2013

Development and optimization of an integrated Faraday modulator and compensator design for continuous polarimetric glucose monitoring

Brandon William Clarke
The University of Toledo

Follow this and additional works at: http://utdr.utoledo.edu/theses-dissertations

Recommended Citation
Clarke, Brandon William, "Development and optimization of an integrated Faraday modulator and compensator design for continuous polarimetric glucose monitoring" (2013). Theses and Dissertations. 47.
http://utdr.utoledo.edu/theses-dissertations/47

This Thesis is brought to you for free and open access by The University of Toledo Digital Repository. It has been accepted for inclusion in Theses and Dissertations by an authorized administrator of The University of Toledo Digital Repository. For more information, please see the repository's About page.
A Thesis
entitled
Development and Optimization of an Integrated Faraday Modulator and Compensator Design for Continuous Polarimetric Glucose Monitoring
by
Brandon William Clarke
Submitted to the Graduate Faculty as partial fulfillment of the requirements for the Master of Science Degree in Bioengineering

Dr. Brent D. Cameron, Committee Chair

Dr. Patricia A. Relue, Committee Member

Dr. Eda Yildirim-Ayan, Committee Member

Dr. Ronald L. Fournier, Committee Member

Dr. Patricia R. Komuniecki, Dean
College of Graduate Studies

The University of Toledo
May 2013
An Abstract of

Development and Optimization of an Integrated Faraday Modulator and Compensator Design for Continuous Polarimetric Glucose Monitoring

by

Brandon William Clarke

Submitted to the Graduate Faculty as partial fulfillment of the requirements for the Master of Science Degree in Bioengineering

The University of Toledo

May 2013

In recent years, significant advances have been made in the development of noninvasive polarimetric glucose detection systems, salutary for the treatment of our rapidly increasing diabetic population. This area of research utilizes the aqueous humor as the detection medium for its strong correlation to blood glucose concentration and highlights three major features: the optical activity of glucose, minimal scattering of the medium, and the ability to detect sub-millidegree rotation in polarized light. However, many of the current polarimetric systems are faced with size and cost constraints based on the paramount optical components (e.g., terbium gallium garnet or terbium doped glass) and custom wound inductive coils. As a step toward developing a low cost hand-held design, a miniaturized integrated single-crystal Faraday modulator/compensator (IFMC) has been designed and optimized. This device is capable of replacing the traditional two component arrangement that has been widely reported on in many Faraday-based polarimetric configurations.
In this work, an electromagnetic (EM) finite element model (FEM) was developed that can simulate various physical parameters such as geometry, inductance, and orientation of an IFMC with respect to the optical components in order to minimize power consumption and size while maintaining appropriate magnetic field strength. The newly designed prototype was compared with the FEM, providing excellent correlation with operational performance shown to be within 1.8% of predicted values. It was shown that the use of FEM simulations allows for the analysis of a vast range of parameters before prototypes are fabricated and can facilitate custom designs as related to development time, anticipated performance, and cost reduction. Furthermore, the performance of the IFMC was evaluated experimentally under both noninvasive static and dynamic glucose monitoring conditions with a custom designed glucose flow system. The dynamic flow system provides a repeatable and controllable testing environment which can recreate in vivo glucose profiles while reducing the need for repetitive, expensive, and time consuming animal experiments. Finally, it was shown that the combined rotator can achieve modulation depths above 1°, and when operating in a compensated closed-loop configuration, it had demonstrated glucose prediction errors of 1.8 mg/dL and 5.4 mg/dL under hypoglycemic and hyperglycemic conditions, respectively. These results demonstrate that such an integrated design can perform similar to, if not better than, its larger two-part predecessors. Overall, this technology is capable of expediting future research and development in providing a fully functional and commercially available noninvasive polarimetric glucose sensor.
To my parents, Tom and Karen. Your undying love and support has always provided me with the motivation needed to accomplish my goals, through the good and the bad, to make me who I am today.
Acknowledgements

First and foremost, I want to thank my parents for the support they have given and the sacrifices they have made to get me where I am today. Second, my girlfriend and best friend, Kristin, has stood by my side through my best and my worst and I couldn’t begin to show my gratitude for that. Furthermore, my sister, Kristel, and the rest of my family and friends have always provided the encouragement that I needed at all of the right times.

The work contained in this thesis would not have been possible without the guidance and support from my advisor, Dr. Cameron. I have found that hard work and motivation does pay off and his rigorous mentoring has gotten me to this point. I would also like to thank my colleagues, friends, and mentors in the Department of Bioengineering, Nathan Reaver, Rui Zheng, Yongsoon Hwang, Niraj Gupta, Tammy Phares, and other students, faculty, and staff for providing insightful feedback, humorous conversations, and a light-hearted environment, adding balance during stressful times.

Finally, I would like to acknowledge the Center for Materials and Sensor Characterization (CMSC) at the University of Toledo (UT) for instrumentation support as well as the Engineering Center for Orthopaedic Research Excellence (E-CORE) at UT and John Jaegly, Laboratory Supervisor in the UT Department of Mechanical, Industrial, and Manufacturing Engineering (MIME), for providing 3D printing and prototyping capabilities. Furthermore, thanks to Dr. Pappada, Scott Jenkins, and Ryan Ariss for their assistance in providing and collecting the data presented in this thesis.
# Table of Contents

Abstract ........................................................................................................................................ iii

Acknowledgements .................................................................................................................. vi

Table of Contents .................................................................................................................... vii

List of Tables .......................................................................................................................... x

List of Figures .......................................................................................................................... xii

List of Abbreviations .............................................................................................................. xix

List of Symbols ......................................................................................................................... xxi

1 Introduction ............................................................................................................................. 1

1.1 Literature Review ............................................................................................................... 4

1.1.1 The Origins of Polarimetry in Glucose Sensing ......................................................... 7

1.1.2 Tackling the Challenges of Polarimetric Sensing ....................................................... 9

1.1.3 Polarimetry in Turbid Media ...................................................................................... 16

1.1.4 The In Vivo Application of Polarimetric Sensing ...................................................... 19

1.2 The Current Objectives .................................................................................................... 21

2 Theory .................................................................................................................................... 26

2.1 Glucose as a Target Molecule ......................................................................................... 27

2.1.1 Continuous Glucose Monitoring .............................................................................. 28

2.1.2 Optical Activity ......................................................................................................... 32

2.2 Optical Polarimetry .......................................................................................................... 41
2.2.1 Faraday Rotation in Polarimetry .................................................. 43
2.2.2 Magnetic Fields in Faraday Rotation ........................................... 45
2.2.3 Modeling the Optical Components .............................................. 47
2.3 Finite Element Modeling (FEM) ..................................................... 49
3 Materials and Methods ................................................................. 52
  3.1 Building the Initial FEM ............................................................. 53
  3.2 FEM of the Compensator and Modulator Coils ........................... 57
  3.3 Validation of the FEM ............................................................... 61
  3.4 IFMC Design and Optimization .................................................. 64
  3.5 Static Validation of the IFMC ...................................................... 67
  3.6 Programmable Flow System Design .......................................... 69
  3.7 Continuous Dynamic Glucose Detection ..................................... 76
4 Results and Discussion ................................................................. 80
  4.1 Building the Initial FEM ............................................................. 81
  4.2 FEM of the Compensator and Modulator Coils ........................... 84
  4.3 Validation of the FEM ............................................................... 92
  4.4 IFMC Design and Optimization .................................................. 95
  4.5 Static Validation of the IFMC ...................................................... 99
  4.6 Programmable Flow System Design .......................................... 101
  4.7 Continuous Dynamic Glucose Detection ..................................... 102
5 Conclusions and Future Work ...................................................... 111
References ......................................................................................... 116
A Coilcraft Inductor Specifications .................................................. 127
<table>
<thead>
<tr>
<th></th>
<th>Title</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>B</td>
<td>Digi-Key Inductor Specifications</td>
<td>130</td>
</tr>
<tr>
<td>C</td>
<td>IFMC Electronic Components</td>
<td>133</td>
</tr>
<tr>
<td>D</td>
<td>IFMC PID Digital Control VI</td>
<td>136</td>
</tr>
<tr>
<td>E</td>
<td>WELCO Peristaltic Pump Specifications</td>
<td>141</td>
</tr>
<tr>
<td>F</td>
<td>LIFA Firmware Code</td>
<td>149</td>
</tr>
<tr>
<td>G</td>
<td>LIFA Header Code</td>
<td>151</td>
</tr>
<tr>
<td>H</td>
<td>LIFA Source Code</td>
<td>156</td>
</tr>
<tr>
<td>I</td>
<td>AFMotor Header Code</td>
<td>172</td>
</tr>
<tr>
<td>J</td>
<td>AFMotor Source Code</td>
<td>175</td>
</tr>
<tr>
<td>K</td>
<td>AccelStepper Header Code</td>
<td>186</td>
</tr>
<tr>
<td>L</td>
<td>AccelStepper Source Code</td>
<td>195</td>
</tr>
<tr>
<td>M</td>
<td>LIFA VIs Used in Flow System Programs</td>
<td>205</td>
</tr>
<tr>
<td>N</td>
<td>Flow System Control VIs</td>
<td>221</td>
</tr>
<tr>
<td>O</td>
<td>Raw Collected Data</td>
<td>246</td>
</tr>
<tr>
<td>Table No.</td>
<td>Title</td>
<td>Page</td>
</tr>
<tr>
<td>----------</td>
<td>----------------------------------------------------------------------</td>
<td>------</td>
</tr>
<tr>
<td>2.1</td>
<td>Definition of parameters used in the insulin/glucose model</td>
<td>31</td>
</tr>
<tr>
<td>2.2</td>
<td>Specific rotation of glucose at different wavelengths</td>
<td>41</td>
</tr>
<tr>
<td>3.1</td>
<td>Design parameters used to build each inductor within FEM</td>
<td>58</td>
</tr>
<tr>
<td>3.2</td>
<td>Material properties input to the FEM</td>
<td>59</td>
</tr>
<tr>
<td>4.1</td>
<td>Initial 2D FEM magnetic field results compared to Biot-Savart calculations</td>
<td>82</td>
</tr>
<tr>
<td>4.2</td>
<td>Error in FEM calculated inductance and impedance for off-the-shelf coils</td>
<td>84</td>
</tr>
<tr>
<td>4.3</td>
<td>Error results from each continuous glucose experiment with the IFMC</td>
<td>105</td>
</tr>
<tr>
<td>O.1</td>
<td>Linear voltage sweep data from the 6.8 mH coil in compensation</td>
<td>246</td>
</tr>
<tr>
<td>O.2</td>
<td>Linear voltage sweep data from the 15 mH coil in compensation</td>
<td>246</td>
</tr>
<tr>
<td>O.3</td>
<td>Linear voltage sweep data from the 68 mH coil in compensation</td>
<td>246</td>
</tr>
<tr>
<td>O.4</td>
<td>Linear voltage sweep data from the 100 mH coil in compensation</td>
<td>246</td>
</tr>
<tr>
<td>O.5</td>
<td>Linear voltage sweep data from the 220 mH coil in compensation</td>
<td>247</td>
</tr>
<tr>
<td>O.6</td>
<td>Linear voltage sweep data from the 470 mH coil in compensation</td>
<td>247</td>
</tr>
<tr>
<td>O.7</td>
<td>Linear voltage sweep data from the 6.8 mH coil in modulation</td>
<td>247</td>
</tr>
<tr>
<td>O.8</td>
<td>Linear voltage sweep data from the 15 mH coil in modulation</td>
<td>247</td>
</tr>
<tr>
<td>O.9</td>
<td>Linear voltage sweep data from the 68 mH coil in modulation</td>
<td>247</td>
</tr>
<tr>
<td>O.10</td>
<td>Linear voltage sweep data from the 100 mH coil in modulation</td>
<td>247</td>
</tr>
<tr>
<td>O.11</td>
<td>Linear voltage sweep data from the 220 mH coil in modulation</td>
<td>248</td>
</tr>
<tr>
<td>O.12</td>
<td>Linear voltage sweep data from the 470 mH coil in modulation</td>
<td>248</td>
</tr>
</tbody>
</table>
O.13 Maximum rotational data from each coil in modulation ........................................248
O.14 Maximum rotational data from each coil in compensation .....................................249
O.15 Compensator output, IFMC, static sampling, hypoglycemic range ..........................249
O.16 Compensator output, IFMC, static sampling, hyperglycemic range .........................250
O.17 YSI data from flow system samples, hypoglycemic range .......................................250
O.18 YSI data from flow system samples, hyperglycemic range ....................................251
O.19 Compensator output, IFMC, dynamic sampling, trial 1 ........................................251
O.20 Compensator output, IFMC, dynamic sampling, trial 2 .......................................252
O.21 Compensator output, IFMC, dynamic sampling, trial 3 ........................................252
### List of Figures

<table>
<thead>
<tr>
<th>Figure</th>
<th>Description</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>2-1</td>
<td>Non-diabetic glucose response following 50 g OGTT</td>
<td>33</td>
</tr>
<tr>
<td>2-2</td>
<td>Component breakdown of polarized light and circular birefringence</td>
<td>34</td>
</tr>
<tr>
<td>2-3</td>
<td>R and S enantiomers of a chiral molecule</td>
<td>36</td>
</tr>
<tr>
<td>2-4</td>
<td>D and L enantiomers of glucose compared to glyceraldehyde</td>
<td>37</td>
</tr>
<tr>
<td>2-5</td>
<td>Anomers of D-glucose</td>
<td>39</td>
</tr>
<tr>
<td>2-6</td>
<td>ORD curve for glucose in the visible spectrum</td>
<td>40</td>
</tr>
<tr>
<td>2-7</td>
<td>A schematic demonstrating the principle of a basic polarimeter</td>
<td>42</td>
</tr>
<tr>
<td>2-8</td>
<td>Faraday rotation in the presence of a DC axial magnetic field component</td>
<td>45</td>
</tr>
<tr>
<td>2-9</td>
<td>Relationship between current and magnetic field in space</td>
<td>46</td>
</tr>
<tr>
<td>2-10</td>
<td>Visual representation of a Faraday-modulated polarimetric signal</td>
<td>48</td>
</tr>
<tr>
<td>3-1</td>
<td>Basic geometry of a single loop of current</td>
<td>54</td>
</tr>
<tr>
<td>3-2</td>
<td>Single loop FEM geometry with 2D axial symmetry</td>
<td>56</td>
</tr>
<tr>
<td>3-3</td>
<td>Geometric orientation of the compensator and modulator FEM</td>
<td>59</td>
</tr>
<tr>
<td>3-4</td>
<td>Schematic of polarimetric system used to measure Faraday rotations</td>
<td>61</td>
</tr>
<tr>
<td>3-5</td>
<td>Final IFMC design fixture</td>
<td>66</td>
</tr>
<tr>
<td>3-6</td>
<td>Schematic of the polarimetric glucose sensing system</td>
<td>68</td>
</tr>
<tr>
<td>3-7</td>
<td>Schematic of the flow system coupled with the polarimeter</td>
<td>71</td>
</tr>
<tr>
<td>3-8</td>
<td>Custom designed 1 cm flow cell for continuous measurements</td>
<td>72</td>
</tr>
<tr>
<td>3-9</td>
<td>Wiring of the microcontroller and stepper motor driver</td>
<td>73</td>
</tr>
<tr>
<td>Page</td>
<td>Description</td>
<td></td>
</tr>
<tr>
<td>------</td>
<td>-------------</td>
<td></td>
</tr>
<tr>
<td>3-10</td>
<td>Front panel view of the flow system purge program ........................................ 74</td>
<td></td>
</tr>
<tr>
<td>3-11</td>
<td>Front panel view of the real-time flow system control program .......................... 74</td>
<td></td>
</tr>
<tr>
<td>3-12</td>
<td>Front panel view of the continuous flow system control program ....................... 75</td>
<td></td>
</tr>
<tr>
<td>3-13</td>
<td>Secondary glucose profiles used to test the IFMC............................................. 78</td>
<td></td>
</tr>
<tr>
<td>4-1</td>
<td>FEM results from a single loop of current ...................................................... 82</td>
<td></td>
</tr>
<tr>
<td>4-2</td>
<td>FEM results from three loops of current ....................................................... 83</td>
<td></td>
</tr>
<tr>
<td>4-3</td>
<td>Correlation of inductance and impedance between predicted and measured values for six off-the-shelf coils ................................................................. 85</td>
<td></td>
</tr>
<tr>
<td>4-4</td>
<td>FEM results from each coil in compensation .................................................... 86</td>
<td></td>
</tr>
<tr>
<td>4-5</td>
<td>FEM results from each coil in modulation ...................................................... 88</td>
<td></td>
</tr>
<tr>
<td>4-6</td>
<td>FEM parametric voltage sweep results from each coil in compensation ................. 89</td>
<td></td>
</tr>
<tr>
<td>4-7</td>
<td>FEM parametric voltage sweep results from each coil in modulation .................... 90</td>
<td></td>
</tr>
<tr>
<td>4-8</td>
<td>Relationship between rotational depth and driving voltage for each coil ............ 91</td>
<td></td>
</tr>
<tr>
<td>4-9</td>
<td>Comparative results from each coil when driven with the same voltage ............. 92</td>
<td></td>
</tr>
<tr>
<td>4-10</td>
<td>Prediction error in compensation and modulation between FEM analysis and physical measured values ................................................................. 93</td>
<td></td>
</tr>
<tr>
<td>4-11</td>
<td>Plots used for selecting coils based on specific rotational requirements ........... 96</td>
<td></td>
</tr>
<tr>
<td>4-12</td>
<td>Final FEM results of the IFMC in compensation and modulation ....................... 98</td>
<td></td>
</tr>
<tr>
<td>4-13</td>
<td>Error results using the IFMC to measure hypoglycemic glucose concentrations .............. 100</td>
<td></td>
</tr>
<tr>
<td>4-14</td>
<td>Error results using the IFMC to measure hyperglycemic glucose concentrations ............ 100</td>
<td></td>
</tr>
<tr>
<td>4-15</td>
<td>Trial 1 of continuous non-diabetic glucose monitoring with the IFMC ........... 103</td>
<td></td>
</tr>
</tbody>
</table>
N-5  Flow System Purge VI block diagram, right view.................................223
N-6  Flow System Concentration Change VI front panel, full view..................223
N-7  Flow System Concentration Change VI block diagram, left view 1 ..........224
N-8  Flow System Concentration Change VI block diagram, left view 2 ..........224
N-9  Flow System Concentration Change VI block diagram, left view 3 ..........225
N-10 Flow System Concentration Change VI block diagram, left view 4 ..........225
N-11 Flow System Concentration Change VI block diagram, left view 5 ..........226
N-12 Flow System Concentration Change VI block diagram, left-center view 1 ....226
N-13 Flow System Concentration Change VI block diagram, left-center view 2 ....227
N-14 Flow System Concentration Change VI block diagram, left-center view 3 ....227
N-15 Flow System Concentration Change VI block diagram, left-center view 4 ....228
N-16 Flow System Concentration Change VI block diagram, left-center view 5 ....228
N-17 Flow System Concentration Change VI block diagram, left-center view 6 ....229
N-18 Flow System Concentration Change VI block diagram, right-center view 1 ...229
N-19 Flow System Concentration Change VI block diagram, right-center view 2 ...230
N-20 Flow System Concentration Change VI block diagram, right-center view 3 ...230
N-21 Flow System Concentration Change VI block diagram, right-center view 4 ...231
N-22 Flow System Concentration Change VI block diagram, right-center view 5 ...231
N-23 Flow System Concentration Change VI block diagram, right-center view 6 ...232
N-24 Flow System Concentration Change VI block diagram, right view ..............232
N-25 Flow System Profile Control VI front panel, full view ........................233
N-26 Flow System Profile Control VI block diagram, left view ........................233
N-27 Flow System Profile Control VI block diagram, left-center view 1 ............234
N-28  Flow System Profile Control VI block diagram, left-center view 2 .................234
N-29  Flow System Profile Control VI block diagram, left-center view 3 .................235
N-30  Flow System Profile Control VI block diagram, left-center view 4 .................235
N-31  Flow System Profile Control VI block diagram, left-center view 5 .................236
N-32  Flow System Profile Control VI block diagram, left-center view 6 .................236
N-33  Flow System Profile Control VI block diagram, center view 1 .......................237
N-34  Flow System Profile Control VI block diagram, center view 2 .......................237
N-35  Flow System Profile Control VI block diagram, center view 3 .......................238
N-36  Flow System Profile Control VI block diagram, center view 4 .......................238
N-37  Flow System Profile Control VI block diagram, center view 5 .......................239
N-38  Flow System Profile Control VI block diagram, center view 6 .......................239
N-39  Flow System Profile Control VI block diagram, right-center view ..................240
N-40  Flow System Profile Control VI block diagram, right view ..........................241
N-41  Flow System Pump Configuration VI front panel, full view ...........................242
N-42  Flow System Pump Configuration VI block diagram, full view ......................242
N-43  Flow System Stepper Disconnect VI front panel, full view ...........................243
N-44  Flow System Stepper Disconnect VI block diagram, full view 1 .................243
N-45  Flow System Stepper Disconnect VI block diagram, full view 2 .................244
N-46  Flow System Step Calculation VI front panel, full view ..............................244
N-47  Flow System Step Calculation VI block diagram, full view ...........................245
# List of Abbreviations

<table>
<thead>
<tr>
<th>Abbreviation</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>Analyzer</td>
</tr>
<tr>
<td>C</td>
<td>Compensator Coil</td>
</tr>
<tr>
<td>CDC</td>
<td>Centers for Disease Control and Prevention</td>
</tr>
<tr>
<td>CEGA</td>
<td>Clarke Error Grid Analysis</td>
</tr>
<tr>
<td>CF</td>
<td>Central Flask</td>
</tr>
<tr>
<td>CMSC</td>
<td>Center for Material and Sensor Characterization</td>
</tr>
<tr>
<td>D</td>
<td>Dextrorotatory, clockwise (with respect to glyceraldehyde)</td>
</tr>
<tr>
<td>DDE</td>
<td>Delay Differential Equation</td>
</tr>
<tr>
<td>DI</td>
<td>Deionized water</td>
</tr>
<tr>
<td>DOP</td>
<td>Degree of Polarization</td>
</tr>
<tr>
<td>DPP</td>
<td>Diabetes Prevention Program</td>
</tr>
<tr>
<td>E-CORE</td>
<td>Engineering Center for Orthopaedic Research Excellence</td>
</tr>
<tr>
<td>EM</td>
<td>Electromagnetic</td>
</tr>
<tr>
<td>FC</td>
<td>Flow Cell</td>
</tr>
<tr>
<td>FEM</td>
<td>Finite Element Model(-ing)</td>
</tr>
<tr>
<td>FPG</td>
<td>Fasting Plasma Glucose</td>
</tr>
<tr>
<td>GDM</td>
<td>Gestational Diabetes Mellitus</td>
</tr>
<tr>
<td>GIF</td>
<td>Glucose Flask</td>
</tr>
<tr>
<td>HPL</td>
<td>Human Placental Lactogen</td>
</tr>
<tr>
<td>IDE</td>
<td>Integro-Differential Equation</td>
</tr>
<tr>
<td>IFMC</td>
<td>Integrated Faraday Modulator and Compensator</td>
</tr>
<tr>
<td>IVGTT</td>
<td>Intravenous Glucose Tolerance Test</td>
</tr>
<tr>
<td>L</td>
<td>Levorotatory, counterclockwise (with respect to glyceraldehyde)</td>
</tr>
<tr>
<td>LIFA</td>
<td>LabVIEW Interface for Arduino</td>
</tr>
<tr>
<td>LS</td>
<td>Light Source</td>
</tr>
<tr>
<td>M</td>
<td>Modulator Coil</td>
</tr>
<tr>
<td>M1</td>
<td>Modulator Coil #1</td>
</tr>
<tr>
<td>M2</td>
<td>Modulator Coil #2</td>
</tr>
<tr>
<td>M3</td>
<td>Modulator Coil #3</td>
</tr>
<tr>
<td>MAE</td>
<td>Mean Absolute Error</td>
</tr>
<tr>
<td>MC</td>
<td>Microcontroller</td>
</tr>
<tr>
<td>MD</td>
<td>Motor Driver</td>
</tr>
<tr>
<td>MIME</td>
<td>Department of Mechanical, Industrial, and Manufacturing Engineering</td>
</tr>
<tr>
<td>NZW</td>
<td>New Zealand White</td>
</tr>
<tr>
<td>OCT</td>
<td>Optical Coherence Tomography</td>
</tr>
</tbody>
</table>
ODE .................................. Ordinary Differential Equation
OGTT .................................. Oral Glucose Tolerance Test
ORT .................................. Optical Rotatory Dispersion
P ......................................... Polarizer
P1 ...................................... Pump #1
P2 ...................................... Pump #2
P3 ...................................... Pump #3
P4 ...................................... Pump #4
PCA .................................... Principal Component Analysis
PD ...................................... Photodetector
PDE .................................... Partial Differential Equation
PEM .................................... Photoelastic Modulator
PID .................................... Proportional-Integral-Derivative
PLS .................................... Partial Least Squares
pt1 ..................................... FEM analysis point 1, (0, 10, 0)
pt2 ..................................... FEM analysis point 2, (0, 10, 10)
pt3 ..................................... FEM analysis point 3, (0, 0, 0)
R ......................................... Rectus, meaning right in Latin
RBC ..................................... Red Blood Cells
RTPR .................................... Rotation To Power Ratio
S ......................................... Sinister, meaning left in Latin
SEC .................................... Standard Error of Calibration
SEP .................................... Standard Error of Prediction
SP ...................................... Stir Plate
TDG .................................... Terbium Doped Glass
TEM .................................... Transverse Electromagnetic Waves
TGG .................................... Terbium Gallium Garnet
UT ..................................... University of Toledo
V ......................................... Terbium Doped Glass rod
VI ...................................... Virtual Instrument
VPR .................................... Volume Per Rotation
WsF .................................... Waste Flask
List of Symbols

\( \nabla \) Del operator

\( \alpha \) Anomer of D-glucose with hydroxyl on the same side in both chiral centers

\( \alpha \) Observed rotation of the polarization plane of light (\(^\circ\))

\( \alpha \) Arbitrary combination used for separation vector magnitude in Biot-Savart Law \((\text{mm}^2)\)

\( [\alpha]_{\lambda, \text{pH}}^T \) Specific rotation of a chiral molecule at a given temperature, wavelength, and pH \((\text{°/(dm}*^g/\text{mL}))\)

\( \beta \) Anomer of D-glucose with hydroxyl on the opposite side in each chiral center

\( \varepsilon_0 \) Vacuum permittivity \((\text{F/m})\)

\( \varepsilon_r \) Relative permittivity

\( \theta \) Angular separation between plane wave and analyzer in polarimeter \((\text{°})\)

\( \theta_m \) Modulation depth \((\text{rad})\)

\( \lambda \) Wavelength \((\text{nm})\)

\( \lambda_0 \) Absorption maximum \((\text{nm})\)

\( \lambda_0 \) Vacuum wavelength \((\text{nm})\)

\( \lambda_i \) Absorption wavelengths \((\text{nm})\)

\( \mu_0 \) Vacuum permeability of free space \((4\pi \cdot 10^{-7} \text{ N/A}^2)\)

\( \mu_r \) Relative permeability

\( \sigma \) Conductivity \((\text{S/m})\)

\( \Phi \) Observed rotation of the polarization plane of light \((\text{°})\)

\( \phi \) Rotational difference between optically active sample and Faraday compensator \((\text{rad})\)

\( \omega \) Frequency \((\text{rad/sec})\)

\( \omega_m \) Modulation frequency \((\text{rad/sec})\)

\( A \) Vector potential \((\text{V}*\text{s/m})\)

\( B \) Magnetic field vector in 3D space \((\text{G})\)

\( B \) Magnetic field magnitude in the direction of travel \((\text{G})\)

\( C \) Capacitance \((\mu\text{F})\)

\( C \) Sample concentration \((\text{mg/dL})\)

\( C \) Set of residuals from a calibration data set \((\text{mg/dL})\)
$D$..........................Total ingested glucose (mg)
$E$............................Electric field component of an electromagnetic wave (V/m)
$E$............................Rate constant of insulin exchange between blood and interstitial fluid (mL/min)
$f$..............................Modulation frequency (Hz)
$f_1$..........................Function that represents the infusion of insulin into the blood (μU/min)
$f_2$..........................Function that represents the insulin independent effect that glucose has on its own utilization (mg/min)
$f_3$..........................Function that represents the insulin dependent effect that glucose has on its own utilization
$f_4$..........................Function that represents the insulin dependent effect that insulin has on glucose utilization (mg/min)
$f_5$..........................Insulin dependent glucose production based on the time delay between their interaction (mg/min)
$G_B$..........................Blood glucose concentration (mg/dL)
$I$............................Current vector (A)
$I$............................Intensity of light
$i$..............................Summation index value
$I_{B}$..........................Blood insulin concentration (μU/mL)
$I_{IF}$..........................Interstitial fluid insulin concentration (μU/mL)
$J_e$..........................External current density (A/m$^2$)
$j$..............................Imaginary unit
$k_0$..........................Rotational proportionality constant at absorption maximum (°nm$^2$/(dm*g/mL))
$k_a$..........................Glucose absorption rate constant (L/min)
$k_e$..........................Glucose elimination rate constant (L/min)
$k_i$..........................Rotational proportionality constants (°nm$^2$/(dm*g/mL))
$L$..............................Light path length (dm or cm)
$L$..............................Inductance (mH)
$n_l$..........................Refractive index of left circularly polarized light
$n_R$..........................Refractive index of right circularly polarized light
$P$............................Set of residuals from a prediction data set (mg/dL)
$Q$.............................Cross product of $dr_0$ and $R$ (mm$^2$)
$R$.............................Vector directed from source point to the current element (mm)
$r$..............................Separation vector between origin and B-field point (mm)
$r_0$..........................Vector directed from the origin to the current element (mm)
$r_{Gin}$..........................Function that represents glucose infusion (mg/min)
$T$.............................Temperature (K)
$t$..............................Time (min)
$t_d$..........................Time delay between insulin infusion into blood and glucose production (min)
$t_i$..........................Time constant for insulin degradation in interstitial fluid (min)
$t_p$..........................Time constant for insulin degradation in blood (min)
$V$.............................Verdet constant (arcminute/G/cm)
$V_{Bg}$..........................Volume of total glucose distribution (dL)
$V_{Bl}$............................. Volume of insulin distribution in the blood (mL)
$V_{IF}$............................. Volume of insulin distribution in the interstitial fluid (mL)
$v$................................. Current velocity of a conductor (m/s)
$x$................................. Rectangular coordinate
$\mathbf{\hat{x}}$........................ Unit vector in the x direction
$x_1$............................... Insulin/glucose time delay variable 1 (μU)
$x_2$............................... Insulin/glucose time delay variable 2 (μU)
$x_3$............................... Insulin/glucose time delay variable 3 (μU)
$y$................................. Rectangular coordinate
$\mathbf{\hat{y}}$........................ Unit vector in the y direction
$z$................................. Rectangular coordinate
$\mathbf{\hat{z}}$........................ Unit vector in the z direction
$z_0$............................... Position of current loop
Chapter 1

Introduction

Diabetes mellitus, or what is more commonly referred to as diabetes, is a disease which is defined by the presence of high levels of glucose circulating in the bloodstream. More specifically, fasting plasma glucose (FPG) levels in non-diabetics are typically less than 100 mg/dL while those in diabetics will exceed 126 mg/dL and may be as high as 600 mg/dL after a meal. The area in between is considered pre-diabetes and occurs in those at high risk of developing the disease. In general, all forms of diabetes stem from one main cause: a lack of or resistance to the hormone known as insulin. Insulin is a product of the β-cells located in the islets of Langerhans within the pancreas. Its sole purpose is in the regulation of glucose metabolism in the blood. When this hormone is not present, or left inactive, glucose is unable to be removed from the blood and its concentration will rise to dangerous levels, leading to hyperglycemia (high blood glucose). When left uncontrolled, these high concentrations of glucose over time can lead to kidney disease, eye disease, nervous system disease, hypertension, high blood cholesterol, heart attack, and stroke. On the other hand, a very serious short-term complication known as hypoglycemia can occur as a side effect of unregulated glucose levels or due to the misuse of treatment medications. Hypoglycemia refers to low blood
glucose concentrations and can rapidly result in headache, confusion, dizziness, seizure, coma, and even death due to cerebral damage if left untreated.\textsuperscript{2,3} Therefore, it can be seen that consistent blood glucose regulation is a very important aspect in the health and stability of those with diabetes.

Diabetes is typically diagnosed as one of four clinical classifications.\textsuperscript{1} Type 1 diabetes is diagnosed when the body produces little to no insulin due to defects in the β-cells. Type 2 diabetes is the most common form of the disease and it is caused by an overall resistance to insulin. Gestational diabetes mellitus (GDM) occurs in women during pregnancy when certain hormones such as human placental lactogen (HPL) are produced that inhibit normal insulin receptor function. Lastly, the remaining clinical group is diagnosed based on other causes such as genetic defects, disease of the pancreas, or drug related interactions.

According to the Centers for Disease Control and Prevention (CDC), it is estimated that more than 25 million Americans (8.3\% of the national population) suffer from some form of diabetes and 79 million American adults show symptoms of pre-diabetes as of 2010.\textsuperscript{4} The disease was also a known contributing factor in more than 230,000 deaths in 2007 alone. This same report estimated the total cost of diabetes on the U.S. economy to be $174 billion in 2007 with the average medical expenses for diabetics to be twice that of the non-diagnosed population. In order to reduce this burden for tomorrow’s population, the Diabetes Prevention Program (DPP) has shown that the delay and prevention of the onset of type 2 diabetes can occur through interventional lifestyle changes by increasing physical activity and controlling diet.\textsuperscript{5} However, given today’s rate of diagnosis, proper disease control and patient compliance can significantly reduce
future healthcare costs stemming from the long-term health risks associated with diabetes as well as the dangerous short-term complications such as hypoglycemia.

Unfortunately, there is currently no cure for diabetes. However, several forms of treatment exist which can help mitigate the risk of future health complications. Such treatments include diet, exercise, glucose-lowering drugs, and insulin injections. In order to administer the proper treatment regimen at the correct time, leading to effective disease control, blood glucose concentrations must be closely monitored throughout the day, especially during times of high physical activity, after meals, and during trauma situations. Several studies have shown that this type of control can significantly decrease long-term health effects leading to a reduction in healthcare costs associated with the disease. Also, consistent monitoring can prevent the onset of serious complications stemming from hypoglycemia.

The conventional method of personalized blood glucose monitoring is done by taking small samples of blood and analyzing them in a hand-held digital glucose meter. Throughout the years, the required sample size to obtain approximate measurements has been reduced to as low as 0.3 µL and it can be done in as little as five seconds. However, it still requires blood to be repetitively extracted by fingertip lancing, resulting in pain, infection, and eventual nerve damage. Also, glucose meters operate based on an electrochemical reaction with an indicator such as glucose oxidase, requiring the purchase of costly disposable test strips which are used each time a measurement is taken (upwards of six times daily). Therefore, in order to promote easier, more patient-friendly methods of glucose monitoring, it is critical to investigate noninvasive and cost effective technology. A simple and affordable in-home glucose monitoring system paralleled with
the proper patient education in disease control can lead to better personalized healthcare, increasing compliance while decreasing long-term diabetes related complications.

The current study serves to investigate a well understood and promising polarimetric-based method of noninvasive continuous glucose detection. This approach operates based on the optical activity of glucose, utilizes a physical phenomenon known as the Faraday effect, and targets the correlation between glucose concentrations in the blood and the aqueous humor within the anterior chamber of the eye. The following section reviews the state of today’s research in the area of noninvasive glucose monitoring with a primary focus on polarimetry. The chapter is then concluded by introducing the specific objectives of the current study.

1.1 Literature Review

Prior to presenting an exhaustive investigation of the major benchmarks in the history of polarimetric glucose monitoring, it is also useful to understand the promising parallel tracks that other research groups have taken in an attempt to provide a necessary form of noninvasive detection. An extensive review article published in October of 2012 by Sandeep Vashist highlights 14 different techniques as well as some of the commercially available devices. These techniques include reverse iontophoresis, bioimpedance spectroscopy, thermal emission spectroscopy, absorption spectroscopy (both near- and mid-infrared), photoacoustic spectroscopy, Raman spectroscopy, ocular spectroscopy, fluorescence, ultrasound, electromagnetic sensing, temperature-regulated localized reflectance, optical coherence tomography (OCT), metabolic heat conformation, and polarimetry, each with its own distinct set of advantages and
disadvantages. An in-depth investigation of each of these methods would provide a collective wealth of knowledge spanning several different fields of physics, chemistry, and engineering, all of which are utilized for achieving a similar goal. However, to all intents and purposes of this review, each one can be briefly summarized.

