THESIS

Electrochemical Biosensor Array Characterization

Submitted by

Matthew W. Jibson

Department of Electrical and Computer Engineering

In partial fulfillment of the requirements

for the Degree of Master of Science

Colorado State Univesity

Fort Collins, Colorado

Summer, 2009

COLORADO STATE UNIVERSITY

May 29, 2009

WE HEREBY RECOMMEND THAT THE THESIS PREPARED UNDER OUR SUPERVISION BY MATTHEW W. JIBSON ENTITLED ELECTROCHEMICAL BIOSENSOR ARRAY CHARACTERIZATION BE ACCEPTED AS FULFILLING IN PART REQUIREMENTS FOR THE DEGREE OF MASTER OF SCIENCE.

Committee on Graduate Work

Committee Member (Dr. Charles Henry)

Committee Member (Dr. Stuart Tobet)

Committee Member (Dr. Anthony Maciejewski)

Advisor (Dr. Tom Chen)

Department Head (Dr. Anthony Maciejewski)

ABSTRACT OF THESIS

Electrochemical Biosensor Array Characterization

Neurotransmitters play an important role in central nervous systems. Nitric oxide, a neurotransmitter, is important in this development. Of interest is detecting molecular gradients that are essential in the development of tissue and organ systems. Molecular gradients are difficult to detect because of the relative large size of the cells compared to the electrochemical sensors used in sensing systems. Furthermore, in order to detect a gradient, an sensor array must be used in order to collect real-time spatial data. Due to this requirement of a sensor array, it is difficult to construct a device with discrete parts, since it would be quite large. Thus, an integrated sensor must be constructed. Integration allows components to be small enough to have many sensors in the area of a cell, and is thus able to sense a chemical image, or gradient.

Previous work has resulted in the production of a chip with an array of 21 sensor sites of individual and specific design with the purpose of testing hypotheses relating the shape, size, distance and configuration to the output signal strength. The electrodes are on the micron scale, and are capable of performing electrochemistry on living cells. The sensor sites were characterized using differential pulse voltammetry to find their relative performance. Based on these results, further tests were performed to test hypotheses regarding the shape, size, distance and configuration of the electrodes. The lower detection limit is found on two of the best sensors. A proof-of-concept test is done with a living mouse-ovary slice, which showed results similar to those in the literature.

Results show that the important design characteristics are working-electrode size (larger is better), and the ratio of the areas of the working to auxiliary electrode (smaller ratio is better). The other design characteristics (distance, shape, configuration) played, in general, did not have much impact on the output. Conclusions about the design of future chips is made based on these findings.

Matthew W. Jibson
Electrical and Computer Engineering Department
Colorado State University
Fort Collins, CO 80523
Summer, 2009

Contents

1 Introduction **1**

1.1 Neurotransmitter Detection 1

1.2 Objectives 2

2 Background and Existing Approaches **3**

2.1 Introduction to Electrochemistry 3

2.2 Existing Approaches 6

2.3 Summary 7

3 Chip Design and Characterization Results **8**

3.1 Sensor Configurations 10

3.1.1 Working Electrodes 10

3.1.2 Sites 11

3.1.3 Sensors 12

3.2 Experiment Setup 19

3.3 Experiment Procedure 24

3.3.1 All Sites 24

3.3.2 Limit of Detection 24

3.3.3 Specific Electrodes 24

3.4 Experiment Results 24

3.4.1 Output vs. Working Electrode Area 24

3.4.2 Output Density vs. Working Electrode Area 26

3.4.3 Ratio of Working to Auxiliary Electrode Areas 26

3.4.4 Working Electrode Perimeter over Area vs. Output Density 28

3.4.5 Distance from Working to Auxiliary Electrode 28

3.5 Limit of Detection Results 28

4 Slice Testing **32**

4.1 Slice Information 32

4.2 Experiment Procedure 32

4.3 Experiment Results 34

4.4 Discussion 34

5 Discussion and Conclusion **36**

A Source Code **40**

B Screenshots **49**

C Solution Mixing **54**

List of Figures

2.1 Schematic representation of potential gradients in a three-electrode cell 4
2.2 Schematic representation of a primitive three-electrode controlled-potential apparatus 4
2.3 Comparison of inputs of electrochemistry techniques 5
2.4 Comparison of results of electrochemistry techniques 6

3.1 Sensors layout of the chip 9
3.2 Full layout of the chip 9
3.3 "C"-type working-electrode pair 11
3.4 "Inverse C"-type working-electrode pair 11
3.5 "F"-type working-electrode pair 11
3.6 Sensor 0 12
3.7 Sensor 1 12
3.8 Sensor 2 13
3.9 Sensor 3 13
3.10 Sensor 4 14
3.11 Sensor 5 14
3.12 Sensor 6 14
3.13 Sensor 7 14
3.14 Sensor 8 15
3.15 Sensor 9 15
3.16 Sensor 10 15
3.17 Sensor 11 15
3.18 Sensor 12 16
3.19 Sensor 13 16
3.20 Sensor 14 17

3.21 Sensor 15 . . . 17
3.22 Sensor 16 . . . 17
3.23 Sensor 17 . . . 17
3.24 Sensor 18 . . . 18
3.25 Sensor 19 . . . 18
3.26 Sensor 20 . . . 18
3.27 Chip with PDMS well attached . . . 19
3.28 Probestation . . . 19
3.29 Pins lowered onto probe pads of the chip . . . 23
3.30 Pins lowered onto probe pads of the chip, top view . . . 23
3.31 Potentiostat leads connected to pins . . . 23
3.32 WE area vs. Output . . . 26
3.33 WE area vs. Density . . . 27
3.34 WEs with low area vs. Density . . . 27
3.35 WEs with high area vs. Density . . . 28
3.36 WE area / AE area vs. Output . . . 29
3.37 WE perimeter / area vs. Density . . . 29
3.38 Distance from WE to AE vs. Density . . . 30
3.39 Detection limits . . . 31

4.1 HDV for norepinephrine . . . 33
4.2 Top view of chip under testing with an ovary slice attached . . . 34
4.3 Amperometric pulses from ovary-slice test. . . . 35
4.4 Amperometric pulses from another group . . . 35

B.1 Web application main page . . . 50
B.2 Web application result detail . . . 51
B.3 Web application sensor results . . . 52
B.4 Web application electrode table . . . 53

List of Tables

3.1 Working-electrode areas and perimeters 12

3.2 Sensor site configurations 17

3.3 Electrode properties for sensors 0–7 20

3.4 Electrode properties for sensors 8–14 21

3.5 Electrode properties for sensors 15–20 22

3.6 DPV results for sensor sites 25

3.7 Results of further DPV tests on specific working electrodes 25

3.8 Detection limit results for sensor 17 30

3.9 Detection limit results for sensor 8 31

Chapter 1

Introduction

1.1 Neurotransmitter Detection

Neurotransmitters play an important role in central nervous systems. Nitric oxide (NO), a neurotransmitter, is important in this development [1–3]. Of interest is detecting molecular gradients that are essential in the development of tissue and organ systems [4, 5]. Molecular gradients are difficult to detect because of the relative large size of the cells compared to the electrochemical sensors used in sensing systems. Furthermore, in order to detect a gradient, an sensor array must be used in order to collect real-time spatial data. Due to this requirement of a sensor array, it is difficult to construct a device with discrete parts, since it would be quite large. Thus, an integrated sensor must be constructed. Integration allows components to be small enough to have many sensors in the area of a cell, and is thus able to sense a chemical image, or gradient.

For example, in order to construct a discrete sensor apparatus, one would first have to acquire individual electrodes on the micron scale, and a device able to hold them in place a few microns apart. This is theoretically possible using a probestation and micromanipulators. Next, some sort of covering would have to be placed sufficient that the cell slice could be mounted onto it, the sensors would be in contact with the slice, and the nutrient solution around the slice would be contained. This is theoretically possible by constructing a well and lowering the many pins (discussed next) into the well. In order to measure a gradient, let us assume a 2×2 array of sensors is used, which is 4 sensors, at 3 electrodes each, or 12 electrodes. (For comparison, our chip has more than 84 sensors, around 200 electrodes.) Positioning twelve micromanipulators on one probestation is unlikely to happen (six is difficult to do). Furthermore, lowering that many pins into a well without accidentally

shorting them, while getting them all within a few microns of each other, will not happen reliably (if ever). Assuming all of this did happen, the gradient measured is too small to be useful due to the small array size. In addition, one must have a potentiostat for every sensor. Potentiostats can cost a few thousand dollars and take up about half the volume of a normal computer case. Stacking 84 potentiostats near each other is prohibitively expensive and logistically unmanageable. Thus, discrete sensors are not practical for this work.

Integrated sensors do not have any of the described problems that discrete sensors do. Sensors can be mounted on chip in a location able to support an ovary and its surrounding solutions. Potentiostats can be integrated in the chip for each sensor [6–8]. The smaller a sensor is, the greater temporal and spatial resolution is possible since more sensors can be put into an identical area. This miniaturization allows finer gradients to be measured, and thus better conclusions can be drawn. However, miniaturization requires further smaller components and fabrication processes. This goal of a fine gradient continues to push interest and research (like this work) toward integrated sensors.

1.2 Objectives

This thesis is one step toward that goal of integrated sensors, specifically an integrated biosensor. Previous work in this lab has resulted in the design and subsequent fabrication of a chip with integrated electrodes [9]. The design of this chip had a number of hypotheses about electrode construction, and used many designs to test them. This thesis discusses those tests, their results, and conclusions. Specifically, we test the effects of varying shape, size, distance and configuration of electrodes, and are able to draw conclusions about the importance of each of those aspects.

The next chapter discusses the scientific background to the research as well as the previous and current work in the area of integrated biosensors. Chapter 3 documents the design and design hypotheses of the chip, the tests performed, and their results, drawing conclusions about the chip design hypotheses. Chapter 4 gives some proof-of-concept tests and results while testing live ovary slices. Chapter 5 draws conclusions about future work.

Chapter 2

Background and Existing Approaches

2.1 Introduction to Electrochemistry

Electrochemistry is a process for determining the compounds in a solution. This is accomplished by causing reduction or oxidation within the solution by raising or lowering the potential near some electrode surfaces and measuring the amount of current that is generated.

Specifically, three electrodes (reference, auxiliary, working) are placed into a solution with an electroactive analyte. The potential difference between the electrodes is raised which creates a potential gradient at both electrode-solution interfaces, but a zero potential gradient in the bulk solution between the electrodes (Figure 2.1). The field strength typically becomes zero at less than 10^{-6}m from the surface of the electrode [10]. When the electroactive analyte nears an electrode, if the potential is high enough, an electron will find favorable conditions in the analyte and transfer from the surface of the electrode to the analyte. At this point there is a concentration gradient between the new, reduced, analyte and the original. The reduced analyte begins to diffuse into the bulk solution, causing more reductions to occur. This happens continuously and at an exponentially decreasing rate until all of the analyte has been reduced. The continuing transfer of electrons causes current to flow. A similar response called oxidation occurs if the potential is lowered and the electron transfers back to the electrode from the analyte. The potential is kept constant by the potentiostat (a device which performs these operations), which measures the current needed to keep the potential constant (Figure 2.2).

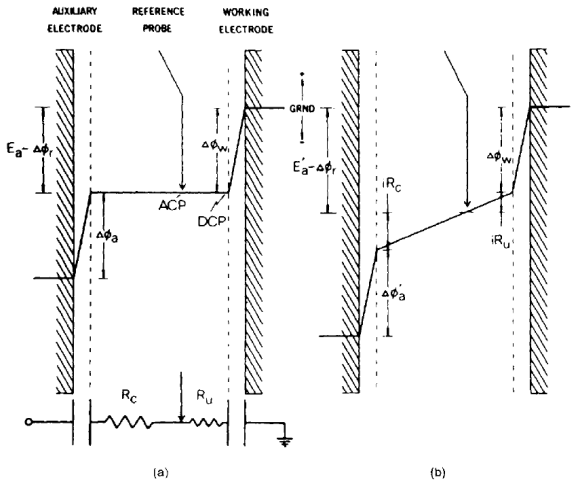


Figure 2.1: Schematic representation of potential gradients in a three-electrode cell: (a) $i = 0$; (b) $i \neq 0$. This is Figure 6.5 from [11].

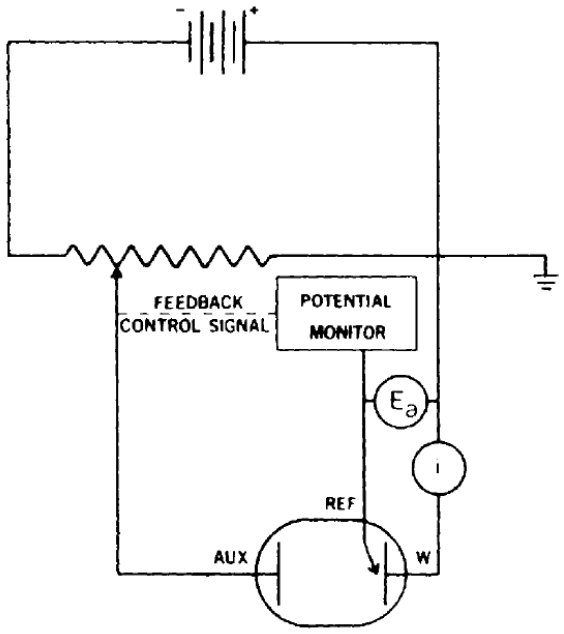
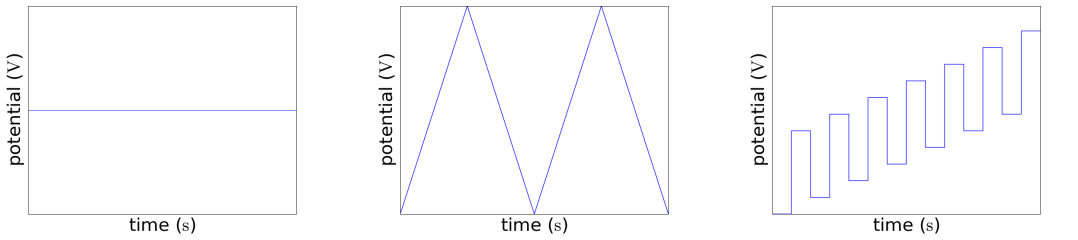


Figure 2.2: Schematic representation of a primitive three-electrode controlled-potential apparatus. This is Figure 6.4 from [11].



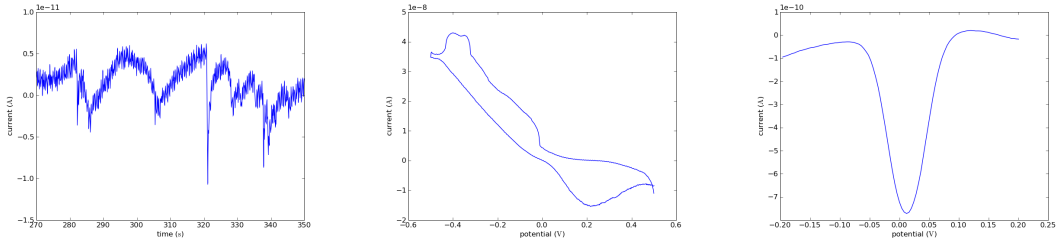
(a) Amperometry (b) Cyclic Voltammetry (c) Differential Pulse Voltammetry

Figure 2.3: Comparison of inputs of electrochemistry techniques

One technique is amperometry, where a constant potential is applied across the electrodes to reduce or oxidize a specific analyte (Figure 2.3(a)). When electroactive chemicals approach the electrode surfaces due to injection or diffusion, the output current changes until those chemicals are fully reduced or oxidized (Figure 2.4(a)). This technique is able to detect low concentrations and has a fast response time. However, its selectivity is low since all analyte that reduce under the potential applied will also reduce, in addition to the target. Furthermore, the electrodes can foul over time, reducing response, and must be cleaned to reproduce results. Electrode surfaces can be made resistive to fouling by applying substances like nafion to the surface. Fouling happens with the other techniques listed here, too, but those allow for discrete experiments to be run over a short time, between which cleaning can be performed.

Another common technique is cyclic voltammetry (CV). Here the potential is swept (at a rate called the scan rate) over a certain range linearly with time, where the range is again dependent on the reduction-oxidation (redox) potential of the analyte (Figure 2.3(b)). Since both reduction and oxidation potentials are within the range, both occur, and there multiple current spikes in opposite directions (Figure 2.4(b)). CV is useful for determining all of the analytes in a solution since each can present their own current spikes. When the scan rate is high, CV is able to detect analytes that have a short lifetime, like neurotransmitters. High scan rates, however, produce a large background current and thus have a poor limit of detection.

A third technique is differential pulse voltammetry (DPV) [12], a technique similar to CV. In CV, the potential across the electrodes is varied linearly with time up and down, in cycles. In DPV, the potential across the electrodes is varied in a step pattern. The start of each subsequent step is higher than the previous (Figure 2.3(c)). The current is measured before and after each step, and the difference is returned (Figure 2.4(c)). This allows the charging current to be removed from the output, yielding a more accurate result than CV.



(a) Amperometry (b) Cyclic Voltammetry (c) Differential Pulse Voltammetry

Figure 2.4: Comparison of results of electrochemistry techniques. All above results were obtained by measurements on our chip.

