE)
G PPcond -> if you catch the convict (IF-CATCH)

#L IF : (context-set)
L IF-SEE : (context-set)
L IF-ARE : (context-set)
L IF-CATCH : (context-set)
```

In this way, each type of condition is clearly marked along with its referents in the MLS, for example:

```
[IDENT,e0.e1]STA/PPcond_Simp;[IF-SEE,e12]PPcond/sil;[-,-]-/-;[-,-]-/-;;/SIL 2
```

which results at the end of the clause “if you see the chair in the alan room” (this chair is indeed denoted by e12).

Using regular expressions, the words and referents are extracted from the MLS, resulting in an output of the following form (where the beta room corresponds to e1):

```
if you see the chair in the alan room go to the beta room
```

```
Command Referents: e1
Conditional Referents: e12
```

It will prove advantageous to demarcate the two clauses with the word “then”; accordingly, a provision is included in the code for MLS processing to insert the line /DH_EH_N_SIL? into the MLS dump in the transition from if-statement to command. As long as this word is added to the pronunciation file, it will be appended to the sentence automatically, resulting in the following.

```
if you see the chair in the alan room then go to the beta room
```

```
Command Referents: e1
Conditional Referents: e12
```

The final step of MLS processing is oddly necessitated by the proficiency of the SLUSH system in reducing information like “the chair in the alan room” to e12, the element to which the phrase refers. While this is the primary strength of SLUSH, it also obscures the location of that chair, which could be a point of interest for a motion planner. In order to retrieve this information as well, we must scan the MLS dump for the entry where the word “in” is resolved, leaving its inverse, INPRIME, and the corresponding room referent. The following example is from the same sentence as above.

```
...[IF-SEE,e12]PPcond/PP_PP*;[INPRIME,e0]PP/sil;[-,-]-/-;;/SIL 3
```

This information is appended to the processed MLS except when the clause deals with an action (such as `IF-CATCH` above); in that case, recording the location is sufficient. We finally arrive at the following format.

if you see the chair in the alan room then go to the chair in the beta room

Command Referents: e1/e10

Conditional Referents: e0/e12

3.3 Quasi-English Templates

While LTMop provides a way for the user to write LTL formulas in a quasi-English medium, even simple scenarios such as those described in [2] require perhaps twenty such formulas to be fully specified. For example, the sentence

`If you are activating Action_1 then visit Region_1`

would be of little use in most applications if not accompanied by additional rules to govern when `Action_1` should be activated, perpetuated, and deactivated. This presents a problem for spoken input, since the advantage of delivering commands verbally is quickly outweighed by the burden of keeping so many particular instructions in mind.

To address this concern, a system to generate many quasi-English sentences from a single spoken instruction has been developed. Every sentence delivered by SLUSH is identified as one of the supported commands, and its information is expressed by an appropriate set of quasi-English template sentences. For example, the verbal direction “if you are in the yard, go to the office” generates the following four sentences.

```
If you are in yard then do yardFlag
If you activated yardFlag and you were not in office then do yardFlag
If you are activating yardFlag then visit office
If you were in office do not yardFlag
```

The custom proposition `yardFlag` is introduced in the first sentence so that the robot “remembers” that it had visited the region yard. The second sentence assures that `yardFlag` is reactivated at every new stage, since its value will otherwise change serendipitously. Of course, the third sentence describes the intended behavior, that the robot visit the office; and the fourth sentence absolves the robot of its duty once this task has been performed.

More elaborate commands involve a greater number of the template sentences, a complete list of which is included in Appendix B. In order to

reduce the number of sentences necessary, the symbols { } are introduced around optional information, which is either removed or retrieved depending on the semantics of the utterance.

Once all utterances have been delivered, the sentences are compiled into a list. Furthermore, any actions which were mentioned are listed separately; the same goes for sensors, which arise in utterances of the form `if you see Sensor_1 then....`. Finally, custom propositions such as `yardFlag` above are indexed in order of utterance, resulting for instance in `yardFlag1`. This ensures that every custom proposition is unique.

In addition to the automatically generated sentences, certain basic conditions are specified by the user beforehand, in the same way that the grammar is customized for any particular scenario. These may include the initial conditions `Environment starts with false` and `Robot starts with false`. Furthermore, liveness conditions such as “visit every room” are currently not supported for spoken input, and must also be specified beforehand.

4. OPERATION AND SIMULATIONS

The system developed in this project provides an intuitive way to issue a set of verbal instructions to a robot. Guided by the SLUSH system map of the world model (Figure 1), the user depresses the control key and delivers a command, repeating as necessary. She then loads the generated file into LTLMop, which automatically creates a controller for the robot, satisfying the specifications of the user. The automaton can then be tested in Stage or Gazebo simulation.

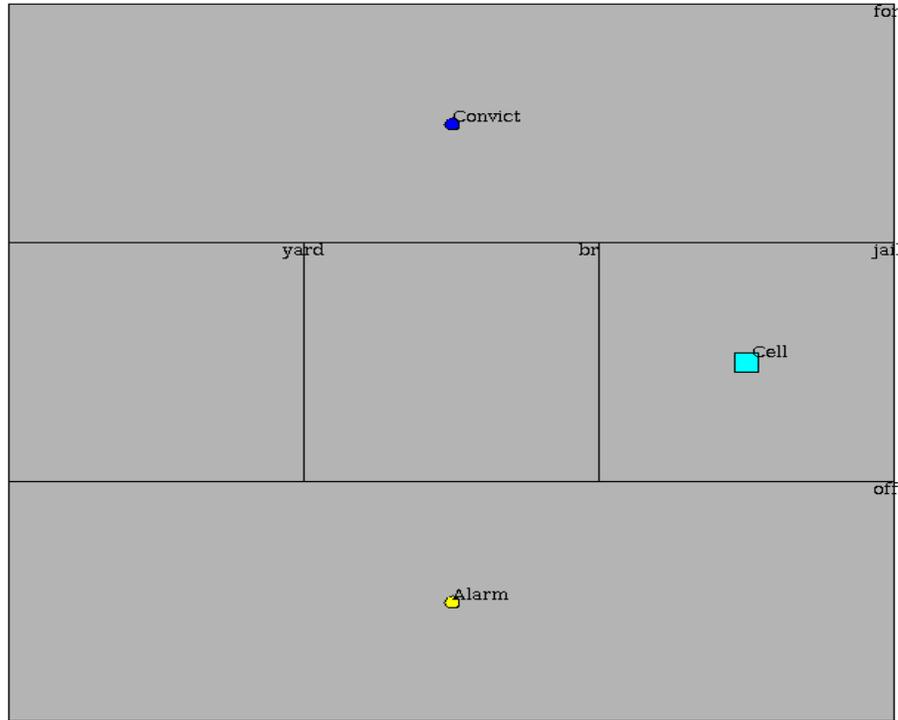


Figure 1 - SLUSH System Map

The following simulations are based on this scenario: a robot is moving about in the world depicted in Figure 2. An alarm is ringing in the office and a convict is hiding in the forest. We issue the following commands:

1. If you see the alarm in the office, catch the convict in the forest.
- 2. If you catch the convict in the forest, lock him up in jail.**

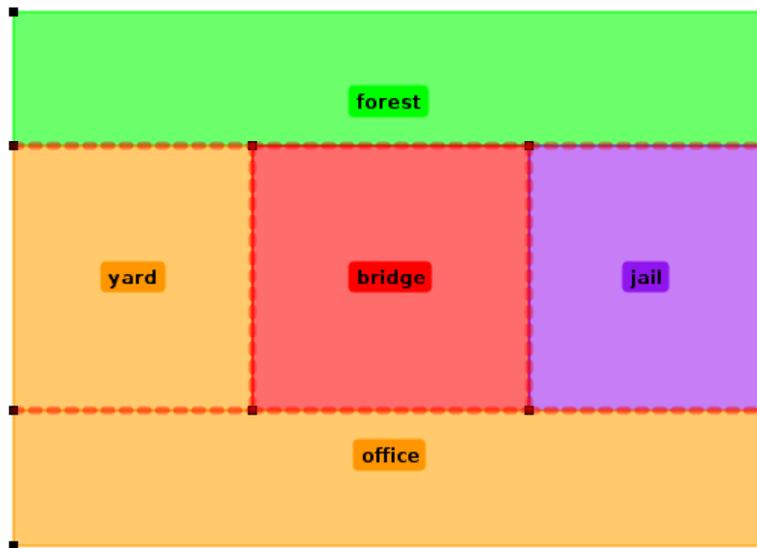


Figure 2 - World Model

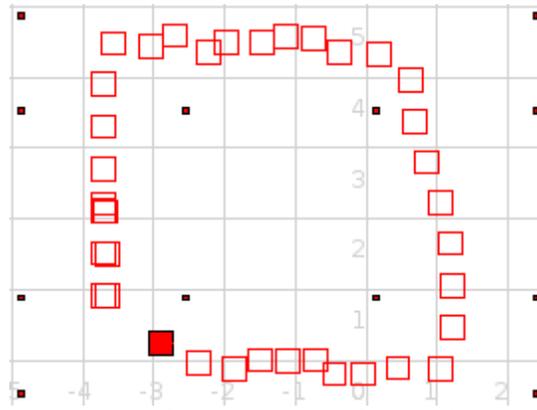


Figure 4 - Bridge closed for repairs

5. DISCUSSION

The expressivity achieved in this project is summed up in Table 1. Of course, any of the clauses in the first column may be paired clauses in the second. The final grammar, which implements the various combinations expressed in above, appears in Appendix A.

Table 1: Expressivity of final grammar

IF	THEN
SENSOR [in LOCATION_1]	go to LOCATION_2
in LOCATION_1	ACTION_2 [in LOCATION_2]
ACTION_1 [in LOCATION_1]	

The two primary goals of this project were met. Using template sentences and semantic analysis of each utterance, the number of spoken commands necessary has been kept small. Furthermore, the system is designed to accommodate customization by the user; a new scenario can be crafted solely by modifying the SLUSH files in Appendix A, and drawing the world map in the graphical region editor of LTLMop.

A number of features could be added to the system to improve its functionality. The additions require modest amounts of coding and should be supported in the next version.

For example, it would be worthwhile to implement support for utterances with multiple referents. It is reasonable to expect that the user may not care to specify a single entity in the world model every time. Accordingly, the MLS processor should generate LTL formulas involving all possible referents, which can be done with no increase in the number of logical propositions.

Conjunction support is another useful aspect for the next revision. By adding

G Simp -> Simp and Simp

to the grammar, some modifications to the MLS processing algorithm will allow this considerable enrichment to the expressivity of the system.

Furthermore, the current system requires text input beforehand to determine liveness conditions. The project encountered difficulties in automatically generating such sentences because the conditions governing them may be complicated: ultimately, there could be a unique formula governing the visitation of each individual region in the world model. Given that the philosophy of the project is economy of spoken commands, the specification of such a scenario would be burdensome indeed. However, support for sentences such as “go everywhere but [region]” would allow the user a measure of control over “visitation rights” without unduly increasing the number of utterances required.

6. RECOMMENDATIONS

The success of this project in simulation prompts a demonstration with real robots. Since LTLMop supports Player, this should be a relatively straightforward extension in the future.

Multi-robot scenarios also provide an interesting opportunity for future work. This is a natural progression from the current system with each robot treating the others as part of its environment [2]. The prospect of instructing many robots with a single command is an intriguing one, which will certainly require attention to the way the temporal logic formulas are currently generated.

A more radical extension to this project is the idea of a robot exploring a previously unmapped area. Establishing a feedback loop between speech recognizer and motion planner (that is, essentially between robot and human), the robot in this new system could coin new terms for the objects and regions it discovered, sending maps and images back to the user. Meanwhile, the user could remotely, verbally direct the robot in further exploration.

7. ACKNOWLEDGMENTS

I would like to thank my advisors, Dr. George Pappas and Hadas Kress-Gazit of the University of Pennsylvania, for their advice and encouragement. I would also like to thank Dr. William Schuler, Stephen Wu, and Luan Nguyen of the University of Minnesota for help and collaboration regarding their SLUSH system. Finally, I would like to thank the National Science Foundation for their continued support of the SUNFEST program.

8. REFERENCES

1. W. Schuler, S. Wu, L. Schwartz, A Framework for Fast Incremental Interpretation during Speech Decoding, *Computational Linguistics*, in press.
2. H. Kress-Gazit, G.E. Fainekos, G.J. Pappas, Where’s Waldo? Sensor-Based Temporal Logic Motion Planning, *IEEE Conference on Robotics and Automation, Rome, Italy, April 2007*.

APPENDIX A

Grammar

G START -> STAA sil
G STAA -> STA
G STA -> sil Simp
G Simp -> Comm
G Comm -> go to NP (GO)
G Comm -> catch the convict PP (DO-CATCH)
G Comm -> catch him PP (DO-CATCH)
G Comm -> lock up the convict PP (DO-LOCK)
G Comm -> lock him up PP (DO-LOCK)
G NP -> the N PP*
G PP* -> PP PP*
G PP* -> PPneg PP*
G PP* -> sil
G PP -> (CONTAIN) containing NP (CONTAINPRIME)
G PP -> (IN) in NP (INPRIME)
G PP -> sil
G PPneg -> not PP (NOT)
G PPneg -> that are not PP (NOT)
G Simp -> PPcond Comm
G PPcond -> if you see NP (IF-SEE)
G PPcond -> if you are in NP (IF-ARE)
G PPcond -> if you catch the convict PP (IF-CATCH)
G PPcond -> if you catch him PP (IF-CATCH)
G PPcond -> if you lock up the convict PP (IF-LOCK)
G PPcond -> if you lock him up PP (IF-LOCK)
G N -> (YARD) yard
G N -> (OFFICE) office
G N -> (FOREST) forest
G N -> (JAIL) jail
G N -> (BRIDGE) bridge
G N -> (CONVICT) convict
G N -> (ALARM) alarm
G N -> (CELL) cell

Lexicon

L YARD : (set-of-all i in (source-set) s-t ((name of i) is (Yard)))
L OFFICE : (set-of-all i in (source-set) s-t ((name of i) is (Office)))
L FOREST : (set-of-all i in (source-set) s-t ((name of i) is (Forest)))
L JAIL : (set-of-all i in (source-set) s-t ((name of i) is (Jail)))
L BRIDGE : (set-of-all i in (source-set) s-t ((name of i) is (Bridge)))
L CONVICT : (set-of-all i in (source-set) s-t ((full of i) is (1)))
L ALARM : (set-of-all i in (source-set) s-t ((empty of i) is (1)))
L CELL : (set-of-all i in (source-set) s-t ((ilk of i) is (chair)))
L CONTAIN : (reach-of (source-set) in (dir-graph-from-all i to j s-t ((i is-not j) and (((abs-of ((y of i) minus (y of j))) l-t-eq (yradius of i)) and ((abs-of ((x of i) minus (x of j))) l-t-eq (xradius of i))))))
L CONTAINPRIME : (intersection-of (context-set) with (reach-of (source-set) in (dir-graph-from-all i to j s-t ((i is-not j) and (((abs-of ((y of i) minus (y of j))) l-t-eq (yradius of j)) and ((abs-of ((x of i) minus (x of j))) l-t-eq (xradius of j))))))
L IN : (reach-of (source-set) in (dir-graph-from-all i to j s-t ((i is-not j) and (((abs-of ((y of i) minus (y of j))) l-t-eq (yradius of j)) and ((abs-of ((x of i) minus (x of j))) l-t-eq (xradius of j))))))
L INPRIME : (intersection-of (context-set) with (reach-of (source-set) in (dir-graph-from-all i to j s-t ((i is-not j) and (((abs-of ((y of i) minus (y of j))) l-t-eq (yradius of i)) and ((abs-of ((x of i) minus (x of j))) l-t-eq (xradius of i))))))
L NOT : (exclusion-of (source-set) from (context-set))
L GO : (source-set)
L DO-CATCH : (source-set)
L DO-LOCK : (source-set)
L IF-SEE : (context-set)
L IF-ARE : (context-set)
L IF-CATCH : (context-set)
L IF-LOCK : (context-set)
L CONTAIN : (reach-of (source-set) in (dir-graph-from-all i to j s-t ((i is-not j) and (((abs-of ((y of i) minus (y of j))) l-t-eq (yradius of i)) and ((abs-of ((x of i) minus (x of j))) l-t-eq (xradius of i))))))
L CONTAINPRIME : (intersection-of (context-set) with (reach-of (source-set) in (dir-graph-from-all i to j s-t ((i is-not j) and (((abs-of ((y of i) minus (y of j))) l-t-eq (yradius of j)) and ((abs-of ((x of i) minus (x of j))) l-t-eq (xradius of j))))))
L IN : (reach-of (source-set) in (dir-graph-from-all i to j s-t ((i is-not j) and (((abs-of ((y of i) minus (y of j))) l-t-eq (yradius of j)) and ((abs-of ((x of i) minus (x of j))) l-t-eq (xradius of j))))))
L INPRIME : (intersection-of (context-set) with (reach-of (source-set) in (dir-graph-from-all i to j s-t ((i is-not j) and (((abs-of ((y of i) minus (y of j))) l-t-eq (yradius of i)) and ((abs-of ((x of i) minus (x of j))) l-t-eq (xradius of i))))))

L NOT : (exclusion-of (source-set) from (context-set))

Pronunciation

P alarm : /AH_L_AA_R_M_SIL? = 1
P are : /ER_SIL? = 1
P bridge : /B_R_IH_JH_SIL? = 1
P catch : /K_AE_CH_SIL? = 0.5
P cell : /S_EH_L_SIL? = 1
P containing : /K_AH_N_T_EY_N_IH_NG_SIL? = 1
P convict : /K_AA_N_V_IH_K_T_SIL? = 0.5
P convict : /K_AH_N_V_IH_K_T_SIL? = 0.5
P forest : /F_AO_R_AH_S_T_SIL? = 0.5
P forest : /F_AO_R_IH_S_T_SIL? = 0.5
P go : /G_OW_SIL? = 1
P having : /HH_AE_V_IH_NG_SIL? = 1
P him : /HH_IH_M_SIL? = 0.5
P him : /IH_M_SIL? = 0.5
P if : /IH_F_SIL? = 1
P in : /IH_N_SIL? = 1
P is : /IH_S_SIL? = 1
P it : /IH_T_SIL? = 1
P jail : /JH_EY_L_SIL = 1
P lock : /L_AA_K_SIL? = 1
P not : /N_AA_T_SIL? = 1
P office : /AH_F_AH_S_SIL? = 0.5
P office : /AH_F_IH_S_SIL? = 0.5
P on : /AH_N_SIL? = 1
P see : /S_IY_SIL? = 1
P sil : /SIL = 1
P that : /DH_AE_T_SIL? = 1
P the : /DH_AH_SIL? = 0.7
P the : /DH_IY_SIL? = 0.3
P then : /DH_EH_N_SIL? = 1
P thing : /TH_IH_NG_SIL? = 1
P to : /T_AH_SIL? = 0.3
P to : /T_IH_SIL? = 0.2
P to : /T_UW_SIL? = 0.5
P up : /AH_P_SIL? = 1
P yard : /Y_AA_R_D/SIL? = 1
P you : /Y_UW_SIL? = 1

Base Ontology

V yard ilk : room
V yard name : Yard
V yard color : light_gray
V yard x : 200
V yard y : 400
V yard xradius : 100
V yard yradius : 100
V yard shape : r
V office ilk : room
V office name : Office
V office color : light_gray
V office x : 400
V office y : 600
V office xradius : 300
V office yradius : 100
V office shape : r
V forest ilk : room
V forest name : Forest
V forest color : light_gray
V forest x : 400
V forest y : 200
V forest xradius : 300
V forest yradius : 100
V forest shape : r
V jail ilk : room
V jail name : Jail
V jail color : light_gray
V jail x : 600
V jail y : 400
V jail xradius : 100
V jail yradius : 100
V jail shape : r
V bridge ilk : room
V bridge name : Bridge
V bridge color : light_gray
V bridge x : 400

