THE EFFECT OF TRANSVERSE MIXING
ON TRACER DISPERSION
IN A FRACTURE

A REPORT

SUBMITTED TO THE DEPARTMENT OF PETROLEUM ENGINEERING

OF STANFORD UNIVERSITY

IN PARTIAL FULFILLMENT OF THE REQUIREMENTS

FORTHEDEGREEOF

MASTER OF SCIENCE

by

LAWRENCE W. BOUETT

June, 1986

TABLE OF CONTENTS

ACKNOWLEDGEMENTS

For his advice and encouragement throughout this project, to Professor Roland **N.** Home **I** give **my** sincerest thanks.

For their work in developing the excellent software by which the graphs in the appendices were produced, to Jeff Wingard and Wes Monroe **I** extend **my** gratitude.

DEDICATION

For their loyalty, and for the time they were deprived of their husband and father, it is but poor compensation to my wife Sybil and my sons Bill and John that I dedicate this work to them with all my love.

ABSTRACT

Laboratory investigations into the physical mechanisms of tracer dispersion and retention in a planar fracture were conducted. Iodide and chloride tracers were flowed in a fracture, first without obstructions and again with a nylon mesh to stimulate transverse mixing and to provide closed pore volumes. Tracer was flowed as a step input across the fracture width. Tracer concentration within the fracture was measured at a series of electrodes embedded in one of the confining plates. The measurements were controlled by and stored in a microcomputer. The theoretical response for linear Taylor dispersion was matched to the data to determine the non-linear parameters velocity and dispersivity. Despite noisy data it was observed that the presence of the nylon mesh did not significantly change the dispersivity. Proposals are presented to improve the quality of the data acquired.

LIST OF FIGURES

LIST OF TABLES

Section 1: INTRODUCTION

The study of the flow of fluids in fractures arises in part out of the need to dispose of waste hot water in geothermal fields. Although reinjected waste hot water may help to support reservoir pressures in some cases ***(Horne,*** 1982a) it may also result in a decline in enthalpy ***(Horne,*** 1982b) and may even decrease reservoir pressure dramatically if steam were to be condensed in the reservoir. **A** decrease in production is due to the premature breakthrough of reinjected water through high-conductivity fractures into the produced fluids. Reinjected water, although usually hot by ambient surface standards, is nevertheless colder than the hotter produced fluids.

Tracer tests have been used in several geothermal fields worldwide in an attempt to characterize the degree of fracturing ***(Horne,*** 1982c); however in order successfully to analyze tracer tests it is necessary to understand **as** completely as possible the transfer mechanisms of tracer transport through fractures. ***Horne and Rodriguez*** (1983) examined four mechanisms of dispersion in a planar fracture: molecular diffusion, convective dispersion, Taylor dispersion and turbulence.

Laboratory Experiments To Model Tracer Dispersion

In order to develop successful reservoir engineering techniques to forecast thermal breakthrough it is necessary to understand the fluid and heat transfer mechanics of the flow of fluids through fractures, to be able to obtain or infer the maximum amount of information from tracer tests, and to develop reliable tracer test techniques. To these ends we must develop theoretical models of low-temperature fluid flow in a fractured, high temperature reservoir, compare the model results with field data, perform experimental investigations in fluid flow in fractures, both in large- and small-scale physical models, and field test the stability of tracers.

The Present Work

In view of the above discussion it may be seen that a large-scale laboratory fracture model is a part of the methodology developed to integrate various parts of the solution to the problem of how to design a technique to forecast thermal breakthrough in a geothermal reservoir. *Gilardi* (1984) conducted investigations into the mechanism of Taylor dispersion in a fracture using a large-scale laboratory model. The present work continues that of Gilardi, broadening the scope of the investigations to include monitoring both dimensions in a planar fracture and to consider the effects of transverse mixing and the creation of dead pore volumes.

Section 2: THE MATHEMATICAL MODEL

Taylor Dispersion

Taylor's now-classic study of the mechanism of dispersion in the **flow** of fluids through capillary tubes (*Taylor*, 1953) gave us what has come to be known as "Taylor dispersion." Taylor dispersion (Fig. 1 [after *Gilardi*, 19841) combines the effects of convective dispersion and transverse molecular diffusion. ***Taylor*** (1953) presented the equation governing effective longitudinal dispersion:

$$\eta \frac{\partial^2 C}{\partial z^2} = \frac{\partial C}{\partial t} , \qquad (1)$$

where

η	=	net longitudinal dispersivity
C	=	tracer concentration
z	=	x-ut (translated distance)
x	=	distance
u	=	mean flow velocity
t	=	time.

Carslaw and Jaeger (1959) have solved Eq. 1 written in terms of heat transfer for a variety of inlet and boundary conditions. **As** Eq. 1 **is** written the initial condition is

$$C_{t=0} = 0, \quad x 2 0$$

and the inner boundary condition is

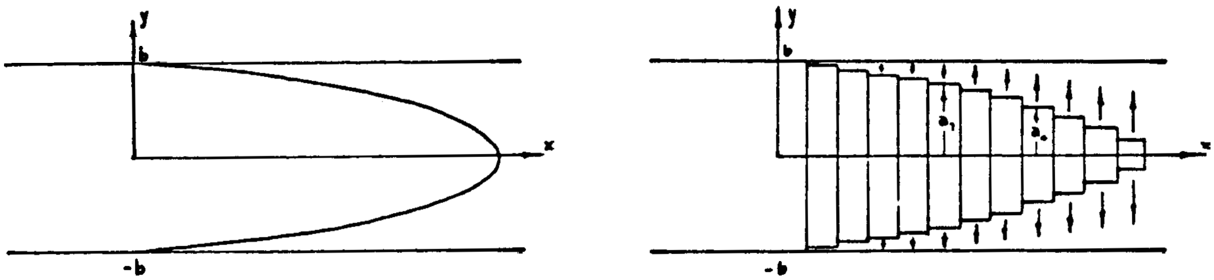
$$C_{x=0} = 0, \quad t 2 0.$$

Carslaw and Jaeger (1959) provide the solution to these conditions on Eq. 1. Written in terms of concentration, for a step input we have

$$C = C_o + \frac{1}{2}\left(C_1 - C_o\right)\left[erfc\left[\frac{x-ut}{2(\eta t)^{1/2}}\right] + exp^{\frac{ux}{\eta}}\, erfc\left[\frac{x+ut}{2(\eta t)^{1/2}}\right]\right], \quad (2)$$

where

C	=	concentration at x
C_o	=	base concentration
C_1	=	injected concentration
erfc	=	complimentary error function.



Convective Dispersion

Taylor Dispersion

Fig. 1: Schematic Representation Of Dispersion Mechanisms.

The Effect Of Taylor Dispersion

The effect of convective dispersion in a fracture is to generate large concentration gradients across the narrow fracture width. ***Horne and Rodriguez*** **(1983)** point out that in the presence of this large concentration gradient molecular diffusion tends to equalize tracer concentration across the fracture, thereby ameliorating the effect of convective dispersion. This important result may be seen graphically as Fig. 2 ***(Horne and Rodriguez,*** **1983),** in which the difference in dimensionless concentration between the centerline and the boundary of the fracture is given as a function of dimensionless time for various slug widths a. It is clear from Fig. 2 that regardless of the slug width the concentration is equalized across the fracture by a dimensionless time t_D = **0.5.** Here

$$t_D = \frac{D}{b^2}\, t \,, \tag{3}$$

where

t_D	=	dimensionless time function
D	=	coefficient of molecular diffusion for tracer
b	=	fracture half-width.

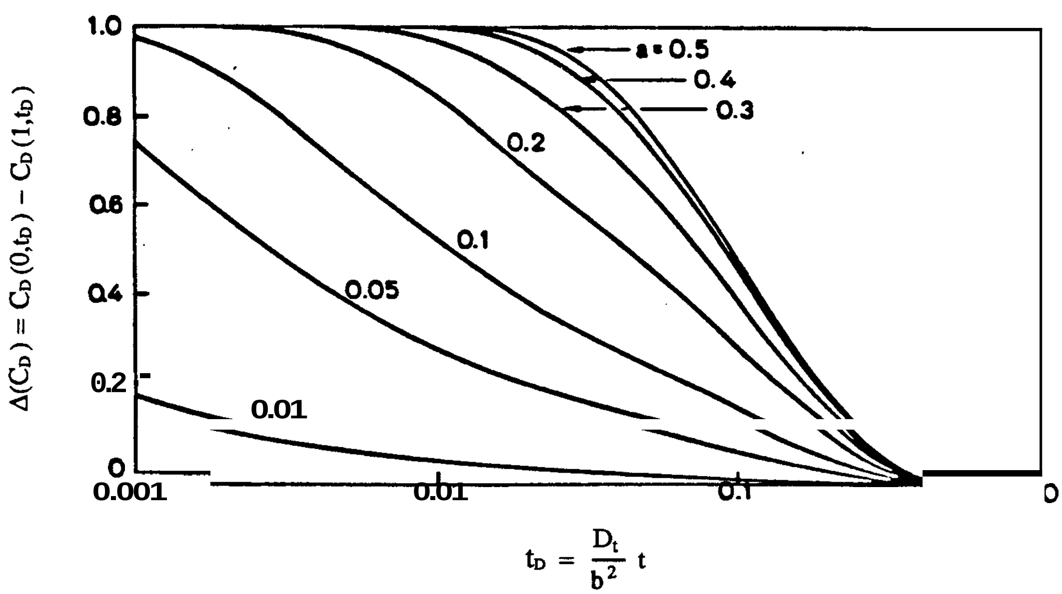


Fig. 2: Time To Equalization For Various Slug Widths.

Solving **Eq. 3** for t gives an expression for the real flowing time it takes for a tracer concentration gradient to become equalized across the fracture width. **In** field'cases this value typically translates **to** distances of order **thirty** meters; in the case of a laboratory bench model it will **be** shown that the distance may be as small as eight centimeters.

Net Longitudinal Dispersivity For Fracture Flow

Taylor (1953) derived an expression for net longitudinal dispersivity for pipe flow, in which the tracer front moves with the mean flow velocity. Using a similar derivation *Horne and Rodriguez* (1983) presented the equivalent expression for fracture flow:

$$\eta = \frac{2}{105}\,\frac{b^2u^2}{D}\,. \tag{4}$$

Eq. 4 gives an expression for predicting η or more importantly, Eq. 4 may be rearranged to solve for b, the fracture half-width, if η is known or calculable. In field examples effective Peclet numbers are of order 150 (*Horne and Rodriguez,* 1983); in a laboratory bench model it will be shown that effective Peclet numbers are of order 200.

Applicability Of The Mathematical Model

Eq. 2 is the exact solution to Eq. 1 for an instantaneous step input. In a series of graphical representations designed to demonstrate the applicability and the limitations of particular approximations *Maloszewski and Zuber* (1985) report that in short-term experiments in a high matrix porosity, diffusion is dominant and dispersion may be neglected. In a low matrix porosity, dispersion only yielded approximately the mean transit time of water. By contrast, in long-term experiments the tracer appeared to exploit the total water volume and the mean transit time of the tracer was extended beyond the mean transit time of water by the retardation factor, given approximately by:

$$\text{retardation factor} = \frac{\text{mobile fissure volume} + \text{stagnant micropore volume}}{\text{mobile fissure volume}}\,, \tag{5}$$

so long as the tracer was non-adsorbable; for adsorbable tracers, the micropore volume acted as a strong sink.

In reporting on the experimental and theoretical determination of matrix diffusion and solute transport properties at the Nevada Test Site, *Walter* (1982) names molecular diffusion as the most important physico-chemical process affecting the degree of nonconvective transport into the matrix porosity, followed by the groundwater composition and the degree of constrictivity and tortuosity of the matrix.

Section 3: SIZING THE LABORATORY MODEL

Recalling **Eq.** 3 and substituting $t_D = 0.5$ and $b = 0.25\ mm\ (8.202 \times 10^{-4}\ ft)$, then if we use iodide tracer $(D = 2.065 \times 10^{-5}\ cm^2/s\ [= 2.223 \times 10^{-8}\ ft^2/s])$, we can see that the time to equalization of the tracer concentration gradient across the fracture is 15.133 seconds. Now if we make the apparatus 30.48 *cm* (1.0 *ft*) wide (cross-sectional area = 1.524 cm^2 $[= 1.640 \times 10^{-3}\ ft^2)$ and use a volumetric flow rate of 50 cm^3/s, the distance to equalization of any concentration gradient is only 8.202 *cm* (0.269 *ft*). Thus an apparatus 182.88 *cm* (6.0 *ft*) long fits conveniently on a laboratory bench.

Peclet Number In The Laboratory Model

Recall the Peclet number:

$$N_{Pe} = \frac{uL}{\eta}\ , \qquad (6)$$

where

u	=	mean flow velocity
L	=	fracture length
η	=	net longitudinal dispersivity.

Calculating values for the variables in **Eq.** 6 from the quantities proposed in the discussion above and using a value of $\eta = 0.5\ cm^2/s\ (5.382 \times 10^{-4} ft^2/s$, a value typically calculated from laboratory data) we can see that Peclet numbers in the laboratory model are of order 200. ***Horne and Rodriguez*** (1983) have shown that under the mechanism of Taylor dispersion, effective Peclet numbers in a field-scale fracture are of order 150.

Section 4: DESIGN OF THE LABORATORY MODEL

The apparatus used in the experiments is a Hele-Shaw cell 182.88 *cm* (6.0 *ft*) long and 30.48 *cm* (1.0 *ft*) wide (Fig. 3 *[Gilardi, 1984]*). It was designed by *Gilardi* (1984) and is constructed of a surface ground, polished, anodized aluminum plate into which have been press-fit three parallel rows of thirty-two electrodes for a total of ninety-six electrodes. The electrodes (Fig. 4 [Gilardi, 19841) are gold-plated; the outer poles are grounded to the aluminum plate.

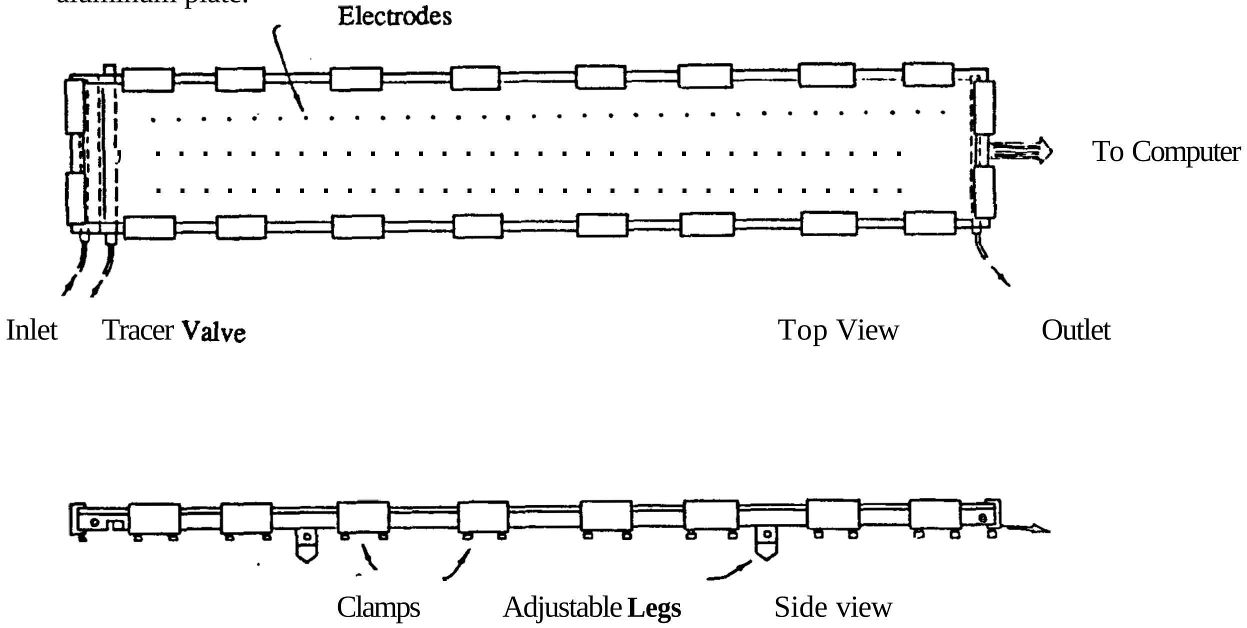


Fig. 3: Hele-Shaw Cell.

The inner poles of the electrodes are wired to wire looms which terminate in plugs at resistor-bank circuit boards. The aluminum plate is grounded through the wire looms to pins in the plugs; in addition the aluminum plate is grounded to earth through water system piping in the laboratory.

A slot cut the width of the fracture (Fig. *5 [Gilardi,* 19841) insures that the base concentration fluid is flowed uniformly across the fracture. The injected fluid is flowed through a second slot from a valve arrangement (Fig. *6 [Gilardi,* 19841) which also insures

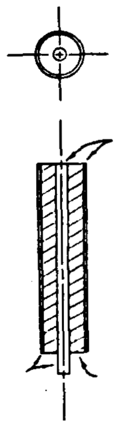


Fig. 4: Electrode Cross-section.

uniform flow across the fracture. The design of the valve is such that it is either on or off, thereby approximating a step input as closely as possible.

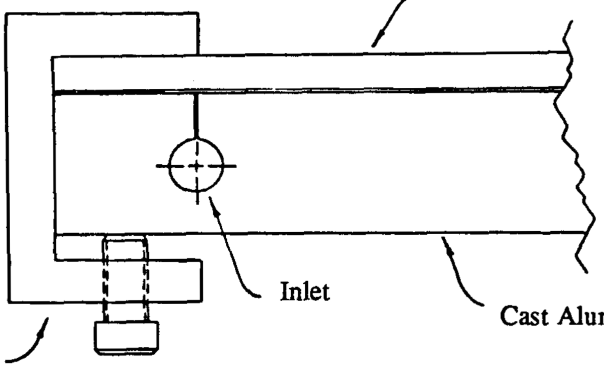


Fig. 5: Detail Of Inlet.

Both the base- and injected-concentration fluids are stored in reservoirs which maintain a constant pressure through a tube arrangement (Fig. 7 [*Gilardi*, 19841). Because the reservoir system is capable of maintaining a constant pressure it is not necessary to keep the reservoir levels the same; nor is it necessary to adapt to the changing head which would result if the pressures were not kept constant.

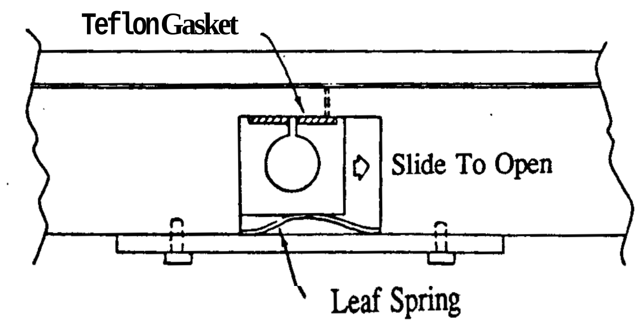


Fig. 6: Detail Of Tracer Valve.

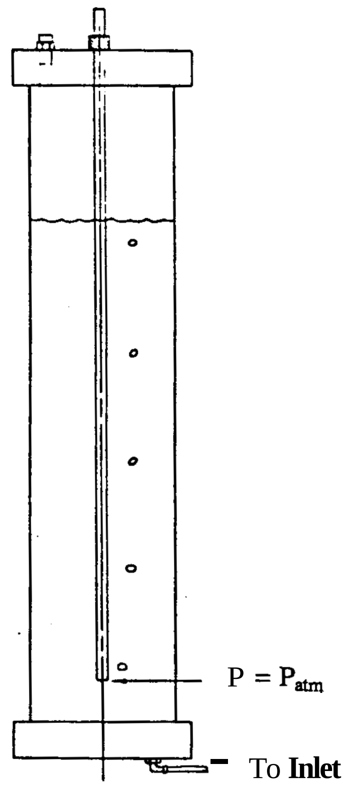


Fig. 7: Constant-pressure Reservoir Design.

Section 5: DATA ACQUISITION HARDWARE

The advance of the tracer front in the Hele-Shaw cell is monitored at each of the cell's ninety-six electrodes as a voltage by a KEITHLEY/das Series 500 Measurement And Control System and ancillary hardware (Fig. **8).** The Series 500 system comprises a microcomputer interface card and connecting cable and a box external to the microcomputer and in which are a self-contained, multiple-range power supply **and** ten slots for a variety of circuit boards or cards. The cards used in the current configuration are an Analog Measurement Module (AMM1), an Analog Output Module (AOM1) and three Analog Input Modules (AIM3's), all manufactured by KEITHLEY/das.

The AMMl card performs the functions of global analog signal conditioning through a programmable gain amplifier (PGA), switching, and 12-bit analog-to-digital (A/D) conversion. The AOMl card provides high-speed analog output through its 12-bit digital-to-analog (D/A) converter over a number of voltage ranges. Sample-and-hold time of both the A/D and **D/A** converters is typically 20 to 25 μs and acquisition time is typically **3** μs. Each AIM3 card offers thirty-two channels of single-ended input and provides high-speed multiplexing and gain amplification. Selection of the proper slot and channel on all cards within the Series 500 box is software-controlled, which makes configuring the system a software function rather than a hardware one.

External to the Series 500 box is a trio of circuit boards, one for each row of thirty-two electrodes. Each board takes a voltage input from the AOMl card and distributes it in parallel through thirty-two 100-kΩ resistors to its bank of thirty-two electrodes. Plugs in the boards allow cables to and from the Series 500 box and to the Hele-Shaw cell to be disconnected easily if necessary. The three external circuit boards were constructed expressly for these experiments. Circuits were made by wire-wrapping on gold-plated pins.

Central to the data acquisition system is a COMPAQ™ personal computer with an internal, 360-kilobyte electronic disk emulator. The internal "disk" stores raw voltage data during an experimental run; since writing to a physical floppy disk does not occur during an actual run the data transfer rate is faster than it would be otherwise. Data are not transfered to a floppy disk for backup **and** storage until a run has been terminated.

Figure **8** illustrates the bridge circuit used to acquire data as a voltage from the electrodes. Each electrode has it own dedicated circuit and input channel. Voltage is applied in parallel to all ninety-six electrodes simultaneously; the grounds are common as well.

The 100-kΩ resistor has about the same impedance as the electrode; therefore about half the voltage drop **occurs** across the resistor and half across the electrode. **As** the conductivity of the tracer solution drops, the voltage drop across the electrode increases.

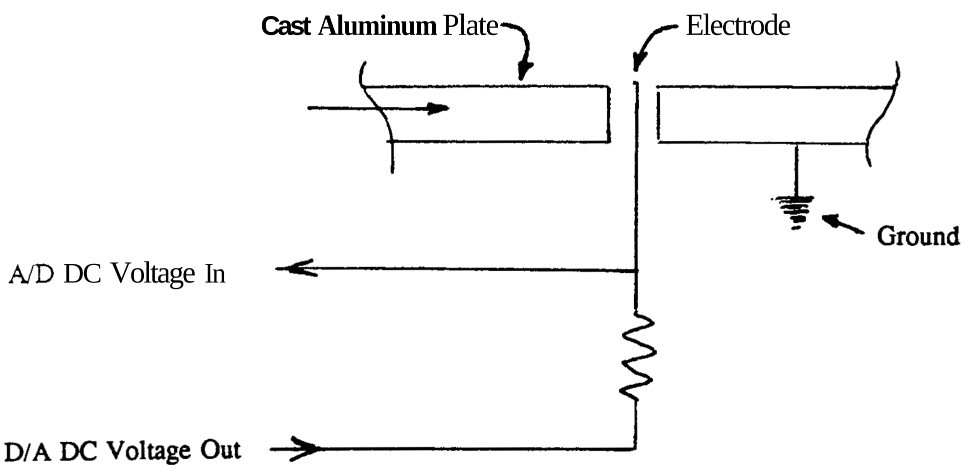


Fig. **8:** Schematic Drawing **Of** Electrode Circuit

Section 6: SOFTWARE FOR DATA ACQUISITION AND ANALYSIS

Data Acquisition Software

It was necessary to develop and test a completely new software driver for data acquisition since the KEITHLEY/das Series 500 Data Acquisition hardware was newly-acquired and the laboratory apparatus was going to be used in its full, 96-electrode configuration. Because of the new data-acquisition hardware, development of the SCAN500 program suite became of primary importance and was therefore addressed first. The reference manual for software development was the reference section of the KEITHLEY/das ***Series*** 500 *Measurement* ***And Control System*** (1984) manual. A suite of programs was developed in compiled Microsoft BASIC™ for use on the COMPAQ™ personal computer. The programs comprise the following:

Table 1. SCAN500 Data Acquisition Software

Program Name	Description
SCAN500	Prints graphic logon; calls system menu
HEAD500	Prints system menu; calls selected program
	Maintains overall program control
DATA500	Configures hardware; drives data acquisition
FIX500	Converts raw data; groups results by electrode
GRAPH500	Graphically displays electrode response versus time
TEST500	Tests individual electrode performance
	Provides for a test electrode
	Provides drivers for hardware calibration

SCAN500 Graphic Logon

The ***SCAN500*** Data Acquisition **Software** System is completely interactive and menu-driven. It was designed in **this** manner ***so*** that it would be easy to operate without extensive knowledge of the code used to generate the programs. SCAN500 is a graphic logon which identifies the program suite and chains HEADSOO, the system menu.

HEAD500 System Menu

HEAD500 is the overall program controller and system main menu. When one program terminates it calls the menu, allowing for continuing analysis of data or a program restart. Any of the programs may be called from the menu or control may be returned to the operating system.

DATA500 Data Acquisition Program

Data acquisition using the KEITHLEY/das Series 500 Data Acquisition System is performed by DATASOO. The program may be toggled to operate in demonstration mode to illustrate the mechanics of data acquisition using the KEITHLEY/das Series 500 hardware and to demonstrate the screen graphic display monitoring and time-delay features. In demonstration mode a dummy dataset is produced using a random number generator which allows the user to continue through the suite of programs, reorganizing and analyzing data and displaying it on the screen.

The primary function of DATA500 is of course acquisition of data from the Hele-Shaw cell for which the program was designed. Initialization of the program includes laboratory conditions which **are** included in a header file unique to each experimental run. Options for data acquisition include all three **rows** of electrodes **(96** electrodes total) or the center row only **(32** electrodes). The program then prompts for a time limit for the run and a time delay between datasets. The delay feature uses the **trio** of countdown timers on the microcomputer interface card and is accurate to any value within the limits of the ability of the software driver **to** scan the timers (on the order of µs). DATASOO then prompts for graphic display and if **this** option is selected, further prompts for the frequency of display.

Once DATASOO has been initialized the program waits for the user to toggle the beginning of data acquisition. At any time after the run has begun and before it has timed out the user may terminate the run with single-stroke function key input. If the graphic display option has been selected another single-stroke function key allows the user to change the frequency of graphic display. This is an important function since screen graphic display requires reconstituting raw data and drawing it on the screen, all of which requires time and therefore reduces the number of datasets obtainable during the run. While in some cases **this** may not pose a serious problem, in other cases it may be desirable to obtain as much data as possible, for example when voltages are changing rapidly.

Once a run has either timed out or been terminated DATA500 prompts for disposition of the data; in any case the header information is added to a permanent log. Raw data which has been stored on the microcomputer's internal electronic disk may be backed up on a floppy disk at **this** time; however it is possible to proceed directly to data reconstituting and sorting, or there is a final escape key which may be used to cancel the run entirely if there is a reason not to save the data, after which control is returned to the program menu.

FIX500 Data Reconstitution And Sorting Program

The **FIX500** program has as its dual purposes **the** reconstitution of raw data from the datafile SCAN500.DAT created on the microcomputer's internal electronic disk during data acquisition and its reorganization into **a** voltage-response-versus-time format by electrode. Since the SCAN500.DAT raw datafile may be large, FIX500 does its work in chunks, the size of each of which is manageable by the microcomputer's random-access memory (RAM) limitation of **64** kilobytes. FIXSOO then writes these chunks to temporary files on a designated disk drive. When the entire set of raw **data** has been read in this manner FIX500 creates its output file, FIX500.DAT, from the temporary chunks and finally erases the temporary files.

Although the original datafile SCAN500.DAT was written simply **as** a sequence of times and high and low bytes of information, FIX500.DAT is actually smaller than SCAN500.DAT; this is because FIX500.DAT writes the run elapsed time to each dataset only once, at the beginning of the file. FIX500.DAT is typically about 70% the size of SCAN500.DAT. This is convenient for two **reasons:** first, it guarantees that FIX500.DAT will fit on any device on which SCAN500.DAT was created; and second, it allows subsequent graphic display and curve-fitting analysis programs to access a smaller database, thereby speeding the time of execution of the more time-consuming downstream programs.

FIX500 prompts for the drive designations for both the output datafile and the temporary files; it finds the SCAN500.DAT datafile and SCAN500.HDR run header files itself. Errors either correct themselves or prompt for remedial action, thereby allowing for the smooth flow of program execution. In the case that program execution must be terminated a single-stroke function key closes and erases unnecessary files and returns control to the program menu. Finally, **on** completion of its functions, **FIX500** prompts for GRAPH500, the screen graphic display program which is the first step toward analysis of the data acquired by DATASOO.

GRAPH500 Optimizing Graphic Display Program

The datafile created by FIX500, **FIX500.DAT,** contains voltage-versus-time information for each electrode. GRAPH500 offers three display options: automatic, time-delayed sequencing of the entire dataset, in which the user may specify the time delay; time-delayed sequencing of the entire dataset, in which the user toggles each dataset; and random-access display of any electrode in the dataset, in which the user specifies the electrode to be viewed. In the latter option display may either be ahead or back in the dataset; the program compensates for both requests.

Data from FIX500.DAT are optimized on the screen; that is, the x- and y-axes are automatically scaled to minimize the voltage and time ranges necessary to display the dataset. This is an important feature since voltage response for each electrode varies considerably. The effect of the automatic scaling is to make the display of the response of each electrode appear at the same scale even if the actual range **is** different.

Unless the automatic time-delay option has been selected, while a particular electrode is being displayed single-stroke function keys allow either or both of the axes to be rescaled, Another single-stroke function key returns the display to its default (maximized) setting. Rescaling the axes is an important diagnostic tool since it allows close scrutiny of any part of the dataset. Time rescaling remains in effect for subsequent datasets unless the maximizing single-stroke function key is used.

If it may be desirable to isolate the dataset for a particular electrode for further analysis another single-stroke function key prompts for a **disk** drive designation and creates and labels a file containing that dataset. This is particularly useful, for example, for taking the derivative of the curve to generate a spike.