2.2 Existing Approaches

Previous work in this field has can be divided into three fields. First, the fabrication of smaller electrodes, which are able to detect lower concentrations and can be used in integrated sensors. In 1990, Shibuki [13] made a 250μm electrode to detect NO, and could detect concentration changes of 1μM, and reported detecting concentrations from 8–58nM. In 1996, Bedioui et al. [14] made a 50μm Pt wire with a detection limit of 170μM of nitrite (which is not low enough for biological samples). In 2002, Zhang et al. [15] made a 2μm electrode on a sensor chip, able to detect down to 0.3nM of NO. In 2002, Zhang et al. [16] made 100nm electrodes with a 2nM detection limit. In 2003, Kim et al. [17] made integrated 300μm gold electrodes able to detect down to 570nM of NO. In 2006, Wang et al. [18] made a 20μm electrode able to detect down to 20nM of NO.

The second field is integrated potentiostats, which allows on-chip signal processing and applications to sensor arrays, where each electrode in the array has its own integrated potentiostat. In 2003, Frey et al. [19] designed and fabricated an integrated potentiostat for use in biosensor chips with output current in the 10s of nA. In 2005, Murari et al. [6] designed and fabricated a 16-channel integrated potentiostat with an input current range down to 8pA. In 2007, Stanacevic et al. [7] designed a 16-channel integrated potentiostat array with an input current range down to 100fA. In 2007, Steffan and Vrba [20] designed an integrated potentiostat for low currents with an off-chip sensor, noting that an on-chip solution would yield better results.

The third field is electrode arrays, which allow chemical gradients to be detected. In 2001, George et al. [21] made an electrode array by screen printing, with sensors about 1 mm apart. In 2003, Naware et al. [22] made a 4×4 array, with square electrodes at $200 \times 200\mu\text{m}^2$. In 2005, Hafez et al. [23] made a 2×2 array with electrodes about 3 μm. In 2005, Zhang et al. [8] made a 3×3 of nine 90μm circular electrodes, however they were all shorted together and read by on on-chip

potentiostat. In 2006, Hassibi and Lee [24] made an integrated sensor array, with square electrodes at $50 \times 50 \mu m^2$.

2.3 Summary

The best recent work, as described above, shows electrodes in the nanometer size, concentration detection in the nanomolar range, fully integrated potentiostats able to detect femtoamps (though at slow timescales), and integrated sensor arrays.

The above approaches show progressing work in the field of microelectrodes and integrated sensors and potentiostats. Smaller devices are being fabricated with lower detection limits. Arrays of sensors and on-chip potentiostats have been designed and shown to work. However, no thorough research has been made into shape, size, distance and configuration of integrated electrodes. The electrodes fabricated above were all either square, round, or of an unknown shape, as they were flamed or shaved down to a specific size.

Chapter 3

Chip Design and Characterization Results

As discussed above, current approaches are either using relatively large electrodes, or have only used microelectrodes that were fabricated without specific regard for shape, size, distance and configuration. Thus, there is a need to identify the relationships among those characteristics and their impact on signal strength. This chapter describes the design goals of the chip, formulates hypothesis about its design, tests them, and draws some conclusions about the importance of shape, size, distance, and configuration. Tests are done with norepinephrine (NE) instead of NO because NO has a short lifespan, and NE performs similarly.

Previous work in this field is insufficient for our needs. We expect signals in the picoamp range on the microsecond timescale, and thus require circuitry able to amplify and measure quickly [25]. Other work has not addressed timing requirements [8, 20], assumes slow concentration change times on the order of seconds [6], or uses oversampling (which is slow) to detect low concentrations [6, 7].

Due to the above limitations, a chip was designed that was able to meet the requirements, and test hypotheses, for integrated biosensing [9]. The process used to achieve these goals was CMOS, the standard process used in microchip fabrication. Our process was able to produce feature definition at the 0.6 μm size. The electrodes are platinum on a glass substrate. There are 21 sensor sites, arranged in a 5×5 grid, with varying configurations and distances between electrodes, making each site unique (Figure 3.1). Sensor sites are 500μm (center-to-center) apart. The size of the chip is 9mm $\times$ 9mm, with bonding pads on the exterior at 160mm $\times$ 160mm each (Figure 3.2).

After the chip was fabricated, characterizations were done on each sensor site to determine the

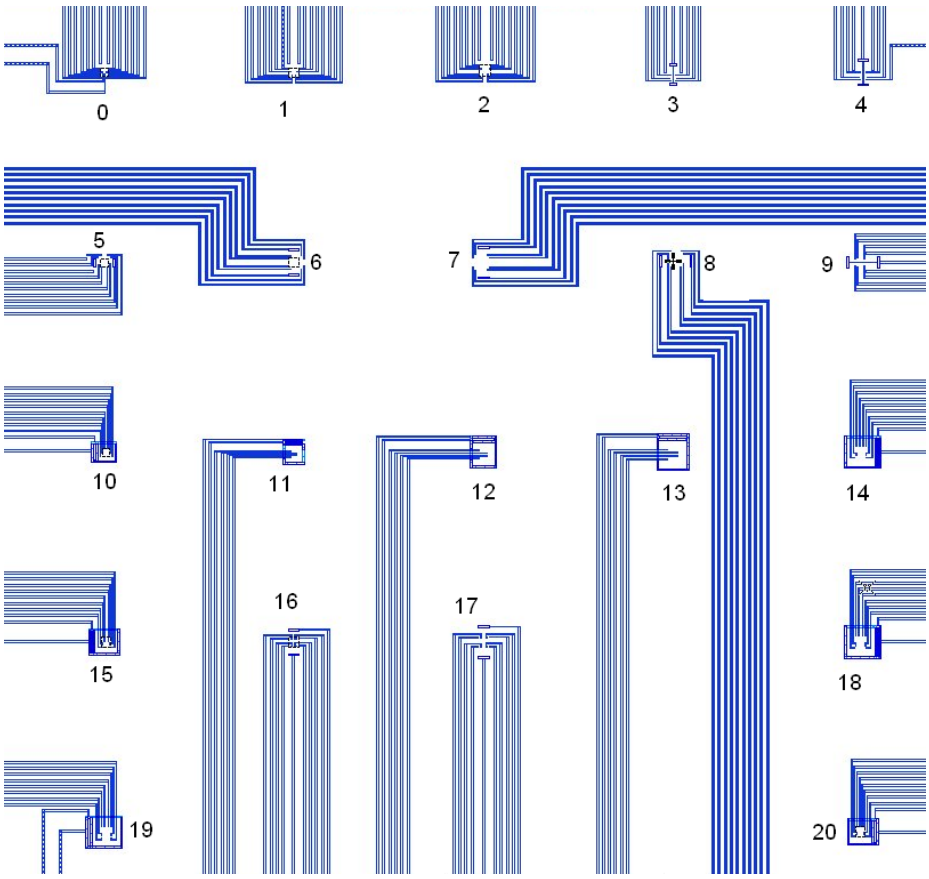


Figure 3.1: Sensors layout of the chip with numbered sensors

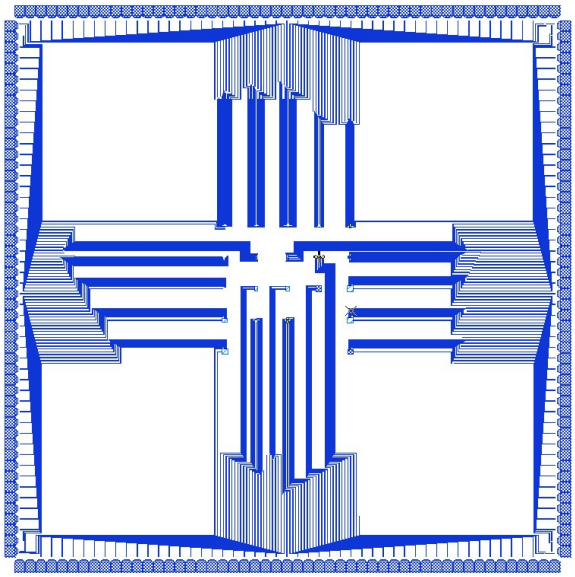


Figure 3.2: Full layout of the chip

best shape, size, and organization of the electrodes. There are 21 sensor sites on the chip. Each sensor site has four working-electrode areas. Each area has either one or two working electrodes, depending on if it is a 3- or 4-electrode system, respectively. This chapter describes the chip, sensor sites, and electrode systems, and the experiments used to characterize them.

3.1 Sensor Configurations

There are 21 sensor sites on the chip. Each site uses one of four sensor-site configurations, and one of four working-electrode configurations.

3.1.1 Working Electrodes

Each site has 4 working-electrode areas. There are four configurations, which are either 3- or 4-electrode systems. Three electrode systems have one auxiliary electrode (AE), one reference electrode (RE), and one working electrode (WE). Four electrode systems have one AE, one RE, and two WEs. These two WEs can work in tandem (one as a generator, one a collector, with appropriate instruments), and are interdigitated [19]. The interdigitated design is done to increase interaction between the two electrodes. No tests here used such an instrument, but work on 4-electrode systems indicates larger output in those systems [26]. To repeat, all 4-electrode systems were tested as if they were 3-electrode systems. The working-electrode configurations are (Table 3.1):

3 electrode where there is one WE, one RE, and one AE. WEs are numbered 1, 2, 3, 4, have areas (μm^2) 1, 2.25, 4, 4, and perimeters (μm) 4, 6, 8, 8, respectively.

4 electrode, C shape where there is one RE, one AE, and two WEs in the "C" shape (Figure 3.3). WEs are numbered 11, 12 for the first pair, 21, 22 for the second pair, 31 32 for the third pair, 41, 42 for the fourth pair. The first in each pair has area 10.885 μm^2 and perimeter 24.700 μm. The second in each pair has area 11.342 μm^2 and perimeter 25.081 μm.

4 electrode, inverse C shape where there is one RE, one AE, and two WEs in the "inverse C" shape (Figure 3.4). WEs are numbered 11, 12 for the first pair, 21, 22 for the second pair, 31 32 for the third pair, 41, 42 for the fourth pair. The first in each pair has area 14.267 μm^2 and perimeter 30.793 μm. The second in each pair has area 15.119 μm^2 and perimeter 32.835 μm.

4 electrode, F shape where there is one RE, one AE, and two WEs in the "F" shape (Figure 3.5). WEs are numbered 11, 12 for the first pair, 21, 22 for the second pair, 31 32 for the third pair,

Figure 3.3: “C”-type working-electrode pair

Figure 3.4: “Inverse C”-type working-electrode pair

41, 42 for the fourth pair. The first in each pair has area 21.771 μm^2 and perimeter 46.636 μm. The second in each pair has area 21.738 μm^2 and perimeter 46.796 μm.

3.1.2 Sites

The sensor-site configurations are:

2 auxiliary where there are four electrodes of the same size and spacing in a square with the WE at the bottom-left, RE at the top-right, and AEs at the top-left and bottom-right. The AEs are shorted together. This configuration is always in a 3-electrode system. This configuration is used as a control, as it is standard and simple.

common reference, top & bottom where there are two large, common RE at the top and bottom of the sensor site. The WE and AE pairs have the same size (by pair), are square, and are placed between the common REs. The common REs are shorted together. This configuration is always in a 3-electrode system. In 3-electrode systems, there is no current into the RE. Thus, this configuration should perform equally to the above, 2-auxiliary configuration.

common reference top, common auxiliary bottom where there is a large, common RE at the top, and a large, common AE at the bottom. WE pairs are placed between. This configuration is always in a 4-electrode system. Since current does flow into the AE, this configuration should perform better than the above 2 configurations, since AE area is less of a limiting factor.

Figure 3.5: “F”-type working-electrode pair

electrode system	**areas** (μm^2)	**perimeters** (μm)
3	1, 2.25, 4, 4	4, 6, 8, 8
4, "C"	10.885, 11.342	24.700, 25.081
4, "inverse C"	14.267, 15.119	30.793, 32.835
4, "F"	21.771, 21.738	46.636, 46.796

Table 3.1: Working-electrode areas and perimeters. The 3-electrode entry lists the different sizes that are in one sensor site. The 4-electrode entries list the areas of the first and second electrode in a pair, respectively.

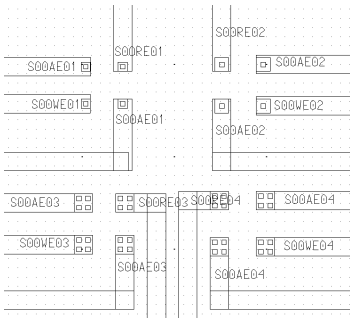


Figure 3.6: Sensor 0

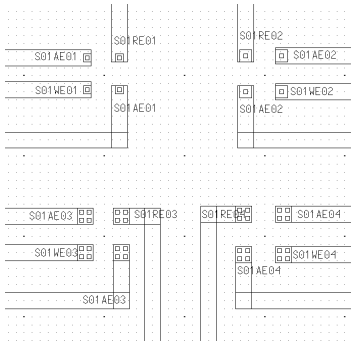


Figure 3.7: Sensor 1

common reference top, common auxiliary on 3 sides where there is a large, common RE at the top, and a larger, common AE completely surrounding the RE, and surrounding the WEs on the top, bottom, and right sides. This configuration is used by both 3- and 4-electrode systems. Since the AE area here is much larger and closer to all WEs, this configuration should perform even betten than the previous. Since no current flows into the RE, it is believed that moving the AE closer to the WEs is a good design. Hence, the RE is surrounded by the AE, and the WEs are all closest to the AE.

3.1.3 Sensors

Of the sixteen possible sensor combinations, nine are used (Table 3.2):

Sensor 0 is a 3-electrode system, with 2 AEs (Figure 3.6). WE areas have electrodes spaced at 3 μm for areas 1, 2, 4, and 2.5 μm for area 3. Pitch (center-to-center distance between WE areas) is 15μm.

Sensor 1 is the same as Sensor 0 (Figure 3.7), but with a pitch of 20μm.

Sensor 2 is the same as Sensor 0 (Figure 3.8), but with a pitch of 25μm..

Sensor 3 is a 3-electrode system, common RE at top and bottom (Figure 3.9). WE areas have electrodes spaced at 3 μm for areas 1, 2, 4, and 2.5 μm for area 3. Pitch is 15μm.

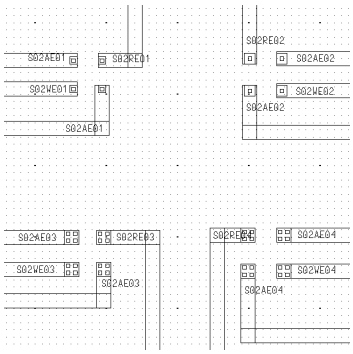


Figure 3.8: Sensor 2

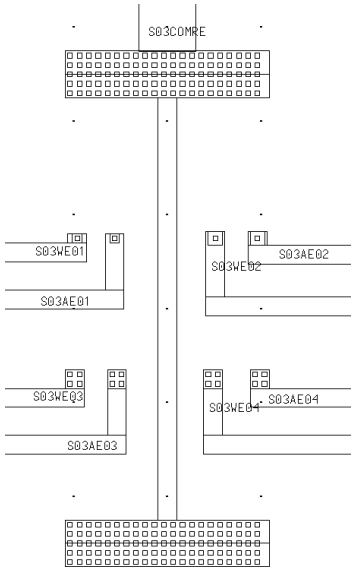


Figure 3.9: Sensor 3

Sensor 4 is the same as Sensor 3 (Figure 3.10), but with a pitch of 20μm.

Sensor 5 is a 4-electrode system, shape C, with a common RE at top, common AE at bottom (Figure 3.11). WEs 11, 12, 21, 22 are 26.55 μm from the AE. WEs 31, 32, 41, 42 are 11.51 μm from the AE. Pitch is 15μm.

Sensor 6 is a 4-electrode system, shape inverse C, with a common RE at top, common AE at bottom (Figure 3.12). WEs 11, 12, 21, 22 are 35.54 μm from the AE. WEs 31, 32, 41, 42 are 15.51 μm from the AE. Pitch is 20μm.

Sensor 7 is a 4-electrode system, shape inverse C, with a common RE at top, common AE at bottom (Figure 3.13). WEs 11, 12, 21, 22 are 45.48 μm from the AE. WEs 31, 32, 41, 42 are 20.40 μm from the AE. Pitch is 25μm.

Sensor 8 is a 4-electrode system, shape F, with a common RE at top, common AE at bottom (Figure 3.14). WEs 11, 12, 21, 22 are 33.53 μm from the AE. WEs 31, 32, 41, 42 are 13.53 μm from the AE. Pitch is 20μm.

Sensor 9 is the same as Sensor 3 (Figure 3.15), but with a pitch of 25μm.

Sensor 10 is a 4-electrode system, shape inverse C, with a common RE at top, common AE on 3 sides (Figure 3.16). WEs are 11.50 μm from the AE. Pitch is 15μm.

Sensor 11 is a 3-electrode system, with a common RE at top, common AE on 3 sides (Figure 3.17). WEs 1, 2, 3, 4 are 14.47, 14.24, 13.96, 13.96 μm from the AE, respectively. Pitch is 15μm.