The first proposed detection method, reverse iontophoresis, is based on a process that produces an electric current across the skin which draws glucose out from within the interstitial fluid to be measured via traditional sensors. It was first employed in a commercially available device known as the GlucoWatch® Biographer (Cygnus, Inc., Redwood City, CA.) in 1999. The disadvantages to this technique include obstruction by sweat and it is known to cause skin irritation. Another method, bioimpedance spectroscopy, measures changes in membrane potential of red blood cells (RBC) due to glucose fluctuations. The downsides to this measurement technique are that it is affected by moisture as well as overall RBC condition and it requires a resting equilibrium period that can be as long as an hour prior to taking initial readings. Then there is thermal emission spectroscopy which measures changes in infrared radiation emitted from the body caused by a changing glucose concentration. However, it is strongly influenced by normal physiological temperature fluctuations and body movements. The next technique, absorbance spectroscopy, is widely studied and observes the changes in light absorption and scattering by the skin due to variations in glucose concentration. Unfortunately, this method is strongly influenced by the presence of other biological molecules, various physiological parameters, and poor skin penetration of light. Another method, photoacoustic spectroscopy, measures glucose-induced changes in pressure vibrations of tissue caused by an increase in temperature when exposed to
pulsed laser light. This technology also suffers from a myriad of external and internal interferences. Following is Raman spectroscopy which is similar to absorption spectroscopy but it focuses on the measurement of scattered light of a higher wavelength than the incident source and requires long measurement periods. Next is ocular spectroscopy which utilizes a boronic acid derivative functionalized contact lens that captures glucose present in tears, causing a change in wavelength of reflected light when illuminated. This wavelength change can be correlated to overall glucose concentration. However, it has been shown that glucose in the blood is not strongly correlated to that in tears. The fluorescence technique, similar to ocular spectroscopy, has been used to measure glucose-related fluorescence signals in tissue as well as tears, maintaining limitations with light scattering, toxicity, and low concentration correlations. Another method, ultrasound, is used to extract glucose through the skin by increasing permittivity which is then measured with conventional technology. This method is affected by problems similar to reverse iontophoresis. Then there is electromagnetic sensing which measures changes in the dielectric properties of blood due to fluctuations in glucose concentration. However, this approach is strongly dependent on temperature and changes in other biological molecules. The next technique presented, temperature-regulated localized reflectance, measures glucose-related changes in the refractive index of tissue in response to temperature variations. This method is strongly influenced by health and other temperature related factors. Another popular approach is OCT which targets delays in backscattered light due to glucose. Unfortunately, it is strongly influenced by motion artifact and temperature. Next is metabolic heat conformation which measures various parameters such as heat, blood flow, hemoglobin, and
oxyhemoglobin as related to glucose concentration.\textsuperscript{35, 36} This method is also affected by various physiological and environmental conditions. However, it shows promise due to its combination of multiple glucose-related factors. Finally, a technique that has been vigorously studied is polarimetry. As previously mentioned, polarimetry utilizes the optical activity of glucose to measure rotations in polarized light which is dependent on the overall concentration and light path length. This approach provides the foundation for the work presented in this thesis and is reviewed in more detail in the following sections.

Given the individual pros and cons of each aforementioned detection method, a promising approach to achieving a physiologically feasible device may require the combination of two or more separate techniques such as absorption spectroscopy and polarimetry.\textsuperscript{37} However, application of such a method will require much more work in the optimization of each technique separately, as well as combined. Therefore, because the focus of this thesis is to further improve the realization of a polarimetric approach, a more extensive review of this technique is given in the following sections.

1.1.1 The Origins of Polarimetry in Glucose Sensing

The phenomenon of optical activity has been studied since the early 1800s when it was first introduced by Dominique F. J. Argo and Jean Baptiste Biot in various solid, liquid, and vaporous materials.\textsuperscript{38} A material is considered to be optically active if it causes the electric field of a beam of linearly polarized light to rotate upon traveling through a portion of the given material. This type of activity is known to occur with glucose and other chiral molecules when dissolved in a liquid medium and has been used as a measurement standard in the sugar industry since the early 1900s.\textsuperscript{39} Another similar
event was discovered in 1845 by Michael Faraday when he noticed a rotation in the plane of linearly polarized light traveling through a piece of glass in the presence of a strong magnetic field.\textsuperscript{38} This phenomenon has come to be known as Faraday rotation and it provided strong evidence for the correlation between electromagnetism and light. It was not until 1956 that Gillham combined the two and first proposed the idea of Faraday rotation to modulate light as a measurement tool in the sugar industry.\textsuperscript{40,41} The theoretical premise of these events is explained in detail in Chapter 2 of this thesis.

Before these methods could be applied to an effective noninvasive blood glucose detection system, a reliable sampling medium had to be established. Due to the high scattering and depolarization effects caused by normal biological tissue, its variations in thickness, its overall composition of other confounding components, as well as the need to measure sub-millidegree rotations, an optically clear path is desired for optimal polarimetric detection. In 1966, the foundation for this idea was provided by the findings of Pohjola who showed that the concentration of glucose in the blood could be correlated to the concentration of glucose in the aqueous humor within the anterior chamber of the eye by an age dependent ratio of about 70\%.\textsuperscript{42} Therefore, given the high scattering of skin and other tissue, the eye was targeted as an optically clear path for observing polarimetric measurements. This idea was brought to life in 1979 by March \textit{et al.} who showed that the concentration of glucose in the aqueous humor could be detected by measuring optical rotation and described methods of noninvasive glucose detection.\textsuperscript{43}

In 1982, the optical activity of glucose, the Faraday effect, and the glucose content of the aqueous humor were combined and the first noninvasive polarimetric physiological glucose sensor which utilized a null-point feedback mechanism was
proposed by March, Rabinovitch, and Adams.44,45 These studies demonstrated the ability to measure optical rotations in glucose concentrations of 20 mg/dl (corresponding to 1.3 millidegrees of rotation) with an accuracy of 0.1 millidegrees. They also showed that the correlation of rapidly changing blood glucose levels with changes in the aqueous humor glucose concentration in rabbit models was on the order of 10 to 20 minutes. Within their work, the authors proposed the miniaturization of the components into a contact lens with a light path laterally through the anterior chamber of the eye that would be coupled to an insulin pump for the continuous regulation of blood glucose in diabetics. This initial study has laid the groundwork for a significant amount of research throughout the 1990s and 2000s devoted to noninvasive polarimetric glucose detection which proceeds to continuously evolve with new optical technology.

1.1.2 Tackling the Challenges of Polarimetric Sensing

Over time, a number of advancements have been made toward increasing system accuracy and stability while addressing the major critiques of polarimetric glucose sensing. In 1992, Coté et al. proposed a system which utilized a rotating polarizer rather than a Faraday modulator in a phase measurement technique for measuring rotational changes.46 Their system improved the signal-to-noise ratio and provided millidegree sensitivity. Around the same time, Goetz et al. demonstrated an increase in sensitivity of a Faraday-based system utilizing a closed-loop feedback controller which achieved microdegree sensitivity.47 This work was further stabilized by Cameron et al. in 1997 which implemented a digital closed-loop controller, producing standard errors of prediction of 8.84 mg/dL in glucose-doped water and 27.47 mg/dL in glucose-doped
bovine aqueous humor.\textsuperscript{48} Similar results were also reported for glucose measurements in cell culture media.\textsuperscript{49} Also, as an extension of the use of lasers in optical polarimetry, Baba \textit{et al.} proposed the use of a quarter-wave plate to help reduce inherent noise in the polarization state of lasers.\textsuperscript{50}

It should be noted that other groups have reported on different methods of optical polarimetry for continuous noninvasive glucose detection through the aqueous humor. Such methods include an optical heterodyne polarimeter by Chou \textit{et al.} in 1998,\textsuperscript{51} measurement of the Brewster reflection of circularly polarized light by Ansari \textit{et al.} in 2004,\textsuperscript{52} and a modulated system that measures light reflected off of the intraocular lens by Rawer \textit{et al.} in 2004.\textsuperscript{53} However, Chou’s work resulted in inconsistent time delay properties of the aqueous humor and the latter two systems propose a light path that reflects off of the lens of the eye which has the potential for retinal damage. Therefore, it is Cameron’s Faraday-based system\textsuperscript{48} which utilizes a lateral path across the eye that provides the basis for the work presented in the current thesis.

One can see that each polarimetric method presented thus far employs different methods for achieving consistent, stable, and continuous measurements of glucose noninvasively through the aqueous humor of the eye. However, they also share similar obstacles in the implementation of an actual \textit{in vivo} system. More specifically, there have been recent studies to address issues involving the secondary optical confounders found in normal aqueous humor, optical rotations due to the cornea, temperature effects, pH effects, glucose transport time delays, subconscious motion artifact of the eye, and time-varying corneal birefringence.
Given the composition of the aqueous humor, it was once thought that other optically active components would contribute to discrepancies in measurements of optical rotation. However, a study by Gough in 1982 revealed that in a combination of over 20 biological molecules present in bovine aqueous humor, only glucose and the total proteins made a significant contribution to overall optical activity.\(^{54}\) These contributions are likely due to their high concentration within the aqueous humor and their large specific rotations. It should be noted that these proteins are typically found in lower concentrations in human aqueous humor and they contribute to rotations that oppose glucose. This was also supported by the work of Rabinovitch et al. which showed that the physiological concentrations of lactic acid, ascorbic acid, and amino acids in the rhesus monkey contribute to rotational errors of less than 10% when compared to glucose.\(^{44}\) Overall, these effects can be deemed negligible if their concentrations are assumed to remain constant in comparison to glucose.

A slightly different approach was taken by King et al. in 1994 in which they proposed using a single Pockels cell for modulation and signal feedback.\(^{55}\) This study demonstrated the use of superposition and multispectral analysis in a two wavelength system that was capable of measuring hyperglycemic levels of glucose \textit{in vitro} in the presence of ascorbic acid and albumin (known to be a primary component of the protein composition in the aqueous humor). This was possible given the characteristic way that each molecule rotates light at different wavelengths. It was also suspected that the same theory could be applied to compensate for corneal rotations and motion artifact as presented by Coté et al. in 1998.\(^{56}\) A study by Cameron et al. in 1999 utilized a Faraday-based multispectral polarimetric system and demonstrated successful glucose prediction
in vitro in the presence of albumin at physiological levels.\textsuperscript{57} Therefore, it can be seen that the contribution of other optical confounders is low and multispectral systems can be used to minimize their effect altogether.

Other issues that have been addressed in optical polarimetry for noninvasive physiological glucose detection involve temperature, pH, and the effects of glucose transport time delay. A study by Baba \textit{et al.} in 2002 revealed that fluctuations in temperature and pH can have an effect on rotations of light from glucose for a given concentration.\textsuperscript{58} However, the values reported for specific rotation are significantly higher than typical accepted values around room temperature, questioning the validity of these results.\textsuperscript{59} Regardless, it was also noted that these effects were negligible within the range of physiological temperature and pH values, especially at higher wavelengths.

To address the issue of the time delay associated with glucose transport from the blood into the aqueous humor, the initial work by March \textit{et al.} revealed it to be on the order of 10 to 20 minutes.\textsuperscript{45} However, only two samples were collected over a spread of two hours, indicating a lack of supporting data. A later \textit{in vivo} rabbit study by Chou \textit{et al.} produced results suggesting that this delay was as high 30 minutes.\textsuperscript{51} It is known that delays of this magnitude would deem polarimetric detection unsuitable for effective insulin treatment and should remain less than 10 to 15 minutes as reported by Sorensen.\textsuperscript{60} However, the study by Chou presents results from three rabbits but only reports time delay data on two of the three. Also, their study does not verify that the system is in fact tracking glucose rather than other optically active components or overall signal drift. Such discrepancies in the literature were put to rest upon a study by Cameron \textit{et al.} in 2001 which resulted in an average time delay of glucose transport into the aqueous humor.
to be less than 5 minutes in New Zealand White (NZW) rabbits.\textsuperscript{61} This study characterized a glucose response protocol using triplicate samples from two rabbits and followed up with the experimental protocol using five rabbits over a period of several weeks in order to minimize effects from invasive aqueous humor sampling. Although prediction errors between data points and the model used to calculate the time delay are not given, a consistency between data sets provides overall support of their findings. Furthermore, a more recent \textit{in vivo} study by Purvinis \textit{et al.} in 2011 also supported these results based on seven rabbits with individual delays between 2.9 and 5.4 minutes.\textsuperscript{62} Therefore, because temperature, pH, and transport time delay are considered negligible effects, it can be seen that the eye provides a valid sensing medium for monitoring physiological blood glucose levels.

Although, physiologically, the eye may be an ideal sensing site for polarimetry, time-varying corneal birefringence caused by motion artifact is arguably the most significant obstacle encountered \textit{in vivo} during polarimetric glucose detection, as seen in the literature. An \textit{in vivo} rabbit study presented by Baba \textit{et al.} revealed that this motion artifact is caused primarily by respiration and, to a lesser extent, the cardiac cycle.\textsuperscript{58} In addition, this group modeled the effect of varying corneal birefringence and characterized it with experimental \textit{in vivo} results to be used in the design of birefringence resistant systems. To compensate for this problem, Wan \textit{et al.} formulated a model and combined the multispectral approach with a closed-loop system that was able to predict physiological glucose concentrations with an error of 20 mg/dL in the presence of varying birefringence in 2005.\textsuperscript{63} This was a great achievement in comparison to the errors seen in single-wavelength systems with varying birefringence which were shown to be
over 1000 mg/dL. However, it is unclear in their experimental design at what frequency the birefringence was varied during sample cell translation meaning that the results could be different in systems displaying physiological time-varying motion. Also, the authors claim to use larger birefringence magnitudes than what is reported physiologically, meaning that the comparative data set is not representative of what would be expected to occur in vivo and the error of the single wavelength system appears inflated. Furthermore, the comparative model used was based on that which was derived by Cameron et al. in 1997 which does not incorporate birefringence into the Jones model. Therefore, the authors arbitrarily assigned a non-time-dependent rotation into the model in order to represent birefringence. For this to be truly representative, a new Jones model must be derived. Finally, the system was calibrated after collecting data meaning that real-time glucose measurements were not collected which is critical for a practical in vivo device.

In 2006, Cameron and Anumula reported on a dual-closed-loop birefringence compensated glucose sensing polarimeter with standard errors in calibration (SEC) as low as 13.8 mg/dL in the presence of varying birefringence and 23.4 mg/dL using a corneal based eye phantom. This system introduced a second feedback loop prior to the glucose sensing path which consisted of a Stokes analyzer for measuring circular polarization caused by sample birefringence and a variable retarder capable of compensating for varying sample birefringence. Although these results appear promising, the SEC was nearly twice as high when excised cornea was used meaning that actual uncontrolled physiological errors may be even higher. Also, the calibration model was not validated against a separate data set and no standard errors of prediction (SEP) were presented which is generally the accepted evaluation method. Furthermore, the corneal
tissue used was not living and reported as translucent, it was not indicated how old the samples were or what portion of the eye they were extracted from, and the time varying movements were performed in one translational direction in a controlled manner. Therefore, the system must be tested in vivo or using a more representative eye model in order to demonstrate better feasibility and the results should be validated against multiple data sets. However, their technology is currently patented and licensed in the United States, Japan, and several European countries and shows promise as a way to reduce the effects of corneal birefringence.

In 2009, Malik and Coté took the multispectral approach another step further by introducing a real-time, closed-loop, dual-wavelength system capable of predicting physiological glucose concentrations in vitro in the presence of varying birefringence with an error of prediction on the order of 12.8 mg/dL. However, the authors utilized a theoretical Jones model that was not representative of their polarimetric system or the birefringence that was implemented. They also, indicate using birefringence that was likely larger than what would be seen in vivo, making the controlled single-wavelength data set appear much worse than what might be expected. Furthermore, glucose predictions made with the combined wavelength system were not actually measured in real-time, but rather fit to a multiple-linear regression model after the data was collected. Therefore, the study does not clearly represent the need for multi-wavelength systems in vivo and more work must be done to show its feasibility with an actual eye model and real-time glucose prediction. As a follow up to this work, Malik and Coté introduced a computational eye model which includes the spatially varying birefringence properties of the cornea and identifies the optimal light path through the anterior chamber of the eye to
be midway between the apex and limbus. This model indicates the need to reduce the effects of corneal birefringence and proposes a light path through the eye which can be used to best achieve this goal.

A slightly different approach was taken by Winkler et al. in 2011 to potentially eliminate the burden of birefringence faced by a traditional polarimetric-based system. This group proposed the use of Sagnac interferometry, rather than crossed polarizers, which is insensitive to linear birefringence and the state of incident polarization. Such a system detects changes in phase caused by optically active molecules, eliminating the need for direct rotational measurements and their inherent interferences. Their initial results appear promising but much more work is required to study the effects of diattenuation at the corneal surface and an overall more robust system must be developed.

Overall, it can be seen that many of the initial problems in polarimetric glucose detection have been reduced or viable solutions have been proposed. However, corneal birefringence remains to be a significant challenge in the development of a practical glucose detection system. The next few sections introduce other methods of polarimetric glucose detection as well as some of the recent in vivo application data.

### 1.1.3 Polarimetry in Turbid Media

Although the optically clear aqueous humor has been the favored sensing medium for the majority of this research, more recent advancements with increasingly sensitive devices have shown that the scattering effects of biological tissues such as the skin and other turbid media do not completely depolarize a polarimetric signal and glucose can be detected through complex analysis methods. In 1998, Mehruşoğu et al. presented
results on the potential use of diffuse reflectance polarimetry to detect changes in polarization pattern images due to concentration differences of optically active species in tissue-mimicking phantoms. Shortly after in 2000, Vitkin and Hoskinson proposed a technique utilizing a photoelastic modulator (PEM) and synchronous detection for quantifying the net degree of polarization (DOP) and optical rotation of linearly polarized light that has been multiply scattered in turbid media containing glucose. However, a considerable improvement in sensitivity of this method was required for physiological feasibility. This work was followed up by Studinski and Vitkin who proposed a Mueller matrix model capable of predicting the DOP and optical rotation in backscattered light in turbid media. Their results demonstrated model correlation and polarization preservation in both tissue-mimicking phantoms and biological tissue (ex vivo and in vivo). They also provided evidence on the effect of tissue optical properties on the polarization characteristics of multiply scattered light. A few years later in 2002, Wang et al. demonstrated that the Mueller matrix patterns of diffusely backscattered and forward-scattered light could be correlated to glucose concentrations in turbid media by means of a Monte Carlo model. They also presented a single-scattering model which was not supported in the diffuse regime. Their work was based on the agreement between experimental and Monte Carlo simulation results presented earlier by Cameron, Raković, et al. This research showed symmetry in backscattered Mueller matrices and demonstrated the ability to predict the backscattered intensity patterns seen when polarized light was cast on turbid media, based on the theory of incoherent light scattering by spherical particles. In 2007, Cameron and Li quantitatively demonstrated a strong correlation in glucose prediction in highly scattering media using diffuse
reflectance polarization imaging and partial least squares (PLS) regression.\textsuperscript{81} This group also used principal component analysis (PCA) to show that the two main glucose-based factors causing changes in the polarimetric images were the overall scattering coefficient and the optical activity. It was suspected that predictive error in these experiments could be reduced with more advanced Mueller matrix measurement techniques.\textsuperscript{82}

As shown in the book chapters by Wood, Ghosh, \textit{et al.}\textsuperscript{70, 71} and a review paper by Ghosh and Vitkin\textsuperscript{72}, there has been a recent research boom in the study and modeling of polarimetric properties within highly scattering media such as biological tissue for use in various medical diagnostic and interventional procedures such as noninvasive glucose detection. It can be seen that a significant contributor to the feasibility of such a system is in the polar decomposition of lumped Mueller matrices (\textit{i.e.,} for extraction of the optical rotation due to glucose) as first proposed by Lu and Chipman in 1996.\textsuperscript{83} Since then, this technique has been utilized frequently in several tissue polarization studies such as those by Manhas \textit{et al.} in 2006\textsuperscript{84} and Ghosh \textit{et al.} in 2008-2012\textsuperscript{85-88} to extract the optical rotation due to glucose from other confounding components such as linear diattenuation and linear retardance in both forward and backscattered light. However, to make this technology a prime candidate for noninvasive glucose monitoring, much more work is required to achieve system sensitivity in the physiological range, to implement a multispectral system for overall specificity, and to validate actual \textit{in vivo} applications.\textsuperscript{89} Therefore, polarimetric sensing through an optically clear medium such as the aqueous humor is a more favorable candidate for current device development. This is further supported in the following section which highlights the \textit{in vivo} validations of this technology.
1.1.4 The *In Vivo* Application of Polarimetric Sensing

Since it was initially introduced in 1982,\textsuperscript{44,45} significant accomplishments have been made in the field of noninvasive polarimetric glucose monitoring. Those which were once thought to be unavoidable setbacks have been overcome or dispelled through experimentation and new developments in technology. It can be seen that this approach provides a truly viable means for accurately facilitating diabetes treatments. This is even more evident given the state of the most recent research in the applications of polarimetric systems *in vivo* as well as the steps being taken toward commercialization of such a device. In 1998, Chou *et al.* reported the first full scale *in vivo* rabbit experiment utilizing an optical heterodyne polarimeter.\textsuperscript{51} However, this study was slightly premature to the major developments in combating corneal birefringence and their results on time delay have since been disproven, as previously mentioned. In the following year, Cameron *et al.* reported on the design of an index matching eye-coupling device capable of detecting a polarimetric signal from a single-wavelength Faraday-based system *in vivo*, through the eye of a NZW rabbit.\textsuperscript{57} Following this study, Webb and Cameron reported on successful continuous *in vivo* polarimetric glucose tracking in NZW rabbits in 2009 which was followed up by a more expansive study in 2011 using a similar eye coupling mechanism and a single-wavelength polarimeter.\textsuperscript{90,91} Their results indicate a SEP of 18.6 mg/dL with an $R^2$ value of 0.95 and 100% of the data points falling within the A and B regions of a Clarke error grid analysis (CEGA). However, this system utilized an open-loop approach and calibration was carried out after the data was collected meaning that the glucose concentration was not predicted in real-time. Also, no birefringence compensation method was utilized and the time delay of glucose diffusion into the
aqueous humor was not factored into the experiment which could have been used to
further decrease the error values.

In a second study in 2011, Purvinis et al. reported on an in vivo study of multiple
NZW rabbits that implemented a single-wavelength Faraday-based optical polarimeter
for noninvasive glucose detection through the aqueous humor. In this work, the
polarized signal was propagated along the fast axis of the eye in order to preserve the
polarization effects of glucose within the aqueous humor. Also, data processing and a
contact eye-coupling device were used to minimize the effects of corneal birefringence
caued by motion artifact. The results of this study show the successful tracking of blood
sugar concentration changes, even in the absence of time delay correction. The
published CEGA utilized 41 data points with 93% falling in zone A and 7% in zone B.
Although this study further provided groundbreaking results demonstrating the feasibility
of a polarimetric glucose monitor, it was also suggested that future in vivo studies need to
incorporate active birefringence compensation as well as a non-contact eye-coupling
mechanism. Also, an open-loop system was used and glucose concentrations were not
measured directly in real time meaning that follow-up studies should investigate real-time
closed-loop control for stable glucose monitoring. This study transitions into the most
recent work published, at the time of this thesis, by Pirnstill et al. in September of 2012. Their group was the first to demonstrate the use of a dual-wavelength polarimeter in vivo
for noninvasive glucose detection in NZW rabbits with 100% of their data points falling
in zone A upon CEGA and an overall standard error of 11.7 mg/dL. However, it was
assumed that corneal birefringence would not cause significant effects in the ellipticity of
the polarized signal and only corneal rotations were implemented into the Jones model.
Furthermore, although a closed-loop system was used, post-processing was done on the data in order to calculate glucose concentrations, calibration models were formed individually for each rabbit, and no continuous data was reported. They also suggest the need for developing a better eye-coupling mechanism in order to create a more robust calibration model. Therefore, it can be seen that the results could have been idealized based on the point selection of the data, given that no continuous measurements were used. Also, because a contact eye-coupling mechanism was used, it is unclear if the multispectral system was fully tested to the limit of normal corneal birefringence. However, these results successfully indicate that the dual-wavelength approach can improve prediction accuracy, bringing this technology another step closer toward eventual commercialization for improved treatment of diabetics around the world. Also, it can be seen that polarimetric device validation and optimization would largely benefit from a more controllable and repeatable testing environment that can recreate physiological conditions while bypassing these expensive and time consuming in vivo conditions.

1.2 The Current Objectives

One can easily see that diabetes is a costly and lethal disease if not handled properly. Although work must done to find ways to prevent type 2 diabetes, the current onset of this condition is on the rise and it is known that there is no cure for type 1 diabetes. Therefore, innovative ways to supplement treatment and increase patient compliance is a necessity. The primary bottleneck to achieving these goals has been identified as the current invasive means of monitoring glucose in the blood which is
required several times throughout the day for the proper administration of treatment. As shown in the literature review, a significant amount of work has gone into the study and development of various methods of detecting the physiological concentrations of glucose noninvasively over the past 30 years. Given the state of today’s research, optical polarimetry has been a leading technology in the field of biomedical sensing. It has been shown that glucose in the aqueous humor within the anterior chamber of the eye correlates well to that in the blood in a matter of minutes, thus providing an optically clear sensing site for minimal signal depolarization. This discovery has paved the way for other advancements such as closed-loop controllers, digital signal processing, and multispectral applications which have proven successful for noise reduction, system stability, and elimination of interference from corneal birefringence and other optical confounders. A culmination of this research has occurred over the past few years with the successful demonstration of polarimetric glucose detection \textit{in vivo} in NZW rabbit models.\textsuperscript{62,92} Therefore, it can be seen that commercialization of such a device will soon be of interest. However, to date, little work has been focused on the development of a commercially viable and manufacturable product. Also, a testing method which could mimic physiological conditions should be developed in order to forgo the need for repetitive animal testing in future research and development and facilitate more controllable experimentation.

As will be further detailed in the following chapters, the primary optical components that are required for physiological polarimetric glucose monitoring are the Faraday modulator for signal detection and the Faraday compensator for continuous null-point feedback control. All of the closed-loop Faraday-based designs presented thus far
have utilized these as two separate components which require two separate optical rods, typically composed of terbium gallium garnet (TGG) or terbium doped glass (TDG), wound in custom inductive coils. Three major drawbacks to this approach are the significant cost of the optical materials, the difficulty of repetitively producing quality optical materials, and the bulky size of the custom inductive coils. In order to produce an economical and competitive device that can be made available as an alternative to today’s invasive glucose detection methods for diabetic patients, the end product must be made affordable as well as miniaturized into a handheld device.

Furthermore, as the current state of research has demonstrated successful \textit{in vivo} noninvasive polarimetric glucose sensing, a more practical testing method is required in order to further advance research and development and move the field forward toward a commercial product. It is known that animal testing can be very expensive and time consuming and is often difficult to repetitively control. Also, it is unnecessary to repeat these tests with each and every new development. Therefore, future research would highly benefit from the use of a more controllable testing environment such that \textit{in vivo} conditions could be easily replicated based on various initial parameters. A study by Malik \textit{et al.} in 2012 utilized an artificial eye anterior chamber as a means to represent the \textit{in vivo} sensing site, allowing for the control of time-varying corneal birefringence.\textsuperscript{93} However, this model can be further expanded by adding a continuous glucose flow system for controlling physiological profiles based on a wide range of desired parameters. Together, these types of systems can be coupled to the feedback loop of a polarimeter in order to track glucose concentrations continuously in real time, similar to the conditions tested in animal studies. Therefore, robust testing systems can facilitate
new developments in polarimetric technology, forgo repetitive animal testing, and be validated quickly and easily in a more controlled manner while speeding up the development of a commercially available device.

The overall objectives of the current research were threefold. First, a finite element model (FEM) and a customizable prototype were developed and validated to be used in the construction of custom Faraday rotators for various applications involving the sensing of optically active materials, such as noninvasive glucose sensing, cell culture bioreactors, or chemical reaction monitoring, as well as in optical communications. Second, it was shown that separate Faraday components are not needed and an integrated Faraday modulator and compensator (IFMC) which utilizes a single TDG rod and off-the-shelf inductors was designed and optimized based on the FEM as a step toward overall size and cost reduction in the development of a noninvasive polarimetric glucose monitor. Third, a continuous and programmable flow system was developed, integrated with the polarimetric glucose detection system, and tested against actual physiological glucose profiles in the creation of a controllable in vitro testing model for real-time, continuous sensing. Overall, the work presented in this thesis provides a transition from the theoretical “proof-of-concept,” seen in the majority of previous research, toward the development of a user-friendly, economically sensible end-product in hopes of bringing this technology another step closer to the hands of the diagnosed and, ultimately, providing a better treatment methodology. Also, the development of a controllable glucose flow system will promote an accelerated advancement in technology by allowing future developments to be validated and optimized quickly without the need for repetitive and expensive animal testing. Finally, the customizable prototype along with the FEM
can facilitate optical polarimetry in a wide range of applications with different rotational requirements which have previously required custom designed and programmed systems. The theory behind each of these objectives is further explained in Chapter 2.
Chapter 2

Theory

As evident in Chapter 1, polarimetry has been the foundation for a significant amount of research toward the development of a noninvasive and continuous physiological glucose sensor. Many other technologies have also been applied to achieve a similar goal but polarimetry remains as a primary candidate for a commercially viable option in the near future. Although these systems have evolved over the past 30 years in order to address different challenges or implement new technology, each one still operates based on the same fundamental principles: the optical activity of glucose as well as the manipulation and measurement of polarized light. More specifically, as a beam of linearly polarized light travels across a sample containing glucose, the plane of polarization rotates proportionally to the path length of the sample as well as the concentration of glucose. However, a change in sample concentration of 10 mg/dL will result in a rotational shift of 0.632 millidegrees in a 543.5 nm linearly polarized beam of light when passed through a 1 cm sample (which is representative of the aqueous humor of the eye). Therefore, very precise techniques are required in order to measure such a small change. One way to achieve this precision is to implement an optical modulator and closed-loop feedback system.\textsuperscript{48} The current research proposes the design of an IFMC
which directly depends on the surrounding magnetic field properties. Given the complex nature of magnetic fields in three dimensional space, the optimization of such a system largely benefits from the use of computer modeling through FEM analysis of electromagnetic (EM) fields. Also, as a way to provide a more controllable testing environment, a continuous glucose flow system was designed that is capable of recreating real physiological glucose profiles based on mathematical models or clinical data. The theory behind each of these phenomena, namely glucose as a target molecule, optical polarimetry and the Faraday effect, and FEM of EM fields, is given in the following sections.

2.1 Glucose as a Target Molecule

The primary characteristic of glucose which allows for polarimetric sensing is known as optical activity. This material property is inherent to many complex biological molecules based on how they are formed in nature. As will be explained in the following sections, the rotations caused by different molecules are unique in how they relate to a spectrum of different wavelengths of light. However, prior to explaining the theory behind this phenomenon, it is useful to understand the application to which continuous polarimetry was applied in the current thesis. In other words, the physiological concentration profiles which a noninvasive polarimetric glucose sensor is expected to track over time must be explored such that the appropriate design requirements can be met. One way of viewing these profiles in terms of an experimental design is by utilizing a mathematical model which allows for easily controllable and quantifiable testing. The application of such a model provides a means to repeatedly control the testing
environment in order to validate the design of an IFMC as a continuous glucose monitor. Therefore, the following sections will provide the model that was used in the design of a glucose flow system for experimental validation before leading into the theory of optical activity and its application to optical polarimetry.

2.1.1 Continuous Glucose Monitoring

As mentioned in Chapter 1, an important aspect of advancing noninvasive polarimetric glucose sensing at a more rapid pace is in the ability to provide a more controllable testing environment. Given that the current state of research has successfully shown in vivo sensing capabilities, it is important to provide new experimental methods which can produce similar results without the need to repeat costly and time consuming animal studies with each new development in technology. Therefore, a system that is capable of reproducing the normal physiological glucose response would prove to be useful in future research and development of polarimetric systems. However, in order to develop a system capable of producing such controlled profiles, it is important to understand the interaction of glucose with other compounds in the body, namely insulin, such that the environment can be successfully recreated. A useful way of providing this understanding is through mathematical modeling. Several such models in the form of ordinary differential equations (ODE), delay differential equations (DDE), integro-differential equations (IDE), and partial differential equations (PDE) have been presented as a means to mathematically explain the physiological interactions as shown in a review paper by Makroglou et al. in 2006.94
Typically, non-diabetics are expected to maintain their blood glucose levels between 80-120 mg/dL which is controlled by the interactions of glucose with insulin.\textsuperscript{95} This control is achieved by a periodic ultradian oscillation of insulin on the order of 100 to 150 minutes. A set of more rapid oscillations may also play a role but on a much lower scale (< 1 mg/dL of glucose regulation over 10 to 15 minutes). A model presented by Sturis \textit{et al.} in 1991 provides a mechanism for these ultradian interactions based on four negative feedback loops.\textsuperscript{96} Each feedback mechanism can be explained by the effects from a single increase in blood glucose. For example, the effects from such an increase lead to an increase in insulin secretion which inhibits glucose production, an increase in insulin secretion which increases glucose utilization, a direct decrease in glucose production, and a direct increase in glucose utilization. The model proposed by Sturis \textit{et al.}\textsuperscript{96} and refined by Fournier\textsuperscript{95} is given as the following six state system:

\begin{align}
V_{Bi} \frac{dI_B}{dt} &= f_1(G_B) - E \cdot (I_B - I_{IF}) - \frac{I_B V_{Bi}}{t_p}, \\
\frac{dI_{IF}}{dt} &= E \cdot \left( \frac{I_B}{V_{IF}} - \frac{I_{IF}}{V_{IF}} \right) - \frac{I_{IF}}{t_i}, \\
V_{Bg} \frac{dG_B}{dt} &= r_{gin}(t) - f_2(G_B) - f_3(G_B) \cdot f_4(I_{IF}) + f_5(x_3), \\
\frac{dx_1}{dt} &= \frac{3}{t_d} (I_B V_{Bi} - x_1), \\
\frac{dx_2}{dt} &= \frac{3}{t_d} (x_1 - x_2), \\
\frac{dx_3}{dt} &= \frac{3}{t_d} (x_2 - x_3),
\end{align}

and

(2.6)
In this model, $G_B$ represents glucose in the blood (mg/dL), $I_B$ represents insulin in the blood (µU/mL), $I_{IF}$ represents insulin in the interstitial fluid (µU/mL), $t$ represents time (minutes), and $x_1$, $x_2$, and $x_3$ together represent the time delay between the introduction of insulin into the blood and its effect on glucose production (µU). The reason for the separation of insulin concentrations in the blood versus in the interstitial fluid is to represent the time delay associated with the equilibrium of insulin from blood into the tissue where it acts. Both of these delays together play an important role on explaining the normal ultradian oscillations observed in physiological glucose profiles.

The secondary equations shown in the model each represent a separate process within the system. $^96$ Function $f_1(G_B)$ represents the infusion of insulin into the blood, such as by islet cell secretion, function $f_2(G_B)$ represents the insulin independent effect that glucose has on its own utilization, function $f_3(G_B)$ represents insulin dependent glucose utilization as a function of glucose, function $f_4(I_{IF})$ represents the insulin dependent glucose utilization as a function of insulin, and function $f_5(x_3)$ represents the insulin dependent glucose production based on the time delay between their interaction. It should be noted that although function $f_1(G_B)$ in this case represents insulin secretion from normal islet cells in non-diabetics, it could also be used to represent insulin infusion from external sources such as bioartificial organs or external insulin pumps. The final function, $r_{Gin}(t)$, is used to represent glucose infusion by absorption through the gastrointestinal tract after a meal or an oral glucose tolerance test (OGTT). $^95$ However, it can also be used to represent glucose infusion directly into the blood in the case of an intravenous glucose tolerance test (IVGTT). Each of these functions are given below as
The set of parameters used in the model equations are defined in Table 2.1 below.\(^96\)

**Table 2.1:** Definition of parameters used in the insulin/glucose model.\(^96\)

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Definition</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>(k_e)</td>
<td>Glucose elimination rate constant</td>
<td>0.0083 L/min</td>
</tr>
<tr>
<td>(k_a)</td>
<td>Glucose absorption rate constant</td>
<td>0.042 L/min</td>
</tr>
<tr>
<td>(D)</td>
<td>Total ingested glucose</td>
<td>50,000 mg</td>
</tr>
<tr>
<td>(V_{Bi})</td>
<td>Volume of insulin distribution in blood</td>
<td>3000 mL</td>
</tr>
<tr>
<td>(V_{IF})</td>
<td>Volume of insulin distribution in interstitial fluid</td>
<td>11,000 mL</td>
</tr>
<tr>
<td>(V_{Bg})</td>
<td>Volume of total glucose distribution</td>
<td>100 dL</td>
</tr>
<tr>
<td>(E)</td>
<td>Rate constant of insulin exchange between blood and interstitial fluid</td>
<td>200 mL/min</td>
</tr>
<tr>
<td>(t_p)</td>
<td>Time constant for insulin degradation in blood</td>
<td>6 min</td>
</tr>
<tr>
<td>(t_i)</td>
<td>Time constant for insulin degradation in interstitial fluid</td>
<td>100 min</td>
</tr>
<tr>
<td>(t_d)</td>
<td>Time delay between insulin infusion into blood and glucose production</td>
<td>36 min</td>
</tr>
</tbody>
</table>

Overall, it can be seen that the mathematical model provides a controllable and
repeatable glucose profile which can be used for testing and validating noninvasive glucose sensors based on various initial conditions and controlling functions. The model demonstrates the interactions and oscillations between glucose and insulin in non-diabetics which can also be modified for displaying various diabetic treatment interactions, particularly for type 1 diabetes. The primary benefit of using such models in device validation is that they show the characteristic physiological changes in glucose concentrations over time that a continuous sensor would have to be able to track, providing a baseline understanding of the design requirements. Also, different parameters and input functions can be controlled in order to provide the desired experimental environment when combined with a fully programmable glucose flow system. An excellent example of a normal, non-diabetic response of blood glucose following a 50 g OGTT with initial conditions for blood glucose, blood insulin, and interstitial insulin of 80 mg/dL, 10 µU/mL, and 6.45 µU/mL, respectively, can be seen in Figure 2-1.95 The model was solved using XPPAUT, an open source numerical integration software package which originated from a group at the University of Pittsburgh (Pittsburgh, PA). The data was then exported and plotted in MATLAB (MathWorks, Natick, MA). It should be noted that the oscillation appears damped because the glucose was ingested as a single does. During a constant infusion of glucose, the oscillation would be maintained over time provided that the bifurcation parameters produced a stable limit cycle.

2.1.2 Optical Activity

Now that it has been shown what a continuous polarimetric glucose monitor must be capable of predicting over time, it is necessary to understand the nature of how
polarimetry is applied as a sensing method for molecules such as glucose. The primary characteristic of glucose that allows for its polarimetric detection is known as optical activity. As mentioned in Chapter 1, optical activity is a phenomenon which causes the E-field of a linear plane wave to rotate around an axis parallel to the direction of travel through an optically active material. Overall, optical activity can be explained by a material property known as circular birefringence. Such materials possess a separate refractive index that affects right ($n_R$) and left ($n_L$) circularly polarized light. Given that linear polarization is the superposition of two opposing circular components of equal magnitude, it can be seen that these components will become out of phase when
travelling through an optically active material, resulting in a rotated plane wave as shown in Figure 2-2. The amount of rotation ($\alpha$) in degrees can be quantified based on the indices of refraction, material path length ($L$) in centimeters, and vacuum wavelength ($\lambda_0$) in centimeters as shown in Equation 2.13,

$$\alpha = \frac{180^\circ L}{\lambda_0} (n_L - n_R).$$

(2.13)
The cause for separation in refractive indices is due to the handedness of electron orientation of an optically active molecule. When the E-field of a polarized wave passes over one such molecule, time-varying electric and magnetic dipole moments form. The dipoles then re-emit a scattered E-field which combines with the original incident wave. The resultant wave rotates around the axis of travel depending on the sense of the dipole moments and the orientation of electrons in the molecule.