Program execution may be terminated at any time by a single-stroke function key and control is returned to the program menu. Error conditions are usually self-correcting; if not, the appropriate error message is displayed briefly and control is returned to the program menu. In any case, no data are lost and files are closed.

TEST500 Electrode Performance, Test Electrode And Hardware Calibration Program

TEST500 was originally written to calibrate the various electronic components on the several cards installed in the KEITHLEY/das Series 500 Data Acquisition System chassis. The primary calibrations necessary were the A/D converter on the AMMl card, and the **D/A** converter and analog output voltage ranges on the AOMl card. Development of the other programs within TEST500 arose out of the needs for test electrode and electrode performance programs.

The calibration programs are expanded versions of program listings in the KEITHLEY/das manual referenced above; however in TEST500 those programs were made interactive and easier to use by adding user input from an on-screen menu into a compiled program. The calibration programs, together with the other programs in the TEST500 suite, were then gathered together under the control **of a** single menu.

The electrode performance program includes electrode charge time and a linearity test between two specified voltage ranges. Since electrode performance may be affected by the fluid with which it comes in contact it is necessary to know the charge time. The linearity test generates an output datafile so that electrode performance, that is, voltage response to a given exciting voltage, may be seen graphically.

Analysis Software

All data acquisition and preliminary analysis programs are under the control of the SCAN500 program suite; however the critical link between the analytical model and the experimental model **is** the ability to analyze the raw data according to the analytical model. For this reason a suite of programs previously written was used for downstream data analysis.

Table 2. FIT500 Data Analysis Software

Program Name	Description
FIT500	Fits voltage raw data
FIT600	Fits the derivative of voltage raw data
PLOTONE	Graphs a single fitted dataset
PLOTALL	Graphs all fitted datasets in sequence

All of the data analysis programs in the FIT500 suite were used in one version or another by Gilardi **(1984).** They were written by **R.** N. Home. All of the programs have been modified, both by the author and by Horne, for their present use.

FIT500 is a driver written to make use of the curve-fitting program VARPRO ***(Stanford University,*** **1973).** VARPRO fits Eq. 2, the complimentary error function solution to Eq. 1, to experimental data from the FIX500.DAT dataset. Linear parameters fitted are C_0 and C_1, the base and injected tracer concentrations; non-linear parameters fitted are u, the velocity and η, the dispersivity. **FIT500** creates individual data output files for each electrode fitted as well as a summary output file and velocity-versus-electrode and **dispersivity-versus-electrode** datafiles. All of the output files are used for analysis and examples are found in the Results section of **this** paper.

FIT600 is a modification of FIT500 which fits the derivative of concentration-versus-time data from the original FIX500.DAT dataset; otherwise it functions in exactly the same way as FIT500.

Output datafiles obtained from either FIT500 or FIT600 may be graphed on the screen of a graphic display terminal either by PLOTONE or by PLOTALL. Diagnostically these two programs are used to verify the coincidence of the concentration-versus-time data and its fit for

a particular electrode. It is **necessary** to eliminate from the complete dataset those electrodes for which the fit **is** not acceptable. Excessive electronic noise, voltage **drift** or an apparent open circuit are possible causes of a bad electrode dataset. **PLOTALL** makes use of a single-stroke function key which removes **a** given electrode dataset from the active file while reserving it for further study.

Section 7: EXPERIMENTAL PROCEDURE

Preparation Of The Experimental Apparatus

In order for it to fill properly the Hele-Shaw cell must be thoroughly cleaned and dried. Cleaning necessitates removal of the clamps and the glass top plate. The glass plate is most easily cleaned with a static-free cloth and acetone on a flat laboratory bench where it is completely supported. If the glass is stored flat it will stay flat for the duration of an experimental run. The constant-pressure tracer reservoirs must be clean and dry as well; they will dry if they are drained following a run.

The apparatus is assembled by first applying a very thin film of silicone stopcock grease to the shim and then laying the glass plate on the shim. The purpose of the grease is to help provide a tight seal between the glass plate and the shim. Once the glass plate is in place the clamps must be positioned and finger-tightened. If this procedure is followed carefully the Hele-Shaw cell will not leak

For the current experiments the Hele-Shaw cell was assembled in one of two configurations: open cell, in which nothing obstructed the flow of tracer down the fracture; and mixing cell, in which a nylon mesh fabric with a regular, 1.5 x 1.5 *mm* grid was laid down within the cell volume, providing a means of inducing transverse mixing as well as providing dead pore volumes within the fracture.

Preparation Of The Tracer Solutions

Tracer solutions of various concentrations are prepared from standard solutions according to the ratio

$$\text{tracer standard} = \frac{0.1 \text{ moles tracer}}{\text{liter distilled water}} . \tag{7}$$

From the standard solution a particular solution of a given concentration may be prepared according to the relation

$$\frac{\text{liters tracer standard}}{\text{liter of solution}} = \frac{\text{concentration (ppm)}}{0.1 \text{ gram–moles tracer}} \; . \tag{8}$$

For iodide standard Eq. 8 may be written

$$\frac{\text{liters iodide standard}}{\text{liter of solution}} = \frac{\text{concentration (ppm)}}{12.690 \text{ grams iodide}} \; ; \tag{9}$$

while for chloride standard it is written

$$\frac{\text{liters chloride standard}}{\text{liter of solution}} = \frac{\text{concentration (ppm)}}{3.5453 \text{ grams chloride}} \; . \tag{10}$$

Charging the Hele-Shaw Cell

Following preparation of the two tracer solutions the constant-pressure reservoirs are filled. All flowlines must be filled as well. The Hele-Shaw cell is then blown out with CO_2 and the flowlines are attached. Prior to filling the Hele-Shaw cell the tracer valve must also be filled with fluid.

The base concentration solution is then flowed slowly into the Hele-Shaw cell. There is a tendency for air bubbles to form on top of the electrodes due to their very slight roughness compared with the surfaces of the cell; however no air bubbles will form if the glass is tapped very gently with a soft rubber hammer as the liquid front passes each electrode. In this manner the entire cell is filled.

Data Acquisition

At this point the KEITHLEY/das box and the microcomputer are turned on. The SCAN500 program diskette is placed in the "A" drive and the SCAN500.LOG diskette is placed in the "B" drive. The "Scan" option is selected from the program main menu; once all the initializing requested by the program has been completed, data acquisition is ready to begin. Data acquisition begins by pressing the **RETURN** key on the microcomputer.

It is possible to run the experiment in several ways. In the equations above the variable r refers to the real time from the beginning of the continuous step injection; however $t = 0$ occurs at some time t_0, the duration of time for which the base concentration fluid has been flowing. If it is desired that t_0 be greater than zero, then the tracer valve is opened after a suitable delay; otherwise it may be opened immediately. For the current experiments t_0 was usually set to zero.

Several experimental runs performed with the graphic display option demonstrated that at the lowest range of flowrates obtainable the injected tracer front had passed the end of the Hele-Shaw cell after no more than six to seven minutes had elaped. Several additional runs performed without the graphic display option produced datasets with more than triple the density of the earlier runs. It was learned that these higher-density runs provided better resolution, especially during the time the injected tracer front was passing a particular electrode; subsequently, most of the runs were performed without the graphic display option.

Cleanup And Storage Of The Apparatus

Disposition of the data acquired during an experimental run has been explained in the section describing software. Following data backup, if another run is not to be performed immediately it is good practice to disconnect the flowlines and drain the constant-pressure reservoirs. The Hele-Shaw cell should then be blown out with CO_2, the glass removed, cleaned and stored and the cell cleaned, dried and covered.

It should be pointed out that the injected tracer solution may be higher or lower than the base solution. This means that it is possible to perform any number of experimental runs in sequence without dismantling the Hele-Shaw cell and without flushing the cell down to the lower concentration. In practice, two runs were usually performed together, since the constant-pressure reservoirs hold enough fluid for two runs without refilling.

Section 8: ANALYSIS PROCEDURE

Part 1: All Datasets

Part of the process of analyzing data from an experimental run has been described in the software section of this report. Analysis begins by running FIX500, the program which takes the raw data written as time and voltage versus electrode number and rewrites it to a new file, FIX500.DAT, as voltage versus time for each electrode in sequence.

Next, FIT500 first converts the voltages to Concentrations, then fits Eq. 2 to the concentration response curve. FIT600 performs the same conversion; however, it then differentiates the data and fits the derivative of Eq. 2 to the result. Meanwhile, GRAPH500 allows the FIX500.DAT file to be viewed graphically. This viewing is an important step, since during no experimental run do all the electrodes behave perfectly. Data transmission errors, excessive drift, and an occasional "open circuit," caused by an air bubble being trapped on the electrode, may be detected. Any electrodes which do not conform to a minimum data transmission standard must be discarded from the dataset.

After FIT500 and FIT600 have been run, the results are viewed using one of the graphic display programs, PLOTALL or PLOTONE. As before with GRAPHSOO, viewing the fits is necessary in order to reject datasets which do not meet a minimum standard for fitting. It is necessary that there not be excessive drift in the original data, which is the primary reason for rejecting a dataset at this point; however any datasets for which the match is not close must be discarded as unuseable. In most cases the same datasets which were rejected using GRAPHSOO are rejected here as well.

Using the remaining datasets, typically about one-half to two-thirds of the total, a comparison of the numerical values of the dispersivity, η, between matched to the step function (FITSOO) and its derivative (FIT600), typically screens several more electrodes. Here some comparison criterion must be applied; typically 10% is used.

Part 2: Using The Numerical Results

Confirming Eq. 4

From the final dataset an average value of velocity is calculated using the statistical mean, along with the standard deviation of the velocity. These values are then used to calculate the range of the dispersivity, q, from Eq. 4. This value appears on the graphs of the results, as in Figs. 8 and 9.

Investigating The Effects Of Transverse Mixing In The Hele-Shaw Cell

The same screening and calculation techniques are applied to the datasets from the runs in which the mesh was included in the fracture aperture. Here it is desired to contrast the results with those from runs in which the fracture aperture was clear of the mesh. The effects of transverse mixing and/or the creation of dead pore volumes would then be made manifest in any changes in the apparent diffusion coefficient calculated from Eq. 4.

Section 9: RESULTS: TAYLOR DISPERSION MODEL

Typical of the results obtained from runs in which the Hele-Shaw cell was clear of the mesh are those given as Table 3 from Run 16. Here the truncated dataset is given so that the useable range of values obtained may be seen. Table 4 gives the results from the same dataset for the derivative analysis. Appendix A contains a full set of the function matches; Appendix B contains a full set of the function derivative matches. The analysis was performed according to the procedure described above.

Results of the analysis and comparison with the values calculated from Eq. 4 are given as Fig. 9. Eq. 4 predicts that $\eta = 0.4103\ cm^2/s$; the mean value obtained from the matches is $\eta = 0.3824\ cm^2/s$. These two values are well within one standard deviation of each other, since the data are scattered.

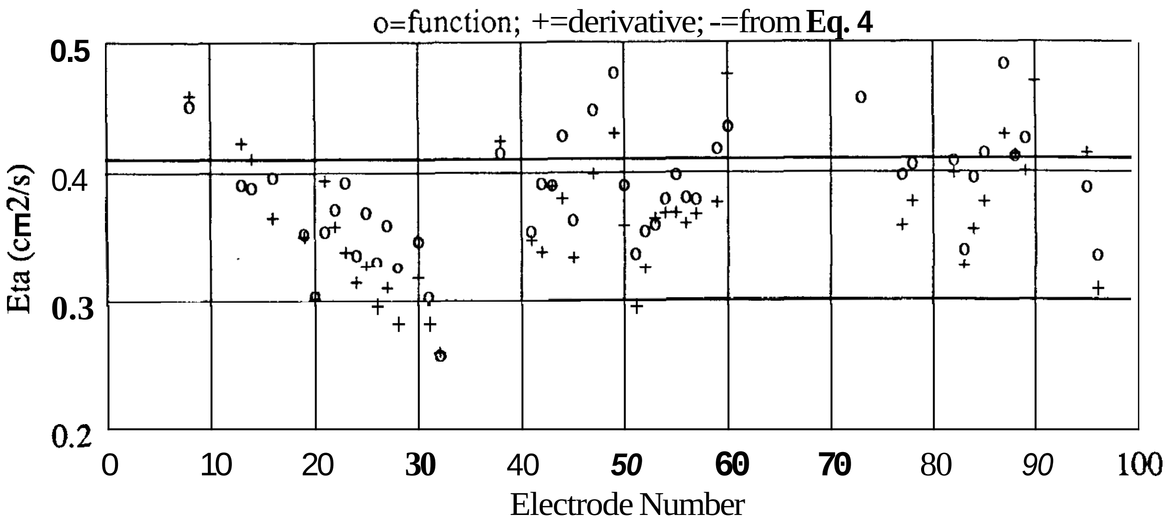


Fig. 9: Results: Run 16 Dispersivity

Table 3. Results: Run 16 Function Match

Electrode Number	C_0, *ppm*	C_1, *ppm*	Velocity, *cm/s*	η, *cm²/s*	t_o, *s*
8	301.4585	99.7838	0.6866	0.4504	-0.5717
13	301.4375	100.0154	0.7047	0.3904	-1.0004
14	301.4592	100.0227	0.6895	0.3879	-0.3701
16	301.9744	99.7453	0.6721	0.3961	-0.1856
19	300.6046	1005613	0.6881	0.3528	1.0154
20	299.8596	101.0646	0.6680	0.3037	-0.3160
21	302.1015	100.0861	0.6934	0.3542	-0.2161
22	302.1749	100.0796	0.6949	0.3711	-0.6864
23	301.8146	100.2608	0.7005	0.3918	-0.0433
24	301.5788	100.2698	0.6976	0.3362	-0.3316
25	301.0849	100.6161	0.6983	0.3686	-0.7350
26	303.9118	99.7314	0.7008	0.3307	-0.7789
27	302.5794	100.0868	0.6954	0.3591	-0.5780
28	301.7254	100.3039	0.6915	0.3261	-0.0507
30	302.8628	100.3007	0.7059	0.3468	-0.8829
31	300.9701	100.6014	0.6964	0.3024	-0.9311
32	301.6063	100.3390	0.6829	0.2573	-0.1706
38	3002421	100.4424	0.6844	0.4145	-0.2420
41	300.1323	100.6178	0.6666	0.3543	-0.0365
42	299.9247	100.8680	0.6658	0.3907	-1.2781
43	300.5925	100.3761	0.6815	0.3895	-0.1421
44	300.5379	100.4582	0.6824	0.4277	-0.9796
45	300.5592	100.4752	0.6681	0.3629	-0.4056
47	300.6788	100.6247	0.6745	0.4471	-1.2631
49	302.3793	99.7439	0.6756	0.4755	-0.5198
50	301.0013	100.4126	0.6679	0.3891	-0.6920
51	300.1586	100.9608	0.6630	0.3366	-0.3787
52	300.4196	100.9523	0.6523	0.3540	-0.4450
53	301.6875	100.0338	0.6607	0.3591	-0.7702
54	300.4023	101.1088	0.6567	0.3789	-0.5098
55	300.9331	100.7822	0.6613	0.3974	-0.3967
56	300.6924	101.0008	0.6544	0.3797	-0.9229
57	301.0000	100.7868	0.6526	0.3781	-0.1659
59	302.7237	99.9000	0.6631	0.4172	-0.5110
60	301.5180	100.7503	0.6633	0.4346	-0.3100
64	301.1286	102.5596	0.6250	0.5928	-0.0133
73	300.3258	100.3682	0.7963	0.4561	-0.7272
77	299.3334	100.8362	0.7573	0.3966	-0.0070
78	300.7867	100.2841	0.7843	0.4052	-0.4521
82	301.2524	100.3085	0.7791	0.4077	-1.0826
83	300.0170	100.5247	0.7567	0.3389	-1.2880
84	299.5644	100.6431	0.7604	0.3946	-0.7872
85	300.7395	100.4554	0.7713	0.4142	-0.1323
87	301.7392	100.4260	0.7777	0.4823	-0.2606
88	303.5022	100.2358	0.7708	0.4114	-0.5526
89	302.1294	100.3910	0.7710	0.4248	-0.6067
90	302.9916	100.3839	0.7769	0.5025	-0.6085
95	307.3179	100.1716	0.7584	0.3873	-0.7452
96	301.5720	100.5215	0.6990	0.3349	-0.8319

Table 4. Results: Run 16 Derivative Match

Electrode Number	C_0, ppm	C_1, ppm	Velocity, cm/s	η, cm^2/s	t_o, s
8	-0.0229	-200.4117	0.6801	0.4638	-2.5817
13	0.0467	-207.3111	0.7028	0.4276	-1.2963
14	0.0355	-205.2367	0.6874	0.4153	-2.0341
16	-0.0956	-191.6559	0.6681	0.3703	-2.2985
19	0.0511	-203.0274	0.6887	0.3557	-0.8903
20	0.0168	-200.6927	0.6706	0.3055	-0.9628
21	0.0506	-209.4178	0.6929	0.3983	-1.6435
22	-0.0980	-194.9319	0.6940	0.3630	-0.9198
23	-0.2402	-179.5442	0.7004	0.3435	-1.3942
24	-0.0778	-193.3969	0.6972	0.3196	-1.7100
25	-0.0974	-189.6121	0.6994	0.3326	-1.7149
26	-0.3249	-177.5298	0.6988	0.2971	-1.3841
27	-0.1648	-182.0753	0.6937	0.3159	-1.0991
28	-0.1436	-184.9567	0.6909	0.2838	-1.5582
30	-0.1788	-185.9182	0.7066	0.3234	-0.6521
31	-0.0053	-199.1477	0.6972	0.2861	-0.6685
32	0.0144	-202.3221	0.6829	0.2650	-1.4611
38	-0.0267	-199.4611	0.6873	0.4290	-1.2947
41	-0.0452	-196.0579	0.6695	0.3525	-0.9985
42	-0.0904	-187.9744	0.6685	0.3436	-0.9029
43	-0.0142	-198.9020	0.6822	0.3938	-1.3538
44	-0.0589	-190.6589	0.6818	0.3844	-1.0641
45	-0.0838	-190.6824	0.6688	0.3394	-1.5970
47	-0.1167	-187.3702	0.6754	0.4033	-2.3811
49	-0.1296	-188.0205	0.6718	0.4343	-1.3258
50	0.0303	-197.3533	0.6664	0.3638	-1.0267
51	-0.0748	-188.2832	0.6650	0.2973	-1.2129
52	-0.0922	-190.2663	0.6536	0.3298	-1.4323
53	0.1143	-207.4534	0.6579	0.3692	-1.5174
54	0.0320	-201.2188	0.6574	0.3731	-1.6025
55	-0.0515	-192.5398	0.6615	0.3732	-1.6537
56	-0.0147	-195.6582	0.6547	0.3652	-2.1265
57	0.0217	-199.0190	0.6521	0.3720	-1.6492
59	-0.0733	-189.3000	0.6607	0.3813	-1.3345
60	0.1166	-214.0118	0.6623	0.4798	-1.9743
64	-0.1295	-189.2347	0.6226	0.5966	-3.6868
73	-0.0450	-201.9783	0.8023	0.5038	-1.4808
77	-0.1532	-187.2571	0.7642	0.3630	-1.7054
78	-0.1973	-187.5193	0.7866	0.3815	-1.4492
82	-0.0132	-199.3645	0.7804	0.4033	-0.8607
83	0.0376	-201.4645	0.7592	0.3324	-0.8143
84	0.0072	-195.4469	0.7626	0.3602	-1.6673
85	-0.0376	-194.3870	0.7727	0.3815	-1.1222
87	-0.0946	-191.2638	0.7785	0.4334	-1.3807
88	0.0130	-202.9410	0.7710	0.4179	-1.8310
89	-0.0623	-195.6217	0.7721	0.4047	-1.6591
90	-0.0607	-196.4829	0.7773	0.4741	-1.7965
95	0.0301	-209.8610	0.7588	0.4191	-1.9445
96	-0.0993	-189.5685	0.6996	0.3099	-0.6353

Section 10: RESULTS: MIXING/DIFFUSION MODEL

Typical of the results obtained from runs in which the Hele-Shaw cell contained the mesh are those given as Fig. 10 from Run 17 and Fig. 11 from Run 18. Here it is interesting to note that, because of the scatter in the data, the calculated effective diffusion coefficients range from $1.15 \times 10^{-5}\ cm^2/s$ to $2.42 \times 10^{-5}\ cm^2/s$. The coefficient of molecular diffusion for sodium chloride is $1.484 \times 10^{-5}\ cm^2/s$.

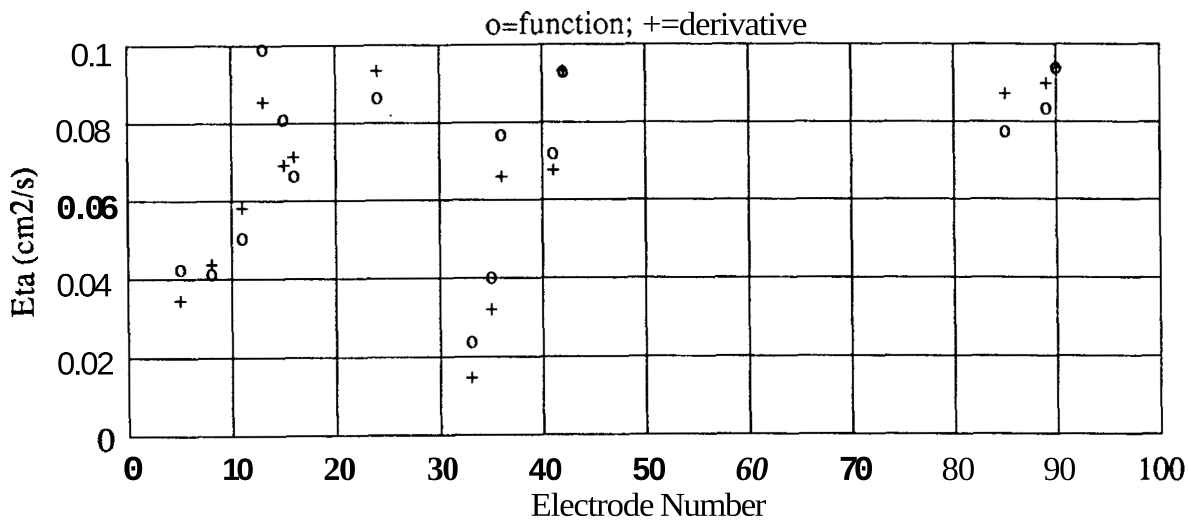


Fig. 10: Results: Run 17 Dispersivity

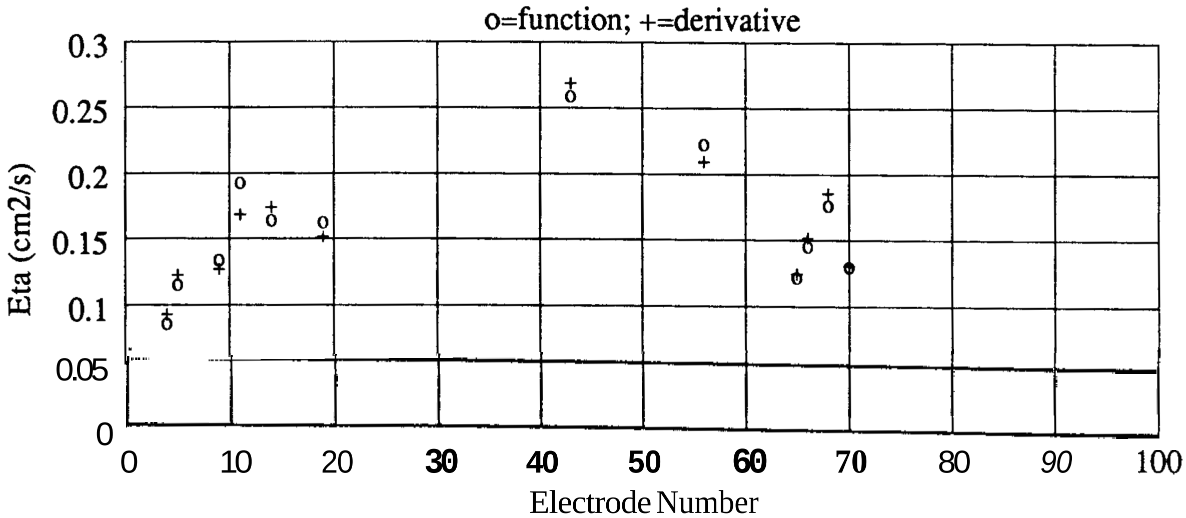


Fig. 11: Results: Run 18 Dispersivity

Section 11: SUMMARY AND CONCLUSIONS

This investigation was initiated to continue that of *Gilardi*'s (1984) into the physical processes which govern the flow of a tracer in a planar fracture. Because new data acquisition hardware had been purchased *this* investigation comprised several parts: data acquisition software had to be developed; the data acquisition system had to be tested against the results of *Gilardi* (1984); and the investigation was broadened to include observing the effects of transverse mixing and the presence of closed pore volumes in the fracture.

To simulate the presence of closed pore volumes in the fracture and to effect transverse mixing a fine nylon mesh was placed within the fracture volume. Comparing data obtained from several runs performed both with and without the mesh showed that analysis was more difficult using data from runs in which the mesh was present since the data were uniformly more noisy. The subroutine VARPRO was able in most cases to curvefit either Eq. 2 or its derivative to the data; however, because of the associated noise, confidence intervals were broader and standard deviations were greater. Still it is possible to draw some conclusions from what was observed.

Table 5 summarizes a portion of the results from Run 16, performed without the mesh in the cell, and Runs 17 and 18, performed with the mesh in the cell.

Table 5. Comparison Of Results

Run Number	Number of datasets	$\bar{u}_{fitted}$, *cm/s*	$\bar{\eta}_{fitted}$, *cm²/s*	$\eta_{calculated}$, *cm²/s*	$D_{calculated}$, 10^5 *cm²/s*	N_{Pe}
16	49	0.7010	0.3824	0.4104	-	312
17	16	0.3861	0.07637		2.418	924
18	12	0.4823	0.1609	-	1.790	548

As may be seen from the table, the values of η, the dispersivity, D, the (apparent) diffusion coefficient, and N_{Pe}, the Peclet number, are all within the same statistical ranges irrespective of the nature of the experiment, namely, whether the mesh was present.

This interesting results suggests that for short-term flow regimes with low matrix porosity, even the fact **of** transverse mixing and the presence of some (low) matrix porosity may not significantly affect the value of effective diffusion coefficients. Thus it may be possible that the simpler, dispersion-only model, that which assumes only Taylor dispersion as the primary flow mechanism of tracer transport in a planar fracture, may adequately describe selected field systems. It is also possible that this simple model may fit adequately into a more general, more complicated model in which effects outside the limits of the fracture's physical boundaries are being investigated, **as** being sufficient to describe flow within the fracture itself.

In general, Eq. **4** tended to predict slightly higher values for η, the dispersivity, than were found experimentally. It should be pointed out again that throughout the investigations the data were noisy, and there were present occasionally other unexplained electrical phenomena such as periodic spikes with frequencies several orders of magnitude less than the frequencies generated by the experiments.

One method which may reduce the noise in the data **is to** take data in a differential mode, that is, to reference the low side of all the electrodes to the low side of a reference electrode while applying a higher frequency, alternating-current **(AC)** signal. The presence of a reference electrode would allow data to be acquired anywhere on the **AC** signal. If there is a capacitance effect from the apparatus itself it would tend to become **minimized** using this technique.

SECTION 12: REFERENCES

Carslaw, H. **S.** and Jaeger, J. C.: ***Conduction Of Heat In Solids,* 2nd** ed., Oxford University Press (1959).

Computer Science Department, Stanford University, Stanford, CA (1973).

Gilardi, J. R.: "Experimental Determination Of The Effective Taylor Dispersivity In **A** Fracture," report SGP-TR-78, Stanford Geothermal Program (1984).

Horne, R. N.: "Geothermal Reinjection Experience In Japan," ***J. Pet. Tech.*** (1982a), **34,** 495.

Horne, R. N.: "Reservoir Engineering Aspects Of Reinjection," ***Japan Geoth. Energy Assn.*** *J.* (1982b), **19,** 23-30.

Horne, R. N.: "Effects Of Water Injection Into Fractured Geothermal Reservoirs - **A** Summary Of Experience Worldwide," ***Fractures In Geothermal Reservoirs,*** Geothermal Resources Council, Davis, CA (1982c).

Horne, R. N. and Rodriguez, F.: "Dispersion In Tracer **Flow** In Fractured Geothermal Systems," ***Geoph. Res. L.*** (April 1983) **10,** 289-92.

Series 500 Measurement And Control System, IBM-H Second Edition (February 1984), Keithley DAS, Boston, MA (1984).

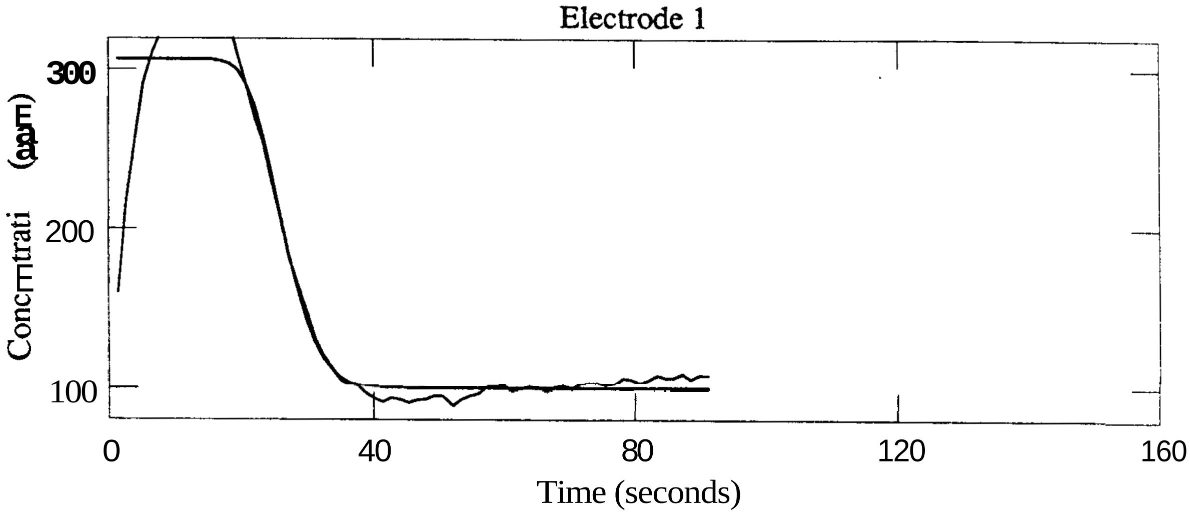
Maloszewski, P. and Zuber, A.: "On **The** Theory Of Tracer Experiments In Fissured Rocks With **A** Porous Matrix", ***J. Hydrol. (Amsterdam)*** (1985), **79** 3-4, 333-58.