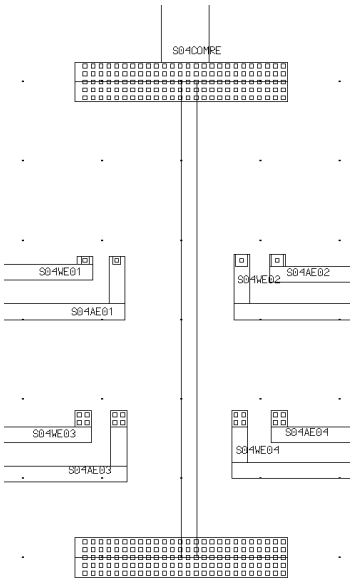


Figure 3.10: Sensor 4

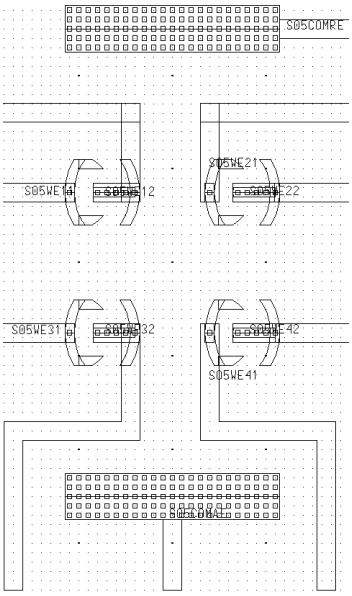


Figure 3.11: Sensor 5

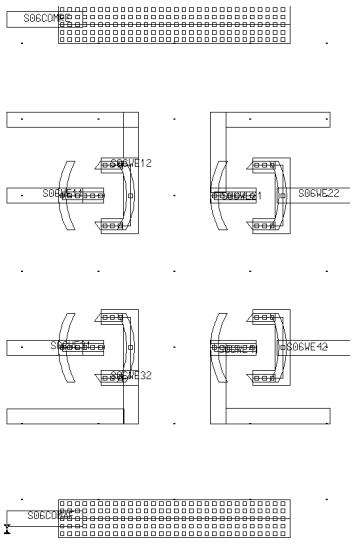


Figure 3.12: Sensor 6

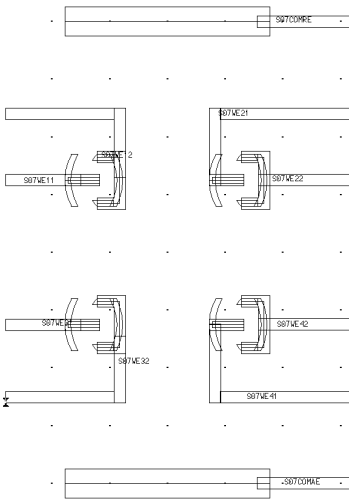


Figure 3.13: Sensor 7

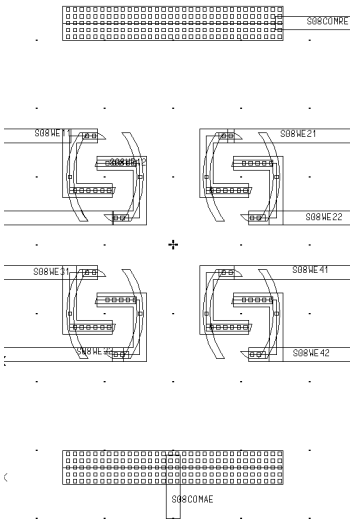


Figure 3.14: Sensor 8

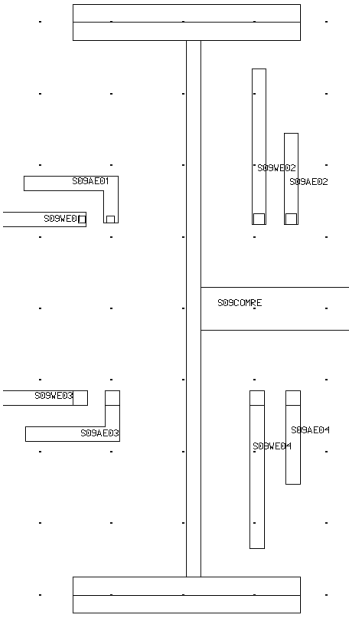


Figure 3.15: Sensor 9

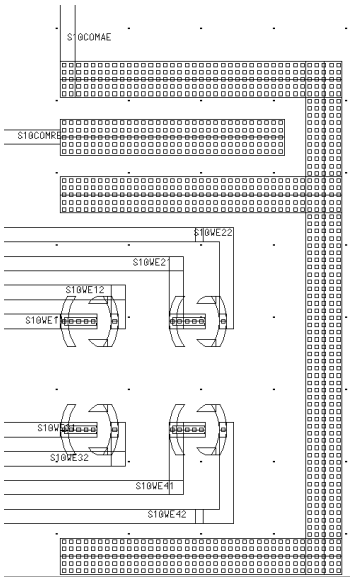


Figure 3.16: Sensor 10

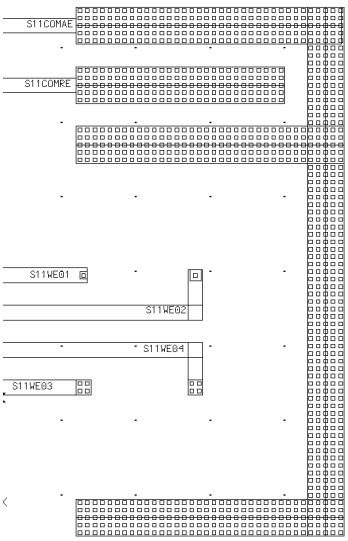


Figure 3.17: Sensor 11

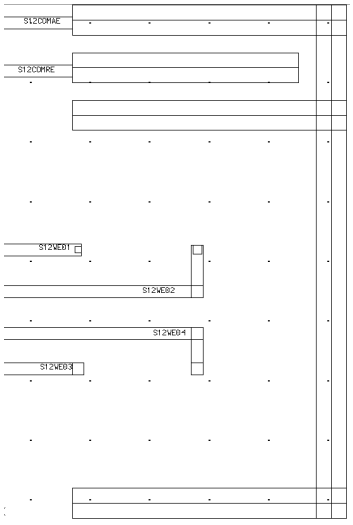


Figure 3.18: Sensor 12

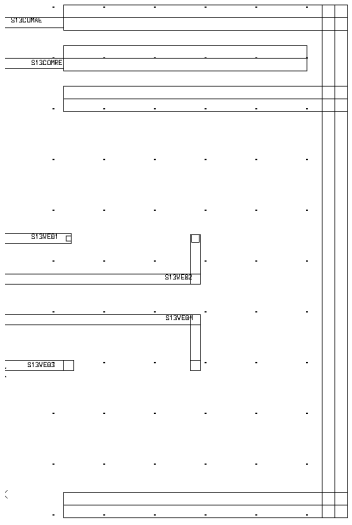


Figure 3.19: Sensor 13

Sensor 12 is a 3-electrode system, with a common RE at top, common AE on 3 sides (Figure 3.18). WEs 1, 2, 3, 4 are 19.5, 19.21, 18.98, 18.98 μm from the AE, respectively. Pitch is 20μm.

Sensor 13 is a 3-electrode system, with a common RE at top, common AE on 3 sides (Figure 3.19). WEs 1, 2, 3, 4 are 24.53, 24.26, 24.05, 24.05 μm from the AE, respectively. Pitch is 25μm.

Sensor 14 is a 4-electrode system, shape C, with a common RE at top, common AE on 3 sides (Figure 3.20). WEs are 20.50 μm from the AE. Pitch is 25μm.

Sensor 15 is a 4-electrode system, shape C, with a common RE at top, common AE on 3 sides (Figure 3.21). WEs are 15.50 μm from the AE. Pitch is 20μm.

Sensor 16 is a 4-electrode system, shape F, with a common RE at top, common AE on 3 sides (Figure 3.22). WEs are 13.50 μm from the AE. Pitch is 20μm.

Sensor 17 is a 4-electrode system, shape inverse C, with a common RE at top, common AE at bottom (Figure 3.23). WEs 11, 12, 21, 22 are 43.47 μm from the AE. WEs 31, 32, 41, 42 are 18.46 μm from the AE. Pitch is 25μm.

Sensor 18 is a 4-electrode system, shape F, with a common RE at top, common AE on 3 sides (Figure 3.24). WEs are 18.50 μm from the AE. Pitch is 25μm.

Sensor 19 is the same as Sensor 3 (Figure 3.25), but with a pitch of 15μm.

Sensor 20 is the same as Sensor 0, except the first WE area has spacing 2 μm (Figure 3.26) and with a pitch of 15μm.

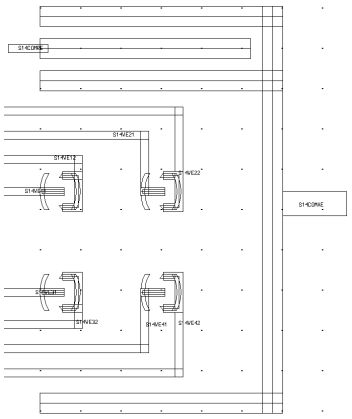


Figure 3.20: Sensor 14

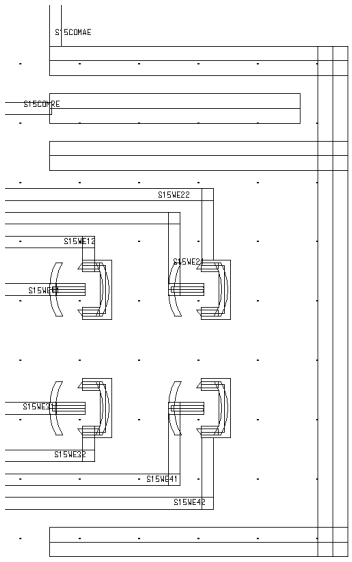


Figure 3.21: Sensor 15

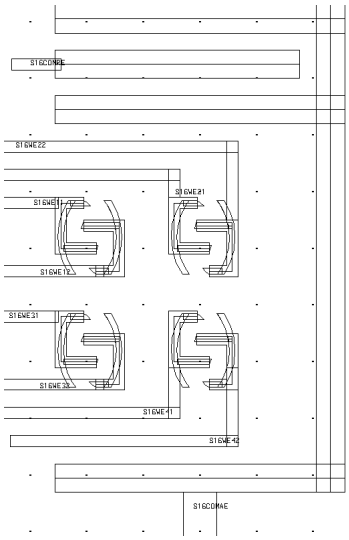


Figure 3.22: Sensor 16

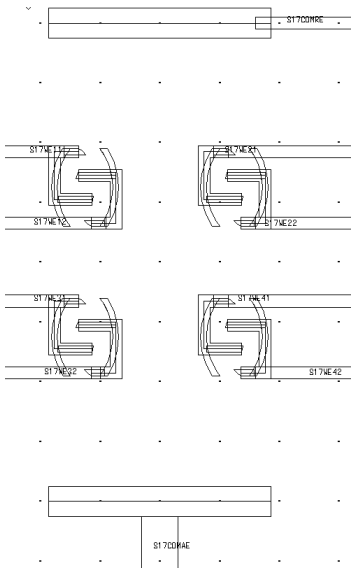


Figure 3.23: Sensor 17

electrode system:	**3**	**4, C**	**4, inverse C**	**4, F**
2 auxiliary	0, 1, 2, 20			
common reference, top & bottom	3, 4, 9, 19			
common reference top, common auxiliary bottom		5	6, 7	8, 17
common reference top, common auxiliary on 3 sides	11, 12, 13	14, 15	10	16, 18

Table 3.2: Sensor site configurations

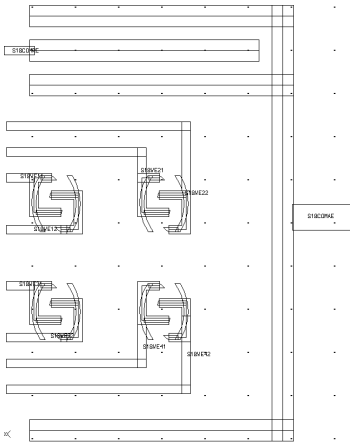


Figure 3.24: Sensor 18

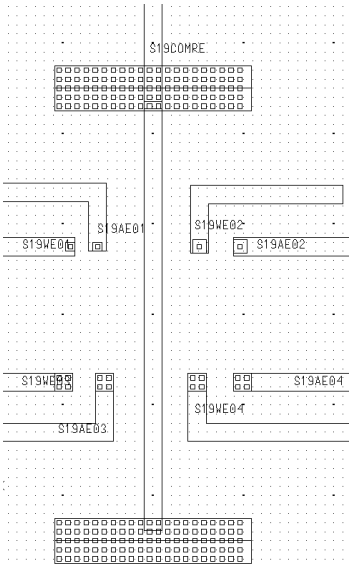


Figure 3.25: Sensor 19

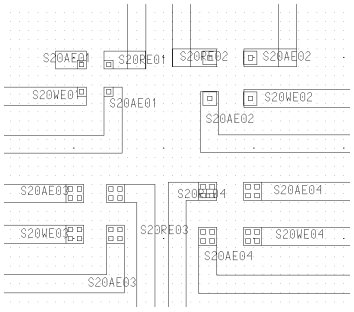


Figure 3.26: Sensor 20

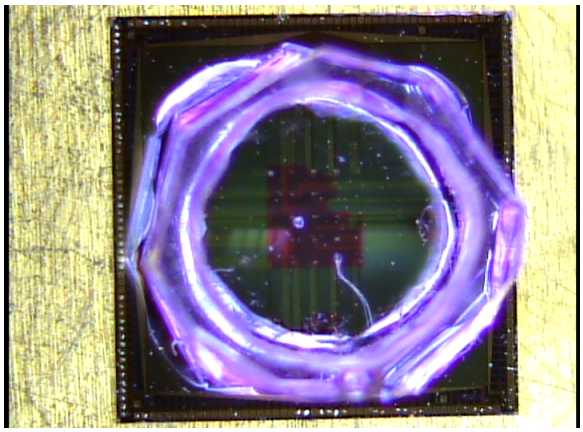


Figure 3.27: Chip with PDMS well attached

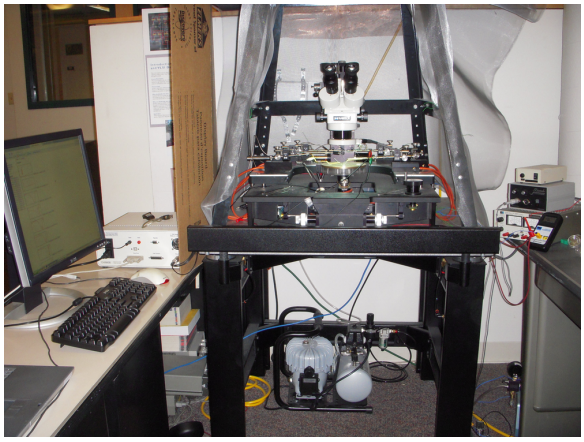


Figure 3.28: Probestation enclosed in a Faraday cage, and connected to the potentiostat

These data, including areas and perimeters of AEs, are summarized in Table 3.3, 3.4, 3.5.

3.2 Experiment Setup

A CH Instruments 660B potentiostat was used to perform all tests. A PDMS well was attached to the chip to prevent solutions from leaking onto probe pads (Figure 3.27). The chip was mounted onto a probestation, which was shielded by enclosing it in a Faraday cage (Figure 3.28). Pins were lowered onto the probe-pads of the chip (Figure 3.29, 3.30), which were connected to the reference, auxiliary, and working electrode leads of the potentiostat (Figure 3.31).

sensor	WE	area	perimeter	WE distance to AE	AE area	AE perimeter
0	1	1	4	3	1	4
0	2	2.25	6	3	2.25	6
0	3	4	8	2.5	4	8
0	4	4	8	3	4	8
1	1	1	4	3	1	4
1	2	2.25	6	3	2.25	6
1	3	4	8	2.5	4	8
1	4	4	8	3	4	8
2	1	1	4	3	1	4
2	2	2.25	6	3	2.25	6
2	3	4	8	2.5	4	8
2	4	4	8	3	4	8
3	1	1	4	3	1	4
3	2	2.25	6	3	2.25	6
3	3	4	8	2.5	4	8
3	4	4	8	3	4	8
4	1	1	4	3	1	4
4	2	2.25	6	3	2.25	6
4	3	4	8	2.5	4	8
4	4	4	8	3	4	8
5	11	10.885	24.7	26.55	115	56
5	12	11.342	25.081	26.55	115	56
5	21	10.885	24.7	26.55	115	56
5	22	11.342	25.081	26.55	115	56
5	31	10.885	24.7	11.51	115	56
5	32	11.342	25.081	11.51	115	56
5	41	10.885	24.7	11.51	115	56
5	42	11.342	25.081	11.51	115	56
6	11	14.267	30.793	35.54	152.5	71
6	12	15.119	32.835	35.54	152.5	71
6	21	14.267	30.793	35.54	152.5	71
6	22	15.119	32.835	35.54	152.5	71
6	31	14.267	30.793	15.51	152.5	71
6	32	15.119	32.835	15.51	152.5	71
6	41	14.267	30.793	15.51	152.5	71
6	42	15.119	32.835	15.51	152.5	71
7	11	14.267	30.793	45.48	177.5	81
7	12	15.119	32.835	45.48	177.5	81
7	21	14.267	30.793	45.48	177.5	81
7	22	15.119	32.835	45.48	177.5	81
7	31	14.267	30.793	20.4	177.5	81
7	32	15.119	32.835	20.4	177.5	81
7	41	14.267	30.793	20.4	177.5	81
7	42	15.119	32.835	20.4	177.5	81

Table 3.3: Electrode properties for sensors 0–7. Areas are in μm^2. Perimeters and distances are in μm. WE distance to AE lists shortest distance between the two electrodes.