```

V bridge y : 400
V bridge xradius : 100
V bridge yradius : 100
V bridge shape : r
V Alarm ilk : glass
V Alarm x : 400
V Alarm y : 600
V Alarm empty : 1
V Alarm shape : e
V Alarm color : yellow
V Convict ilk : glass
V Convict x : 400
V Convict y : 200
V Convict full : 1
V Convict shape : e
V Convict color : blue
V Cell shape : r
V Cell color : cyan
V Cell ilk : chair
V Cell x : 600
V Cell y : 400
V Cell xradius : 8
V Cell yradius : 8

```

APPENDIX B

Template Sentences

1. If you are in Location1 then do Location1Flag
2. If you activated Location1Flag and you were not in Location2 then do Location1Flag
3. {If you are activating Location1Flag then visit Location2}
4. If you were in Location2 do not Location1Flag
5. Do Action2 if and only if you are activating Location1Flag {and you were in Location2}
6. If you did not activate Action2 and you are activating Action2 then stay there
7. If you activated Action2 then do not Location1Flag
8. If you are sensing Sensor {and you were in Location1} then do SensorFlag
9. If you activated SensorFlag and you were not in Location2 then do SensorFlag
10. {If you are activating SensorFlag then visit Location2}
11. If you were in Location2 then do not SensorFlag
12. If you activated SensorFlag and you did not activate Action2 then do SensorFlag
13. If you activated Action2 then do not SensorFlag
14. Do Action2 if and only if you are activating SensorFlag {and you were in Location2}
15. If you activated Action1 {and you were in Location1} then do Action1Flag
16. If you activated Action1Flag and you were not in Location2 then do Action1Flag
17. {If you are activating Action1Flag then visit Location2}
18. If you were in Location2 then do not Action1Flag
19. If you activated Action1Flag and you did not activate Action2 then do Action1Flag
20. If you activated Action2 then do not Action1Flag
21. Do Action2 if and only if you are activating Action1Flag {and you were in Location2}

Structured English Example

Environment starts with false
 Robot starts with false

```

Visit yard
Visit office
Visit forest
Visit jail
Visit bridge

```

```

If you did not activate Catch and you are activating Catch then stay there
If you are sensing alarm and you were in office then do alarmFlag1
If you are activating alarmFlag1 then visit forest
If you activated alarmFlag1 and you did not activate Catch then do alarmFlag1
If you activated Catch then do not alarmFlag1
Do Catch if and only if you are activating alarmFlag1 and you were in forest

```

```

If you did not activate Lock and you are activating Lock then stay there
If you activated Catch and you were in forest then do catchFlag2
If you are activating catchFlag2 then visit jail
If you activated catchFlag2 and you did not activate Lock then do catchFlag2
If you activated Lock then do not catchFlag2
Do Lock if and only if you are activating catchFlag2 and you were in jail

```

Propositions