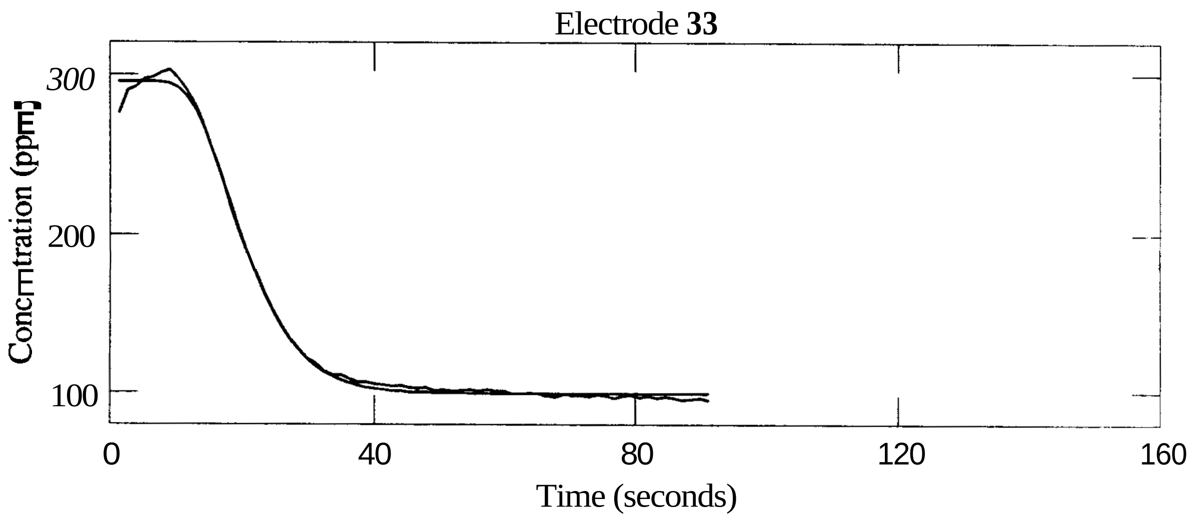
Soc. (1953), A219, 186-203.

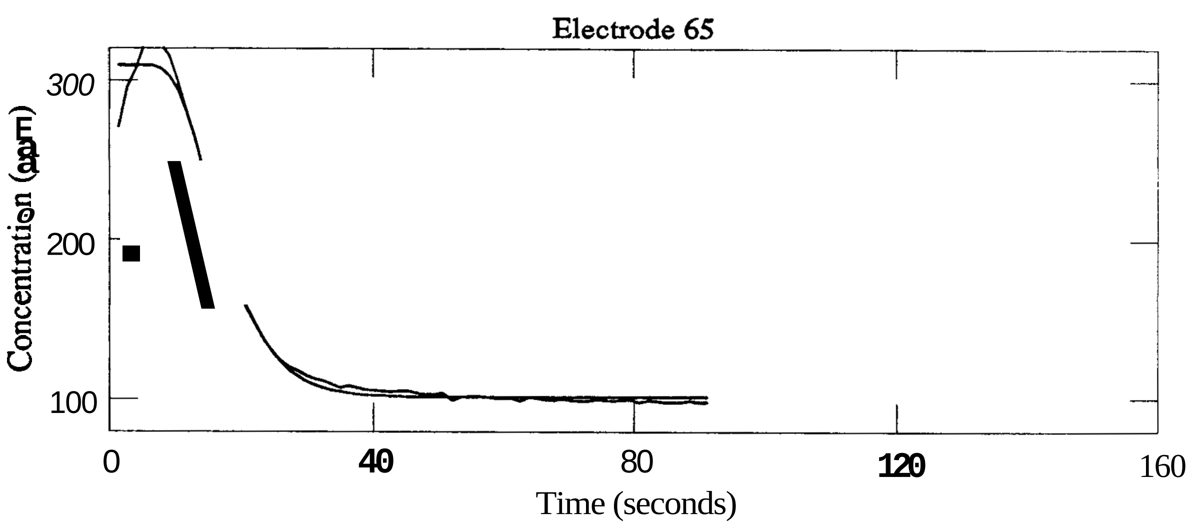
Walter, G. R.: "Theoretical And Experimental Determination Of Mamx Diffusion And Related Solute Transport Properties Of Fractured Tuffs From The Nevada Test Site", report LA-9471-MS, *Energy Res. Abs.* (1983), 8 (5), Abstract No. 10813.

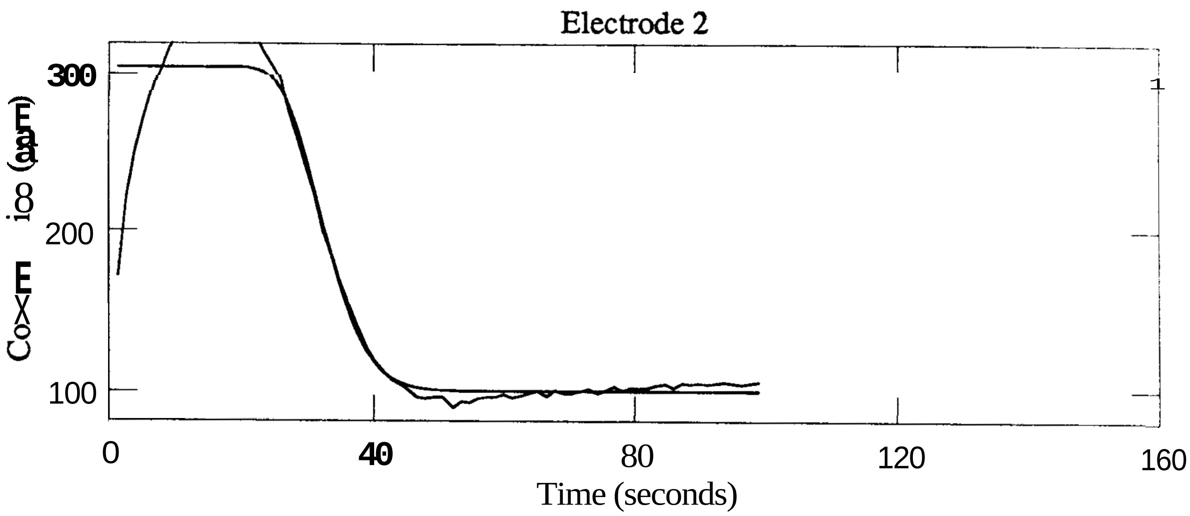
Appendix A: GRAPHS OF RUN 16 FUNCTION FITS

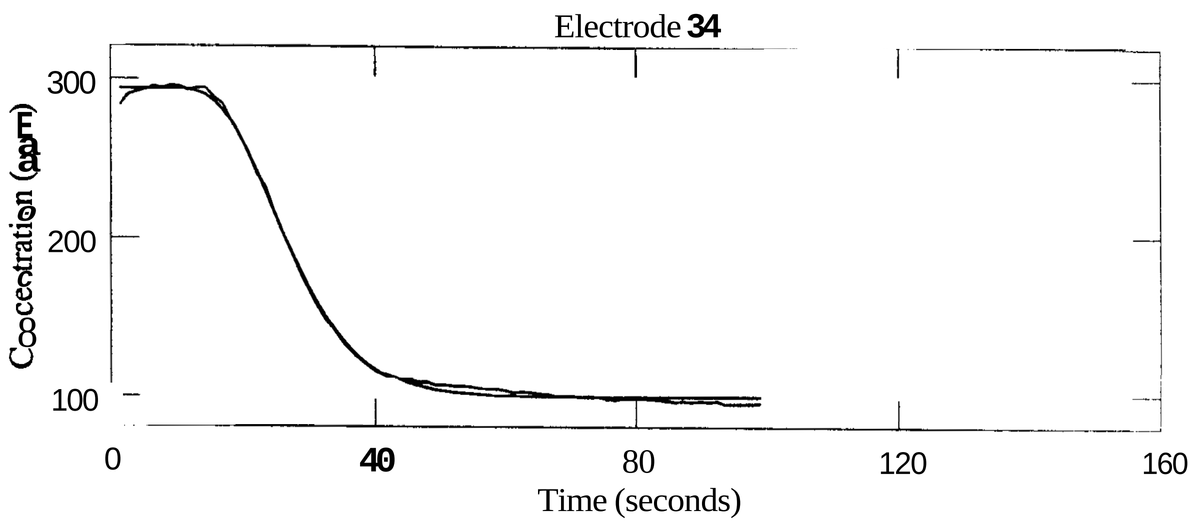
This appendix comprises a complete set of graphs of the matches of **Eq. 2** for Run 16. Each page is in effect a cross-section of the Hele-Shaw cell at a particular distance along the cell from the inlet valve. The electrodes are in three groups numbered 1-32, 33-64, and 65-96 along the cell from the valve. Recall that Table **3** lists the electrodes which were included in the final calculations for this run. Any electrode not included in Table **3** was eliminated from consideration. From the graphs in this appendix it is usually apparent why a particular electrode was not included.

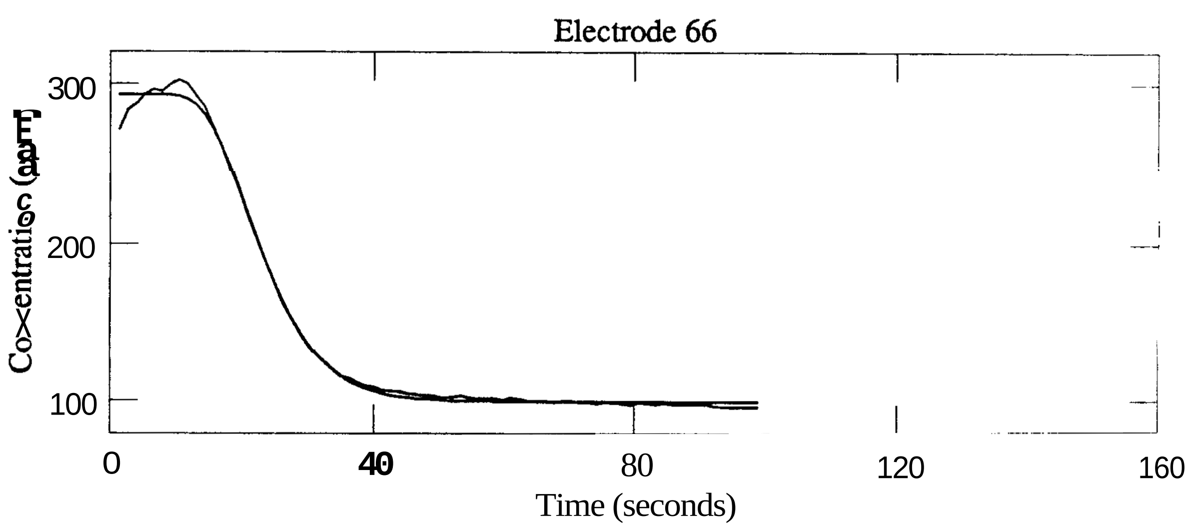

Electrode 1
Concentrati (ppm)
300
200
100
0
40
80
120
160
Time (seconds)

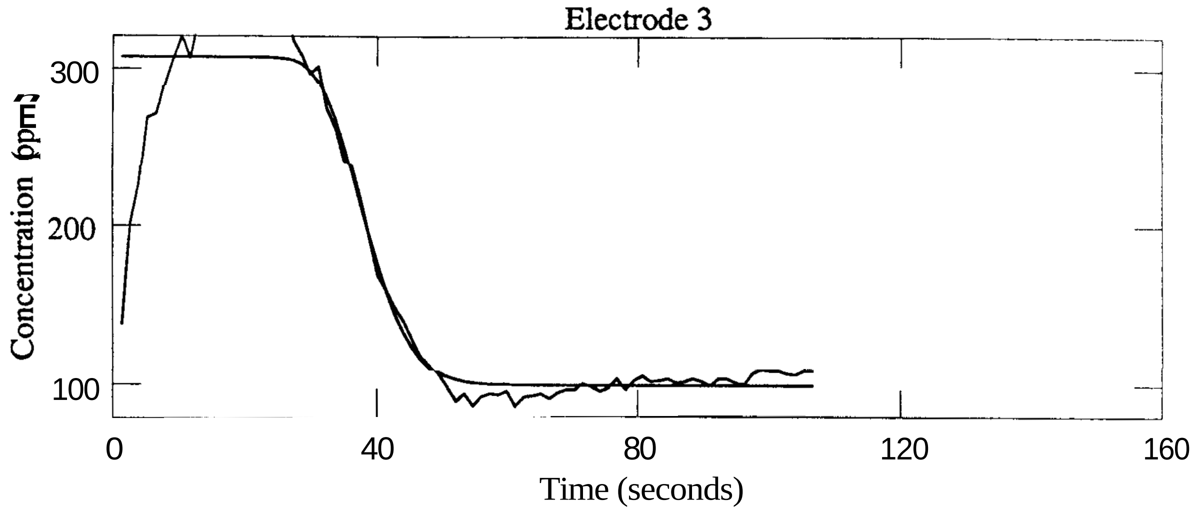

Electrode 33
Concπtration (ppm)
300
200
100
0
40
80
120
160
Time (seconds)

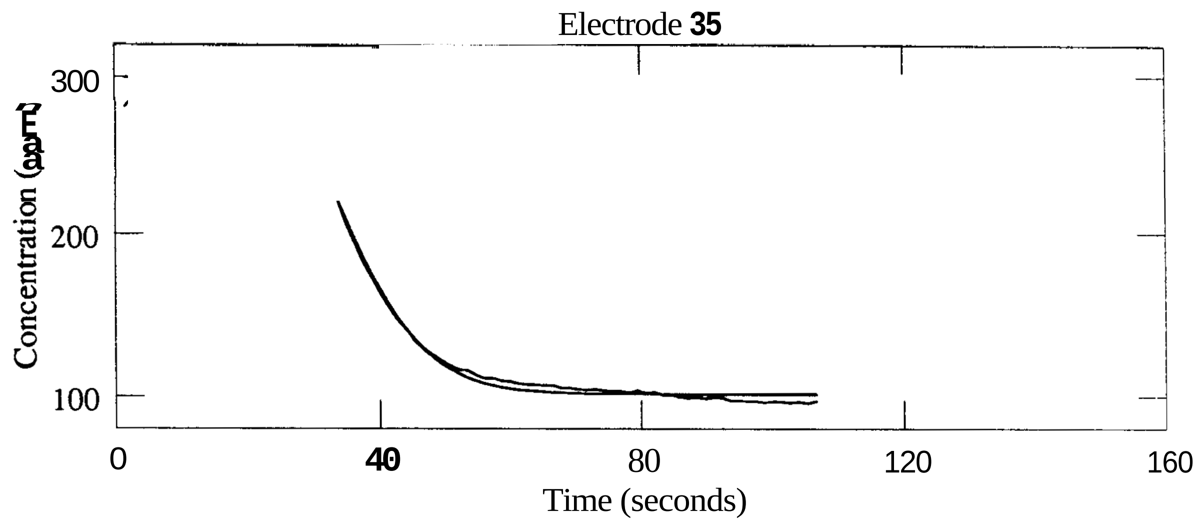

Electrode 65
Concentration (ppm)
300
200
100
0
40
80
120
160
Time (seconds)

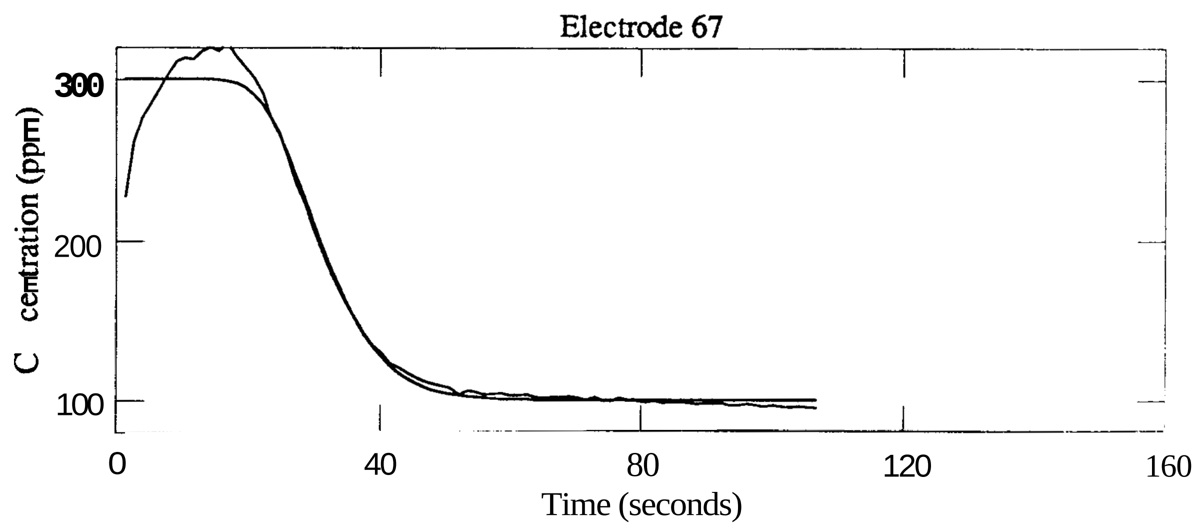

Electrode 2
300
200
100
0
40
80
120
160
Time (seconds)

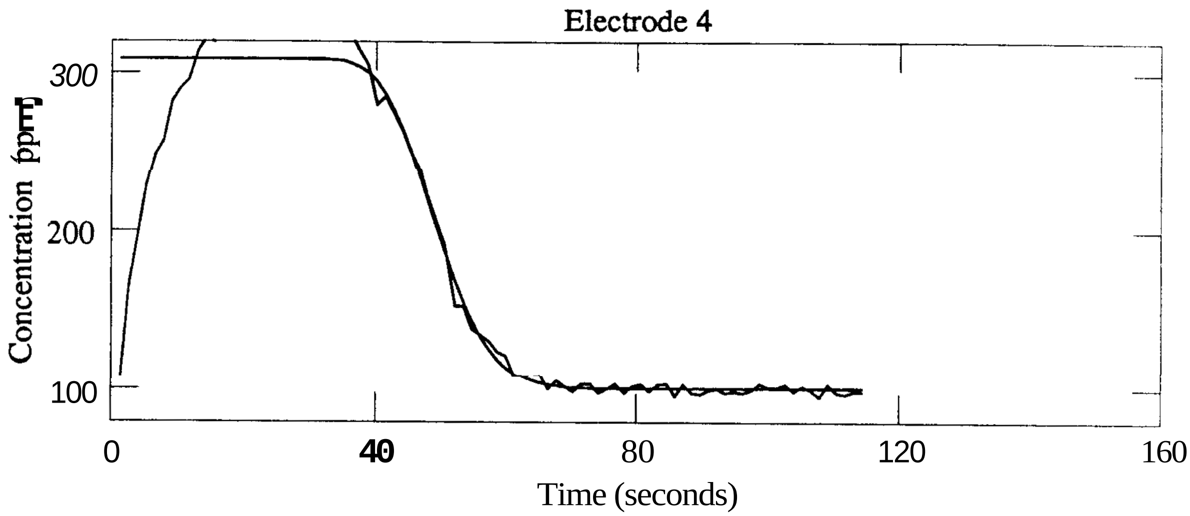

Electrode 34
300
200
100
0
40
80
120
160
Time (seconds)

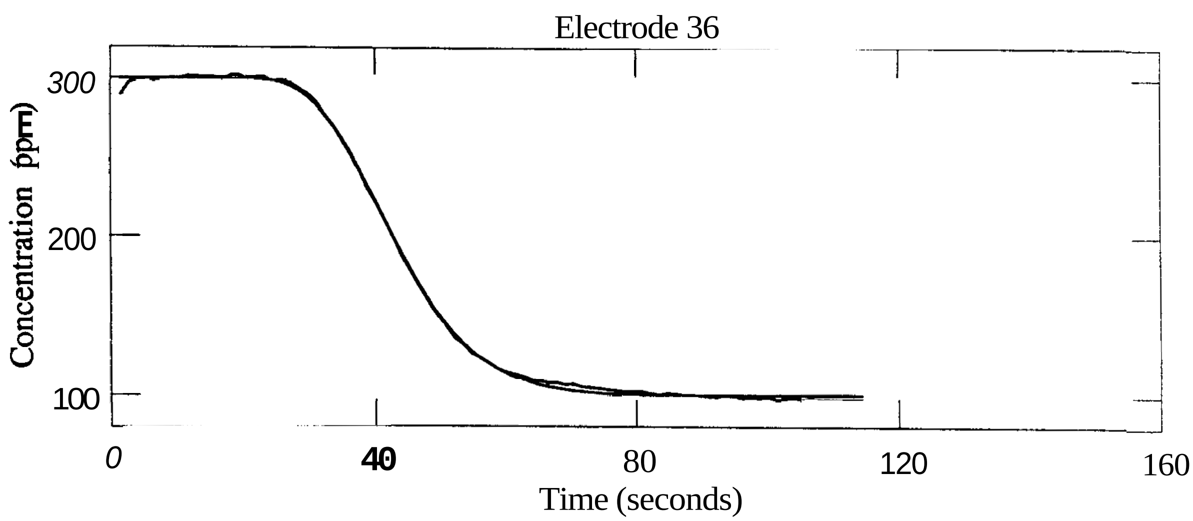

Electrode 66
300
200
100
0
40
80
120
160
Time (seconds)

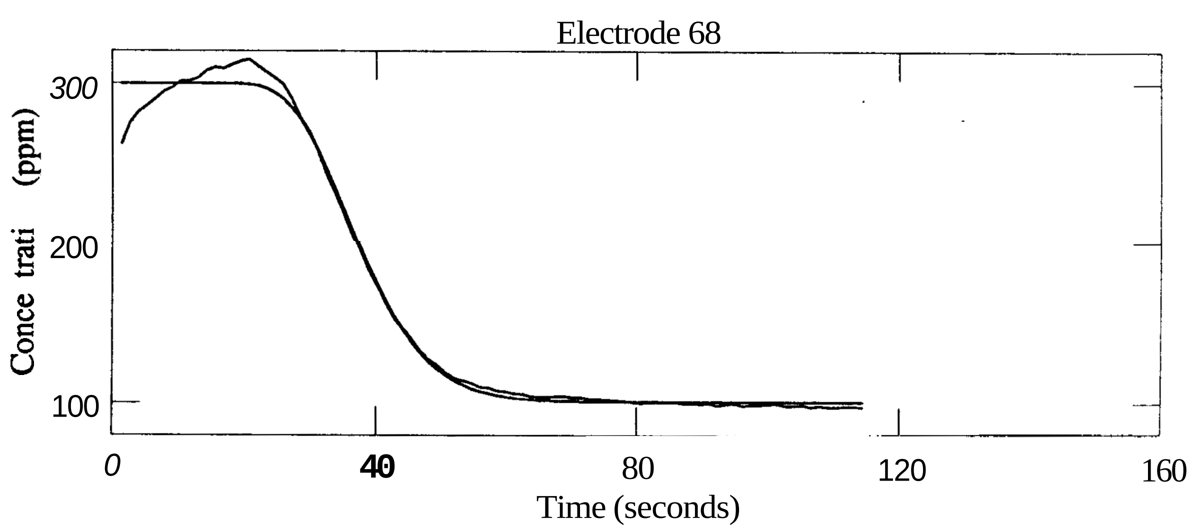

Electrode 3
Concentration (ppm)
300
200
100
0
40
80
120
160
Time (seconds)

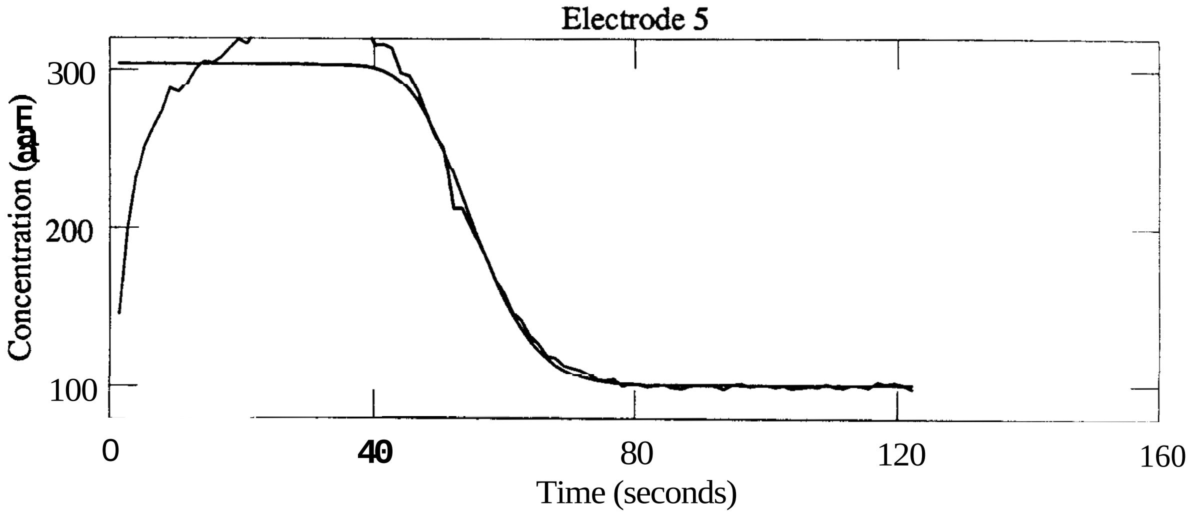

Electrode 35
300
200
100
0
40
80
120
160
Time (seconds)

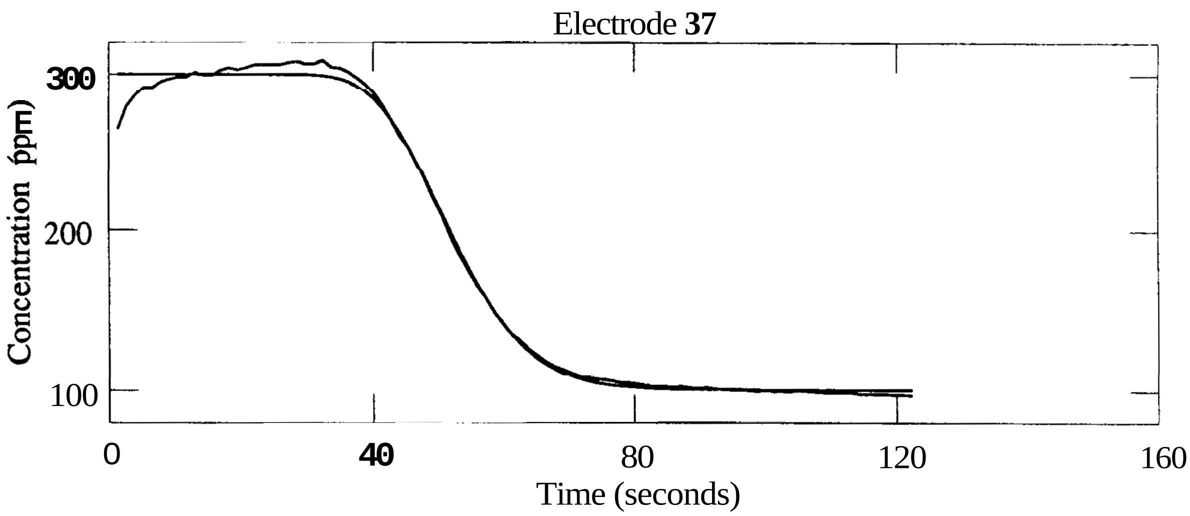

Electrode 67
300
200
100
0
40
80
120
160
Time (seconds)

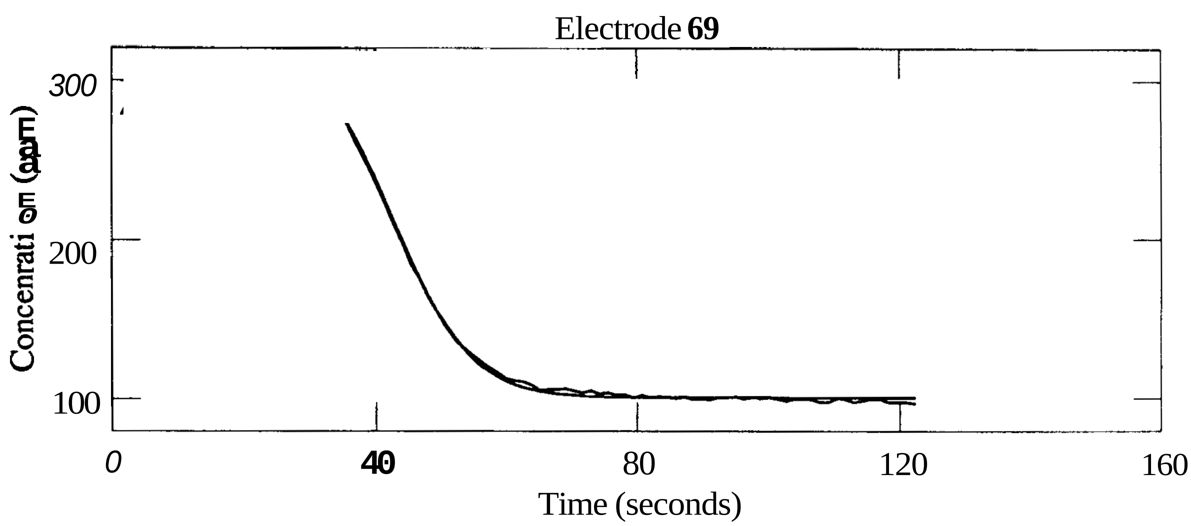

Electrode 4
Concentration (ppm)
300
200
100
0
40
80
120
160
Time (seconds)