sensor	WE	WE area	WE perimeter	WE distance to AE	AE area	AE perimeter
8	11	21.771	46.636	33.53	161.25	74.5
8	12	21.738	46.796	33.53	161.25	74.5
8	21	21.771	46.636	33.53	161.25	74.5
8	22	21.738	46.796	33.53	161.25	74.5
8	31	21.771	46.636	13.53	161.25	74.5
8	32	21.738	46.796	13.53	161.25	74.5
8	41	21.771	46.636	13.53	161.25	74.5
8	42	21.738	46.796	13.53	161.25	74.5
9	1	1	4	3	1	4
9	2	2.25	6	3	2.25	6
9	3	4	8	2.5	4	8
9	4	4	8	3	4	8
10	11	14.267	30.793	11.5	865	356
10	12	15.119	32.835	11.5	865	356
10	21	14.267	30.793	11.5	865	356
10	22	15.119	32.835	11.5	865	356
10	31	14.267	30.793	11.5	865	356
10	32	15.119	32.835	11.5	865	356
10	41	14.267	30.793	11.5	865	356
10	42	15.119	32.835	11.5	865	356
11	1	1	4	14.47	820	338
11	2	2.25	6	14.24	820	338
11	3	4	8	13.96	820	338
11	4	4	8	13.96	820	338
12	1	1	4	19.5	1045	428
12	2	2.25	6	19.21	1045	428
12	3	4	8	18.98	1045	428
12	4	4	8	18.98	1045	428
13	1	1	4	24.53	1270	518
13	2	2.25	6	24.26	1270	518
13	3	4	8	24.05	1270	518
13	4	4	8	24.05	1270	518
14	11	10.885	24.7	20.5	1333.75	543.5
14	12	11.342	25.081	20.5	1333.75	543.5
14	21	10.885	24.7	20.5	1333.75	543.5
14	22	11.342	25.081	20.5	1333.75	543.5
14	31	10.885	24.7	20.5	1333.75	543.5
14	32	11.342	25.081	20.5	1333.75	543.5
14	41	10.885	24.7	20.5	1333.75	543.5
14	42	11.342	25.081	20.5	1333.75	543.5

Table 3.4: Electrode properties for sensors 8–14. Areas are in μm^2. Perimeters and distances are in μm. WE distance to AE lists shortest distance between the two electrodes.

sensor	WE	area	perimeter	WE distance to AE	AE area	AE perimeter
15	11	10.885	24.7	15.5	1108.75	453.5
15	12	11.342	25.081	15.5	1108.75	453.5
15	21	10.885	24.7	15.5	1108.75	453.5
15	22	11.342	25.081	15.5	1108.75	453.5
15	31	10.885	24.7	15.5	1108.75	453.5
15	32	11.342	25.081	15.5	1108.75	453.5
15	41	10.885	24.7	15.5	1108.75	453.5
15	42	11.342	25.081	15.5	1108.75	453.5
16	11	21.771	46.636	13.5	1121.875	458.75
16	12	21.738	46.796	13.5	1121.875	458.75
16	21	21.771	46.636	13.5	1121.875	458.75
16	22	21.738	46.796	13.5	1121.875	458.75
16	31	21.771	46.636	13.5	1121.875	458.75
16	32	21.738	46.796	13.5	1121.875	458.75
16	41	21.771	46.636	13.5	1121.875	458.75
16	42	21.738	46.796	13.5	1121.875	458.75
17	11	14.267	30.793	43.47	186.25	84.5
17	12	15.119	32.835	43.47	186.25	84.5
17	21	14.267	30.793	43.47	186.25	84.5
17	22	15.119	32.835	43.47	186.25	84.5
17	31	14.267	30.793	18.46	186.25	84.5
17	32	15.119	32.835	18.46	186.25	84.5
17	41	14.267	30.793	18.46	186.25	84.5
17	42	15.119	32.835	18.46	186.25	84.5
18	11	21.771	46.636	18.5	1346.875	548.75
18	12	21.738	46.796	18.5	1346.875	548.75
18	21	21.771	46.636	18.5	1346.875	548.75
18	22	21.738	46.796	18.5	1346.875	548.75
18	31	21.771	46.636	18.5	1346.875	548.75
18	32	21.738	46.796	18.5	1346.875	548.75
18	41	21.771	46.636	18.5	1346.875	548.75
18	42	21.738	46.796	18.5	1346.875	548.75
19	1	1	4	2	1	4
19	2	2.25	6	3	2.25	6
19	3	4	8	2.5	4	8
19	4	4	8	3	4	8
20	1	1	4	2	1	4
20	2	2.25	6	3	2.25	6
20	3	4	8	2.5	4	8
20	4	4	8	3	4	8

Table 3.5: Electrode properties for sensors 15–20. Areas are in μm^2. Perimeters and distances are in μm. WE distance to AE lists shortest distance between the two electrodes.

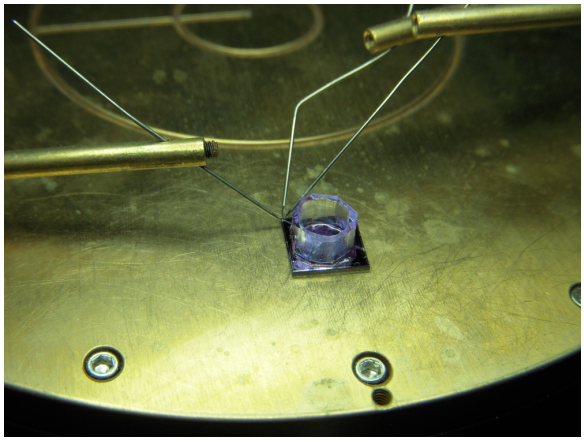


Figure 3.29: Pins lowered onto probe pads of the chip

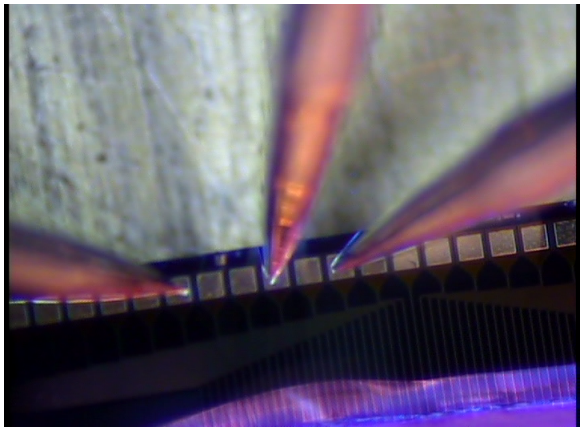


Figure 3.30: Pins lowered onto probe pads of the chip, top view

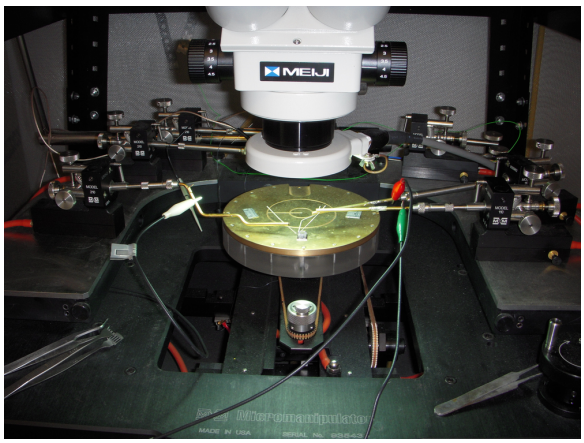


Figure 3.31: Potentiostat leads for reference, auxiliary, and working electrodes connected to chip pins (white, red, and green, respectively)

3.3 Experiment Procedure

The 21 sensor sites were characterized to find the best general electrode construction. This was done using DPV and varying the electrodes and solution concentrations.

3.3.1 All Sites

The first set of tests was done by performing DPV from -0.2V to 0.2V on two randomly-chosen WE areas of each sensor site, three times each (six tests on each sensor site). The solution used was 0.3mM NE in 0.1M H_2SO_4. Before a WE area was used, it was cleaned by CV from -0.3V to 1.5V at 1V/s with 20 cycles, with a solution of 0.1M H_2SO_4. The result of each individual test was taken to be the distance of the baseline to the bottom of the peak centered around 0.0V. The result of each sensor site was taken to be the average of the six tests on that site.

3.3.2 Limit of Detection

The second set of tests was done on sensors 17 and 8, which were the two highest-performing sensors from the previous test. The same procedure was used as above, except decreasing concentrations of NE were used.

3.3.3 Specific Electrodes

The third set of tests was done on specific WE areas (instead of randomly chosen areas), based on their performance and our hypotheses about their performance. The procedure used was the same as above, except specific WE areas were chosen, and multiple chips were tested to control against process variation while manufacturing the chips.

3.4 Experiment Results

Results for the first set of tests (all sites) are summarized in Table 3.6. Results for the third set of tests are summarized in Table 3.7. Below are our hypotheses about electrode configuration, and the results that correspond to those hypotheses.

3.4.1 Output vs. Working Electrode Area

This hypothesis is that the output current is proportional to the WE area. Figure 3.32 shows all 126 tests and a linear-fit curve. Although there is a large variation in this correlation, it is clear that there

Sensor	% of max	Avg. value
17	100.0 ± 4.0	7.04e-10 ± 2.83e-11
8	77.9 ± 5.5	5.48e-10 ± 3.85e-11
16	75.7 ± 1.8	5.32e-10 ± 1.28e-11
7	71.5 ± 3.0	5.03e-10 ± 2.13e-11
18	58.8 ± 2.5	4.14e-10 ± 1.73e-11
6	57.6 ± 4.0	4.05e-10 ± 2.79e-11
15	55.2 ± 3.5	3.89e-10 ± 2.43e-11
14	50.8 ± 2.1	3.57e-10 ± 1.46e-11
10	47.6 ± 3.6	3.35e-10 ± 2.50e-11
5	43.0 ± 1.2	3.03e-10 ± 8.27e-12
1	36.7 ± 3.5	2.58e-10 ± 2.46e-11
20	26.1 ± 1.0	1.84e-10 ± 7.29e-12
0	23.8 ± 5.0	1.68e-10 ± 3.55e-11
19	21.9 ± 2.0	1.54e-10 ± 1.44e-11
12	19.5 ± 1.5	1.37e-10 ± 1.05e-11
4	19.2 ± 1.8	1.35e-10 ± 1.28e-11
13	19.1 ± 1.0	1.34e-10 ± 7.37e-12
2	17.6 ± 2.1	1.24e-10 ± 1.49e-11
9	17.5 ± 0.4	1.23e-10 ± 2.83e-12
11	14.4 ± 1.9	1.01e-10 ± 1.33e-11
3	11.3 ± 1.3	7.93e-11 ± 9.48e-12

Table 3.6: DPV results for sensor sites sorted by decreasing value

sensor	electrode	chip 1	chip 3	chip 4	distance	area
02	01		1.493e-10 ± 7.3e-12	1.632e-10 ± 1.6e-12	3.00	1
02	02		1.921e-10 ± 4.1e-12	2.120e-10 ± 8.9e-12	3.00	2.25
02	03		1.940e-10 ± 1.1e-11	2.250e-10 ± 2.0e-12	2.50	4
02	04		1.419e-10 ± 8.5e-12	1.850e-10 ± 4.6e-12	3.00	4
07	12	5.169e-10	4.774e-10 ± 2.1e-11	4.994e-10 ± 1.2e-11	45.48	15.119
07	32	3.245e-10	4.963e-10 ± 1.1e-11	4.072e-10 ± 2.2e-12	20.40	15.119
16	12	5.157e-10	7.212e-10 ± 2.4e-11	6.188e-10 ± 1.3e-11	13.50	21.738
16	32	6.494e-10	6.349e-10 ± 5.2e-12	6.989e-10 ± 1.0e-11	13.50	21.738
17	11	6.562e-10			43.47	21.771
17	12	4.949e-10	5.533e-10 ± 8.5e-12	6.420e-10 ± 9.2e-12	43.47	21.738
17	31	6.913e-10			18.46	21.771
17	32	6.593e-10	6.360e-10 ± 5.9e-12	6.284e-10 ± 1.2e-11	18.46	21.738
18	12	5.570e-10	5.377e-10 ± 8.8e-12	6.130e-10 ± 8.5e-12	18.50	21.738
18	32	5.683e-10		4.965e-10 ± 2.8e-12	18.50	21.738

Table 3.7: Results of further DPV tests on specific working electrodes. Outputs are given from averaged data from runs on chips 1, 3, and 4 with standard error in amperes. The distance column lists the distance of the working to the auxiliary electrode from Table 3.3, 3.4, 3.5 in μm. The area column lists the area of the working electrode from Table 3.1 in μm^2. Chip 2 was not performing well and was skipped. Chips 3 and 4 used the same solution. Chip 1 used a separate solution. They were both mixed to be identical, but at concentrations this low, that is difficult to guarantee. Thus, do not use these data without adjusting for relative performance.

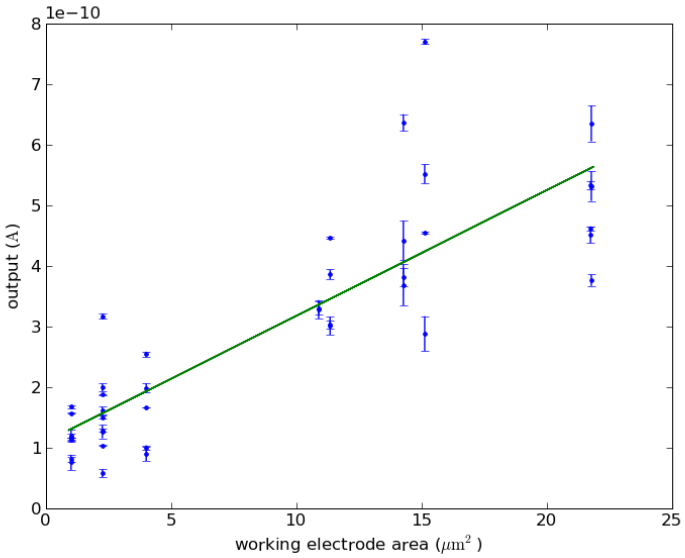


Figure 3.32: WE area vs. Output

is a general trend towards larger WE area leading to greater output, supporting this hypothesis. However, due to variations in sensor configurations, AE areas, perimeters, and distances, there is not much more we can say about this.

3.4.2 Output Density vs. Working Electrode Area

This hypothesis is that the output current density is proportional to the WE area. Figure 3.33 shows all 126 tests and a linear-fit curve. Here there appears to be a general trend. However, this is likely not the case. The areas above 10 microns are 4-electrode systems; the areas below 10 microns are 3-electrode systems. The two systems differ significantly, so it is reasonable to separate them for analysis. When viewed this way (Figure 3.34, 3.35), it is clear that there is no conclusive data to support this hypothesis.

3.4.3 Ratio of Working to Auxiliary Electrode Areas

This hypothesis is that the output current is proportional to the ratio of the WE area to the AE area. Figure 3.36 shows the tests. Here we can clearly see that when the ratio is unity (which occurs on only 3-electrode systems), the output appears to be limited by the AE area. When the AE is much larger than the WE (which includes both 3- and 4-electrode systems), the output appears to be limited by factors other than the AE area. Hence, we can conclude that this hypothesis holds, and say that the AE area should be much larger than the WE area. A hypothesis to test further is

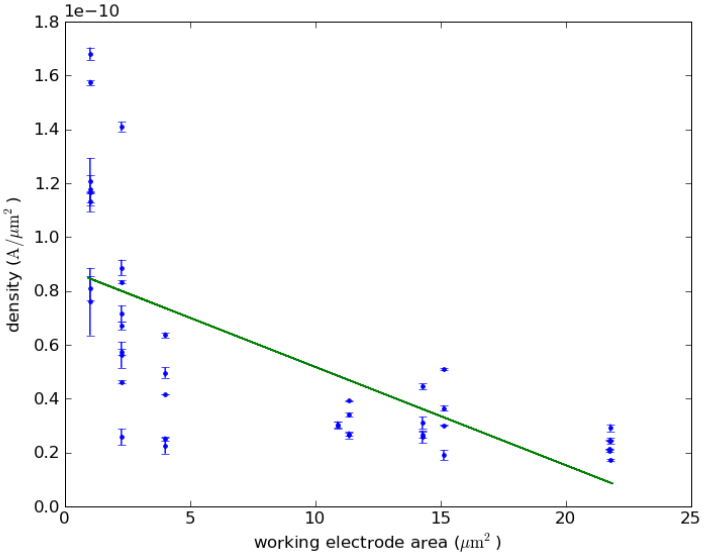


Figure 3.33: WE area vs. Density

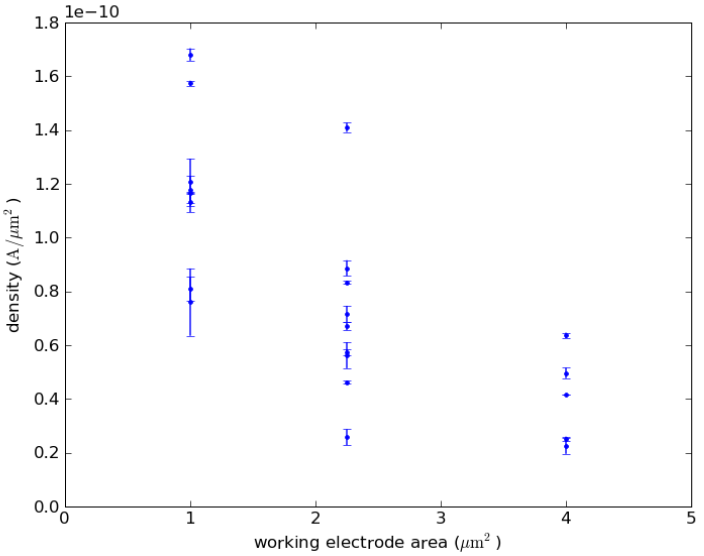


Figure 3.34: WEs with low area vs. Density

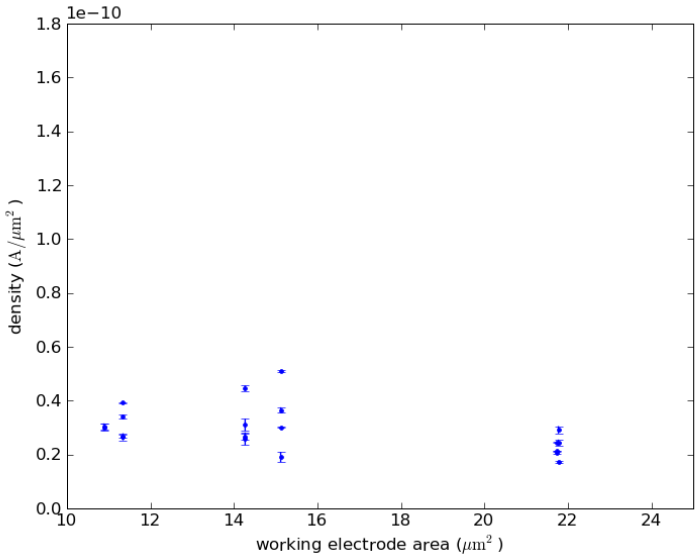


Figure 3.35: WEs with high area vs. Density

that there is some ratio above which the output is not limited by the AE, so that the graph would have a step at some point between our data.