The concept of optical activity also holds true for glucose and other chiral molecules when dissolved in solution based on their overall concentration. The term chirality is derived from the Greek word for hand which is indicative of its meaning. In general, chirality refers to the right or left handedness of an object and its inability to be superimposed on a mirror imagine of itself. On the molecular level, this refers to two molecules of a single chemical composition which differ only in the orientation of one or more chemical bonds around a central atom known as a chiral center. Such a difference in orientation produces two molecules that are exact mirror images of one another, known as enantiomers. Enantiomers are chemically identical compounds except in how they react with other chiral molecules and in their ability to rotate linearly polarized light, both of which are based on the chirality of each orientation. Due to the complexity of many organic biological molecules, chirality is a common occurrence in nature. Furthermore, since molecular orientation is a direct product of how a particular compound is formed, biological molecules are naturally found in a specific orientation. This is supported by the ability of humans to distinguish between the smell of spearmint versus caraway, which are the R and S enantiomers of carvone, respectively, suggesting that various olfactory receptors also possess chirality.
The R and S nomenclature is commonly used to distinguish between enantiomer compounds. The substituents surrounding a chiral center are designated with a priority based on atomic number and assigned either an R (for rectus, meaning right in Latin) or S (for sinister, meaning left in Latin) depending on their orientation when the lowest priority is facing away from the viewer as shown in Figure 2-3. In the case of biological compounds such as amino acids and carbohydrates, different enantiomers are designated as D or L for consistency of what is found in nature. This convention in carbohydrates is based on the orientation of the chiral center in the most basic aldose, glyceraldehyde. In particular, D-glyceraldehyde causes a dextrorotatory (clockwise) rotation in polarized light when looking toward the source while L-glyceraldehyde causes a levorotatory...
(counterclockwise) rotation. Therefore, all other carbohydrates are assigned $\text{d}$ or $\text{l}$ enantiomers based on the chiral center furthest from the carbonyl group with respect to the orientation of $\text{d}$- and $\text{l}$-glyceraldehyde, regardless of their polarization rotation. An example of this configuration is shown in Figure 2-4.

Figure 2-4: The $\text{d}$ and $\text{l}$ enantiomers of glucose as compared to glyceraldehyde. Chiral centers furthest from the carbonyl group are circled in red. Adapted from Ben Mills and NEUROtiker in the public domain.\textsuperscript{101-103}

In nature, it is known that most carbohydrates, including glucose, occur as their $\text{d}$ enantiomer.\textsuperscript{99} This fact makes polarimetry plausible as a detection method because if
glucose in the blood was composed of a racemic mixture of \(D\) and \(L\) enantiomers, the rotations from each enantiomer would cancel the other out and a zero net rotation would occur. However, when dissolved in an aqueous solution, \(D\)-glucose forms a cyclic hemiacetal, resulting in two chiral anomers designated as \(\alpha\)-\(D\)-glucose (hydroxyl groups at the anomeric center and furthest chiral center are on the same side of the molecule) and \(\beta\)-\(D\)-glucose (hydroxyl groups at the anomeric center and furthest chiral center are on the opposite side of the molecule).\(^9\) The cyclic anomers of \(D\)-glucose also go by the designations of \(\alpha\)-\(D\)-glucopyranose and \(\beta\)-\(D\)-glucopyranose, indicating their six-membered ring form which is similar to the structure of pyran as shown in Figure 2-5. When either anomer is dissolved in an aqueous solution, the molecules will undergo an equilibrium process known as mutarotation, resulting in a final mixture of about one-third \(\alpha\)-\(D\)-glucose, two-thirds \(\beta\)-\(D\)-glucose, and small amounts of \(D\)-glucose in the linear and five-membered ring forms. Once at equilibrium, the rotation of polarized light caused by a sample of dissolved glucose (or any chiral molecule for that matter) is proportional to the path length and concentration of the sample for a given wavelength by a constant known as the specific rotation. This relationship can be seen in Equation 2.14 below where \([\alpha]^T_{\lambda,\text{pH}}\) is the specific rotation constant, \(\alpha\) is the observed rotation, \(L\) is the path length, and \(C\) is the sample concentration,\(^4\)

\[
[\alpha]^T_{\lambda,\text{pH}} = \frac{\alpha}{LC}.
\]

Therefore it can be seen that if optical rotation can be measured in a sample of fixed path length, then the concentration can be established based on the specific rotation at a given wavelength. This provides the primary relationship that is used for noninvasive polarimetric glucose sensing applications.
As shown by the sub- and superscripts attached to the symbol for specific rotation, it can be seen that it is dependent on temperature, pH, and wavelength. However, because the targeted area of polarimetric glucose detection is the aqueous humor of the eye, it can be assumed that physiological temperature and pH will remain constant.\(^{58}\) Therefore, the primary dependence of specific rotation is on the wavelength of the source used as shown mathematically in Drude’s equation,

\[
\alpha_{\lambda, pH} = \sum_i \frac{k_i}{\lambda^2 - \lambda_i^2}, \tag{2.15}
\]

where \(k_i\) are rotational proportionality constants, \(\lambda_i\) are absorption wavelengths, and \(\lambda\) is the source wavelength.\(^{59}\) This relationship between specific rotation and wavelength is known as optical rotatory dispersion (ORD) and it can be approximated as a more basic form when outside the vicinity of absorption bands which display the Cotton effect.\(^{106}\) The Cotton effect is a characteristic region on the ORD curve in which the specific rotation peaks before crossing zero at the absorption wavelength. The simplified ORD
model is shown in Equation 2.16,

\[
[\alpha]_{\lambda,\text{pH}}^T = k_0 \frac{\lambda^2 - \lambda_0^2}{\lambda^2 - \lambda^2_0},
\]

(2.16)

where \(\lambda_0\) is the absorption maximum for a given compound and \(k_0\) is the proportionality constant at that wavelength. Based on this equation, if the specific rotations at two or more wavelengths are known, a curve can be fit to the data and the specific rotation can be predicted for any wavelength within the fitted region. An example of this is shown above in Figure 2-6 in which data for \(\text{d}-\text{glucose}\) was fit to the model by minimizing the sum of the squares of the errors based on actual data given in Table 2.2.\(^{59}\) The figure compares the raw data to the fitted model and was plotted in MATLAB. This model also

Figure 2-6: ORD curve for glucose in the visible spectrum and outside of its absorption bands.
demonstrates the ability to use multispectral analysis in systems designed to detect a single component amongst other optically active materials with different ORD curves\textsuperscript{55-57} or in the case of time-varying corneal birefringence.\textsuperscript{63, 66-68, 92}

Table 2.2: Specific rotation data for glucose at different wavelengths in the visible spectrum.\textsuperscript{59}

<table>
<thead>
<tr>
<th>Wavelength (nm)</th>
<th>656</th>
<th>589</th>
<th>535</th>
<th>508</th>
<th>479</th>
<th>447</th>
</tr>
</thead>
<tbody>
<tr>
<td>Specific Rotation ($^\circ/(\text{dm}^*\text{g/mL})$)</td>
<td>41.89</td>
<td>52.76</td>
<td>65.35</td>
<td>73.61</td>
<td>83.88</td>
<td>96.62</td>
</tr>
</tbody>
</table>

\subsection{2.2 Optical Polarimetry}

Section 2.1 of this chapter has demonstrated the normal physiological behavior of glucose that a polarimetric detection system must be capable of continuously tracking over time. It also provided the theory behind optical activity and why glucose is a viable target for polarimetry. However, in order to successfully develop a system that is suited to measure the sub-millidegree rotations in polarized light caused by glucose, it is necessary to understand the concepts involved in Faraday-based optical polarimetry.

Optical polarimetry is a term that refers to the measurement of the polarization state of transverse electromagnetic waves (TEM).\textsuperscript{38} More specifically, a polarimeter is a device capable of measuring rotation in the state of linear polarization when a TEM passes through an optically active material such as glucose. The basic configuration of a polarimeter is shown in Figure 2-7. The schematic shows that unpolarized light is first passed through a linear polarizer before traveling through the chiral sample of a fixed path length. The plane-polarized light rotates around the axis of the direction of travel
Figure 2-7: A schematic demonstrating the principle of a basic polarimeter. Reprinted from Heesung Shim under the terms of the Creative Commons Attribution-Noncommercial-Share Alike 3.0 United States License (http://creativecommons.org/licenses/by-nc-sa/3.0/us/).

Based on the concentration of enantiomers within the sample. The rotated plane wave is then passed through an analyzer which consists of a second polarizer oriented 90° to the initial polarizer prior to being measured by a detection source (typically a photodetector). Based on this orientation, when no chiral molecules are present in the sample tube and no rotation occurs in the plane wave, the full intensity of the incident polarized light is blocked by the analyzer and the detector measures zero intensity. However, when the concentration of enantiomers in the sample is not zero, rotation will occur and a fraction of the incident light will pass through the analyzer to the detector based on Malus’s Law,

\[ I(\theta) = I(0)\cos^2(\theta), \]

(2.17)

where \( I(\theta) \) is the intensity of light seen by the detector as a function of \( \theta \), the angle between the analyzer and the plane wave, and \( I(0) \) is the maximum intensity at an angle of zero.\(^{38}\) It should be noted that the intensity is zero when the plane wave is 90° from the analyzer (i.e., no optical rotation has occurred due to the sample).
It can be seen that when rotations in polarization are small and $\theta$ is close to 90°, very little light is cast on the detector. Based on the ORD model shown in Figure 2-6, the specific rotation for glucose with a 543.5 nm light source is 63.2 °/(dm*g/mL) which corresponds to a rotational shift of 0.632 millidegrees ($\theta = 89.999368°$, $\cos^2(\theta) = 1e^{-10}$) in polarization through a 1 cm sample (characteristic of the aqueous humor) containing 10 mg/dL of glucose. Therefore, in order to accurately monitor physiological glucose concentrations over the dynamic range of 0 – 600 mg/dL, a polarimetric system must be able to achieve sub-millidegree sensitivity in the presence of ambient EM noise. As shown in Chapter 1, optical rotational modulation can successfully be used to reduce noise interference and detect the signal of interest. A modulated signal can then be detected through a photodetector attached to a lock-in amplifier which is capable of targeting signals of a known frequency in the presence of excessive noise. Lock-in amplifiers operate by comparing the frequency of a reference source with a detected signal based on the orthogonality of sine waves.\textsuperscript{108} In other words, any signal of a frequency differing from the reference signal that is input to the lock-in amplifier is filtered out, much like a very narrow band-pass filter. Therefore, various EM noise such as ambient light and normal power lines will not interfere with signal detection with an appropriate frequency selection.

### 2.2.1 Faraday Rotation in Polarimetry

Although signal modulation allows for significant noise rejection, the detection source of the polarimeter must still be capable of measuring sub-millidegree rotations. Therefore, much of the previous research has implemented a null-point, closed-loop
signal feedback mechanism which can controllably compensate for signal rotation caused by glucose.\textsuperscript{48} When digitally controlled, the feedback mechanism can detect glucose concentrations in real time. In order to control rotational modulation and compensation in a real-time, closed-loop system, a phenomenon known as Faraday rotation can be used, as shown in Chapter 1. This method of rotation is similar to that which occurs in an optically active medium.\textsuperscript{38} However, the oscillating electric field component of the propagating light causes elastically bound electrons within an optical material to vibrate in a circular orbit. The generation of this current in combination with the axial component of an external magnetic field produces a force on the electrons. This force, along with the elastic restoring force of the electrons, creates two separate dipole moments, leading to two separate indices of refraction ($n_R$ and $n_L$), and ultimately causing circular birefringence. The relationship between rotation ($\alpha$), path length ($L$), and the magnetic field strength in the direction of travel ($B$) is held by a temperature and wavelength dependent proportionality constant known as the Verdet constant ($V$) as shown in Figure 2-8 and Equation 2.18,

$$\alpha = VBL.$$  \hspace{1cm} (2.18)

As mentioned previously, materials which typically experience this phenomenon, such as TGG and TDG, can be very expensive and difficult to produce such that stress and temperature induced birefringence do not play a role in rotational interference. Therefore, it is ideal to minimize the total number of components which require these materials in order to optimize Faraday-based polarimeters, thus providing the motivation behind developing an IFMC.
2.2.2 Magnetic Fields in Faraday Rotation

Faraday rotation is most useful in polarimetry for its ability to controllably modulate and compensate polarized light. The precise control required for sub-millidegree rotational sensitivity is achieved through voltage driven inductive coils based on the relationship between a moving charge and magnetic field as shown in the Biot-Savart Law,

\[
B(r) = \frac{\mu_0}{4\pi} \oint \frac{I \times R}{R^3} \, dr_0,
\]

(2.19)

Bolded terms in this equation represent vector quantities, \(B(r)\) is the magnetic field at any point in space a distance \(r\) from the origin, \(\mu_0\) is the permeability of free space \((4\pi \cdot 10^{-7} \text{ N/A}^2)\), \(I\) is the current, \(R\) is the vector directed from the source point to \(r\), and \(dr_0\) is an element of length along the current path. As seen in this equation, the magnitude of the
magnetic field is dependent on the distance from the source as well as the magnitude of the driving current while the direction of the field will be perpendicular to the plane formed by the intersection of the current and the separation vectors. This relationship is illustrated in Figure 2-9 above. Traditional Faraday modulators and compensators produce the necessary axial magnetic field component for a given rotational depth by winding custom inductive coils such that an optical material can be placed in the center of the coil. However, as will be discussed in the proceeding chapters, ferrite-core inductors can be placed within proximity of the material in order to provide the necessary field strength along the axis of the material. Furthermore, it was shown that separate
Faraday components are not needed and a single optical rod can be used for both modulation and compensation based on the superposition of magnetic fields from separate inductive coils.

2.2.3 Modeling the Optical Components

The operation of a Faraday-based, closed-loop, null-point feedback polarimeter can be modeled mathematically using Jones vectors and matrices which are used to represent the electric field of polarized light and how it interacts with various optical components. A Jones vector is a component representation of the electric field of a polarized wave. Different optical components are represented as Jones matrices which describe the behavior of polarized light through a specific component. By multiplying the vector representation of the polarized electric field by various Jones matrices, the behavior of an optical system such as a polarimeter can be predicted. This technique is especially useful to represent the light intensity that is measured by a photodetector in Faraday-based optical polarimetry. The Jones model used to represent the detected signal intensity in such a system can be seen in Equation 2.20,

\[ E^2 = \left( \phi^2 + \frac{\theta_m^2}{2} \right) + 2\phi\theta_m\sin(\omega_mt) - \frac{\theta_m^2}{2}\cos(2\omega_mt), \quad (2.20) \]

where \( \phi \) is the difference in rotation between the optically active sample and that of the Faraday compensator, \( \theta_m \) is the modulation depth, \( \omega_m \) is the modulation frequency, and \( t \) is time. The model consists of three overall components: a DC offset, a modulated component, and a \( 2\omega_f \) component. The \( 2\omega_f \) component is present because photodetectors are only sensitive to light intensity which is always a positive value. Therefore, when the rotational modulation is centered on the null plane of the analyzer, it appears as though
the detected signal has a frequency that is twice the driving frequency due to rotational symmetry as shown on the left side of Figure 2-10. However, when an optically active sample is introduced, the rotational modulation is no longer centered on the null plane so a single frequency component exists as well, as shown on the right side of Figure 2-10. When this occurs, the lock-in amplifier detects the single frequency component and produces an output based on the non-orthogonal relationship to the reference sine wave. The principle of the null-point feedback system is to continuously apply a DC voltage to the compensator coil that is proportional to the magnetic field and, in turn, the amount of rotation caused by the optically active sample in order to force $\phi$ to zero. This is done by continuously measuring the lock-in output and compensating for rotations in polarization by a digital proportional-integral-derivative (PID) controller. When the signal is completely nulled, only the $2\omega_f$ component is present, the lock-in output is zero, and the PID output corresponds to the concentration of glucose within a fixed-length sample. The
compensator voltage can then be correlated to the glucose concentration through least-squares linear regression and the model can be used to predict unknown concentrations.

2.3 Finite Element Modeling (FEM)

As mentioned in the previous section, the primary means of control of Faraday rotation in the polarimetric system described in this thesis is through the magnetic field which is generated from current travelling around an inductive coil. Therefore, in order to properly design a system which can meet various rotational requirements it is necessary to be able to accurately predict the magnetic field in space, particularly through the optical materials. Although the Biot-Savart law can be used to make these predictions, the addition of non-uniform shapes, various materials, and non-symmetric orientations make such calculations extremely complex. However, the use of the finite element method can easily handle such complex boundary value problems in all types of engineering, providing a simple way to predict the magnetic field in various applications, and in turn can be used to select components in order to meet specific rotational requirements, such as those involved in noninvasive glucose sensing.

The benefit of using a FEM is that various spatially dependent variables can be solved within a complex geometry with varying material properties through discrete methods. The principle of this process is evident based on its name in which the volume (or surface) of interest is divided into an array of discrete elements of finite size and shape. Each element is assigned a specific set of material properties based on the physics that are being solved, such as electrical conductivity, magnetic permeability, and electrical permittivity in the case of electromagnetic fields. The array of elements
together is known as the mesh which is used to collectively define the overall system. The size of individual elements throughout the mesh may vary depending on the intricacies of the model in order for the analysis to converge to a representative result. The intersections of elements create analysis points, or nodes, which provide points in space that can numerically represent a solution. Various systems of equations can then be assigned to the model which are solved using discrete numerical and computational methods based on a set on initial boundary conditions applied to the model and the interactions between individual elements. Such models allow for multiple ordinary and partial differential as well as integral equations to be solved simultaneously depending on the approximations within each finite element. Finally, the results can be compiled as a fully integrated model in order to represent collective interactions at various regions of interest. The mathematical theory involved is based on the Galerkin method which can be used to convert continuous problems into discrete approximations. An in depth discussion of these theories can be found in a textbook by Hughes.\textsuperscript{113}

Several software applications are currently available for solving FEMs with a heavy focus on structural mechanics. However, COMSOL Multiphysics (COMSOL, Inc., Burlington, MA) is one such package which can solve models implementing several complementary physics such as structural mechanics, thermodynamics, fluid flow, chemical reactions, and electromagnetics based on independent or dependent interactions. The AC/DC Module within COMSOL is of particular interest because it allows for the analysis of magnetic field physics in three dimensional structures. The magnetic fields interface within this module solves Ampère’s Law for the magnetic vector potential based on the following equations:
\[ J_e = (j \omega \sigma - \omega^2 \varepsilon_0 \varepsilon_r)A + \nabla \times (\mu_0^{-1} \mu_r^{-1}B) - \sigma \nu \times B \]  \hspace{1cm} (2.21)

and

\[ B = \nabla \times A \]  \hspace{1cm} (2.22)

where \( J_e \) is the external current density, \( j \) is the imaginary unit, \( \omega \) is frequency, \( \sigma \) is conductivity, \( \varepsilon_0 \) is the vacuum permittivity, \( \varepsilon_r \) is the relative permittivity, \( A \) is the vector potential, \( \nabla \) is the del operator, \( \mu_0 \) is the vacuum permeability, \( \mu_r \) is the relative permeability, \( B \) is the magnetic field vector, and \( \nu \) is the current velocity vector of the conductor. Therefore, it can be seen that these equations can be solved in both static and frequency dependent conditions, providing an ideal system for predicting Faraday rotations in compensation as well as frequency-dependent modulation. Such results can be used to optimize a Faraday-based optical polarimeter for a variety of applications, as will be shown in the following chapters. More specifically, a FEM was developed in order to predict the magnetic fields caused by various inductive coils. The model along with the theories outlined in this chapter were utilized to design and optimize an IFMC as a component in noninvasive polarimetric glucose detection which was validated against dynamic physiological glucose profiles. The methodology used for applying these theories and meeting these objectives is described in the following chapter.
Chapter 3

Materials and Methods

As was shown in the previous two chapters, Faraday-based optical polarimetry has been widely used as a measurement platform for optical activity, capable of achieving sub-millidegree linear polarization sensitivity, similar to what is required in noninvasive physiological glucose detection. However, a commercially available component is currently unavailable. Given the multitude of possible applications that would benefit from such a device, an EM FEM was proposed and validated to predict which combination of design parameters would be necessary for meeting specific rotational requirements. Furthermore, it is known that the cost and size constraints in developing such a device remain to be the optical materials and the custom fabricated inductive coils. Therefore, an IFMC design was proposed which utilized a single TDG rod and off-the-shelf ferrite core inductors. The final design was optimized for use as a noninvasive polarimetric glucose detection system and validated against glucose doped water in a static sample cell. Also, a continuous glucose flow system was designed and developed as a way to promote continuing research and development in the optimization of a noninvasive glucose sensor. The flow system was capable of producing various physiological glucose profiles over time through a custom flow cell such that continuous
monitoring could be tested without the need to repeat expensive and time consuming animal experiments. Finally, the flow system was integrated with the polarimeter and the IFMC was validated against dynamic conditions. The methods used in the design and optimization of the combined Faraday component are described in detail in the following sections.

3.1 Building the Initial FEM

As was mentioned in Chapter 2, Faraday rotation can be predicted if the magnitude of the magnetic field component that is parallel to the direction of travel of a beam of polarized light is known. Also, one way of controlling this magnitude is by applying a voltage to an inductor such that current will flow, producing the magnetic field based on the Biot-Savart Law shown in Equation 2.19. In order to solve this equation to predict the field produced from an inductor, an approximation can be made that an inductor is composed of several loops of wire of negligible pitch based on the total number of turns within the inductor. A representation of a single loop of wire (which translates to a single loop of current) as applied to the Biot-Savart Law is shown in Figure 3-1. Based on this geometry and a combination of rectangular and cylindrical coordinates, it can be seen that $r_0 = r_0 \cos(\theta)\hat{x} + r_0 \sin(\theta)\hat{y} + z_0\hat{z}$, $r = x\hat{x} + y\hat{y} + z\hat{z}$, $R = r - r_0 = [x - r_0 \cos(\theta)]\hat{x} + [y - r_0 \sin(\theta)]\hat{y} + [z - z_0]\hat{z}$, and $dr_0 = [-r_0 \sin(\theta)\hat{x} + r_0 \cos(\theta)\hat{y}]d\theta$ since $r_0$ is constant around the loop. By holding the current constant, as is the case in the Faraday compensator, it can be moved outside of the integral and the cross product of $dr_0$ and $R$ can be solved. The vector quantity $Q$ was arbitrarily assigned as the solution to this cross product as given by
Figure 3-1: Basic geometry of a single loop of current as applied to the Biot-Savart Law.

\[ Q_x = [zr_0\cos(\theta) - z_0r_0\cos(\theta)]d\theta \hat{x}, \]  
\[ Q_y = [zr_0\sin(\theta) - z_0r_0\sin(\theta)]d\theta \hat{y}, \]  
and
\[ Q_z = [r_0^2 - yr_0\sin(\theta) - xr_0\cos(\theta)]d\theta \hat{z}. \]  

By letting \( \alpha = x^2 + y^2 + z^2 + z_0^2 + r_0^2 - 2zz_0 \), the magnitude of the separation vector \( R \) can be calculated to be
\[ R = \sqrt{\alpha - 2xr_0\cos(\theta) - 2yr_0\sin(\theta)}. \]  

Finally, Equations 3.1 through 3.4 can be substituted into Equation 2.19 and the component representation of the magnetic field from a single, uniform loop of current at any point in space can be given as
Based on Equations 3.5, 3.6, and 3.7, it can be seen that solving for the magnetic field at a single point in space due to a loop of current will benefit from complex numerical integration. Given that a coil inductor can be viewed as a series of current loops when in steady state, superposition can be used to evaluate the total field strength at various points in space in a multi-turn coil. However, as previously mentioned, these calculations get progressively more complex when different materials, shapes, and orientations are evaluated with multiple inductors at several points in space. Also, the model no longer holds true in an AC driven coil, such as that required for Faraday modulation. Finite element analysis is a computational method capable of solving thousands of ODEs, PDEs, and IDEs simultaneously over complicated domains by solving a matrix of nodes applied with initial conditions based on the materials and physics used. Therefore, a FEM was designed for accurate and efficient field calculations in both compensator and modulator coils.

The FEM was designed using the AC/DC Module in COMSOL Multiphysics (COMSOL, Inc., Burlington, MA). To begin, a simple model was created based on the geometry of Figure 3-1 to represent a single loop of current using the 2D axisymmetric dimension scheme as shown in Figure 3-2, the magnetic fields physics interface, and the
stationary solver sequence. A cross section of the loop of wire was drawn 5 mm away from the rotational axis and it was assigned as a single-turn coil domain with an initial vector potential of zero, the conductivity of copper, and an applied current of 71 mA (based on initial coil testing). Then the material properties of air were defined within the boundary region (200 mm radius half-circle) and a free triangular mesh was applied to break the model into finite elements before launching the solver. After solving the FEM, Equations 3.5, 3.6, and 3.7 were solved in Mathcad (PTC, Needham, MA) with the same initial conditions at various points in space in order to compare the FEM with the mathematical model. This process was then repeated for a three-loop coil domain in COMSOL and using superposition of the Biot-Savart model for three different loops.

Figure 3-2: Single loop FEM geometry as shown with 2D axial symmetry around the r-axis where the grey region represents the boundary region (200 mm radius half-circle). The various points represent analysis points while the wire cross section is 5 mm from the origin.
3.2 FEM of the Compensator and Modulator Coils

After the 2D axisymmetric FEM was related to the Biot-Savart Law calculations for a single loop and three loops of current, more complex 3D models were built and tested in COMSOL to evaluate the magnetic field produced by different coils for use in Faraday compensation and modulation. In order to evaluate the models against physical rotational measurements, six ferrite core inductors of various size, shape, and inductance were purchased. The rated inductance of each coil used was 6.8 mH (PCH-45X-685_LT, Coilcraft, Inc., Cary, IL), 15 mH (PCH-27X-756_LT, Coilcraft, Inc., Cary, IL), 68 mH (DN4546-ND, Digi-Key Corporation, Theif River Falls, MN), 100 mH (PCH-45X-107_LT, Coilcraft, Inc., Cary, IL), 220 mH (M8397-ND, Digi-Key Corporation, Theif River Falls, MN), and 470 mH (M8408-ND, Digi-Key Corporation, Theif River Falls, MN). The component specification sheets for the Coilcraft and Digi-Key inductors can be seen in Appendix A and Appendix B, respectively. Prior to modeling the components, the inductance and impedance of each coil was measured using a digital multimeter (Beckman Industrial, Fullerton, CA) and the overall dimensions were measured with digital calipers (Mitutoyo, Aurora, IL). Then, a 3D model of each coil was generated separately in COMSOL, using the multi-turn coil domain feature and the parameters listed in Table 3.1 such that the measured parameters corresponded to the values within the model.

Once the correct geometry was established for each coil, they were implemented into separate stationary FEMs to evaluate the magnetic field potential in space when used as a compensator. In order to accurately represent how the coils would be physically tested in the stationary domain, modulation coils as well as the TDG rod were also
### Table 3.1: Design parameters used to build each inductor within the FEM.

<table>
<thead>
<tr>
<th>Inductor (mH)</th>
<th>6.8</th>
<th>15</th>
<th>68</th>
<th>100</th>
<th>220</th>
<th>470</th>
</tr>
</thead>
<tbody>
<tr>
<td>Number of Turns</td>
<td>407</td>
<td>875</td>
<td>1320</td>
<td>1591</td>
<td>2815</td>
<td>4110</td>
</tr>
<tr>
<td>Wire Radius (mm)</td>
<td>0.095</td>
<td>0.0365</td>
<td>0.059</td>
<td>0.053</td>
<td>0.0352</td>
<td>0.0352</td>
</tr>
<tr>
<td>DC Voltage (VDC)</td>
<td>1.62</td>
<td>3.89</td>
<td>2.92</td>
<td>5.91</td>
<td>10</td>
<td>10</td>
</tr>
<tr>
<td>AC Source Voltage (V&lt;sub&gt;rms&lt;/sub&gt;)</td>
<td>0.068</td>
<td>0.15</td>
<td>0.15</td>
<td>0.15</td>
<td>0.50</td>
<td>0.50</td>
</tr>
<tr>
<td>AC Coil Voltage (V&lt;sub&gt;pk&lt;/sub&gt;)</td>
<td>22.3</td>
<td>17.9</td>
<td>68.2</td>
<td>71.4</td>
<td>100</td>
<td>117</td>
</tr>
<tr>
<td>Capacitance (μF)</td>
<td>3.14</td>
<td>1.416</td>
<td>0.321</td>
<td>0.222</td>
<td>0.1056</td>
<td>0.0477</td>
</tr>
<tr>
<td>Frequency (kHz)</td>
<td>1.101</td>
<td>1.103</td>
<td>1.080</td>
<td>1.080</td>
<td>1.035</td>
<td>1.073</td>
</tr>
</tbody>
</table>

modeled into the system due to the material properties and inherent mutual inductance between the modulator and compensator. This is necessary because the signal had to be modulated in the physical system in order to take comparative measurements in rotation. However, it should be noted that the modulation coils were not active in this simulation as only the DC component was being measured; they were only used to represent the surrounding materials within the test fixture. Three 100 mH inductors were selected to be used for modulation and they were oriented annularly, along with the compensator coil, around a TDG rod (13.5 mm long by 5.4 mm in diameter, MR32, Xi’an AoFa Optoelectronics Technology, Inc., Xi’an, China) at 90° intervals, based on a previously made, 3D printed test fixture, as seen in the left side of Figure 3-3. This orientation was selected to maximize the field generation along the axis of the TDG rod.

The models were then surrounded by a 100 mm (in radius) spherical air domain with a 50 mm thick infinite element shell domain to represent the analysis over all-space
Figure 3-3: The geometric orientation of the compensator analysis (left) and modulator analysis (right) used in the FEM. The letters stand for compensator (C), modulator (M, M1, M2, and M3), and TDG (V). The letters in red indicate the particular coil that was tested in a given configuration.

Table 3.2: Material properties input to the FEM as provided by COMSOL. The permeability of TDG was slightly increased compared to glass due to its magnetic properties.

<table>
<thead>
<tr>
<th>Material Domain</th>
<th>Air</th>
<th>Ferrite Core</th>
<th>Copper Wire</th>
<th>TDG</th>
</tr>
</thead>
<tbody>
<tr>
<td>Permeability</td>
<td>1</td>
<td>2000</td>
<td>1</td>
<td>1.06</td>
</tr>
<tr>
<td>Permittivity</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>4.2</td>
</tr>
<tr>
<td>Conductivity (S/m)</td>
<td>0</td>
<td>0</td>
<td>5.96*10^-7</td>
<td>1*10^-14</td>
</tr>
</tbody>
</table>

and eliminate distortion of field lines along the boundary layer. Once the geometry was complete, the material properties provided by COMSOL were assigned to each domain as shown in Table 3.2 and the multi-turn coil domain was given the necessary parameters, such as wire conductivity, wire cross section, wire direction, and number of overall turns as given in Table 3.1. The material properties of glass were assigned to the TDG rod with a slight increase to relative permability due to the magnetic properties of the material. Directionality of the current was determined using a built-in coil current calculation.
feature within the solver sequence, given a defined voltage input surface. Then, the initial vector potential was set to zero and the driving voltage was set so as to not exceed the saturation current of the coil in steady state while remaining below 10 VDC, which is the upper limit of the multifunction data acquisition device (DAQ; NI USB-6212, National Instruments, Austin, TX) that was used. The voltages used can be seen in Table 3.1. Finally, a custom mesh was designed such that each component, as well as the spaces between, could be resolved (in order to produce an accurate solution) before the solver sequence was initiated. The results were then evaluated to show consistent magnetic field generation and the average axial component along the length of the TDG rod was calculated. This could then be correlated to an overall rotation in polarized light of a given wavelength based on Equation 2.18. The results were later used in comparison to physical rotational measurements in order to validate the model. This process was repeated for each coil under consideration.

Each coil was then evaluated in a similar FEM to determine the time dependent magnetic field generation when used as a modulator. These models were created under the frequency domain solver sequence and contained only the coil of interest and the TDG rod, based on the orientation used for physical measurements as shown in the right side of Figure 3-3. This orientation did not require other coils to be modeled for modulation because the purpose of the simulation was to measure the modulation from a single coil, rather than a DC offset component in the presence of modulation. These models were also assigned with A-field gauge fixing in the magnetic field domains which was necessary to reach a stable solution within the frequency solver sequence. This provides an additional variable for potential and its affiliated conservation equation.
Then, each model was driven with a capacitive load at its resonant frequency based on physical system measurements and the coil values given in Table 3.1. The results were then used to calculate the modulation depth along the length of the TDG rod and the overall power consumption was determined. Once the compensation and modulation depths of each coil were established, they were compared to physical measurements in order to validate the accuracy of the FEM as explained in the following section. Finally, once a model was developed for each coil in compensation and modulation, a parametric sweep was run on each model in order to obtain a relationship between magnetic field strength and driving voltage.

### 3.3 Validation of the FEM

After the FEM was used to predict rotational measurements, each coil was physically tested to measure modulation depth in a standard polarimeter as depicted in Figure 3-4. The test system utilized a 5 mW, 543.5 nm HeNe laser (CVI Melles Griot, Albuquerque, NM) as the light source which passed through a linear polarizer (Newport,

![Figure 3-4: Schematic representation of the polarimetric system used to measure Faraday rotations (both modulation and compensation). The thick lines represent the path of light whereas the thin lines represent electrical communication.](image-url)
Irvine, CA) with a vertically oriented polarization plane. The polarized signal proceeded through a 1.35 cm long TDG rod (MR32, Xi’an Aofa Optoelectronics Technology, Inc., Xi’an, China) before traveling through the analyzer, consisting of a second linear polarizer (Newport, Irvine, CA) with a horizontally oriented polarization plane. The analyzer was mounted to a rotational stage controlled through a DC servo motor controller (Thorlabs, Newton, NJ) attached to a PC (Dell, Round Rock, TX) running the manufacturer’s servo control software. The final signal was detected with a high speed Si photodetector (Thorlabs, Newton, NJ) and amplified with a wide bandwidth amplifier (CVI Melles Griot, Albuquerque, NM). The inductor was oriented parallel to the TDG rod, matching the orientation used in the FEM. Prior to testing the coil, a series capacitance was chosen using off-the-shelf Mylar capacitors to achieve resonance at a frequency between 1.0 and 1.1 kHz based on Equation 3.8,

\[ 2\pi f = \frac{1}{\sqrt{LC}} \]  

(3.8)

where \( f \) is the modulation frequency, \( L \) is inductance, and \( C \) is capacitance. The corresponding measured parameters for each coil are shown in Table 3.1. Once the circuit was assembled, it was driven with the sinusoidal reference output of a lock-in amplifier (Stanford Research Systems, Sunnyvale, CA) at resonance and a maximum permitted inductor voltage based on the saturation current of the coil through a custom power amplifier (Marchand Electronics Inc., Rochester, NY) with a gain of 40 V/V. The detected signal was observed visually on an oscilloscope (Agilent, Santa Clara, CA) and input back to the lock-in amplifier based on Equation 2.20 and Figure 2-10.

Once the inductive circuit was energized, the analyzer was rotated such that the \( 2\omega_m \) component was dominant and the lock-in output was zero. Then, the angular
position of the analyzer was recorded, the stage was rotated until the $2\omega_m$ signal was eliminated, and the final angular position was recorded. The change in rotation represents the total modulation depth for a given inductor, driving voltage, and orientation when the polarizer and analyzer were initially crossed and the detected output was centered around zero. The $2\omega_m$ signal was completely eliminated when the analyzer had rotated just enough so that the peak of the modulated signal no longer crossed the null plane of the analyzer and the detected signal was of the same frequency as the reference signal ($\omega_m$). This process was carried out in triplicate and averaged for each coil before comparing values to the FEM output.

After the modulation depth of each coil was measured, they were tested to measure compensation depth when driven with a DC voltage source. The same polarimeter system described previously was used with the addition of the modulation component which consisted of three 100 mH coils, surrounding the TDG rod as shown in the left side of Figure 3-3, and a 0.0823 µF series capacitance. Once the system was assembled into the test fixture, the modulator was driven with a 20 V$_{\text{rms}}$ sinusoidal signal at a frequency of 1.1 kHz and the analyzer was rotated until the lock-in output was zero. Then, the angular position was recorded and the compensator was powered with a DC power supply (Hewlett-Packard Company, Palo Alto, CA) with the same inductor voltage used in the equivalent FEM. The analyzer was rotated again until the lock-in output was zero and the $2\omega_m$ signal was dominant before recording the final angular position. The total rotation represents how far from the null plane that the compensator coil caused the signal to rotate based on the Faraday effect. This process was also carried out in triplicate and averaged for each coil before comparing values to the equivalent FEM output.
3.4 IFMC Design and Optimization

After demonstrating the relationship between the FEM and physical rotational measurements, it was desired to apply the model to a useful application which involves Faraday-based optical polarimetry. As is evident from Chapters 1 and 2, one such application that has shown much promise is noninvasive polarimetric glucose detection. A commercial noninvasive glucose sensor would benefit the diabetic community by offering an accurate alternative to the current invasive means of detection, saving costs in healthcare and improving overall patient health and compliance. Therefore, this thesis presents a method for optimizing an IFMC for use in this application based on the FEM analysis.