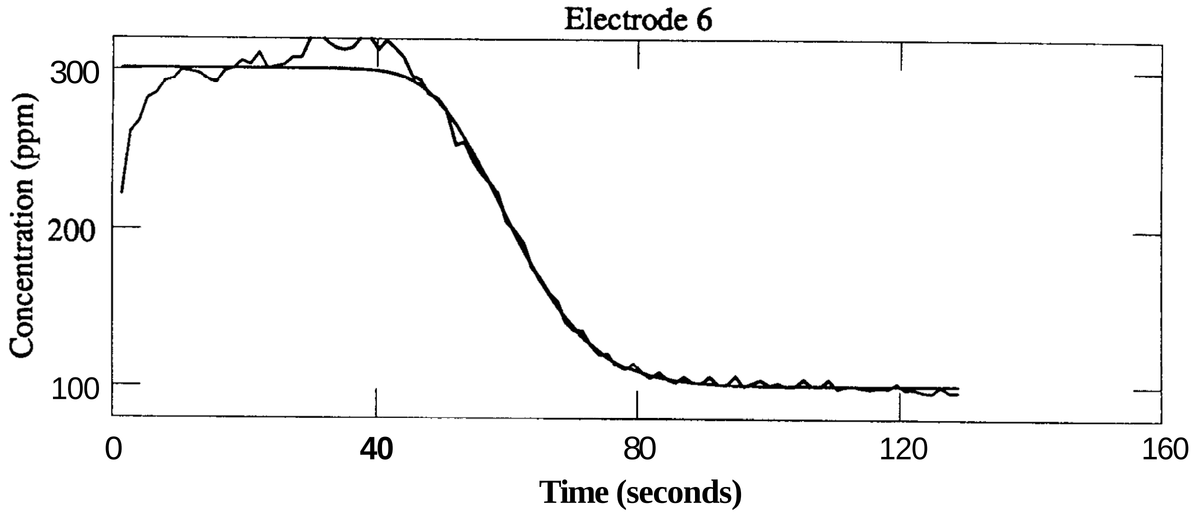

Electrode 36
Concentration (ppm)
300
200
100
0
40
80
120
160
Time (seconds)

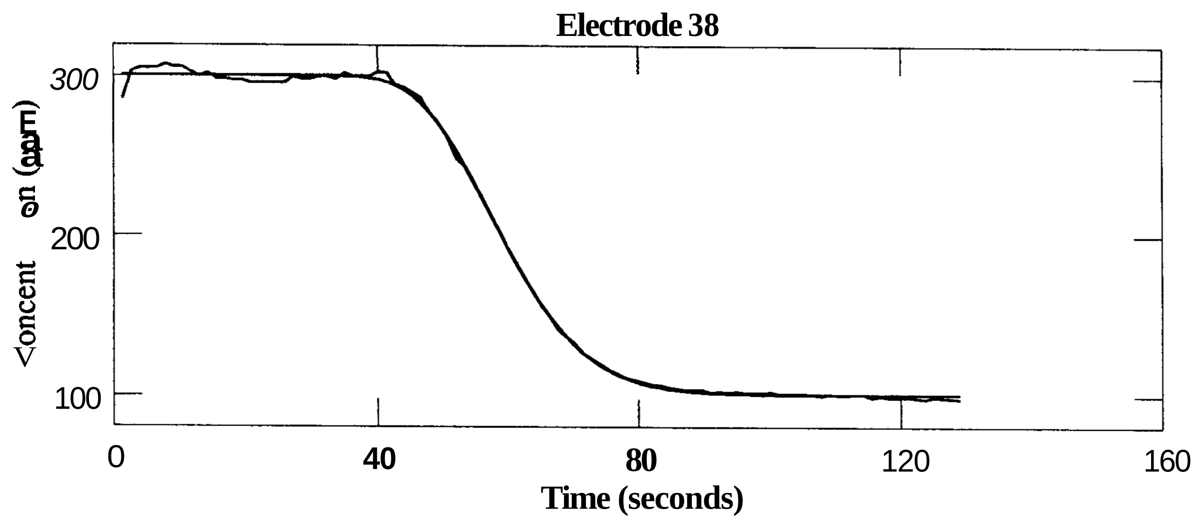

Electrode 68
(ppm)
300
200
100
0
40
80
120
160
Time (seconds)

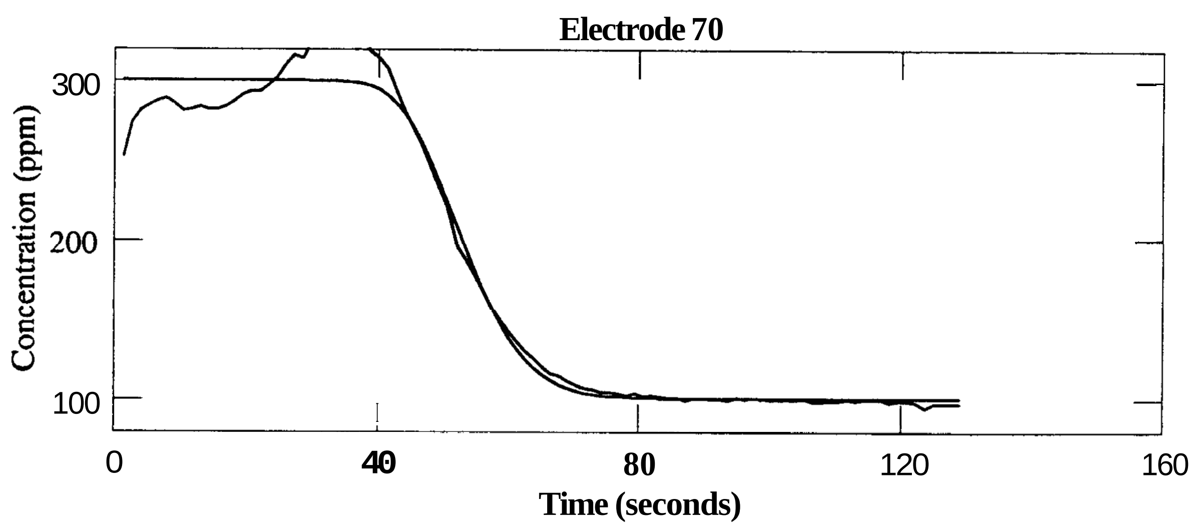

Electrode 5
300
200
100
Concentration (ppm)
0
40
80
120
160
Time (seconds)


Electrode 37
300
200
100
Concentration (ppm)
0
40
80
120
160
Time (seconds)

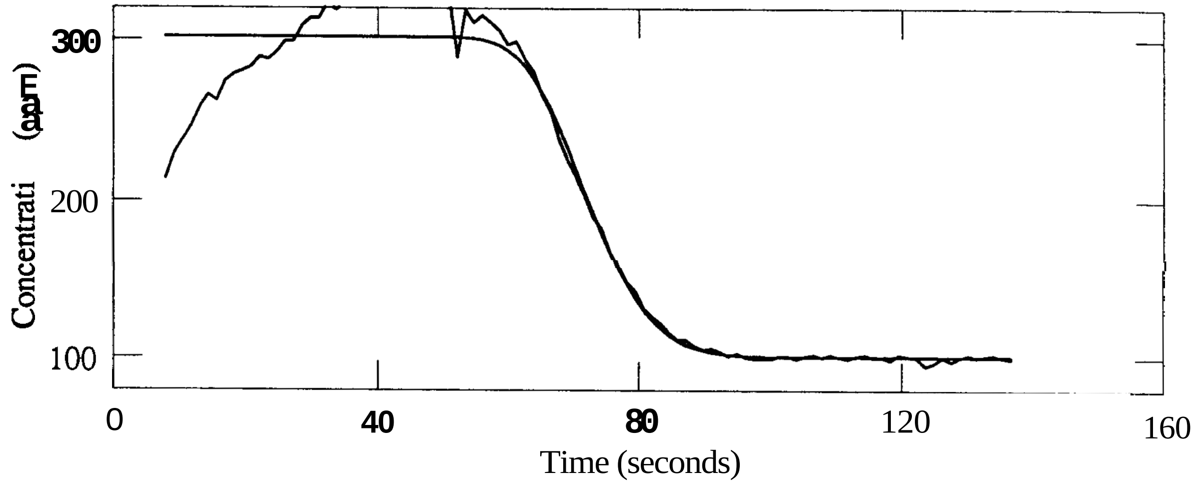

Electrode 69
300
200
100
0
40
80
120
160
Time (seconds)


Electrode 6
Concentration (ppm)
300
200
100
0
40
80
120
160
Time (seconds)

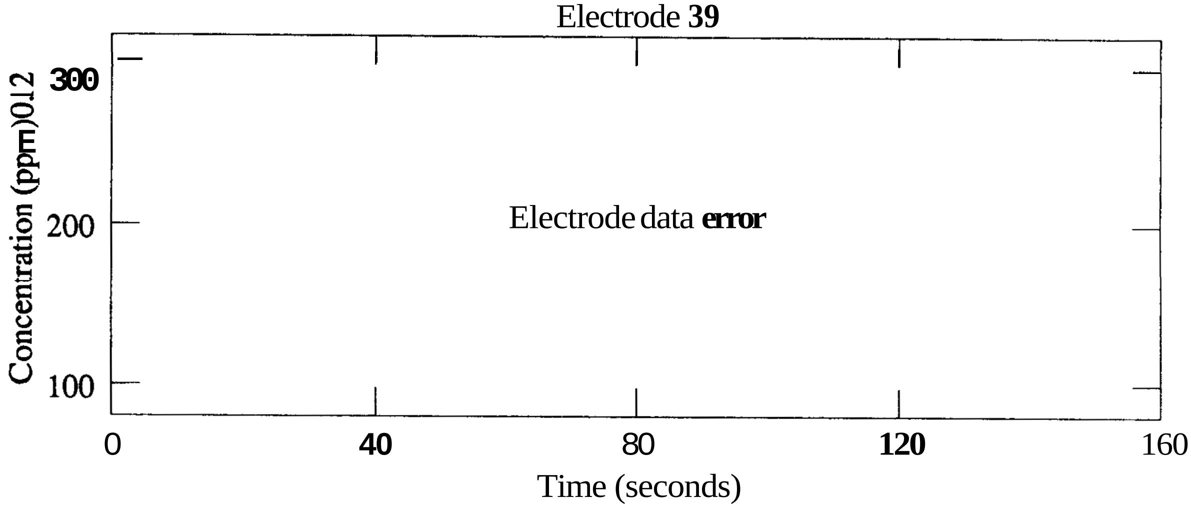

Electrode 38
300
200
100
0
40
80
120
160
Time (seconds)


Electrode 70
Concentration (ppm)
300
200
100
0
40
80
120
160
Time (seconds)

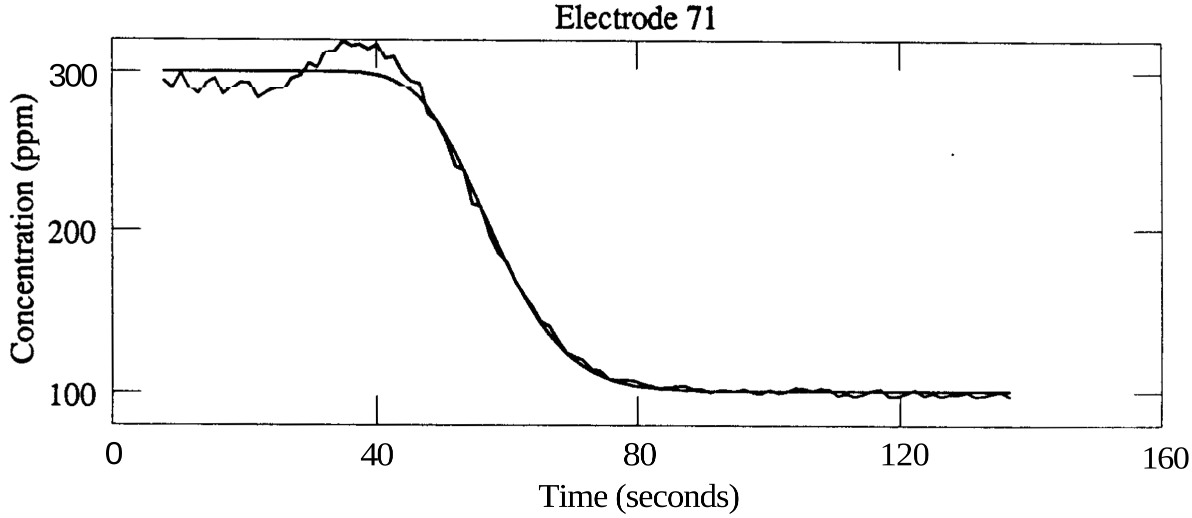
Electrode 7

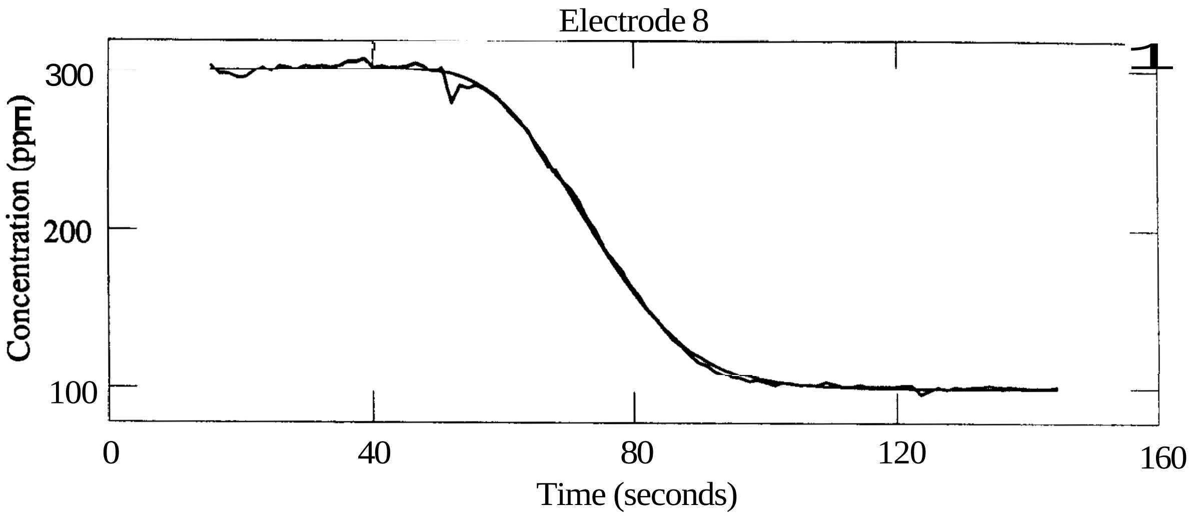


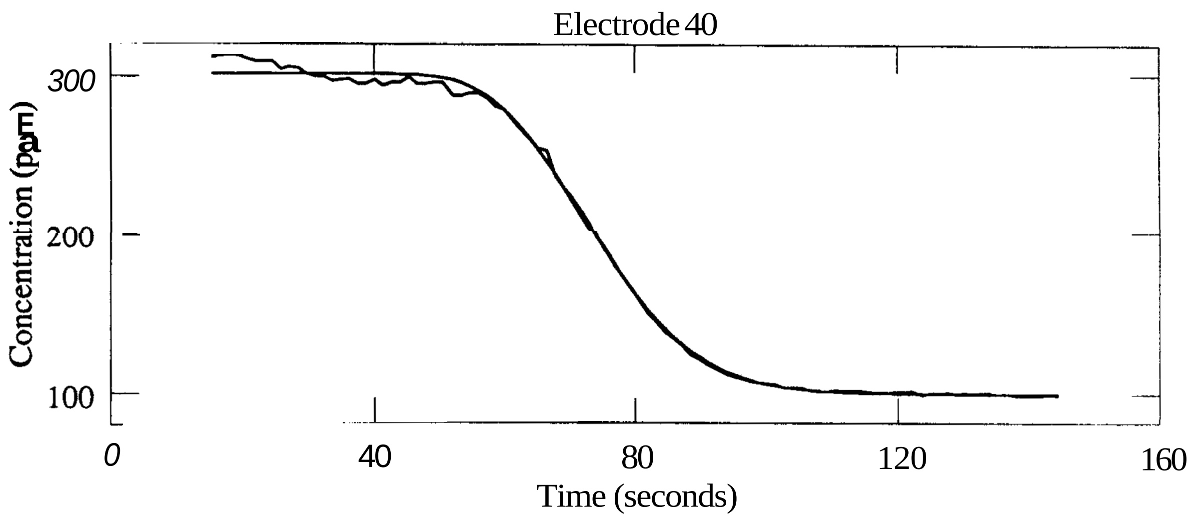
Electrode 39

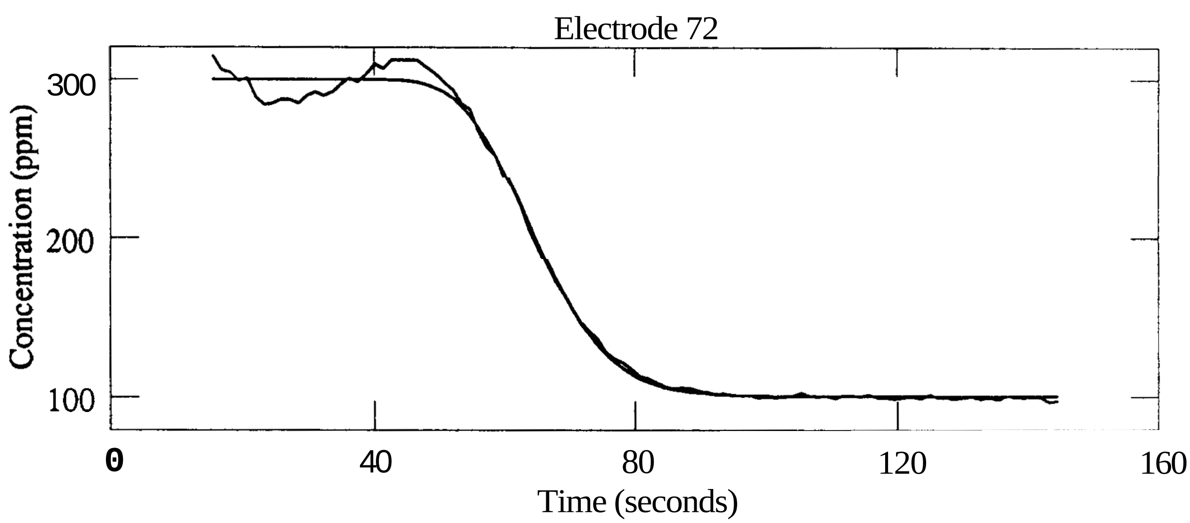


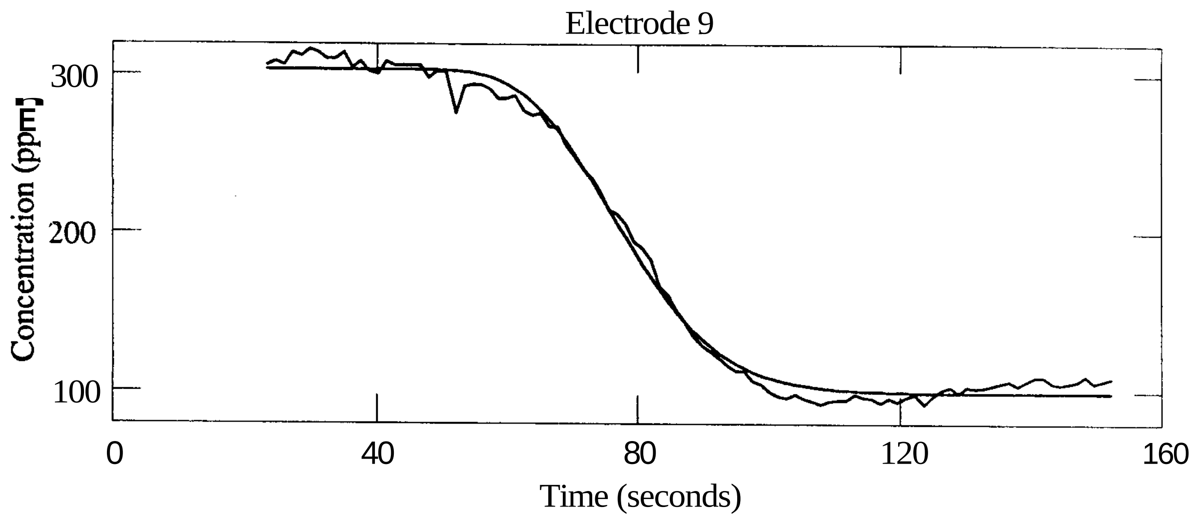
Electrode 71

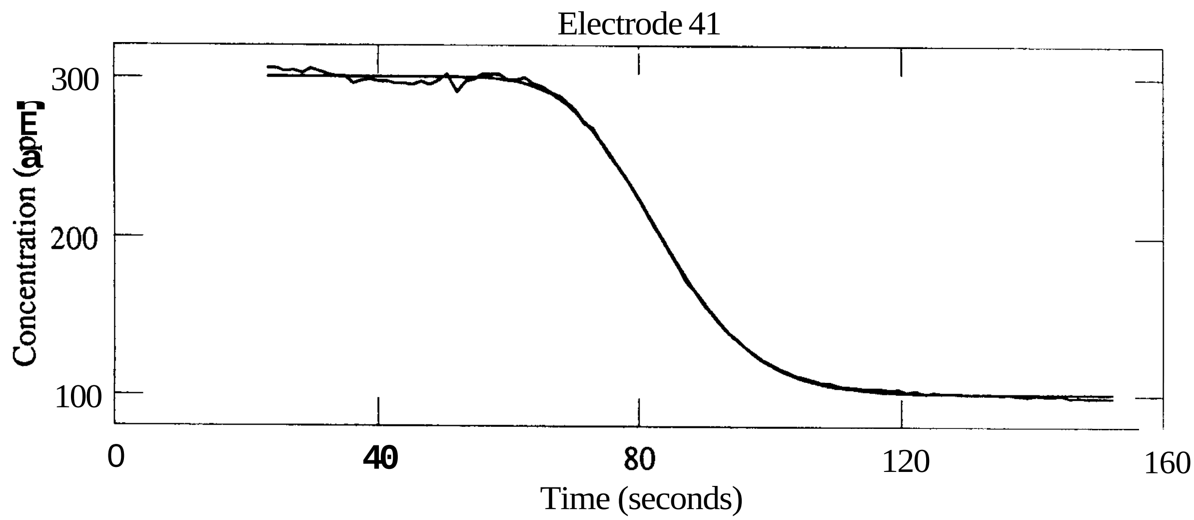


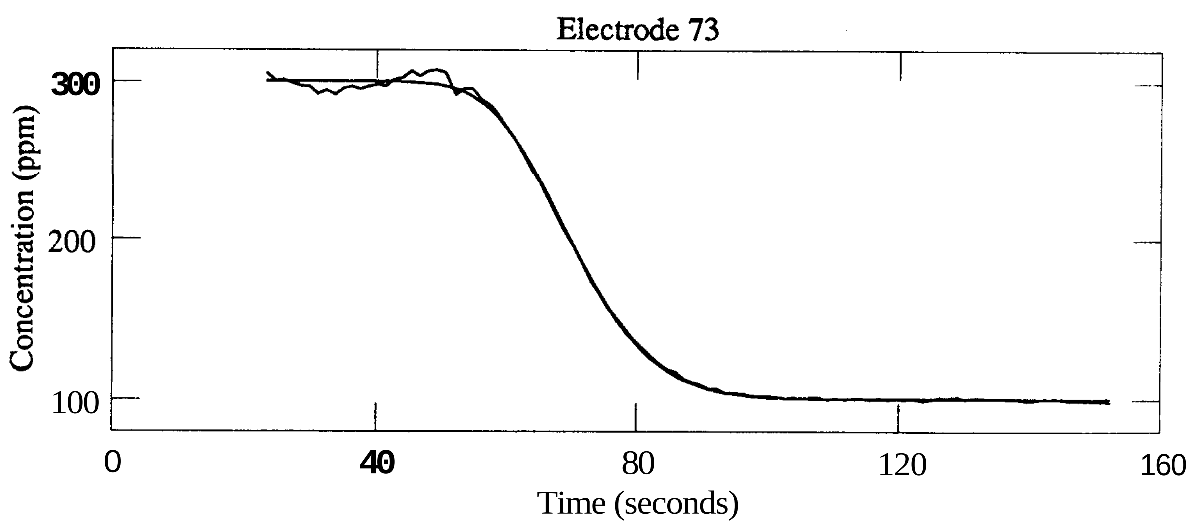

Electrode 8
Concentration (ppm)
300
200
100
0
40
80
120
160
Time (seconds)

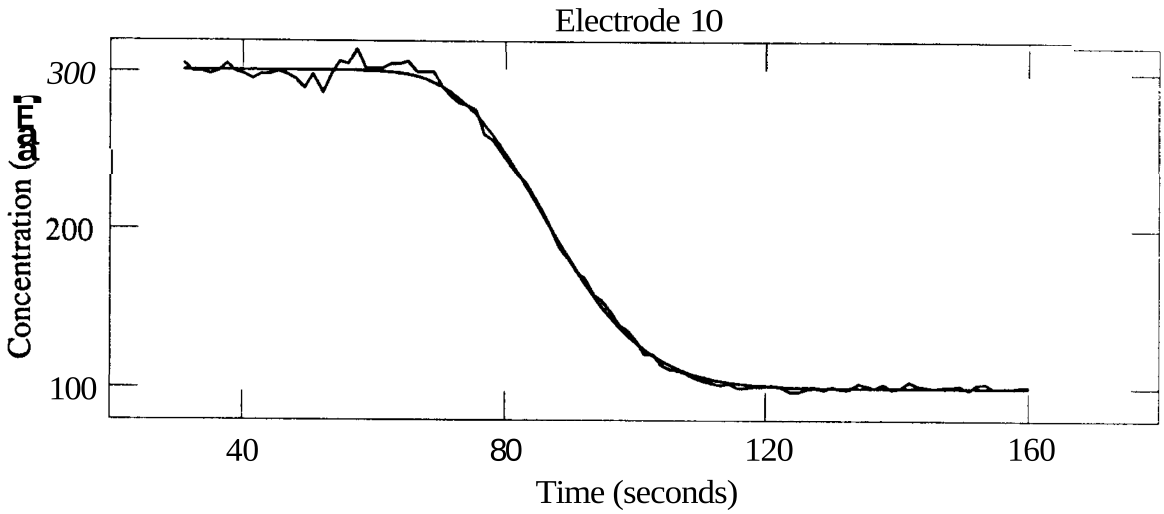

Electrode 40
Concentration (ppm)
300
200
100
0
40
80
120
160
Time (seconds)

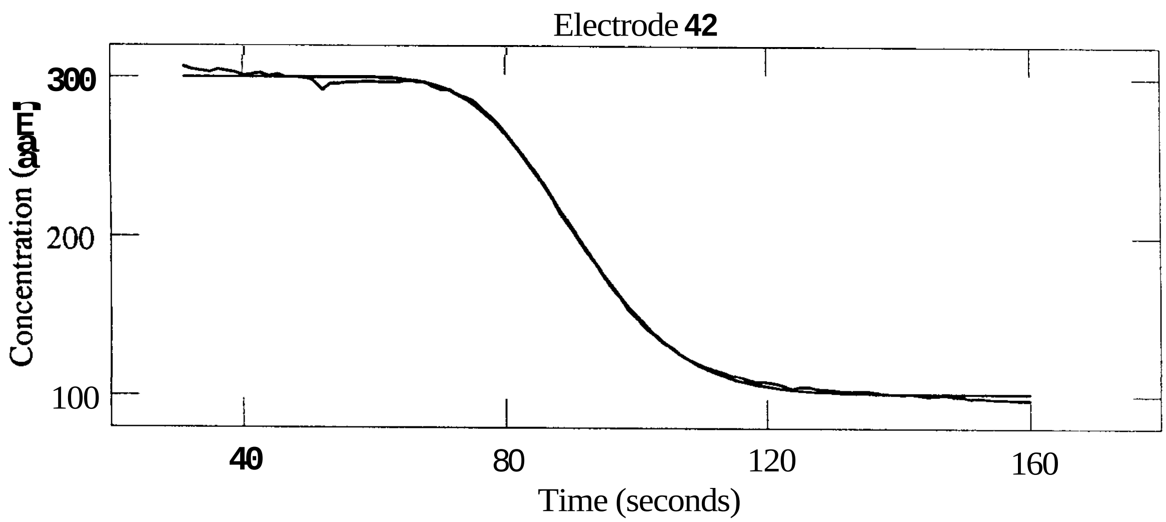

Electrode 72
Concentration (ppm)
300
200
100
0
40
80
120
160
Time (seconds)

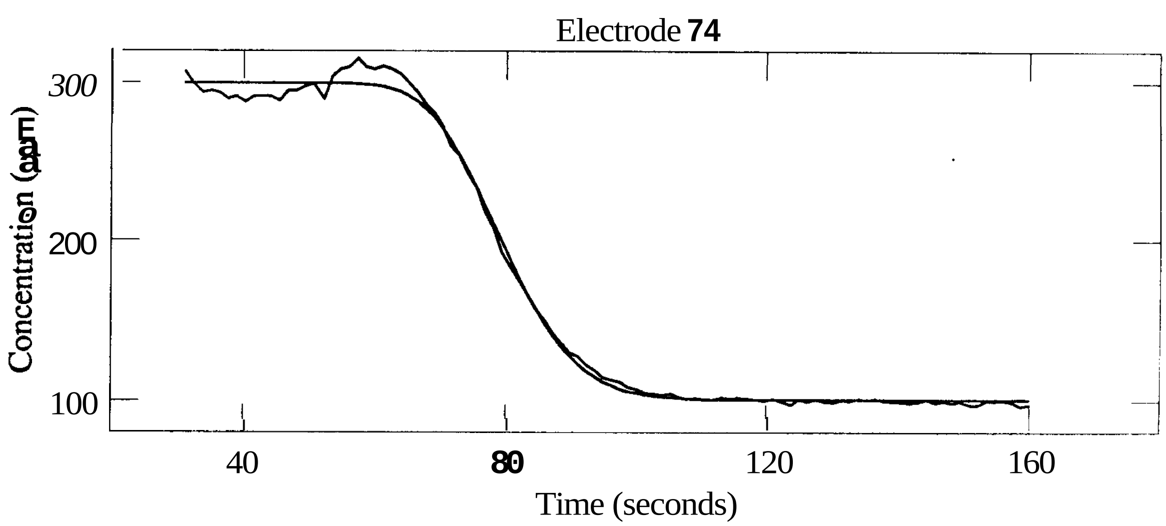

Electrode 9
Concentration (ppm)
300
200
100
0
40
80
120
160
Time (seconds)


Electrode 41
Concentration (ppm)
300
200
100
0
40
80
120
160
Time (seconds)