```

Sensors: alarm
Actions: Catch, Lock
Customs: alarmFlag1, catchFlag2

```

INVESTIGATION OF MSC DIFFERENTIATION ON ELECTROSPUN NANOFIBROUS SCAFFOLDS

NSF Summer Undergraduate Fellowship in Sensor Technologies
Emily Wible (Bioengineering) – University of Pennsylvania
Advisor: Robert L. Mauck and Nandan L. Nerurkar

ABSTRACT

Tissue engineering has been utilizing electrospun nanofibrous scaffolds as a platform for growing mesenchymal stem cells (MSCs). Electrospinning allows for the fabrication of scaffolds with a controllable degree of fiber alignment. Previous work has established that variation in the degree of fiber alignment directly effects the shape of the cells seeded onto these scaffolds. This study explored the effect of scaffold fiber alignment, and hence cell shape, on MSC differentiation. Two groups were observed in this study—aligned and non-aligned scaffolds. The scaffolds were seeded with MSCs and grown in a chemically defined culture medium for ten days. The effect of alignment was inspected using fluorescent microscopy to visualize cell morphology, biochemical assays to measure DNA content and glycosaminoglycan production, and real time PCR to determine gene expression. It was hypothesized that aligned scaffolds, in which cells are elongated in shape, would result in an up regulation of collagen type I, a marker of fibrous tissue differentiation, while non-aligned scaffolds would induce an up regulation of collagen type II, a cartilaginous marker. The data supported the second part of our hypothesis, that collagen type II would be up regulated in non-aligned scaffolds. However, collagen type I was not different between the two scaffold types. Because collagen type I is not unique to fibrous tissues, more specific markers are currently being investigated, including tenomodulin and lysyl oxidase. In conclusion, scaffold disorganization promoted chondrogenic differentiation of MSCs when compared to highly aligned fibers. These results support the contention that cell shape plays an important role in the differentiation of MSCs along fibro-cartilagenous phenotypes.

1. INTRODUCTION

The long term goal of tissue engineering is to create functional tissue that replaces injured or degenerated native tissue. Currently, mesenchymal stem cells (MSCs) have become a popular cell source due to their ability to differentiate into various cells, including chondrocytes, osteoblasts, and adipocytes. However, the mechanisms that dictate the differentiation process remain poorly understood. Therefore, it is necessary to explore the mechanisms that direct a cell towards a specific lineage.

Recently, nanofibrous electrospun scaffolds have become a possible platform for growing new tissue. Unfortunately, it is not well understood which scaffold features direct a cell towards a specific cell type. In previous research, it has been shown that cell shape is important in differentiation [1]. For example, cells residing in fibrous tissues have an elongated shape, while cartilaginous cells have a more rounded contour. In order to help encourage the MSCs on scaffolds to adopt a particular cell shape, fiber alignment has been manipulated to be either highly aligned or non-aligned. The hope is that aligned fibers will encourage the MSCs to adopt a fibrous phenotype, while the non-aligned fibers will encourage the MSCs to adopt a chondrogenic phenotype. This research study focuses on scaffold fiber orientation as a possible means to direct MSCs towards a specific phenotype.

2. BACKGROUND

MSCs are able to differentiate into various cells including chondrocytes, osteoblasts, and adipocytes. This variability makes MSCs an important resource for generating new tissues. However, it is important to determine the environmental conditions which direct MSCs towards a particular cell type. In order to assess the effect of an environmental condition, the expression of tissue specific genes can be used to indicate the phenotype of differentiated cells. SOX 9, a

transcription factor, and collagen type II, an extracellular matrix protein, are both present primarily in chondrocytes, and therefore serve as markers of chondrogenic differentiation. In comparison, collagen I is more common in fibrous tissues such as tendon and ligament, and is therefore a marker of fibrous differentiation.

The MSCs in this research project were grown on electrospun nanofibrous scaffolds. Electrospun scaffolds have become an important area of research due to their ability to have an aligned or non-aligned fiber arrangement [2]. Aligned scaffolds have fibers oriented parallel to each other, while non-aligned scaffolds have fibers oriented randomly. Previous work has shown that cells grow along the fiber orientation [4]. This study hypothesized that aligned scaffolds, in which cells are elongated in shape, would result in an up regulation of collagen type I, a marker of fibrous tissue differentiation, while non-aligned scaffolds would induce an up regulation of collagen type II and SOX9, both cartilaginous markers.

3. METHODS

3.1 CELL CULTURE

Bovine MSCs were seeded on nanofibrous electrospun scaffolds for 10 days. Initially, bovine marrow was harvested from the femur and tibia. This marrow was then placed in tissue culture plastic dishes, containing basal media to allow the MSCs to crawl out. Once the MSCs were expanded to passage 2, they were seeded on the aligned and nonaligned electrospun scaffolds. From this point on, the MSCs were grown in a chemically defined growth media containing TGF- β 3. At 3, 7, and 10 days of culture, real time PCR, actin staining, and biochemical assays were performed. For each time point, an extra group was added that was treated with cytochalasin. Cytochalasin depolymerizes actin, the primary cytoskeletal element of cells which gives rise to and maintains cell shape.

3.2 FLOURESENCE MICROSCOPY

Fluorescence microscopy was used to visualize both the nucleus and stress fibers of the cells on the scaffolds. The stress fibers provided not only a rough outline of the cell shape, but also an indication of whether the MSCs were growing along the fiber alignment. Samples were stained with phalloidin conjugated alexafluor to visualize the actin cytoskeleton, and counterstained with DAPI to visualize cell nuclei. Blue light was used to visualize the actin, while UV light was used to visualize the nucleus of the cells. Pictures were collected from both types of light at 20x and then overlaid into a single picture displaying both the nucleus and actin.

3.3 REAL TIME PCR (POLYMERASE CHAIN REACTION)

Real time PCR was used to simultaneously amplify and quantify a specified DNA product (determined by the genetic markers). The DNA content was quantified based on the intensity of fluorescence. SYBR Green, a DNA-binding dye, attached to any double-stranded DNA, causing the dye to fluoresce. As the amount of double-stranded DNA increased, so did the fluorescence. In order to make the results more accurate, a house-keeping gene was used in comparison to the gene of interest. The challenge with SYBR Green is that it is non-specific and will attach to all double-stranded DNA. Therefore, RNA samples were treated with DNase prior to amplification to prevent genomic contamination. The genes observed in this study were SOX9, collagen type I, collagen type II, and GAPDH (house-keeping gene).

3.4 BIOCHEMICAL ASSAYS

Two different biochemical assays were used in this study. The first biochemical assay determined the amount of GAG in the scaffolds. Before the scaffolds were analyzed for GAG content, the scaffolds had to undergo a papain digest for 24 hours in a 60°C water bath. The papain digest consisted of 20 µl of papain per ml of papain digest buffer. After digestion, the

scaffolds were analyzed for GAG content using 1,9-dimethylmethylene blue dye-binding (DMMB) assay. Standard samples of known concentrations of chondroitin-6-sulfate were run with the scaffold samples in order to calibrate the measurements. The presence of GAG served as a measure of chondrogenic differentiation, as GAG is present in high quantities in cartilaginous tissues.

The second biochemical assay determined the amount of DNA present in the scaffolds. Before the scaffolds were analyzed for DNA content, they had to undergo a papain digest as described above. After digestion, the scaffold samples were analyzed for DNA content using the Picogreen ds-DNA assay. Standard samples of known concentrations were run with the scaffold in order to calibrate the measurements. The presence of DNA allowed for the determination of GAG/DNA so that GAG production could be analyzed without being obscured by varying cell quantities.

4. RESULTS

Figure 1, 2, and 3 display the ratio of the specified genetic marker (COLI, COLII, or SOX9) to the house-keeping gene (GAPDH). The house-keeping gene was used to normalize the results and account for varying cell contents. Significance in this study was determined by performing a two-tailed t-test with an $\alpha=0.05$. Although there is no significant difference at day 10 in Figure 1, collagen type II is indeed significantly higher at day 10 in non-aligned scaffolds. SOX9, a transcription factor known to up regulate collagen type II expression was observed to confirm our finding that chondrogenesis is enhanced on nonaligned scaffolds. As shown on Figure 3, the collagen type II up regulation at day 10 coincided with up regulation of SOX9. It should be noted that the non-aligned group for day 10 only had a sample size of $n=1$.

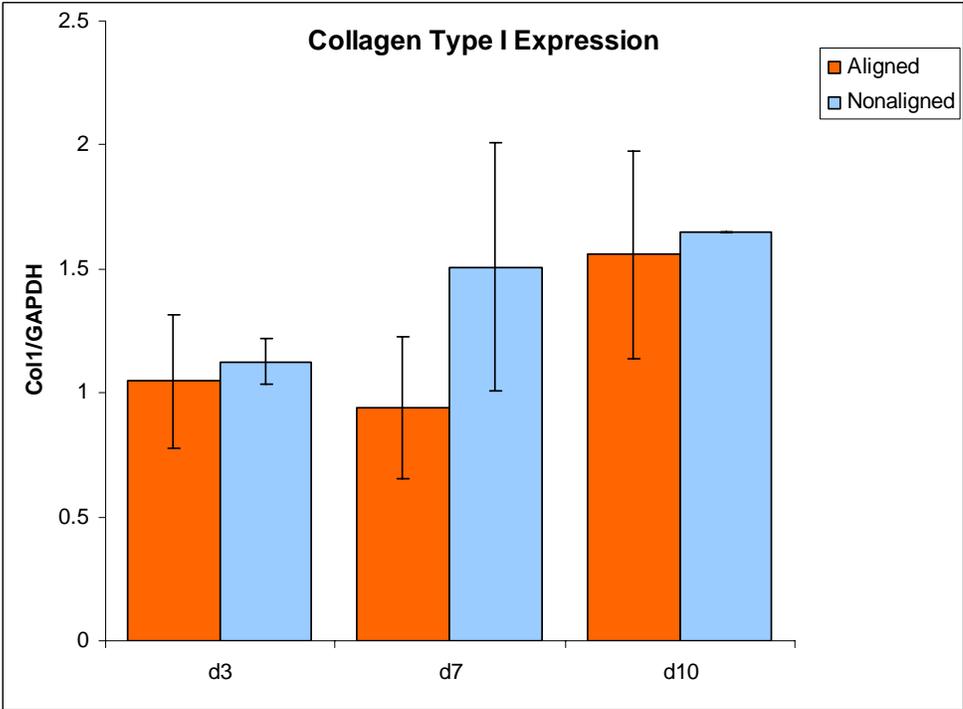


Figure 1.

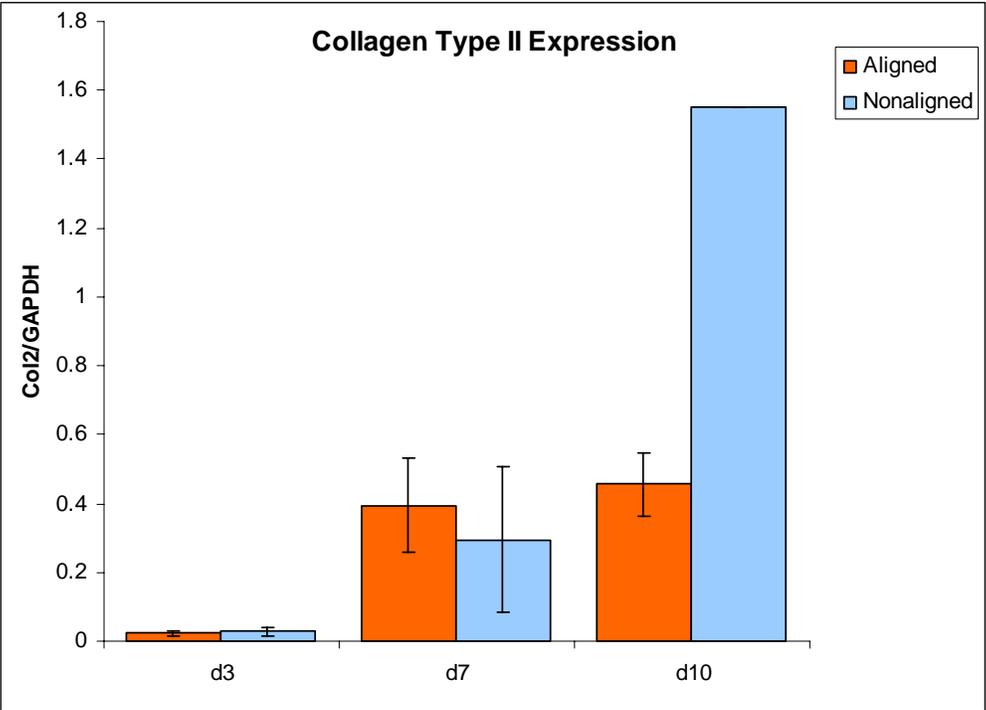


Figure 2.

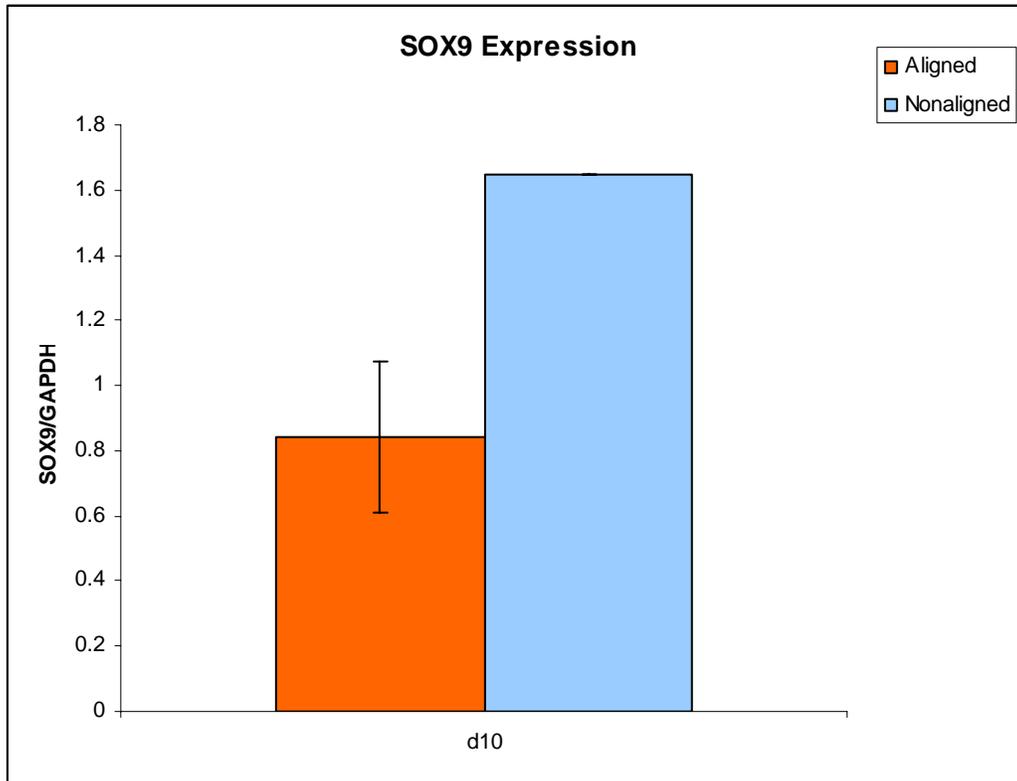
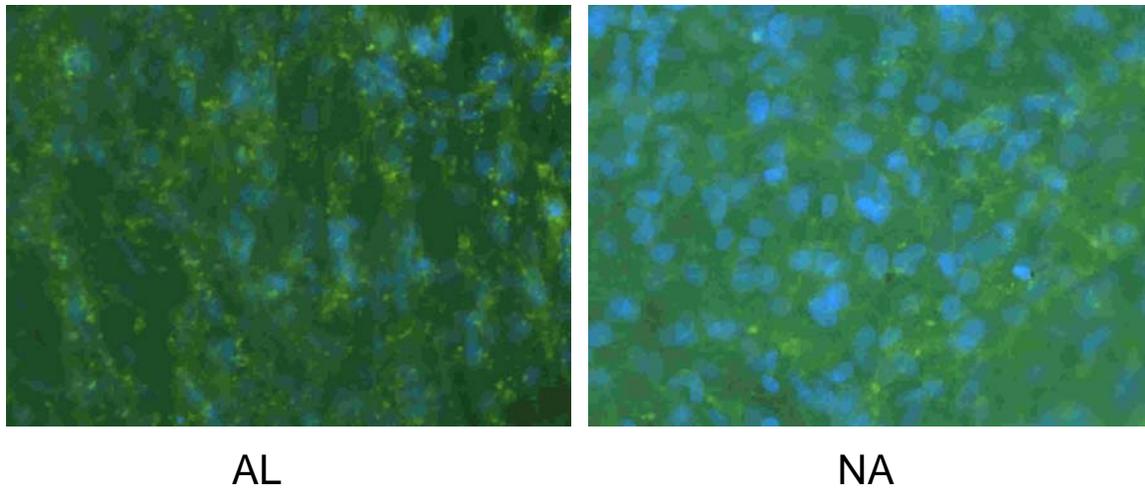


Figure 3.

Figure 4 is a picture of the cytochalasin treated groups (aligned and non-aligned) at day 10. The stress fibers have been disassembled and the green is seen scattered throughout the image, with blue nuclei indicating the presence of a cell.



AL

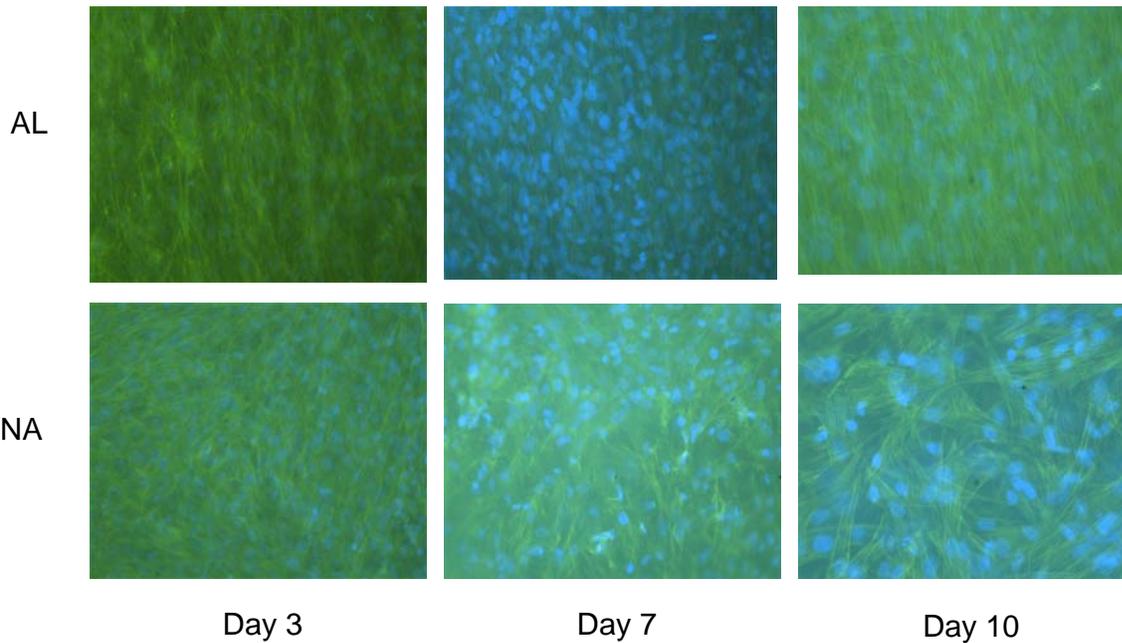
NA

Day 10

Figure

4.

Figure 5 displays the fluorescence microscopy of the non-aligned (bottom row) and aligned (top row) scaffolds for day 3, 7, and 10.



AL

NA

Day 3

Day 7

Day 10

Figure 5.

The biochemical assay results are not depicted in the final paper; however, there were no significant differences observed between aligned and nonaligned scaffolds. There was, however,

a consistent reduction of GAG production on both scaffold types when the scaffolds were treated with cytochalasin.

5. DISCUSSION AND CONCLUSIONS

As indicated by Figures 2 and 3, we observed an up regulation of SOX9 and collagen type II on the non-aligned scaffolds when compared to that observed on the aligned scaffolds. This partially supported our hypothesis that non-aligned scaffolds would encourage cells toward a cartilaginous type. The cells were encouraged to become cartilaginous because of the non-aligned fibers, mimicking a more rounded shape. Hence, shape plays an important part in directing the phenotype of a differentiated MSC.

On the contrary, as indicated in Figure 1, we did not observe any change in the presence of collagen type I on the aligned scaffolds when compared to the non-aligned scaffolds. Although the bar graph shows a difference on d7, this difference becomes insignificant at time point day 10 (tested with two-tailed t-test). Because collagen type I is expressed in undifferentiated MSCs, as well as chondrogenically differentiated MSCs, it may not be a good marker for fibrous differentiation. Currently, other markers are being investigated that might better confirm a fibrous cell's phenotype.

Figure 4 confirms that cytochalasin interrupts the actin filaments in the cell. After the actin has been disrupted, the cell no longer displays its original shape. The stress fibers are not as visible as they normally are on the scaffolds. These observations coincided with a loss of GAG production, and with a reduction in type II collagen expression.

Actin staining (Figure 5) visually showed the elongated shape of MSCs on aligned scaffolds and the rounded shape of MSCs on non-aligned scaffolds. This was seen for all time points (days 3, 7, and 10). These morphologies were outlined by the stress fibers' florescence in