As mentioned in Section 2.2, it is known that a typical physiological glucose detection system may have to operate within a maximum dynamic range of rotation between 0 and 63.2 millidegrees based on the specific rotation of D-glucose (63 °/(dm*g/mL) at 543.5 nm), given a 1 cm path length (characteristic of the aqueous humor), and an absolute worst case blood glucose concentration of 1000 mg/dL. Also, in order to minimize error below 10 mg/dL, sub-millidegree sensitivity below 0.632 millidegrees must be maintained. Therefore, to have an effective Faraday compensator component, it is desired to obtain a maximum DC rotation of at least 0.0632° so as to fully encompass the range of rotation due to physiological glucose concentrations. Also, to have an effective Faraday modulator component, it is desired to obtain a modulation depth of at least 1° which is much larger than the maximum rotation which may occur due to glucose. In order to build a system capable of meeting these requirements, an overall FEM was designed based on the results of Sections 3.2 and 3.3. The model
implemented three serially connected 100 mH inductors as the modulator and one 220 mH inductor as the compensator which were oriented in space similar to that shown in the left side of Figure 3-3 with slight modifications to the distance from the TDG rod. The geometry was implemented into both a stationary and frequency domain study and the properties were assigned as explained in Section 3.2. Initially, the stationary study was computed in order to determine the rotational capabilities of the compensator coil with an applied 10 VDC voltage, which was the upper limit of the DAQ (National Instruments, Austin, TX) used in feedback control of the polarimetric system. Then, the DC axial component of the magnetic field was averaged along the length of the TDG rod and the corresponding rotation was calculated. Next, a capacitance was chosen to be serially connected to the three modulator coils such that signal resonance could be achieved at a frequency between 1.0 and 1.1 kHz when driven with a sinusoidal voltage source, based on Equation 3.8. Circuit analysis was then done to determine the voltage drop across each coil when driven with a 20 V$_{\text{rms}}$ signal at resonance. Finally, these parameters were assigned to the modulator coils within the FEM such that current would travel in the same direction around each coil and the frequency domain study was computed. The results were used to determine the AC axial component of the magnetic field along the length of the TDG rod and the corresponding modulation depth was calculated.

Once the FEM was finalized, the physical system was built based on the results. The four coils were fixed in a custom hand-held housing that was designed in SolidWorks (SolidWorks Corporation, Waltham, MA) and prototyped with a 3D printer (Objet, Rehovot, Israel) as shown in Figure 3-5. The housing was designed with a 6 pin
mini circular connector (CP-2860-ND and CP-2060-ND, Digi-Key Corporation, Thief River Falls, MN) for quick-release connections, shown in Figure C-1 and C-2 of Appendix C, as well as with standard 30 mm optical cage mounts and ¼-20 threaded post mounts for easy integration with standard optical equipment. Also, the interior was designed such that different inserts would allow for a variety of off-the-shelf inductors to be used at various distances from the centered TDG rod, providing a universal kit for a wide range of different polarimetric applications. The four coils were assembled in the housing to replicate the FEM and all connections were soldered. The three modulator
coils were connected in series such that current would flow in the same direction around each coil, producing a collectively maximized axial component of the magnetic field along the TDG rod which is necessary for effective Faraday modulation. The final assembly was then coupled to the polarimetric system, compensation depth was measured from a 10 VDC load, and modulation depth was measured from a 20 V\textsubscript{rms} sinusoidal signal at resonance across the modulator circuit using the techniques described in Section 3.3. The measured results were compared to the output from the FEM and the overall system optimization was determined based on the requirements for a physiological glucose sensor.

3.5 Static Validation of the IFMC

After the rotational requirements of the IFMC were verified for use in a noninvasive polarimetric glucose sensor, the design was validated against static glucose measurements. This procedure was carried out with a system similar to what was described in Section 3.3 with the addition of a digital closed feedback loop, similar to what was proposed by Cameron \textit{et al.} in 1997.\textsuperscript{48} A 1 cm open air sample cell was added after the first polarizer that allowed for liquid to be easily pipetted into and out of the light path, contained between two glass microscope cover slides (Thermo Fisher Scientific, Waltham, MA). The modulated signal was monitored with the lock-in amplifier which output a signal to the DAQ (NI USB-6212, National Instruments, Austin, TX) and onto a PC. The information was processed in a PID continuous feedback virtual instrument (VI) created in LabVIEW (National Instruments, Austin, TX; shown in Appendix D). The feedback loop controlled a voltage buffer circuit (LT1010, Linear
Technology, Milpitas, CA; shown in Figure C-3 of Appendix C) powered with a DC power supply (Hewlett-Packard Company, Palo Alto, CA) through the DAQ which would apply a DC voltage to the compensator coil in order to align the polarized signal perpendicularly to the analyzer by magnetic field generation, centering the modulated signal on the null plane, and producing a $2\omega_m$ detected signal. As shown in Chapter 2, this applied voltage was proportional to any rotation in the polarized signal that would occur within the system, such as that due to glucose within the sample cell. Therefore, the output from the PID controller could be directly used to predict the glucose concentration in unknown samples of fixed path length with a linear calibration model over a known range. A schematic representation of this system can be seen in Figure 3-6 below.

Figure 3-6: Schematic representation of the polarimetric glucose sensing system used to validate the IFMC. The thick lines represent the path of light whereas the thin lines represent electrical communication.

To begin, glucose solutions were prepared in 2 mL volumes using deionized (DI) water and a 1000 mg/dL glucose stock. The stock solution was created with a powder form of $\alpha$-($\pm$)-glucose (Sigma-Aldrich, St. Louis, MO) 24 hours prior to polarimetric testing in order to achieve complete mutarotation equilibrium. Standard dilutions were made in the hypoglycemic range from 0 to 100 mg/dL in intervals of 10 mg/dL as well as
in the hyperglycemic range from 0 to 600 mg/dL in intervals of 50 mg/dL. The solutions were pipetted at random into the sample cell for each dilution range, the compensator voltage was noted, and the sample cell was rinsed with DI water between measurements. Two separate data sets were collected for both concentration ranges. Calibration models were formulated using least-squares linear regression for each concentration range after subtracting the baseline measurement. Then, the data was plotted and the SEC and SEP were calculated in MATLAB (MathWorks, Natick, MA) based on Equations 3.9 and 3.10,

\[
SEC = \sqrt{\frac{1}{(N-1)} \sum_{i=1}^{N} (C_i - \bar{C})^2}
\]  

(3.9)

and

\[
SEP = \sqrt{\frac{1}{(N-1)} \sum_{i=1}^{N} (P_i - \bar{P})^2},
\]  

(3.10)

where \(C\) is the set of residuals from the calibration data and \(P\) is the set of residuals from the prediction data. The residuals are the difference between the predicted and actual concentration.

3.6 Programmable Flow System Design

As previous work with Faraday-based glucose sensors has shown, static sample testing provides an easily repeatable method for formulating calibration models and providing error analyses for basic system validation. However, the downside to this approach is in the use of single data points for each sample concentration observed.
Given the progress with successful continuous in vivo animal studies, it is desired to provide more robust testing methods when validating continuous monitoring systems, similar to the capabilities of animal models. The downside to repetitive in vivo experiments is in the overall cost and time commitment associated with them each time new developments are made in polarimetric systems with continuing device optimization. This burden would be unnecessary if a continuous system was capable of mimicking physiological characteristics such that real-time measurements could be taken to provide more meaningful data. Therefore, a programmable glucose flow system was developed that could be integrated with the polarimeter, providing real-time output as it corresponds to changing glucose concentrations.

The programmable flow system was composed of four WPX1 peristaltic pumps with stepper motors (WELCO Co., Ltd., Tokyo, Japan; shown in Appendix E) and all-purpose silicone Tygon tubing (Saint-Gobain S.A., Courbevoie, France). A design schematic and photograph of the system can be seen in Figure 3-7. Pump 1 (WPX1-S3/16FA4-W4C-CP) was setup to run continuously to circulate glucose solutions from a well-mixed central flask, through a custom designed and 3D printed 1 cm thick flow cell as shown in Figure 3-8. The flow cell was sealed on either side with glass microscope cover slides (Thermo Fisher Scientific, Waltham, MA) such that light from the polarimeter could pass through the fluid path. Then, liquid from the flow cell was continuously circulated back into the central flask for continuous mixing. Pump 2 (WPX1-S1/8FB4-W4C-YP) was programmed to remove waste from the central flask in order to maintain a constant liquid volume during concentration changes. Pump 3 (WPX1-S3/32FB4-WM4-BP) would add DI water to the central flask during a
Figure 3-7: A schematic of the flow system (top) and the overall design coupled to the polarimeter (bottom). In the schematic, thick lines represent the fluid path while thin lines correspond to electronic communication. The component labels are as follows: light source (LS), polarizer (P), flow cell (FC), integrated Faraday modulator/compensator (IFMC), analyzer (A), photodetector (PD), pump 1 (P1), pump 2 (P2), pump 3 (P3), pump 4 (P4), central flask (CF), waste flask (WsF), water flask (WaF), glucose flask (GlF), stir plate (SP), microcontroller (MC), and motor driver (MD). It should be noted in the photo that P3 is located behind P2 and P4 is located behind P1.
concentration decrease. Pump 4 (WPX1-S3/32FB4-WM4-GP) would add a 1000 mg/dL glucose stock solution to the central flask during a concentration increase. The stepper motors on each pump were driven with a Quadstepper Motor Driver Board (SparkFun Electronics, Boulder, CO) which was controlled through an Arduino Mega 2560 R3 Microcontroller Board (Officine Arduino Torino, Torino, Italy) as shown in Figure 3-9. The microcontroller was programmed with a customized version of the open source LabVIEW Interface for Arduino (LIFA) toolkit through the Arduino development software which was capable of controlling the Quadstepper Motor Driver Board. The modified firmware code as well as the programming libraries that were uploaded to the microcontroller and used in control of the flow system can be seen in Appendices F, G, H, I, J, K, and L. Furthermore, the LIFA toolkit VIs are displayed in Appendix M as used in the custom flow system programs. Both the Arduino programming environment and the LIFA toolkit were downloaded as open source applications from the internet.117, 118
Finally, three separate LabVIEW VIs were developed and integrated to the PID VI for collecting real-time glucose measurement data based on the polarimeter response as related to the actual concentration in the flow cell. The first VI, shown in Figure 3-10, was used to purge the tubing prior to each experiment in order to fill the lines with the corresponding solutions and remove air bubbles. The second VI, shown in Figure 3-11, was designed for changing glucose concentrations on-the-fly for calibration and testing purposes. The final VI, as shown in Figure 3-12, was used for programming predefined glucose profiles to run continuously over a designated period of time in order to mimic the physiological interactions between insulin and glucose, similar to that shown in Section 2.1.1. Each VI was designed with the appropriate fields for updating real-time concentration, monitoring or changing the stepper motor status, tracking flask concentration and volume, and visualizing the polarimetric output. The custom designed
LabVIEW VIs along with the corresponding sub-VIs are shown in Appendix N.

Figure 3-10: The front panel of the LabVIEW VI designed for purging the fluid lines of the glucose flow system prior to use.

Figure 3-11: The front panel of the LabVIEW VI designed for changing glucose concentrations on-the-fly as the system was running for calibration and testing purposes.
Figure 3-12: The front panel of the LabVIEW VI designed for programming predefined glucose profiles to run continuously over a designated period of time.

When the flow system was running, the central flask was maintained at a consistent 50 mL volume and continuously mixed on a magnetic stir plate (Corning, Tewksbury, MA). When the concentration was changed in the central flask, the waste pump would pull out the allotted volume of liquid prior to the addition of glucose stock or water in order to minimize equilibrium time. Also, the circulation pump utilized a minimal length of tubing, it was set to run at a flow rate of 51 mL/min, and all concentration changes were spaced at two minute intervals in order to allow adequate time for system equilibrium during changes. Prior to initial use, each pump was calibrated using DI water and an analytical balance by weighing a sample dispensed from 10 rotations of the pump head and calculating a volume per rotation (VPR) figure. The VPR would allow for maximum accuracy over long periods of time when programmed into the flow system VI. The system was then validated by mixing glucose solutions of
known concentration using the flow system. The glucose concentration in the central flask was ramped up from 0 to 100 mg/dL in 10 mg/dL intervals and again from 0 to 600 mg/dL in 50 mg/dL intervals. After the concentration equilibrated with each change, a 2 mL sample was removed and the flask volume within the LabVIEW VI was adjusted accordingly for correct future concentration calculations. The samples were then compared to measurements taken with a YSI 2300 STAT Plus Glucose and Lactate Analyzer (YSI Life Sciences, Inc., Yellow Springs, OH) in triplicate and the system error was calculated.

3.7 Continuous Dynamic Glucose Detection

Once the flow system was validated, a new calibration model was created for the polarimeter with the custom flow cell and the IFMC. To begin, 50 mL glucose solutions were made in DI water between 0 and 200 mg/dL in 20 mg/dL intervals. Each solution was purged through the flow cell at random while running the PID VI and the voltage output to the IFMC was noted. The flow cell was flushed with DI water between each measurement and the baseline voltage was noted. This process was carried out in triplicate, the baseline was subtracted from each data point, the data sets were averaged, and a final least-squares linear regression calibration model was formed. The calibration model was then programmed into the PID VI so that continuous glucose measurements could be taken and monitored through the flow system VI in order to track a physiological concentration profile over time.

The mathematical model presented in Section 2.1.1 was used to produce the glucose profile shown in Figure 2-1. This profile reflects the glucose response in a non-
diabetic person following a 50 g oral glucose tolerance test (OGTT) beginning with fasting glucose and insulin levels using the gastrointestinal absorption rate equation as the glucose input parameter. The profile data was exported from XPPAUT to MATLAB and saved as a two column text file containing time in two minute intervals for a duration of 500 minutes (beginning with $t = 2$ min) and the corresponding glucose concentrations in mg/dL. The data was then uploaded into the flow system profile VI which would update the glucose concentration in the central flask every two minutes, producing a continuous physiological profile for the polarimeter to monitor through the flow cell.

Prior to beginning the 500 minute continuous glucose monitoring experiment, 50 mL of DI water was added to the central flask and purged through the flow cell so that no air bubbles remained in the circulation path. With the circulation pump and stir plate running, the polarimeter was turned on and the initial baseline measurement was set in the calibration model within the PID VI. Then, the glucose concentration in the central flask was brought to an initial 80 mg/dL and the flow system profile VI was started at $t = 0$ min. The polarimeter output was plotted in real-time at a frequency of 1 Hz along with the actual concentration in the central flask. After the 500 minute testing period, the system was shut down and the final data set containing the polarimeter output was exported and saved. The data was then imported into MATLAB where it was plotted and the errors for continuous measurements over time were calculated. The same continuous test was repeated two more times for a total of three data sets utilizing this model.

After the non-diabetic profile was monitored in real time with the polarimeter, it was also of interest to test the flow system in combination with the polarimeter under two more different conditions. Given that the end goal of such a noninvasive device is to
monitor the glucose levels in diabetics, a second set of data was used to create a profile based on actual clinical data. Glucose data was collected from a previous study of a patient with type 1 diabetes which was given in five minute intervals over a period of several hours. A 500 minute region was selected which reflects the ingestion of two meals followed by two bolus insulin injections. Space between the original data points was then linearly interpolated in MATLAB to provide a final data set reflecting the overall profile spaced at two minute intervals for purposes of the glucose flow system. This profile can be seen below in Figure 3-13, shown as the solid blue line, in which the two peaks represent the peak glucose seen following each meal and insulin injection.

Figure 3-13: Secondary glucose profiles which were used to test the IFMC continuously as a polarimetric glucose sensor. The solid line represents clinical data from a type 1 diabetic and the dashed line represents a steep step change.
Finally, a third data set was used in order to test the limits of the polarimeter as well as the flow system. Based on the digital feedback system used, it is known that the polarimeter is capable of responding to change much quicker than the flow system which is limited by the speed of the pumps, volume of liquid used, desired rate of change in concentration, etc. Therefore, a sigmoidal function was used to create a near-step change in concentration in order to establish the limitations of the system. Because a concentrated stock solution (1000 mg/dL) of glucose was used to step up the concentration, it was determined that diluting the solution in the central flask with water was the limiting factor. Given the length of tubing, liquid volume, concentration range, pump speeds, and time spacing, it was also determined that the maximum rate of change that the flow system could keep up with was about 28 mg/dL/min in order for equilibrium to occur. This was established by timing the response of the polarimeter during a real-time test. Therefore, a profile was created in MATLAB which could reproduce this scenario in the flow cell such that the polarimeter could monitor its progress over time, shown as the dashed red line in Figure 3-13. This profile produced a maximum rate of change of 27.5 mg/dL/min over a single two minute interval with an average rate of change of 10 mg/dL/min over each 12 minute step-change period. It should be noted that this profile provides a much more rapid response than would be expected in most physiological systems as compared to the clinical profile, meaning that the designed flow system is more than capable of testing the expectations of the polarimetric glucose sensor.
Chapter 4

Results and Discussion

This chapter serves to present the results of the experimental methods described in Chapter 3. Also, a meaningful discussion will be proposed along with the results as a way to explain them towards the overall objectives of this research. As previously mentioned, a FEM was designed and validated such that magnetic fields caused by inductive currents could be predicted for a wide range of off-the-shelf inductors. The model was then used to predict Faraday rotations in a polarimetric system which was validated against actual measurements. These results were used to select specific coils to be implemented as an IFMC based on the requirements of a noninvasive physiological glucose sensor which corresponds to $1^\circ$ for modulation depth and $0.0632^\circ$ for maximum compensation depth. The system was built and validated against static glucose measurements in order to verify that sub-millidegree sensitivity could be achieved. Finally, the IFMC was further validated in physiological dynamic conditions with a custom designed and programmed glucose flow system. This system provides a method for recreating real physiological conditions based on predetermined glucose profiles as a means to forgo unnecessary, expensive, and time consuming animal testing in future research and development and device optimization. Also, the IFMC largely benefits the commercialization of a
noninvasive polarimetric glucose sensor by implementing a more compact design with a single TDG rod and off-the-shelf inductors, reducing the size and cost. Furthermore, the FEM provides a way to further customize these polarimetric systems to be used in a wide range of applications that require sub-millidegree rotational sensitivity.

4.1 Building the Initial FEM

The initial FEM was built in COMSOL to determine its relationship to results calculated using the Biot-Savart Law during a stationary analysis. Because the model was intended to evolve into a combination of various inductive coils, which can be viewed as a collection of several current loops based on the superposition of magnetic fields, a single loop of current with a radius of 5 mm and a magnitude of 71 mA was evaluated in both models. Also, for the purpose of this investigation, three coordinate points were chosen as analysis points. With the loop centered at the origin around the z-axis (as shown in Figure 3-1), the selected analysis points were (0, 10, 0), (0, 10, 10), and (0, 0, 0) given in mm in the y-z plane and referenced as pt1, pt2, and pt3, respectively (as shown in Figure 3-2). The FEM resulted in a magnetic field at these points of magnitude 7.57 mG, 3.02 mG, and 88.11 mG, respectively. The same coordinates were evaluated in Equations 3.5, 3.6, and 3.7 which resulted in a magnetic field of magnitude 7.60 mG, 3.03 mG, and 88.09 mG, respectively. Based on these results, the average error between the FEM and the Biot-Savart Law was calculated to be 0.25%, as summarized in Table 4.1. The FEM results can also be seen in Figure 4-1 which shows a concentration of field strength in close proximity to the coil (left) with field lines extending perpendicularly to the current direction (right), as expected. However, it should be noted that the field lines
Table 4.1: Initial 2D FEM predicted magnetic field results as compared to the Biot-Savart Law calculations.

<table>
<thead>
<tr>
<th>Coordinate Point (mm)</th>
<th>Single Loop Analysis</th>
<th></th>
<th>Triple Loop Analysis</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>FEM (mG)</td>
<td></td>
<td>Biot-Savart (mG)</td>
<td>Error (%)</td>
</tr>
<tr>
<td>(0, 10, 0)</td>
<td>7.57</td>
<td></td>
<td>7.60</td>
<td>0.39</td>
</tr>
<tr>
<td>(0, 10, 10)</td>
<td>3.02</td>
<td></td>
<td>3.03</td>
<td>0.33</td>
</tr>
<tr>
<td>(0, 0, 0)</td>
<td>88.11</td>
<td></td>
<td>88.09</td>
<td>0.02</td>
</tr>
</tbody>
</table>

Figure 4-1: Colormap magnitude plot given in Gauss (left) and streamline direction plot (right) of the magnetic field from the FEM analysis of a single loop of current.

appear distorted around the boundary layer, indicating the need for an infinite element domain surrounding this boundary in order to resolve proper directionality. Furthermore, it is known that the total current-induced magnetic field at any point in space is the vector sum of all the individual contributions. Because the final COMSOL model will need to demonstrate this concept, a simple sum of three identical current loops was also investigated. This FEM resulted in a magnetic field at pt1, pt2, and pt3 of magnitude 22.75 mG, 9.04 mG, and 264.13 mG, respectively. The Biot-Savart Law resulted in magnitudes of 22.76 mG, 9.10 mG, and 264.03 mG, respectively. Based on these results, the average error between the FEM and the Biot-Savart Law was calculated to be 0.25%,
as summarized in Table 4.1. The FEM results can also be seen in Figure 4-2 below with similar expectations as the single loop of current.

Figure 4-2: Colormap magnitude plot given in Gauss (left) and streamline direction plot (right) of the magnetic field from the FEM analysis of three loops of current.

Although the error between these models was negligible, the root cause for it was likely due to the boundary effect and distortion of field lines. Also, a FEM by nature is based on the approximation of interactions between elements of finite size, rather than an infinite number of infinitesimal elements (as is the case in standard integral equations), meaning that the result is largely dependent on its meshing characteristics. However, given the high resolution of the mesh used, the results show that there is a good correlation between the FEM prediction and the actual calculated value for inductive magnetic fields. Therefore, it was established that the model can be used for more elaborate 3D coil configurations with the addition of an infinite element domain as shown in the following sections.
4.2 FEM of the Compensator and Modulator Coils

Once it was shown that the FEM was capable of producing results similar to the theoretical model, the geometry of six off-the-shelf inductors was built within COMSOL so that they could be analyzed under both DC and AC conditions for magnetic field generation. Parameters such as relative permeability, conductivity, wire radius, and number of turns were assigned to each model and the FEM calculated impedance and inductance were compared to their respective measured values from the physical components. The comparison of these values along with the associated errors can be seen in Table 4.2 and Figure 4-3 (created in MATLAB). The average errors in inductance and impedance between the physical inductor and the FEM were 0.30% and 0.54%, respectively. Therefore, the models can be expected to accurately represent the physical inductors under normal operating conditions.

Table 4.2: Error between measured and FEM values of inductance and impedance for each off-the-shelf coil.

<table>
<thead>
<tr>
<th>Expected Inductance (mH)</th>
<th>Measured Inductance (mH)</th>
<th>Calculated Inductance (mH)</th>
<th>Inductance Error (%)</th>
<th>Measured Impedance (Ω)</th>
<th>Calculated Impedance (Ω)</th>
<th>Impedance Error (%)</th>
</tr>
</thead>
<tbody>
<tr>
<td>6.8</td>
<td>6.65</td>
<td>6.64</td>
<td>0.15</td>
<td>5.14</td>
<td>5.16</td>
<td>0.39</td>
</tr>
<tr>
<td>15</td>
<td>14.7</td>
<td>14.61</td>
<td>0.61</td>
<td>45.80</td>
<td>45.73</td>
<td>0.15</td>
</tr>
<tr>
<td>68</td>
<td>67.7</td>
<td>67.96</td>
<td>0.38</td>
<td>48.70</td>
<td>48.67</td>
<td>0.06</td>
</tr>
<tr>
<td>100</td>
<td>97.8</td>
<td>97.62</td>
<td>0.18</td>
<td>72.96</td>
<td>73.17</td>
<td>0.29</td>
</tr>
<tr>
<td>220</td>
<td>224</td>
<td>223</td>
<td>0.45</td>
<td>399</td>
<td>391</td>
<td>2.01</td>
</tr>
<tr>
<td>470</td>
<td>462</td>
<td>462</td>
<td>0</td>
<td>624</td>
<td>626</td>
<td>0.32</td>
</tr>
</tbody>
</table>

After the models were validated against measured inductance and impedance values, a stationary analysis was carried out in the presence of three modulator coils as well as the TDG rod. The reason for including these materials was to accurately represent
the fixture that would be used for physical measurements, allowing for the necessary interactions to occur, such as mutual induction between coils. Once the collective model was built, material properties were assigned to each coil and the analysis was computed using a maximum DC voltage across the compensator coil of a magnitude depending on the rated saturation current of each respective coil. After the analysis was run, a 3D plot was created in COMSOL to show the magnetic field amplitude and the direction of the field lines as compared to current travel through the inductor. Also, a second plot was created to visualize the axial component of the magnetic field along the length of the TDG rod. An example of these plots can be seen in Figure 4-4 which shows the 3D results based on the 470 mH coil on the left (created in COMSOL) and the collective axial B-field results from each coil on the right (created in MATLAB). It can be seen that

Figure 4-3: Correlation of inductance and impedance between physical measurements and FEM calculations of six off-the-shelf inductors.
Figure 4-4: The 3D colormap plot of magnetic field (G) from the 470 mH inductor (left) and the 2D plot of the axial component of the magnetic field through the TDG rod from each inductor (right) after stationary FEM analysis. The green coils in the left plot represent the direction of current and the red arrows represent the field direction.

The magnetic field direction was perpendicular to the direction of current travel in the coil and the largest B-field strength was localized within the magnetic core domain, as expected. Similar results were obtained from each coil and the peak magnetic field was shown to occur along the middle of the inductor when \( y=0 \) mm as expected. The offset in symmetry shown on the right side of Figure 4-4 was due to the position of the TDG rod with respect to the coil which was inherent to the test fixture used in the physical system. Had the TDG rod been centered along the inductor, the resulting B-field plots would have been symmetric along the \( y=0 \) mm coordinate. It should also be noted that there was no overall relationship between individual inductors because they varied in size and shape.

Furthermore, the results in Figure 4-4 demonstrate the maximum B-field from each inductor which represents a range of driving voltages based on the rated saturation current. These results can then be averaged along the length of the TDG rod in order to predict the maximum compensation depth seen in the polarized light due to each
configuration based on Equation 2.18, a Verdet constant of 0.46 arcmin/G/cm, and a path length of 1.35 cm, as will be shown in the proceeding sections.

Once the stationary results for each coil were established, a frequency domain analysis was performed in order to determine the maximum modulation depth. These models implemented the individual coil and TDG rod geometry without extra modulation coils in order to reflect the orientation of the physical test fixture. The same material properties were assigned and a capacitance was calculated such that signal resonance could be achieved at a frequency between 1.0 and 1.1 kHz when driven with a sinusoidal source based on Equation 3.8. The model was then setup to run at the selected frequency with a maximum AC voltage across the coil of a magnitude that would prevent current saturation. The steady state results were then displayed on a 3D plot during peak modulation as shown on the left of Figure 4-5 which presents the results from the 470 mH inductor (created in COMSOL). Also, a 2D plot was created of the average axial magnetic field component along the length of the TDG rod with respect to the phase angle of the driving source, as shown on the right of Figure 4-5 (created in MATLAB). This plot displays the maximum results from each coil. Again, it can be seen that the magnetic field direction is perpendicular to the current travel and the field intensity follows a sinusoidal path dependent on the frequency of the voltage source, as expected. These results also show no overall relationship between individual inductors due to the same reasons described for the compensator models. However, the peak results can be used to predict the maximum modulation depth seen in polarized light due to each configuration based on Equation 2.18, a Verdet constant of 0.46 arcmin/G/cm, and a path length of 1.35 cm as will be shown in the proceeding sections.
Figure 4-5: The 3D colormap plot of magnetic field (G) from the 470 mH inductor (left) and the 2D plot of the average axial component of the magnetic field through the TDG rod with respect to the phase angle of the driving source from each inductor (right) after frequency domain FEM analysis. The green coils in the left plot represent the direction of current and the red arrows represent the field direction.

After each FEM was tested for maximum compensation and modulation depth, a parametric voltage sweep was run to determine the relationship between the driving source and the resulting Faraday rotation. Each model was tested over a linear voltage range below the maximum rated value in both compensation and modulation. The results from each coil in compensation and modulation can be seen in Figure 4-6 and Figure 4-7, respectively. These results indicate that the different size, shape, and inductance of each coil provide a unique operational range of rotations based on the maximum voltage. It should also be noted that the 15 mH coil should not be used in such Faraday-based polarimetric applications because of the variation in sign of the magnetic field along the length of the TDG rod. Due to the short length of the coil, a portion of the magnetic field along the length of the TDG rod was negative, meaning that it will work against itself to achieve a collective rotation in one direction. However, the magnetic field results from
Figure 4-6: FEM parametric voltage sweep results from each coil in compensation. Starting with the upper left plot and proceeding from left to right, top to bottom: 6.8 mH coil, 15 mH coil, 68 mH coil, 100 mH coil, 220 mH coil, and 470 mH coil.
Figure 4-7: FEM parametric voltage sweep results from each coil in modulation. Starting with the upper left plot and proceeding from left to right, top to bottom: 6.8 mH coil, 15 mH coil, 68 mH coil, 100 mH coil, 220 mH coil, and 470 mH coil.
each coil were averaged and used to calculate Faraday rotation based on the driving voltage. The plots which relate rotation to driving voltage can be seen below in Figure 4-8 based on the FEM data shown in Tables O.1 through O.12 of Appendix O. These results demonstrate a linear relationship between rotation and voltage in both compensation and modulation within the operational range of each inductor. Plots such as these would be useful for selecting the necessary driving voltage in order to achieve a required rotational depth in polarimetric systems with specific inductors. Given that the lines pass directly through zero, the slope can be used for making these predictions.

![Figure 4-8](image)

**Figure 4-8:** The relationship between compensation depth and driving voltage (left) and modulation depth and driving voltage (right) for each inductor based on the FEM results.

Finally, the voltage sweep results were utilized to provide a comparison in magnetic field strength between each inductor when driven with a common voltage source, as seen in Figure 4-9. The 1 VDC results were compared between coils in compensation and the 10 Vpk results were compared between the coils in modulation. These results indicate that there was an inverse relationship between inductance and field strength. The exception to this rule is shown to be the 15 mH coil which is likely due to
the smaller size as compared to the other five coils. Overall, this type of relationship is expected because coils with a higher inductance are generally manufactured with more turns of wire, producing larger resistances, and reducing the amount of current that can flow through the inductor. Although these coils have more turns which also affect the total magnetic field generated, the larger resistance seems to be the limiting factor. Therefore, as a general rule, in order to produce larger Faraday rotations, coils with lower inductances should typically be used. Then, the linear relationship between rotation and voltage can be used to pinpoint the exact rotation needed in a given application.

![Figure 4-9](image)

**Figure 4-9:** The FEM results comparing each coil in compensation while being driven with a 1 VDC source (left) and in modulation while being driven with a 10 Vpk source (right).

### 4.3 Validation of the FEM

The previous section demonstrates how a FEM could be used to predict Faraday rotation in different coils used as a polarimetric modulator or compensator. Also, Section 4.1 shows that the FEM of a single loop of current correlates to the calculated magnetic
field values based on the Biot-Savart Law. However, it is also desired to show that the more complex 3D models can accurately predict Faraday rotation based on magnetic field analysis in both the stationary and frequency domains. Given that Equation 2.19 becomes very complex when non-uniform geometry is implemented, the polarimetric system shown in Figure 3-4 was used for making rotational measurements to compare with the FEM output. Initially, the maximum modulation depth of each coil was measured using the polarimetric system described in Section 3.3. The coils were oriented in a test fixture, parallel to the TDG rod. The series capacitance required for signal resonance was connected to the coil and it was energized with a sinusoidal signal of equivalent amplitude and frequency used in the FEM. Then, the analyzer was rotated until the $2\omega_m$ component was gone and the modulation depth was noted. This process was done in triplicate, the results were averaged, and measured rotation versus the FEM calculated rotation was plotted in MATLAB as shown in the right side of Figure 4-10. The

![Figure 4-10](image-url): The prediction error in compensation (left) and modulation (right) between the FEM analysis and physical measured values.
rotational data based on these measurements and the FEM magnetic field calculations are shown in Table O.13 of Appendix O. The standard deviation of the residuals of these values was then calculated to determine a prediction error between measurements and the FEM calculations. The error in prediction of modulation was calculated to be 0.0044° which is 0.71% of the largest rotation seen in the data. Therefore, it can be seen that the developed FEM can accurately predict the depth of Faraday modulation for various off-the-shelf inductors.

A similar process was used for measurement of the maximum compensation depth that each coil was capable of achieving with the addition of a modulation component to be used for signal detection. Modulation was necessary due to the sensitivity required for measuring millidegree polarization rotations, as previously explained. The modulator was driven with a 20 \( V_{\text{rms}} \) sinusoidal signal at a frequency of 1.1 kHz and each compensator coil was powered with the same DC voltage used in the FEM so as to avoid current saturation and without exceeding a 10 VDC limit of the DAQ. The analyzer was then rotated to eliminate the \( \omega_m \) offset seen in modulation due to the compensator and the amount of rotation was noted. The process was repeated in triplicate for each coil, the results were averaged, and measured rotation versus the FEM calculated rotation was plotted in MATLAB as shown in the left side of Figure 4-10. The rotational data based on these measurements and the FEM magnetic field calculations are shown in Table O.14 of Appendix O. Based on these results, the error in prediction of compensation was calculated to be 0.0087° which is 3.26% of the largest rotation seen in the data series. Although this was higher than the modulator prediction error, it was still within reasonable accuracy for successful compensator prediction and coil selection. The
primary source of this error was due to the difficulty of measuring angles with sub-millidegree sensitivity, rather than general discrepancies within the FEM. Using a mechanical rotational stage for measuring polarization angles of this magnitude was reasonable but still produces inherent error. Also, the placement of the coil with respect to the TDG rod largely affects the rotational results meaning that any slight difference between the test fixture and the FEM geometry could also produce error. However, it was shown that the overall model provides an accurate representation of the physical system.

4.4 IFMC Design and Optimization

The results presented in the previous sections indicate that the FEM can successfully predict Faraday rotation in both compensation and modulation components in a polarimetric system. However, it is also beneficial to show the application of such a model in the development and optimization of a useful device. As mentioned in Chapter 1, much research has been devoted to the study of Faraday-based noninvasive polarimetric glucose monitoring as a means for promoting better overall treatment for diabetics. It is known that the primary size and cost factor in these systems has been in the separation of Faraday modulation and compensation components which require multiple optical rods and custom built inductive coils. Therefore, the FEM proposed in this thesis was used to design and optimize an IFMC which would utilize a single TDG rod and off-the-shelf inductors while still maintaining the necessary design requirements. To achieve this, the maximum FEM data shown in Table O.13 and O.14 of Appendix O was plotted in MATLAB as shown in Figure 4-11. These plots were used to select the optimal coils for device design based on the requirements given in Section 3.4. The plots
Figure 4-11: A plot displaying maximum compensation depth versus driving voltage (left) and a plot displaying maximum modulation depth versus average power consumption (right) for different off-the-shelf inductors based on the FEM output. The horizontal blue line on the plot on the left indicates the minimum rotational requirement necessary for successful physiological glucose measurements.

represent compensation depth versus driving voltage (left) as well as modulation depth versus average power consumption (right). These types of plots would be useful for selecting a compensator coil to be used in an IFMC when a minimum rotation is needed and there is a maximum voltage allotted by the driving electronics. The minimum rotational requirement for a physiological glucose sensor is represented as the blue horizontal line on the plot on the left. They would also be useful for selecting a modulator configuration to be used in an IFMC when a minimum depth must be achieved while maintaining low power. Points on the upper-left side of the plot on the right represent the highest rotation to power ratio (RTPR).

The coil selected to operate as the compensator was the 220 mH inductor because it was able to achieve a rotation of 0.0740° which is larger than the minimum requirement of 0.0632° and it can utilize the full 10 VDC range of the DAQ used in
feedback control. Due to the fact that no single coil was able to achieve a modulation depth larger than 1°, three 100 mH coils connected in series were selected to operate as the modulator component. These coils were selected because they produce an individual modulation depth of 0.4881° and have the highest RTPR of 0.9961°/W.

Once the necessary coils were selected, a FEM was built to reflect the properties of the coils, surrounding the parallel TDG rod. Initially, a stationary study was computed with a 10 VDC source applied to the 220 mH coil in order to predict the maximum possible compensation depth. Then, a frequency domain study was computed with a 45 V$_{\text{rms}}$ sinusoidal source at a frequency of 1.073 kHz applied to each 100 mH coil. These studies resulted in a total compensation depth of 0.0780° and modulation depth of 1.2404°, indicating that both requirements were successfully met by the IFMC design. The average power calculated for the modulation circuit was 1.99 W which produces a RTPR of 0.6233°/W. The decrease in this figure when compared to individual 100 mH coils was due to the mutual induction losses between coils which also explains why the total modulation depth was not simply three times as large as the depth produced by a single coil. A 3D colormap plot of the IFMC in compensation and modulation from COMSOL can be seen in Figure 4-12. Both images indicate the expected interaction between coil currents and magnetic field which further supports the FEM.

Once the FEM indicated that the IFMC would meet the design requirements, a physical prototype was built in a custom 3D printed hand-held housing in order to mimic the FEM results. The components were soldered together and fixed in the housing. The final design was then integrated with the polarimetric system used for rotational measurements and the total compensation and modulation depths were determined. The
compensator portion of the IFMC produced a total inductance of 227 mH, an impedance of 358 Ω, and an average rotation of 0.0794° when energized with a 10 VDC source. These results correspond to a measurement error of 1.79% which is well within acceptable limits based on the sensitivity of the rotational stage and fixture geometry used. The modulator portion of the IFMC produced a measured inductance of 245 mH, an impedance of 201 Ω, and was attached in series to a 0.0898 µF capacitance for resonance. The resulting modulation depth when driven with a 20 V\text{rms} sinusoidal source at 1.073 kHz averaged out to be 1.2226°, producing a measurement error of 1.44%. The low overall error associated with the IFMC demonstrates that a FEM can be used to successfully design and optimize the components within a specific polarimetric application. Although the IFMC may be optimized, in theory, to predict physiological glucose concentrations, its capability as a practical alternative to the widely reported separate modulator/compensator systems must be demonstrated. Therefore, the final IFMC design was validated under two separate conditions. First, static measurements

Figure 4-12: The 3D colormap plot of magnetic field (G) of the IFMC in compensation (left) and modulation (right) after FEM analysis. The green coils represent the direction of current and the red arrows represent the field direction.
were taken to formulate a calibration model and calculate errors in prediction. Second, a
dynamic flow system was designed to control glucose concentrations real-time so that
measurements could be taken continuously. The results of these validations are given in
the following sections.