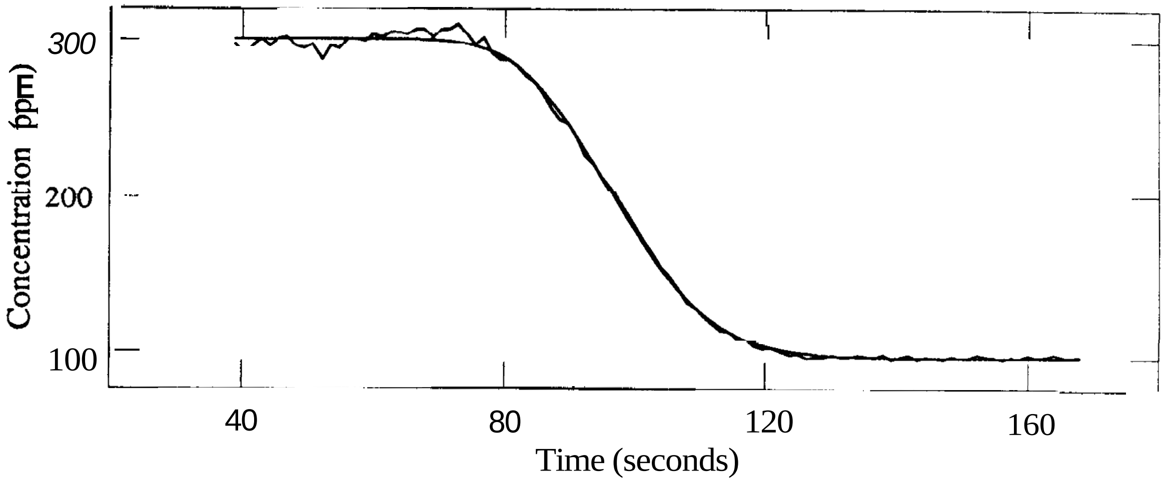

Electrode 73
Concentration (ppm)
300
200
100
0
40
80
120
160
Time (seconds)


Electrode 10
300
200
100
Concentration (ppm)
40
80
120
160
Time (seconds)

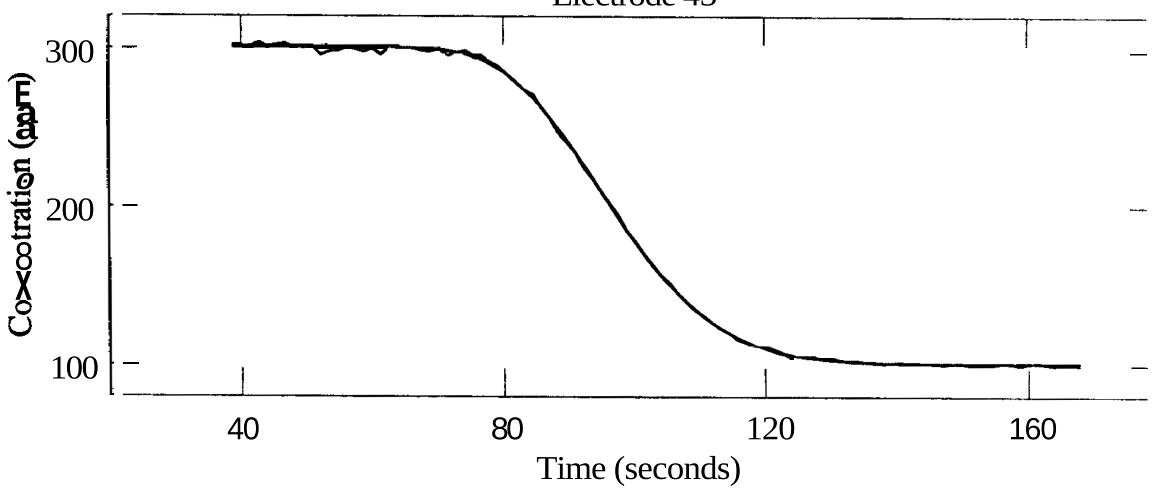

Electrode 42
300
200
100
Concentration (ppm)
40
80
120
160
Time (seconds)


Electrode 74
300
200
100
Concentration (ppm)
40
80
120
160
Time (seconds)

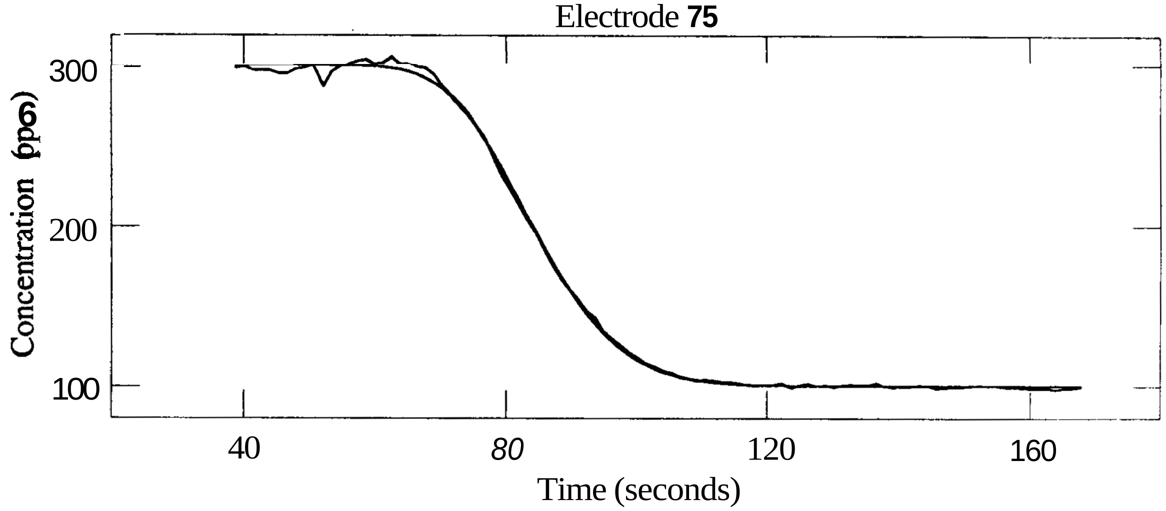
Electrode 11

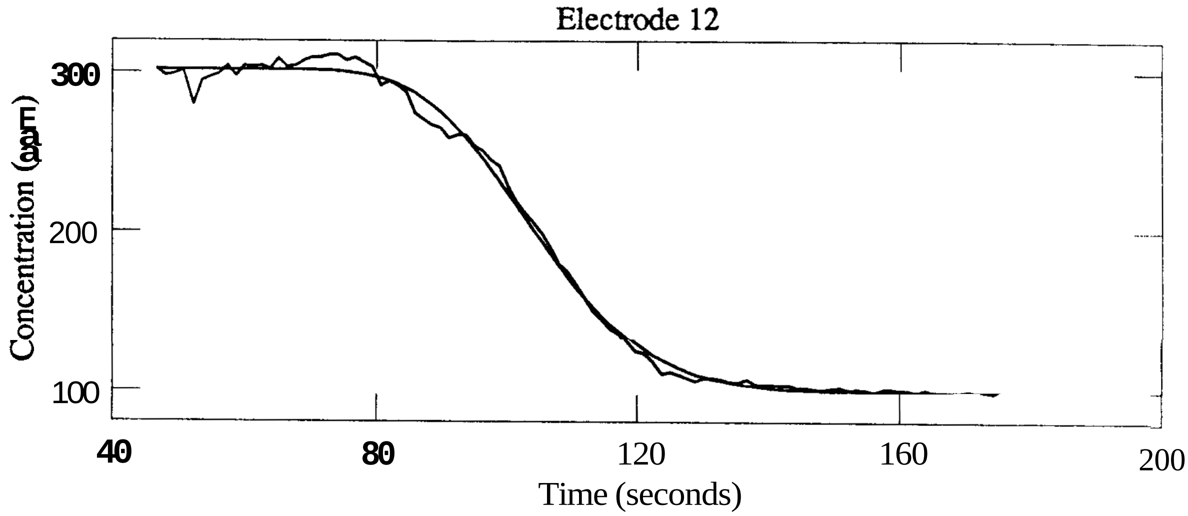


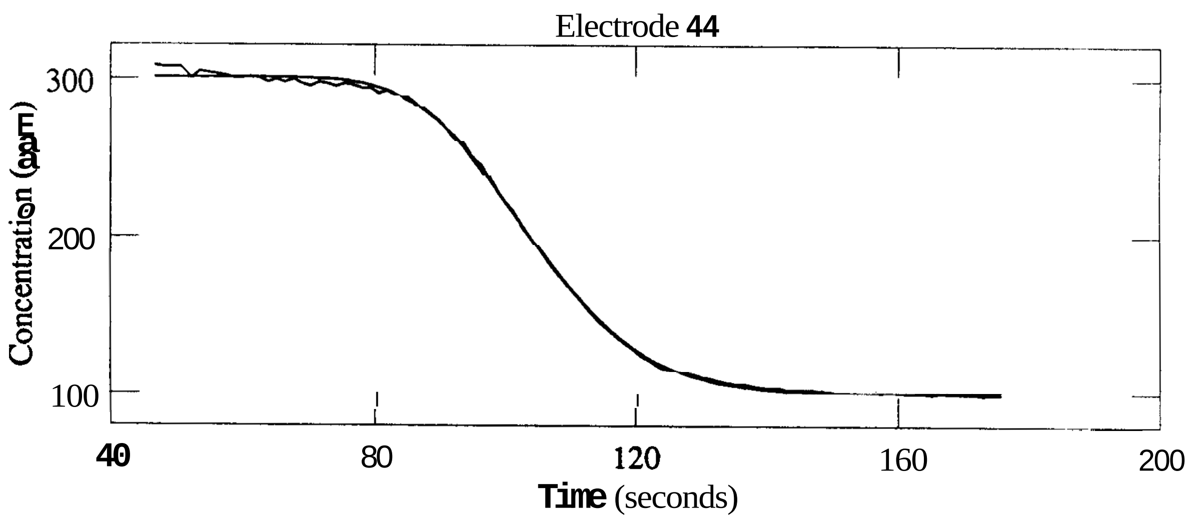
Electrode 43

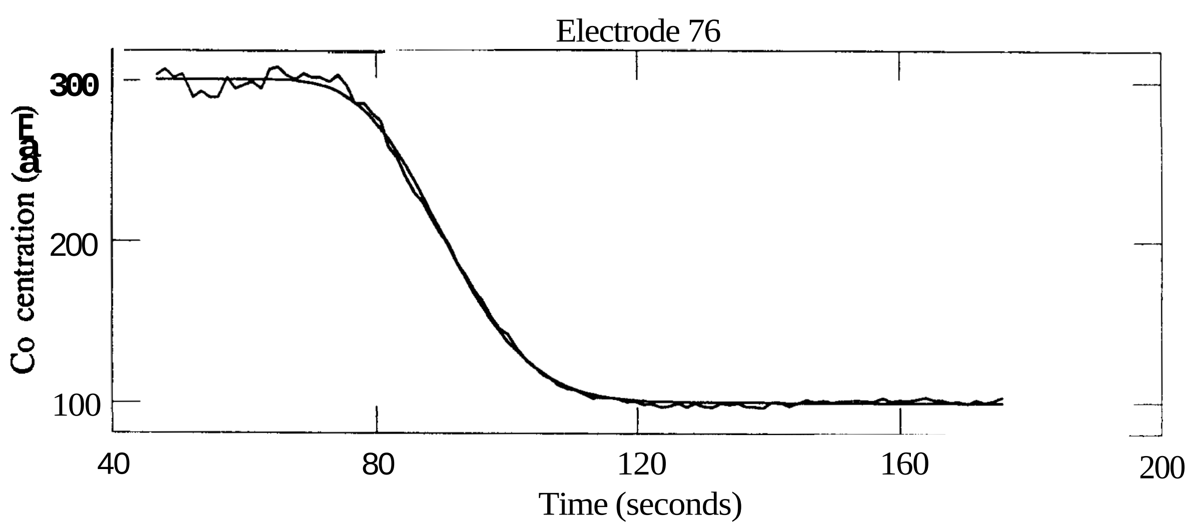


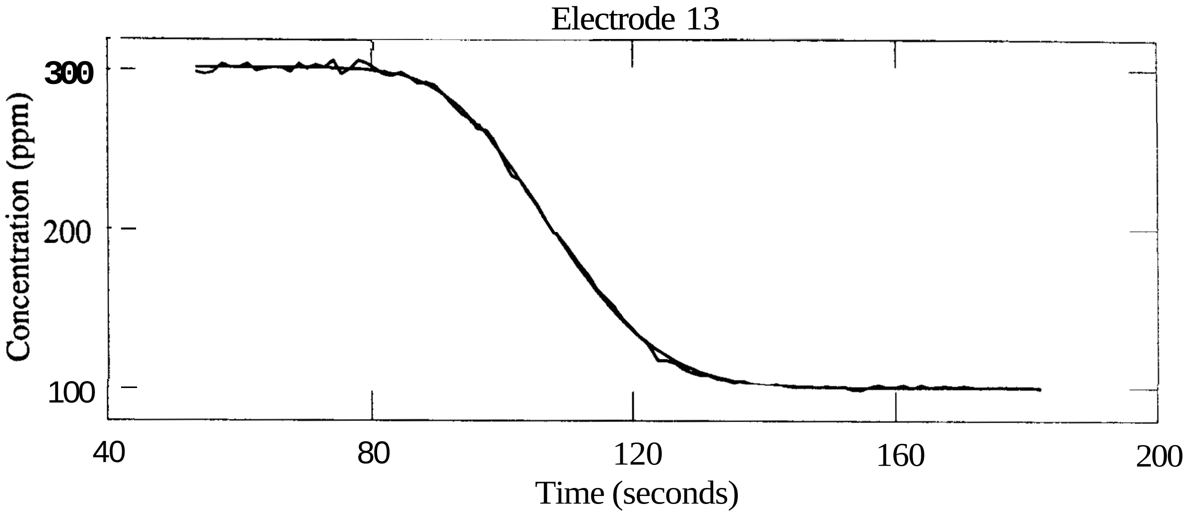
Electrode 75

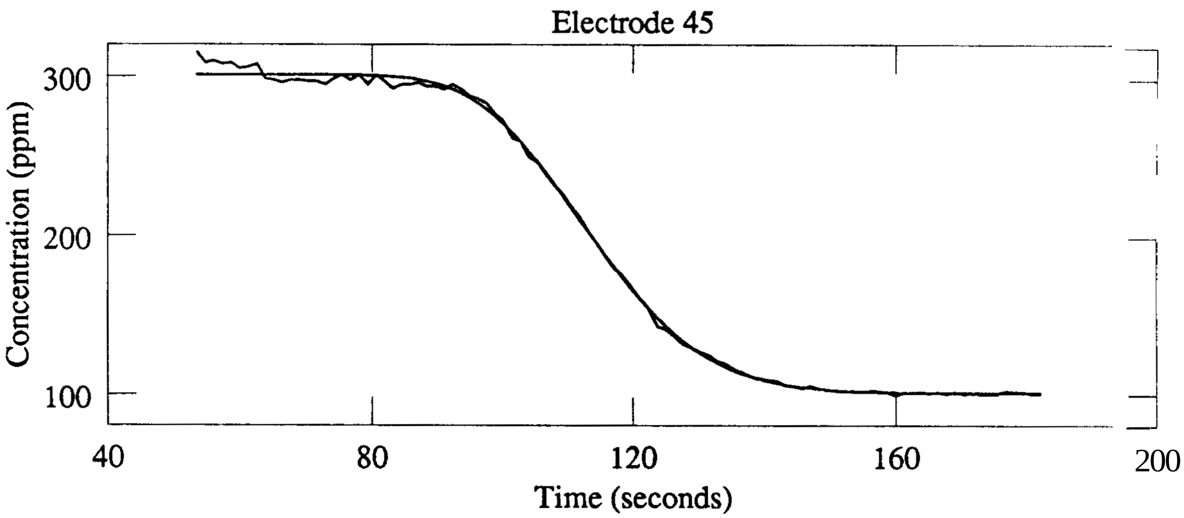


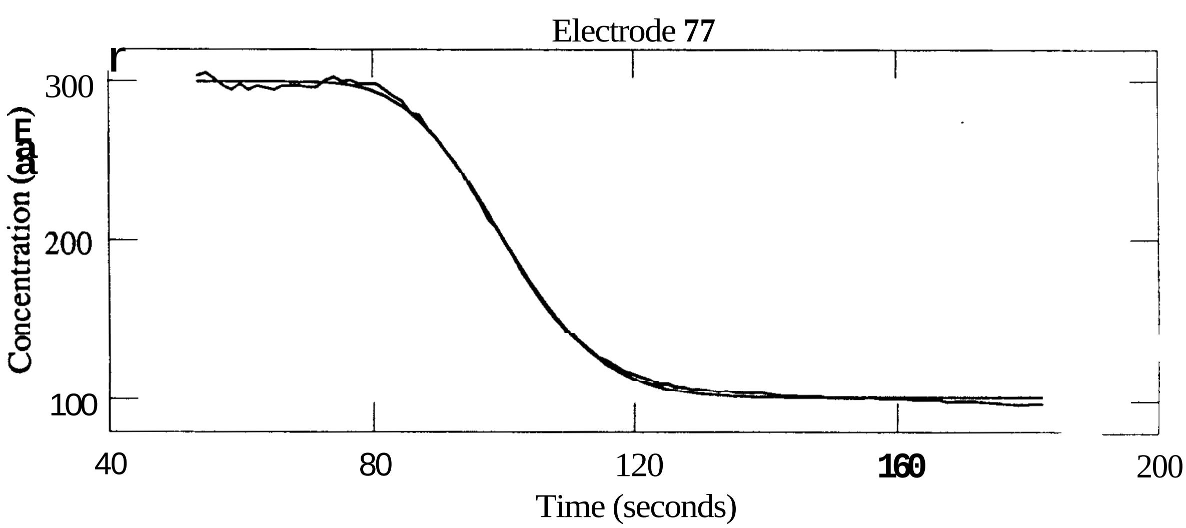

Electrode 12
Concentration
300
200
100
40
80
120
160
200
Time (seconds)

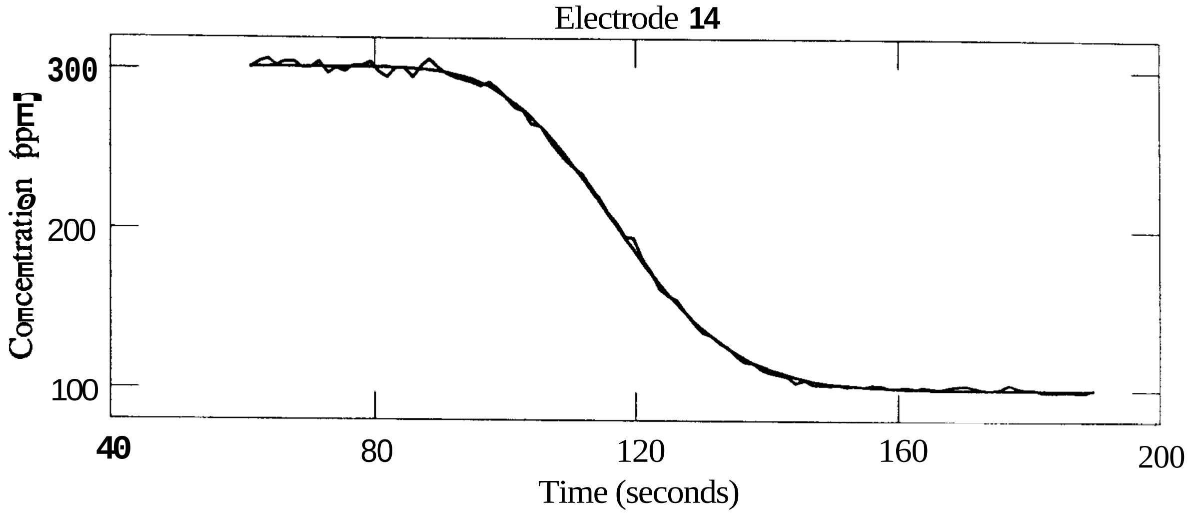

Electrode 44
Concentration
300
200
100
40
80
120
160
200
Time (seconds)

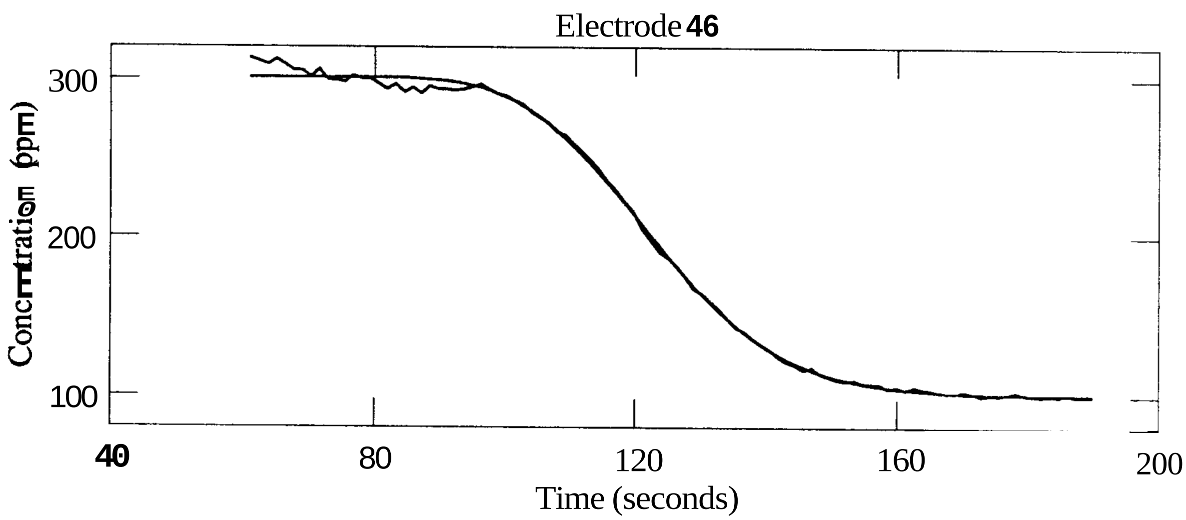

Electrode 76
300
200
100
40
80
120
160
200
Time (seconds)

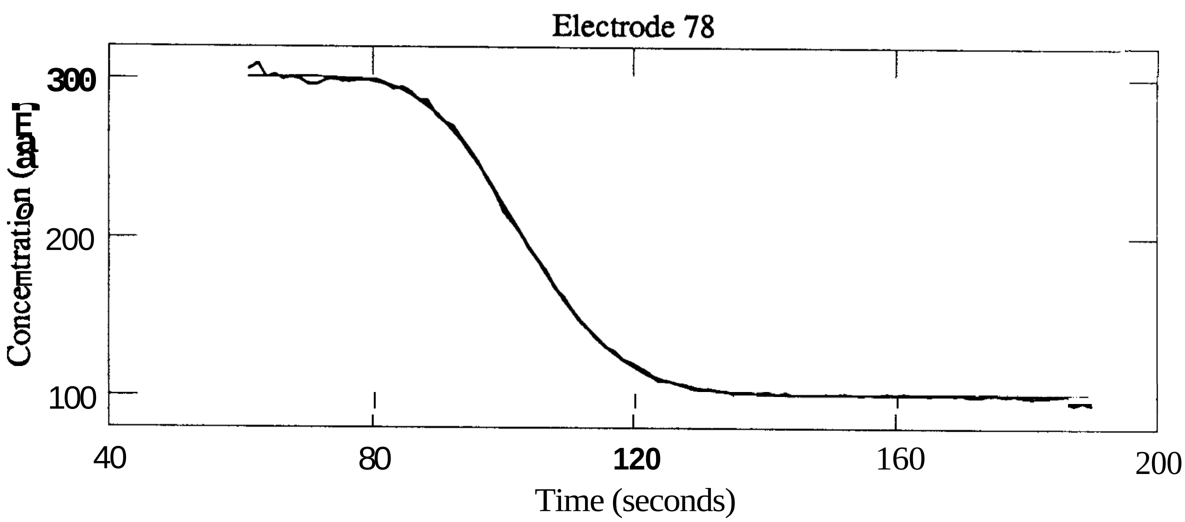

Electrode 13
300
200
100
Concentration (ppm)
40
80
120
160
200
Time (seconds)

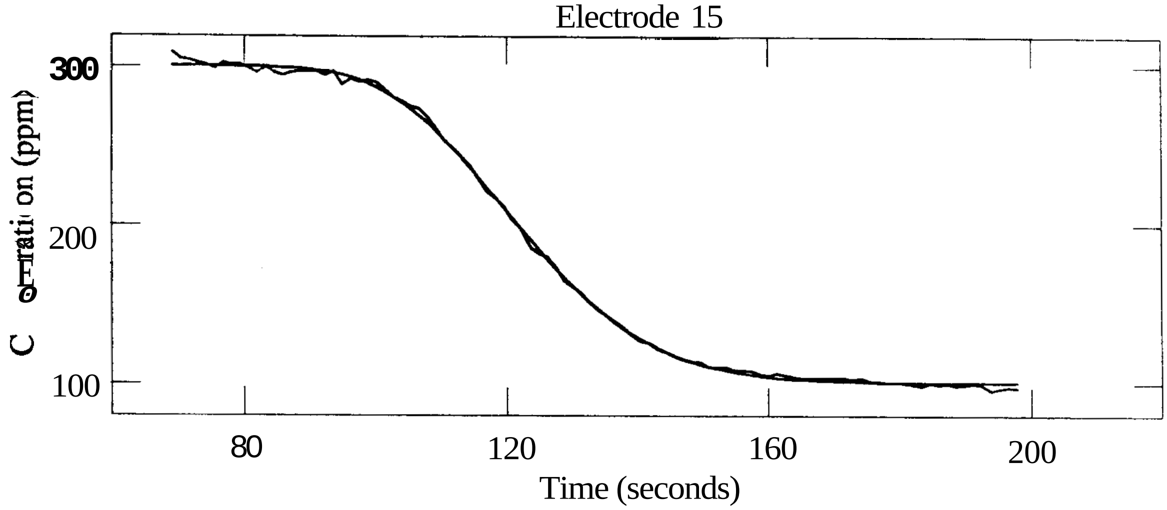

Electrode 45
300
200
100
Concentration (ppm)
40
80
120
160
200
Time (seconds)

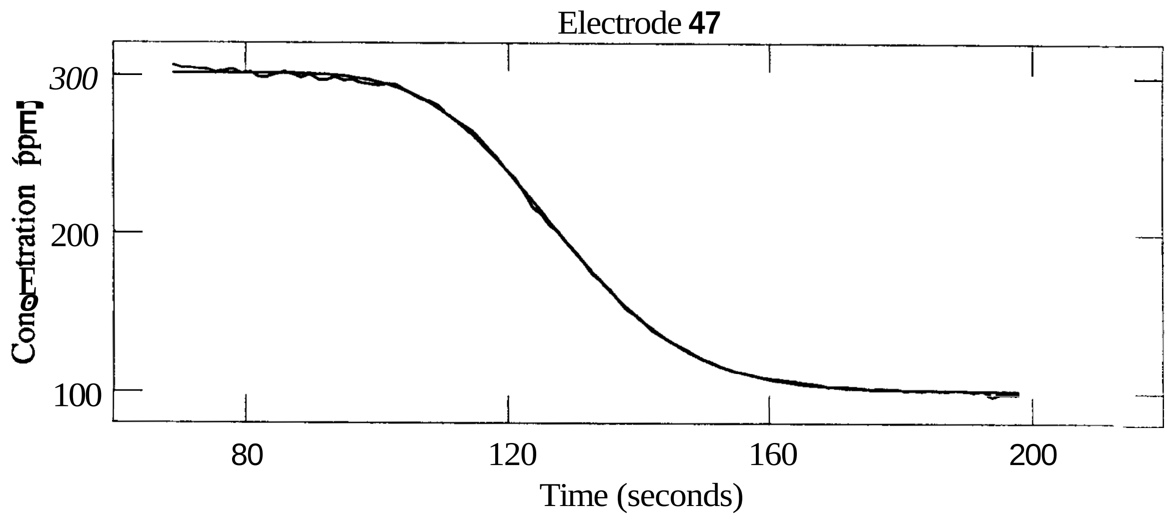

Electrode 77
300
200
100
Concentration
40
80
120
160
200
Time (seconds)

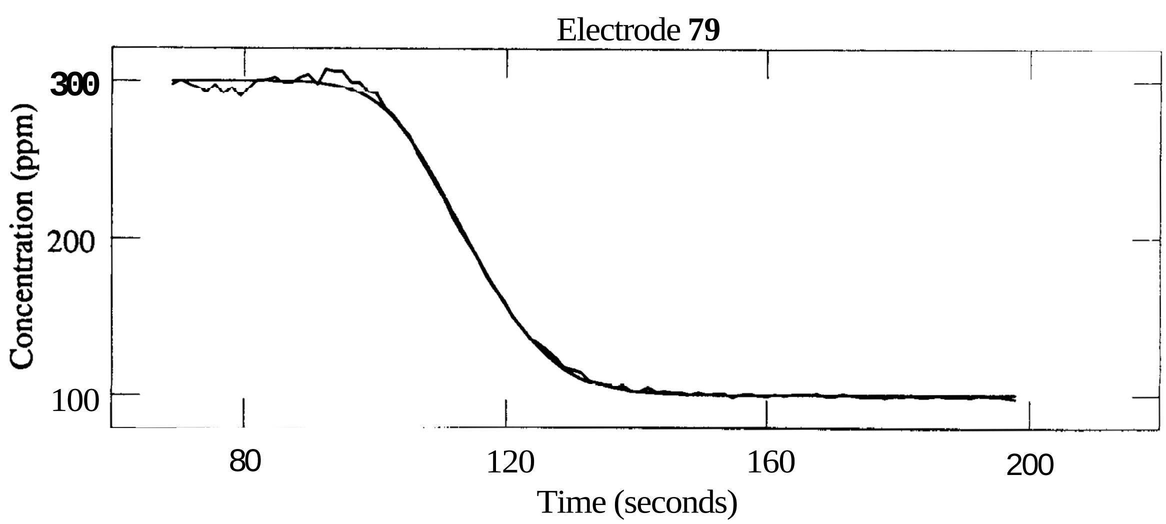

Electrode 14
300
200
100
Concentration (ppm)
40
80
120
160
200
Time (seconds)