green and the nuclei's fluorescence in blue. The actin staining also confirmed that the MSCs were growing along the fiber direction (aligned or non-aligned).

6. RECOMMENDATIONS

Currently, more specific genes for fibrous differentiation are being investigated to better indicate a fibrous phenotype. The two genes being investigated currently are tenomodulin and lysyl oxidase. These genes are being considered because they are up regulated in fibrous tissue.

7. ACKNOWLEDGMENTS

I would first like to thank the National Science Foundation for their support through an NSF-REU grant and Microsoft Corporation for their financial support. I would also like to thank my advisor, Robert L. Mauck, for selecting me as a summer research student in his lab. I would finally like to thank my graduate student, Nandan L. Nerurkar, for all of his assistance with my summer project.

8. REFERENCES

- [1] J. Settleman, "Tension Precedes Commitment—Even for a Stem Cell," *Molecular Cell*, vol. 14, pp. 148-150, 4/23. 2004.
- [2] T. J. Sill and H. A. von Recum, "Electrospinning: Applications in drug delivery and tissue engineering," *Biomaterials*, vol. 29, pp. 1989-2006, 5. 2008.
- [3] Xu, J., Wang, W., Ludeman, M., Cheng, K., Hayami, T., Lotz, J. C., Kapila, S, "Chondrogenic differentiation of human mesenchymal stem cells in three-dimensional alginate gels," *Tissue Engineering-Part A*, vol. 14, pp. 667-680, 5. 2008.
- [4] N. L. Nerurkar, D. M. Elliott and R. L. Mauck, "Mechanics of oriented electrospun nanofibrous scaffolds for annulus fibrosus tissue engineering," *Journal of Orthopaedic Research*, vol. 25, pp. 1018-1028, 2007.

COMPACT ATTITUDE SENSOR SYSTEM USING SR-UKF

NSF Summer Undergraduate Fellowship in Sensor Technologies
Christopher Baldassano (Electrical Engineering) – Princeton University
Advisor: Dr. Mark Yim

ABSTRACT

This paper describes the selection and integration of attitude sensors for a novel flying device. The overall goal of the experimental flyer project is to create a small, cheap, and lightweight device that is capable of controlled flight. This project focuses on the development of a state estimation system for the flyer that allows for precise and rapid attitude measurement. A sensor package consisting of a three-axis magnetometer, a three-axis accelerometer, and three one-axis rate gyroscopes is presented. The sensor measurements are filtered and fused together using a Square-Root Unscented Kalman Filter (SR-UKF) developed in MATLAB. The feasibility of porting the filter to a Microchip dsPIC30F microcontroller is discussed, and the performance of the filter is evaluated.

1. INTRODUCTION

The objective of the experimental flyer project is to develop a miniature, highly-maneuverable flying craft for indoor use. The main body of the flyer is cylindrical, with a length of about seven inches and a diameter of about three inches (see figure 1). Lift is generated by two propellers (one at each end of the cylinder) which spin in opposite directions. The flyer's novel torque generation scheme relies on a hinge in the center of the flyer, which allows the upper and bottom halves of the cylinder to pivot with respect to one another. The direction and magnitude of the torque can be easily calculated by conservation of angular momentum; the momenta of the propellers normally cancel since they are oriented in precisely opposite directions, but when the hinge pivots this cancelation is no longer exact, producing a compensating torque to maintain zero total angular momentum. The design of the experimental flyer is much simpler than that of traditional helicopters, which should allow it to be fabricated more cheaply and easily.



Figure 1: The Experimental Flyer

This paper describes the development of the attitude estimation sensor system for the flyer. The project consisted of the design, implementation, and testing of both the hardware and software elements of the system. The overall design goals of the system were:

- Size – must fit within the diameter of the flyer and be thin enough not to interfere with other components
- Weight – must not contribute significantly to the load on the flyer (which has a mass on the order of 300g)
- Speed – must be able to sense disturbances on the order of 100 Hz
- Precision and accuracy – must have sufficient resolution to allow for visibly stable flight
- Power – must require much less battery power than the motors (which consume power on the order of 50W)

Section 2 discusses the hardware and algorithms currently available for attitude estimation. Section 3 describes the selection and performance of the sensor hardware. Section 4 covers the implementation of the filtering algorithm and its feasibility for use on a microcontroller. Section 5 reviews the results of the system, and Section 6 offers suggestions for future development.

2. BACKGROUND

2.1 Attitude Sensors

The size, weight, and power requirements of the project necessitate the use of miniature state sensing devices. A variety of small sensors based on Micro-Electrical-Mechanical Systems (MEMS) are available for attitude and rate of rotation sensing. The most common are magnetometers, accelerometers, and gyroscopes.

2.1.1 Magnetometers

Miniature magnetometers can generate measurements by taking advantage of a number of different types of magnetic effects. Magnetostrictive magnetometers take advantage of the fact that certain ferromagnetic materials undergo an elongation or contraction along an axis parallel to a magnetic field and an opposite deformation along in the transverse direction, which can be sensed through the deflection of an attached cantilever [1]. Lorentz-force magnetometers measure the magnetic force on a current through a suspended aluminum bar, which causes the bar to vibrate in its fundamental mode [1]. Anisotropic magnetoresistive sensors utilize a thin film that changes resistance in the presence of a magnetic field [2]. Magnetoinductive magnetometers measure the inductance of a solenoid whose core changes permeability with a magnetic field [2].

2.1.2 Accelerometers

Micromachined accelerometers are produced in great volume for a wide variety of industrial and consumer applications. They are generally constructed of a suspension beam with an effective spring constant attached to a proof mass, whose position can be determined by measuring characteristics such as 1) Varying resistivity of silicon piezoresistors in the suspension beam as it is stressed; 2) Varying capacitance between the proof mass and an electrode as the mass moves; and 3) Varying quantum-mechanical tunneling current between the proof mass and an electrode as the mass moves [3]. Capacitive sensors are among the most popular, since they have high sensitivity and low power dissipation, and are offered with interface circuitry on a single integrated chip [3].

2.1.3 Gyroscopes

Most micromachined gyroscopes sense rotation by vibrating a mechanical element in one mode and then detecting the transfer of energy to other modes due to the Coriolis acceleration. For example, in a tuning-fork vibratory gyroscope, the tines are driven at a fixed amplitude, and the Coriolis force produces a bending of the tines or rotation around the stem of the tuning fork. Capacitive, piezoresistive, or piezoelectric sensors can be used to measure vibrations in these modes [3].

2.2 Sensor Fusion

Choosing the best attitude estimate for the flyer based on noisy sensor measurements is a nontrivial problem. One widely-used approach for modeling state dynamics and measurements is Kalman filtering.

2.2.1 Kalman Filters

Kalman filters are a method for estimating a dynamic state using indirect measurements [4]. At every discrete time step, two operations are performed: first, the current state of the system is predicted from the last state based on a process model, and then the state is corrected based on a measurement. Kalman filters explicitly take noise into account, and adjust the influence of various measurements based on their statistical properties. Both the mean vector and covariance matrix for the state are recorded at each step.

2.2.2 Euler-Rodrigues Symmetric Parameters

The state of the Kalman filter must be represented internally by some set of coordinates. One common parameterization of attitude is Euler-Rodrigues Symmetric Parameters, usually represented as a four-dimensional quaternion with a normalization constraint to give three degrees of freedom [5], [6]. Quaternions are a four-dimensional extension of complex numbers, and are used here to represent a scalar and a three-dimensional vector.

$$q = [q_0 \quad \rho] \quad \rho = [q_1 \quad q_2 \quad q_3] \quad (1),(2)$$

Quaternions have a number of advantages over traditional Euler rotation angles. The most important for this application is that they do not have singularities under translation (Euler angles are degenerate every 2π) and do not require the use of transcendental functions [7], which are very expensive on microprocessors; the cosine function, for example, can take up to 3200 cycles [8].

The time-evolution equation for quaternions, given a rotation vector $\omega(t)$, is given in [9]:

$$\dot{q} = \frac{1}{2} \Xi(q) * \omega = \frac{1}{2} \Omega(\omega) * q \quad (3)$$

where

$$\Xi(q) = \begin{bmatrix} -\rho^T \\ q_0 * I_{3 \times 3} + [\rho \times] \end{bmatrix} \quad \Omega(\omega) = \begin{bmatrix} 0 & -\omega^T \\ \omega & -[\omega \times] \end{bmatrix} \quad (4),(5)$$

and the cross-product matrix is given by

$$[a \times] = \begin{bmatrix} 0 & -a_3 & a_2 \\ a_3 & 0 & -a_1 \\ -a_2 & a_1 & 0 \end{bmatrix} \quad (6)$$

The measurement model for the magnetometer and accelerometer (when stationary) amounts to a rotation of a reference vector (magnetic or gravitational field direction) into the frame of the sensors, given in [10]:

$$x_{meas} = A(q) * x_{ref} \quad (7)$$

where the rotation matrix is defined by

$$A(q) = \Xi^T(q) * \Psi(q) \quad \Psi(q) = \begin{bmatrix} & -\rho^T \\ q_0 * I_{3 \times 3} & -[\rho \times] \end{bmatrix} \quad (8),(9)$$

2.2.3 SR-UKF

Since the classic Kalman filter assumes that the process and measurement equations are linear in the state variables, a generalization of the Kalman filter must be used that can account for the non-linear quaternion measurement model. One approach is to approximate the non-linearities to first order using an Extended Kalman Filter (EKF), but this approach has been mostly replaced by sigma-point methods such as the Unscented Kalman Filter (UKF) [11]. Sigma-point filters pass a set of points representing the input distribution through the non-linear functions, and then approximate the output statistics. The UKF is accurate to third order for any nonlinearity, but only requires computational resources on par with the EKF [12].

Figure 2 gives an example of poor modeling by an EKF. The mean and covariance calculated by the sigma-point filter (green) much more closely approximate the actual statistics (black) than those of the EKF (pink).

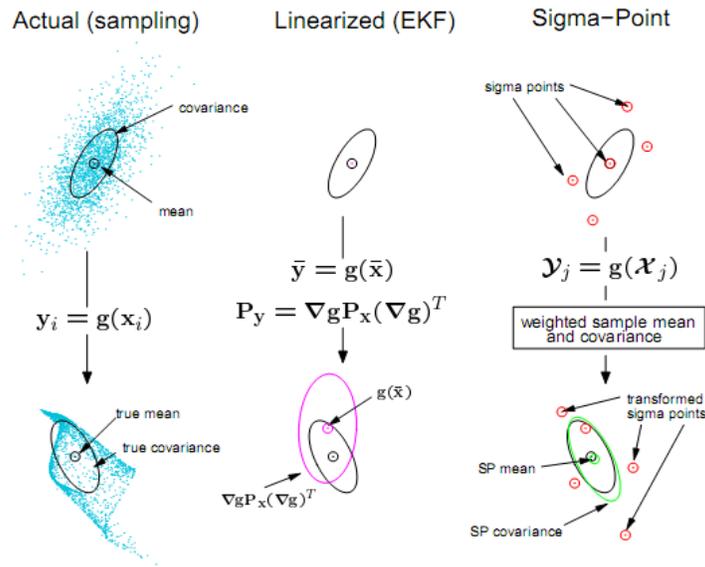


Figure 2: Comparison of EKF and UKF [12]

Due to numerical round-off errors, it is possible for the state covariance matrix to cease to be positive definite, causing the UKF algorithm to fail when taking a square root of the covariance to calculate the sigma points. To prevent this possibility, the square root of the covariance matrix can be calculated directly during every time step, without using the actual covariance matrix. The efficient implementation of van der Merwe and Wan makes use of Cholesky factor updating, allowing for better performance than even a normal UKF [11]. The equations from van der Merwe and Wan's paper are described below.

The filter is initialized with a mean vector and the square root of a covariance.

$$\hat{x}_0 = E[x_0] \quad S_0 = chol\{E[(x_0 - \hat{x}_0)(x_0 - \hat{x}_0)^T]\} \quad (10),(11)$$

The Cholesky factorization decomposes a symmetric, positive-definite matrix into the product of a lower-triangular matrix and its transpose. This triangular matrix is used directly to calculate the sigma points:

$$\chi_{k-1} = [\hat{x}_{k-1} \quad \hat{x}_{k-1} + \eta S_k \quad \hat{x}_{k-1} - \eta S_k] \quad (12)$$

The scaling constant η is calculated from

$$\eta = \sqrt{L\alpha^2} \quad (13)$$