4.5 Static Validation of the IFMC

In order to compare the IFMC to previous Faraday-based glucose detection
systems, polarimetric measurements were taken in the hypoglycemic and hyperglycemic
concentration ranges. The raw compensator voltage data as collected with the IFMC can
be seen in Table O.15 and Table O.16 of Appendix O. Once two sets of data were
collected in each range (one for calibration and one for validation), they were imported
into MATLAB to formulate a least-squares linear regression calibration model and to
calculate the SEC and SEP for each range. The calibration and prediction results for the
hypoglycemic and hyperglycemic concentration ranges against the actual concentration
values are shown below in Figure 4-13 and Figure 4-14, respectively. The hypoglycemic
data resulted in a SEC of 1.6 mg/dL with an $R^2$ of 0.9977 and a SEP of 1.8 mg/dL with
an $R^2$ of 0.9970. The hyperglycemic data resulted in a SEC of 5.2 mg/dL with an $R^2$ of
0.9993 and a SEP of 5.4 mg/dL with an $R^2$ of 0.9992. The main source of error within the
system was due to the sensitivity required to measure sub-millidegree rotations. Although
optical systems are capable of accurately detecting these measurements, electromagnetic
noise and physical vibrations play a large role in the overall capabilities of these systems.
Based on these results, it is evident that the IFMC is capable of achieving measurement
sensitivity with similar, if not better, accuracy than its larger two-part predecessors.
Figure 4-13: The calibration (left) and prediction (right) results using a hypoglycemic glucose concentration range as measured with the IFMC.

Figure 4-14: The calibration (left) and prediction (right) results using a hyperglycemic glucose concentration range as measured with the IFMC.

However, due to the fact that the current research with Faraday-based glucose sensors is in the *in vivo* stage of development, it is desired to demonstrate the capabilities of the IFMC in a similar fashion. Therefore, the design and development of a glucose flow
system as well as continuous monitoring results using the IFMC are presented in the following sections.

4.6 Programmable Flow System Design

As previously stated, in vivo experiments can be very expensive and time consuming. Therefore, it is desired to reproduce the physiological conditions experienced in animal models without the need for actual animal testing. A primary testing characteristic of in vivo systems is the capability to track physiological glucose profiles in real-time. In order to further validate the IFMC, a glucose flow system was developed, providing robust and continuous measurement capabilities. The system was calibrated against DI water and it was determined that pumps 1, 2, 3, and 4 would dispense 0.910, 0.472, 0.337, and 0.336 mL per revolution of the pump head, respectively. These VPR values were programmed into the flow system VI and it was used to create standard glucose dilutions in the hypoglycemic and hyperglycemic ranges. The samples were measured in triplicate using the YSI Glucose Analyzer and the errors between expected and actual values were calculated. The measurement data can be seen in Table O.17 and O.18 of Appendix O. The averaged data resulted in a standard error of 0.5 mg/dL in the hypoglycemic range and 6.4 mg/dL in the hyperglycemic range. It should be noted that the precision of the YSI is quoted in the manufacturer’s documentation to be within 2.5 mg/dL for measurements up to 100 mg/dL and 12 mg/dL for measurements up to 600 mg/dL. Therefore, the standard errors are well within the limits of the YSI. Given the precision control of the stepper motor driven pump heads and the individual calibration of each head, the primary source of the error was likely centered on the YSI rather than
the flow system. Overall, the custom designed flow system was capable of producing accurately controlled real-time glucose solutions in order to facilitate robust continuous polarimetric measurements similar to standard *in vivo* experiments.

### 4.7 Continuous Dynamic Glucose Detection

The final objective of the current study was to evaluate the IFMC design in a real-time dynamic system using the glucose flow system and a custom 1 cm flow cell. Prior to running the 500 minute physiological glucose profile with the IFMC, a new calibration model was formulated for the flow cell. This was done by averaging three collected data sets, less the baseline measurements, over a range of 0 to 200 mg/dL and fitting the data to a least-squares linear regression model. The raw compensator voltage and baseline data as collected with the IFMC can be seen in Table O.19, Table O.20, and Table O.21 of Appendix O. The new calibration model resulted in a SEC of 3.5 mg/dL with an $R^2$ of 0.9972 which is in similar agreement with the static glucose data. This model was then programmed into the PID VI and the central flask in the flow system was filled with 50 mL of DI water. Once the flow cell was purged, the polarimeter was powered on and the glucose concentration in the central flask was brought to an initial 80 mg/dL. Then, the flow system profile VI was started and the changing concentrations were monitored over a 500 minute period with the IFMC. The results of the continuous polarimeter output can be seen in Figure 4-15 as compared to the actual real-time glucose concentration in the central flask. As shown in the figure, the polarimeter tracks the overall profile with precision. However, as time goes on, the output signal drifts with respect to the actual profile. In order to quantify the error over time, the SEP was calculated at 100 minute
intervals using the raw data. The SEPs during the first 100, 200, 300, 400, and 500 minutes were determined to be 4.3, 3.7, 3.2, 3.1, and 3.2 mg/dL, respectively. Based on these results, it can be seen that the system is capable of tracking concentration changes with a precision that is consistent with the calibration model. In other words, the calibration model remained valid but the baseline has noticeably shifted. In order to quantify the baseline drift, the mean absolute error (MAE) of the residuals was also calculated over the same 100 minute intervals to be 5.1, 5.8, 5.5, 6.0, and 6.4 mg/dL, respectively. These results show a consistent decrease in system accuracy as drift occurs over time. The cause for the drift can easily be explained due to the physical sensitivity of the system. Although the optical components were mounted on an optical table and supported with 30 mm cage mounts, the slightest table movement can lead to spikes in the system output (as shown around the 210 minute mark on the left of Figure 4-15). This coupled with consistent vibrations of the driving electronics, the stir plate, and the
peristaltic pumps on the table can cause the position of each component in the system to slightly shift over time, resulting in a change in the polarization state of the initial signal. A solution to this problem would be to fix each component in such a way to resist this type of vibrational movement in the final hand-held device.

A final analysis on the data was done to realign the baseline every 20 minutes to the calibration model before running a 60 second moving average. This type of baseline correction can be rationalized due to the nature of the final system that is to be used by diabetics. Continuous measurements are ideal for achieving accurate results. However, a hand-held glucose meter would be used by patients on the order of a few minutes at a time, meaning that baseline measurements can be taken before and after each reading. Applications that require long-term monitoring, such as use with sugar monitoring in cell culture bioreactors, can be designed to implement single point baseline corrections which are common for many commercial glucose meters such as the YSI. The baseline corrected results are shown in the right side of Figure 4-15 and demonstrate a much better fit to the actual concentration profile. It should be noted that the system was not recalibrated as the model was still valid based on the SEP, but rather the baseline was shifted accordingly and the excessive noise was masked. The final baseline corrected SEP was calculated to be 2.4 mg/dL with the overall MAE of the residuals to be 1.6 mg/dL. These results suggest that the baseline corrected data maintains similar error to the calibration model as well as to the raw data while the drift was negated. It should also be noted that the data begins to branch off track during the last 20 minutes of the experiment. This was caused by an air bubble that became trapped in the flow cell. Problems such as these could be minimized by redesigning the interior volume of the
flow cell. However, air bubbles are primarily an issue with the sample cell design and would not be a problem during actual physiological sensing. A summary of the continuous results can be seen in Table 4.3 below.

Table 4.3: Error results from each continuous glucose experiment with the IFMC, including both raw data and the single-point baseline corrected data. All values are given in mg/dL.

<table>
<thead>
<tr>
<th></th>
<th>Raw Data</th>
<th>Baseline Corrected Data</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>SEP</td>
<td>MAE</td>
</tr>
<tr>
<td>Non-Diabetic 1</td>
<td>3.2</td>
<td>6.4</td>
</tr>
<tr>
<td>Non-Diabetic 2</td>
<td>7.9</td>
<td>6.3</td>
</tr>
<tr>
<td>Non-Diabetic 3</td>
<td>10.3</td>
<td>10.1</td>
</tr>
<tr>
<td>Diabetic</td>
<td>4.8</td>
<td>3.8</td>
</tr>
<tr>
<td>Near-Step Change</td>
<td>11.1</td>
<td>23.5</td>
</tr>
<tr>
<td>Average</td>
<td>7.5</td>
<td>10.0</td>
</tr>
</tbody>
</table>

The real-time test was repeated two more times and the results are shown in Figure 4-16 and Figure 4-17, respectively. Both of these plots display various baseline drift over time due to vibrational effects, similar to the initial results, further indicating the overall sensitivity of the polarimetric system. The second trial resulted in an overall SEP of 7.9 mg/dL and a MAE of 6.3 mg/dL from the raw data and a SEP of 4.2 mg/dL and a MAE of 2.8 mg/dL after performing the same baseline correction and moving average. The third trial resulted in an overall SEP of 10.3 mg/dL and a MAE of 10.1 mg/dL from the raw data and a SEP of 3.6 mg/dL and a MAE of 2.6 mg/dL after baseline correction. Therefore, it can be seen that the results are repeatable and a single point baseline correction can be used to realign any vibrational baseline drift, which is already common in many commercially available systems.
After it was shown that the IFMC could successfully track a non-diabetic physiological glucose profile over time based on a controllable mathematical model, the
test was repeated with two different profiles. First, the system was tested against a clinical profile from a patent with type 1 diabetes, providing information following meal ingestion and insulin injections. Then, the limitations of the flow system and IFMC were tested against a near-step change profile with a maximum rate of change in concentration of 27.5 mg/dL/min and an average change of 10 mg/dL/min. The results from these remaining experiments can be seen in Figure 4-18 and Figure 4-19, respectively. The diabetic profile resulted in an overall SEP of 4.8 mg/dL, MAE of 3.8 mg/dL, baseline corrected SEP of 3.4 mg/dL, and baseline corrected MAE of 2.5. The near-step profile resulted in an overall SEP of 11.1 mg/dL, MAE of 23.5, baseline corrected SEP of 5.4 mg/dL, and baseline corrected MAE of 4.0 mg/dL. The diabetic profile demonstrates good tracking, even in the raw data, while the step profile shows a variable amount of baseline drift. It can be seen that the SEP and MAE encountered from the raw data are variable depending on the overall amount of drift. However, the single-point baseline corrected data provides a better representation of the system with an average SEP of 3.8 mg/dL and an average MAE of 2.7 mg/dL from all data sets. This indicates that the relative amount of drift that occurs in the system is situational and the single-point baseline correction can successfully combat this issue. However, better fixation of the system will ultimately reduce this effect. Furthermore, these results indicate that the IFMC can successfully track a clinical diabetic profile during normal conditions as well as during the worst case conditions which are represented by the high rate of change seen in the near-step change profile. Regardless of the drift, it can be seen that the IFMC is capable of achieving the necessary rotational sensitivity and stability required for continuous physiological glucose detection.
Figure 4-18: The raw data resulting from a 500 minute continuous polarimetric glucose detection test using a clinical diabetic profile as measured with the IFMC (left) and the 20 minute baseline corrected, 60 second moving averaged data profile (right). Both data sets are plotted against the actual physiological glucose profile used during the test.

Figure 4-19: The raw data resulting from a 500 minute continuous polarimetric glucose detection test using a near-step change profile as measured with the IFMC (left) and the 20 minute baseline corrected, 60 second moving averaged data profile (right). Both data sets are plotted against the actual physiological glucose profile used during the test.
Finally, the functionality of the flow system during rapid concentration changes was also observed during the final test. A zoomed in view of these results during the first increase and decrease in concentration can be seen in Figure 4-20. It should be noted that there was a slight delay between when a change in the flow system occurred and when the polarimeter registered the change. The reason for this being that a portion of the liquid volume was contained within the flow cell at any given point during the test. Also, there was a slight initial over/under-shoot in the polarimetric signal prior to reaching equilibrium of the concentration. However, it can be seen that the two minute intervals allow adequate time for this to occur before dispensing the next change. These results further support the functionality of the flow system and the ability of the polarimeter to
keep up with such rapid changes. Therefore, the flow system can be used to create a wide range of different profiles in a controlled manner, even during a worst-case situation.

This can be utilized in order to validate polarimetric systems such as the IFMC without the need to follow through with multiple expensive and time consuming animal experiments.
Chapter 5

Conclusions and Future Work

The purpose of the work presented in this thesis was threefold. The first objective was to develop a FEM that could be used to design and optimize a customizable IFMC prototype for use in any application that requires sub-millidegree rotational measurement sensitivity. Second, it was shown that separate Faraday components are not needed in order to achieve the necessary rotational sensitivity. Therefore, an IFMC which utilized a single TDG rod and off-the-shelf inductors was designed and optimized based on the FEM as a step toward overall size and cost reduction in the development of a noninvasive polarimetric glucose monitor. Finally, a continuous and programmable glucose flow system was developed, integrated with the polarimetric glucose detection system, and tested against actual physiological glucose profiles in the creation of a controllable in vitro testing model for real-time, continuous sensing.

In order to achieve the first objective, a model was developed in the AC/DC Module of COMSOL Multiphysics and it was shown to accurately predict magnetic field magnitude throughout space based on the Biot-Savart Law. The FEM was then expanded to show that the magnetic field predicted within complex inductor models could be correlated to physical rotational measurements in a Faraday-based polarimetric system in
both compensation and modulation with errors as low as 3.26% and 0.71%, respectively. The primary source of error was based on the rotational accuracy of the mechanical stage used to collect the measurement data and the overall component fixturing, rather than discrepancies in the model.

For the second objective, data collected from the FEM was used to design and optimize an IFMC based on the requirements of a polarimetric physiological glucose sensor. The final design was implemented into a collective FEM and the results indicate that the rotational requirements were successfully met. To confirm this analysis, a customizable prototype of the model was also built and tested for depth in compensation and modulation, resulting in errors of 1.79% and 1.44%, respectively. The prototype was then validated in standard static conditions as well as through a unique dynamic arrangement. The dynamic testing was performed with a custom designed glucose flow system which was integrated with the polarimetric control software in order to monitor a physiological glucose profile continuously. The results indicate that not only does that IFMC meet the minimum design requirements for physiological detection, but it also maintains a degree of sensitivity similar to, if not better than, previously tested multi-component configurations with overall SEPs of 1.8 mg/dL and 5.4 mg/dL in the hypoglycemic and hyperglycemic concentration ranges, respectively.

The final objective was achieved by designing a custom flow system which was shown to accurately control glucose concentrations and could be programmed with various glucose profiles such as the human ultradian oscillations with insulin or actual clinical diabetic data. It was also shown that the system was capable of maintaining rapidly changing glucose profiles on the order of 27.5 mg/dL/min, which is much faster
than the typical expectations of physiological systems. Ultimately, when the flow system was coupled to the feedback loop of the polarimeter, glucose concentrations could be monitored continuously over long periods of time with an average prediction error of 7.5 mg/dL. However, vibrational movements over time caused slight changes to the polarization state of the detected signal, leading to overall baseline drift. Eventually, this drift accumulated to an average MAE of 10.0 mg/dL which was reduced through periodic single-point baseline corrections. The final SEP and MAE were calculated to be 3.8 mg/dL and 2.7 mg/dL, respectively. These results indicate that the IFMC is capable of monitoring continuous glucose changes with high precision but baseline corrections may be required to eliminate drift and increase accuracy without requiring recalibration of the system. This procedure is currently used in many commercialized glucose sensors and would be simple to implement in a diabetic meter before and after measurements are taken. However, the drift itself can be minimized by more rigidly coupling the optical components together and eliminating excess vibration in a compact, hand-held device. Also, in secondary applications that require long term monitoring, such as cell culture bioreactors, single point baseline corrections can be implemented to correct any inherent drift.

Overall, a FEM was developed that is capable of being used to design and optimize components as Faraday modulators and compensators in polarimetric systems. Also, a prototype was built to demonstrate the benefit of using such a model when designing a system for a specific application. The final design incorporated the separate components into a single hand-held device, eliminating the need for multiple TDG rods and custom designed inductive coils. The 3D printed housing used to fix the components
together was designed such that various inserts and spacers could be used when different coils are required for specific applications. The combination of the FEM with an easily customizable design allows for a full range of products to be made for several applications based on their rotational requirements. Also, integrating the modulator and compensator together into one hand-held device saves on cost and size which is beneficial in applications such as noninvasive glucose monitoring. Therefore, it can be seen that the design of the FEM can facilitate the commercialization of IFMCs to be used across all areas of optical polarimetry, eliminating the need to design and build custom systems.

Finally, a novel IFMC design was successfully demonstrated as a replacement to the multi-component noninvasive polarimetric glucose meter. Such a device is beneficial in personalized glucose monitors for diabetic patients in that it consolidates size and cost requirements and can achieve the same effect as its larger two-part predecessors. The design was validated under continuous monitoring conditions, providing a robust testing system for non in vivo experiments. The results of this work were also presented at the 2013 SPIE Photonics West Conference as shown in the conference proceedings.115, 116

Future research has the potential to combine the glucose flow system with an artificial eye model93 for recreating physiological conditions, such as concentration profiles and corneal birefringence, in a controlled manner. The availability of such a controlled system would provide a simple means for future development and optimization of polarimetric components as an alternative to expensive and time consuming animal testing. Also, the IFMC could be implemented into multispectral systems for further component consolidation in multi-analyte or birefringent conditions. This step in the
development process provides a good foundation for movement toward a commercially viable and manufacturable noninvasive glucose meter to ultimately reduce associated healthcare costs and improve the condition of those stricken with diabetes.
References


[78] Cameron, B. D., Rakovic, M. J., Mehrübeoglu, M., Kattawar, G. W., Rastegar, S., Wang, L. V. and Coté, G. L., “Measurement and calculation of the two-


Appendix A

Coilcraft Inductor Specifications

Figure A-1: Coilcraft inductor sheet 1.
<table>
<thead>
<tr>
<th>Part number</th>
<th>Inductance nom.</th>
<th>Percent tolerance</th>
<th>SRF type (MHz)</th>
<th>DCR max (Ohms)</th>
<th>Isat (A)</th>
<th>Irms (A)</th>
</tr>
</thead>
<tbody>
<tr>
<td>PCH-27X-392KLT</td>
<td>3.9 μH</td>
<td>10</td>
<td>36</td>
<td>0.017</td>
<td>7.01</td>
<td>1.7</td>
</tr>
<tr>
<td>PCH-27X-472KLT</td>
<td>4.7 μH</td>
<td>10</td>
<td>28</td>
<td>0.024</td>
<td>5.17</td>
<td>1.3</td>
</tr>
<tr>
<td>PCH-27X-562KLT</td>
<td>5.6 μH</td>
<td>10</td>
<td>25</td>
<td>0.020</td>
<td>5.79</td>
<td>1.3</td>
</tr>
<tr>
<td>PCH-27X-652KLT</td>
<td>6.5 μH</td>
<td>10</td>
<td>25</td>
<td>0.026</td>
<td>5.20</td>
<td>1.3</td>
</tr>
<tr>
<td>PCH-27X-752KLT</td>
<td>7.5 μH</td>
<td>10</td>
<td>22</td>
<td>0.030</td>
<td>4.20</td>
<td>1.3</td>
</tr>
<tr>
<td>PCH-27X-852KLT</td>
<td>8.2 μH</td>
<td>10</td>
<td>17</td>
<td>0.033</td>
<td>4.30</td>
<td>1.3</td>
</tr>
<tr>
<td>PCH-27X-100KLT</td>
<td>10 μH</td>
<td>10.5</td>
<td>16</td>
<td>0.037</td>
<td>3.00</td>
<td>1.3</td>
</tr>
<tr>
<td>PCH-27X-120KLT</td>
<td>12 μH</td>
<td>10.5</td>
<td>12</td>
<td>0.031</td>
<td>3.31</td>
<td>1.3</td>
</tr>
<tr>
<td>PCH-27X-150KLT</td>
<td>15 μH</td>
<td>10.5</td>
<td>11</td>
<td>0.028</td>
<td>3.29</td>
<td>1.3</td>
</tr>
<tr>
<td>PCH-27X-220KLT</td>
<td>22 μH</td>
<td>10.5</td>
<td>10</td>
<td>0.064</td>
<td>2.52</td>
<td>1.3</td>
</tr>
<tr>
<td>PCH-27X-270KLT</td>
<td>27 μH</td>
<td>10.5</td>
<td>7</td>
<td>0.071</td>
<td>2.59</td>
<td>1.3</td>
</tr>
<tr>
<td>PCH-27X-330KLT</td>
<td>33 μH</td>
<td>10.5</td>
<td>7</td>
<td>0.076</td>
<td>2.54</td>
<td>1.3</td>
</tr>
<tr>
<td>PCH-27X-390KLT</td>
<td>39 μH</td>
<td>10.5</td>
<td>6</td>
<td>0.094</td>
<td>1.88</td>
<td>1.3</td>
</tr>
<tr>
<td>PCH-27X-470KLT</td>
<td>47 μH</td>
<td>10.5</td>
<td>6</td>
<td>0.12</td>
<td>1.74</td>
<td>1.3</td>
</tr>
<tr>
<td>PCH-27X-680KLT</td>
<td>68 μH</td>
<td>10.5</td>
<td>7</td>
<td>0.16</td>
<td>1.57</td>
<td>0.67</td>
</tr>
<tr>
<td>PCH-27X-820KLT</td>
<td>82 μH</td>
<td>10.5</td>
<td>4</td>
<td>0.22</td>
<td>1.41</td>
<td>0.52</td>
</tr>
<tr>
<td>PCH-27X-100KLT</td>
<td>100 μH</td>
<td>10.5</td>
<td>3</td>
<td>0.26</td>
<td>1.38</td>
<td>0.52</td>
</tr>
<tr>
<td>PCH-27X-120KLT</td>
<td>120 μH</td>
<td>10.5</td>
<td>3</td>
<td>0.35</td>
<td>1.17</td>
<td>0.42</td>
</tr>
<tr>
<td>PCH-27X-150KLT</td>
<td>150 μH</td>
<td>10.5</td>
<td>3</td>
<td>0.39</td>
<td>1.04</td>
<td>0.42</td>
</tr>
<tr>
<td>PCH-27X-180KLT</td>
<td>180 μH</td>
<td>10.5</td>
<td>3</td>
<td>0.43</td>
<td>0.943</td>
<td>0.42</td>
</tr>
<tr>
<td>PCH-27X-220KLT</td>
<td>220 μH</td>
<td>10.5</td>
<td>3</td>
<td>0.47</td>
<td>0.848</td>
<td>0.42</td>
</tr>
<tr>
<td>PCH-27X-270KLT</td>
<td>270 μH</td>
<td>10.5</td>
<td>2</td>
<td>0.56</td>
<td>0.746</td>
<td>0.39</td>
</tr>
<tr>
<td>PCH-27X-330KLT</td>
<td>330 μH</td>
<td>10.5</td>
<td>2</td>
<td>0.72</td>
<td>0.600</td>
<td>0.33</td>
</tr>
<tr>
<td>PCH-27X-390KLT</td>
<td>390 μH</td>
<td>10.5</td>
<td>2</td>
<td>0.76</td>
<td>0.600</td>
<td>0.33</td>
</tr>
<tr>
<td>PCH-27X-470KLT</td>
<td>470 μH</td>
<td>10.5</td>
<td>2</td>
<td>1.04</td>
<td>0.573</td>
<td>0.26</td>
</tr>
<tr>
<td>PCH-27X-560KLT</td>
<td>560 μH</td>
<td>10.5</td>
<td>1</td>
<td>1.14</td>
<td>0.520</td>
<td>0.26</td>
</tr>
<tr>
<td>PCH-27X-680KLT</td>
<td>680 μH</td>
<td>10.5</td>
<td>1</td>
<td>1.49</td>
<td>0.472</td>
<td>0.21</td>
</tr>
<tr>
<td>PCH-27X-820KLT</td>
<td>820 μH</td>
<td>10.5</td>
<td>1</td>
<td>2.14</td>
<td>0.427</td>
<td>0.21</td>
</tr>
<tr>
<td>PCH-27X-100KLT</td>
<td>100 μH</td>
<td>10.5</td>
<td>1</td>
<td>2.36</td>
<td>0.395</td>
<td>0.16</td>
</tr>
</tbody>
</table>

1. Please specify sleeve, tolerance, termination and packaging codes.

2. Inductance tested at 15.75 kHz, 0.1 Vrms, 25°C ambient.
   
3. Tolerances in bold are stocked for immediate shipment.
   
4. DC current at which the inductance drops 10% (typ.) from its value without current.

5. Rated current based on 200 micromaxs per Ampere.

6. Operating temperature range: -40°C to +80°C.

7. Electrical specifications at 25°C.

8. Parts in bold type are included in Coilcraft Designer’s Kit No. P4699.

---

Figure A-2: Coilcraft inductor sheet 2.
# Axial Lead Power Chokes – PCH-45

<table>
<thead>
<tr>
<th>Part number</th>
<th>Inductance nom$^a$</th>
<th>Percent tolerance$^b$</th>
<th>SRF typ (MHz)</th>
<th>DCR max (Ohms)</th>
<th>Iext$^a$ (A)</th>
<th>Ims$^a$ (A)</th>
</tr>
</thead>
<tbody>
<tr>
<td>PCH-45X-392KLT</td>
<td>3.9 μH</td>
<td>10</td>
<td>34</td>
<td>0.038</td>
<td>14.3</td>
<td>4.3</td>
</tr>
<tr>
<td>PCH-45X-392KLT</td>
<td>3.9 μH</td>
<td>10</td>
<td>31</td>
<td>0.039</td>
<td>13.0</td>
<td>4.3</td>
</tr>
<tr>
<td>PCH-45X-662KLT</td>
<td>6.6 μH</td>
<td>10</td>
<td>29</td>
<td>0.039</td>
<td>11.0</td>
<td>4.3</td>
</tr>
<tr>
<td>PCH-45X-662KLT</td>
<td>6.6 μH</td>
<td>10</td>
<td>26</td>
<td>0.010</td>
<td>9.6</td>
<td>4.3</td>
</tr>
<tr>
<td>PCH-45X-822KLT</td>
<td>8.2 μH</td>
<td>10</td>
<td>29</td>
<td>0.011</td>
<td>9.4</td>
<td>4.3</td>
</tr>
<tr>
<td>PCH-45X-103KLT</td>
<td>10 μH</td>
<td>10</td>
<td>20</td>
<td>0.016</td>
<td>8.4</td>
<td>4.3</td>
</tr>
<tr>
<td>PCH-45X-123KLT</td>
<td>12 μH</td>
<td>10</td>
<td>17</td>
<td>0.013</td>
<td>7.5</td>
<td>4.3</td>
</tr>
<tr>
<td>PCH-45X-153KLT</td>
<td>15 μH</td>
<td>10</td>
<td>14</td>
<td>0.016</td>
<td>6.5</td>
<td>4.3</td>
</tr>
<tr>
<td>PCH-45X-203KLT</td>
<td>20 μH</td>
<td>10</td>
<td>11</td>
<td>0.016</td>
<td>6.5</td>
<td>4.3</td>
</tr>
<tr>
<td>PCH-45X-253KLT</td>
<td>25 μH</td>
<td>10</td>
<td>10</td>
<td>0.017</td>
<td>6.3</td>
<td>4.3</td>
</tr>
<tr>
<td>PCH-45X-273KLT</td>
<td>27 μH</td>
<td>10</td>
<td>9</td>
<td>0.019</td>
<td>6.0</td>
<td>4.3</td>
</tr>
<tr>
<td>PCH-45X-323KLT</td>
<td>32 μH</td>
<td>10</td>
<td>9</td>
<td>0.021</td>
<td>5.5</td>
<td>4.3</td>
</tr>
<tr>
<td>PCH-45X-393KLT</td>
<td>39 μH</td>
<td>10</td>
<td>8</td>
<td>0.025</td>
<td>4.8</td>
<td>4.3</td>
</tr>
<tr>
<td>PCH-45X-473KLT</td>
<td>47 μH</td>
<td>10</td>
<td>7</td>
<td>0.034</td>
<td>4.2</td>
<td>4.3</td>
</tr>
<tr>
<td>PCH-45X-563KLT</td>
<td>56 μH</td>
<td>10</td>
<td>6</td>
<td>0.037</td>
<td>3.7</td>
<td>4.3</td>
</tr>
<tr>
<td>PCH-45X-683KLT</td>
<td>68 μH</td>
<td>10</td>
<td>5</td>
<td>0.044</td>
<td>3.7</td>
<td>4.3</td>
</tr>
<tr>
<td>PCH-45X-823KLT</td>
<td>82 μH</td>
<td>10</td>
<td>4</td>
<td>0.060</td>
<td>2.9</td>
<td>1.7</td>
</tr>
<tr>
<td>PCH-45X-103KLT</td>
<td>100 μH</td>
<td>10</td>
<td>3</td>
<td>0.066</td>
<td>2.6</td>
<td>1.7</td>
</tr>
<tr>
<td>PCH-45X-124KLT</td>
<td>120 μH</td>
<td>10</td>
<td>3</td>
<td>0.095</td>
<td>2.5</td>
<td>1.7</td>
</tr>
<tr>
<td>PCH-45X-154KLT</td>
<td>150 μH</td>
<td>10</td>
<td>3</td>
<td>0.146</td>
<td>2.4</td>
<td>1.7</td>
</tr>
<tr>
<td>PCH-45X-184KLT</td>
<td>180 μH</td>
<td>10</td>
<td>2</td>
<td>0.165</td>
<td>2.6</td>
<td>1.7</td>
</tr>
<tr>
<td>PCH-45X-224KLT</td>
<td>220 μH</td>
<td>10</td>
<td>2</td>
<td>0.17</td>
<td>2.0</td>
<td>1.7</td>
</tr>
<tr>
<td>PCH-45X-275KLT</td>
<td>275 μH</td>
<td>10</td>
<td>2</td>
<td>0.185</td>
<td>1.9</td>
<td>1.7</td>
</tr>
<tr>
<td>PCH-45X-335KLT</td>
<td>335 μH</td>
<td>10</td>
<td>2</td>
<td>0.21</td>
<td>1.9</td>
<td>1.7</td>
</tr>
<tr>
<td>PCH-45X-404KLT</td>
<td>400 μH</td>
<td>10</td>
<td>1</td>
<td>0.26</td>
<td>1.9</td>
<td>1.7</td>
</tr>
<tr>
<td>PCH-45X-474KLT</td>
<td>4.7 μH</td>
<td>10</td>
<td>1</td>
<td>0.33</td>
<td>1.9</td>
<td>1.7</td>
</tr>
<tr>
<td>PCH-45X-564KLT</td>
<td>5.6 μH</td>
<td>10</td>
<td>1</td>
<td>0.39</td>
<td>1.9</td>
<td>1.7</td>
</tr>
<tr>
<td>PCH-45X-824KLT</td>
<td>8.2 μH</td>
<td>10</td>
<td>1</td>
<td>0.51</td>
<td>1.9</td>
<td>1.7</td>
</tr>
<tr>
<td>PCH-45X-105KLT</td>
<td>10 μH</td>
<td>10</td>
<td>1</td>
<td>0.69</td>
<td>1.9</td>
<td>1.7</td>
</tr>
<tr>
<td>PCH-45X-125KLT</td>
<td>12 μH</td>
<td>10</td>
<td>1</td>
<td>0.78</td>
<td>1.9</td>
<td>1.7</td>
</tr>
<tr>
<td>PCH-45X-155KLT</td>
<td>15 μH</td>
<td>10</td>
<td>0.75</td>
<td>0.92</td>
<td>0.75</td>
<td>0.67</td>
</tr>
<tr>
<td>PCH-45X-205KLT</td>
<td>20 μH</td>
<td>10</td>
<td>0.70</td>
<td>1.37</td>
<td>0.60</td>
<td>0.42</td>
</tr>
<tr>
<td>PCH-45X-275KLT</td>
<td>27 μH</td>
<td>10</td>
<td>0.85</td>
<td>2.28</td>
<td>0.54</td>
<td>0.33</td>
</tr>
<tr>
<td>PCH-45X-335KLT</td>
<td>33 μH</td>
<td>10</td>
<td>0.80</td>
<td>2.39</td>
<td>0.48</td>
<td>0.33</td>
</tr>
<tr>
<td>PCH-45X-495KLT</td>
<td>49 μH</td>
<td>10</td>
<td>0.56</td>
<td>2.94</td>
<td>0.40</td>
<td>0.33</td>
</tr>
<tr>
<td>PCH-45X-475KLT</td>
<td>4.7 μH</td>
<td>10</td>
<td>0.50</td>
<td>3.22</td>
<td>0.41</td>
<td>0.33</td>
</tr>
<tr>
<td>PCH-45X-565KLT</td>
<td>5.6 μH</td>
<td>10</td>
<td>0.40</td>
<td>4.25</td>
<td>0.37</td>
<td>0.26</td>
</tr>
<tr>
<td>PCH-45X-685KLT</td>
<td>6.8 μH</td>
<td>10</td>
<td>0.36</td>
<td>5.88</td>
<td>0.34</td>
<td>0.26</td>
</tr>
<tr>
<td>PCH-45X-825KLT</td>
<td>8.2 μH</td>
<td>10</td>
<td>0.30</td>
<td>6.46</td>
<td>0.31</td>
<td>0.21</td>
</tr>
<tr>
<td>PCH-45X-106KLT</td>
<td>10 μH</td>
<td>10</td>
<td>0.30</td>
<td>9.10</td>
<td>0.28</td>
<td>0.18</td>
</tr>
<tr>
<td>PCH-45X-126KLT</td>
<td>12 μH</td>
<td>10</td>
<td>0.30</td>
<td>9.10</td>
<td>0.28</td>
<td>0.18</td>
</tr>
<tr>
<td>PCH-45X-156KLT</td>
<td>15 μH</td>
<td>10</td>
<td>0.30</td>
<td>9.10</td>
<td>0.28</td>
<td>0.18</td>
</tr>
<tr>
<td>PCH-45X-196KLT</td>
<td>19 μH</td>
<td>10</td>
<td>0.20</td>
<td>21.2</td>
<td>0.21</td>
<td>0.10</td>
</tr>
<tr>
<td>PCH-45X-226KLT</td>
<td>22 μH</td>
<td>10</td>
<td>0.20</td>
<td>16.0</td>
<td>0.19</td>
<td>0.13</td>
</tr>
<tr>
<td>PCH-45X-276KLT</td>
<td>27 μH</td>
<td>10</td>
<td>0.15</td>
<td>18.6</td>
<td>0.17</td>
<td>0.13</td>
</tr>
<tr>
<td>PCH-45X-336KLT</td>
<td>33 μH</td>
<td>10</td>
<td>0.15</td>
<td>24.7</td>
<td>0.15</td>
<td>0.10</td>
</tr>
<tr>
<td>PCH-45X-396KLT</td>
<td>39 μH</td>
<td>10</td>
<td>0.15</td>
<td>29.0</td>
<td>0.14</td>
<td>0.10</td>
</tr>
<tr>
<td>PCH-45X-476KLT</td>
<td>47 μH</td>
<td>10</td>
<td>0.10</td>
<td>61.2</td>
<td>0.12</td>
<td>0.10</td>
</tr>
<tr>
<td>PCH-45X-566KLT</td>
<td>56 μH</td>
<td>10</td>
<td>0.10</td>
<td>42.5</td>
<td>0.119</td>
<td>0.032</td>
</tr>
<tr>
<td>PCH-45X-686KLT</td>
<td>68 μH</td>
<td>10</td>
<td>0.10</td>
<td>46.9</td>
<td>0.138</td>
<td>0.085</td>
</tr>
<tr>
<td>PCH-45X-826KLT</td>
<td>82 μH</td>
<td>10</td>
<td>0.10</td>
<td>64.9</td>
<td>0.128</td>
<td>0.087</td>
</tr>
<tr>
<td>PCH-45X-107KLT</td>
<td>100 μH</td>
<td>10</td>
<td>0.10</td>
<td>71.7</td>
<td>0.099</td>
<td>0.087</td>
</tr>
</tbody>
</table>

1. Please specify sleeve, tolerance, termination, and packaging codes.

   **PCH-45X-10P**

<table>
<thead>
<tr>
<th>Sleeve</th>
<th>Tolerance</th>
<th>Termination</th>
<th>Packaging</th>
</tr>
</thead>
<tbody>
<tr>
<td>X</td>
<td>J</td>
<td>L</td>
<td>R</td>
</tr>
</tbody>
</table>

2. Inductance tested at 15.75 kHz, 0.1 Vrms, 25°C ambient.
3. Tolerances in bold are stocked for immediate shipment.
4. DC current at which the inductance drops 10% (typ) from its value without current.
5. Rated current based on 200 circular millimeters per Amp.
6. Operating temperature range –40°C to +85°C.
7. Electrical specifications at 25°C.
8. Parts in bold type are included in Coilcraft’s Design Sheet No. P409.

---

**Figure A-3:** Coilcraft inductor sheet 3.
Appendix B

Digi-Key Inductor Specifications

Figure B-1: Digi-Key inductor sheet, 68 mH.
Figure B-2: Digi-Key inductor sheet, 220 mH.
Figure B-3: Digi-Key inductor sheet, 470 mH.
Appendix C

IFMC Electronic Components

Figure C-1: Digi-Key round connector sheet, female.
Figure C-2: Digi-Key round connector sheet, male.

<table>
<thead>
<tr>
<th>Datasheets</th>
<th>MD-60</th>
</tr>
</thead>
<tbody>
<tr>
<td>Product Photos</td>
<td>MD50-90</td>
</tr>
<tr>
<td>RoHS Information</td>
<td>MD-60 Cert of Compliance</td>
</tr>
<tr>
<td>Catalog Drawings</td>
<td>MD Series Plug-Front, MD Series Plug-Cable End, MD Series Plug-Side</td>
</tr>
<tr>
<td>3D Model</td>
<td>MD-60</td>
</tr>
<tr>
<td>Standard Package</td>
<td>100</td>
</tr>
<tr>
<td>Connector Type</td>
<td>Plug, Male Pins</td>
</tr>
<tr>
<td>Number of Positions</td>
<td>6</td>
</tr>
<tr>
<td>Shell Size - Insert</td>
<td>Mini DIN</td>
</tr>
<tr>
<td>Shell Size, MIL</td>
<td></td>
</tr>
<tr>
<td>Mounting Type</td>
<td>Free Hanging (In-Line)</td>
</tr>
<tr>
<td>Termination</td>
<td>Solder, Crimp Pin Adapter Available</td>
</tr>
<tr>
<td>Fastening Type</td>
<td>Push-Pull</td>
</tr>
<tr>
<td>Orientation</td>
<td>Keyed</td>
</tr>
<tr>
<td>Ingress Protection</td>
<td></td>
</tr>
<tr>
<td>Shell Material, Finish</td>
<td>Plastic</td>
</tr>
<tr>
<td>Contact Finish</td>
<td>Silver</td>
</tr>
<tr>
<td>Features</td>
<td>Shielded</td>
</tr>
<tr>
<td>Packaging</td>
<td>Shell</td>
</tr>
<tr>
<td>Contact Finish Thickness</td>
<td></td>
</tr>
<tr>
<td>Current Rating</td>
<td>1A, 2A</td>
</tr>
<tr>
<td>Voltage - Rated</td>
<td>100VAC, 125VDC</td>
</tr>
<tr>
<td>Operating Temperature</td>
<td>20°C – 105°C</td>
</tr>
<tr>
<td>Dynamic Catalog</td>
<td>MD</td>
</tr>
</tbody>
</table>
Figure C-3: Linear Technology buffer circuit sheet.
Appendix D

IFMC PID Digital Control VI

Figure D-1: IFMC PID control VI front panel, full view.
Figure D-2: IFMC PID control VI block diagram, left view, true.