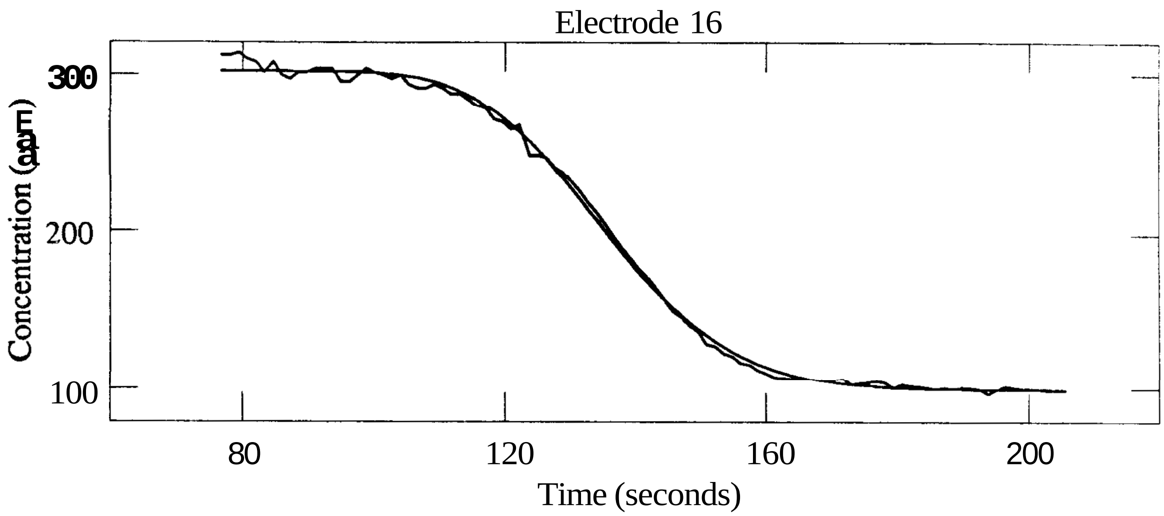

Electrode 46
300
200
100
Concentration (ppm)
40
80
120
160
200
Time (seconds)

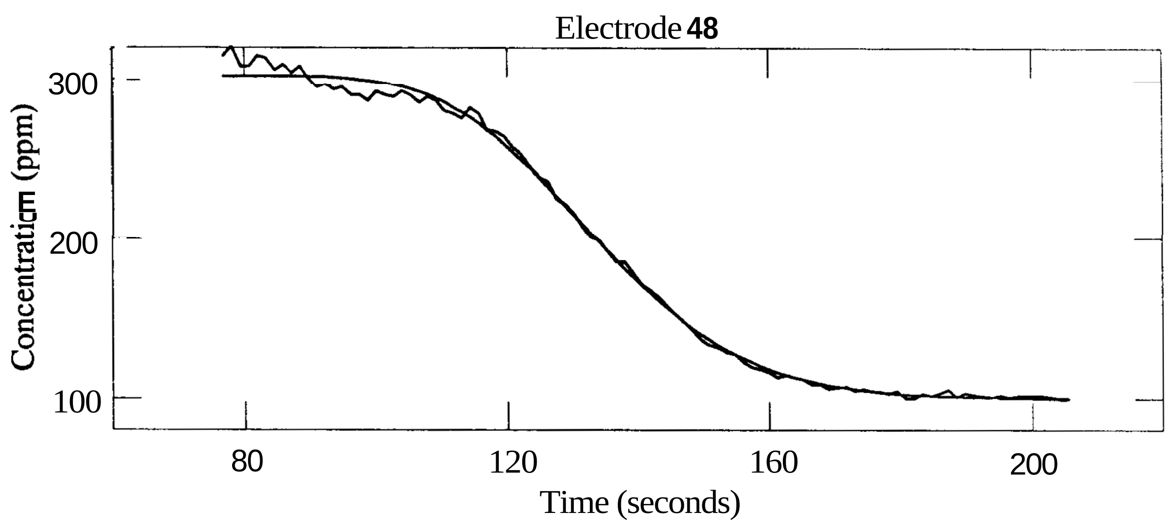

Electrode 78
300
200
100
Concentration (ppm)
40
80
120
160
200
Time (seconds)

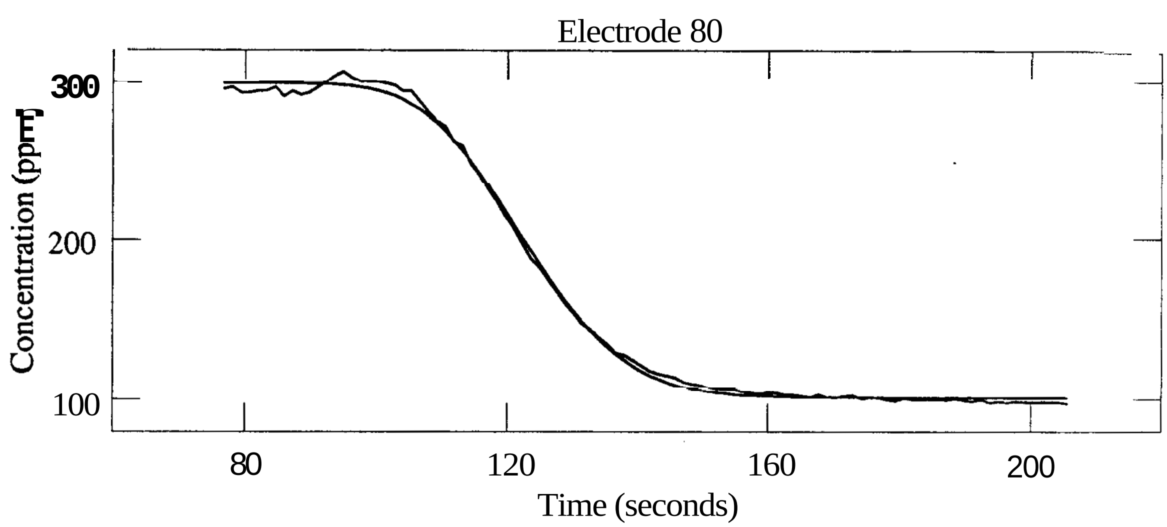

Electrode 15
300
200
100
80
120
160
200
Time (seconds)

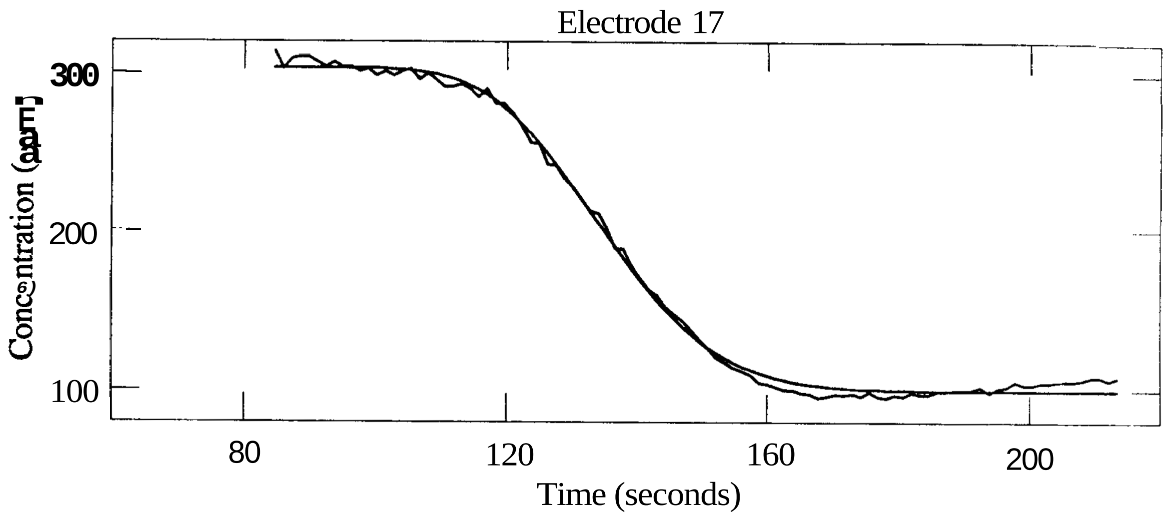

Electrode 47
300
200
100
80
120
160
200
Time (seconds)

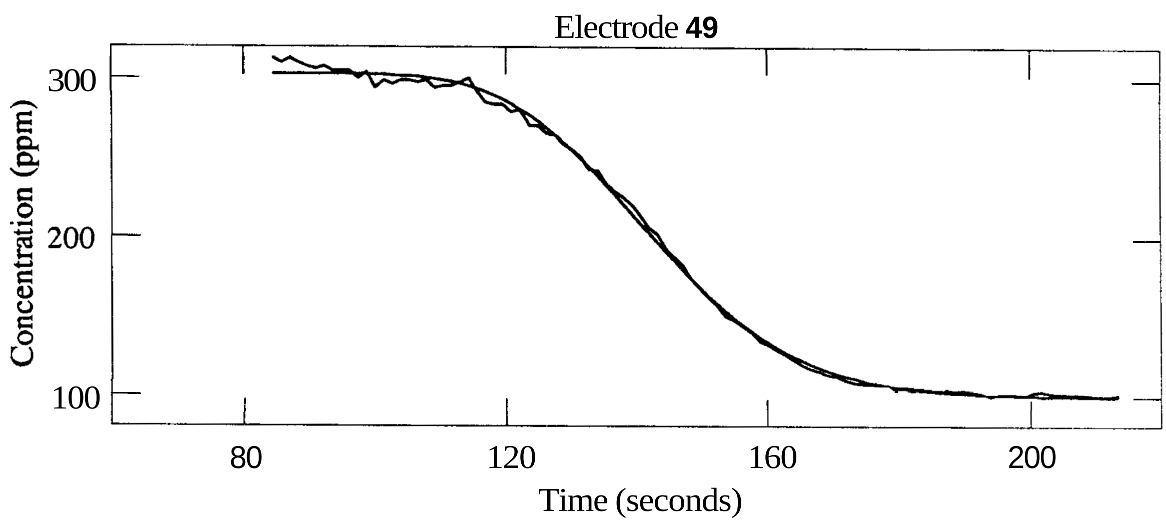

Electrode 79
Concentration (ppm)
300
200
100
80
120
160
200
Time (seconds)

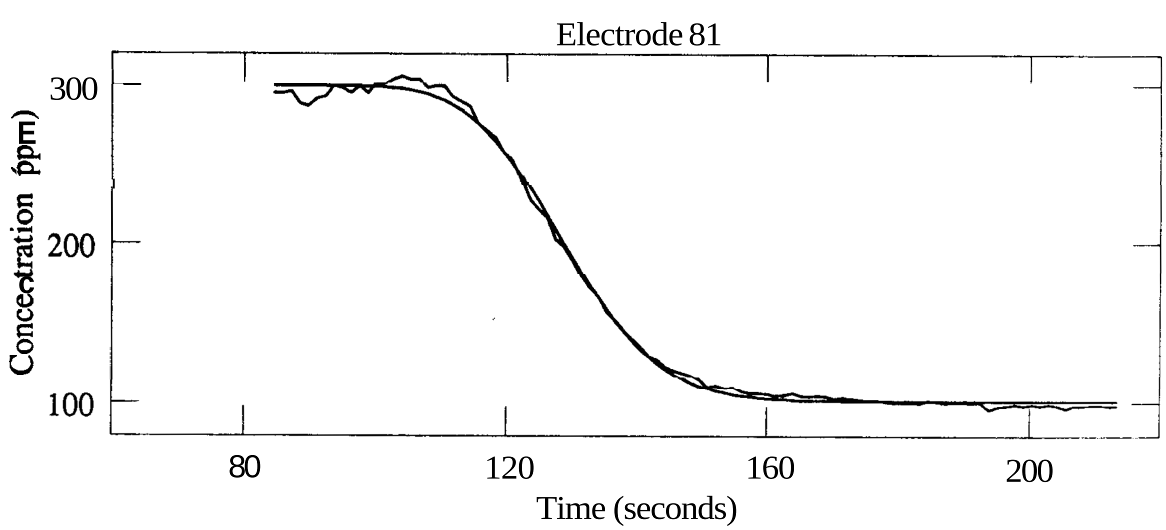

Electrode 16
300
200
100
Concentration (ppm)
80
120
160
200
Time (seconds)

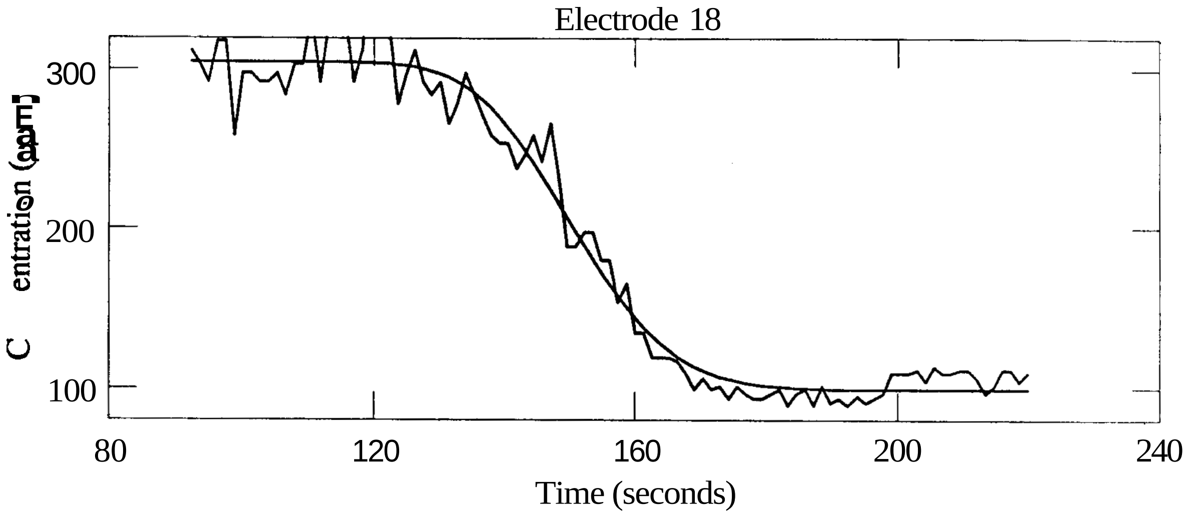

Electrode 48
300
200
100
Concentration (ppm)
80
120
160
200
Time (seconds)


Electrode 80
300
200
100
Concentration (ppm)
80
120
160
200
Time (seconds)

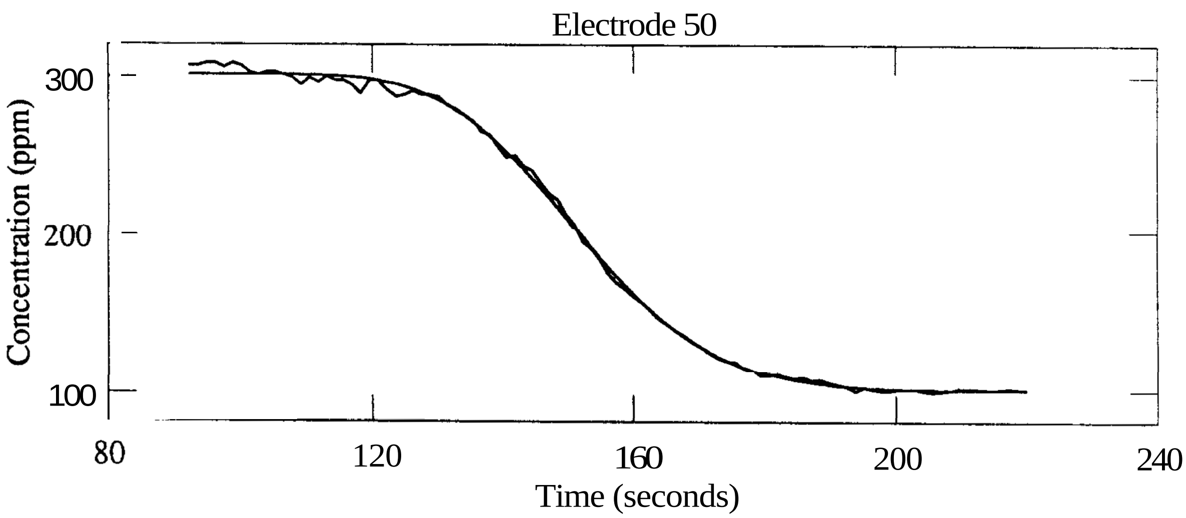

Electrode 17
300
200
100
Concentration (ppm)
80
120
160
200
Time (seconds)


Electrode 49
300
200
100
Concentration (ppm)
80
120
160
200
Time (seconds)

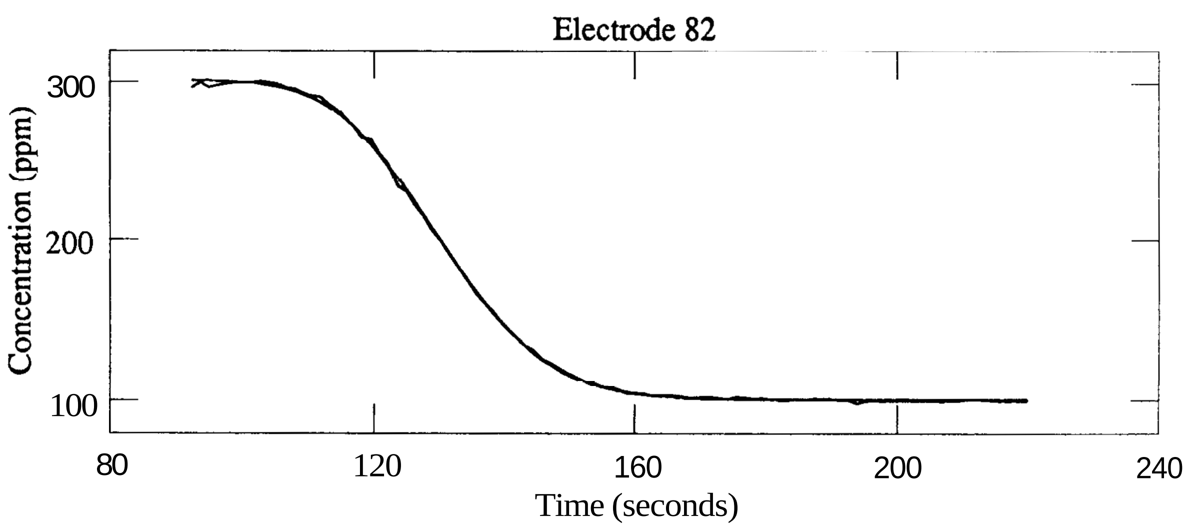

Electrode 81
300
200
100
Concentration (ppm)
80
120
160
200
Time (seconds)



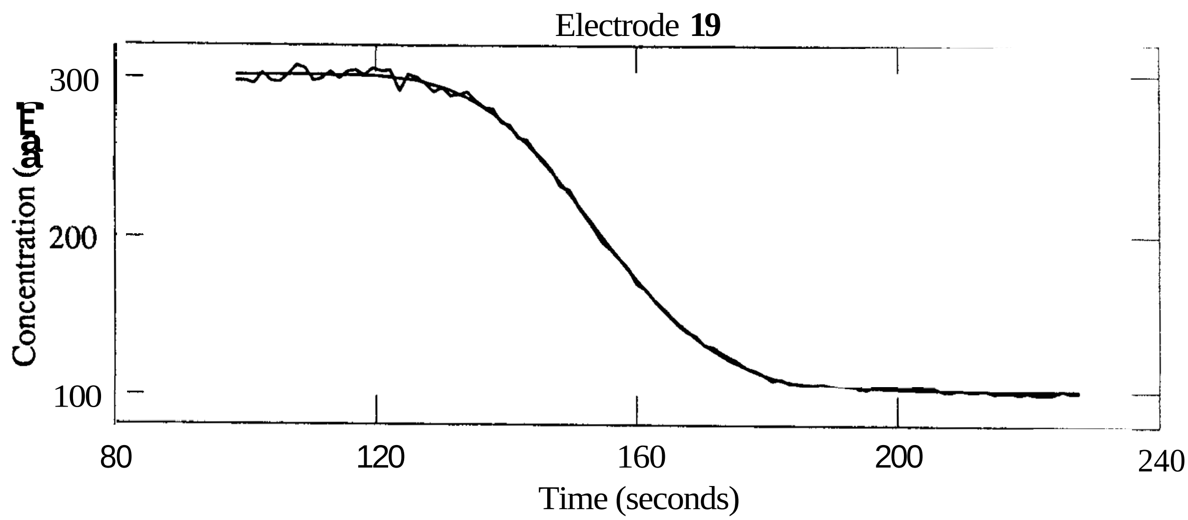
Electrode 18
C entration (ppm)
300
200
100
80
120
160
200
240
Time (seconds)



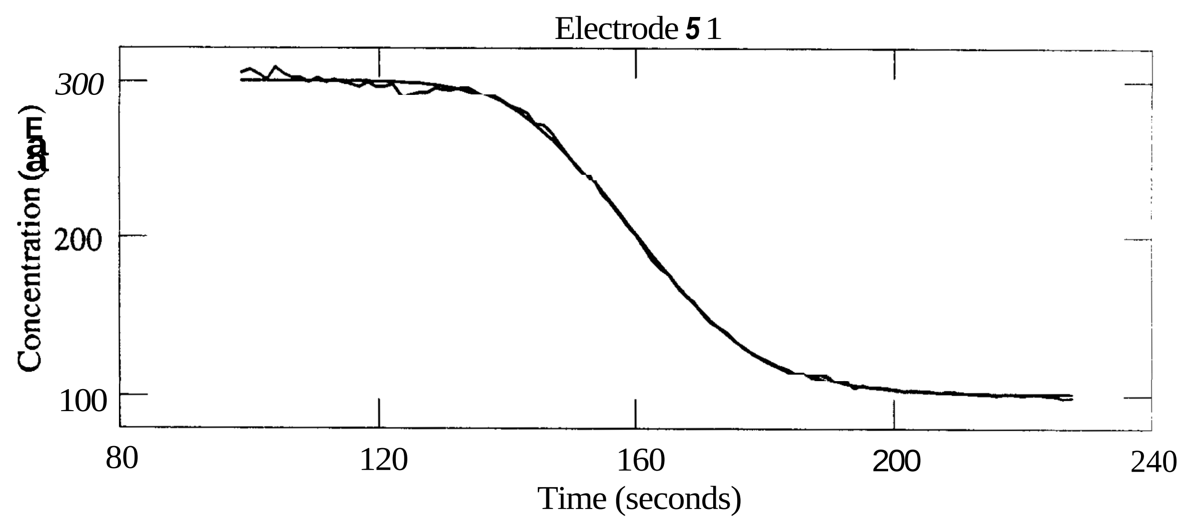
Electrode 50
Concentration (ppm)
300
200
100
80
120
160
200
240
Time (seconds)



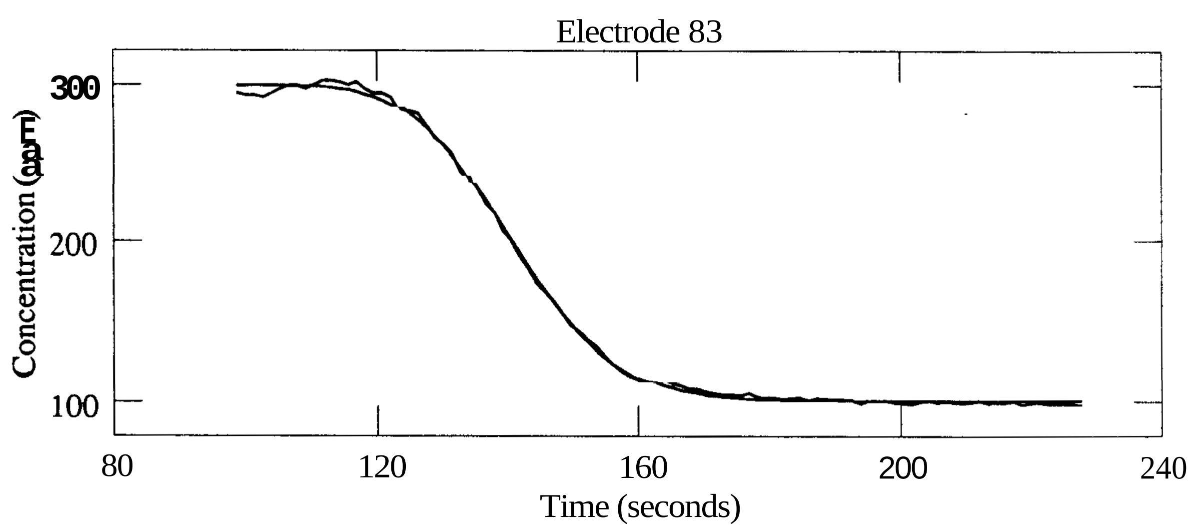
Electrode 82
Concentration (ppm)
300
200
100
80
120
160
200
240
Time (seconds)



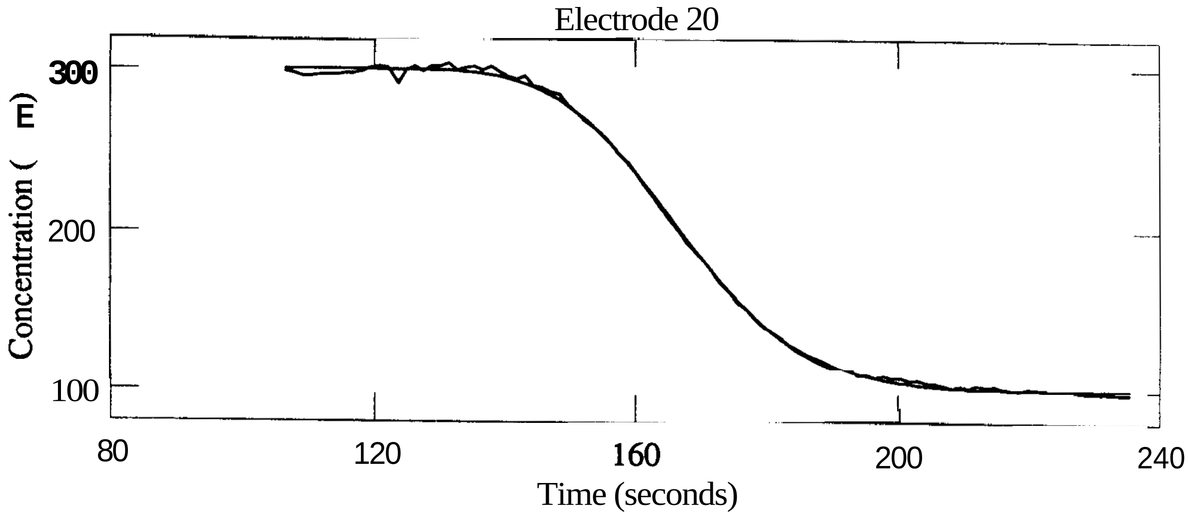
Electrode 19
300
200
100
Concentration (μm)
80
120
160
200
240
Time (seconds)

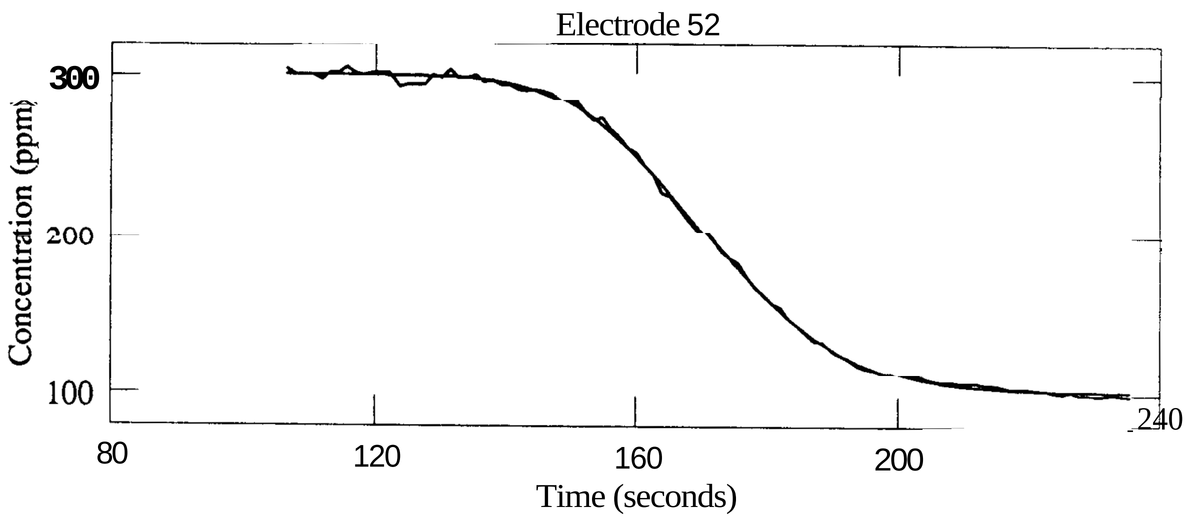


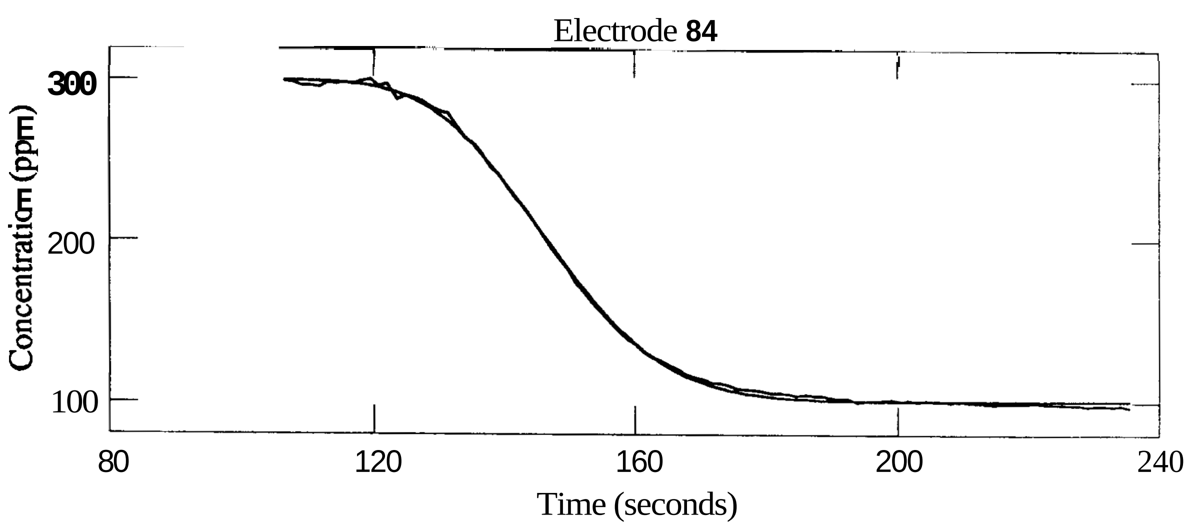
Electrode 51
300
200
100
Concentration (μm)
80
120
160
200
240
Time (seconds)