where α is a tunable parameter less than one. The sigma points are then passed through the non-linear process model, which predicts the current attitude based on each sigma point and the last command from the flyer's control system, u_{k-1} .

$$\chi_{k|k-1} = F[\chi_{k-1}, u_{k-1}] \quad (14)$$

The estimated mean and square root covariance are calculated from the transformed sigma points using

$$\begin{aligned} \hat{x}_k^- &= \sum_{i=0}^{2L} W_i^{(m)} \chi_{i,k|k-1} & S_k^- &= qr\{[\sqrt{W_1^{(c)}} (\chi_{1:2L,k|k-1} - \hat{x}_k^-) \quad \sqrt{R^v}]\} \\ S_k^- &= cholupdate\{S_k^-, \chi_{0,k} - \hat{x}_k^-, W_0^{(c)}\} \end{aligned} \quad (15),(16),(17)$$

where

$$W_0^{(c)} = 2(1 - \alpha^2 + \frac{1}{2}\beta) \quad W_0^{(m)} = 1 - \alpha^2 \quad W_i^{(m)} = W_i^{(c)} = \frac{1}{2L\alpha^2} \quad (18),(19),(20)$$

The parameter α is the same as above, and β is another tunable parameter used to incorporate prior knowledge of the state distribution ($\beta=2$ is optimal for Gaussian distributions). The matrix R^v is the process noise covariance. The QR factorization decomposes a matrix into the product of an orthogonal matrix and a triangular matrix; only the triangular matrix is used here. Since the zero weight may be negative, the separate Cholesky update operation is needed; the Cholesky update operation efficiently transforms the Cholesky decomposition of the matrix A into the Cholesky decomposition of the matrix $A + x^T x$, where x is a row vector.

The transformed sigma points are then used to predict what measurements the sensors will make, using the nonlinear measurement model:

$$Y_{k|k-1} = H[\chi_{k|k-1}] \quad (21)$$

The expected measurement \hat{y}_k^- and square root covariance of $\tilde{y}_k = y_k - \hat{y}_k^-$ (the difference between the actual and expected measurements, also called the innovation) are given by the unscented transform equations just as for the process model:

$$\hat{y}_k^- = \sum_{i=0}^{2L} W_i^{(m)} Y_{i,k|k-1} \quad S_{\tilde{y}_k} = qr\{[\sqrt{W_1^{(c)}} [Y_{1:2L,k} - \hat{y}_k] \quad \sqrt{R_k^n}]\}$$

$$S_{\tilde{y}_k} = cholupdate\{S_{\tilde{y}_k}, Y_{0,k} - \hat{y}_k, W_0^{(c)}\} \quad (22),(23),(24)$$

In order to determine how much to adjust the predicted mean and covariance based on the actual sensor input, the Kalman gain matrix K_k is calculated:

$$P_{x_k y_k} = \sum_{i=0}^{2L} W_i^{(c)} [\chi_{i,k|k-1} - \hat{x}_k^-][Y_{i,k|k-1} - \hat{y}_k^-]^T \quad (25)$$

$$K_k = (P_{x_k y_k} / S_{\tilde{y}_k}^T) / S_{\tilde{y}_k} \quad (26)$$

Note that $S_{\tilde{y}_k}$ is square and triangular, so efficient back-substitutions can be used rather than a costly matrix inversion. Finally, the state mean and covariance are updated using the actual sensor input and the Kalman gain matrix:

$$\hat{x}_k = \hat{x}_k^- + K_k (y_k - \hat{y}_k^-) \quad (27)$$

$$U = K_k S_{\tilde{y}_k} \quad S_k = cholupdate\{S_k^-, U, -1\} \quad (28),(29)$$

3. ATTITUDE SENSOR SELECTION AND INTEGRATION

In this project, two categories of sensors have been selected. A magnetometer and accelerometer are used to give an absolute orientation estimate by measuring the direction of the magnetic field and gravitational field vectors, and three one-axis gyroscopes are used to estimate relative changes in orientation between updates.

Sensor Type	Sensor Name	Voltage (V)	Typical Current (mA)	Units/LSB	Interface
Magnetometer	MicroMag3	3 or 5	<.5	.032 μ T	SPI
Accelerometer	SCA3000-E04	3 \pm .6	.12	2 mg	SPI
Gyroscope	MLX90609	5 \pm .25	16	.31 $^\circ$ /sec	SPI

Table 1: Sensor specifications

3.1 Magnetometer

PNI Corporation's MicroMag3 magnetometer uses three magneto-inductive sensors, one oriented along each axis [14]. It uses an SPI interface, draws very little current, and has a high maximum resolution.

Communicating with the MicroMag3 requires six wires: the SS line, which must be pulled low to enable SPI communication; the RESET line, which must be pulsed high at the start of each measurement; the MOSI line, which receives an SPI command; the DRDY line, which is raised high by the MicroMag3 after a successful measurement; the MISO line, which outputs the measurement back to the microcontroller; and the SCLK line, which provides timing information for the MOSI and MISO lines [14]. The protocol is summarized in Figure 3.

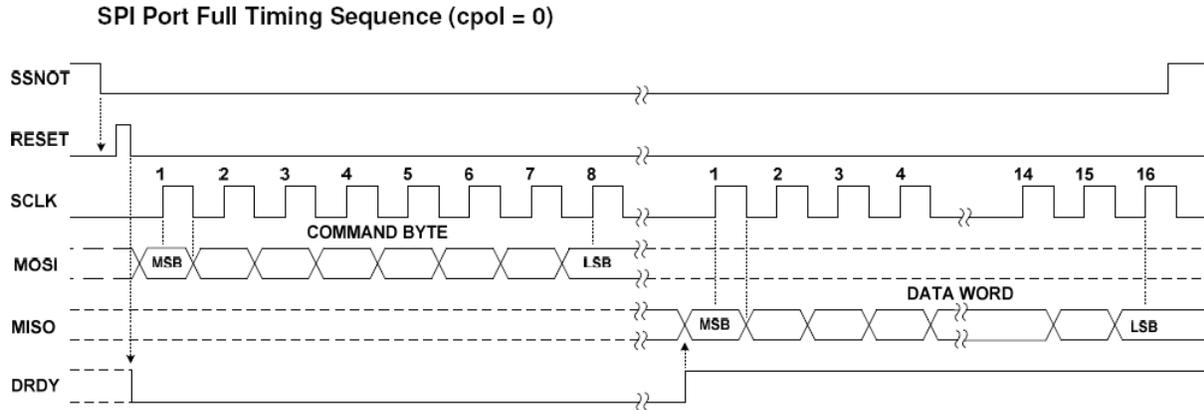


Figure 3: MicroMag3 SPI protocol [13]

The command sent to the MicroMag3 specifies both the axis to measure (X, Y, or Z) and the period division ratio. Magneto-inductive sensors measure the frequency of an LR relaxation oscillator whose inductance depends on the external magnetic field, taking advantage of the nonlinear permeability $\mu(H)$ of the core material [14]. The period division ratio determines how long the oscillator is run; measuring more cycles gives a more accurate field measurement, but takes a longer amount of time. A ratio of 1024 was selected, which guarantees that each axis can be measured in less than 15 ms. For non-zero magnetic fields, the actual delay is measured to be approximately 10 ms.

3.2 Accelerometer

VTI Technologies' SCA3000-E04 uses a capacitive sensor to measure the deflection of a proof mass due to acceleration [15]. It features an SPI interface, low power consumption, and high precision of 2 mg steps between $\pm 6g$.

The SPI interface uses five wires: the XRESET line, which must be raised high at power-on (before the first measurement only); the CSB line, which must be pulled low to enable SPI communication; and MOSI, MISO, and SCLK lines as in section 3.1 [16].

3.3 Gyroscopes

Melexis's MLX90609 uses a capacitive sensor to detect secondary mode oscillations due to the Coriolis force from rotation [17]. The R2 version has a measurement range of ± 300 °/sec in increments of 0.3 °/sec, accessible simultaneously over both SPI and an analog voltage output. The SPI interface is the simplest of the sensors, with only four lines: SS, MOSI, MISO, and SCLK as in section 3.1 [18]. Since each gyroscope only measures rotation around one axis, however, three separate MLX90609 sensors were needed. Also, an interface circuit had to be constructed to allow these 5V sensors to share an SPI bus with the 3.3V sensors (see section 3.5).

3.4 SPI Time Division

In order to achieve the design goal of speeds on the order of 100 Hz, measurements and computation had to be appropriately arranged in time. Since each axis of the magnetometer takes approximately 10 ms to measure, only one axis was measured per SR-UKF cycle. The actual filter processing takes place while waiting for each magnetometer measurement to complete. Figure 4 shows the SPI inputs and outputs as measured by an oscilloscope.

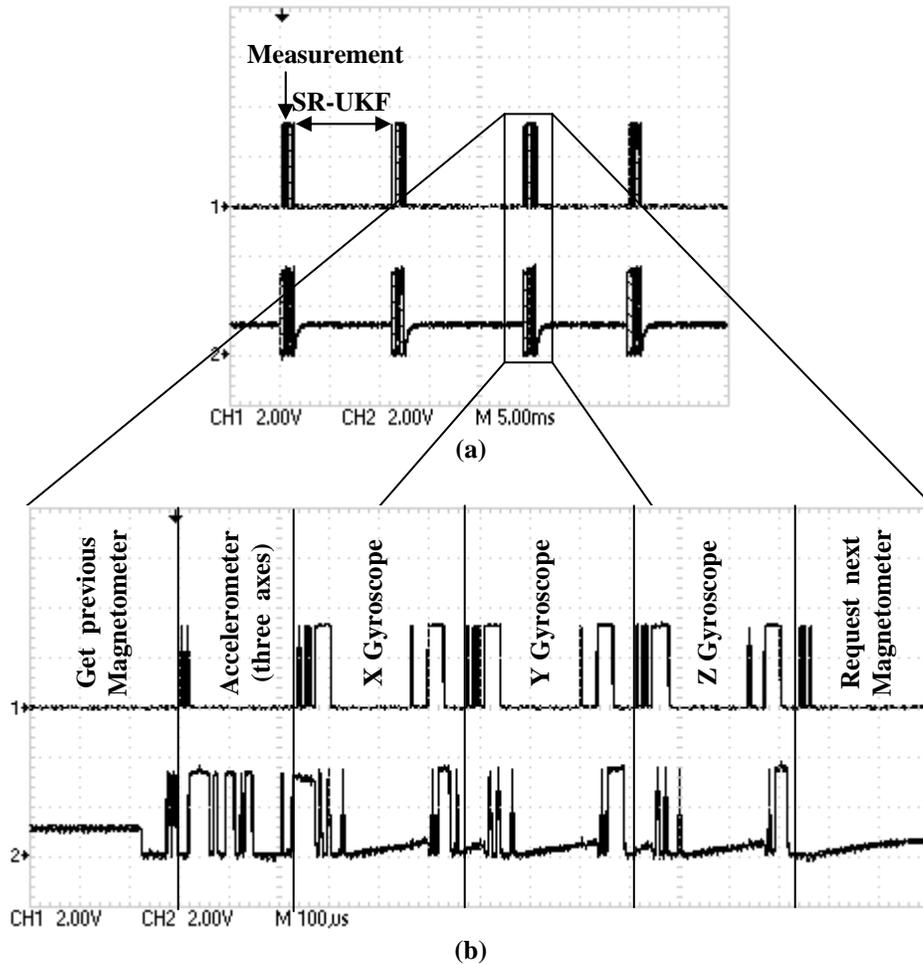


Figure 4: SPI MOSI (master output, top) and MISO (master input, bottom) during filter operation

The final circuit is shown in Figure 5. The SPI lines for each sensor are controlled by the dedicated SPI ports on the dsPIC, and the other lines (SS, XRESET, etc.) are controlled by the general I/O pins. Some noteworthy elements of the circuit:

- The MLX90609 gyroscopes run on 5V, rather than 3.3V like the other sensors. The outputs from the microcontroller to the gyroscopes – the three SS lines and the MOSI line – are stepped up using XOR gates with one input grounded. These act level shifters, recognizing 3.3V as a logical high and thus mapping 0V → 0V and 3.3V → 5V. No level shifting is required for the SCLK line, since the gyroscopes’ logical high threshold for the clock line is only 2.8V (rather than 3.5V as on the other lines).
- To prevent the gyroscopes from driving the output line (MISO) to 5V, potentially damaging the 3.3V sensors and microcontroller, a diode to 3.3V was added such that the excess voltage is dropped across a 2K resistor before reaching the 3.3V components.
- A universal asynchronous receiver/transmitter (UART) interface allows the microcontroller to communicate with a PC over RS-232 using a serial cable. The RS-232 standard defines voltages between -3V and -25V as logical high (“marking”) and voltages between 3V and 25V as logical low (“spacing”) [18]. Therefore, an additional chip (the MAX232ACSE) must be used to convert the 0-3.3V signal from the dsPIC’s UART port into an RS-232-compatible signal; the MAX232 is able to generate voltages of ±10V with only a +5V source line, using dual charge-pump DC-DC converters [19]. Once the cable load for this application is attached, the actual signals transmitted are measured to be about ±7V (well within the specification). Although this is a 5V chip, it recognizes signals greater than 2V as logical high so no interface circuitry is needed for the transmission line. A diode/resistor pair is added to the receiver line as on the gyroscopes’ MISO line to prevent damage to the dsPIC.

A photograph of the final circuit is shown in Figure 6.

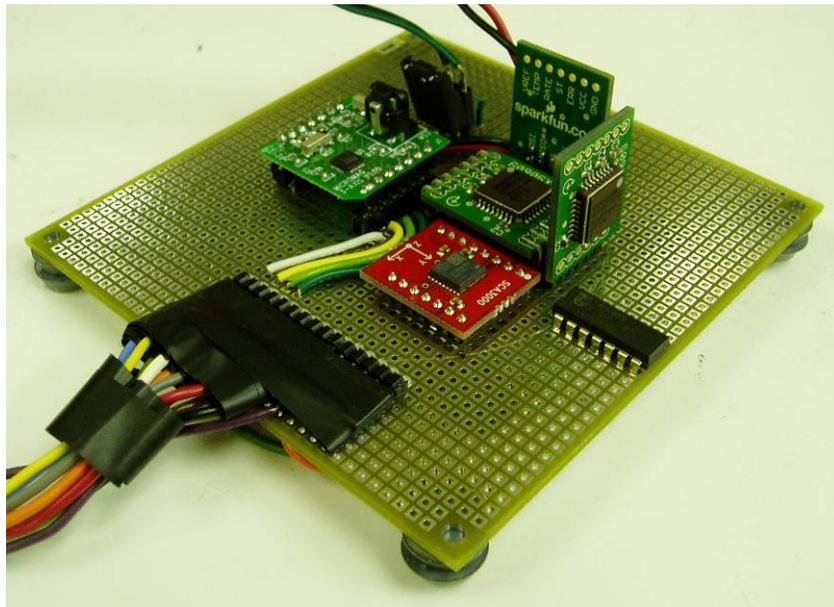


Figure 6: Photograph of Sensor Board

4. SRUKF IMPLEMENTATION

4.1 MATLAB Prototyping and GUI

In order to develop and test the SR-UKF, the filter was implemented in MATLAB. Sensor data were streamed from the dsPIC to the PC over a RS-232 serial cable, and the processing was performed in real time on the PC.

A number of visualizations were developed to aid in the debugging process. Figure 7 shows the four most commonly used windows. Three of the windows show the data being output by the dsPIC compared to the result of the measurement prediction \hat{y}_k^- in the SR-UKF (see section 5.2 for analysis). The lower-right window shows a 3D view of the current attitude estimate, in which the solid blue, green, and red lines show the current orientations of the X, Y, and Z axes (respectively) and the dotted lines show the reference (unrotated) orientation.

In addition to the real-time GUI, the MATLAB program logs the input data, covariance matrices, Kalman gain matrices, and filter output for each step, allowing for offline analysis. See Appendix B for a complete listing of the MATLAB code, with annotations.

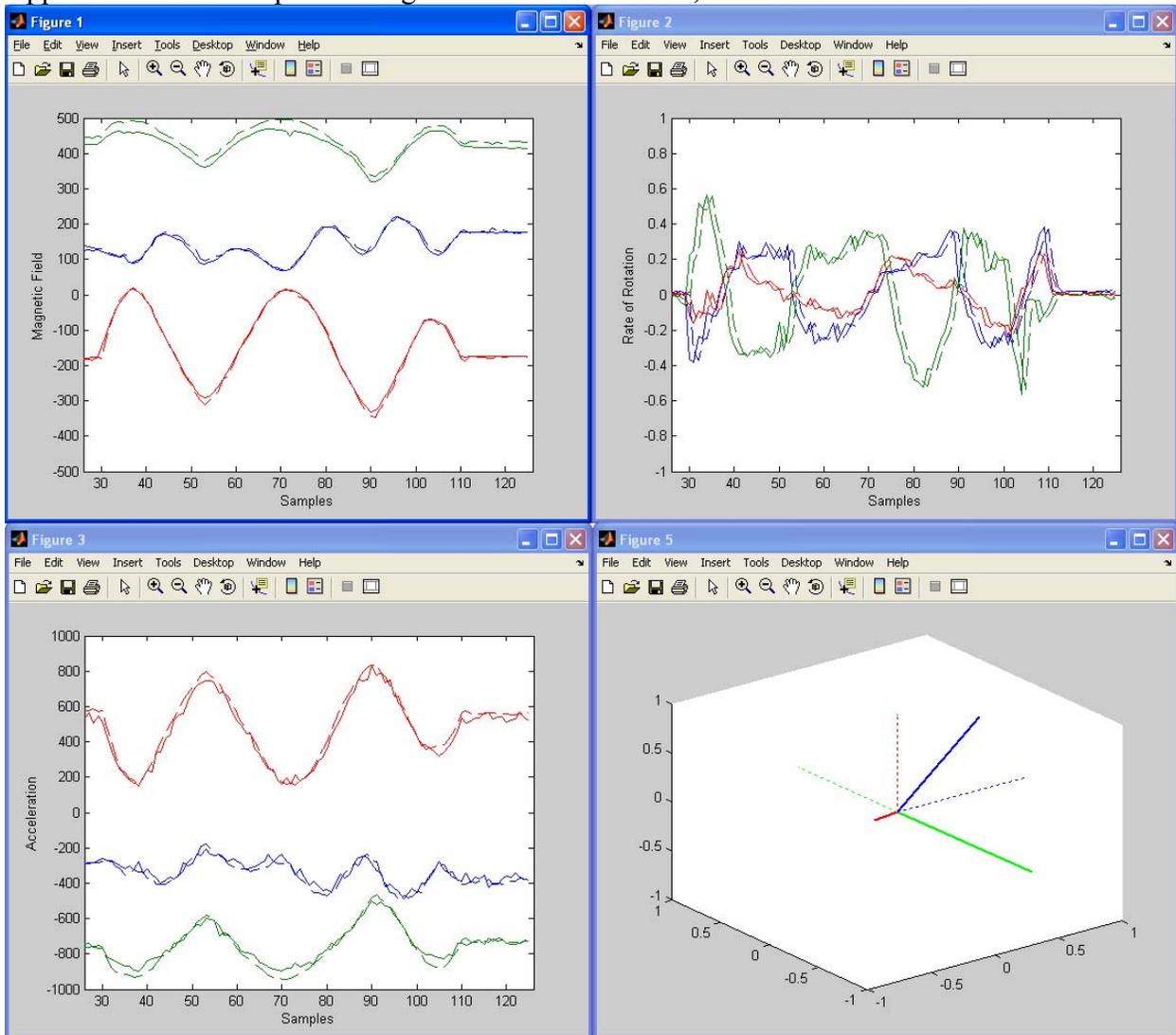


Figure 7: MATLAB GUI for SR-UKF Testing

4.2 dsPIC Implementation

An attempt to implement the filter on the dsPIC30F4012 was made using the Embedded MATLAB Coder (EMLC) from the MATLAB Real-Time Workshop. The EMLC allows for the automatic translation of MATLAB files (meeting certain prerequisites) into C files suitable for execution on an embedded system. In addition to a C file containing the converted MATLAB code and C versions of all necessary matrix functions, the EMLC generates header files that provide wrappers for the specific data types available on the dsPIC.

Although the SR-UKF code was converted correctly, it failed immediately upon execution due to a stack overflow. The available RAM for the dsPIC30F4012 is only 2KB (data space 0x0800 to data space 0x1000) [20]. The required matrices for the SR-UKF alone, however, consume over 3KB of memory (see Table 2) since every floating-point number uses 4 bytes.

Dynamic allocation (not implemented by the EMLC) would allow for a reduction in the amount of memory required, since not all the matrices are required simultaneously. Even with this approach, however, the memory consumed by the required matrices still exceeds 2KB, as shown in Table 3; this peak memory usage occurs just before the calculation of the Kalman gain matrix K . Note that these calculations are underestimates, since they do not take into account stack overhead from function calls or temporary variables.

Matrix Type	Bytes per Matrix	Total bytes
1x7: xMeanUT, yMeanUT, state, data	28	112
15x1: Wm, Wc	60	120
7x7: S, Sy, Q, R, Pxy, K, U	196	1372
15x7: xSigmaPts, xSigmaPtsUT, ySigmaPtsUT	420	1260
21x7: Input to QR function	588	588
	Total Memory Required:	3452

Table 2: SR-UKF Memory Requirements without dynamic allocation

Matrix Type	Bytes per Matrix	Total bytes
1x7: xMeanUT, yMeanUT, state, data	28	112
15x1: Wm, Wc	60	120
7x7: S, Sy, Q, R, Pxy	196	980
15x7: xSigmaPtsUT, ySigmaPtsUT	420	840
	Total Memory Required:	2052

Table 3: SR-UKF Memory Requirements with Dynamic Allocation

Since the full SR-UKF cannot be run on this dsPIC, a possible compromise is to run a simplified version of the filter. MATLAB tests were performed to determine the feasibility of running the SR-UKF using a constant Kalman gain matrix K and a constant square-root covariance S . Since most of the time- and memory-consuming operations of the filter involve updating these matrices, the simplified filter would be more appropriate for the dsPIC. This approach, however, led to instabilities in the filter and was not implemented. Figure 8 shows an example of these instabilities, in the predicted x measurement of the magnetic field (solid line).