3.4.4 Working Electrode Perimeter over Area vs. Output Density

This hypothesis is that the output current density is proportional to the WE perimeter divided by the WE area. Figure 3.37 shows the tests and a linear-fit curve. We can see a correlation, indicating that most of the electron transfer happens near the edge of the electrode.

3.4.5 Distance from Working to Auxiliary Electrode

This hypothesis is that the output current density is proportional to the distance from the WE to the AE. Figure 3.38 shows the tests. The vertical column below 5 microns contains all "2 auxiliary" sensor configurations, which are only 3-electrode systems. It is clear that the data are inconclusive, and that distance plays an unimportant part for most electrodes.

3.5 Limit of Detection Results

Sensor 17 (Table 3.8) was able to detect down to about 3μM with a linear fit of $A = 1.88996 * 10^{-6} * M - 2.36855 * 10^{-12}$, where A is the resulting current in amperes and M is the concentration molarity. Sensor 8 (Table 3.9) was able to detect down to about 10μM with a linear fit of $A =$

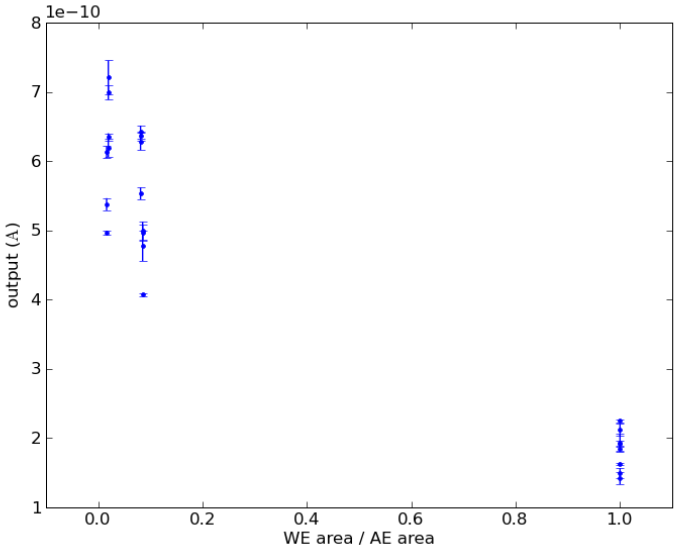


Figure 3.36: WE area / AE area vs. Output

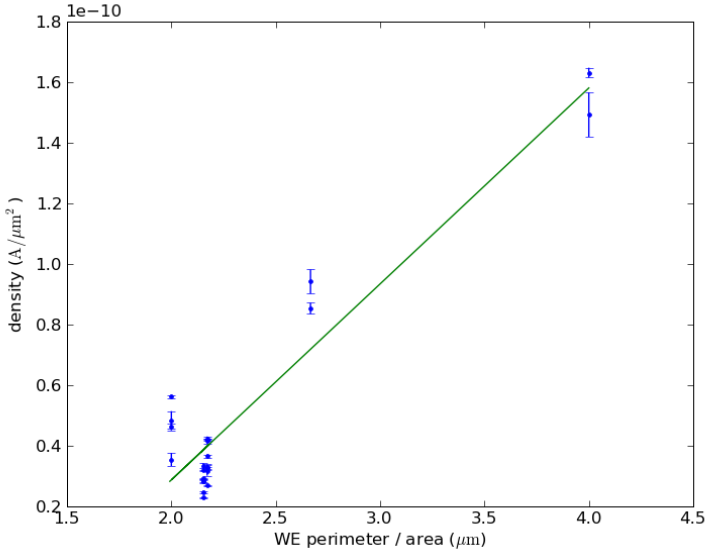


Figure 3.37: WE perimeter / area vs. Density

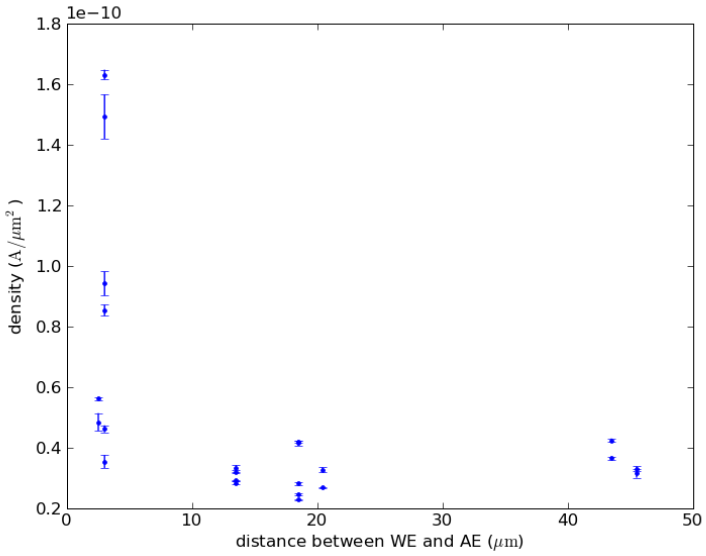


Figure 3.38: Distance from WE to AE vs. Density

Concentration (M)	Peak value
0.0003	5.609E-10
0.0003	5.591E-10
0.0001	1.988E-10
0.0001	2.082E-10
0.000033	5.701E-11
0.000033	4.788E-11
0.000011	1.306E-11
0.000011	1.439E-11

Table 3.8: Detection limit results for sensor 17 sorted by decreasing concentration

$8.29117 * 10^{-7} * M + 4.3417 * 10^{-11}$, where A is the resulting current in amperes and M is the concentration molarity. These are summarized in Figure 3.39.

Concentration (M)	Peak value
0.0003	2.926E-10
0.0003	2.663E-10
0.0001	1.786E-10
0.0001	1.753E-10
0.000033	3.276E-11
0.000033	3.296E-11

Table 3.9: Detection limit results for sensor 8 sorted by decreasing concentration

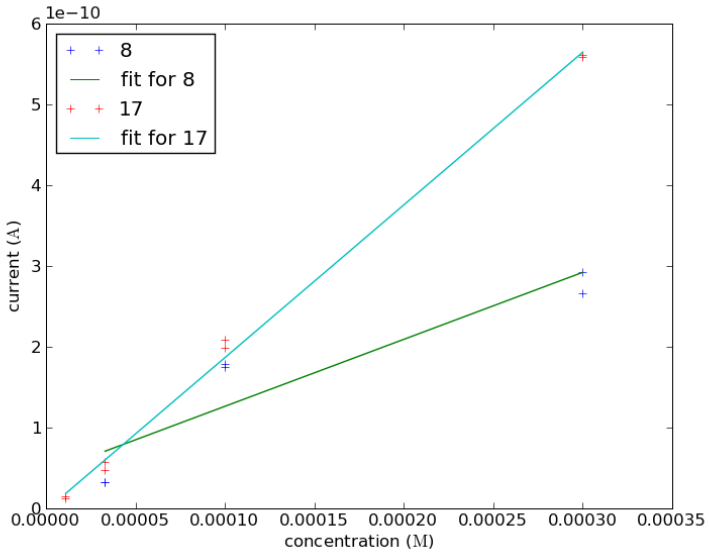


Figure 3.39: Detection limits

Chapter 4

Slice Testing

The long-term goal of this work is to create an integrated biosensor for use with living cells. Thus, proof-of-concept tests were performed. This chapter describes these tests and results.

4.1 Slice Information

Mouse-ovary slices were prepared and attached by the Department of Biomedical Sciences at Colorado State University. Slices were prepared as described in [27], but with changes to use ovary instead of brain slices. Those changes include: they were not cut on a specific plane and they were not plated on glass coverslips (rather, on the chips). The ovaries were sectioned at 200μm thick in adult, female mice.

4.2 Experiment Procedure

Since the slices had to be tested while alive, and thus at a temperature of 37 °C, we were forced to use a heating device. We chose a heat lamp, placed in the Faraday cage, next to the chip. Tests were performed using the method below (amperometry for 30 minutes), but without ovary, which showed that the lamp did not contribute noticable noise to the output.

To find the ideal potential at which to run the amperometry, a hydrodynamic voltammogram (HDV) was performed to find the optimum potential for amperometry. This was done by running multiple amperometry experiments at identical conditions except for potential, which was varied between 0.6V and 1.25V. In order to ensure identical conditions, the electrode was cleaned before each run by CV from 1.5V to -0.8V at 1V/s for 100 cycles in H_2SO_4. The chip well was filled with

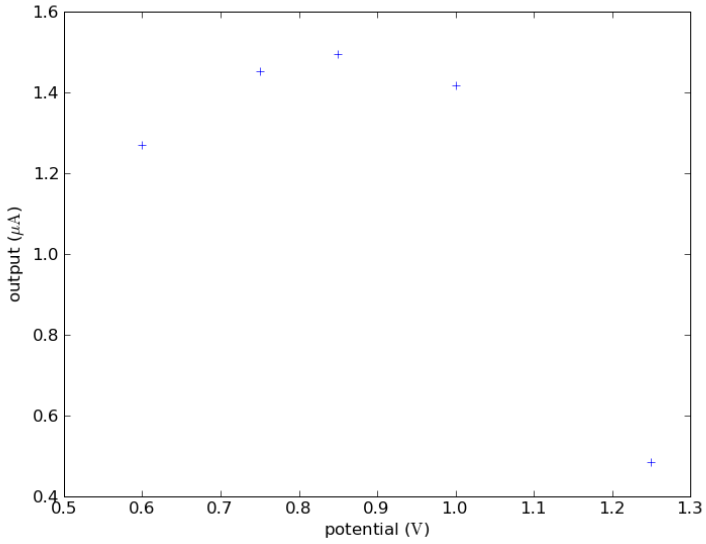


Figure 4.1: HDV for norepinephrine

50μL 0.1M KCl. A syringe of 600μM NE was loaded into a pump. The end of the syringe was connected to a pipette, the end of which was lowered into the well so that it was below the surface of the solution. An amperometry experiment was carried out where the syringe pump would be enabled for 10 seconds at 100μL/min. The value of the experiment was taken to be the difference between the idle point just before the syringe pump was enabled (generally close to 0) and the lowest point of the resulting curve. The optimum potential was found by taking the largest result, which was at 0.85V (Figure 4.1).

These tests were done by preparing a chip with the PDMS well, doing a basic CV to determine if the chip was working (although not how well it worked), and attaching a living slice of a mouse ovary to the surface of the chip, over the sensor sites. Care was taken to ensure the slice would stay living up to and during the tests by keeping it at a temperature of 37 °C, and submerged in neurobasal. After the ovary slice was attached the chip was moved into a different lab. The temperature was measured by lowering the end of a thermometer below the surface of the neurobasal, and heated with a lamp to the appropriate temperature (Figure 4.2). The entire testing device was enclosed in a Faraday cage. Previously, a chip had been prepared in the same fashion, but with no ovary, and tested with the lamp on. This test showed that the lamp did not contribute noise to the results. After these preparations, the chip was connected to the 660B potentiostat at sensor 17, working electrode 41. Amperometry was performed at 0.85V for 30 minutes.

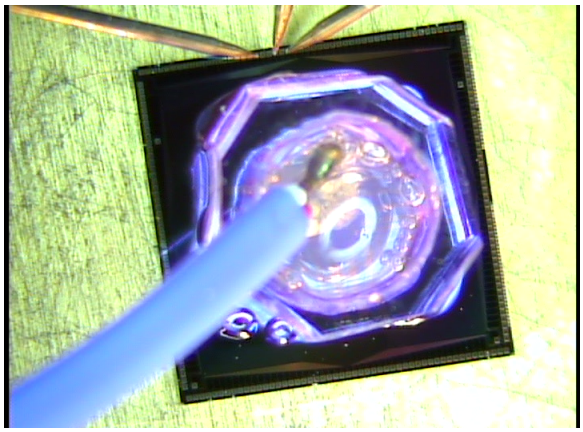


Figure 4.2: Top view of chip under testing with an ovary slice attached. The blue wire is a temperature sensor.

4.3 Experiment Results

Figure 4.3 shows an excerpt of a test. Thes results are consistent with the release profile: it exhibits multiple sharp changes, followed by an exponential decrease back to the baseline.

4.4 Discussion

Figure 4.4 shows a similar test from another group. Our pulses have a longer time duration (5s compared to 50ms) and a smaller peak height (10pA compared to 100pA). However, our electrodes are much smaller, which accounts for some of these differences. It appears that our chip is able to detect chemical releases from a living cell. Hence, with the right instruments, future work could pursue obtaining a chemical gradient. This would require many electrodes being used at once, either with many potentiostats or a multi-channel potentiostat. Further work should also characterize the performance of the sensors with attached ovary slices. Currently we do not have any reliable data to show the concentrations of chemicals that are released from the cell. However, our results show proof-of-concept results.

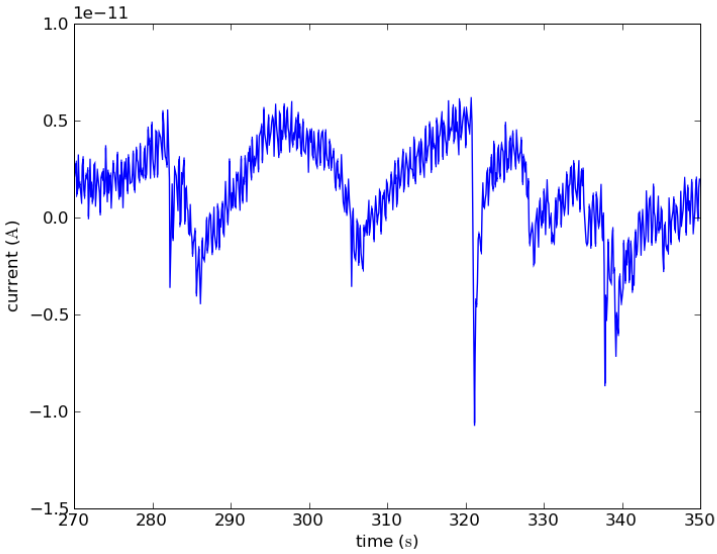


Figure 4.3: Amperometric pulses from ovary-slice test.

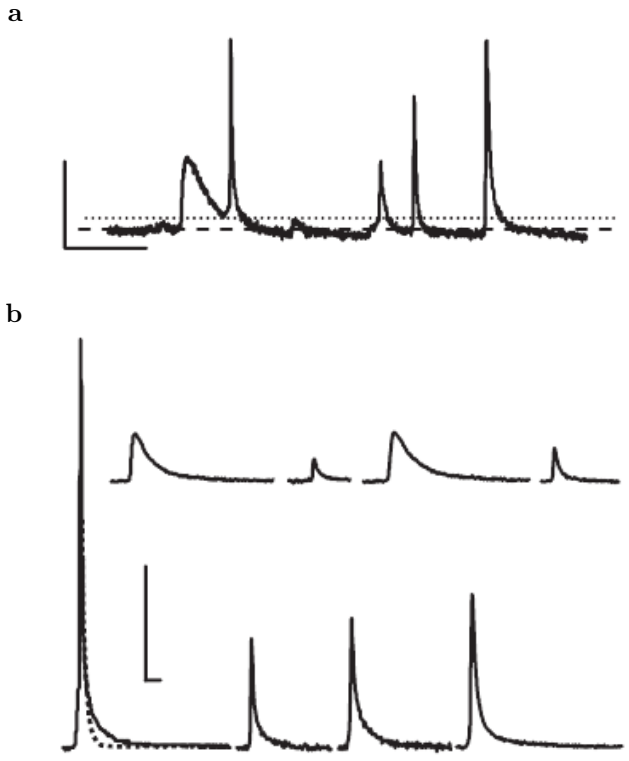


Figure 4.4: Amperometric pulses from another group. (a) Figure 2 (a) of [25], which used a carbon fiber electrode placed against the surface of a secretory cell. Scale bars, 50pA and 100ms for the timescale. (b) Figure 4 of [25]: amperometric events belonging to the same experimental trace. Scale bars, 50pA and 10ms for the timescale.

Chapter 5

Discussion and Conclusion

We fabricated a chip with an array of electrochemical sensor cells of varying make. This chip was characterized to find high-performing electrodes which were then each characterized by the response to various concentrations of expected chemicals that would be released by living cells. In addition, more characterizations were done to test the hypotheses put forward during initial chip design, and some conclusions were drawn from the results which led to design goals of future chips.

The results above show that the chip performs well under the anticipated conditions for integrated biosensing (i.e., low concentrations of solution). During the overall site test, we found sensors 17 and 8 to perform the best, and showed their limit of detection down to 3 and 10μM, respectively. The design hypotheses of the chip were tested, and some were found to be conclusive. Specifically, it is clear that 1) output increases with increased WE area, and 2) AE area should be much larger than WE area to prevent saturation. There is not conclusive evidence to support the hypotheses that distance from WE to AE, or the shape of the sensor site has any major impact on the result. Thus, further chips designed should have large WEs, and larger AEs.