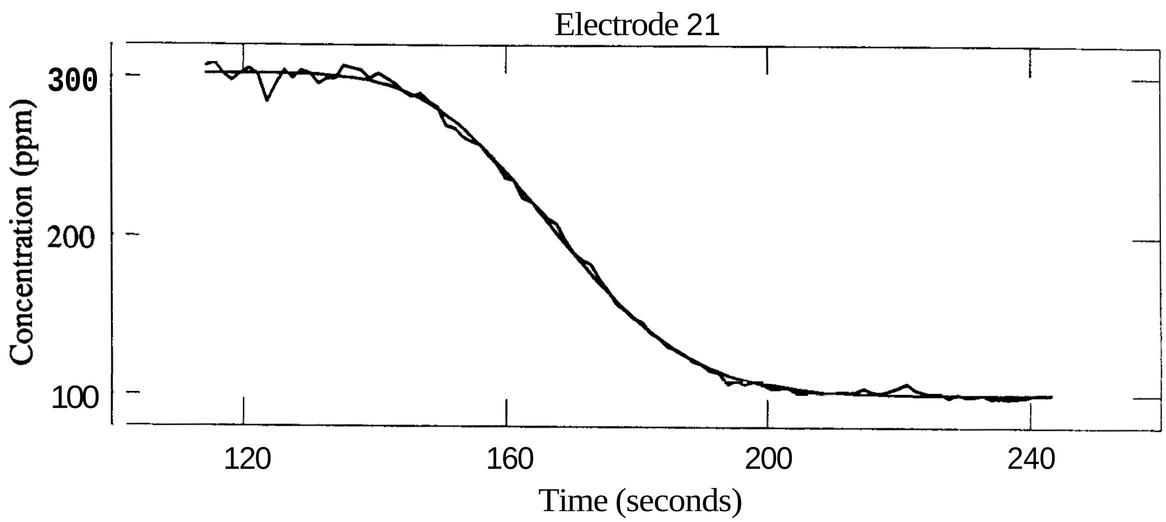


Electrode 83
300
200
100
Concentration (μm)
80
120
160
200
240
Time (seconds)

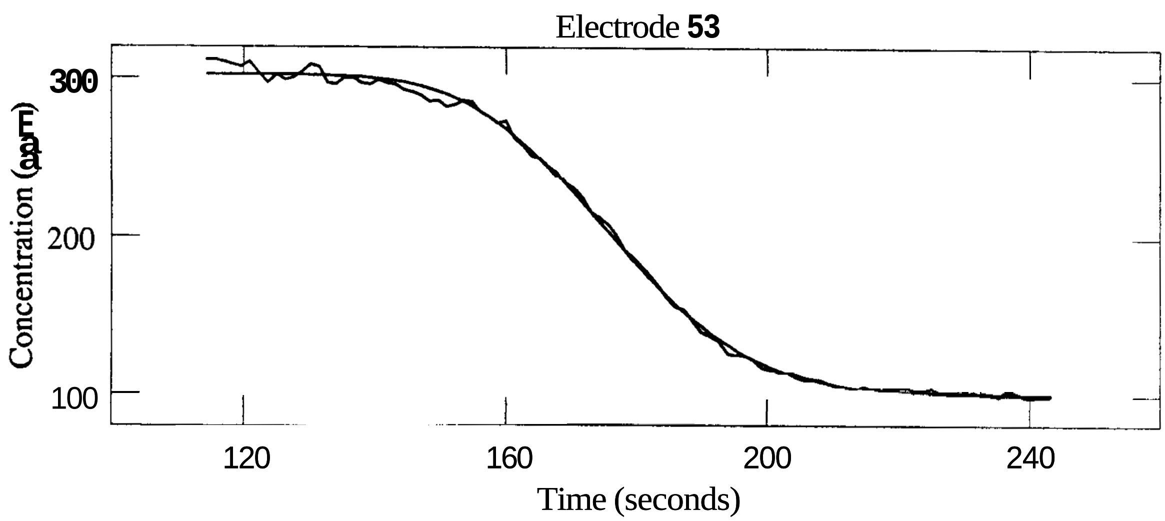

Electrode 20
300
200
100
Concentration (m)
80
120
160
200
240
Time (seconds)


Electrode 52
300
200
100
Concentration (ppm)
80
120
160
200
240
Time (seconds)

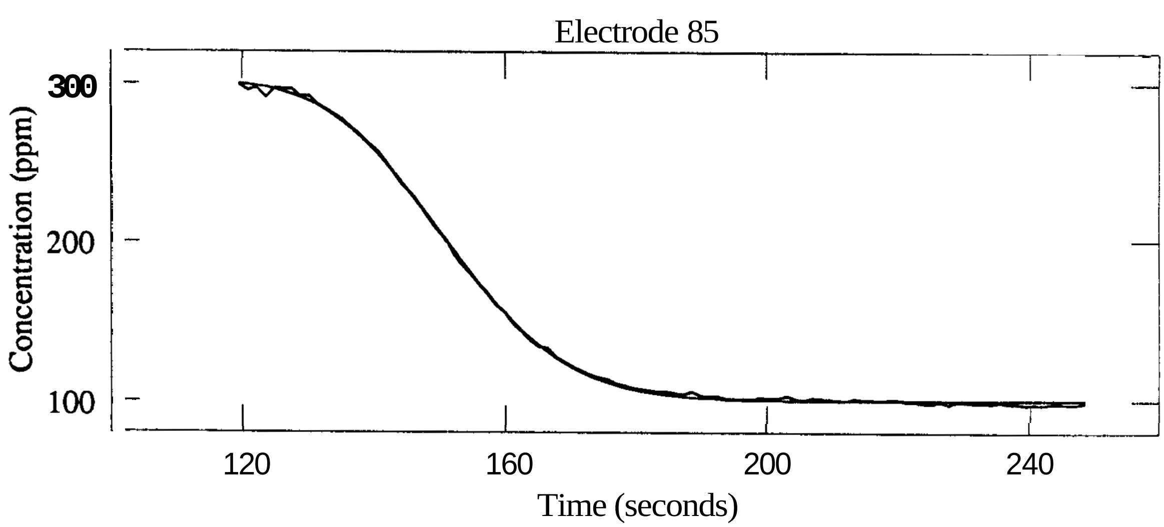

Electrode 84
300
200
100
Concentration (ppm)
80
120
160
200
240
Time (seconds)



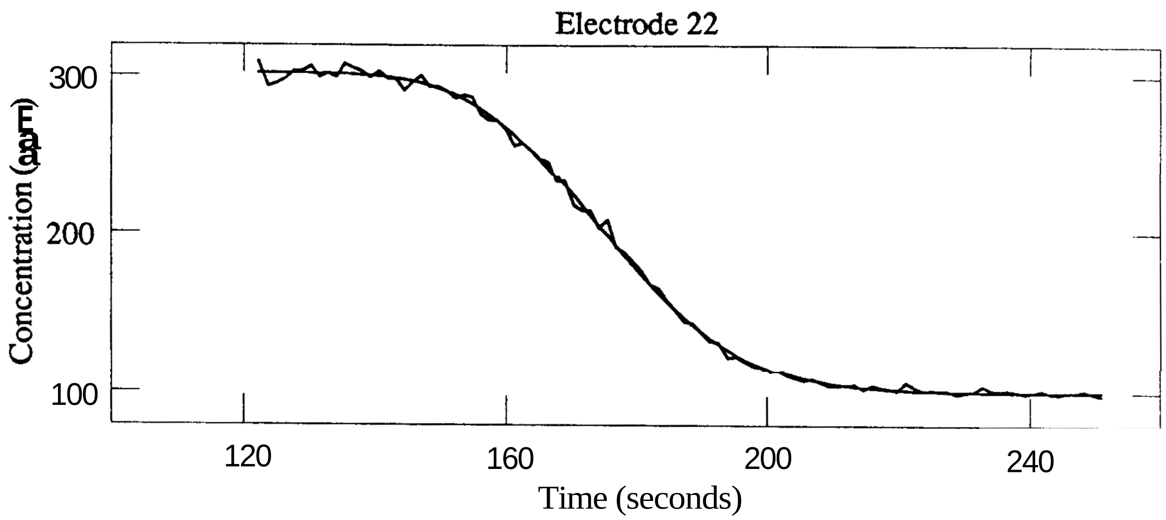
Electrode 21
300
200
100
Concentration (ppm)
120
160
200
240
Time (seconds)

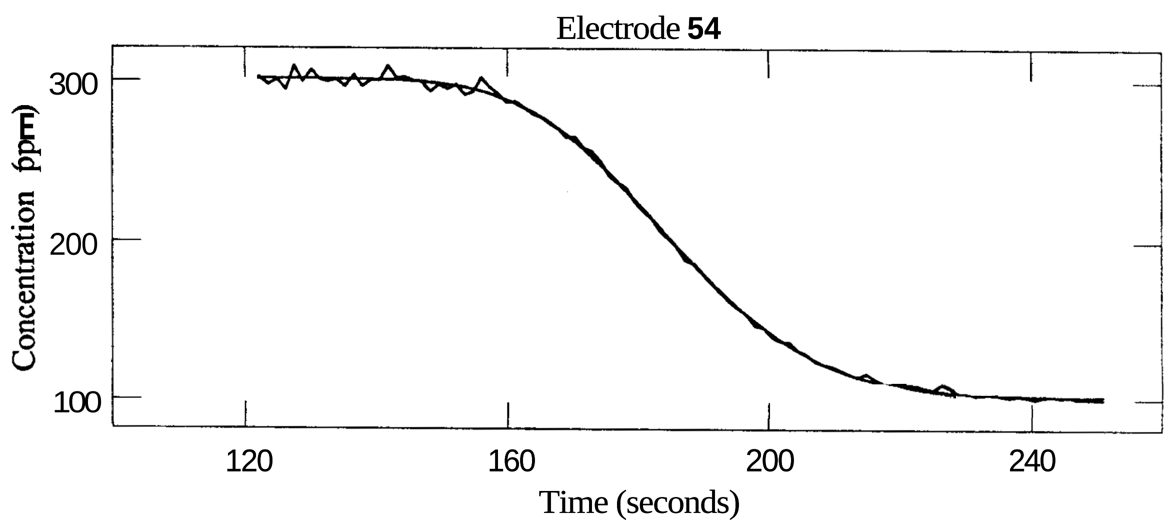


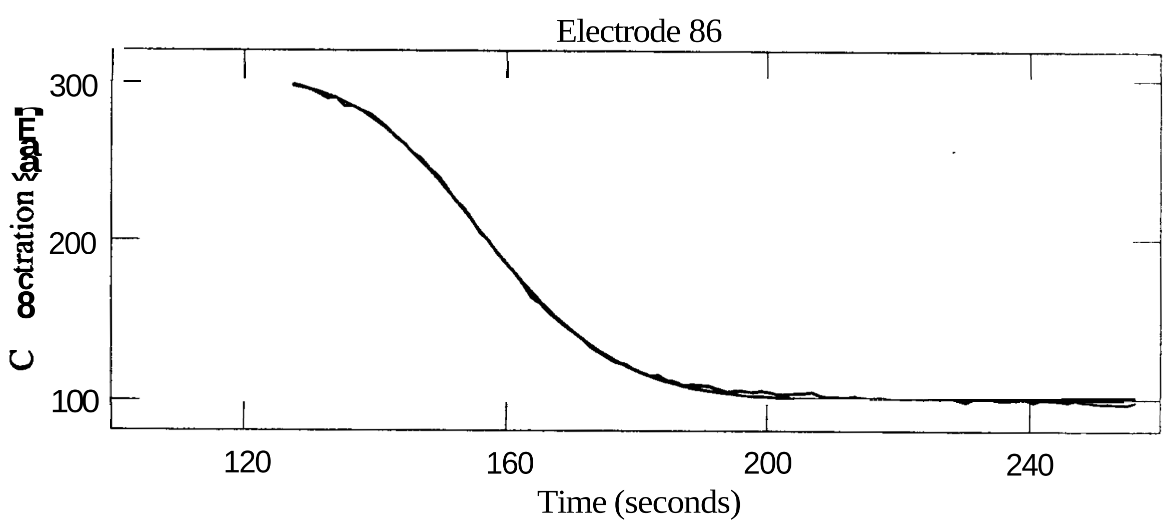
Electrode 53
300
200
100
Concentration (ppm)
120
160
200
240
Time (seconds)



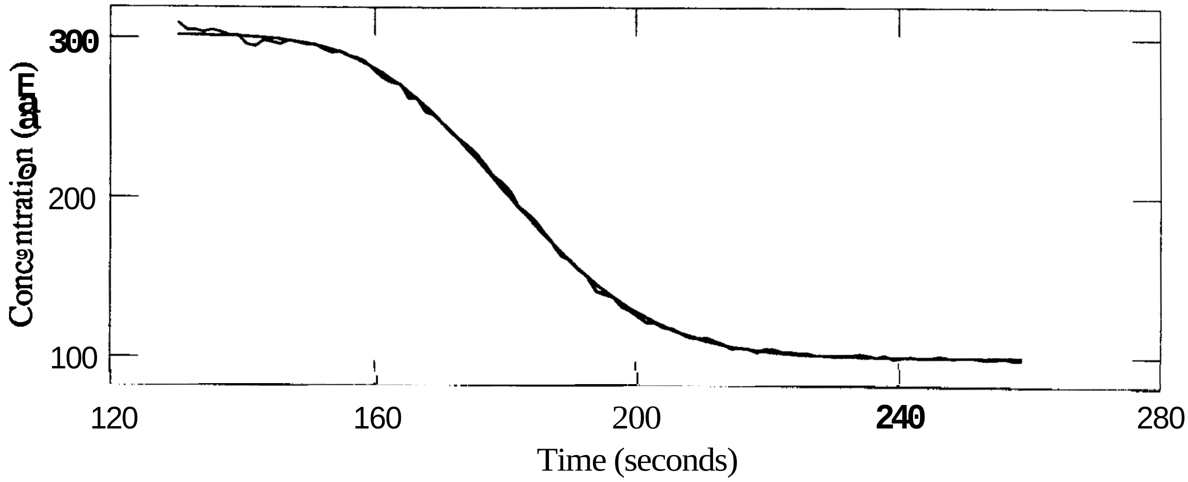
Electrode 85
300
200
100
Concentration (ppm)
120
160
200
240
Time (seconds)


Electrode 22
300
200
100
Concentration (ppm)
120
160
200
240
Time (seconds)

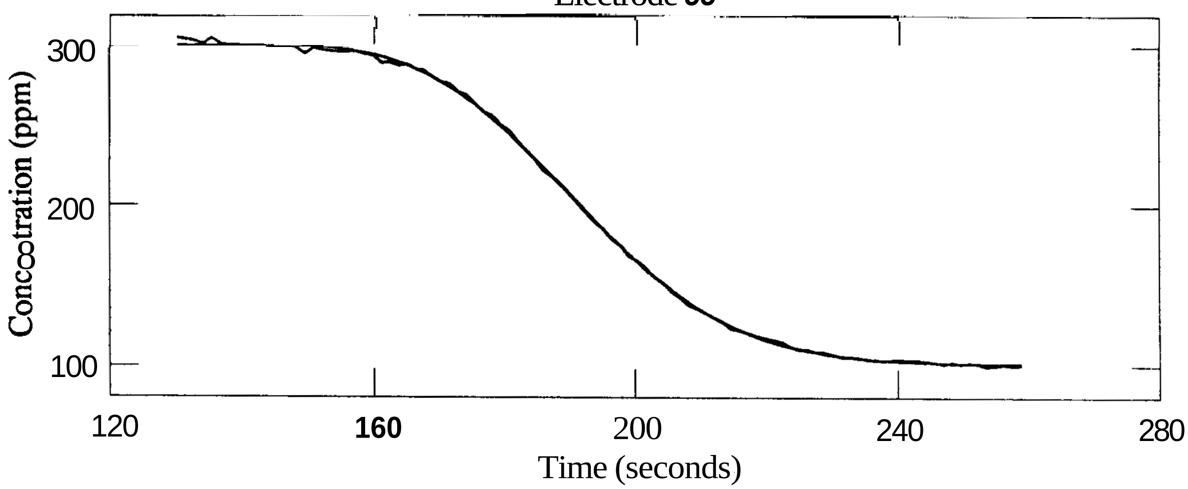

Electrode 54
300
200
100
Concentration (ppm)
120
160
200
240
Time (seconds)


Electrode 86
300
200
100
120
160
200
240
Time (seconds)

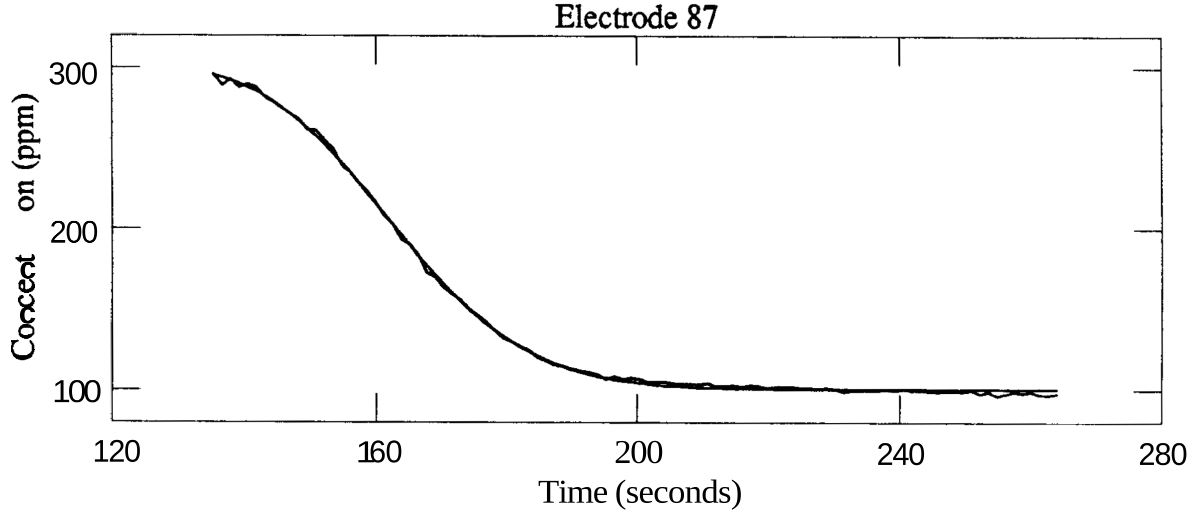
Electrode 23

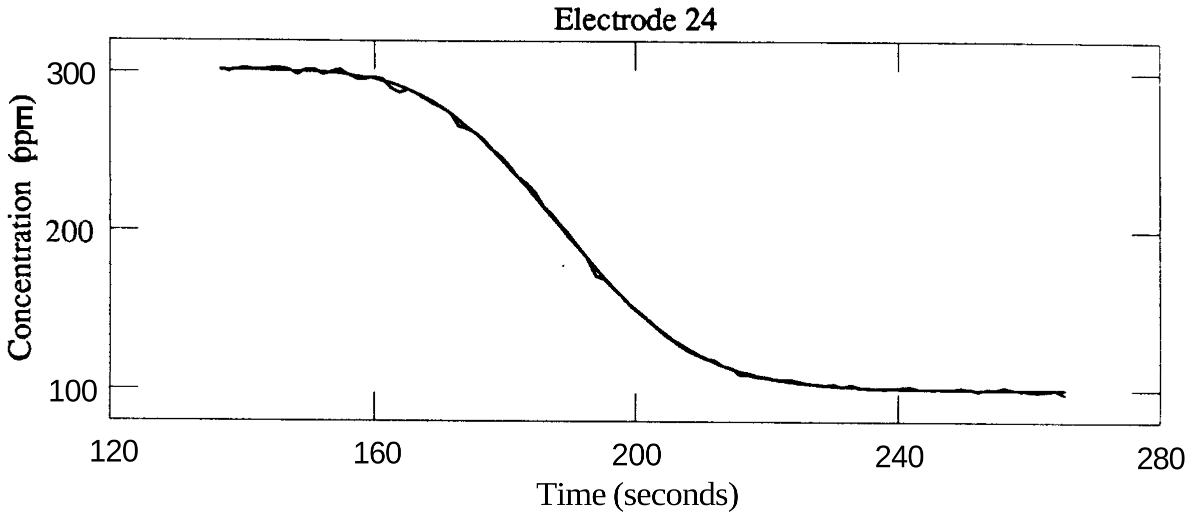


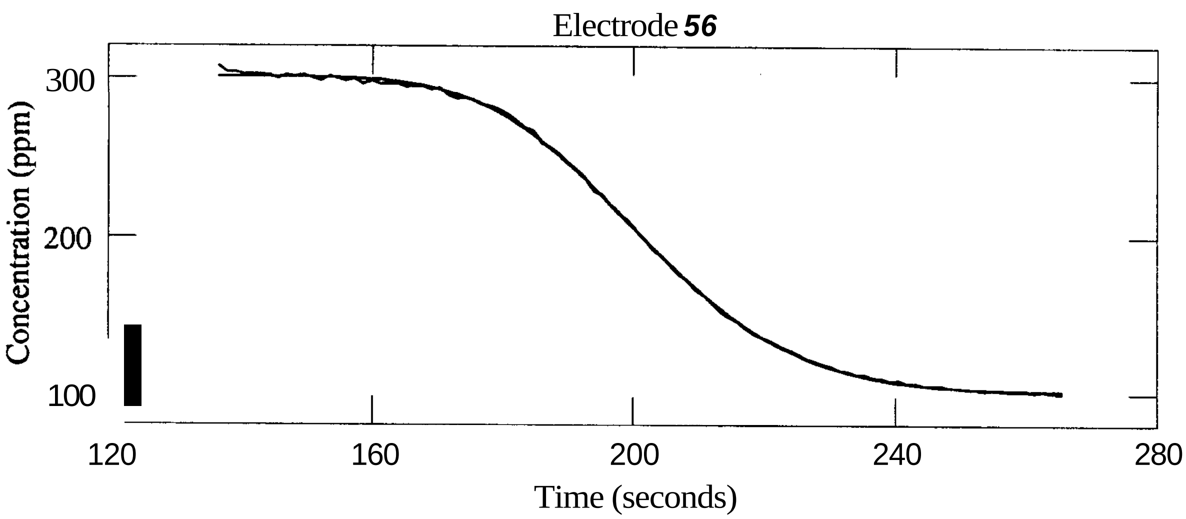
Electrode 55

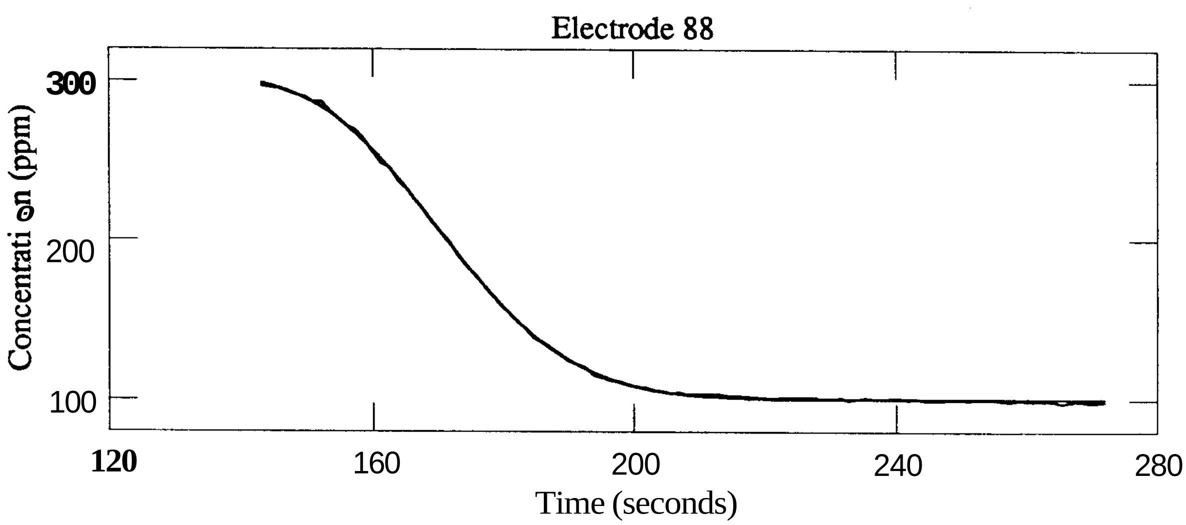


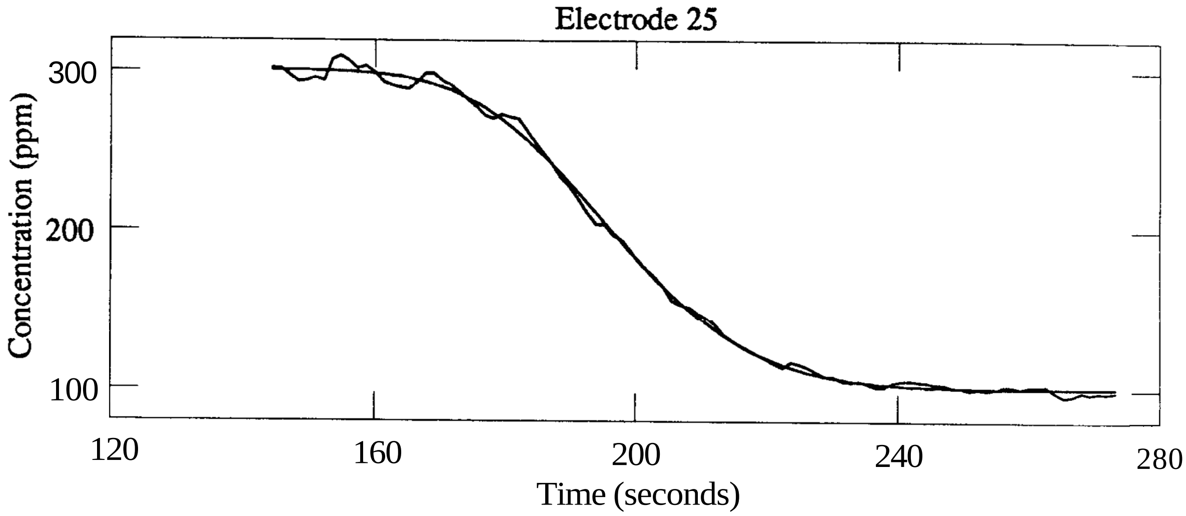
Electrode 87

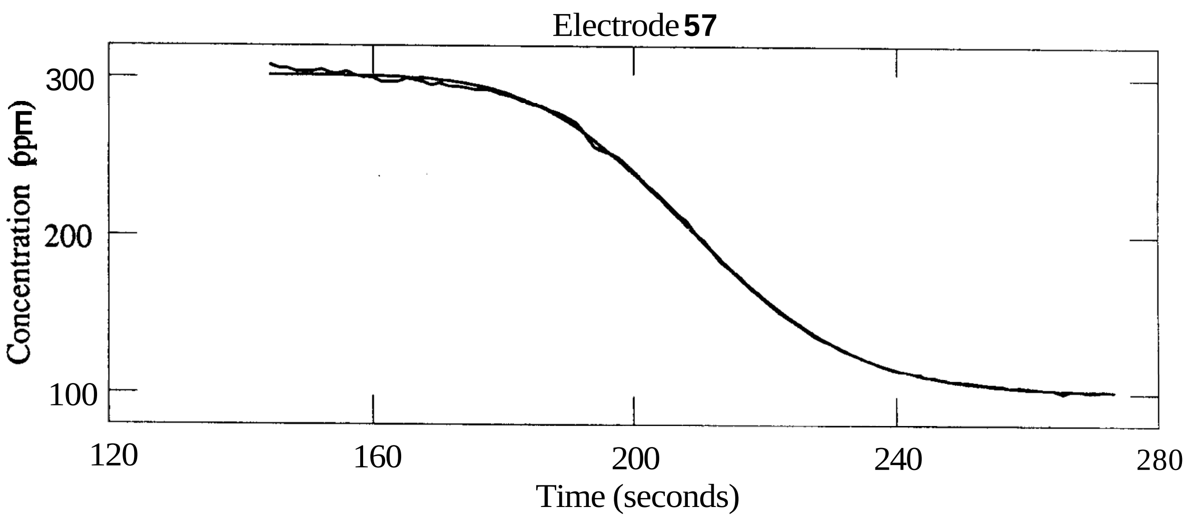


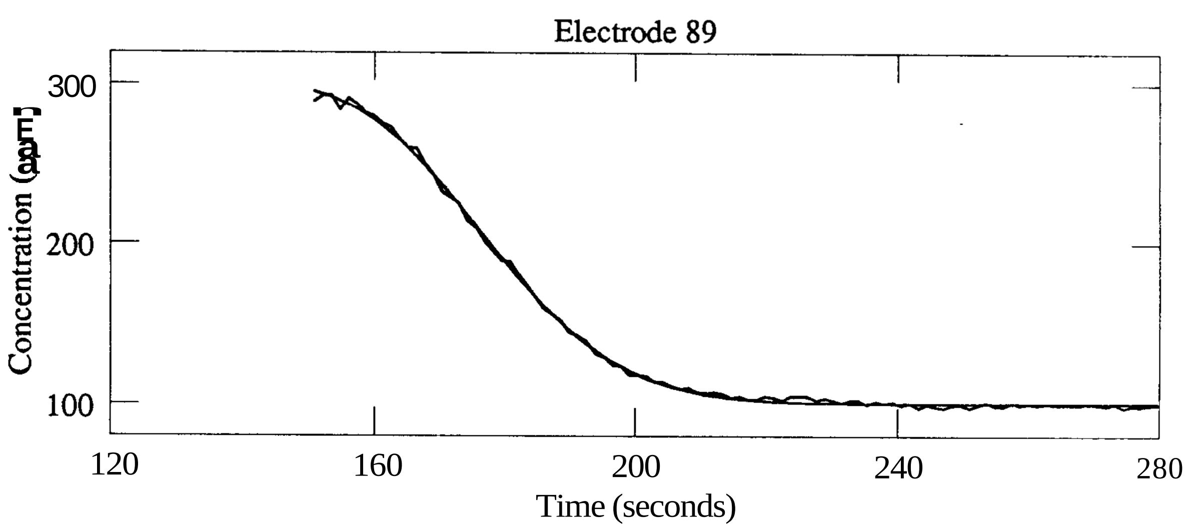

Electrode 24
Concentration (ppm)
300
200
100
120
160
200
240
280
Time (seconds)