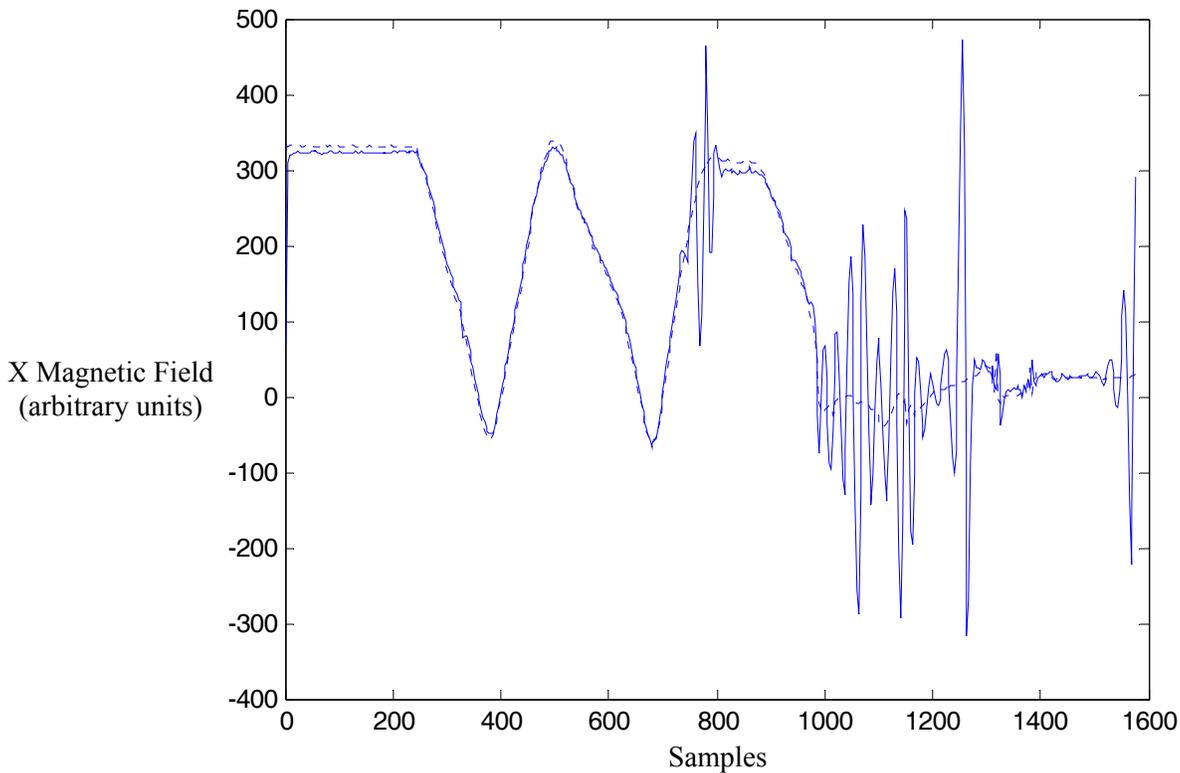


Figure 8: Actual (solid) vs. Predicted (dotted) Measurements for the Constant-Gain SR-UKF

5. RESULTS AND DISCUSSION

5.1 Low-Speed Sensor Tests

The following graphs show typical low-speed sensor measurements taken simultaneously. The sensor board was powered on with the y axis pointing straight up. A series of rotations were performed on the x-axis, then rotations were performed on the z-axis, and finally (after one-half x-axis rotation to reset to the initial position) rotations were performed on the y-axis.

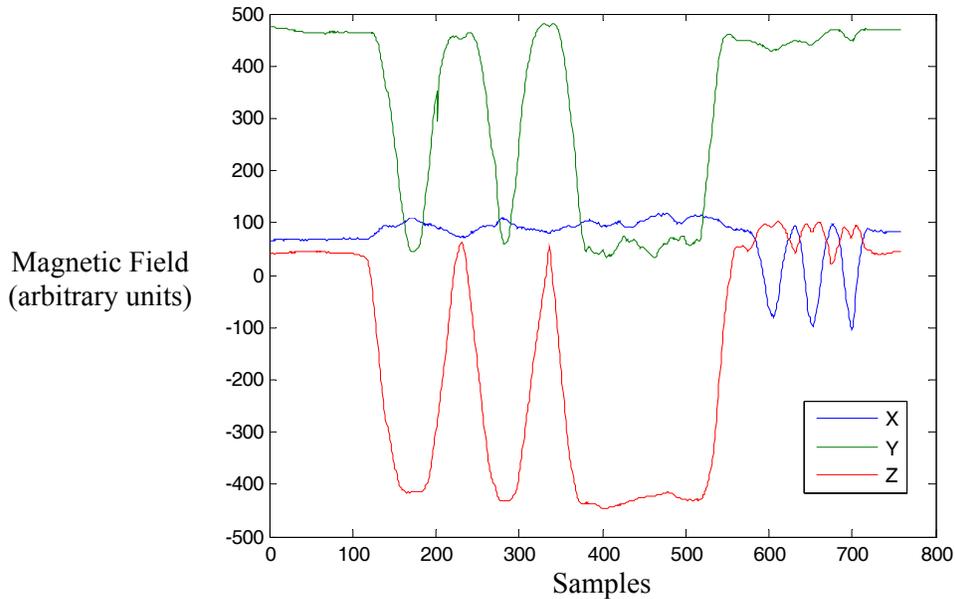


Figure 9: Example MicroMag3 Measurement

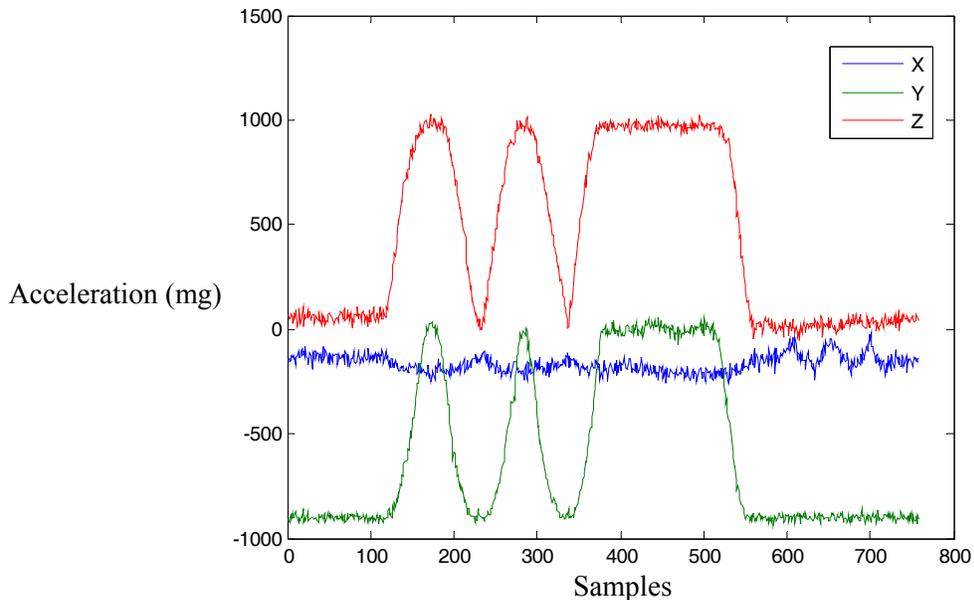


Figure 10: Example SCA3000 Measurement

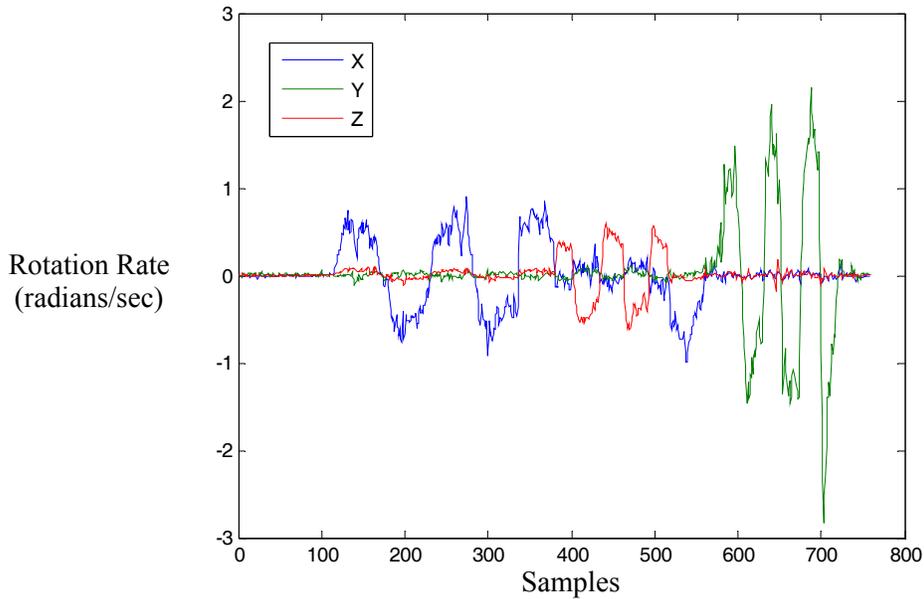


Figure 11: Example MLX90609 Measurement

Although the MicroMag3 takes longer to make a measurement compared to the SCA3000 accelerometer, these data reveal that it has a much higher signal-to-noise ratio. It does have very infrequent spikes (as can be seen on the Y-axis at around sample 200) but these can be easily filtered out.

The variation of the gyroscope measurements from pure sinusoids is not due to measurement errors by the MLX90609, but reflects physical variation in rotation speed (since the test rig was not powered, rotations had to be induced by hand).

5.2 MATLAB Tests

Comparing the actual sensor data and the result \hat{y}_k^- of the measurement prediction step in the SR-UKF allows for verification of the filter in two ways. Showing that the estimated state can generate measurement predictions that closely match all of the sensor data *simultaneously* provides strong evidence that the model parameterization and measurement function are physically realistic. An incorrect measurement function for an SR-UKF with a single sensor might not be detected, since it still could be possible for the filter to match predicted and actual measurements by choosing a different (erroneous) state. As shown in Figure 12 and Figure 13, however, the MATLAB SR-UKF is able to choose a state that matches all six magnetic and acceleration sensors, verifying the correctness of the model (predicted measurements are indicated by solid lines, while actual measurements are shown as dots). See section 5.3 for quantitative analysis of the prediction matching.

Additionally, the time-dependent behavior of the prediction validates the time-update process model. If the SR-UKF were not modeling and correctly incorporating the rotation rate, the predicted measurements would be seen to lag behind the actual measurements since any changes would not be anticipated appropriately. See section 5.3 for quantitative analysis of the prediction delay.

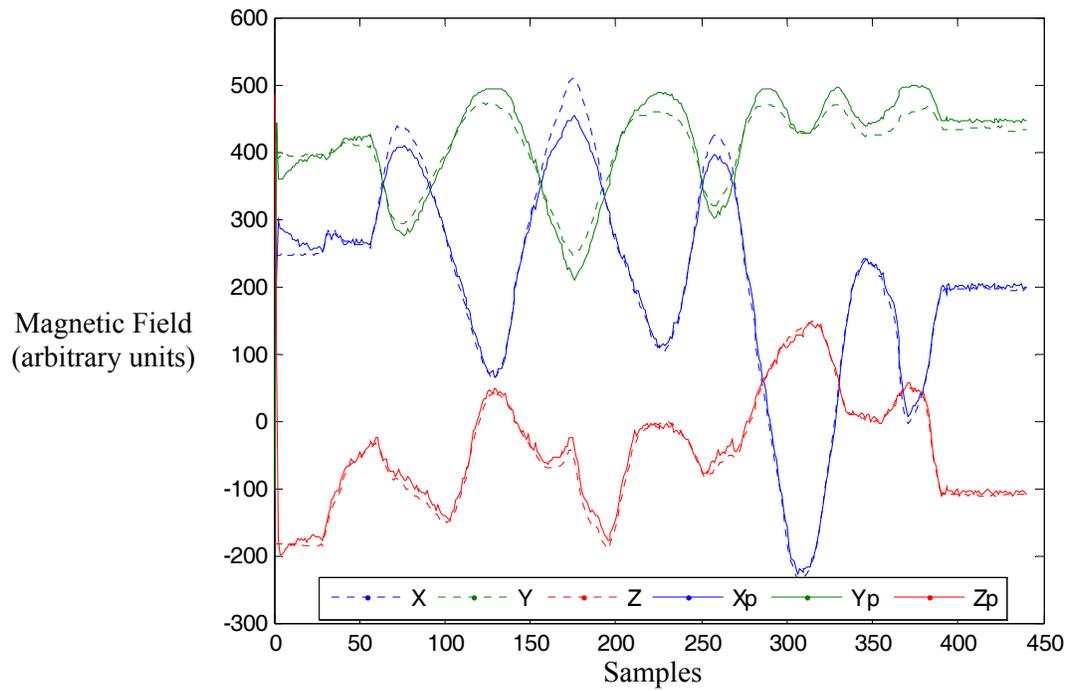


Figure 12: Actual vs. Predicted Measurements for MicroMag3

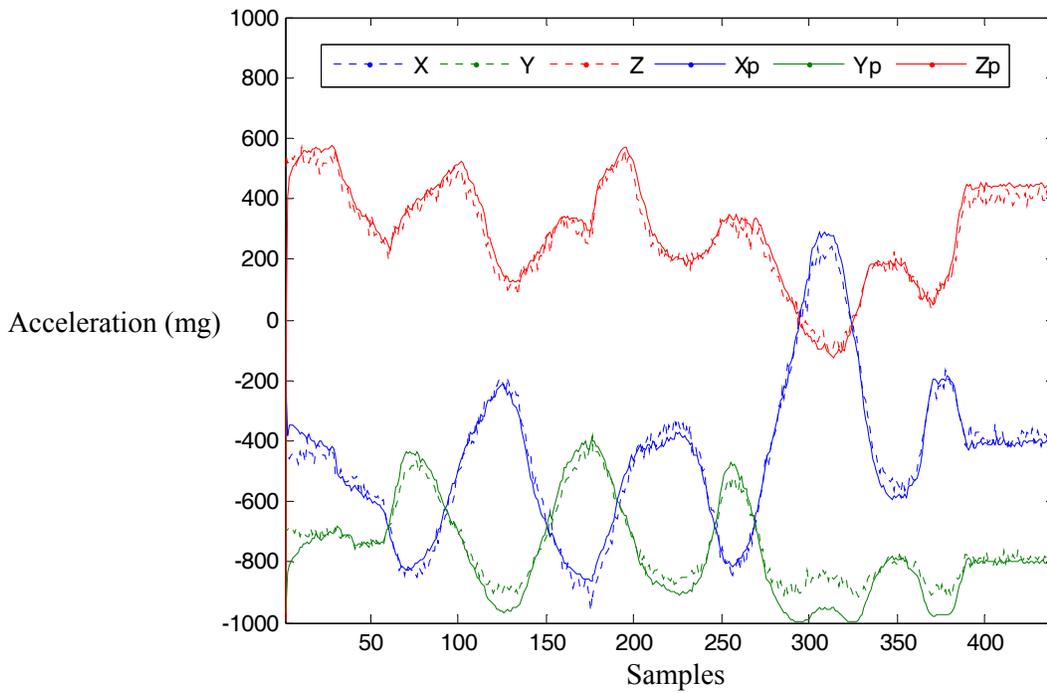


Figure 13: Actual vs. Predicted Measurements for SCA3000

5.3 High-Speed MATLAB Tests

As described in section 3.4, the measurement system was designed to read seven measurements – three gyroscope axes, three accelerometer axes, and one magnetometer axis – at approximately 100Hz. Running at this measurement rate, however, caused the noise on the SCA3000 accelerometer to increase dramatically. The characteristics of this transition were investigated by inserting a delay of varying length before requesting a sample from the accelerometer and then measuring the noise in all three accelerometer axes. The results are shown in Figure 14. Noise was quantified by calculating the root-mean-square error over a window of 50 samples between the measurements and their average values over the window. Note that the noise level drops off sharply at a delay of about 5 ms. Since this is unacceptably long, attempts were made to only query the axes on a rotating basis (much like the magnetometer), but the noise level did not appreciably decrease unless the SPI bus was completely silent for about 5 ms before requesting a sample. The data sheet claims that update rates of up to 93 Hz are possible, but does not indicate whether or not this specification still holds when the SCA3000 is sharing an SPI bus with other sensors [21]. Since this issue could not be resolved, and the noisy accelerometer measurements are not useful for attitude estimation, the high-speed MATLAB tests were conducted using only the magnetometer and gyroscopes.

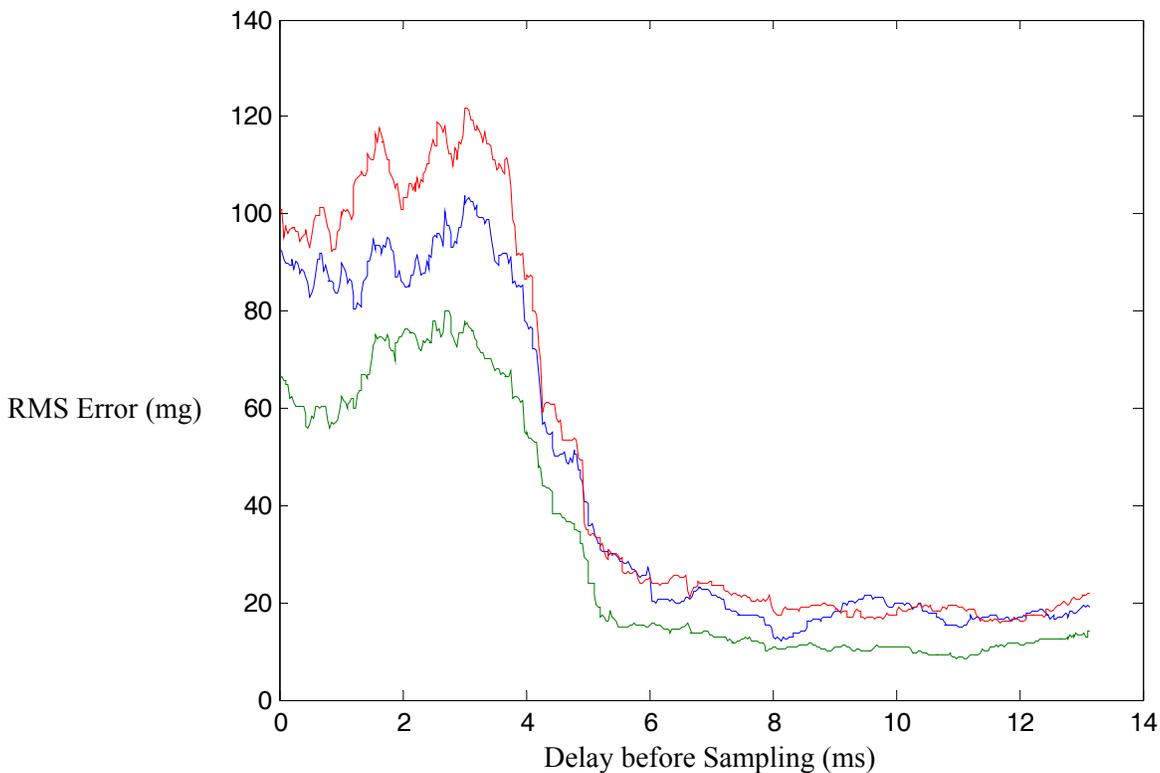


Figure 14: Noise level of SCA3000 Accelerometer vs. Delay before measurement

The MATLAB SR-UKF was able to run in real-time as the data was streamed from the dsPIC at speeds of approximately 90Hz, as long as the GUI was not displayed. Use of the MATLAB profiler showed that the longest single steps in the filter cycle were the time and measurement prediction functions, which involve repeated matrix multiplications.

The match between actual and predicted magnetometer measurements (discussed in section 5.2) was investigated analytically for the high-speed filter. The root-mean-square error between the two plots was less than 16.7 units in each axis, or less than 3.4% of the full magnetic magnitude of 500 units.

To demonstrate that there was no lag between the two plots, the RMS error was evaluated for various horizontal translations of the predicted measurements. Figure 15 shows that no translation improves the match between the actual and predicted measurements, so the estimated attitude does not lag behind the measured attitude. In contrast, Figure 16 shows that there is a definite lag of one cycle for the gyroscope measurements, since changes in rotational velocity are not being modeled. The difference between these plots demonstrates the benefit of explicitly modeling the rotation rates in the attitude state.

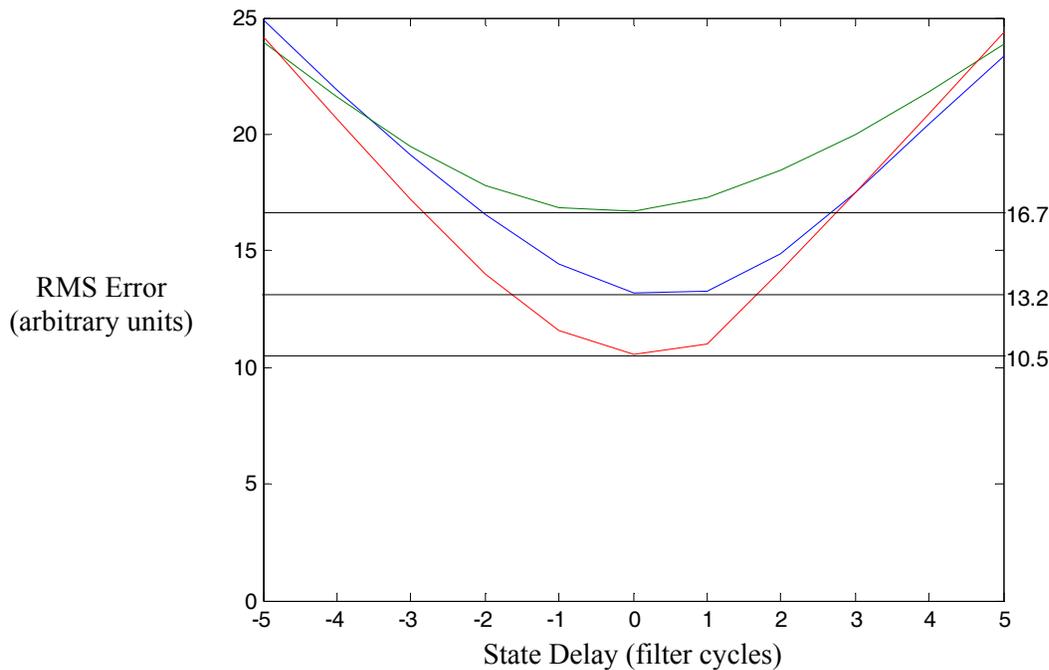


Figure 15: RMS Error between Actual and Predicted Magnetic Measurements vs. Lag Times

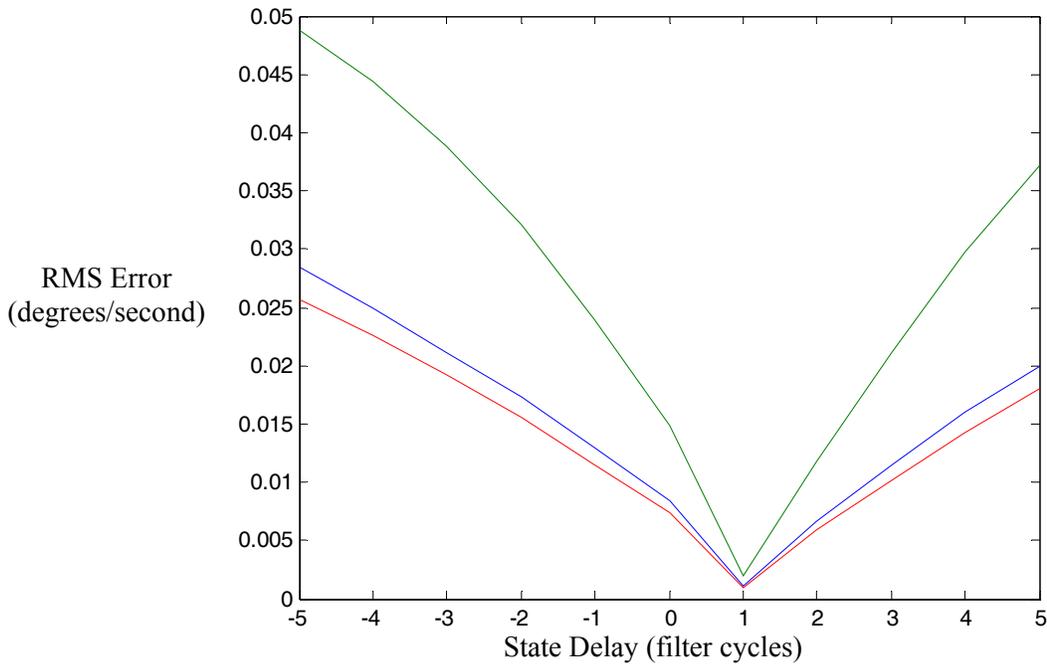


Figure 16: RMS Error between Actual and Predicted Gyroscope Measurements vs. Lag Times

6. FUTURE WORK

A dsPIC-based sensor board and MATLAB implementation of an SR-UKF were successfully completed, but several questions about the sensor system remain unanswered:

- Is it possible to run the filter on a more powerful microcontroller than the dsPIC30F4012? The space requirements for the filter are not unreasonable, but the processor speed necessary to complete a SR-UKF cycle in less than 10 ms has not yet been evaluated. A possible replacement microcontroller is the dsPIC30F6014A, which offers 8KB of RAM and has a similar architecture.
- Should an accelerometer be included in the final design? The SCA3000 can be exchanged with an accelerometer that does not lose reliability at high polling rates, but the usefulness of an acceleration measurement in the highly dynamic environment of the flyer has not been proven.
- How will the magnetometer measurements be affected by the flyer motors? The filter currently relies heavily on the magnetometer for absolute position measurements, but the introduction of magnetic devices nearby may impair its measurement ability. Sensor placement and electromagnetic shielding should be investigated to improve the signal-to-noise ratio.
- Can the performance of the SR-UKF be improved further by incorporating the flyer's control outputs into the process model? This would allow modeling of the rotation rates.

7. ACKNOWLEDGMENTS

I would like to thank my advisor, Dr. Mark Yim, for providing me with a challenging project and the resources I needed to complete it. I thank Bill Mather for acting as a sounding board for all of the problems I encountered and for always being available to offer advice. I am grateful to the National Science Foundation and Dr. Jan Van der Spiegel for making SUNFEST possible and giving me the opportunity to be involved in advanced research.

8. REFERENCES

- [1] D. K. Wickenden, T. J. Kristenmacher, R. Osiander, S. A. Ecelberger, R.B. Givens, J C. Murphy. (1997). Development of Miniature Magnetometers. *John Hopkins APL Technical Digest* [Online]. 18(2), 271-278. Available: <http://www.jhuapl.edu/techdigest/td1802/wickend.pdf>
- [2] M. J. Caruso, T. Bratland, C. H. Smith, R. Schneider. (1998, Dec.). A New Perspective on Magnetic Field Sensing. *Sensors* [Online]. Available: <http://www.sensorsmag.com/articles/1298/mag1298/main.shtml>
- [3] N. Yazdi, F. Ayazi, K. Najafi. (1998, Aug.). Micromachined Inertial Sensors. *Proceedings of the IEEE* [Online]. 86(8), 1640-1659. Available: <http://dx.doi.org/10.1109/5.704269>
- [4] E. J. Lefferts, F. L. Markley, M.D. Shuster. (1982, Jan.). Kalman Filtering for Spacecraft Attitude Estimation. *AIAA 20th Aerospace Sciences Meeting* [Online]. Available: <http://www.aiaa.org/content.cfm?pageid=406&gTable=mtgpaper&gID=90156>
- [5] M. D. Shuster. (1993, Oct.-Dec.) A Survey of Attitude Representations. *Journal of Astronautical Sciences* [Online]. 41(4), 439-517. Available: http://home.comcast.net/~mdshuster/Pub_1993h_J_Repsurv_scan.pdf
- [6] S. Hsieh, L. K. Wang, F. Hsaio, K. Huang, F. Tsai. (2004, March). Airborne Attitude/Ground Target Location Determinations Using Unscented Kalman Filter. *2004 IEEE Aerospace Conference Proceedings* [Online]. 3, 1561-1568. Available: <http://dx.doi.org/10.1109/AERO.2004.1367931>
- [7] J. L. Crassidis, F. L. Markley. (1995). Attitude Estimation Using Modified Rodrigues Parameters. *Proceedings of the Flight Mechanics/Estimation Theory Symposium* [Online]. 419-433. Available: http://ntrs.nasa.gov/archive/nasa/casi.ntrs.nasa.gov/19960035754_1996055864.pdf
- [8] Microchip Technology Inc. PIC24 MCU / dsPIC DSC Math Library [Online]. Available: http://www.microchip.com/stellent/idcplg?IdcService=SS_GET_PAGE&nodeId=1406&DocName=en022432
- [9] J. L. Crassidis, F. L. Markley, Y. Cheng, "Survey of Nonlinear Attitude Estimation Methods," *Journal of Guidance, Control, and Dynamics*, vol. 30, no. 1, pp. 12-28, Jan.-Feb. 2007.
- [10] J. L. Crassidis, F. L. Markley, "Unscented Filtering for Spacecraft Attitude Estimation," *Journal of Guidance, Control, and Dynamics*, vol. 26, no.4, pp. 536-542, July-Aug. 2003.
- [11] R. van der Merwe and E. A. Wan. (2001, May). The Square-Root Unscented Kalman Filter for State and Parameter-Estimation. *International Conference on Acoustics, Speech, and Signal Processing* [Online]. Available: <http://speech.bme.ogi.edu/publications/ps/merwe01a.pdf>
- [12] R. van der Merwe and E. Wan. (2004, June). Sigma-Point Kalman Filters for Integrated Navigation. *Proceedings of the 60th Annual Meeting of The Institute of Navigation (ION)* [Online]. Available: <http://speech.bme.ogi.edu/publications/ps/merwe04a.pdf>
- [13] PNI Corporation. MicroMag3 Data Sheet [Online]. Available: <http://www.pnicorp.com/productDetail?nodeId=cMM3>
- [14] Zero-offset magnetometer having coil and core sensor controlling period of an oscillator circuit, by T. Hawks. (August 24, 1993). *Patent 5 239 264* [Online]. Available: <http://patft.uspto.gov/>

-
- [15] VTI Technologies. Why 3D Mems [Online]. Available:
<http://www.vti.fi/en/products/technology/3d-mems/why-3d-mems/>
- [16] VTI Technologies. SCA3000 Accelerometer Product Family Specification [Online]. Available: http://www.vti.fi/en/products/accelerometers/consumer_electronics/
- [17] Melexis Microelectronic Systems. MLX90609 DataSheet [Online]. Available:
http://www.melexis.com/Sensor_ICs_Inertia/General/MLX90609_582.aspx
- [18] C. E. Strangio. (2006). The RS232 Standard [Online]. Available:
http://www.camiresearch.com/Data_Com_Basics/RS232_standard.html
- [19] Maxim Integrated Products. +5V Powered, Multichannel RS-232 Drivers/Receivers [Online]. Available: http://www.maxim-ic.com/quick_view2.cfm/qv_pk/1798
- [20] Microchip Technology Inc. dsPIC30F4011/12 Data Sheet [Online]. Available:
http://www.microchip.com/stellent/idcplg?IdcService=SS_GET_PAGE&nodeId=1335&dDocName=en010338
- [21] VTI Technologies. SCA3000-E04 Data Sheet [Online]. Available:
http://www.vti.fi/en/products/accelerometers/consumer_electronics/

Appendix A: dsPIC Source Code