Figure D-3: IFMC PID control VI block diagram, left view, false.
Figure D-4: IFMC PID control VI block diagram, center view, true, false.

Figure D-5: IFMC PID control VI block diagram, center view, true, true.
Figure D-6: IFMC PID control VI block diagram, center view, false.

Figure D-7: IFMC PID control VI block diagram, right view, false.
Figure D-8: IFMC PID control VI block diagram, right view, true.
Appendix E

WELCO Peristaltic Pump Specifications

Figure E-1: WELCO peristaltic pump selection guide sheet 1.
WELCO Peristaltic pumps use a custom ordering system that enables part types and sizes to be selected according to the desired application.

Selection method for customization of pumps
Select the part number according to the following guide

Figure E-2: WELCO peristaltic pump selection guide sheet 2.
## Pump tube type: Material
(Selectable according to fluid type)

<table>
<thead>
<tr>
<th>Tube type</th>
<th>Product Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>TYGON 3355S</td>
<td>“Long service life silicon tubes” with excellent interior flatness</td>
</tr>
<tr>
<td></td>
<td>Meets USP Class VI, FDA, 3-A and NSF 51 criteria</td>
</tr>
<tr>
<td>TYGON 2075X</td>
<td>“Chemical-resistant tubes” that require excellent chemical resistance, transparency, and flexibility</td>
</tr>
<tr>
<td></td>
<td>Meets USP Class VI</td>
</tr>
<tr>
<td>TYGON 2275Y</td>
<td>“High purity tubes” suitable for sending easily-changeable liquids</td>
</tr>
<tr>
<td></td>
<td>Meets USP Class VI, FDA, 3-A and NSF 51 criteria</td>
</tr>
<tr>
<td>TYGON LFL</td>
<td>“High durability tubes” with excellent wear resistance</td>
</tr>
<tr>
<td></td>
<td>Meets USP Class VI, FDA, 3-A</td>
</tr>
<tr>
<td>PHARMED BPT</td>
<td>“Chemical manufacturing and bio-tubes” with long service life and excellent acid and alkali resistance</td>
</tr>
<tr>
<td></td>
<td>Meets USP Class VI, FDA, 3-A and NSF 51 criteria</td>
</tr>
<tr>
<td>Noprane A-60G</td>
<td>“Industrial tubes” with excellent weather resistance for a wide range of applications</td>
</tr>
<tr>
<td>F</td>
<td>“Fluorine tubes” that are resistant to corrosive chemicals, oils, and fuels, etc.</td>
</tr>
<tr>
<td>W Tube</td>
<td>“Dual wall tubes” that are resistant to chemicals and high temperature: Inner layers: Polyolefin Outer layers: Thermosetting Elastomers</td>
</tr>
</tbody>
</table>

Note: TYGON, Pharmed, Noprane and Fluran are manufactured by Saint-Gobain Group.

## Pump tube type: Tube size (inner diameter) (Selectable according to the tube material and number of rollers)

<table>
<thead>
<tr>
<th>Model name/inner diameter</th>
<th>1/16</th>
<th>3/32</th>
<th>1/8</th>
<th>3/16</th>
</tr>
</thead>
<tbody>
<tr>
<td>Inner diameter</td>
<td>1.6mm (1/16&quot;)</td>
<td>2.4mm (3/32&quot;)</td>
<td>3.2mm (1/8&quot;)</td>
<td>4.8mm (3/16&quot;)</td>
</tr>
<tr>
<td>Available tube material</td>
<td>P</td>
<td>S / P</td>
<td>All type (6)</td>
<td>X / Y Not suitable</td>
</tr>
<tr>
<td>Number of rollers</td>
<td>2 / 4</td>
<td>2 / 4</td>
<td>2 / 4</td>
<td>2 / 4</td>
</tr>
</tbody>
</table>

Caution: Tube type F4.9 cannot be used with four rollers due to its high hardness.

## Flow amount benchmark (flow amount per rotation)

<table>
<thead>
<tr>
<th>Inside diameter of tube (inches)</th>
<th>1.6mm (1/16&quot;)</th>
<th>2.4mm (3/32&quot;)</th>
<th>3.2mm (1/8&quot;)</th>
<th>4.8mm (3/16&quot;)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Number of rollers</td>
<td>2</td>
<td>4</td>
<td>2</td>
<td>4</td>
</tr>
<tr>
<td>WPK1 Flow amount (mL)</td>
<td>0.17</td>
<td>0.15</td>
<td>0.35</td>
<td>0.6</td>
</tr>
</tbody>
</table>

Caution: The above table describes the initial benchmark flow amounts during water suction. This may vary considerably depending on the tube type, use period, ambient temperature, and lot tolerance, etc. Measure the specifications with reasonable certainty.

Figure E-3: WELCO peristaltic pump selection guide sheet 3.
Figure E-4: WELCO peristaltic pump selection guide sheet 4.
Figure E-5: WELCO peristaltic pump selection guide sheet 5.
Figure E-6: WELCO peristaltic pump selection guide sheet 6.
Figure E-7: WELCO peristaltic pump selection guide sheet 7.
Color variation
A 5-color lineup that can be classified for use according to the type of liquids used

B: Blue
G: Green
C: Clear
R: Red
Y: Yellow
UV: Black
(Special order item)

※There is also a lineup of panels to which the pump can be easily mounted

Using an optional panel
There is also a lineup of panels to which the pump can be easily mounted

P = with bracket
N or Blank = without bracket

General specifications

<table>
<thead>
<tr>
<th>Recommended installation height</th>
<th>2.0m max</th>
</tr>
</thead>
<tbody>
<tr>
<td>Liquid temperature range</td>
<td>5 to 50°C (41°F to 122°F)</td>
</tr>
<tr>
<td>Specified environment temperature range</td>
<td>0 to 50°C (32°F to 122°F)</td>
</tr>
<tr>
<td>Specified ambient humidity range</td>
<td>20% to 80% (with no condensation)</td>
</tr>
</tbody>
</table>

Certifications & Approvals
UL (E209554)
CE (EN50115 Class B Gc1/EN61000-6-2)
NSF (pump tube type: S/X/Y/U/P/N/F)
RoHS

Precautions
1. When selecting a tube, the customer should perform a verification test to verify the chemical suitability according to the usage environment and the intended application.
2. Regardless of the pump tube type, the phenomenon of peeling from inside of the tube starts with small amounts.
3. This product was not designed for medical use. Do not use for medical applications.
4. This product is not waterproof. If using in water-filled environments, design to protect against water.
5. Numerical data listed in this catalog reflect conditions measured over short periods of time. Their accuracy for long-term use is not assured.
6. There is a tendency for the flow rate to increase until the tube becomes saturated, and even among the same models, different lots may have different flow rates within the specified tolerances. Also, the rotating speed of the DC motor may fluctuate depending on the load conditions and changes in the motor temperature. During the design stage, be sure to select a motor with ample capacity.

Figure E-8: WELCO peristaltic pump selection guide sheet 8.
Appendix F

LIFA Firmware Code

/*******************************************************************************/
** LVFA_Firmware - Provides Basic Arduino Sketch For Interfacing With LabVIEW.**
** Written By:  Sam Kristoff - National Instruments**
** Written On:  November 2010**
** Last Updated: Mar 2013 - Brandon Clarke - University of Toledo**
** This File May Be Modified And Re-Distributed Freely. Original File Content**
** Written By Sam Kristoff And Available At www.ni.com/arduino.**
*******************************************************************************/

/*********************************************************************************/
** Includes.**
********************************************************************************/
// Standard includes. These should always be included.
#include <Wire.h>
#include <SPI.h>
#include <Servo.h>
#include "LabVIEWInterface.h"

/*********************************************************************************/
** setup()**
********************************************************************************/
** Initialize the Arduino and setup serial communication.**
** Input: None**
** Output: None**
*******************************************************************************/
void setup()
{
    // Initialize Serial Port With The Default Baud Rate
    syncLV();

    // Place your custom setup code here
void loop()
{
    // Check for commands from LabVIEW and process them.
    checkForCommand();

    // Place your custom loop code here (this may slow down communication with LabVIEW)

    if (acqMode == 1)
    {
        sampleContinously();
    }
}
Appendix G

LIFA Header Code

/*********************************************************************************/
**
**  LVFA_Firmware - Provides Functions For Interfacing With The Arduino Uno
**
**  Written By:    Sam Kristoff - National Instruments
**  Written On:    November 2010
**  Last Updated:  Mar 2013 - Brandon Clarke - University of Toledo
**
**  This File May Be Modified And Re-Distributed Freely. Original File Content
**  Written By Sam Kristoff And Available At www.ni.com/arduino.
**
*********************************************************************************/

/******************************************************************************
  **  Define Constants
  **  Define directives providing meaningful names for constant values.
*******************************************************************************/
#define FIRMWARE_MAJOR 02
#define FIRMWARE_MINOR 00
#if defined(__AVR_ATmega1280__) || defined(__AVR_ATmega2560__)
#define DEFAULTBAUDRATE 9600    // Defines The Default Serial Baud Rate (This must match the
  baud rate specified in LabVIEW)
#else
#define DEFAULTBAUDRATE 115200
#endif
#define MODE_DEFAULT 0
#define COMMANDLENGTH 15       // Defines The Number Of Bytes In A Single LabVIEW Command
  (This must match the packet size specified in LabVIEW)
#define STEPPER_SUPPORT 1      // Defines Whether The Stepper Library Is Included - Comment This
  Line To Exclude Stepper Support

// Declare Variables
unsigned char currentCommand[COMMANDLENGTH];    // The Current Command For The Arduino To Process

}
//Globals for continuous acquisition
unsigned char acqMode;
unsigned char contAcqPin;
float contAcqSpeed;
float acquisitionPeriod;
float iterationsFlt;
int iterations;
float delayTime;

/*************************--------------------------------------------------------*
** syncLV
**
** Synchronizes with LabVIEW and sends info about the board and firmware (Unimplemented)
**
** Input: None
** Output: None
******************************************************************************/
void syncLV();

/*************************--------------------------------------------------------*
** setMode
**
** Sets the mode of the Arduino (Reserved For Future Use)
**
** Input: Int - Mode
** Output: None
******************************************************************************/
void setMode(int mode);

/*************************--------------------------------------------------------*
** checkForCommand
**
** Checks for new commands from LabVIEW and processes them if any exists.
**
** Input: None
** Output: 1 - Command received and processed
** 0 - No new command
******************************************************************************/
int checkForCommand(void);

/*************************--------------------------------------------------------*
** processCommand
**
** Processes a given command
**
** Input: command of COMMANDLENGTH bytes
** Output: 1 - Command received and processed
** 0 - No new command
******************************************************************************/
void processCommand(unsigned char command[]);

/*************************--------------------------------------------------------*
** writeDigitalPort
**
** Write values to DIO pins 0 - 13. Pins must first be configured as outputs.
void writeDigitalPort(unsigned char command[]);

void analogReadPort();

void sevenSegment_Config(unsigned char command[]);

void sevenSegment_Write(unsigned char command[]);

void spi_setClockDivider(unsigned char divider);

void spi_sendReceive(unsigned char command[]);
/*******************************************************/
** checksum_Compute
**
** Compute Packet Checksum
**
** Input: Command Packet
** Output: Char Checksum Value
**************************************************************************
unsigned char checksum_Compute(unsigned char command[]);
**************************************************************************
** checksum_Test
**
** Compute Packet Checksum And Test Against Included Checksum
**
** Input: Command Packet
** Output: 0 If Checksums Are Equal, Else 1
**************************************************************************
int checksum_Test(unsigned char command[]);
**************************************************************************
** AccelStepper_Write
**
** Parse command packet and write speed, direction, and number of steps to travel
** Modified for Sparkfun Quadstepper Motor Driver Board
**
** Input: Command Packet
** Output: None
**************************************************************************
void AccelStepper_Write(unsigned char command[]);
**************************************************************************
** SampleContinously
**
** Returns several analog input points at once.
**
** Input: void
** Output: void
**************************************************************************
void sampleContinously(void);
**************************************************************************
** finiteAcquisition
**
** Returns the number of samples specified at the rate specified.
**
** Input: pin to sample on, speed to sample at, number of samples
** Output: void
**************************************************************************
void finiteAcquisition(int analogPin, float acquisitionSpeed, int numberOfSamples);
**************************************************************************
** lcd_print
**
** Prints Data to the LCD With The Given Base
**
** Input: Command Packet
** Output: None

154
void lcd_print(unsigned char command[]);
Appendix H

LIFA Source Code

/*---------------------------------------------------------------------------------------------
 **
 ** LVIFA_Firmware - Provides Functions For Interfacing With The Arduino Uno
 **
 ** Written By:    Sam Kristoff - National Instruments
 ** Written On:    November 2010
 ** Last Updated:  Mar 2013 - Brandon Clarke - University of Toledo
 **
 ** This File May Be Modified And Re-Distributed Freely. Original File Content
 ** Written By Sam Kristoff And Available At www.ni.com/arduino.
 **
 ** ------------------------------------------------------------------------------------------*/

#include <Wire.h>
#include <SPI.h>
#include <LiquidCrystal.h>

/*---------------------------------------------------------------------------------------------
 ** Optionally Include And Configure Stepper Support
 ** ------------------------------------------------------------------------------------------*/
#ifndef STEPPER_SUPPORT

// Stepper Modifications
#include "AFMotor.h"
#include "AccelStepper.h"

// Adafruit shield
AF_Stepper motor1(200, 1);
AF_Stepper motor2(200, 2);

// you can change these to DOUBLE or INTERLEAVE or MICROSTEP
// wrappers for the first motor
void forwardstep1() {
    motor1.onestep(FORWARD, SINGLE);
}
void backwardstep1() {
    motor1.onestep(BACKWARD, SINGLE);
}
// wrappers for the second motor
void forwardstep2() {
    motor2.onestep(FORWARD, SINGLE);
}
void backwardstep2() {
    motor2.onestep(BACKWARD, SINGLE);
}

AccelStepper steppers[8]; //Create array of 8 stepper objects

#endif

// Variables
unsigned int retVal;
int sevenSegmentPins[8];
int currentMode;
unsigned int freq;
unsigned long duration;
int i2cReadTimeouts = 0;
char spiBytesToSend = 0;
char spiBytesSent = 0;
char spiCSPin = 0;
char spiWordSize = 0;
Servo *servos;
byte customChar[8];
LiquidCrystal lcd(0,0,0,0,0,0,0);
int moveType[8]; // Added integer fir motor control
// Sets the mode of the Arduino (Reserved For Future Use)
void setMode(int mode)
{
    currentMode = mode;
}

// Checks for new commands from LabVIEW and processes them if any exists.
int checkForCommand(void)
{
    #ifdef STEPPER_SUPPORT
    // Call run function as fast as possible to keep motors turning.
    // Modified to implement continuous movement when moveType == 1.
    for (int i=0; i<8; i++){
        if (moveType[i] == 1){
            steppers[i].runSpeed();
        } else{
            steppers[i].run();
        }
    }
    #endif

    int bufferBytes = Serial.available();
    if(bufferBytes >= COMMANDLENGTH) {
        // New Command Ready, Process It
// Build Command From Serial Buffer
for(int i=0; i<COMMANDLENGTH; i++)
{
    currentCommand[i] = Serial.read();
}
processCommand(currentCommand);
return 1;
}
else
{
    return 0;
}

// Processes a given command
void processCommand(unsigned char command[])
{
    // Determine Command
    if(command[0] == 0xFF && checksum_Test(command) == 0)
    {
        switch(command[1])
        {
        /*************************************************************************/
        ** LIFA Maintenance Commands
        /*************************************************************************/
        case 0x00:    // Sync Packet
            Serial.print("sync");
            Serial.flush();
            break;
        case 0x01:    // Flush Serial Buffer
            Serial.flush();
            break;
        /*************************************************************************/
        ** Low Level - Digital I/O Commands
        /*************************************************************************/
        case 0x02:    // Set Pin As Input Or Output
            pinMode(command[2], command[3]);
            Serial.write('0');
            break;
        case 0x03:    // Write Digital Pin
            digitalWrite(command[2], command[3]);
            Serial.write('0');
            break;
        case 0x04:    // Write Digital Port 0
            writeDigitalPort(command);
            Serial.write('0');
            break;
        case 0x05:    //Tone
            freq = ( (command[3]<<8) + command[4]);
            if(freq > 0)
            {
                tone(command[2], freq, duration);
            }
else
{
    noTone(command[2]);
}
Serial.write('0');
break;
case 0x06:    // Read Digital Pin
    retVal = digitalRead(command[2]);
    Serial.write(retVal);
    break;
case 0x07:    // Digital Read Port
    retVal = 0x0000;
    for(int i=0; i <=13; i++)
    {
        if(digitalRead(i))
        {
            retVal += (1<<i);
        }
    }
    Serial.write((retVal & 0xFF));
    Serial.write((retVal >> 8));
    break;
/***********************
**************
** Low Level - Analog Commands
***********************************************************************/
case 0x08:    // Read Analog Pin
    retVal = analogRead(command[2]);
    Serial.write((retVal >> 8));
    Serial.write((retVal & 0xFF));
    break;
case 0x09:    // Analog Read Port
    analogReadPort();
    break;
/***********************
**************
** Low Level - PWM Commands
***********************************************************************/
case 0x0A:    // PWM Write Pin
    analogWrite(command[2], command[3]);
    Serial.write('0');
    break;
case 0x0B:    // PWM Write 3 Pins
    analogWrite(command[2], command[5]);
    analogWrite(command[3], command[6]);
    analogWrite(command[4], command[7]);
    Serial.write('0');
    break;
/***********************
**************
** Sensor Specific Commands
***********************************************************************/
case 0x0C:    // Configure Seven Segment Display
    sevenSegment_Config(command);
    Serial.write('0');
    break;
case 0x0D: // Write To Seven Segment Display
    sevenSegment_Write(command);
    Serial.write('0');
    break;

/*********************************************************************************
** I2C
*********************************************************************************/

case 0x0E: // Initialize I2C
    Wire.begin();
    Serial.write('0');
    break;

case 0x0F: // Send I2C Data
    Wire.beginTransmission(command[3]);
    for(int i=0; i<command[2]; i++)
    {
        #if defined(ARDUINO) && ARDUINO >= 100
            Wire.write(command[i+4]);
        #else
            Wire.send(command[i+4]);
        #endif
    }
    Wire.endTransmission();
    Serial.write('0');
    break;

case 0x10: // I2C Read
    i2cReadTimeouts = 0;
    Wire.requestFrom(command[3], command[2]);
    while(Wire.available() < command[2])
    {
        i2cReadTimeouts++;
        if(i2cReadTimeouts > 100)
        {
            return;
        }
        else
        {
            delay(1);
        }
    }
    for(int i=0; i<command[2]; i++)
    {
        #if defined(ARDUINO) && ARDUINO >= 100
            Serial.write(Wire.read());
        #else
            Serial.write(Wire.receive());
        #endif
    }
    break;

/*********************************************************************************
** SPI
*********************************************************************************/

160
case 0x11:  // SPI Init
SPI.begin();
Serial.write('0');
break;
case 0x12:  // SPI Set Bit Order (MSB LSB)
if(command[2] == 0)
{
    SPI.setBitOrder(LSBFIRST);
}
else
{
    SPI.setBitOrder(MSBFIRST);
}
Serial.write('0');
break;
case 0x13:  // SPI Set Clock Divider
spi_setClockDivider(command[2]);
Serial.write('0');
break;
case 0x14:  // SPI Set Data Mode
switch(command[2])
{
    case 0:
        SPI.setDataMode(SPI_MODE0);
        break;
    case 1:
        SPI.setDataMode(SPI_MODE1);
        break;
    case 2:
        SPI.setDataMode(SPI_MODE2);
        break;
    case 3:
        SPI.setDataMode(SPI_MODE3);
        break;
    default:
        break;
}
Serial.write('0');
break;
case 0x15:  // SPI Send / Receive
spi_sendReceive(command);
break;
case 0x16:  // SPI Close
SPI.end();
Serial.write('0');
break;

/*********************************************************************************
** Servos
*********************************************************************************/
case 0x17:  // Set Num Servos
free(servos);
servos = (Servo*) malloc(command[2]*sizeof(Servo));
for(int i=0; i<command[2]; i++)
{
    servos[i] = Servo();
if(servos == 0)
{
    Serial.write('1');
}
else
{
    Serial.write('0');
}
break;

} /* LCD */

/*********************************************************************************
**                                      LCD                                      
************************************************
*****************************************************************************/

}
case 0x18: // Configure Servo
    servos[command[2]].attach(command[3]);
    Serial.write('0');
    break;
case 0x19: // Servo Write
    servos[command[2]].write(command[3]);
    Serial.write('0');
    break;
case 0x1A: // Servo Read Angle
    Serial.write(servos[command[2]].read());
    break;
case 0x1B: // Servo Write uS Pulse
    servos[command[2]].writeMicroseconds( (command[3] + (command[4]<<8)) );
    Serial.write('0');
    break;
case 0x1C: // Servo Read uS Pulse
    retVal = servos[command[2]].readMicroseconds();
    Serial.write( (retVal & 0xFF));
    Serial.write( (retVal >> 8));
    break;
case 0x1D: // Servo Detach
    servos[command[2]].detach();
    Serial.write('0');
    break;

/*********************************************************************************
**                                      LCD                                      
************************************************
*****************************************************************************/

case 0x1E: // LCD Init
    lcd.init(command[2], command[3], command[4], command[5], command[6], command[7],
    command[8], command[9], command[10], command[11], command[12], command[13]);
    Serial.write('0');
    break;
case 0x1F: // LCD Set Size
    lcd.begin(command[2], command[3]);
    Serial.write('0');
    break;
case 0x20: // LCD Set Cursor Mode
    if(command[2] == 0)
    {
        lcd.noCursor();
    }
    else
    {
        lcd.cursor();
    }
if(command[3] == 0)
{
    lcd.noBlink();
}
else
{
    lcd.blink();
}
Serial.write('0');
break;
case 0x21:  // LCD Clear
    lcd.clear();
    Serial.write('0');
    break;
case 0x22:  // LCD Set Cursor Position
    lcd.setCursor(command[2], command[3]);
    Serial.write('0');
    break;
case 0x23:  // LCD Print
    lcd_print(command);
    break;
case 0x24:  // LCD Display Power
    if(command[2] == 0)
    {
        lcd.noDisplay();
    }
    else
    {
        lcd.display();
    }
    Serial.write('0');
    break;
case 0x25:  // LCD Scroll
    if(command[2] == 0)
    {
        lcd.scrollDisplayLeft();
    }
    else
    {
        lcd.scrollDisplayRight();
    }
    Serial.write('0');
    break;
case 0x26:  // LCD Autoscroll
    if(command[2] == 0)
    {
        lcd.noAutoscroll();
    }
    else
    {
        lcd.autoscroll();
    }
    Serial.write('0');
    break;
case 0x27:  // LCD Print Direction
    if(command[2] == 0)
{  
    lcd.rightToLeft();  
}  
else  
{  
    lcd.leftToRight();  
}  
Serial.write('0');  
break;  
case 0x28: // LCD Create Custom Char  
    for(int i=0; i<8; i++)  
    {  
        customChar[i] = command[i+3];  
    }  
    lcd.createChar(command[2], customChar);  
    Serial.write('0');  
    break;  
case 0x29: // LCD Print Custom Char  
    lcd.write(command[2]);  
    Serial.write('0');  
    break;  
/** Continuos Aquisition  
********************************************/  
  
case 0x2A: // Continuos Aquisition Mode On  
    acqMode=1;  
    contAcqPin=command[2];  
    contAcqSpeed=(command[3])+(command[4]<<8);  
    acquisitionPeriod=1/contAcqSpeed;  
    iterationsFlt =.08/acquisitionPeriod;  
    iterations=(int)iterationsFlt;  
    if(iterations<1)  
    {  
        iterations=1;  
    }  
    delayTime= acquisitionPeriod;  
    if(delayTime<0)  
    {  
        delayTime=0;  
    }  
    break;  
case 0x2B: // Continuos Aquisition Mode Off  
    acqMode=0;  
    break;  
case 0x2C: // Return Firmware Revision  
    Serial.write(byte(FIRMWARE_MAJOR));  
    Serial.write(byte(FIRMWARE_MINOR));  
    break;  
case 0x2D: // Perform Finite Aquisition  
    Serial.write('0');  
    break;  
/*****************************************************************************/
** Stepper, Modified to function with Sparkfun Quadstepper Motor Driver Board

#ifdef STEPPER_SUPPORT

    case 0x30:  // Configure Stepper
        if (command[2] == 5){  // Support AFMotor Shield
            switch (command[3]){
                case 0:
                    steppers[command[3]] = AccelStepper(forwardstep1, backwardstep1);
                    break;
                case 1:
                    steppers[command[3]] = AccelStepper(forwardstep2, backwardstep2);
                    break;
                default:
                    break;
            }
        // Modified to utilize enable and step size pins with Quadstepper:
        else if(command[2]==6) {                   // All other stepper configurations
            steppers[command[3]] = AccelStepper(1, command[4], command[5]);
            steppers[command[3]].setEnablePin(command[6]);
            steppers[command[3]].setPinsInverted(0, 0, 1);
            pinMode(command[7], OUTPUT);
            pinMode(command[8], OUTPUT);
            pinMode(command[9], OUTPUT);
            digitalWrite(command[7], LOW);
            digitalWrite(command[8], LOW);
            digitalWrite(command[9], LOW);
        }
        else{
            steppers[command[3]] = AccelStepper(command[2], command[4],command[5]);
            steppers[command[3]].setEnablePin(command[6]);
            steppers[command[3]].setPinsInverted(0, 0, 1);
            pinMode(command[7], OUTPUT);
            pinMode(command[8], OUTPUT);
            pinMode(command[9], OUTPUT);
            digitalWrite(command[7], LOW);
            digitalWrite(command[8], LOW);
            digitalWrite(command[9], LOW);
        }
        Serial.write('0');
        break;
    case 0x31:  // Stepper Write
        AccelStepper_Write(command);
        Serial.write('0');
        break;
    case 0x32:  // Stepper Detach
        //if (steppers[command[2]].distanceToGo()==0) {
        //    steppers[command[2]].disableOutputs();
        //}
        Serial.write('0');
        break;
    case 0x33:  // Stepper steps to go
        retVal = 0;
        //for(int i=0; i<8; i++){
        //    retVal += steppers[command[3]].distanceToGo();
        //}

#endif
Serial.write((retVal & 0xFF));
Serial.write((retVal >> 8));
break;

case 0x34: // Disable all stepper outputs (Quadstepper)
    for(int i=0; i<8; i++){
        steppers[i].disableOutputs();
    }
    Serial.write('0');
#endif

/*********************************************************************************
** Unknown Packet
*********************************************************************************/
default:      // Default Case
    Serial.flush();
    break;
}
}
else{
    // Checksum Failed, Flush Serial Buffer
    Serial.flush();
}

/*********************************************************************************
** Functions
*********************************************************************************/

// Writes Values To Digital Port (DIO 0-13). Pins Must Be Configured As Outputs Before Being Written To
void writeDigitalPort(unsigned char command[])
{
    digitalWrite(13, ((command[2] >> 5) & 0x01));
    digitalWrite(12, ((command[2] >> 4) & 0x01));
    digitalWrite(11, ((command[2] >> 3) & 0x01));
    digitalWrite(10, ((command[2] >> 2) & 0x01));
    digitalWrite(9, ((command[2] >> 1) & 0x01));
    digitalWrite(8, (command[2] & 0x01));
    digitalWrite(7, ((command[3] >> 7) & 0x01));
    digitalWrite(6, ((command[3] >> 6) & 0x01));
    digitalWrite(5, ((command[3] >> 5) & 0x01));
    digitalWrite(4, ((command[3] >> 4) & 0x01));
    digitalWrite(3, ((command[3] >> 3) & 0x01));
    digitalWrite(2, ((command[3] >> 2) & 0x01));
    digitalWrite(1, ((command[3] >> 1) & 0x01));
    digitalWrite(0, (command[3] & 0x01));
}

// Reads all 6 analog input ports, builds 8 byte packet, send via RS232.
void analogReadPort()
{
    // Read Each Analog Pin
    int pin0 = analogRead(0);
    int pin1 = analogRead(1);
    int pin2 = analogRead(2);
int pin3 = analogRead(3);
int pin4 = analogRead(4);
int pin5 = analogRead(5);

// Build 8-Byte Packet From 60 Bits of Data Read
char output0 = (pin0 & 0xFF);
char output1 = ((pin1 << 2) & 0xFC) | ((pin0 >> 8) & 0x03);
char output2 = ((pin2 << 4) & 0xF0) | ((pin1 >> 6) & 0x0F);
char output3 = ((pin3 << 6) & 0xC0) | ((pin2 >> 4) & 0x3F);
char output4 = ((pin3 >> 2) & 0xFF);
char output5 = (pin4 & 0xFF);
char output6 = ((pin5 << 2) & 0xFC) | ((pin4 >> 8) & 0x03);
char output7 = (pin5 >> 6);

// Write Bytes To Serial Port
Serial.print(output0);
Serial.print(output1);
Serial.print(output2);
Serial.print(output3);
Serial.print(output4);
Serial.print(output5);
Serial.print(output6);
Serial.print(output7);
}

// Configure digital I/O pins to use for seven segment display
void sevenSegment_Config(unsigned char command[])
{
  // Configure pins as outputs and store in sevenSegmentPins array for use in sevenSegment_Write
  for(int i=2; i<10; i++)
  {
    pinMode(command[i], OUTPUT);
    sevenSegmentPins[(i-1)] = command[i];
  }
}

// Write values to sevenSegment display. Must first use sevenSegment_Config
void sevenSegment_Write(unsigned char command[])
{
  for(int i=1; i<9; i++)
  {
    digitalWrite(sevenSegmentPins[(i-1)], command[i]);
  }
}

// Set the SPI Clock Divisor
void spi_setClockDivider(unsigned char divider)
{
  switch(divider)
  {
    case 0:
      SPI.setClockDivider(SPI_CLOCK_DIV2);
      break;
    case 1:
      SPI.setClockDivider(SPI_CLOCK_DIV4);
      break;
  }
case 2:
    SPI.setClockDivider(SPI_CLOCK_DIV8);
    break;
case 3:
    SPI.setClockDivider(SPI_CLOCK_DIV16);
    break;
case 4:
    SPI.setClockDivider(SPI_CLOCK_DIV32);
    break;
case 5:
    SPI.setClockDivider(SPI_CLOCK_DIV64);
    break;
case 6:
    SPI.setClockDivider(SPI_CLOCK_DIV128);
    break;
default:
    SPI.setClockDivider(SPI_CLOCK_DIV4);
    break;
}

void spi_sendReceive(unsigned char command[])
{
    if(command[2] == 1)        //Check to see if this is the first of a series of SPI packets
    {
        spiBytesSent = 0;
        spiCSPin = command[3];
        spiWordSize = command[4];

        // Send First Packet's 8 Data Bytes
        for(int i=0; i<command[5]; i++)
        {
            // If this is the start of a new word toggle CS LOW
            if( (spiBytesSent == 0) || (spiBytesSent % spiWordSize == 0) )
            {
                digitalWrite(spiCSPin, LOW);
            }
            // Send SPI Byte
            Serial.print(SPI.transfer(command[i+6]));
            spiBytesSent++;

            // If word is complete set CS High
            if(spiBytesSent % spiWordSize == 0)
            {
                digitalWrite(spiCSPin, HIGH);
            }
        }
    }
    else
    {
        // SPI Data Packet - Send SPI Bytes
        for(int i=0; i<command[3]; i++)
        {
            // If this is the start of a new word toggle CS LOW
            if( (spiBytesSent == 0) || (spiBytesSent % spiWordSize == 0) )
            {
                digitalWrite(spiCSPin, LOW);
            }
            // Send SPI Byte
            Serial.print(SPI.transfer(command[i]));
            spiBytesSent++;

            // If word is complete set CS High
            if(spiBytesSent % spiWordSize == 0)
            {
                digitalWrite(spiCSPin, HIGH);
            }
        }
    }
}
digitalWrite(spiCSPin, LOW);
}
// Send SPI Byte
Serial.write(SPI.transfer(command[i+4]));
spiBytesSent++;

// If word is complete set CS High
if (spiBytesSent % spiWordSize == 0)
{
    digitalWrite(spiCSPin, HIGH);
}
}
}

// Synchronizes with LabVIEW and sends info about the board and firmware (Unimplemented)
void syncLV()
{
    Serial.begin(DEFAULTBAUDRATE);
i2cReadTimeouts = 0;
spiBytesSent = 0;
spiBytesToSend = 0;
Serial.flush();
}

// Compute Packet Checksum
unsigned char checksum_Compute(unsigned char command[])
{
    unsigned char checksum;
    for (int i = 0; i < (COMMANDLENGTH - 1); i++)
    {
        checksum += command[i];
    }
    return checksum;
}

// Compute Packet Checksum And Test Against Included Checksum
int checksum_Test(unsigned char command[])
{
    unsigned char checksum = checksum_Compute(command);
    if (checksum == command[COMMANDLENGTH - 1])
    {
        return 0;
    }
    else
    {
        return 1;
    }
}

// Stepper Functions, Modified for Sparkfun Quadstepper Motor Driver Board
#ifdef STEPPER_SUPPORT
void AccelStepper_Write(unsigned char command[])
{
    int steps = 0;
    int step_speed = 0;
    int acceleration = 0;
}
Number of steps & speed are a 16 bit values, split for data transfer. Reassemble 2 bytes to an int 16
steps = (int)(command[5] << 8) + command[6];
step_speed = (int)(command[2] << 8) + command[3];
acceleration = (int)(command[7] << 8) + command[8];

steppers[command[4]].setMaxSpeed(step_speed);
moveType[command[4]] = command[9]; // For continuous stepper movement.

if (acceleration == 0) {
    // Workaround AccelStepper bug that requires negative speed for negative step direction
    if (steps < 0) step_speed = -step_speed;
    steppers[command[4]].setSpeed(step_speed);
    steppers[command[4]].move(steps);
    steppers[command[4]].enableOutputs(); // For enable pin
} else {
    steppers[command[4]].setAcceleration(acceleration);
    steppers[command[4]].move(steps);
    steppers[command[4]].enableOutputs(); // For enable pin
}
#endif

void sampleContinuously()
{
    for(int i=0; i<iterations; i++)
    {
        retVal = analogRead(contAcqPin);
        if(contAcqSpeed>1000) // delay Microseconds is only accurate for values less that 16383
        {
            Serial.write( (retVal >> 2));
            delayMicroseconds(delayTime*1000000); // Delay for necessary amount of time to achieve desired sample rate
        }
        else
        {
            Serial.write( (retVal & 0xFF) );
            Serial.write( (retVal >> 8));
            delay(delayTime*1000);
        }
    }
}

void finiteAcquisition(int analogPin, float acquisitionSpeed, int numberOfSamples)
{
    // want to exit this loop every 8ms
    acquisitionPeriod=1/acquisitionSpeed;
    for(int i=0; i<numberOfSamples; i++)
    {
        retVal = analogRead(analogPin);
        if(acquisitionSpeed>1000)
        {
Serial.write( (retVal >> 2));
delayMicroseconds(acquisitionPeriod*1000000);
}
else
{
    Serial.write( (retVal & 0xFF) );
    Serial.write( (retVal >> 8));
    delay(acquisitionPeriod*1000);
}
}
}

void lcd_print(unsigned char command[])
{
    if(command[2] != 0)
    {
        // Base Specified By User
        int base = 0;
        switch(command[2])
        {
            case 0x01:  // BIN
                base = BIN;
                break;
            case 0x02:  // DEC
                base = DEC;
                break;
            case 0x03:  // OCT
                base = OCT;
                break;
            case 0x04:  // HEX
                base = HEX;
                break;
            default:
                break;
        }
        for(int i=0; i<command[3]; i++)
        {
            lcd.print(command[i+4], base);
        }
    }
    else
    {
        for(int i=0; i<command[3]; i++)
        {
            lcd.print((char)command[i+4]);
        }
    }
    Serial.write('0');
}
Appendix I

AFMotor Header Code

// Adafruit Motor shield library
// copyright Adafruit Industries LLC, 2009
// this code is public domain, enjoy!