Future work needs to be done in two main areas: sensitivity characterization during live, ovary slice amperometry, and an on-chip potentiostat design. Our slice testing was only qualitative, and we currently do not know what our data indicate about how much chemical was released. Furthermore, we need a many-channel, on-chip potentiostat capable of reading picoamps in microseconds. Current designs in this area have good sensitivity, but are slow.

Bibliography

[1] G. Bicker, "Stop and go with no: nitric oxide as a regulator of cell motility in simple brains," *Bioessays*, vol. 27, no. 5, 2005.

[2] S. Bulotta, C. Perrotta, A. Cerullo, C. De Palma, E. Clementi, and N. Borgese, "A cellular system to study the role of nitric oxide in cell death, survival, and migration," *Neurotoxicology*, vol. 26, no. 5, pp. 841–845, 2005.

[3] A. Contestabile and E. Ciani, "Role of nitric oxide in the regulation of neuronal proliferation, survival and differentiation," *Neurochemistry international*, vol. 45, no. 6, pp. 903–914, 2004.

[4] L. Wolpert, "One hundred years of positional information," *Trends in Genetics*, vol. 12, no. 9, pp. 359–364, 1996.

[5] J. Gurdon and P. Bourillot, "Morphogen gradient interpretation," *Nature*, vol. 413, no. 6858, pp. 797–803, 2001.

[6] K. Murari, M. Stanacevic, G. Cauwenberghs, and N. Thakor, "Integrated potentiostat for neurotransmitter sensing," *IEEE Engineering in Medicine and Biology Magazine*, vol. 24, no. 6, pp. 23–29, 2005.

[7] M. Stanacevic, K. Murari, A. Rege, G. Cauwenberghs, and N. Thakor, "Vlsi potentiostat array with oversampling gain modulation for wide-range neurotransmitter sensing," *IEEE Transactions on Biomedical Circuits and Systems*, vol. 1, no. 1, pp. 63–72, 2007.

[8] J. Zhang, Y. Huang, N. Trombly, C. Yang, and A. Mason, "Electrochemical array microsystem with integrated potentiostat," in *IEEE Sensors*, p. 4, 2005.

[9] A. Karegar, "Electrochemical, multi-analyte biosensor array for neurotransmitter detection," Master's thesis, Colorado State University, 2007.

[10] P. Kissinger, C. Preddy, R. Shoup, and W. Heineman, "Fundamental concepts of analytical electrochemistry," in *Laboratory Techniques in Electroanalytical Chemistry* (P. Kissinger and W. Heineman, eds.), pp. 11–50, Marcel Deker, 1996.

[11] P. Kissinger, "Introduction to analog instrumentation," in *Laboratory Techniques in Electroanalytical Chemistry* (P. Kissinger and W. Heineman, eds.), pp. 165–194, Marcel Deker, 1996.

[12] P. Kissinger and T. Ridgway, "Small-amplitude controlled-potential techniques," in *Laboratory Techniques in Electroanalytical Chemistry* (P. Kissinger and W. Heineman, eds.), pp. 141–164, Marcel Deker, 1996.

[13] K. Shibuki, "An electrochemical microprobe for detecting nitric oxide release in brain tissue.," *Neuroscience research*, vol. 9, no. 1, p. 69, 1990.

[14] F. Bedioui, S. Trevin, and J. Devynck, "Chemically modified microelectrodes designed for the electrochemical determination of nitric oxide in biological systems," *Electroanalysis*, vol. 8, no. 12, 1996.

[15] X. Zhang, J. Lin, L. Cardoso, M. Broderick, and V. Darley-Usmar, "A novel microchip nitric oxide sensor with sub-nm detection limit," *Electroanalysis*, vol. 14, no. 10, 2002.

[16] X. Zhang, Y. Kislyak, J. Lin, A. Dickson, L. Cardosa, M. Broderick, and H. Fein, "Nanometer size electrode for nitric oxide and s-nitrosothiols measurement," *Electrochemistry Communications*, vol. 4, no. 1, pp. 11–16, 2002.

[17] I. Kim, H. Bae, G. Oh, H. Chung, Y. Kim, and H. Chun, "Integrated microdisk gold electrode modified with metal-porphyrin and metal-phthalocyanines for nitric oxide determination in biological media," *BULLETIN-KOREAN CHEMICAL SOCIETY*, vol. 24, no. 11, pp. 1579–1584, 2003.

[18] Y. Wang, Q. Li, and S. Hu, "A multiwall carbon nanotubes film-modified carbon fiber ultramicroelectrode for the determination of nitric oxide radical in liver mitochondria," *Bioelectrochemistry*, vol. 65, no. 2, pp. 135–142, 2005.

[19] A. Frey, M. Jenkner, M. Schienle, C. Paulus, B. Holzapfl, P. Schindler-Bauer, F. Hofmann, D. Kuhlmeier, J. Krause, J. Albers, *et al.*, "Design of an integrated potentiostat circuit for cmos bio sensor chips," in *Circuits and Systems, 2003. ISCAS'03. Proceedings of the 2003 International Symposium on*, vol. 5, 2003.

[20] P. Steffan and R. Vrba, "Single chip potentiostat measurement system," *International Federation for Information Processing*, vol. 245, p. 653, 2007.

[21] P. George, J. Muthuswamy, J. Currie, N. Thakor, and M. Paranjape, "Fabrication of screen-printed carbon electrode arrays for sensing neuronal messengers," *Biomedical Microdevices*, vol. 3, no. 4, pp. 307–313, 2001.

[22] M. Naware, N. Thakor, R. Orth, K. Murari, and P. Passeraub, "Design and microfabrication of a polymer-modified carbon sensor array for the measurement of neurotransmitter signals," in *Engineering in Medicine and Biology Society, 2003. Proceedings of the 25th Annual International Conference of the IEEE*, vol. 2, 2003.

[23] I. Hafez, K. Kisler, K. Berberian, G. Dernick, V. Valero, M. Yong, H. Craighead, and M. Lindau, "Electrochemical imaging of fusion pore openings by electrochemical detector arrays," *Proceedings of the National Academy of Sciences*, vol. 102, no. 39, pp. 13879–13884, 2005.

[24] A. Hassibi and T. Lee, "A programmable 0.18-μm cmos electrochemical sensor microarray for biomolecular detection," *IEEE Sensors Journal*, vol. 6, no. 6, pp. 1380–1388, 2006.

[25] E. Mosharok and D. Sulzer, "Analysis of exocytotic events recorded by amperometry," *Nature Methods*, vol. 2, no. 9, pp. 651–658, 2005.

[26] Y. Iwasaki and M. Morita, "Electrochemical measurements with interdigitated array microelectrodes," *Current Separations*, vol. 14, no. 1, p. 3, 1995.

[27] S. Tobet, H. Walker, M. Seney, and K. Yu, "Viewing cell movements in the developing neuroendocrine brain 1," *Integrative and Comparative Biology*, vol. 43, no. 6, pp. 794–801, 2003.

Appendix A

Source Code

The following code was used to manage, analyze, and graph output results from CH Instruments 1207 and 660B potentiostats that had been converted to text. It supports cyclic voltammetry, amperometry, and differential pulse voltammetry files. It is written for Python 2.5 and uses the Django package. Included here are only the files that perform management, analysis, and graphing. It is available in its entirety online at http://github.com/mjibson/biosensor/.

biosensor/models.py:

```
from decimal import Decimal
from django.conf import settings
from django.db import models
from django import forms
import os

def get_point(time, target):
  for i in time:
    if target < i:
      return time.index(i)

MODE_HI_LOW = 1

def calc_range(fname, mode):
  f = open(fname)

  time = []
  value = []

  for line in f:
    l = line.split()
    time.append(float(l[0]))
    value.append(float(l[1]))

  f.close()

  if mode == MODE_HI_LOW:
    minv = min(value)
    maxv = max(value)
    mini = value.index(minv)
    maxi = value.index(maxv)
    return [[minv, time[mini]], [maxv, time[maxi]]]
  else:
    raise ValueError, 'unknown mode: %s' %mode

class Result(models.Model):
  sensor = models.IntegerField(null=True, blank=True)
  electrode = models.IntegerField(null=True, blank=True)
  run_date = models.DateTimeField()
  upload_date = models.DateTimeField()
  upload_file = models.FileField(upload_to="uploads")
  solution = models.CharField(max_length=100, blank=True)
  notes = models.TextField(max_length=500, blank=True)
  filename = models.CharField(max_length=100)
  analysis = models.CharField(max_length=100)
  init_e = models.DecimalField(null=True, blank=True, max_digits=4, decimal_places=2)
  high_e = models.DecimalField(null=True, blank=True, max_digits=4, decimal_places=2)
  low_e = models.DecimalField(null=True, blank=True, max_digits=4, decimal_places=2)
  init_pn = models.CharField(null=True, blank=True, max_length=1)
  scan_rate = models.DecimalField(null=True, blank=True, max_digits=10, decimal_places=3)
  sample_interval = models.DecimalField(null=True, blank=True, max_digits=6, decimal_places=5)
```

```
    final_e = models.DecimalField(null=True, blank=True, max_digits=4, decimal_places=2)
    incr_e = models.DecimalField(null=True, blank=True, max_digits=6, decimal_places=4)
    amplitude = models.DecimalField(null=True, blank=True, max_digits=6, decimal_places=4)
    pulse_width = models.DecimalField(null=True, blank=True, max_digits=4, decimal_places=3)
    sample_width = models.DecimalField(null=True, blank=True, max_digits=6, decimal_places=5)
    pulse_period = models.DecimalField(null=True, blank=True, max_digits=4, decimal_places=3)
    sensitivity = models.DecimalField(null=True, blank=True, max_digits=12, decimal_places=11)
    use = models.BooleanField(null=True, default=True)
    high_val = models.DecimalField(null=True, blank=True, max_digits=20, decimal_places=18)
    low_val = models.DecimalField(null=True, blank=True, max_digits=20, decimal_places=18)
    high_time = models.DecimalField(null=True, blank=True, max_digits=20, decimal_places=16)
    low_time = models.DecimalField(null=True, blank=True, max_digits=20, decimal_places=16)
    characterize = models.BooleanField(null=True, blank=True, default=False)
    characterize_low = models.DecimalField(null=True, blank=True, max_digits=10, decimal_places=4)
    characterize_mid = models.DecimalField(null=True, blank=True, max_digits=10, decimal_places=4)
    characterize_high = models.DecimalField(null=True, blank=True, max_digits=10, decimal_places=4)
    characterize_peak = models.DecimalField(null=True, blank=True, max_digits=20, decimal_places=18)
    characterize_value = models.DecimalField(null=True, blank=True, max_digits=20, decimal_places=18)
    characterize_base = models.DecimalField(null=True, blank=True, max_digits=20, decimal_places=18)
    characterize_time = models.DecimalField(null=True, blank=True, max_digits=20, decimal_places=10)

    def analyze(self):
      name = self.upload_file.name
      os.popen(settings.PROG_AWK + ' -v analysis="' + self.analysis + '" -f ' + settings.MEDIA_ROOT + '
          results/plot.awk ' + name)

      r = calc_range(name + '.avg', MODE_HI_LOW)
      self.low_val = str(r[0][0])
      self.low_time = str(r[0][1])
      self.high_val = str(r[1][0])
      self.high_time = str(r[1][1])

      for l in self.notes.splitlines():
        p = l.partition('=')
        if p[0] == 'ip':
          self.characterize_value = Decimal(p[2])
          break

      self.save()
      os.popen(settings.PROG_GNUPLOT + ' ' + name + '.plt')

    def __unicode__(self):
      return self.filename + ': ' + self.run_date.strftime('%d %b %y %H:%M:%S')

    class Admin:
      pass

WE_3 = 0
WE_C = 1
WE_I = 2
WE_F = 3

WE_CHOICES = (
    (WE_3, 'three'),
    (WE_C, 'four: C'),
    (WE_I, 'four: inverse C'),
    (WE_F, 'four: F')
)

SEN_2_AUX       = 0
SEN_COMR        = 1
SEN_COMR_COMA   = 2
SEN_COMR_COMA3 = 3

SENSOR_CHOICES = (
    (SEN_2_AUX, '2 aux'),
    (SEN_COMR, 'com ref at top and bottom'),
    (SEN_COMR_COMA, 'com ref at top, com aux at bottom'),
    (SEN_COMR_COMA3, 'com ref at top, com aux on 3 sides')
)

class Sensor(models.Model):
    sensor = models.PositiveSmallIntegerField()
    sensor_type = models.PositiveSmallIntegerField(choices=SENSOR_CHOICES)
    we_type = models.PositiveSmallIntegerField(choices=WE_CHOICES)

    def __unicode__(self):
      return '%02i: %s, %s electrode' %(self.sensor, self.get_sensor_type_display(), self.
          get_we_type_display())

    class Admin:
      pass

class Electrode(models.Model):
    sensor = models.ForeignKey(Sensor)
    we = models.PositiveSmallIntegerField()
    area = models.DecimalField(max_digits=5, decimal_places=3)
    area_ae = models.DecimalField(max_digits=8, decimal_places=3, help_text='auxiliary electrode area')
    perimeter = models.DecimalField(max_digits=5, decimal_places=3)
    perimeter_ae = models.DecimalField(max_digits=8, decimal_places=3, help_text='auxiliary electrode
        perimeter')
    distance = models.DecimalField(max_digits=4, decimal_places=2, help_text='shortest distance from working
        to aux electrode')

    def __unicode__(self):
      return 's%02dw%02d - area: %s, perimeter: %s, distance: %s' %(self.sensor.sensor, self.we, self.area,
          self.perimeter, self.distance)

    class Admin:
      pass

class UploadForm(forms.Form):
    upload_file = forms.FileField()
    sensor = forms.IntegerField(required=False)
    electrode = forms.IntegerField(required=False)
    solution = forms.CharField(max_length=100, required=False)
    notes = forms.CharField(max_length=500, widget=forms.Textarea, required=False)
```

```
    use = forms.BooleanField(required=False)
```

biosensor/plot.awk:

```
BEGIN {
  line = -1;
  FS = ", ";
  avg = 1;

  if(analysis == "Cyclic_Voltammetry")
  {
    xlabel = "Potential/V";
  }
  else if(analysis == "i_-_t_Curve")
  {
    xlabel = "Time/sec";
  }
  else if(analysis == "Differential_Pulse_Voltammetry")
  {
    xlabel = "Potential/V";
    avg = 0;
  }
}

FNR == 1 {
  fplt = FILENAME ".plt";
  print "set_terminal_png_size_640,_480" > fplt;
  print "set_xlabel_\"" xlabel "\"" > fplt;
  print "set_ylabel_\"Current/A\"" > fplt;

  print "set_output_\"" FILENAME ".avg.png\"" > fplt;
  print "plot_\"" FILENAME ".avg\"_with_lines" > fplt;
  print "set_output_\"" FILENAME ".png\"" > fplt;

  if(avg == 1)
  {
    print "plot_\\" > fplt;
    print "\"" FILENAME ".dat1\"_with_lines_,_\\" > fplt;
    print "\"" FILENAME ".dat2\"_with_lines,_\\" > fplt;
    print "\"" FILENAME ".dat3\"_with_lines,_\\" > fplt;
    print "\"" FILENAME ".dat4\"_with_lines,_\\" > fplt;
    print "\"" FILENAME ".dat5\"_with_lines" > fplt;
  }

  if(analysis == "i_-_t_Curve")
  {
    print "set_output_\"" FILENAME ".+15.png\"" > fplt;
    print "plot_[15:]_\"" FILENAME ".avg\"_with_lines" > fplt;
    print "set_output_\"" FILENAME ".r5.png\"" > fplt;
    print "plot_[:][-5e-10:5e-10]_\"" FILENAME ".avg\"_with_lines" > fplt;
  }

  print "set_terminal_png_size_200,_100" > fplt;
  print "set_output_\"" FILENAME ".tn.png\"" > fplt;
  print "set_lmargin_.2" > fplt;
  print "set_rmargin_.2" > fplt;
  print "set_bmargin_.2" > fplt;
  print "set_tmargin_.2" > fplt;
  print "unset_xlabel" > fplt;
  print "unset_ylabel" > fplt;
  print "unset_tics" > fplt;
  print "plot_\"" FILENAME ".avg\"_notitle_with_lines" > fplt;
}

/^[-0-9]/ {
  if(avg == 0)
  {
    avgname = FILENAME ".avg";
    print $1 "_" $2 > avgname;
  }
  else
  {
    line += 1;
    idx = line % 5;
    data[idx] = $2;
    time[idx] = $1;
    fname = FILENAME ".dat" idx + 1;
    print $1 "_" $2 > fname;

    if(idx == 4)
    {
      avgname = FILENAME ".avg";
      range1 = FILENAME ".-1_1";
      range2 = FILENAME ".-2_2";

      avgtime = (time[0] + time[1] + time[2] + time[3] + time[4]) / 5;
      avgdata = (data[0] + data[1] + data[2] + data[3] + data[4]) / 5;
      print avgtime "_" avgdata > avgname;

      if(avgtime >= -.2 && avgtime <= .2)
        print avgtime "_" avgdata > range2;

      if(avgtime >= -.1 && avgtime <= .1)
        print avgtime "_" avgdata > range1;
    }
  }
}
```