```
#include "p30f4012.h"
/*****
SPI Interface for Sensor Board
Author: Chris Baldassano, chrisb@princeton.edu
Language Toolsuite: Microchip C30
Date: August 2008
Details: Reads data from MicroMag3 magnetometer, 3 MLX90609 gyroscopes,
        and SCA3000 accelerometer and outputs the results over UART.
        For use with MATLAB SR-UKF Filter
Config. bits: Primary Oscillator, FRC w/ PLL 16x

Port map:
    RB0 -> Accelerometer XReset
    RD0 -> Magnetometer SS
    RE0 -> Accelerometer SS
    RE1 -> Z Gyroscope SS
    RE2 -> Y Gyroscope SS
    RE3 -> X Gyroscope SS
    RE4 -> Magnetometer Reset
    RE5 -> Magnetometer DRDY
*****/

// Send a 16-bit integer over UART
void SendIntUART(int toSend);

int main()
{
    int magAxes[3];
    int gyroAxes[3];
    int accelAxes[3];
    int dummy;
    int i;
    long counter;
    char j;
    char magAxis;
    int sign;

    TRISE = 0b000100000; //Set RE5 as input, all others as output
    PORTE = 0b000001111; //Set RE0-3 high

    TRISB = 0b0000000; //Set all as output
    PORTB = 0b0000000; //Set all low

    TRISD = 0b00; //Set all as output
    PORTDbits.RD0 = 1; //Set RD0 high

    //Nominally:
        //Int. Osc = 7.37 MHz
        //Fcy = 7.37 * 16 PLL / 4 = 29.48 MHz
    //Measured: Fcy = 31.3344 MHz

    /* UART Setup */
    //BRG = Fcy / (16*Baud) - 1
    U1BRG = 33; //57600 Baud
```

```

U1MODEbits.PDSEL = 0b00;    //8-bit data, no parity bit
U1MODEbits.STSEL = 0b0;     //1 stop bit
U1MODEbits.ALTIO = 0b1;    //Use alternate ports (RC13-14)
U1MODEbits.UARTEN = 0b1;   //Enable UART
U1STAbits.UTXEN = 0b1;     //Enable transmission

/* SPI Setup */
SPI1CONbits.CKE = 0b1;     //Change MOSI on falling CLK edge
SPI1CONbits.MSTEN = 0b1;   //Master mode
//SPICLK = Fcy / (Primary*Secondary)
SPI1CONbits.SPRE = 0b000;  //Secondary = 8
SPI1CONbits.PPRE = 0b01;   //Primary = 16
SPI1CONbits.MODE16 = 0b0;  //8-bit mode
SPI1STATbits.SPIROV = 0b0; //Reset overflow flag
SPI1STATbits.SPIEN = 0b1;  //Enable SPI

PORTBbits.RB0 = 1;        //Start up accelerometer
// Loops measured to take .25 uS per count
for (counter = 0; counter < 35*4000; counter++); //wait 35ms

//Wait for UART character to begin loop
while (!U1STAbits.URXDA);
dummy = U1RXREG;

//Request first magnetometer reading
PORTDbits.RD0 = 0;        //Magnetometer SS
PORTEbits.RE4 = 1;       //Pulse Reset
for(i=0;i<20;i++);       //5 uS pause
PORTEbits.RE4 = 0;       //Clear Reset
for(i=0;i<20;i++);       //5 uS pause

//Request x measurement
SPI1BUF = 0b01010001;
while (!SPI1STATbits.SPIRBF);
dummy = SPI1BUF;
PORTDbits.RD0 = 1;

//Only one magnetometer axis is read per cycle, to increase speed
for (magAxis = 0; magAxis < 3; magAxis = (magAxis + 1) % 3)
{
    //Wait for magnetometer reading to complete
    while (!PORTEbits.RE5);

    //Make accelerometer measurement
    for(i=0;i<20;i++);     //5 uS pause
    PORTEbits.RE0 = 0;
    for(i=0;i<20;i++);     //5 uS pause

    //Request all three axes
    SPI1BUF = 0b00100100;
    while (!SPI1STATbits.SPIRBF);
    dummy = SPI1BUF;

    //Receive and process all three axes
    for (j = 0; j < 3; j++)

```

```

{
    SPI1BUF = 0b00000000; //dummy
    while (!SPI1STATbits.SPIRBF);
    accelAxes[j] = SPI1BUF;
    accelAxes[j] <<= 8;

    SPI1BUF = 0b00000000; //dummy
    while (!SPI1STATbits.SPIRBF);
    accelAxes[j] |= SPI1BUF;

    sign = accelAxes[j] & 0b1000000000000000;
    accelAxes[j] >>= 3;
    accelAxes[j] = (accelAxes[j] & 0b0000111111111111);
    if (sign != 0)
        accelAxes[j] = accelAxes[j] - 4096; //4096 = 2^12
}
PORTEbits.RE0 = 1;

```

```

//Receive magnetometer measurement
PORTDbits.RD0 = 0;
for(i=0;i<20;i++); //5 uS pause
SPI1BUF = 0b00000000; //dummy
while (!SPI1STATbits.SPIRBF);
magAxes[magAxis] = SPI1BUF;
magAxes[magAxis] <<= 8;
SPI1BUF = 0b00000000; //dummy
while (!SPI1STATbits.SPIRBF);
magAxes[magAxis] |= SPI1BUF;
PORTDbits.RD0 = 1;

```