#ifndef _AFMotor_h_
define _AFMotor_h_
#include <inttypes.h>
#include <avr/io.h>

#define MOTORDEBUG 1
#include <inttypes.h>
#include <avr/io.h>

//#define MOTORDEBUG 1

#define MICROSTEPS 16 //8 or 16

#define MOTOR12_64KHZ _BV(CS20) //no prescale
#define MOTOR12_8KHZ _BV(CS21) //divide by 8
#define MOTOR12_2KHZ _BV(CS21) | _BV(CS20) //divide by 32
#define MOTOR12_1KHZ _BV(CS22) //divide by 64

#define MOTOR34_64KHZ _BV(CS00) //no prescale
#define MOTOR34_8KHZ _BV(CS01) //divide by 8
#define MOTOR34_1KHZ _BV(CS01) | _BV(CS00) //divide by 64

#define MOTOR1_A 2
#define MOTOR1_B 3
#define MOTOR2_A 1
#define MOTOR2_B 4
#define MOTOR4_A 0
#define MOTOR4_B 6
#define MOTOR3_A 5
#define MOTOR3_B 7

#define FORWARD 1
#define BACKWARD 2
#define BRAKE 3
#define RELEASE 4

#define SINGLE 1
#define DOUBLE 2
#define INTERLEAVE 3
#define MICROSTEP 4

/*
#define LATCH 4
#define LATCH_DDR DDRB
#define LATCH_PORT PORTB
#define CLK_PORT PORTD
#define CLK_DDR DDRD
#define CLK 4
#define ENABLE_PORT PORTD
#define ENABLE_DDR DDRD
#define ENABLE 7
#define SER 0
#define SER_DDR DDRB
#define SER_PORT PORTB
*/

// Arduino pin names
#define MOTORLATCH 12
#define MOTORCLK 4
#define MOTORENABLE 7
#define MOTORDATA 8

class AFMotorController
{
    public:
        AFMotorController(void);
        void enable(void);
        friend class AF_DCMotor;
        void latch_tx(void);
};

class AF_DCMotor
{
    public:
        AF_DCMotor(uint8_t motornum, uint8_t freq = MOTOR34_8KHZ);
        void run(uint8_t_t);
        void setSpeed(uint8_t_t);

    private:
        uint8_t motornum,
        pwmfreq;
};

class AF_Stepper
{
    public:
        AF_Stepper(uint16_t, uint8_t);
        void step(uint16_t steps, uint8_t dir, uint8_t style = SINGLE);
        void setSpeed(uint16_t);
        uint8_t onestep(uint8_t dir, uint8_t style);
        void release(void);
        uint16_t revsteps; // # steps per revolution
        uint8_t steppernum;
};
uint32_t usperstep, steppingcounter;
private:
  uint8_t currentstep;
};

uint8_t getlatchstate(void);

#endif
Appendix J

AFMotor Source Code

// Adafruit Motor shield library
// copyright Adafruit Industries LLC, 2009
// this code is public domain, enjoy!

#include <avr/io.h>
#if defined(ARDUINO) && ARDUINO >= 100
#include "Arduino.h"
#else
#include "WProgram.h"
#endif
#include "AFMotor.h"

static uint8_t latch_state;
#if (MICROSTEPS == 8)
uint8_t microstepcurve[] = {0, 50, 98, 142, 180, 212, 236, 250, 255};
#elif (MICROSTEPS == 16)
uint8_t microstepcurve[] = {0, 25, 50, 74, 98, 120, 141, 162, 180, 197, 212, 225, 236, 244, 250, 253, 255};
#endif

AFMotorController::AFMotorController(void) {}

void AFMotorController::enable(void) {
    // setup the latch
    /*
     * LATCH_DDR |= _BV(LATCH);
     * ENABLE_DDR |= _BV(ENABLE);
     * CLK_DDR |= _BV(CLK);
     * SER_DDR |= _BV(SER);
     */
    pinMode(MOTORLATCH, OUTPUT);
    pinMode(MOTORENABLE, OUTPUT);
    pinMode(MOTORDATA, OUTPUT);
    pinMode(MOTORCLK, OUTPUT);

    latch_state = 0;
}
latch_tx(); // "reset"

//ENABLE_PORT &= ~_BV(ENABLE); // enable the chip outputs!
digitalWrite(MOTORENABLE, LOW);
}

void AFMotorController::latch_tx(void) {
  uint8_t i;

  //LAT
  LAT_PORT &= ~_BV(LATCH);
digitalWrite(MOTORLATCH, LOW);

  //SER_PORT &= ~_BV(SER);
digitalWrite(MOTORDATA, LOW);

  for (i=0; i<8; i++) {
    //CLK_PORT &= ~_BV(CLK);
digitalWrite(MOTORCLK, LOW);

    if (latch_state & _BV(7-i)) {
      //SER_PORT |= _BV(SER);
digitalWrite(MOTORDATA, HIGH);
    } else {
      //SER_PORT &= ~_BV(SER);
digitalWrite(MOTORDATA, LOW);
    }
    //CLK_PORT |= _BV(CLK);
digitalWrite(MOTORCLK, HIGH);
  }
  //LATCH_PORT |= _BV(LATCH);
digitalWrite(MOTORLATCH, HIGH);
}

static AFMotorController MC;

/*****************************/
MOTORS
/*****************************/
inline void initPWM1(uint8_t freq) {
  #if defined(__AVR_ATmega8__) ||
defined(__AVR_ATmega48__) ||
defined(__AVR_ATmega88__) ||
defined(__AVR_ATmega168__) ||
defined(__AVR_ATmega328P__)
// use PWM from timer2A on PB3 (Arduino pin #11)
  TCCR2A |= _BV(COM2A1) | _BV(WGM20) | _BV(WGM21); // fast PWM, turn on oc2a
  TCCR2B = freq & 0x7;
  OCR2A = 0;
  #elif defined(__AVR_ATmega1280__) || defined(__AVR_ATmega2560__)
// on arduino mega, pin 11 is now PB5 (OC1A)
  TCCR1A |= _BV(COM1A1) | _BV(WGM10); // fast PWM, turn on oc1a
  TCCR1B = (freq & 0x7) | _BV(WGM12);
  OCR1A = 0;
  #endif
}
#else
    #error "This chip is not supported!"
#endif

pinMode(11, OUTPUT);

} inline void setPWM1(uint8_t s) {
    #if defined(__AVR_ATmega8__) ||
        defined(__AVR_ATmega48__) ||
        defined(__AVR_ATmega88__) ||
        defined(__AVR_ATmega168__) ||
        defined(__AVR_ATmega328P__)
        // use PWM from timer2A on PB3 (Arduino pin #11)
    OCR2A = s;
    #elif defined(__AVR_ATmega1280__) || defined(__AVR_ATmega2560__) 
        // on arduino mega, pin 11 is now PB5 (OC1A)
    OCR1A = s;
    #else
        #error "This chip is not supported!"
    #endif

} inline void initPWM2(uint8_t freq) {
    #if defined(__AVR_ATmega8__) ||
        defined(__AVR_ATmega48__) ||
        defined(__AVR_ATmega88__) ||
        defined(__AVR_ATmega168__) ||
        defined(__AVR_ATmega328P__)
        // use PWM from timer2B (pin 3)
    TCCR2A |= _BV(COM2B1) | _BV(WGM20) | _BV(WGM21); // fast PWM, turn on oc2b
    TCCR2B = freq & 0x7;
    OCR2B = 0;
    #elif defined(__AVR_ATmega1280__) || defined(__AVR_ATmega2560__) 
        // on arduino mega, pin 3 is now PE5 (OC3C)
    TCCR3A |= _BV(COM1C1) | _BV(WGM10); // fast PWM, turn on oc3c
    TCCR3B = (freq & 0x7) | _BV(WGM12);
    OCR3C = 0;
    #else
        #error "This chip is not supported!"
    #endif

    pinMode(3, OUTPUT);

} inline void setPWM2(uint8_t s) {
    #if defined(__AVR_ATmega8__) ||
        defined(__AVR_ATmega48__) ||
        defined(__AVR_ATmega88__) ||
        defined(__AVR_ATmega168__) ||
        defined(__AVR_ATmega328P__)
        // use PWM from timer2A on PB3 (Arduino pin #11)
    OCR2B = s;
    #elif defined(__AVR_ATmega1280__) || defined(__AVR_ATmega2560__) 
        // on arduino mega, pin 11 is now PB5 (OC1A)
    OCR3C = s;
    #else

inline void initPWM3(uint8_t freq) {
#if defined(__AVR_ATmega8__) ||
defined(__AVR_ATmega48__) ||
defined(__AVR_ATmega88__) ||
defined(__AVR_ATmega168__) ||
defined(__AVR_ATmega328P__)
  // use PWM from timer0A / PD6 (pin 6)
  TCCR0A |= _BV(COM0A1) | _BV(WGM00) | _BV(WGM01); // fast PWM, turn on OC0A
  // TCCR0B = freq & 0x7;
  OCR0A = 0;
#else
#error "This chip is not supported!"
#endif
  pinMode(6, OUTPUT);
}

inline void setPWM3(uint8_t s) {
#if defined(__AVR_ATmega8__) ||
defined(__AVR_ATmega48__) ||
defined(__AVR_ATmega88__) ||
defined(__AVR_ATmega168__) ||
defined(__AVR_ATmega328P__)
  // use PWM from timer0A on PB3 (Arduino pin #6)
  OCR0A = s;
#elif defined(__AVR_ATmega1280__) || defined(__AVR_ATmega2560__) // on arduino mega, pin 6 is now PH3 (OC4A)
  OCR4A = s;
#else
#error "This chip is not supported!"
#endif
}

inline void initPWM4(uint8_t freq) {
#if defined(__AVR_ATmega8__) ||
defined(__AVR_ATmega48__) ||
defined(__AVR_ATmega88__) ||
defined(__AVR_ATmega168__) ||
defined(__AVR_ATmega328P__)
  // use PWM from timer0B / PD5 (pin 5)
  TCCR0A |= _BV(COM0B1) | _BV(WGM00) | _BV(WGM01); // fast PWM, turn on OC0A
  // TCCR0B = freq & 0x7;
  OCR0B = 0;
#else
#error "This chip is not supported!"
#endif
  pinMode(6, OUTPUT);
}
TCCR3A |= _BV(COM1A1) | _BV(WGM10); // fast PWM, turn on oc3a
TCCR3B = (freq & 0x7) | _BV(WGM12);
//TCCR4B = 1 | _BV(WGM12);
OCR3A = 0;
#else
#error "This chip is not supported!"
#endif
pinMode(5, OUTPUT);
}

inline void setPWM4(uint8_t s) {
#if defined(__AVR_ATmega8__) ||
  defined(__AVR_ATmega48__) ||
  defined(__AVR_ATmega88__) ||
  defined(__AVR_ATmega168__) ||
  defined(__AVR_ATmega328P__)
  // use PWM from timer0A on PB3 (Arduino pin #6)
  OCR0B = s;
#elif defined(__AVR_ATmega1280__) || defined(__AVR_ATmega2560__) || defined(__AVR_ATmega2560__)
  // on arduino mega, pin 6 is now PH3 (OC4A)
  OCR3A = s;
#else
#error "This chip is not supported!"
#endif
}

AF_DCMotor::AF_DCMotor(uint8_t num, uint8_t freq) {
  motornum = num;
pwmfreq = freq;
  MC.enable();

  switch (num) {
  case 1:
    latch_state &= ~_BV(MOTOR1_A) & ~_BV(MOTOR1_B); // set both motor pins to 0
    MC.latch_tx();
    initPWM1(freq);
    break;
  case 2:
    latch_state &= ~_BV(MOTOR2_A) & ~_BV(MOTOR2_B); // set both motor pins to 0
    MC.latch_tx();
    initPWM2(freq);
    break;
  case 3:
    latch_state &= ~_BV(MOTOR3_A) & ~_BV(MOTOR3_B); // set both motor pins to 0
    MC.latch_tx();
    initPWM3(freq);
    break;
  case 4:
    latch_state &= ~_BV(MOTOR4_A) & ~_BV(MOTOR4_B); // set both motor pins to 0
    MC.latch_tx();
    initPWM4(freq);
    break;
  }
}
void AF_DCMotor::run(uint8_t cmd) {
    uint8_t a, b;
    switch (motornum) {
    case 1:
        a = MOTOR1_A; b = MOTOR1_B; break;
    case 2:
        a = MOTOR2_A; b = MOTOR2_B; break;
    case 3:
        a = MOTOR3_A; b = MOTOR3_B; break;
    case 4:
        a = MOTOR4_A; b = MOTOR4_B; break;
    default:
        return;
    }
    switch (cmd) {
    case FORWARD:
        latch_state |= _BV(a);
        latch_state &= ~_BV(b);
        MC.latch_tx();
        break;
    case BACKWARD:
        latch_state &= ~_BV(a);
        latch_state |= _BV(b);
        MC.latch_tx();
        break;
    case RELEASE:
        latch_state &= _BV(a);
        latch_state &= _BV(b);
        MC.latch_tx();
        break;
    }
}

void AF_DCMotor::setSpeed(uint8_t speed) {
    switch (motornum) {
    case 1:
        setPWM1(speed); break;
    case 2:
        setPWM2(speed); break;
    case 3:
        setPWM3(speed); break;
    case 4:
        setPWM4(speed); break;
    }
}

/******************************************
 * STEPPE RS
 ******************************************/

AF_Stepper::AF_Stepper(uint16_t steps, uint8_t num) {
    MC.enable();

    revsteps = steps;
    stepperNum = num;
}
currentstep = 0;

if (steppernum == 1) {
latch_state &= ~_BV(MOTOR1_A) & ~_BV(MOTOR1_B) &
~_BV(MOTOR2_A) & ~_BV(MOTOR2_B); // all motor pins to 0
MC.latch_tx();

// enable both H bridges
pinMode(11, OUTPUT);
pinMode(3, OUTPUT);
digitalWrite(11, HIGH);
digitalWrite(3, HIGH);

// use PWM for microstepping support
initPWM1(MOTOR12_64KHZ);
initPWM2(MOTOR12_64KHZ);
setPWM1(255);
setPWM2(255);
}
else if (steppernum == 2) {
latch_state &= ~_BV(MOTOR3_A) & ~_BV(MOTOR3_B) &
~_BV(MOTOR4_A) & ~_BV(MOTOR4_B); // all motor pins to 0
MC.latch_tx();

// enable both H bridges
pinMode(5, OUTPUT);
pinMode(6, OUTPUT);
digitalWrite(5, HIGH);
digitalWrite(6, HIGH);

// use PWM for microstepping support
// use PWM for microstepping support
initPWM3(1);
initPWM4(1);
setPWM3(255);
setPWM4(255);
}

void AF_Stepper::setSpeed(uint16_t rpm) {
usperstep = 60000000 / ((uint32_t)revsteps * (uint32_t)rpm);
steppingcounter = 0;
}

void AF_Stepper::release(void) {
if (steppernum == 1) {
latch_state &= ~_BV(MOTOR1_A) & ~_BV(MOTOR1_B) &
~_BV(MOTOR2_A) & ~_BV(MOTOR2_B); // all motor pins to 0
MC.latch_tx();
} else if (steppernum == 2) {
latch_state &= ~_BV(MOTOR3_A) & ~_BV(MOTOR3_B) &
~_BV(MOTOR4_A) & ~_BV(MOTOR4_B); // all motor pins to 0
MC.latch_tx();
}
```c
void AF_Stepper::step(uint16_t steps, uint8_t dir, uint8_t style) {
  uint32_t uspers = usperstep;
  uint8_t ret = 0;

  if (style == INTERLEAVE) {
    uspers /= 2;
  } else if (style == MICROSTEP) {
    uspers /= MICROSTEPS;
    steps *= MICROSTEPS;
    #ifdef MOTORDEBUG
    Serial.print("steps = "); Serial.println(steps, DEC);
    #endif
  }

  while (steps--) {
    ret = onestep(dir, style);
    delay(uspers/1000); // in ms
    steppingcounter += (uspers % 1000);
    if (steppingcounter >= 1000) {
      delay(1);
      steppingcounter -= 1000;
    }
  }

  if (style == MICROSTEP) {
    while ((ret != 0) && (ret != MICROSTEPS)) {
      ret = onestep(dir, style);
      delay(uspers/1000); // in ms
      steppingcounter += (uspers % 1000);
      if (steppingcounter >= 1000) {
        delay(1);
        steppingcounter -= 1000;
      }
    }
  }

  uint8_t AF_Stepper::onestep(uint8_t dir, uint8_t style) {
    uint8_t a, b, c, d;
    uint8_t ocrb, ocra;
    ocra = ocrb = 255;

    if (steppernum == 1) {
      a = _BV(MOTOR1_A);
      b = _BV(MOTOR2_A);
      c = _BV(MOTOR1_B);
      d = _BV(MOTOR2_B);
    } else if (steppernum == 2) {
      a = _BV(MOTOR3_A);
      b = _BV(MOTOR4_A);
      c = _BV(MOTOR3_B);
      d = _BV(MOTOR4_B);
    } else {
      return 0;
    }
  }
```

// next determine what sort of stepping procedure we're up to
if (style == SINGLE) {
    if ((currentstep/(MICROSTEPS/2)) % 2) { // we're at an odd step, weird
        if (dir == FORWARD) {
            currentstep += MICROSTEPS/2;
        } else {
            currentstep -= MICROSTEPS/2;
        }
    } else { // go to the next even step
        if (dir == FORWARD) {
            currentstep += MICROSTEPS;
        } else {
            currentstep -= MICROSTEPS;
        }
    }
} else if (style == DOUBLE) {
    if (! (currentstep/(MICROSTEPS/2) % 2)) { // we're at an even step, weird
        if (dir == FORWARD) {
            currentstep += MICROSTEPS/2;
        } else {
            currentstep -= MICROSTEPS/2;
        }
    } else { // go to the next odd step
        if (dir == FORWARD) {
            currentstep += MICROSTEPS;
        } else {
            currentstep -= MICROSTEPS;
        }
    }
} else if (style == INTERLEAVE) {
    if (dir == FORWARD) {
        currentstep += MICROSTEPS/2;
    } else {
        currentstep -= MICROSTEPS/2;
    }
}

if (style == MICROSTEP) {
    if (dir == FORWARD) {
        currentstep++;
    } else {
        // BACKWARDS
        currentstep--;
    }
}

currentstep += MICROSTEPS*4;
currentstep %= MICROSTEPS*4;

ocra = ocrb = 0;
if ( (currentstep >= 0) && (currentstep < MICROSTEPS)) {
    ocra = microstepcurve[MICROSTEPS - currentstep];
    ocrb = microstepcurve[currentstep];
} else if ( (currentstep >= MICROSTEPS) && (currentstep < MICROSTEPS*2)) {
ocra = microstepcurve[currentstep - MICROSTEPS];
ocrb = microstepcurve[MICROSTEPS*2 - currentstep];
} else if ( (currentstep >= MICROSTEPS*2) && (currentstep < MICROSTEPS*3)) {
ocra = microstepcurve[MICROSTEPS*3 - currentstep];
ocrb = microstepcurve[currentstep - MICROSTEPS*2];
} else if ( (currentstep >= MICROSTEPS*3) && (currentstep < MICROSTEPS*4)) {
ocra = microstepcurve[currentstep - MICROSTEPS*3];
ocrb = microstepcurve[MICROSTEPS*4 - currentstep];
}
}
currentstep += MICROSTEPS*4;
currentstep %= MICROSTEPS*4;

#ifdef MOTORDEBUG
Serial.print("current step: "); Serial.println(currentstep, DEC);
Serial.print(" pwmA = "); Serial.print(ocra, DEC);
Serial.print(" pwmB = "); Serial.println(ocrb, DEC);
#endif

if (steppernum == 1) {
setPWM1(ocra);
setPWM2(ocrb);
} else if (steppernum == 2) {
setPWM3(ocra);
setPWM4(ocrb);
}

// release all
latch_state &= ~a & ~b & ~c & ~d; // all motor pins to 0

//Serial.println(step, DEC);
if (style == MICROSTEP) {
if ((currentstep >= 0) && (currentstep < MICROSTEPS))
latch_state |= a | b;
if ( (currentstep >= MICROSTEPS) && (currentstep < MICROSTEPS*2))
latch_state |= b | c;
if ((currentstep >= MICROSTEPS*2) && (currentstep < MICROSTEPS*3))
latch_state |= c | d;
if ((currentstep >= MICROSTEPS*3) && (currentstep < MICROSTEPS*4))
latch_state |= d | a;
} else {
switch (currentstep/(MICROSTEPS/2)) {
case 0:
latch_state |= a; // energize coil 1 only
break;
case 1:
latch_state |= a | b; // energize coil 1+2
break;
case 2:
latch_state |= b; // energize coil 2 only
break;
case 3:
latch_state |= b | c; // energize coil 2+3
break;
}
case 4:
    latch_state |= c; // energize coil 3 only
    break;

case 5:
    latch_state |= c | d; // energize coil 3+4
    break;

case 6:
    latch_state |= d; // energize coil 4 only
    break;

case 7:
    latch_state |= d | a; // energize coil 1+4
    break;

 MC.latch_tx();
 return currentstep;
 }
Appendix K

AccelStepper Header Code

// AccelStepper.h
//
//--\mainpage AccelStepper library for Arduino
//--\ This is the Arduino AccelStepper library.
//--\ It provides an object-oriented interface for 2 or 4 pin stepper motors.
//--\ The standard Arduino IDE includes the Stepper library
//--\ (http://arduino.cc/en/Reference/Stepper) for stepper motors. It is
//--\ perfectly adequate for simple, single motor applications.
//--\ AccelStepper significantly improves on the standard Arduino Stepper library in several ways:
//--\ \li Supports acceleration and deceleration
//--\ \li Supports multiple simultaneous steppers, with independent concurrent stepping on each stepper
//--\ \li API functions never delay() or block
//--\ \li Supports 2 and 4 wire steppers, plus 4 wire half steppers.
//--\ \li Supports alternate stepping functions to enable support of AFMotor
//--\ \li Supports stepper drivers such as the Sparkfun EasyDriver (based on 3967 driver chip)
//--\ \li Very slow speeds are supported
//--\ \li Extensive API
//--\ \li Subclass support
//--\ The latest version of this documentation can be downloaded from
//--\ Example Arduino programs are included to show the main modes of use.
//--\ The version of the package that this documentation refers to can be downloaded
//--\ You can find the latest version at http://www.open.com.au/mikem/arduino/AccelStepper
//--\ You can also find online help and discussion at http://groups.google.com/group/accelstepper
//--\ Please use that group for all questions and discussions on this topic.
//--\ Do not contact the author directly, unless it is to discuss commercial licensing.
//--\ Tested on Arduino Diecimila and Mega with arduino-0018 & arduino-0021
//--\ on OpenSuSE 11.1 and avr-libc-1.6.1-1.15,
//--\ cross-avr-binutils-2.19-9.1, cross-avr-gcc-4.1.3_20080612-26.5.
///
/// \par Installation
/// Install in the usual way: unzip the distribution zip file to the libraries
/// sub-folder of your sketchbook.
///
/// This software is Copyright (C) 2010 Mike McCauley. Use is subject to license
/// conditions. The main licensing options available are GPL V2 or Commercial:
///
/// \par Open Source Licensing GPL V2
/// This is the appropriate option if you want to share the source code of your
/// application with everyone you distribute it to, and you also want to give them
/// the right to share who uses it. If you wish to use this software under Open
/// Source Licensing, you must contribute all your source code to the open source
/// community in accordance with the GPL Version 2 when your application is
/// distributed. See http://www.gnu.org/copyleft/gpl.html
///
/// \par Commercial Licensing
/// This is the appropriate option if you are creating proprietary applications
/// and you are not prepared to distribute and share the source code of your
/// application. Contact info@open.com.au for details.
///
/// \par Revision History
/// version 1.0 Initial release
///
/// version 1.1 Added speed() function to get the current speed.
/// version 1.2 Added runSpeedToPosition() submitted by Gunnar Arndt.
/// version 1.3 Added support for stepper drivers (ie with Step and Direction inputs) with _pins == 1
/// version 1.4 Added functional constructor to support AFMotor, contributed by Limor, with example
/// sketches.
/// version 1.5 Improvements contributed by Peter Mousley: Use of microsecond steps and other speed
/// improvements
/// to increase max stepping speed to about 4kHz. New option for user to set the min allowed pulse
/// width.
/// version 1.6 Fixed a problem with wrapping of microsecond stepping that could cause stepping to hang.
/// Reporte by Sandy Noble.
/// Removed redundant _lastRunTime member.
/// version 1.7 Fixed a bug where setCurrentPosition() did always work as expected. Reported by Peter
/// Linhart.
/// Reporte by Sandy Noble.
/// Removed redundant _lastRunTime member.
/// version 1.8 Added support for 4 pin half-steppers, requested by Harvey Moon
/// version 1.9 setPosition() now also sets motor speed to 0.
/// version 1.10 Builds on Arduino 1.0
/// version 1.11 Improvements from Michael Ellison:
/// Added optional enable line support for stepper drivers
/// Added inversion for step/direction/enable lines for stepper drivers
/// version 1.12 Announce Google Group
/// version 1.13 Improvements to speed calculation. Cost of calculation is now less in the worst case,
/// and more or less constant in all cases. This should result in slightly better high speed performance, and
/// reduce anomalous speed glitches when other steppers are accelerating.
/// However, its hard to see how to replace the sqrt() required at the very first step from 0 speed.
/// version 1.14 Fixed a problem with compiling under arduino 0021 reported by EmbeddedMan
/// version 1.15 Fixed a problem with runSpeedToPosition which did not correctly handle
/// running backwards to a smaller target position. Added examples
/// version 1.16 Fixed some cases in the code where abs() was used instead of fabs().
#ifndef AccelStepper_h
#define AccelStepper_h

#include <stdlib.h>

#if ARDUINO >= 100
#include <Arduino.h>
#else
#include <Wire.h>
#endif

// These defs cause trouble on some versions of Arduino
#undef round

class AccelStepper

// brief Support for stepper motors with acceleration etc.
// This defines a single 2 or 4 pin stepper motor, or stepper motor with fdriver chip, with optional
// acceleration, deceleration, absolute positioning commands etc. Multiple
// simultaneous steppers are supported, all moving
// at different speeds and accelerations.

// This module operates by computing a step time in microseconds. The step
// time is recomputed after each step and after speed and acceleration
// parameters are changed by the caller. The time of each step is recorded in
// microseconds. The run() function steps the motor if a new step is due.
// The run() function must be called frequently until the motor is in the
// desired position, after which time run() will do nothing.

// Positions are specified by a signed long integer. At
// construction time, the current position of the motor is consider to be 0. Positive
// positions are clockwise from the initial position; negative positions are
// anticlockwise. The curent position can be altered for instance after
// initialization positioning.

// This is an open loop controller: If the motor stalls or is oversped,
// AccelStepper will not have a correct
// idea of where the motor really is (since there is no feedback of the motor's
// real position. We only know where we _think_ it is, relative to the
// initial starting point).

// The fastest motor speed that can be reliably supported is 4000 steps per
// second (4 kHz) at a clock frequency of 16 MHz. However, any speed less than that
// down to very slow speeds (much less than one per second) are also supported.
/// provided the run() function is called frequently enough to step the motor
class AccelStepper
{
public:
  // Constructor. You can have multiple simultaneous steppers, all moving
  // at different speeds and accelerations, provided you call their run()
  // functions at frequent enough intervals. Current Position is set to 0, target
  // position is set to 0. MaxSpeed and Acceleration default to 1.0.
  // The motor pins will be initialised to OUTPUT mode during the
  // constructor by a call to enableOutputs().
  // param[in] pins Number of pins to interface to. 1, 2 or 4 are
  // supported. 1 means a stepper driver (with Step and Direction pins).
  // If an enable line is also needed, call setEnablePin() after construction.
  // You may also invert the pins using setPinsInverted().
  // 2 means a 2 wire stepper. 4 means a 4 wire stepper. 8 means a 4 wire half stepper
  // Defaults to 4 pins.
  // param[in] pin1 Arduino digital pin number for motor pin 1. Defaults
  // to pin 2. For a driver (pins==1), this is the Step input to the driver. Low to high transition means to step)
  // to pin 3. For a driver (pins==1), this is the Direction input the driver. High means forward.
  // to pin 4.
  // to pin 5.

  AccelStepper(uint8_t pins = 4, uint8_t pin1 = 2, uint8_t pin2 = 3, uint8_t pin3 = 4, uint8_t pin4 = 5);
  // Alternate Constructor which will call your own functions for forward and backward steps.
  // You can have multiple simultaneous steppers, all moving
  // at different speeds and accelerations, provided you call their run()
  // functions at frequent enough intervals. Current Position is set to 0, target
  // position is set to 0. MaxSpeed and Acceleration default to 1.0.
  // Any motor initialization should happen before hand, no pins are used or initialized.
  // param[in] forward void-returning procedure that will make a forward step
  // param[in] backward void-returning procedure that will make a backward step
  AccelStepper(void (*forward)(), void (*backward)());

  // Set the target position. The run() function will try to move the motor
  // from the current position to the target position set by the most
  // recent call to this function. Caution: moveTo() also recalculates the speed for the next step.
  // If you are trying to use constant speed movements, you should call setSpeed() after calling moveTo().
  // param[in] absolute The desired absolute position. Negative is
  // anticlockwise from the 0 position.
  void moveTo(long absolute);

  // Set the target position relative to the current position
  // param[in] relative The desired position relative to the current position. Negative is
  // anticlockwise from the current position.
  void move(long relative);

  // Poll the motor and step it if a step is due, implementing
  // accelerations and decelerations to achieve the target position. You must call this as
  // frequently as possible, but at least once per minimum step interval,
  // preferably in your main loop.
  // 
  // return true if the motor is at the target position.
}
boolean run();

/// Poll the motor and step it if a step is due, implementing a constant
/// speed as set by the most recent call to setSpeed(). You must call this as
/// frequently as possible, but at least once per step interval,
/// return true if the motor was stepped.
boolean runSpeed();

/// Sets the maximum permitted speed. The run() function will accelerate
/// up to the speed set by this function.
/// param[in] speed The desired maximum speed in steps per second. Must
/// be > 0. Speeds of more than 1000 steps per second are unreliable.
void setMaxSpeed(float speed);

/// Sets the acceleration and deceleration parameter.
/// param[in] acceleration The desired acceleration in steps per second
/// per second. Must be > 0.
void setAcceleration(float acceleration);

/// Sets the desired constant speed for use with runSpeed().
/// param[in] speed The desired constant speed in steps per
/// second. Positive is clockwise. Speeds of more than 1000 steps per
/// second are unreliable. Very slow speeds may be set (eg 0.00027777 for
/// once per hour, approximately. Speed accuracy depends on the Arduino
/// crystal. Jitter depends on how frequently you call the runSpeed() function.
void setSpeed(float speed);

/// The most recently set speed
/// return the most recent speed in steps per second
float speed();

/// The distance from the current position to the target position.
/// return the distance from the current position to the target position
/// in steps. Positive is clockwise from the current position.
long distanceToGo();

/// The most recently set target position.
/// return the target position
/// in steps. Positive is clockwise from the 0 position.
long targetPosition();

/// The currently motor position.
/// return the current motor position
/// in steps. Positive is clockwise from the 0 position.
long currentPosition();

/// Resets the current position of the motor, so that wherever the motor
/// happens to be right now is considered to be the new 0 position. Useful
/// for setting a zero position on a stepper after an initial hardware
/// positioning move.
/// Has the side effect of setting the current motor speed to 0.
/// param[in] position The position in steps of wherever the motor
/// happens to be right now.
void setCurrentPosition(long position);
/// Moves the motor at the currently selected constant speed (forward or reverse)
/// to the target position and blocks until it is at
/// position. Dont use this in event loops, since it blocks.
void runToPosition();

/// Runs at the currently selected speed until the target position is reached
/// Does not implement accelerations.
/// return true if it stepped
boolean runSpeedToPosition();

/// Moves the motor to the new target position and blocks until it is at
/// position. Dont use this in event loops, since it blocks.
/// \param[in] position The new target position.
void runToNewPosition(long position);

/// Disable motor pin outputs by setting them all LOW
/// Depending on the design of your electronics this may turn off
/// the power to the motor coils, saving power.
/// This is useful to support Arduino low power modes: disable the outputs
/// during sleep and then reenable with enableOutputs() before stepping
/// again.
void disableOutputs();

/// Enable motor pin outputs by setting the motor pins to OUTPUT
/// mode. Called automatically by the constructor.
void enableOutputs();

/// Sets the minimum pulse width allowed by the stepper driver.
/// \param[in] minWidth The minimum pulse width in microseconds.
void setMinPulseWidth(unsigned int minWidth);

/// Sets the enable pin number for stepper drivers.
/// 0xFF indicates unused (default).
/// Otherwise, if a pin is set, the pin will be turned on when
/// enableOutputs() is called and switched off when disableOutputs() is called.
/// \param[in] enablePin Arduino digital pin number for motor enable
/// \sa setPinsInverted
void setEnablePin(uint8_t enablePin = 0xff);

/// Sets the inversion for stepper driver pins
/// \param[in] direction True for inverted direction pin, false for non-inverted
/// \param[in] step True for inverted step pin, false for non-inverted
/// \param[in] enable True for inverted enable pin, false (default) for non-inverted
void setPinsInverted(bool direction, bool step, bool enable = false);

protected:

/// Forces the library to compute a new instantaneous speed and set that as
/// the current speed. Calls
/// desiredSpeed(), which can be overridden by subclasses. It is called by
/// the library:
/// after each step
/// after change to maxSpeed through setMaxSpeed()
/// after change to acceleration through setAcceleration()
/// after change to target position (relative or absolute) through
void computeNewSpeed();

virtual void step(uint8_t step);

virtual void step0(void);

virtual void step1(uint8_t step);

virtual void step2(uint8_t step);

virtual void step4(uint8_t step);

virtual void step8(uint8_t step);

virtual float desiredSpeed();

private:
  uint8_t _pins; // 2 or 4
  // Arduino pin number for the 2 or 4 pins required to interface to the
/// Stepper motor.
uint8_t _pin1, _pin2, _pin3, _pin4;

/// The current absolute position in steps.
long _currentPos; // Steps

/// The target position in steps. The AccelStepper library will move the
/// motor from the _currentPos to the _targetPos, taking into account the
/// max speed, acceleration and deceleration
long _targetPos; // Steps

/// The current motor's speed in steps per second
/// Positive is clockwise
float _speed; // Steps per second

/// The maximum permitted speed in steps per second. Must be > 0.
float _maxSpeed;

/// The acceleration to use to accelerate or decelerate the motor in steps
/// per second per second. Must be > 0
float _acceleration;

/// The current interval between steps in microseconds
unsigned long _stepInterval;

/// The last step time in microseconds
unsigned long _lastStepTime;

/// The minimum allowed pulse width in microseconds
unsigned int _minPulseWidth;

/// Is the direction pin inverted?
bool _dirInverted;

/// Is the step pin inverted?
bool _stepInverted;

/// Is the enable pin inverted?
bool _enableInverted;

/// Enable pin for stepper driver, or 0xFF if unused.
uint8_t _enablePin;

/// The pointer to a forward-step procedure
void (*_forward)();

/// The pointer to a backward-step procedure
void (*_backward)();
};

/// @example Random.pde
/// Make a single stepper perform random changes in speed, position and acceleration

/// @example Overshoot.pde
/// Check overshoot.pde
/// which sets a new target position and then waits until the stepper has
// achieved it. This is used for testing the handling of overshoots

// @example MultiStepper.pde
// Shows how to multiple simultaneous steppers
// Runs one stepper forwards and backwards, accelerating and decelerating
// at the limits. Runs other steppers at the same time

// @example ConstantSpeed.pde
// Shows how to run AccelStepper in the simplest,
// fixed speed mode with no accelerations

// @example Blocking.pde
// Shows how to use the blocking call runToNewPosition
// Which sets a new target position and then waits until the stepper has
// achieved it.

// @example AFMotor_MultiStepper.pde
// Control both Stepper motors at the same time with different speeds
// and accelerations.

// @example AFMotor_ConstantSpeed.pde
// Shows how to run AccelStepper in the simplest,
// fixed speed mode with no accelerations

// @example ProportionalControl.pde
// Make a single stepper follow the analog value read from a pot or whatever
// The stepper will move at a constant speed to each newly set position,
// depending on the value of the pot.