biosensor/views.py:

```
from django.core.paginator import QuerySetPaginator
from django.http import HttpResponse
from django.shortcuts import render_to_response, get_object_or_404
from biosensor.results.models import *
from biosensor import settings
```

```
from decimal import Decimal
from numpy import polyfit, polyval
import datetime
import re
import os
import math

import matplotlib
matplotlib.use('AGG')
import matplotlib.pyplot as plt
from pylab import setp

def render(request, template, dict={}):
  r = Result.objects.all().order_by('-run_date', '-upload_date')
  p = QuerySetPaginator(r, 50)

  try:
    page_num = int(request.GET['p'])
    if page_num < 1:
      page_num = 1
    elif page_num > p.num_pages:
      page_num = p.num_pages
  except:
    page_num = 1

  page = p.page(page_num)

  dict['p'] = p
  dict['page'] = page
  dict['results'] = result_list(page.object_list)
  dict['r'] = result_list(r)

  return render_to_response(template, dict)

def result_list(results):
  result_list = []
  d = ''

  for res in results:
    nd = res.run_date.strftime("%d %b %y")
    if nd != d:
      d = nd
      result_list.append([])

    result_list[-1].append(res)

  return result_list

def index(request):
  return render(request, 'results/base.html')

def electrode(request):
  return render(request, 'results/electrode.html', {'electrodes': Electrode.objects.all().order_by('sensor
      ', 'we')})

def limit(request):
  lod = Result.objects.filter(notes__contains='lod = ')
  limits = {}

  for s in lod:
    for l in s.notes.splitlines():
      p = l.partition(' = ')
      if p[0] == 'lod':
        if s.sensor not in limits:
          limits[s.sensor] = []

        c = s.solution.partition('M')[0]
        conc = Decimal(c[:-1])

        if c[-1] == 'm':
          conc *= Decimal('1e-3')
        elif c[-1] == 'u':
          conc *= Decimal('1e-6')
        else:
          raise ValueError, 'unknown modifier'

        limits[s.sensor].append((conc, Decimal(p[2])))
        break

  for sensor, dat in limits.iteritems():
    xdat = map(float, zipcol(dat, 0))
    ydat = map(float, zipcol(dat, 1))

    plt.plot(xdat, ydat, '+', label='%s' %sensor)
    plt.plot(xdat, polyval(polyfit(xdat, ydat, 1), xdat), label='fit for %s' %sensor)

  plt.xlabel(r'concentration ($\mathrm{M}$)')
  plt.ylabel(r'current ($\mathrm{A}$)')
  plt.axis(xmin=0, xmax=0.00035)
  plt.legend(loc='upper left')
  plt.savefig(settings.MEDIA_ROOT + 'uploads/sensors/limit.png')
  plt.clf()

  limlist = []
  for s, lim in limits.iteritems():
    for conc, value in lim:
      limlist.append((s, conc, value))

  limlist.sort(cmp=lambda x, y: cmp(x[1], y[1]), reverse=True)

  return render(request, 'results/limit.html', {'limits': limlist})

def mean(list):
  return sum(list) / len(list)

def stdev(list):
  a = mean(list)
```

```
  v = [(i - a) ** 2.0 for i in list]
  return math.sqrt(sum(v) / len(list))

def sterror(list):
  return stdev(list) / math.sqrt(len(list))

def sensor(request):
  sensors = Result.objects.filter(use=True)

  count = {}
  avg = {}
  dev = {}
  error = {}

  for s in sensors:
    area = Electrode.objects.get(sensor__sensor=s.sensor, we=s.electrode).area
    if s.sensor not in count:
      count[s.sensor] = 0
      avg[s.sensor] = 0

    avg[s.sensor] += float(s.characterize_value)
    count[s.sensor] += 1

  sensors = sensors.order_by('-characterize_value')

  for k in avg.keys():
    avg[k] /= count[k]
    v = list(float(i.characterize_value) for i in sensors.filter(sensor=k))
    dev[k] = stdev(v)
    error[k] = sterror(v)

  senlist = []
  for k, v in avg.iteritems():
    senlist.append((k, v))
  senlist.sort(cmp=lambda x, y: cmp(x[1], y[1]), reverse=True)

  perc = []
  m = senlist[0][1] / 100.0

  for (id, v) in senlist:
    se = error[id]
    perc.append([id, v, v / m, se / m, se])

  # matplotlib data for use=True

  res = {}

  for r in sensors:
    s = 's%02iw%02i' %(r.sensor, r.electrode)
    if s not in res:
      res[s] = []
    res[s].append(float(r.characterize_value))

  s = res.keys()
  s.sort()
  pltdat = []

  for k in s:
    d = [k[1:3], k[4:6]]
    electrode = Electrode.objects.get(sensor__sensor=d[0], we=d[1])
    m = mean(res[k])
    e = sterror(res[k])
    density = [i / float(electrode.area) for i in res[k]]
    d.append('%.3e_$\pm$_%.3e' %(m, e))
    pltdat.append((
      electrode.sensor.sensor,
      electrode.we,
      None, # chip
      m, # output mean
      e, # output standard error
      mean(density),
      sterror(density),
      float(electrode.area),
      float(electrode.perimeter),
      float(electrode.distance),
      float(electrode.perimeter / electrode.area),
      float(electrode.area / electrode.area_ae),
      electrode.sensor.sensor_type
    ))

  p_output = plot('area_v_output', pltdat, PLT_AREA, PLT_MEAN, fit=True)
  p_density = plot('area_v_density', pltdat, PLT_AREA, PLT_MEAN_DEN, fit=True)
  plot('area_low_v_density', pltdat, PLT_AREA, PLT_MEAN_DEN, axis = {'xmin': 0, 'xmax': 5})
  plot('area_high_v_density', pltdat, PLT_AREA, PLT_MEAN_DEN, axis = {'xmin': 10})

  # matplotlib data for specific tests

  chips = ['3', '4']
  res = {}

  f = Result.objects.filter(run_date__gte=datetime.datetime(2009, 5, 8), run_date__lte=datetime.datetime
      (2009, 5, 10))
  for chip in chips:
    for r in f.filter(filename__contains='chip%s' %chip):
      s = 's%02iw%02i' %(r.sensor, r.electrode)
      if s not in res:
        res[s] = {}
      if chip not in res[s]:
        res[s][chip] = []
      res[s][chip].append(float(r.characterize_value))

  s = res.keys()
  s.sort()
  specific = [['sensor', 'electrode']]
  pltdat = []
  for c in chips:
    specific[0].append('chip_%s' %c)
```

```
    for k in s:
      d = [k[1:3], k[4:6]]
      electrode = Electrode.objects.get(sensor__sensor=d[0], we=d[1])
      for c in chips:
        if c not in res[k]:
          d.append('')
        else:
          m = mean(res[k][c])
          e = sterror(res[k][c])
          density = [i / float(electrode.area) for i in res[k][c]]
          d.append('%.3e_$\pm$_%.3e' %(m, e))
          pltdat.append((
            electrode.sensor.sensor,
            electrode.we,
            c, # chip
            m, # output mean
            e, # output standard error
            mean(density),
            sterror(density),
            float(electrode.area),
            float(electrode.perimeter),
            float(electrode.distance),
            float(electrode.perimeter / electrode.area),
            float(electrode.area / electrode.area_ae),
            electrode.sensor.sensor_type
          ))
      specific.append(d)

    plot('perim_area_v_density', pltdat, PLT_PERIM_AREA, PLT_MEAN_DEN, axis = {'xmin': 1.5, 'xmax': 4.5},
        fit=True)
    plot('distance_v_density', pltdat, PLT_DIST, PLT_MEAN_DEN)
    plot('area_ratio_v_output', pltdat, PLT_AREA_RATIO, PLT_MEAN, axis = {'xmin': -0.1, 'xmax': 1.1})
    plot('shape_v_density', pltdat, PLT_SENSOR_SHAPE, PLT_MEAN_DEN, shape_hack=True, axis = {'xmin': -0.5, '
        xmax': 3.5})

    return render(request, 'results/sensors.html', {'sensors': sensors, 'perc': perc, 'chips': chips, '
        specific': specific, 'p_output': '%.3e_*_x_+_%.3e' %(p_output[0], p_output[1]), 'p_density': '%.3e_
        *_x_+_%.3e' %(p_density[0], p_density[1])})

def zipcol(lst, col):
    return [i[col] for i in lst]

PLT_SENSOR = 0
PLT_WE = 1
PLT_CHIP = 2
PLT_MEAN = 3
PLT_STERR = 4
PLT_MEAN_DEN = 5
PLT_STERR_DEN = 6
PLT_AREA = 7
PLT_PERIM = 8
PLT_DIST = 9
PLT_PERIM_AREA = 10
PLT_AREA_RATIO = 11
PLT_SENSOR_SHAPE = 12

colnames = [
    'sensor',
    'working_electrode',
    'chip',
    r'output_($\mathrm{A}$)',
    'output_standard_error',
    r'density_($\mathrm{A}_/_\mu_\mathrm{m}^2$)',
    'density_standard_error',
    r'working_electrode_area_($\mu_\mathrm{m}^2$)',
    r'working_electrode_perimeter_($\mu_\mathrm{m}$)',
    r'distance_between_WE_and_AE_($\mu_\mathrm{m}$)',
    r'WE_perimeter_/_area_($\mu_\mathrm{m}$)',
    'WE_area_/_AE_area',
    'sensor_shape'
]

def plot(name, lst, x, y, axis={}, shape_hack=False, fit=False):
    xdat = zipcol(lst, x)
    ydat = zipcol(lst, y)

    plt.errorbar(
        xdat,
        ydat,
        yerr=zipcol(lst, y + 1),
        fmt='.'
    )

    if fit:
        p = polyfit(xdat, ydat, 1)
        plt.plot(xdat, polyval(p, xdat))

    plt.xlabel(colnames[x])
    plt.ylabel(colnames[y])
    plt.axis(**axis)

    if x == PLT_SENSOR_SHAPE:
        xa = plt.axes().xaxis
        xa.set_ticklabels(['2_AE', 'RE_top_&_bottom', 'RE_top,_AE_bottom', 'RE_top,_AE_on_3_sides'])
        xa.set_ticks([0, 1, 2, 3])
        labels = plt.axes().get_xticklabels()
        setp(labels, fontsize=8)

    plt.savefig(settings.MEDIA_ROOT + 'uploads/sensors/' + name)
    plt.clf()

    if fit:
        return p

def detail(request, result_id):
    r = get_object_or_404(Result, pk=result_id)
```

```
    try:
      if request.POST['reanalyze'] == 'reanalyze':
        if r.analysis == 'i_-_t_Curve':
          r.characterize = 'characterize' in request.POST and request.POST['characterize'] == 'on'
          if(request.POST['low']):
            r.characterize_low = request.POST['low']
          else:
            r.characterize_low = None
          if(request.POST['mid']):
            r.characterize_mid = request.POST['mid']
          else:
            r.characterize_mid = None
          if(request.POST['high']):
            r.characterize_high = request.POST['high']
          else:
            r.characterize_high = None
          r.save()

        r.analyze()
    except KeyError:
      pass

    try:
      s = Sensor.objects.get(sensor=r.sensor)
      e = Electrode.objects.get(sensor=s, we=r.electrode)
    except:
      e = None

    return render(request, 'results/detail.html', {'result': r, 'e': e})

def upload(request):
    months = {
      'Jan': 1,
      'Feb': 2,
      'Mar': 3,
      'Apr': 4,
      'May': 5,
      'June': 6,
      'July': 7,
      'Aug': 8,
      'Sept': 9,
      'Oct': 10,
      'Nov': 11,
      'Dec': 12
    }

    form = UploadForm()

    if request.method == 'POST' and 'all' in request.POST:
      res = Result.objects.all()

      for r in res:
        r.analyze()
    elif request.method == 'POST':
      form = UploadForm(request.POST, request.FILES)
      if form.is_valid():
        f = request.FILES['upload_file']
        s = f.read().splitlines()
        d = re.split('[\.,_:]+', s[0])

        r = Result(
          sensor = form.cleaned_data['sensor'],
          electrode = form.cleaned_data['electrode'],
          solution = form.cleaned_data['solution'],
          notes = form.cleaned_data['notes'],
          upload_date = datetime.datetime.now(),
          run_date = datetime.datetime(int(d[2]), months[d[0]], int(d[1]), int(d[3]), int(d[4]), int(d[5])),
          filename = f.name,
          analysis = s[1],
          use = form.cleaned_data['use'],
          init_e = s[8].split('_=_')[1],
        )

        if s[1] == 'Cyclic_Voltammetry':
          r.high_e = s[9].split('_=_')[1]
          r.low_e = s[10].split('_=_')[1]
          r.init_pn = s[11].split('_=_')[1]
          r.scan_rate = s[12].split('_=_')[1]
          r.sample_interval = s[14].split('_=_')[1]
          r.sensitivity = s[16].split('_=_')[1]

        elif s[1] == 'i_-_t_Curve':
          r.sample_interval = s[9].split('_=_')[1]
          r.sensitivity = s[12].split('_=_')[1]
        elif s[1] == 'Differential_Pulse_Voltammetry':
          r.final_e = s[9].split('_=_')[1]
          r.incr_e = s[10].split('_=_')[1]
          r.amplitude = s[11].split('_=_')[1]
          r.pulse_width = s[12].split('_=_')[1]
          r.sample_width = s[13].split('_=_')[1]
          r.pulse_period = s[14].split('_=_')[1]
          r.sensitivity = s[16].split('_=_')[1]

        if form.cleaned_data['sensor'] is None and len(r.filename) >= 3 and r.filename[0] == 's':
          r.sensor = r.filename[1:3]

        if form.cleaned_data['electrode'] is None and len(r.filename) >= 6 and r.filename[3] == 'w':
          r.electrode = r.filename[4:6]

        r.save()
        r.upload_file.save(str(r.id), f)
        r.save()

        r.analyze()

        return render(request, 'results/upload.html', {'form': UploadForm(), 'upload': r})
```

```
  return render(request, 'results/upload.html', {'form': form})


def syncdata(request):
  # (WE, area, perimeter)
  ap = {
    WE_3: [
      (1, 1, 4),
      (2, 2.25, 6),
      (3, 4, 8),
      (4, 4, 8)
    ],
    WE_C: [
      (1, 10.885, 24.7),
      (2, 11.342, 25.081)
    ],
    WE_I: [
      (1, 14.267, 30.793),
      (2, 15.119, 32.835)
    ],
    WE_F: [
      (1, 21.771, 46.636),
      (2, 21.738, 46.796)
    ]
  }

  # (sensor, working electrode)
  sensors = {
    0: (SEN_2_AUX, WE_3),
    1: (SEN_2_AUX, WE_3),
    2: (SEN_2_AUX, WE_3),
    3: (SEN_COMR, WE_3),
    4: (SEN_COMR, WE_3),
    5: (SEN_COMR_COMA, WE_C),
    6: (SEN_COMR_COMA, WE_I),
    7: (SEN_COMR_COMA, WE_I),
    8: (SEN_COMR_COMA, WE_F),
    9: (SEN_COMR, WE_3),
    10: (SEN_COMR_COMA3, WE_I),
    11: (SEN_COMR_COMA3, WE_3),
    12: (SEN_COMR_COMA3, WE_3),
    13: (SEN_COMR_COMA3, WE_3),
    14: (SEN_COMR_COMA3, WE_C),
    15: (SEN_COMR_COMA3, WE_C),
    16: (SEN_COMR_COMA3, WE_F),
    17: (SEN_COMR_COMA, WE_I),
    18: (SEN_COMR_COMA3, WE_F),
    19: (SEN_COMR, WE_3),
    20: (SEN_2_AUX, WE_3)
  }

  # (area, perimeter), or None if same as WE
  ae = [
    None,                 # 00
    None,                 # 01
    None,                 # 02
    None,                 # 03
    None,                 # 04
    (115, 56),            # 05
    (152.5, 71),          # 06
    (177.5, 81),          # 07
    (161.25, 74.5),       # 08
    None,                 # 09
    (865, 356),           # 10
    (820, 338),           # 11
    (1045, 428),          # 12
    (1270, 518),          # 13
    (1333.75, 543.5),     # 14
    (1108.75, 453.5),     # 15
    (1121.875, 458.75),   # 16
    (186.25, 84.5),       # 17
    (1346.875, 548.75),   # 18
    None,                 # 19
    None                  # 20
  ]

  Sensor.objects.all().delete()
  Electrode.objects.all().delete()

  for sensor, (sen, we) in sensors.iteritems():
    s = Sensor(sensor=sensor, sensor_type=sen, we_type=we)
    s.save()

    if we == WE_3:
      elist = [0]
    else:
      elist = [10, 20, 30, 40]

    for electrode, area, perimeter in ap[we]:
      for i in elist:
        e = Electrode(sensor=s, we=Decimal(str(i + electrode)), area=Decimal(str(area)), perimeter=Decimal
            (str(perimeter)))

        if ae[sensor] is None:
          e.area_ae = e.area
          e.perimeter_ae = e.perimeter
        else:
          e.area_ae = Decimal(str(ae[sensor][0]))
          e.perimeter_ae = Decimal(str(ae[sensor][1]))

        if sensor <= 4 or sensor == 9:
          if electrode == 3:
            d = 2.5
          else:
            d = 3.0
        elif sensor == 5:
```

```
            if i <= 20:
              d = 26.55
            else:
              d = 11.51
          elif sensor == 6:
            if i <= 20:
              d = 35.54
            else:
              d = 15.51
          elif sensor == 7:
            if i <= 20:
              d = 45.48
            else:
              d = 20.40
          elif sensor == 8:
            if i <= 20:
              d = 33.53
            else:
              d = 13.53
          elif sensor == 10:
            d = 11.5
          elif sensor == 11:
            if electrode == 1:
              d = 14.47
            elif electrode == 2:
              d = 14.24
            else:
              d = 13.96
          elif sensor == 12:
            if electrode == 1:
              d = 19.50
            elif electrode == 2:
              d = 19.21
            else:
              d = 18.98
          elif sensor == 13:
            if electrode == 1:
              d = 24.53
            elif electrode == 2:
              d = 24.26
            else:
              d = 24.05
          elif sensor == 14:
            d = 20.50
          elif sensor == 15:
            d = 15.50
          elif sensor == 16:
            d = 13.50
          elif sensor == 17:
            if i <= 20:
              d = 43.47
            else:
              d = 18.46
          elif sensor == 18:
            d = 18.50
          elif sensor == 19 or sensor == 20:
            if electrode == 1:
              d = 2.00
            elif electrode == 3:
              d = 2.50
            else:
              d = 3.00

          e.distance = Decimal(str(d))
          del d # throw exception next time if not set

          e.save()

    return HttpResponse('<html><body>syncdata successful.</body></html>')
```

Appendix B

Screenshots

The following screenshots are selected from the application described above.