```

//Make gyroscope measurements
for (j = 0; j < 3; j++)
{
    if (j==0)
        PORTEbits.RE3 = 0; //x-axis
    else if (j==1)
        PORTEbits.RE2 = 0; //y-axis
    else
        PORTEbits.RE1 = 0; //z-axis

    for(i=0;i<50;i++); //12.5 uS pause

    //Start ADC
    SPI1BUF = 0b10010100;
    while(!SPI1STATbits.SPIRBF);
    dummy = SPI1BUF;

    //Receive 16 bits (not used)
    SPI1BUF = 0b11111111; //dummy
    while(!SPI1STATbits.SPIRBF);
    dummy = SPI1BUF;
    SPI1BUF = 0b00000000; //dummy
    while(!SPI1STATbits.SPIRBF);
    dummy = SPI1BUF;
}

```

```

        //wait for measurement
        for(i=0;i<500;i++);           //125 uS pause

        //Request measurement
        SPI1BUF = 0b10000000;
        while(!SPI1STATbits.SPIRBF);
        dummy = SPI1BUF;
        //Receive measurement
        SPI1BUF = 0b11111111; //dummy
        while (!SPI1STATbits.SPIRBF);
        gyroAxes[j] = SPI1BUF;
        gyroAxes[j] <<= 8;
        SPI1BUF = 0b00000000; //dummy
        while (!SPI1STATbits.SPIRBF);
        gyroAxes[j] |= SPI1BUF;

        PORTE = 0b000001111;

        //Isolate data bits
        gyroAxes[j] >>= 1;
        gyroAxes[j] &= 0b0000011111111111;
    }

    //Request next magnetometer measurement
    PORTDbits.RD0 = 0;    //Magnetometer SS
    PORTEbits.RE4 = 1;    //Pulse Reset
    for(i=0;i<20;i++);    //5 uS pause
    PORTEbits.RE4 = 0;    //Clear Reset
    for(i=0;i<20;i++);    //5 uS pause

    if (magAxis == 2)      //x-axis
        SPI1BUF = 0b01010001;
    else if (magAxis == 0) //y-axis
        SPI1BUF = 0b01010010;
    else if (magAxis == 1) //z-axis
        SPI1BUF = 0b01010011;
    while (!SPI1STATbits.SPIRBF);
    dummy = SPI1BUF;
    PORTDbits.RD0 = 1;

    //Send measurements over UART while waiting for magnetometer
    for (j = 0; j < 3; j++)
    {
        SendIntUART(accelAxes[j]);
    }
    for (j = 0; j < 3; j++)
    {
        SendIntUART(gyroAxes[j]);
    }
    SendIntUART(magAxes[magAxis]);
}

return 0;

```

```
}  
  
void SendIntUART(int toSend)  
{  
    char UARTchar;  
    UARTchar = toSend >> 8; //MSB  
    U1TXREG = UARTchar;  
    while(U1STAbits.UTXBF==1);  
    UARTchar = (toSend << 8) >> 8; //LSB  
    U1TXREG = UARTchar;  
    while(U1STAbits.UTXBF==1);  
}
```

Appendix B: MATLAB Source Code

```
function x = Embedded_SRUKF_full()
% Square-Root Unscented Kalman Filter for Sensor Board
% Author: Chris Baldassano
% MATLAB version: 7.5.0 (R2007b)
% Based on: "THE SQUARE-ROOT UNSCENTED KALMAN FILTER
%           FOR STATE AND PARAMETER-ESTIMATION" by Rudolph van der Merwe and
%           Eric A. Wan
% Details: Designed for attitude estimation system with MicroMag3
%           magnetometer, SCA3000 accelerometer, and 3 MLX90609 gyroscopes

% Reset logs
clear global data;
clear global tocLog;
clear global SList;
clear global filtered;
clear global yMeanList;
clear global KList;
global filtered;
global SList;
global KList;
global yMeanList;
global data;
global tocLog;

% Setup serial port
delete(instrfindall);
s = serial('COM1', 'BAUD', 57600);
s.ByteOrder = 'bigEndian';
fopen(s);

% Define initial vectors and matrices, and weights for mean/covariance
stateDim = 7;
measDim = 7;
x = [0;0;0;1;0;0;0]; %initial state
S = chol(.5^2*eye(stateDim)); %initial state covariance
Q=.05^2*eye(stateDim); %process covariance
Q(1,1) = .1^2;
Q(2,2) = .1^2;
Q(3,3) = .1^2;
R=200^2*eye(measDim); %measurement covariance
R(4,4) = .01^2;
R(5,5) = .01^2;
R(6,6) = .01^2;
R(7,7) = 5^2;
sqQ = chol(Q);
sqR = chol(R);
alpha=.5;
beta=2;
Wm=[1-alpha^(2) (alpha^(2))/(2*stateDim)+zeros(1,2*stateDim)];
Wc=Wm;
Wc(1)=Wc(1)+(1-alpha^2+beta);
eta = sqrt(stateDim*alpha^2);

magAxis = 2;
tic; %start timing
```

```

fwrite(s,'M','char','async'); %request measurement
i = 0;
while(1)
    i = i+1;
    while (s.BytesAvailable < 14) %wait for measurement
    end
    tau = toc; %compute time elapsed
    tocLog(i) = tau;
    tic;
    currAxes = fread(s,7,'short');

    %data(1:3) -> acceleration
    %data(4:6) -> rotation
    %data(7) -> current axis of magnetic field
    data(i,1) = -2*currAxes(3); %X inverted
    data(i,2) = 2*currAxes(2);
    data(i,3) = -2*currAxes(1); %Z inverted
    data(i,4:6) = (((25/12)*currAxes(4:6)+400)-2500)*300/2000)*pi/180;
    data(i,7) = currAxes(7);

    %gyroscope calibration (drifts slowly over time)
    data(i,4) = data(i,4) + .01;
    data(i,5) = data(i,5) - .06;
    data(i,6) = data(i,6) + 0.0;

    magAxis = mod(magAxis+1,3); %get current magnetic axis

    %Calculate sigma points
    A = eta*S;
    xSigmaPts = [x x(:,ones(1,stateDim))+A x(:,ones(1,stateDim))-A];

    %Perform state update using unscented transform
    numpts=2*stateDim+1;
    xMeanUT=zeros(stateDim,1);
    xSigmaPtsUT=zeros(stateDim,numpts);
    for k=1:numpts
        xSigmaPtsUT(:,k)=processModel(xSigmaPts(:,k),tau);
        xMeanUT=xMeanUT+Wm(k)*xSigmaPtsUT(:,k);
    end
    [Temp S] = qr([sqrt(Wc(2))*(xSigmaPtsUT(:,2:numpts)-
xMeanUT(:,ones(1,numpts-1))) sqQ]',0);
    if (Wc(1) < 0)
        S = chol(S'*S-(((1*Wc(1))^(1/4))*(xSigmaPtsUT(:,1)-xMeanUT(:)))*(((1*
1*Wc(1))^(1/4))*(xSigmaPtsUT(:,1)-xMeanUT(:))));
    else
        S = chol(S'*S+((Wc(1)^(1/4))*(xSigmaPtsUT(:,1)-
xMeanUT(:)))*((Wc(1)^(1/4))*(xSigmaPtsUT(:,1)-xMeanUT(:))));
    end

    %Predict current measurements using unscented transform
    yMeanUT=zeros(measDim,1);
    ySigmaPtsUT=zeros(measDim,numpts);
    for k=1:numpts
        ySigmaPtsUT(:,k)=measurementModel(xSigmaPtsUT(:,k),magAxis);
        yMeanUT=yMeanUT+Wm(k)*ySigmaPtsUT(:,k);
    end
end

```

```

    [Temp Sy] = qr([sqrt(Wc(2))*(ySigmaPtsUT(:,2:numpts)-
yMeanUT(:,ones(1,numpts-1))) sqrtR]',0);
    if (Wc(1) < 0)
        Sy = chol(Sy'*Sy - (((-1*Wc(1))^(1/4))*(ySigmaPtsUT(:,1)-
yMeanUT(:)))*((-1*Wc(1))^(1/4))*(ySigmaPtsUT(:,1)-yMeanUT(:))));
    else
        Sy = chol(Sy'*Sy + ((Wc(1)^(1/4))*(ySigmaPtsUT(:,1)-
yMeanUT(:)))*((Wc(1)^(1/4))*(ySigmaPtsUT(:,1)-yMeanUT(:))));
    end

    Sy = Sy';    %Switch to lower triangular matrix

    %Calculate Kalman gain matrix
    Pxy = (xSigmaPtsUT(:,1:numpts)-
xMeanUT(:,ones(1,numpts)))*diag(Wc)*(ySigmaPtsUT(:,1:numpts)-
yMeanUT(:,ones(1,numpts))));
    K = (Pxy/(Sy)')/Sy;

    %Update state estimate
    x = xMeanUT + K*(data(i,:) - yMeanUT);
    x = normalizeQ(x);

    %Update state covariance
    upMat = K*Sy;
    for k=1:measDim
        S = chol(S'*S - upMat(:,k)*upMat(:,k));
    end
    S = S';    %Switch to lower triangular matrix

    %Log data for analysis
    SList{i} = S;
    KList{i} = K;
    filtered(i,:) = x;
    yMeanList(i,:) = yMeanUT;

    %Plot data for GUI
    start = i - 99;
    if (start < 1)
        start = 1;
    end
    start = start - mod(start,3) + 1;
    if (mod(i,10) == 0)
        figure(1);
        plot(start:i,data(start:i,1:3),'-');
        hold on;
        plot(start:i,yMeanList(start:i,1:3),'--');
        axis([start i+1 -1000 1000]);
        hold off;
        xlabel('Samples')
        ylabel('Acceleration')

        figure(2);
        plot(start:i,data(start:i,4:6),'-');
        hold on;
        plot(start:i,yMeanList(start:i,4:6),'--');

```

```

axis([start i+1 -1 1]);
hold off;
xlabel('Samples')
ylabel('Rate of Rotation')

figure(3);
plot(start:3:i,data(start:3:i,7),'-
',start+1:3:i,data(start+1:3:i,7),'-',start+2:3:i,data(start+2:3:i,7),'-');
hold on;
plot(start:3:i,yMeanList(start:3:i,7),'--
',start+1:3:i,yMeanList(start+1:3:i,7),'--
',start+2:3:i,yMeanList(start+2:3:i,7),'--');
axis([start i+1 -500 500]);
hold off;
xlabel('Samples')
ylabel('Magnetic Field')

figure(5);
clf;
axis([-1 1 -1 1 -1 1]);
disp3d(x);
end

end

function x = normalizeQ(nonUnit)
%Maintain quaternion normalization
mag = sqrt(nonUnit(4)^2+nonUnit(5)^2+nonUnit(6)^2+nonUnit(7)^2);
x = [0;0;0;0;0;0;0];
x(1:3) = nonUnit(1:3);
x(4:7) = nonUnit(4:7)/mag;

function xp = processModel(x,tau)
%Time-update function
xp = (eye(7) + (tau/2)*...
[0 0 0 0 0 0 0;...
0 0 0 0 0 0 0;...
0 0 0 0 0 0 0;...
0 0 0 0 -x(1) -x(2) -x(3);...
0 0 0 x(1) 0 x(3) -x(2);...
0 0 0 x(2) -x(3) 0 x(1);...
0 0 0 x(3) x(2) -x(1) 0])*x;

function yp = measurementModel(x,magAxis)
%Measurement prediction functions
yp = [0;0;0;0;0;0;0];
magneticMagnitude = 500;
gravMagnitude = 1000;
currR = RotationMat(x);
yp(1:3) = currR*gravMagnitude*[0 0 -1]';
yp(4) = x(1);
yp(5) = x(2);
yp(6) = x(3);
threeMag = currR*magneticMagnitude* [.1371 -.2059 .9689]';
yp(7) = threeMag(magAxis+1);

```

```

function disp3d(x)
%Show 3D view of attitude estimate
currR = inv(RotationMat(x));
t = currR*[1 0 0]';
line([0 t(1)],[0 t(2)],[0 t(3)], 'Color', 'b','LineWidth',2);
t = currR*[0 1 0]';
line([0 t(1)],[0 t(2)],[0 t(3)], 'Color', 'g','LineWidth',2);
t = currR*[0 0 1]';
line([0 t(1)],[0 t(2)],[0 t(3)], 'Color', 'r','LineWidth',2);

line([0 1],[0 0],[0 0], 'Color', 'b','LineWidth',1,'LineStyle', ':');
line([0 0],[0 1],[0 0], 'Color', 'g','LineWidth',1,'LineStyle', ':');
line([0 0],[0 0],[0 1], 'Color', 'r','LineWidth',1,'LineStyle', ':');

function R = RotationMat(x)
%Calculate rotation between reference frame and state estimate
R = ([x(4)^2+x(5)^2-x(6)^2-x(7)^2,2*(x(5)*x(6)+x(4)*x(7)),2*(x(5)*x(7)-
x(4)*x(6));2*(x(5)*x(6)-x(4)*x(7)),x(4)^2-x(5)^2+x(6)^2-
x(7)^2,2*(x(6)*x(7)+x(4)*x(5));2*(x(5)*x(7)+x(4)*x(6)),2*(x(6)*x(7)-
x(4)*x(5)),x(4)^2-x(5)^2-x(6)^2+x(7)^2]);

```



SUNFEST

Symposium

Thursday, August 7, 2008

10:00 A.M.

Wu and Chen Aud., Levine

You are invited to join the SUNFEST students for their final presentations.

A NSF-REU program

University of Pennsylvania

Center for Sensor Technologies



SUNFEST Symposium

August 7, 2008

Schedule

- 10am: Chris Baldasanno, *Compact Attitude Sensor System using SR-UKF*
(Prof. Mark Yim)
- 10:15: Clarence Agbi, *Enabling Force Control for Cooperative Towing Robots* (V. Kumar and J. Keller)
- 10:30 Ramon Figueroa-Diaz, *Dye-Sensitized ZnO fibers from Electrospinning and Photovoltaic Cells* (J. Santiago)
- 10:45 Alexei Matyushov *Growth of Long Carbon Nanotubes Via Chemical Vapor Deposition* (A. Johnson and Z Luo)
- 11:00 Erika Martinez, *Development of a micro PCR reactor for Lab-on-chip devices* (H. Bau, J. Thompson)
- 11:15 Rony Kamruzzaman, *Optimizing Legged Locomotion Using Tunable Leg Stiffness* (K. Galloway and D. Koditschek)
- 11:30 Emily Wible, *Investigation Of MSC Differentiation On Electrospun Nanofibrous Scaffolds* (R. Mauck and N. Nerurkar)
- 11:45: Uchenna Anyanwu, *A Scalable Traffic Simulator for Congestion Probing and Prediction* (R. Mangharam)
- 12:00: **Lunch**
- 1:00 Anil Venkatesh, *Talking to Robots: Speech-Directed Motion Planning* (G. Pappas)
- 1:15pm David Joffe *NeoNur: A Feeding Device for Premature Neonatal Nursing* (J. Zemel)
- 1:30 Alta Berger, *The Effects of Interleukin-1 β on the Rat Lumbar Intervertebral Disc* (Prof. D. Elliott)



SUMMER RESEARCH OPPORTUNITY

University of Pennsylvania



SUNFEST



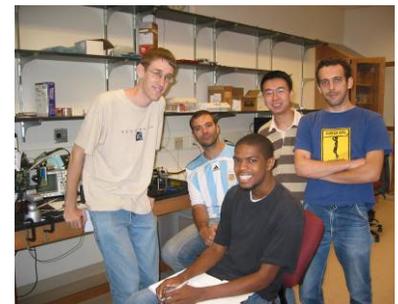
Summer Undergraduate Fellowships in Sensor Technologies
May 28 (est.), 2008 – August 9, 2008



The Center for Sensor Technologies is offering Summer Fellowships for Undergraduate Students ending their sophomore or junior year of college in engineering or science. The fellowships are funded by the National Science Foundation's REU Program. Projects involve:

- Micro and Nano Electromechanical Structures
- Nanotechnology and materials for Sensors
- Robotics and Control oriented projects
- Wireless Sensor Networks
- Sensors for Medical and Biomedical Applications
- Optical and Vision Sensors

1. Summer research experience with a stipend of \$4,400
2. Subsidized Housing on campus; partial travel support
3. Opportunity to work on an interesting interdisciplinary topic
4. Work under the mentorship of a faculty member and graduate students
5. Learn How to Give Effective Presentations, How to Write Technical Reports, Applying to Graduate School, and Ethics in Engineering and Science



Application Information is available at: <http://www.seas.upenn.edu/~sunfest>



Application Deadline: March 2, 2008

Applications by women, minority students and students with disabilities are encouraged!

sunfest@seas.upenn.edu