#endif
Appendix L

AccelStepper Source Code

// AccelStepper.cpp
//
// Copyright (C) 2009 Mike McCauley
// $Id: AccelStepper.cpp,v 1.5 2012/01/28 22:45:25 mikem Exp mikem$

#include "AccelStepper.h"

void AccelStepper::moveTo(long absolute)
{
    _targetPos = absolute;
    computeNewSpeed();
}

void AccelStepper::move(long relative)
{
    moveTo(_currentPos + relative);
}

// Implements steps according to the current speed
// You must call this at least once per step
// returns true if a step occurred
boolean AccelStepper::runSpeed()
{
    // Don't do anything unless we actually have a speed
    if (_speed == 0.0f)
        return false;

    unsigned long time = micros();
    // Gymnastics to detect wrapping of either the nextStepTime and/or the current time
    unsigned long nextStepTime = _lastStepTime + _stepInterval;
    if ( ((nextStepTime >= _lastStepTime) && ((time >= nextStepTime) || (time < _lastStepTime)))
        || ((nextStepTime < _lastStepTime) && ((time >= nextStepTime) && (time < _lastStepTime))))
    {
        if (_speed > 0.0f)
        {
            // Clockwise
            _currentPos += 1;
        }
    }
else if (_speed < 0.0f)
{
    // Anticlockwise
    _currentPos = 1;
}
step(_currentPos & 0x7); // Bottom 3 bits (same as mod 8, but works with + and - numbers)

_lastStepTime = time;
return true;
} else
{
    return false;
}
}

long AccelStepper::distanceToGo()
{
    return abs(_targetPos - _currentPos);
}

long AccelStepper::targetPosition()
{
    return _targetPos;
}

long AccelStepper::currentPosition()
{
    return _currentPos;
}

// Useful during initialisations or after initial positioning
void AccelStepper::setCurrentPosition(long position)
{
    _targetPos = _currentPos = position;
    computeNewSpeed(); // Expect speed of 0
}

void AccelStepper::computeNewSpeed()
{
    setSpeed(desiredSpeed());
}

// Work out and return a new speed.
// Subclasses can override if they want
// Implement acceleration, deceleration and max speed
// Negative speed is anticlockwise
// This is called:
// after each step
// after user changes:
// maxSpeed
// acceleration
// target position (relative or absolute)
float AccelStepper::desiredSpeed()
{
    float requiredSpeed;
long distanceTo = distanceToGo(); // +ve is clockwise from current location

if (distanceTo == 0)
    return 0.0f; // We're there

// sqrSpeed is the signed square of _speed.
float sqrSpeed = sq(_speed);
if (_speed < 0.0)
    sqrSpeed = -sqrSpeed;
float twoa = 2.0f * _acceleration; // 2ag
// if v^2/2as is the left of target, we will arrive at 0 speed too far -ve, need to accelerate clockwise
if ((sqrSpeed / twoa) < distanceTo)
{
    // Accelerate clockwise
    // Need to accelerate in clockwise direction
    if (_speed == 0.0f)
        requiredSpeed = sqrt(twoa);
    else
        requiredSpeed = _speed + fabs(_acceleration / _speed);
    if (requiredSpeed > _maxSpeed)
        requiredSpeed = _maxSpeed;
}
else
{
    // Decelerate clockwise, accelerate anticlockwise
    // Need to accelerate in clockwise direction
    if (_speed == 0.0f)
        requiredSpeed = -sqrt(twoa);
    else
        requiredSpeed = _speed - fabs(_acceleration / _speed);
    if (requiredSpeed < -_maxSpeed)
        requiredSpeed = -_maxSpeed;
}

// Serial.println(requiredSpeed);
return requiredSpeed;

// Run the motor to implement speed and acceleration in order to proceed to the target position
// You must call this at least once per step, preferably in your main loop
// If the motor is in the desired position, the cost is very small
// returns true if we are still running to position
boolean AccelStepper::run()
{
    if (_targetPos == _currentPos)
        return false;

    if (runSpeed())
        computeNewSpeed();
    return true;
}

AccelStepper::AccelStepper(uint8_t pins, uint8_t pin1, uint8_t pin2, uint8_t pin3, uint8_t pin4)
{
    _pins = pins;
    _currentPos = 0;
_targetPos = 0;
_speed = 0.0;
_maxSpeed = 1.0;
_acceleration = 1.0;
_stepInterval = 0;
_minPulseWidth = 1;
_dirInverted = false;
_stepInverted = false;
_enablePin = 0xff;
_lastStepTime = 0;
_pin1 = pin1;
_pin2 = pin2;
_pin3 = pin3;
_pin4 = pin4;
//_stepInterval = 20000;
//_speed = 50.0;
//_lastRunTime = 0xffffffff - 20000;
//_lastStepTime = 0xffffffff - 20000 - 10000;
enableOutputs();
}

AccelStepper::AccelStepper(void (*forward)(), void (*backward)())
{
  _pins = 0;
  _currentPos = 0;
  _targetPos = 0;
  _speed = 0.0;
  _maxSpeed = 1.0;
  _acceleration = 1.0;
  _stepInterval = 0;
  _minPulseWidth = 1;
  _dirInverted = false;
  _stepInverted = false;
  _enablePin = 0xff;
  _lastStepTime = 0;
  _pin1 = 0;
  _pin2 = 0;
  _pin3 = 0;
  _pin4 = 0;
  _forward = forward;
  _backward = backward;
}

void AccelStepper::setMaxSpeed(float speed)
{
  _maxSpeed = speed;
  computeNewSpeed();
}

void AccelStepper::setAcceleration(float acceleration)
{
  _acceleration = acceleration;
  computeNewSpeed();
}

void AccelStepper::setSpeed(float speed)
{  
  if (speed == _speed)  
    return;

  if ((speed > 0.0f) && (speed > _maxSpeed))  
    _speed = _maxSpeed;
  else if ((speed < 0.0f) && (speed < -_maxSpeed))  
    _speed = -_maxSpeed;
  else  
    _speed = speed;

  _stepInterval = fabs(1000000.0 / _speed);
}

float AccelStepper::speed()  
{
  return _speed;
}

// Subclasses can override
void AccelStepper::step(uint8_t step)
{
  switch (_pins)
  {
  case 0:
    step0();
    break;
  case 1:
    step1(step);
    break;

  case 2:
    step2(step);
    break;

  case 4:
    step4(step);
    break;

  case 8:
    step8(step);
    break;
  }
}

// 0 pin step function (ie for functional usage)
void AccelStepper::step0()
{
  if (_speed > 0)
    _forward();
  else
    _backward();
}

// 1 pin step function (ie for stepper drivers)
// This is passed the current step number (0 to 7)
// Subclasses can override
void AccelStepper::step1(uint8_t step)
{
    digitalWrite(_pin2, (_speed > 0) ^ _dirInverted); // Direction
    // Caution 200ns setup time
    digitalWrite(_pin1, HIGH ^ _stepInverted);
    // Delay the minimum allowed pulse width
    delayMicroseconds(_minPulseWidth);
    digitalWrite(_pin1, LOW ^ _stepInverted);
}

// 2 pin step function
// This is passed the current step number (0 to 7)
// Subclasses can override
void AccelStepper::step2(uint8_t step)
{
    switch (step & 0x3)
    {
    case 0: /* 01 */
        digitalWrite(_pin1, LOW);
        digitalWrite(_pin2, HIGH);
        break;

    case 1: /* 11 */
        digitalWrite(_pin1, HIGH);
        digitalWrite(_pin2, HIGH);
        break;

    case 2: /* 10 */
        digitalWrite(_pin1, HIGH);
        digitalWrite(_pin2, LOW);
        break;

    case 3: /* 00 */
        digitalWrite(_pin1, LOW);
        digitalWrite(_pin2, LOW);
        break;
    }
}

// 4 pin step function for half stepper
// This is passed the current step number (0 to 7)
// Subclasses can override
void AccelStepper::step4(uint8_t step)
{
    switch (step & 0x3)
    {
    case 0: // 1010
        digitalWrite(_pin1, HIGH);
        digitalWrite(_pin2, LOW);
        digitalWrite(_pin3, HIGH);
        digitalWrite(_pin4, LOW);
        break;

    case 1: // 0110
        digitalWrite(_pin1, LOW);
        digitalWrite(_pin2, HIGH);
case 2: //0101
  digitalWrite(_pin1, LOW);
  digitalWrite(_pin2, HIGH);
  digitalWrite(_pin3, LOW);
  digitalWrite(_pin4, HIGH);
  break;

case 3: //1001
  digitalWrite(_pin1, HIGH);
  digitalWrite(_pin2, LOW);
  digitalWrite(_pin3, LOW);
  digitalWrite(_pin4, HIGH);
  break;

// 4 pin step function
// This is passed the current step number (0 to 7)
// Subclasses can override
void AccelStepper::step8(uint8_t step)
{
  switch (step & 0x7)
  {
    case 0: // 1000
      digitalWrite(_pin1, HIGH);
      digitalWrite(_pin2, LOW);
      digitalWrite(_pin3, LOW);
      digitalWrite(_pin4, LOW);
      break;

    case 1: // 1010
      digitalWrite(_pin1, HIGH);
      digitalWrite(_pin2, LOW);
      digitalWrite(_pin3, HIGH);
      digitalWrite(_pin4, LOW);
      break;

    case 2: // 0010
      digitalWrite(_pin1, LOW);
      digitalWrite(_pin2, LOW);
      digitalWrite(_pin3, HIGH);
      digitalWrite(_pin4, LOW);
      break;

    case 3: // 0110
      digitalWrite(_pin1, LOW);
      digitalWrite(_pin2, HIGH);
      digitalWrite(_pin3, HIGH);
      digitalWrite(_pin4, LOW);
      break;
  }
}
case 4: // 0100
  digitalWrite(_pin1, LOW);
  digitalWrite(_pin2, HIGH);
  digitalWrite(_pin3, LOW);
  digitalWrite(_pin4, LOW);
  break;

case 5: // 0101
  digitalWrite(_pin1, LOW);
  digitalWrite(_pin2, HIGH);
  digitalWrite(_pin3, LOW);
  digitalWrite(_pin4, HIGH);
  break;

case 6: // 0001
  digitalWrite(_pin1, LOW);
  digitalWrite(_pin2, LOW);
  digitalWrite(_pin3, LOW);
  digitalWrite(_pin4, HIGH);
  break;

case 7: // 1001
  digitalWrite(_pin1, HIGH);
  digitalWrite(_pin2, LOW);
  digitalWrite(_pin3, LOW);
  digitalWrite(_pin4, HIGH);
  break;

} // Prevents power consumption on the outputs
void AccelStepper::disableOutputs()
{
  if (!_pins) return;

  if (_pins == 1)
  {
    // Invert only applies for stepper drivers.
    digitalWrite(_pin1, LOW ^ _stepInverted);
    digitalWrite(_pin2, LOW ^ _dirInverted);
  }
  else
  {
    digitalWrite(_pin1, LOW);
    digitalWrite(_pin2, LOW);
  }

  if (_pins == 4 || _pins == 8)
  {
    digitalWrite(_pin3, LOW);
    digitalWrite(_pin4, LOW);
  }

  if (_enablePin != 0xff)
  {
    digitalWrite(_enablePin, LOW ^ _enableInverted);
  }
void AccelStepper::enableOutputs()
{
    if (! _pins)
        return;

    pinMode(_pin1, OUTPUT);
    pinMode(_pin2, OUTPUT);
    if (_pins == 4 || _pins == 8)
    {
        pinMode(_pin3, OUTPUT);
        pinMode(_pin4, OUTPUT);
    }

    if (_enablePin != 0xff)
    {
        pinMode(_enablePin, OUTPUT);
        digitalWrite(_enablePin, HIGH ^ _enableInverted);
    }
}

void AccelStepper::setMinPulseWidth(unsigned int minWidth)
{
    _minPulseWidth = minWidth;
}

void AccelStepper::setEnablePin(uint8_t enablePin)
{
    _enablePin = enablePin;

    // This happens after construction, so init pin now.
    if (_enablePin != 0xff)
    {
        pinMode(_enablePin, OUTPUT);
        digitalWrite(_enablePin, HIGH ^ _enableInverted);
    }
}

void AccelStepper::setPinsInverted(bool direction, bool step, bool enable)
{
    _dirInverted    = direction;
    _stepInverted   = step;
    _enableInverted = enable;
}

// Blocks until the target position is reached
void AccelStepper::runToPosition()
{
    while (run())
    {
    }
}

boolean AccelStepper::runSpeedToPosition()
if (_targetPos > _currentPos)
    _speed = fabs(_speed);
else
    _speed = -fabs(_speed);
return _targetPos!=_currentPos ? runSpeed() : false;
}

// Blocks until the new target position is reached
void AccelStepper::runToNewPosition(long position)
{
    moveTo(position);
    runToPosition();
}
Appendix M

LIFA VIs Used in Flow System Programs

Figure M-1: LIFA Auto Detect VI front panel, full view.

Figure M-2: LIFA Auto Detect VI block diagram, full view, true.
Figure M-3: LIFA Auto Detect VI block diagram, full view, false.

Figure M-4: LIFA Close VI front panel, full view.

Figure M-5: LIFA Close VI block diagram, full view.
Figure M-6: LIFA Initialize VI front panel, full view.

Figure M-7: LIFA Initialize VI block diagram, left view, default, true.
Figure M-8: LIFA Initialize VI block diagram, left view, Mega 2560, false.

Figure M-9: LIFA Initialize VI block diagram, right view, no error, USB/serial, default.
Figure M-10: LIFA Initialize VI block diagram, right view, no error, default, no timeout, default.

Figure M-11: LIFA Initialize VI block diagram, right view, no error, default, no timeout, sync.
Figure M-12: LIFA Initialize VI block diagram, right view, error.

Figure M-13: LIFA Packetize VI front panel, full view.
Figure M-14: LIFA Packetize VI block diagram, full view.

Figure M-15: LIFA Send/Receive VI front panel, full view.
Figure M-16: LIFA Send/Receive VI block diagram, full view, no error, false, false.

Figure M-17: LIFA Send/Receive VI block diagram, full view, no error, true, true.

Figure M-18: LIFA Send/Receive VI block diagram, full view, error.
Figure M-19: LIFA Stepper Close VI front panel, full view.

Figure M-20: LIFA Stepper Close VI block diagram, full view.
Figure M-21:  LIFA Stepper ToGo VI front panel, full view.

Figure M-22:  LIFA Stepper ToGo VI block diagram, full view.
Figure M-23: LIFA Stepper Write VI front panel, full view.

Figure M-24: LIFA Stepper Write VI block diagram, full view, case 1.
Figure M-25: LIFA Stepper Write VI block diagram, full view, default.

Figure M-26: LIFA Wait for Bytes VI front panel, full view.
Figure M-27: LIFA Wait for Bytes VI block diagram, full view.

Figure M-28: LIFA Stepper Configure VI front panel, full view.
Figure M-29: LIFA Stepper Configure VI block diagram, full view, default.

Figure M-30: LIFA Stepper Configure VI block diagram, full view, case 1.

Figure M-31: LIFA Check Pins VI front panel, full view.
Figure M-32: LIFA Check Pins VI block diagram, full view, Mega 2560, analog.

Figure M-33: LIFA Check Pins VI block diagram, full view, Mega 2560, digital.

Figure M-34: LIFA Check Pins VI block diagram, full view, Uno, analog.
Figure M-35: LIFA Check Pins VI block diagram, full view, Uno, digital.
Appendix N

Flow System Control VIs

Figure N-1: Flow System Purge VI front panel, full view.
Figure N-2: Flow System Purge VI block diagram, left view.

Figure N-3: Flow System Purge VI block diagram, center view, OK button.

Figure N-4: Flow System Purge VI block diagram, center view, timeout.
Figure N-5: Flow System Purge VI block diagram, right view.

Figure N-6: Flow System Concentration Change VI front panel, full view.
Figure N-7: Flow System Concentration Change VI block diagram, left view, true, change concentration, false, false, 0.

Figure N-8: Flow System Concentration Change VI block diagram, left view, false, timeout, false, false, 1.
Figure N-9: Flow System Concentration Change VI block diagram, left view, false, timeout, false, false, 2.

Figure N-10: Flow System Concentration Change VI block diagram, left view, false, timeout, false, true.
Figure N-11: Flow System Concentration Change VI block diagram, left view, false, timeout, true.

Figure N-12: Flow System Concentration Change VI block diagram, left-center view, false, true, false, false, 0, true.
Figure N-13: Flow System Concentration Change VI block diagram, left-center view, false, false, false, false, 1, false.

Figure N-14: Flow System Concentration Change VI block diagram, left-center view, false, false, false, false, 1, false.
Figure N-15: Flow System Concentration Change VI block diagram, left-center view, false, false, false, true, false.

Figure N-16: Flow System Concentration Change VI block diagram, left-center view, false, false, true, false.
Figure N-17: Flow System Concentration Change VI block diagram, left-center view, true.

Figure N-18: Flow System Concentration Change VI block diagram, right-center view, false, true, false, false, 0.
Figure N-19: Flow System Concentration Change VI block diagram, right-center view, false, false, false, false, 1.

Figure N-20: Flow System Concentration Change VI block diagram, right-center view, false, false, false, false, 2.
Figure N-21: Flow System Concentration Change VI block diagram, right-center view, false, false, false, true.

Figure N-22: Flow System Concentration Change VI block diagram, right-center view, false, false, true.
Figure N-23: Flow System Concentration Change VI block diagram, right-center view, true.

Figure N-24: Flow System Concentration Change VI block diagram, right view.
Figure N-25: Flow System Profile Control VI front panel, full view.

Figure N-26: Flow System Profile Control VI block diagram, left view.
Figure N-27: Flow System Profile Control VI block diagram, left-center view, true, true, 0, true, false, false, 0, true.

Figure N-28: Flow System Profile Control VI block diagram, left-center view, false, true, 1, false, false, false, 1, false.
Figure N-29: Flow System Profile Control VI block diagram, left-center view, false, false, false, false, false, 2, false.

Figure N-30: Flow System Profile Control VI block diagram, left-center view, false, false, false, false, true, false.
Figure N-31: Flow System Profile Control VI block diagram, left-center view, false, false, false, true, false.

Figure N-32: Flow System Profile Control VI block diagram, left-center view, false, false, false, true.
Figure N-33: Flow System Profile Control VI block diagram, center view, false, false, false, 0, true, false, false, 0.

Figure N-34: Flow System Profile Control VI block diagram, center view, false, false, false, 1, false, false, false, 1.
Figure N-35: Flow System Profile Control VI block diagram, center view, false, false, false, 2, false, false, false, 2.

Figure N-36: Flow System Profile Control VI block diagram, center view, false, false, true, false, false, true.
Figure N-37: Flow System Profile Control VI block diagram, center view, false, true, false, true.

Figure N-38: Flow System Profile Control VI block diagram, center view, true.
Figure N-39: Flow System Profile Control VI block diagram, right-center view.
Figure N-40: Flow System Profile Control VI block diagram, right view.
Figure N-41: Flow System Pump Configuration VI front panel, full view.

Figure N-42: Flow System Pump Configuration VI block diagram, full view.
Figure N-43: Flow System Stepper Disconnect VI front panel, full view.

Figure N-44: Flow System Stepper Disconnect VI block diagram, full view, false, false, false.
Figure N-45: Flow System Stepper Disconnect VI block diagram, full view, true, true, true.

Figure N-46: Flow System Step Calculation VI front panel, full view.
Figure N-47: Flow System Step Calculation VI block diagram, full view.
### Appendix O

#### Raw Collected Data

**Table O.1:** Linear voltage sweep data from the 6.8 mH coil in compensation collected from the FEM.

<table>
<thead>
<tr>
<th>Voltage (VDC)</th>
<th>0.1</th>
<th>0.3</th>
<th>0.5</th>
<th>1.0</th>
<th>1.5</th>
<th>2.0</th>
</tr>
</thead>
<tbody>
<tr>
<td>Depth (deg.)</td>
<td>0.0165</td>
<td>0.0494</td>
<td>0.0823</td>
<td>0.1646</td>
<td>0.2470</td>
<td>0.3292</td>
</tr>
</tbody>
</table>

**Table O.2:** Linear voltage sweep data from the 15 mH coil in compensation collected from the FEM.

<table>
<thead>
<tr>
<th>Voltage (VDC)</th>
<th>0.1</th>
<th>0.5</th>
<th>1.0</th>
<th>2.0</th>
<th>3.0</th>
<th>4.0</th>
</tr>
</thead>
<tbody>
<tr>
<td>Depth (deg.)</td>
<td>0.0026</td>
<td>0.0131</td>
<td>0.0264</td>
<td>0.0527</td>
<td>0.0791</td>
<td>0.1055</td>
</tr>
</tbody>
</table>

**Table O.3:** Linear voltage sweep data from the 68 mH coil in compensation collected from the FEM.

<table>
<thead>
<tr>
<th>Voltage (VDC)</th>
<th>0.1</th>
<th>0.3</th>
<th>0.5</th>
<th>1.0</th>
<th>2.0</th>
<th>3.0</th>
</tr>
</thead>
<tbody>
<tr>
<td>Depth (deg.)</td>
<td>0.0050</td>
<td>0.0149</td>
<td>0.0249</td>
<td>0.0499</td>
<td>0.0997</td>
<td>0.1496</td>
</tr>
</tbody>
</table>

**Table O.4:** Linear voltage sweep data from the 100 mH coil in compensation collected from the FEM.

<table>
<thead>
<tr>
<th>Voltage (VDC)</th>
<th>0.1</th>
<th>0.5</th>
<th>1.0</th>
<th>2.0</th>
<th>4.0</th>
<th>6.0</th>
</tr>
</thead>
<tbody>
<tr>
<td>Depth (deg.)</td>
<td>0.0039</td>
<td>0.0199</td>
<td>0.0397</td>
<td>0.0796</td>
<td>0.1592</td>
<td>0.2387</td>
</tr>
</tbody>
</table>
Table O.5: Linear voltage sweep data from the 220 mH coil in compensation collected from the FEM.

<table>
<thead>
<tr>
<th>Voltage (VDC)</th>
<th>0.1</th>
<th>1.0</th>
<th>3.0</th>
<th>5.0</th>
<th>7.0</th>
<th>10.0</th>
</tr>
</thead>
<tbody>
<tr>
<td>Depth (deg.)</td>
<td>0.007</td>
<td>0.0075</td>
<td>0.0223</td>
<td>0.0371</td>
<td>0.0518</td>
<td>0.0741</td>
</tr>
</tbody>
</table>

Table O.6: Linear voltage sweep data from the 470 mH coil in compensation collected from the FEM.

<table>
<thead>
<tr>
<th>Voltage (VDC)</th>
<th>0.1</th>
<th>1.0</th>
<th>3.0</th>
<th>5.0</th>
<th>7.0</th>
<th>10.0</th>
</tr>
</thead>
<tbody>
<tr>
<td>Depth (deg.)</td>
<td>0.0056</td>
<td>0.0168</td>
<td>0.0279</td>
<td>0.0390</td>
<td>0.0558</td>
<td></td>
</tr>
</tbody>
</table>

Table O.7: Linear voltage sweep data from the 6.8 mH coil in modulation collected from the FEM.

<table>
<thead>
<tr>
<th>Voltage (V_{pk})</th>
<th>0.1</th>
<th>1.0</th>
<th>5.0</th>
<th>10.0</th>
<th>22.34</th>
<th>30.0</th>
</tr>
</thead>
<tbody>
<tr>
<td>Depth (deg.)</td>
<td>0.0028</td>
<td>0.0275</td>
<td>0.1379</td>
<td>0.2757</td>
<td>0.6159</td>
<td>0.8271</td>
</tr>
</tbody>
</table>

Table O.8: Linear voltage sweep data from the 15 mH coil in modulation collected from the FEM.

<table>
<thead>
<tr>
<th>Voltage (V_{pk})</th>
<th>1.0</th>
<th>10.0</th>
<th>25.0</th>
<th>40.0</th>
<th>68.24</th>
<th>100.0</th>
</tr>
</thead>
<tbody>
<tr>
<td>Depth (deg.)</td>
<td>0.0082</td>
<td>0.0816</td>
<td>0.2039</td>
<td>0.3262</td>
<td>0.5566</td>
<td>0.8157</td>
</tr>
</tbody>
</table>

Table O.9: Linear voltage sweep data from the 68 mH coil in modulation collected from the FEM.

<table>
<thead>
<tr>
<th>Voltage (V_{pk})</th>
<th>1.0</th>
<th>10.0</th>
<th>25.0</th>
<th>40.0</th>
<th>71.42</th>
<th>100.0</th>
</tr>
</thead>
<tbody>
<tr>
<td>Depth (deg.)</td>
<td>0.0068</td>
<td>0.0683</td>
<td>0.1709</td>
<td>0.2733</td>
<td>0.4881</td>
<td>0.6835</td>
</tr>
</tbody>
</table>

Table O.10: Linear voltage sweep data from the 100 mH coil in modulation collected from the FEM.

<table>
<thead>
<tr>
<th>Voltage (V_{pk})</th>
<th>1.0</th>
<th>10.0</th>
<th>25.0</th>
<th>40.0</th>
<th>71.42</th>
<th>100.0</th>
</tr>
</thead>
<tbody>
<tr>
<td>Depth (deg.)</td>
<td>0.0068</td>
<td>0.0683</td>
<td>0.1709</td>
<td>0.2733</td>
<td>0.4881</td>
<td>0.6835</td>
</tr>
</tbody>
</table>
Table O.11: Linear voltage sweep data from the 220 mH coil in modulation collected from the FEM.

<table>
<thead>
<tr>
<th>Voltage (V&lt;sub&gt;pk&lt;/sub&gt;)</th>
<th>1.0</th>
<th>10.0</th>
<th>25.0</th>
<th>40.0</th>
<th>70.0</th>
<th>100.0</th>
</tr>
</thead>
<tbody>
<tr>
<td>Depth (deg.)</td>
<td>0.0034</td>
<td>0.0343</td>
<td>0.0855</td>
<td>0.1368</td>
<td>0.2395</td>
<td>0.3421</td>
</tr>
</tbody>
</table>

Table O.12: Linear voltage sweep data from the 470 mH coil in modulation collected from the FEM.

<table>
<thead>
<tr>
<th>Voltage (V&lt;sub&gt;pk&lt;/sub&gt;)</th>
<th>1.0</th>
<th>10.0</th>
<th>25.0</th>
<th>40.0</th>
<th>70.0</th>
<th>117.0</th>
</tr>
</thead>
<tbody>
<tr>
<td>Depth (deg.)</td>
<td>0.0021</td>
<td>0.0204</td>
<td>0.0510</td>
<td>0.0817</td>
<td>0.1429</td>
<td>0.2390</td>
</tr>
</tbody>
</table>

Table O.13: Maximum rotational data from each coil in modulation.

<table>
<thead>
<tr>
<th>Inductor (mH)</th>
<th>6.8</th>
<th>15</th>
<th>68</th>
<th>100</th>
<th>220</th>
<th>470</th>
<th>IFMC</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Coil Voltage (V&lt;sub&gt;pk&lt;/sub&gt;)</strong></td>
<td>22.3</td>
<td>17.9</td>
<td>68.2</td>
<td>71.4</td>
<td>100.0</td>
<td>117.0</td>
<td>192</td>
</tr>
<tr>
<td><strong>Power (W)</strong></td>
<td>1.44</td>
<td>0.79</td>
<td>0.74</td>
<td>0.49</td>
<td>1.00</td>
<td>0.64</td>
<td>1.99</td>
</tr>
<tr>
<td><strong>FEM Depth (deg.)</strong></td>
<td>0.6159</td>
<td>0.2530</td>
<td>0.5566</td>
<td>0.4881</td>
<td>0.3421</td>
<td>0.2389</td>
<td>1.2404</td>
</tr>
<tr>
<td><strong>RTPR (°/W)</strong></td>
<td>0.4277</td>
<td>0.3203</td>
<td>0.7522</td>
<td>0.9962</td>
<td>0.3421</td>
<td>0.3734</td>
<td>0.6233</td>
</tr>
<tr>
<td><strong>1&lt;sup&gt;st&lt;/sup&gt; Measured Depth (deg.)</strong></td>
<td>0.6215</td>
<td>0.2651</td>
<td>0.5616</td>
<td>0.4866</td>
<td>0.3449</td>
<td>0.2381</td>
<td>1.2158</td>
</tr>
<tr>
<td><strong>2&lt;sup&gt;nd&lt;/sup&gt; Measured Depth (deg.)</strong></td>
<td>0.6126</td>
<td>0.2626</td>
<td>0.5506</td>
<td>0.4965</td>
<td>0.3444</td>
<td>0.2250</td>
<td>1.2023</td>
</tr>
<tr>
<td><strong>3&lt;sup&gt;rd&lt;/sup&gt; Measured Depth (deg.)</strong></td>
<td>0.6168</td>
<td>0.2578</td>
<td>0.5657</td>
<td>0.4923</td>
<td>0.3511</td>
<td>0.2401</td>
<td>1.2497</td>
</tr>
<tr>
<td><strong>AVG Measured Depth (deg.)</strong></td>
<td>0.6170</td>
<td>0.2618</td>
<td>0.5593</td>
<td>0.4918</td>
<td>0.3468</td>
<td>0.2344</td>
<td>1.2226</td>
</tr>
</tbody>
</table>
Table O.14: Maximum rotational data from each coil in compensation.

<table>
<thead>
<tr>
<th>Inductor (mH)</th>
<th>6.8</th>
<th>15</th>
<th>68</th>
<th>100</th>
<th>220</th>
<th>470</th>
<th>IFMC</th>
</tr>
</thead>
<tbody>
<tr>
<td>Voltage (VDC)</td>
<td>1.62</td>
<td>3.89</td>
<td>2.92</td>
<td>5.91</td>
<td>9.98</td>
<td>9.99</td>
<td>10.00</td>
</tr>
<tr>
<td>FEM Depth (deg.)</td>
<td>0.2667</td>
<td>0.1026</td>
<td>0.1455</td>
<td>0.2351</td>
<td>0.0740</td>
<td>0.0558</td>
<td>0.0780</td>
</tr>
<tr>
<td>1&lt;sup&gt;st&lt;/sup&gt; Measured Depth (deg.)</td>
<td>0.2511</td>
<td>0.1037</td>
<td>0.1406</td>
<td>0.2256</td>
<td>0.0787</td>
<td>0.0588</td>
<td>0.0792</td>
</tr>
<tr>
<td>2&lt;sup&gt;nd&lt;/sup&gt; Measured Depth (deg.)</td>
<td>0.2501</td>
<td>0.1084</td>
<td>0.1443</td>
<td>0.2167</td>
<td>0.0791</td>
<td>0.0568</td>
<td>0.0792</td>
</tr>
<tr>
<td>3&lt;sup&gt;rd&lt;/sup&gt; Measured Depth (deg.)</td>
<td>0.2495</td>
<td>0.1026</td>
<td>0.1428</td>
<td>0.2256</td>
<td>0.0777</td>
<td>0.0578</td>
<td>0.0797</td>
</tr>
<tr>
<td>AVG Measured Depth (deg.)</td>
<td>0.2502</td>
<td>0.1049</td>
<td>0.1426</td>
<td>0.2226</td>
<td>0.0785</td>
<td>0.0578</td>
<td>0.0794</td>
</tr>
</tbody>
</table>

Table O.15: Compensator output voltage from the IFMC during static polarimetric glucose sampling in the hypoglycemic range.

<table>
<thead>
<tr>
<th>Concentration (mg/dL)</th>
<th>Trial 1 Voltage (VDC)</th>
<th>Trial 2 Voltage (VDC)</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>2.36</td>
<td>2.37</td>
</tr>
<tr>
<td>10</td>
<td>2.49</td>
<td>2.47</td>
</tr>
<tr>
<td>20</td>
<td>2.58</td>
<td>2.57</td>
</tr>
<tr>
<td>30</td>
<td>2.67</td>
<td>2.64</td>
</tr>
<tr>
<td>40</td>
<td>2.75</td>
<td>2.71</td>
</tr>
<tr>
<td>50</td>
<td>2.85</td>
<td>2.84</td>
</tr>
<tr>
<td>60</td>
<td>2.94</td>
<td>2.96</td>
</tr>
<tr>
<td>70</td>
<td>3.08</td>
<td>3.04</td>
</tr>
<tr>
<td>80</td>
<td>3.13</td>
<td>3.13</td>
</tr>
<tr>
<td>90</td>
<td>3.23</td>
<td>3.24</td>
</tr>
<tr>
<td>100</td>
<td>3.34</td>
<td>3.30</td>
</tr>
</tbody>
</table>
Table O.16: Compensator output voltage from the IFMC during static polarimetric glucose sampling in the hyperglycemic range.

<table>
<thead>
<tr>
<th>Concentration (mg/dL)</th>
<th>Trial 1 Voltage (VDC)</th>
<th>Trial 2 Voltage (VDC)</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>2.37</td>
<td>2.34</td>
</tr>
<tr>
<td>50</td>
<td>2.87</td>
<td>2.90</td>
</tr>
<tr>
<td>100</td>
<td>3.41</td>
<td>3.38</td>
</tr>
<tr>
<td>150</td>
<td>3.87</td>
<td>3.88</td>
</tr>
<tr>
<td>200</td>
<td>4.39</td>
<td>4.34</td>
</tr>
<tr>
<td>250</td>
<td>4.87</td>
<td>4.84</td>
</tr>
<tr>
<td>300</td>
<td>5.34</td>
<td>5.31</td>
</tr>
<tr>
<td>350</td>
<td>5.80</td>
<td>5.86</td>
</tr>
<tr>
<td>400</td>
<td>6.32</td>
<td>6.27</td>
</tr>
<tr>
<td>450</td>
<td>6.72</td>
<td>6.65</td>
</tr>
<tr>
<td>500</td>
<td>7.21</td>
<td>7.16</td>
</tr>
<tr>
<td>550</td>
<td>7.61</td>
<td>7.62</td>
</tr>
<tr>
<td>600</td>
<td>8.08</td>
<td>8.07</td>
</tr>
</tbody>
</table>

Table O.17: YSI glucose measurement data from solutions created with the custom flow system in the hypoglycemic range.

<table>
<thead>
<tr>
<th>Predicted Concentration (mg/dL)</th>
<th>Measured Concentration Trial 1 (mg/dL)</th>
<th>Measured Concentration Trial 2 (mg/dL)</th>
<th>Measured Concentration Trial 3 (mg/dL)</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0.046</td>
<td>0.000</td>
<td>0.091</td>
</tr>
<tr>
<td>10</td>
<td>10.5</td>
<td>10.6</td>
<td>10.5</td>
</tr>
<tr>
<td>20</td>
<td>20.2</td>
<td>20.3</td>
<td>20.6</td>
</tr>
<tr>
<td>30</td>
<td>30.1</td>
<td>30.2</td>
<td>30.1</td>
</tr>
<tr>
<td>40</td>
<td>40.2</td>
<td>39.6</td>
<td>39.6</td>
</tr>
<tr>
<td>50</td>
<td>49.8</td>
<td>49.9</td>
<td>49.9</td>
</tr>
<tr>
<td>60</td>
<td>59.8</td>
<td>59.8</td>
<td>59.4</td>
</tr>
<tr>
<td>70</td>
<td>69.7</td>
<td>69.0</td>
<td>70.6</td>
</tr>
<tr>
<td>80</td>
<td>81.2</td>
<td>80.9</td>
<td>80.3</td>
</tr>
<tr>
<td>90</td>
<td>91.2</td>
<td>89.3</td>
<td>90.0</td>
</tr>
<tr>
<td>100</td>
<td>98.8</td>
<td>98.8</td>
<td>98.9</td>
</tr>
</tbody>
</table>
Table O.18: YSI glucose measurement data from solutions created with the custom flow system in the hyperglycemic range.

<table>
<thead>
<tr>
<th>Predicted Concentration (mg/dL)</th>
<th>Measured Concentration Trial 1 (mg/dL)</th>
<th>Measured Concentration Trial 2 (mg/dL)</th>
<th>Measured Concentration Trial 3 (mg/dL)</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0.000</td>
<td>0.000</td>
<td>0.000</td>
</tr>
<tr>
<td>50</td>
<td>47.7</td>
<td>48.9</td>
<td>48.5</td>
</tr>
<tr>
<td>100</td>
<td>100</td>
<td>101</td>
<td>99.8</td>
</tr>
<tr>
<td>150</td>
<td>146</td>
<td>145</td>
<td>145</td>
</tr>
<tr>
<td>200</td>
<td>194</td>
<td>197</td>
<td>197</td>
</tr>
<tr>
<td>250</td>
<td>251</td>
<td>246</td>
<td>249</td>
</tr>
<tr>
<td>300</td>
<td>291</td>
<td>296</td>
<td>295</td>
</tr>
<tr>
<td>350</td>
<td>340</td>
<td>335</td>
<td>341</td>
</tr>
<tr>
<td>400</td>
<td>400</td>
<td>399</td>
<td>396</td>
</tr>
<tr>
<td>450</td>
<td>450</td>
<td>440</td>
<td>448</td>
</tr>
<tr>
<td>500</td>
<td>478</td>
<td>481</td>
<td>489</td>
</tr>
<tr>
<td>550</td>
<td>524</td>
<td>524</td>
<td>544</td>
</tr>
<tr>
<td>600</td>
<td>596</td>
<td>580</td>
<td>592</td>
</tr>
</tbody>
</table>

Table O.19: Compensator output voltage from the IFMC during dynamic polarimetric glucose sampling through the custom flow cell, trial 1.

<table>
<thead>
<tr>
<th>Concentration (mg/dL)</th>
<th>Measured Voltage (VDC)</th>
<th>Baseline Voltage (VDC)</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>2.62</td>
<td>2.62</td>
</tr>
<tr>
<td>20</td>
<td>3.06</td>
<td>2.85</td>
</tr>
<tr>
<td>40</td>
<td>3.04</td>
<td>2.7</td>
</tr>
<tr>
<td>60</td>
<td>3.36</td>
<td>2.77</td>
</tr>
<tr>
<td>80</td>
<td>3.68</td>
<td>2.82</td>
</tr>
<tr>
<td>100</td>
<td>3.63</td>
<td>2.7</td>
</tr>
<tr>
<td>120</td>
<td>3.95</td>
<td>2.7</td>
</tr>
<tr>
<td>140</td>
<td>4.17</td>
<td>2.74</td>
</tr>
<tr>
<td>160</td>
<td>4.35</td>
<td>2.85</td>
</tr>
<tr>
<td>180</td>
<td>4.45</td>
<td>2.7</td>
</tr>
<tr>
<td>200</td>
<td>4.74</td>
<td>2.83</td>
</tr>
<tr>
<td>Concentration (mg/dL)</td>
<td>Measured Voltage (VDC)</td>
<td>Baseline Voltage (VDC)</td>
</tr>
<tr>
<td>----------------------</td>
<td>------------------------</td>
<td>------------------------</td>
</tr>
<tr>
<td>0</td>
<td>2.88</td>
<td>2.88</td>
</tr>
<tr>
<td>20</td>
<td>2.88</td>
<td>2.70</td>
</tr>
<tr>
<td>40</td>
<td>3.20</td>
<td>2.80</td>
</tr>
<tr>
<td>60</td>
<td>3.27</td>
<td>2.78</td>
</tr>
<tr>
<td>80</td>
<td>3.56</td>
<td>2.93</td>
</tr>
<tr>
<td>100</td>
<td>3.74</td>
<td>2.80</td>
</tr>
<tr>
<td>120</td>
<td>3.99</td>
<td>2.70</td>
</tr>
<tr>
<td>140</td>
<td>4.19</td>
<td>2.79</td>
</tr>
<tr>
<td>160</td>
<td>4.35</td>
<td>2.80</td>
</tr>
<tr>
<td>180</td>
<td>4.52</td>
<td>2.74</td>
</tr>
<tr>
<td>200</td>
<td>4.70</td>
<td>2.64</td>
</tr>
</tbody>
</table>

Table O.21: Compensator output voltage from the IFMC during dynamic polarimetric glucose sampling through the custom flow cell, trial 3.

<table>
<thead>
<tr>
<th>Concentration (mg/dL)</th>
<th>Measured Voltage (VDC)</th>
<th>Baseline Voltage (VDC)</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>2.79</td>
<td>2.79</td>
</tr>
<tr>
<td>20</td>
<td>2.82</td>
<td>2.65</td>
</tr>
<tr>
<td>40</td>
<td>3.11</td>
<td>2.69</td>
</tr>
<tr>
<td>60</td>
<td>3.31</td>
<td>2.72</td>
</tr>
<tr>
<td>80</td>
<td>3.37</td>
<td>2.64</td>
</tr>
<tr>
<td>100</td>
<td>3.80</td>
<td>2.80</td>
</tr>
<tr>
<td>120</td>
<td>3.82</td>
<td>2.60</td>
</tr>
<tr>
<td>140</td>
<td>4.20</td>
<td>2.71</td>
</tr>
<tr>
<td>160</td>
<td>4.25</td>
<td>2.65</td>
</tr>
<tr>
<td>180</td>
<td>4.57</td>
<td>2.75</td>
</tr>
<tr>
<td>200</td>
<td>4.65</td>
<td>2.62</td>
</tr>
</tbody>
</table>