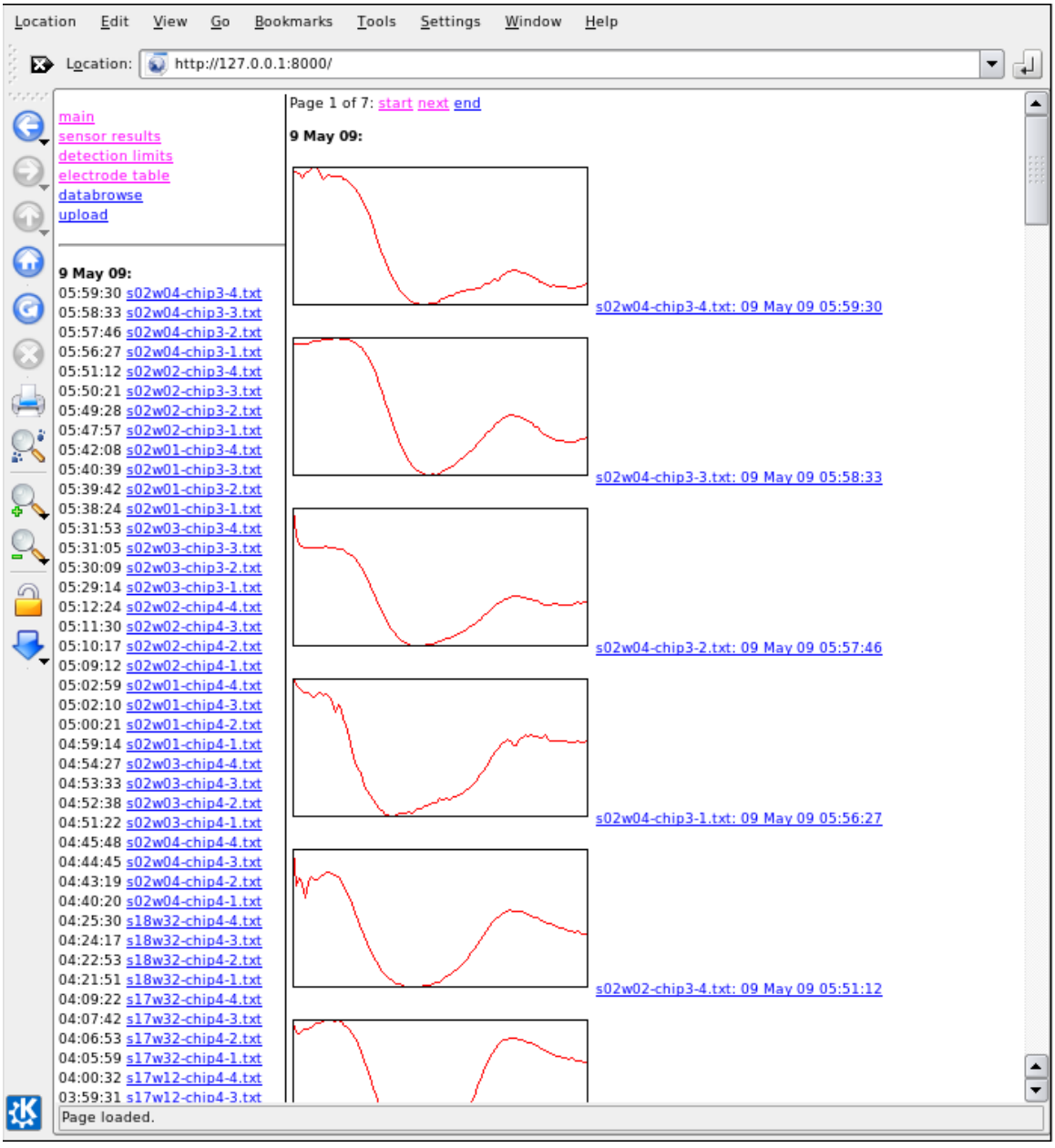


Figure B.1: Web application main page

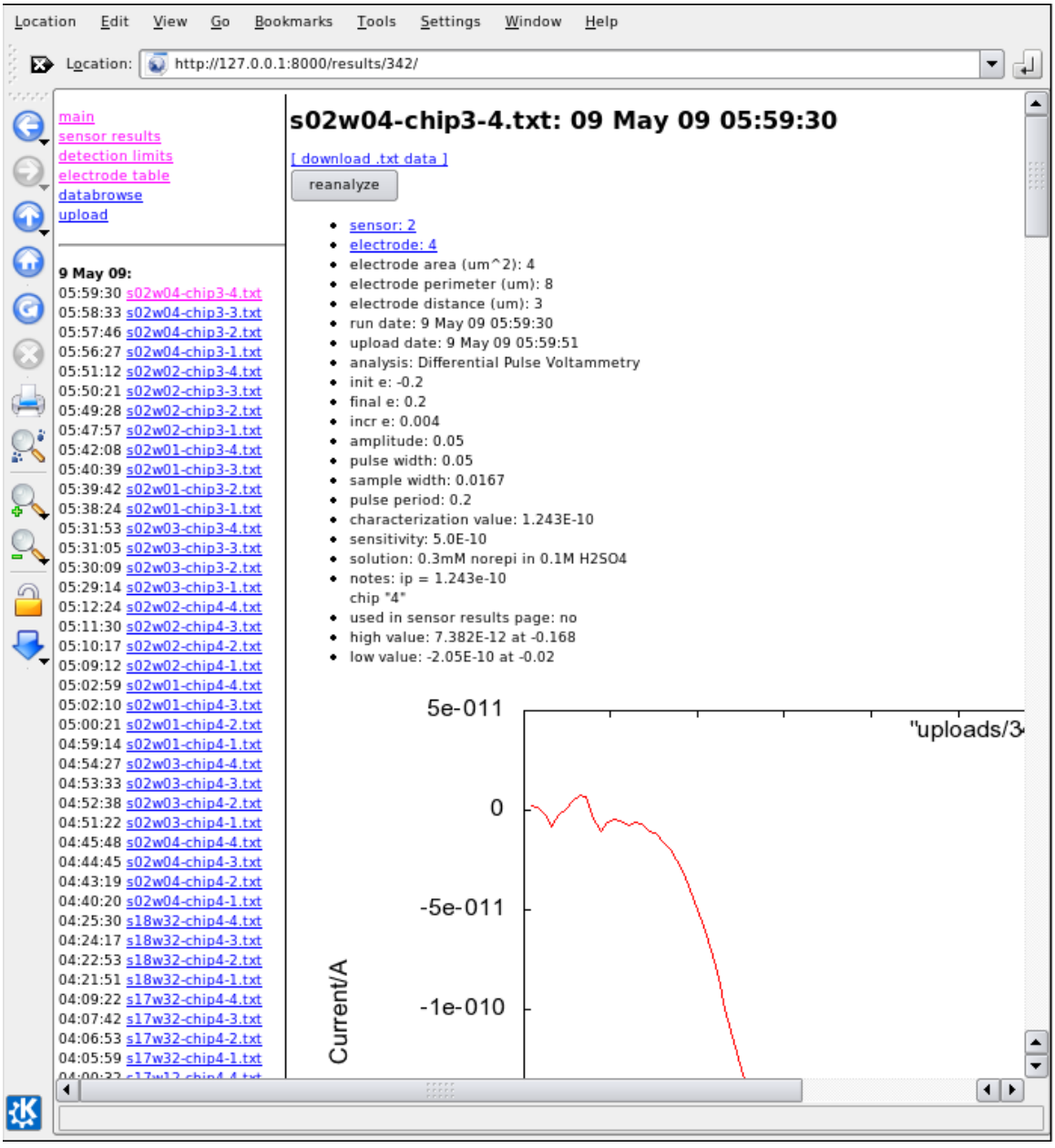


Figure B.2: Web application result detail

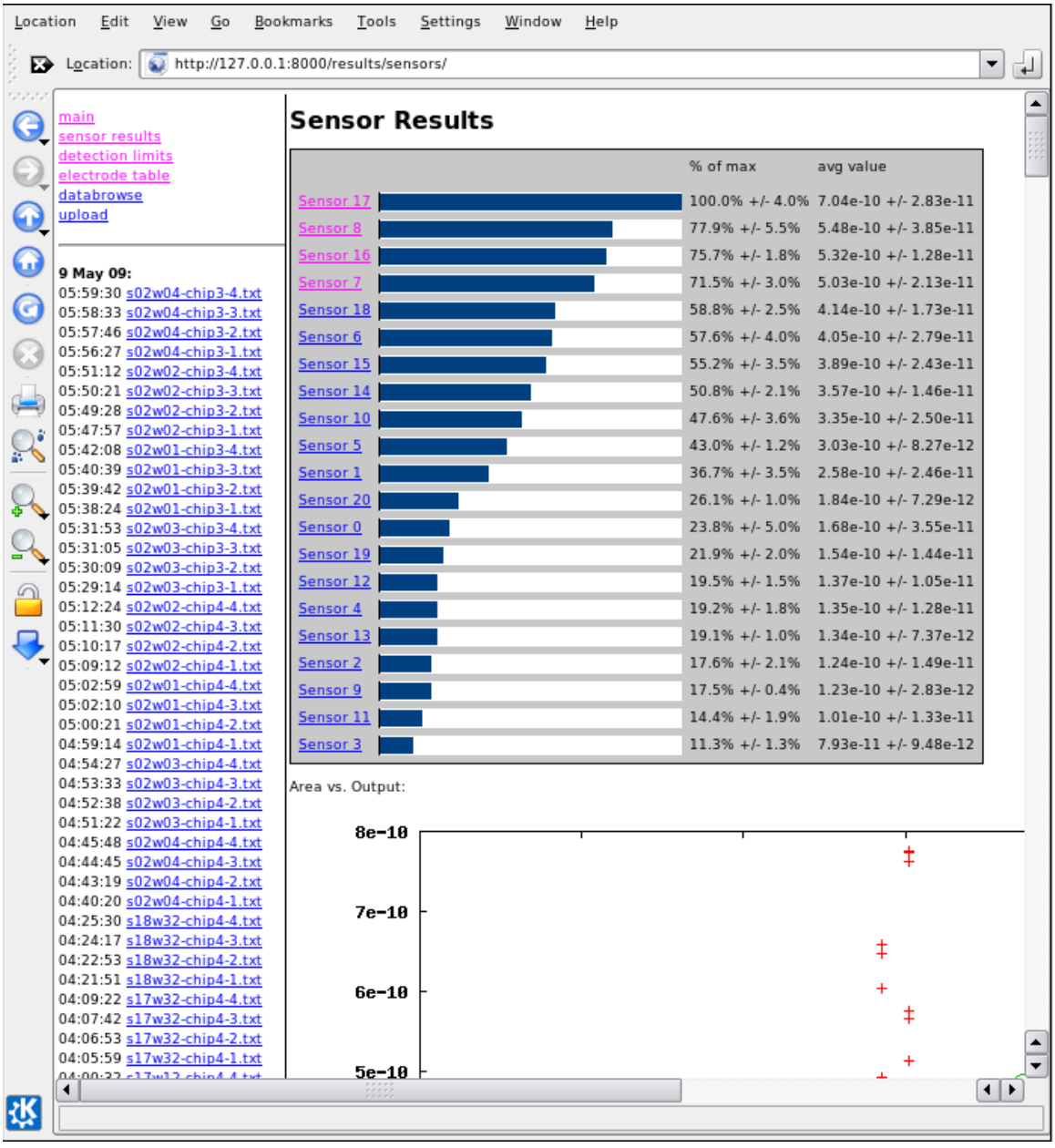


Figure B.3: Web application sensor results

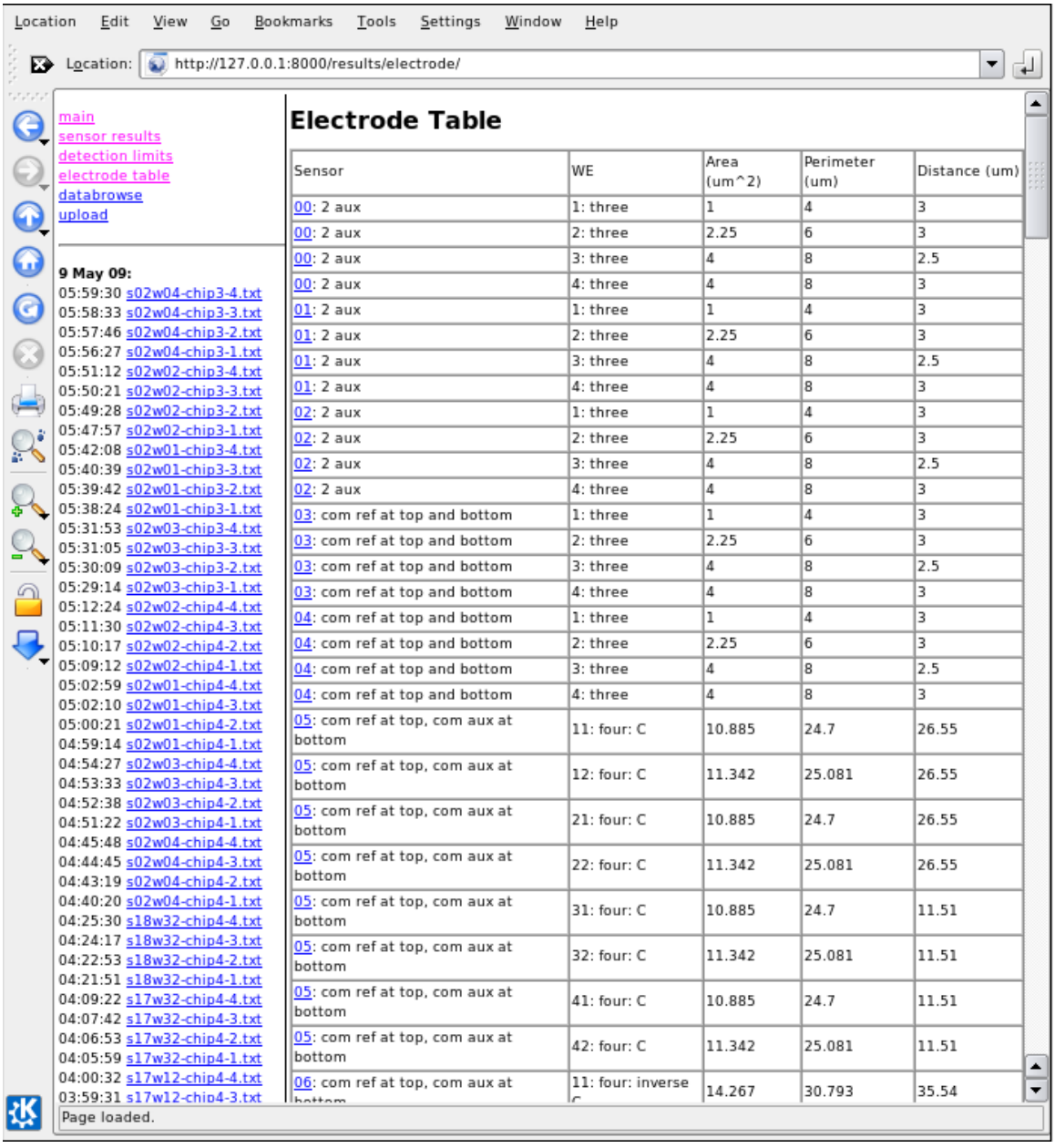


Figure B.4: Web application electrode table

Appendix C

Solution Mixing

This chapter describes the process for mixing the solutions used during testing. If the desired solution is 20mL of 0.3mM norepinephrine in 0.1M H_2SO_4:

1. Measure the correct amount of norepinephrine (labeled as arterenol in the chemistry lab) by finding its molecular weight, listed on the vial, which is 319.3g/mol. Multiply the weight, desired solution amount, and desired molarity together:

   $$0.3\text{mM} * 319.3\text{g/mol} * 20\text{mL} = 0.00192\text{g}$$

   Hence, measure 0.00192g of norepinephrine, and put it in a vial.

2. Create your 0.1M H_2SO_4 solution by first determining how much distilled water is needed. Find the molarity of H_2SO_4, which is 18.4M. Divide the product of the desired final molarity and the desired volume by that molarity:

   $$(0.1\text{M} * 20\text{mL})/18.4\text{M} = 0.1087\text{mL}$$

   Hence, 0.1087mL is the amount of H_2SO_4 needed to make 20mL of 0.1M H_2SO_4.

3. Add the appropriate amount of distilled water to the vial with the norepinephrine. Since we are getting 20mL, subtract the number found in the previous step, and add that much distilled water:

   $$20\text{mL} - 0.1087\text{mL} = 19.89\text{mL}$$

   Hence, add 19.89mL of distilled water to the vial. Always add the water first. Do not add the acid before the water.

4. Now, after adding the water, add the acid. Add 0.1087mL of H_2SO_4 to the vial.

You should now have 20mL of 0.3mM norepinephrine in 0.1M H_2SO_4. It will last a few days before degrading to an unusable point. If kept in a fridge it will last longer.