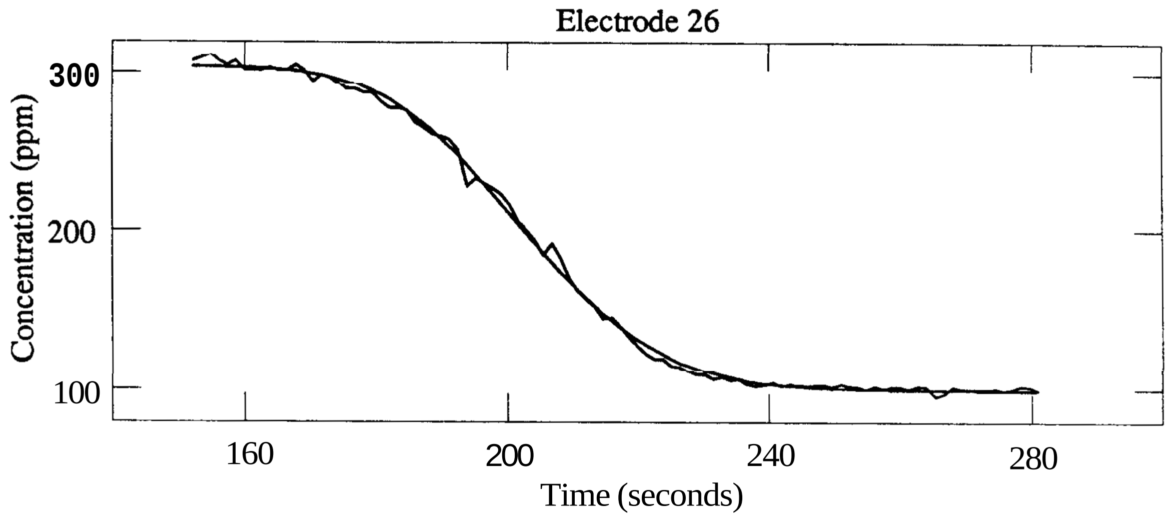

Electrode 56
Concentration (ppm)
300
200
100
120
160
200
240
280
Time (seconds)

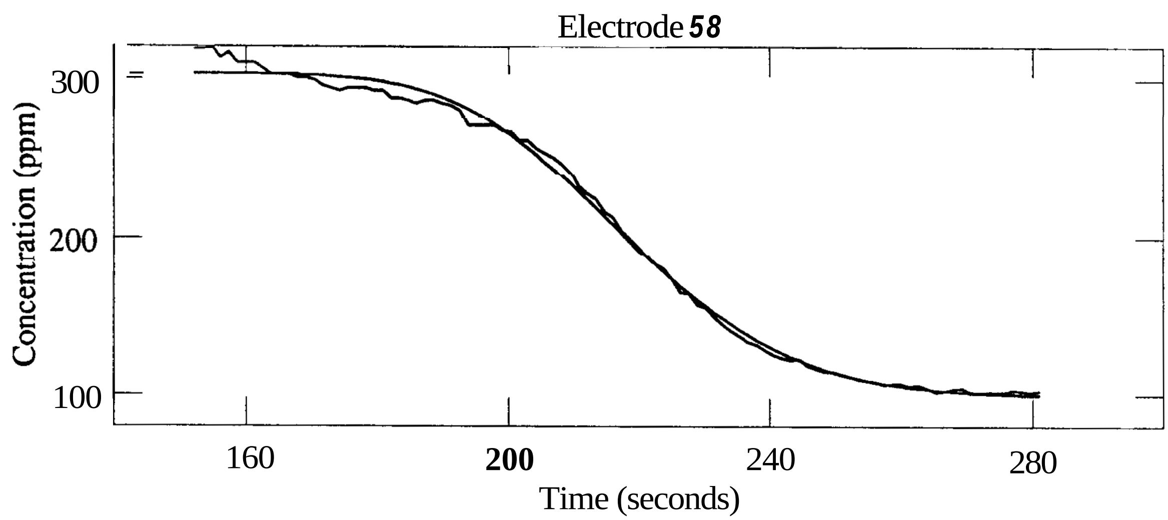

Electrode 88
Concentration (ppm)
300
200
100
120
160
200
240
280
Time (seconds)

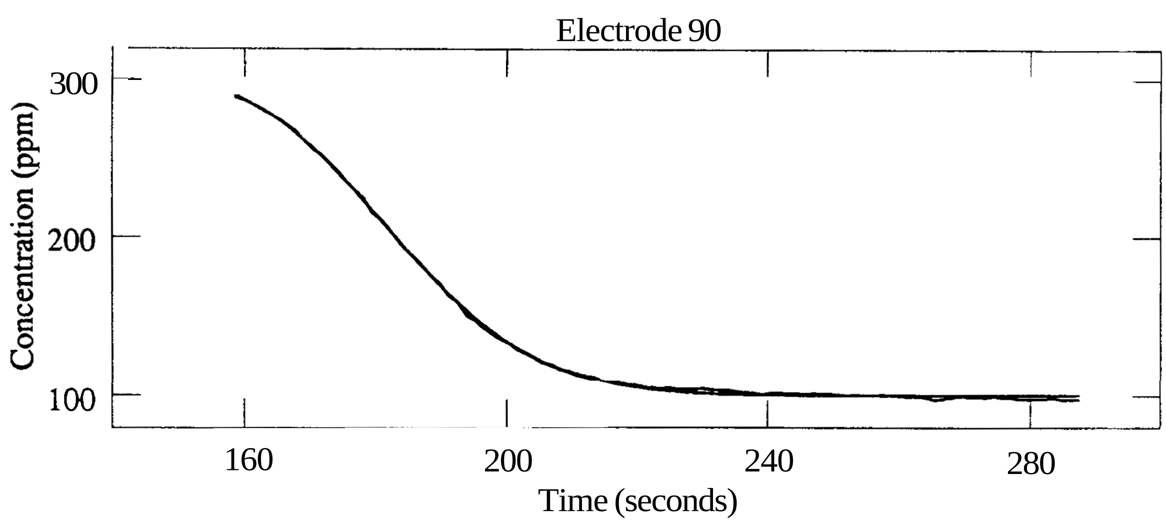

Electrode 25
Concentration (ppm)
300
200
100
120
160
200
240
280
Time (seconds)

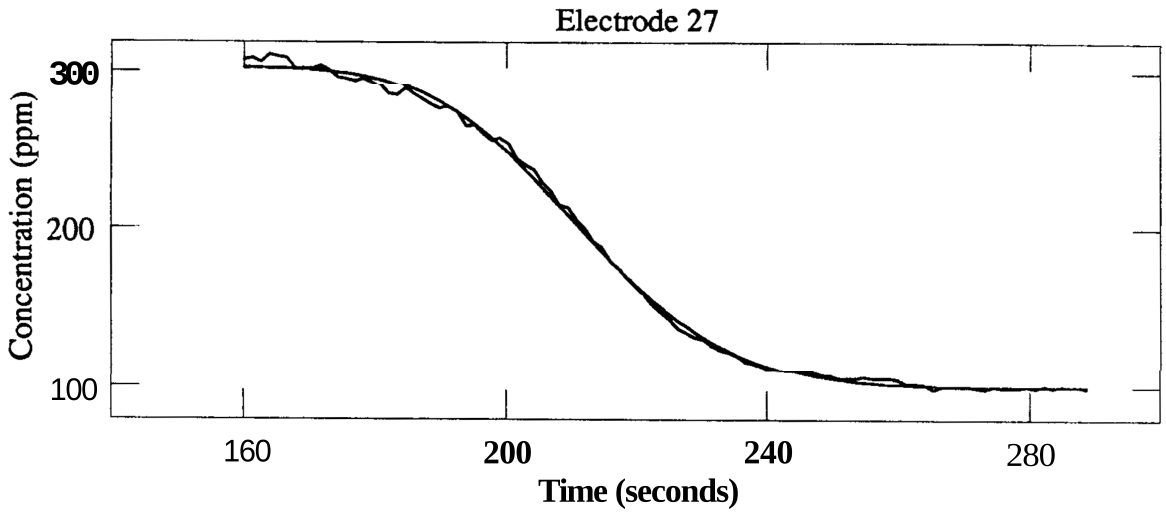

Electrode 57
Concentration (ppm)
300
200
100
120
160
200
240
280
Time (seconds)

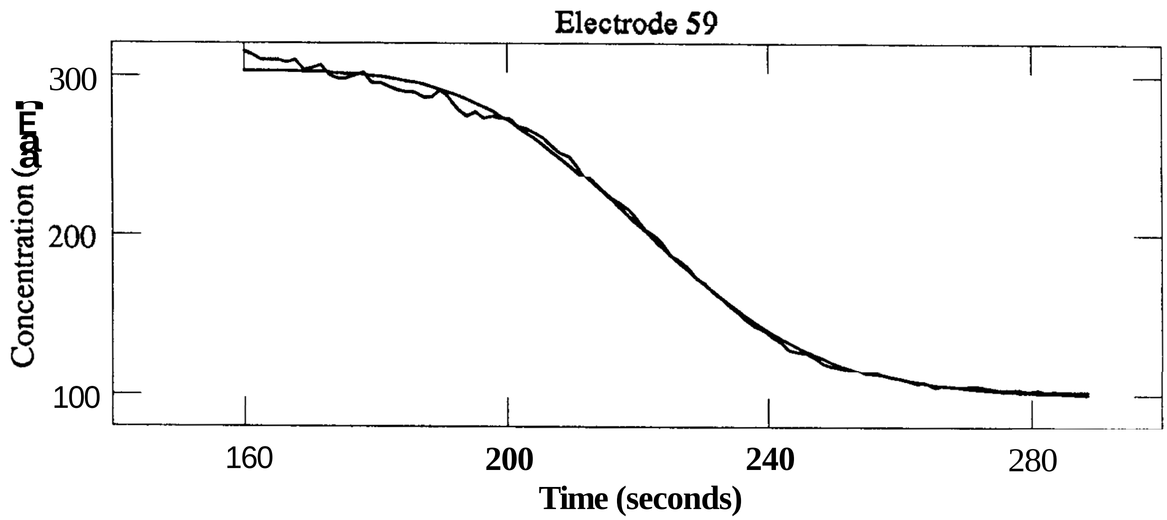

Electrode 89
Concentration (ppm)
300
200
100
120
160
200
240
280
Time (seconds)

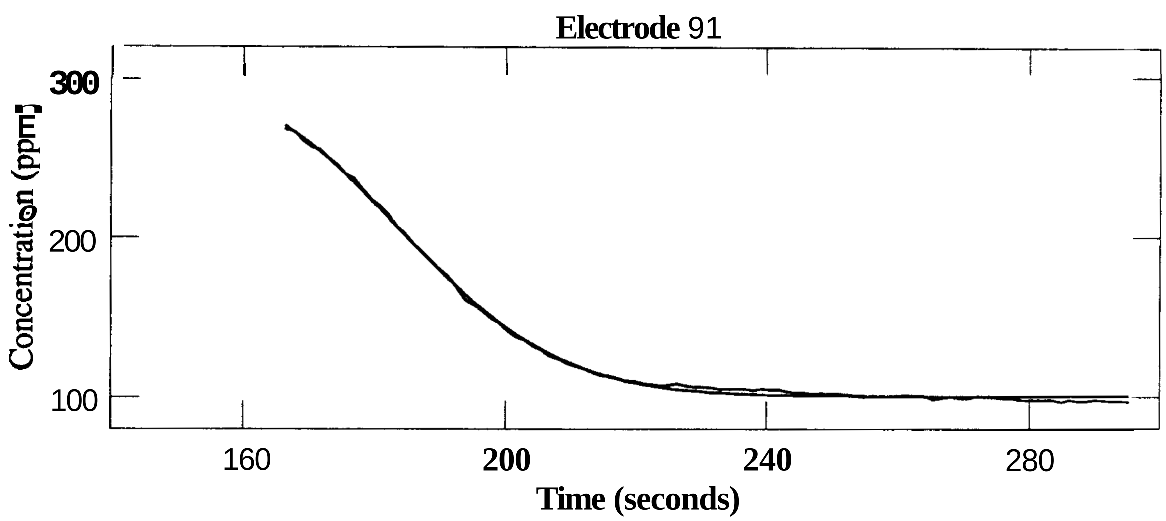

Electrode 26
300
200
100
Concentration (ppm)
160
200
240
280
Time (seconds)

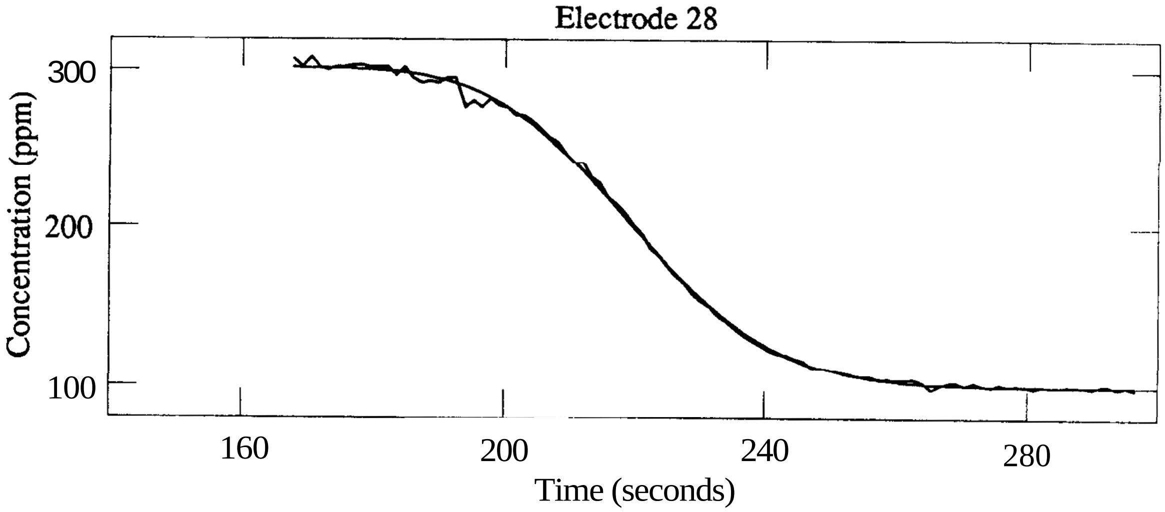

Electrode 58
300
200
100
Concentration (ppm)
160
200
240
280
Time (seconds)

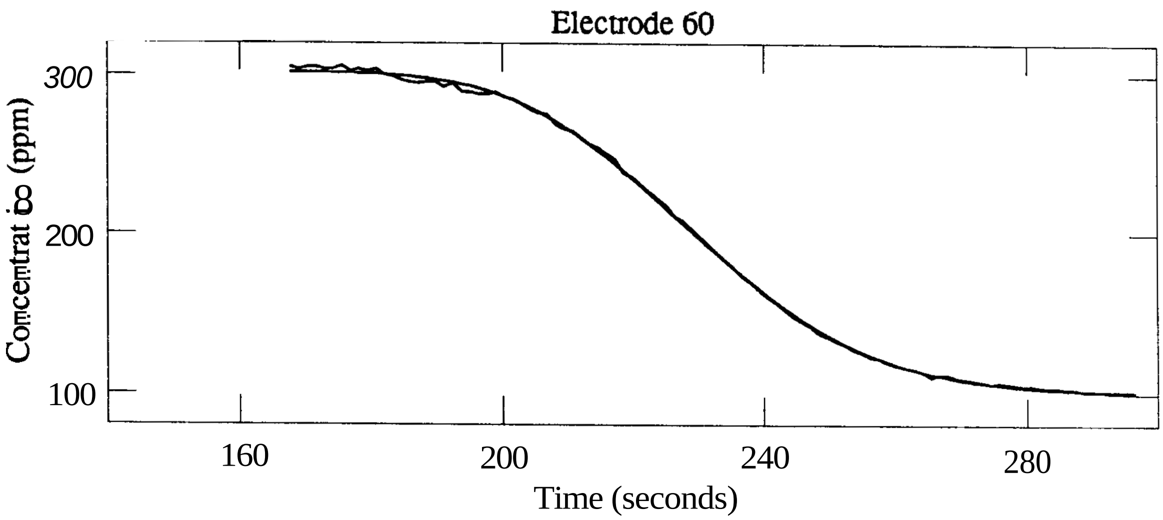

Electrode 90
300
200
100
Concentration (ppm)
160
200
240
280
Time (seconds)

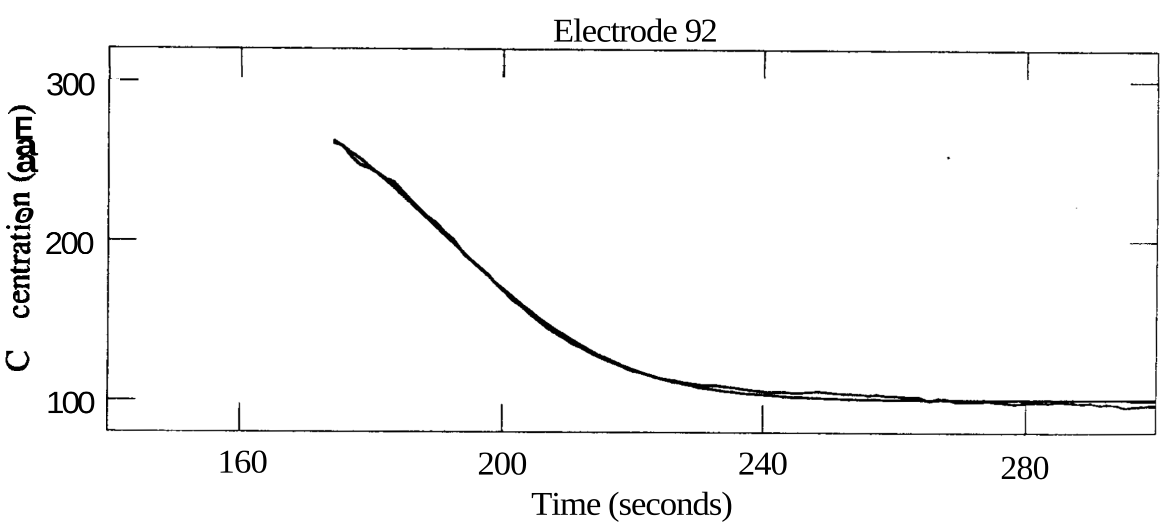

Electrode 27
Concentration (ppm)
300
200
100
160
200
240
280
Time (seconds)

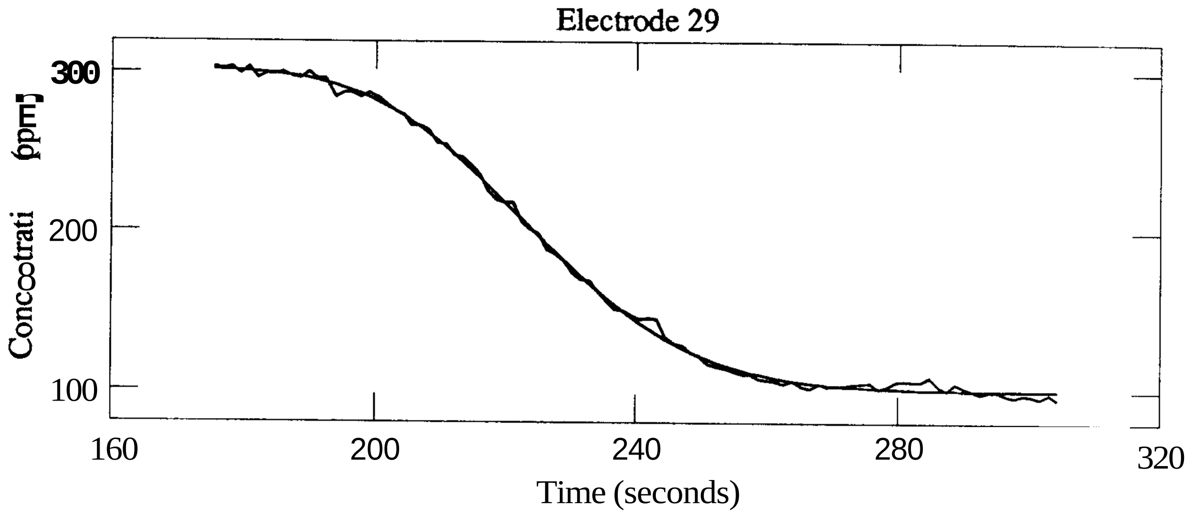

Electrode 59
Concentration (ppm)
300
200
100
160
200
240
280
Time (seconds)

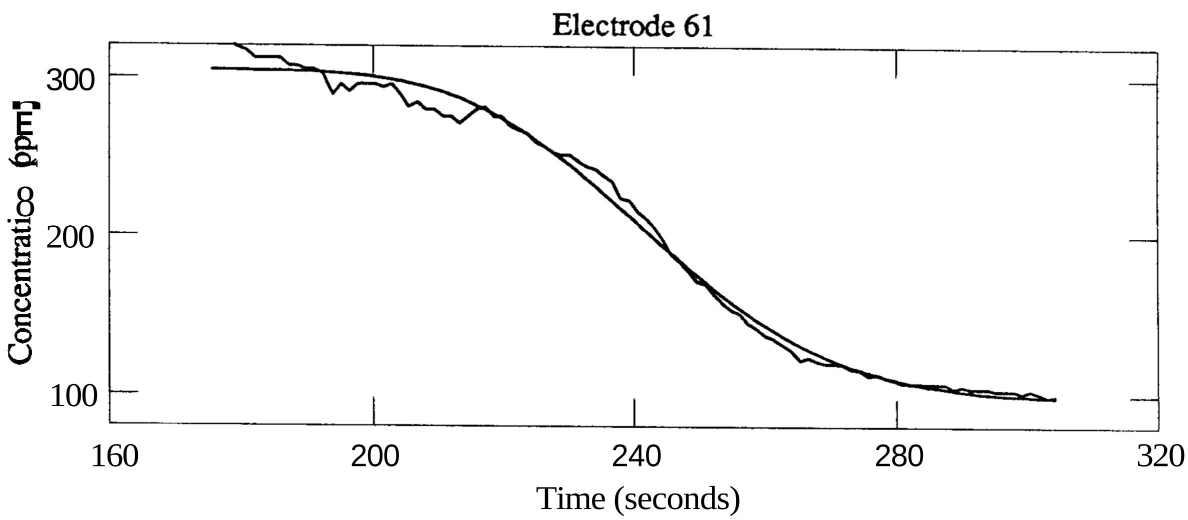

Electrode 91
Concentration (ppm)
300
200
100
160
200
240
280
Time (seconds)

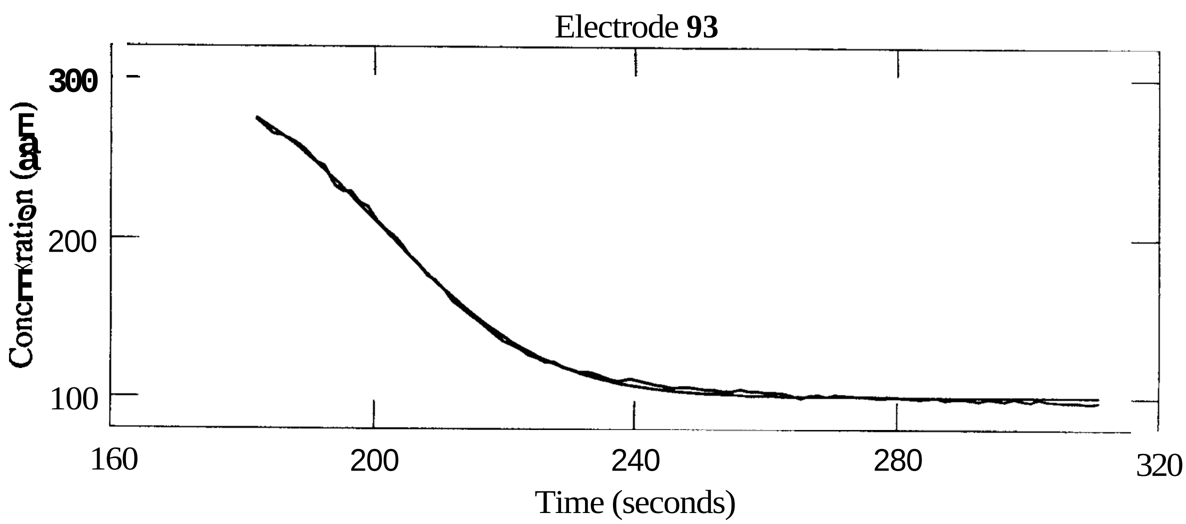

Electrode 28
Concentration (ppm)
300
200
100
160
200
240
280
Time (seconds)

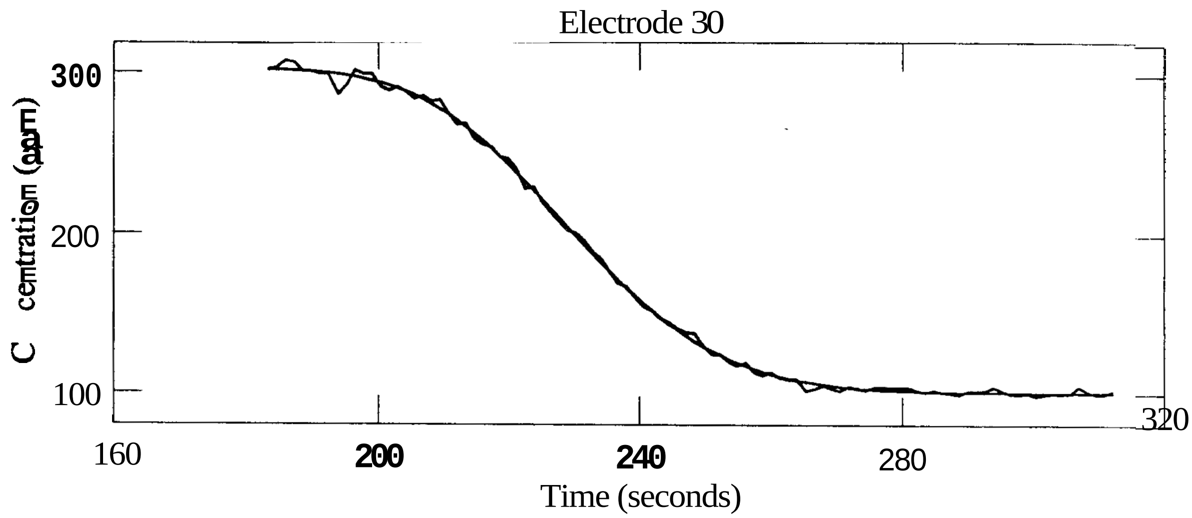

Electrode 60
Concentration (ppm)
300
200
100
160
200
240
280
Time (seconds)

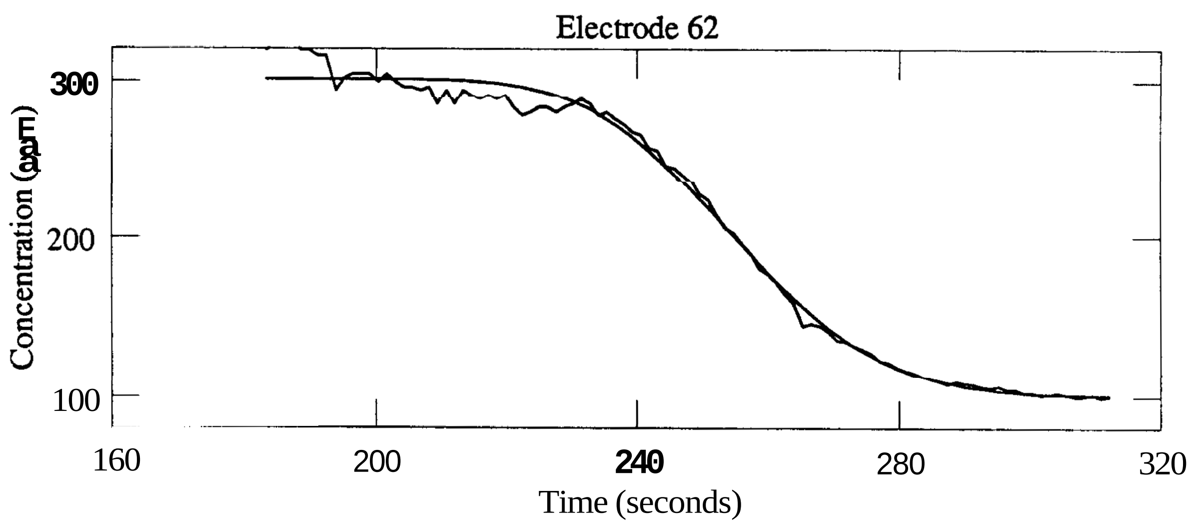

Electrode 92
300
200
100
160
200
240
280
Time (seconds)

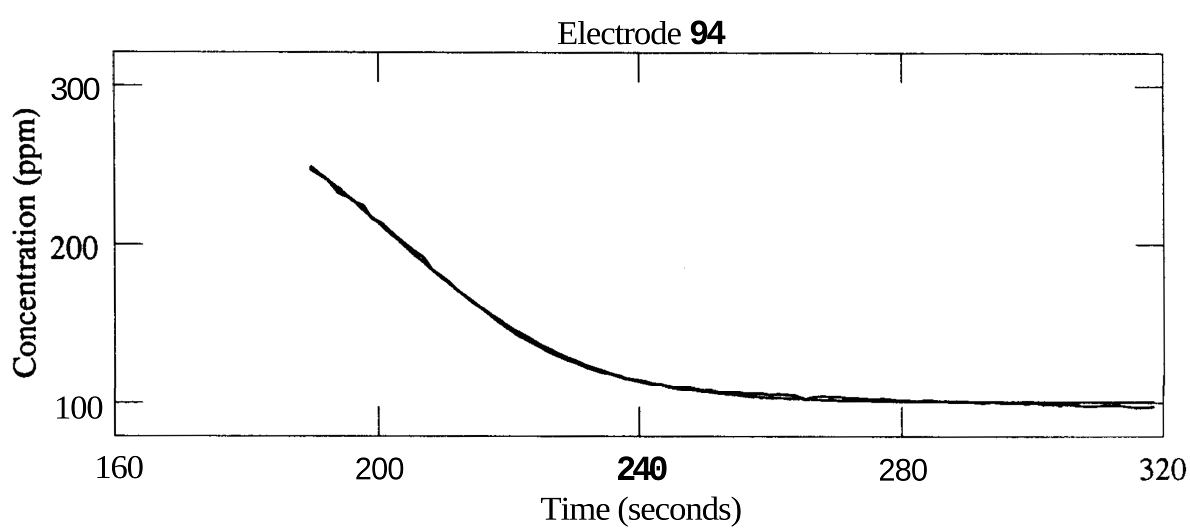

Electrode 29
300
200
100
160
200
240
280
320
Time (seconds)

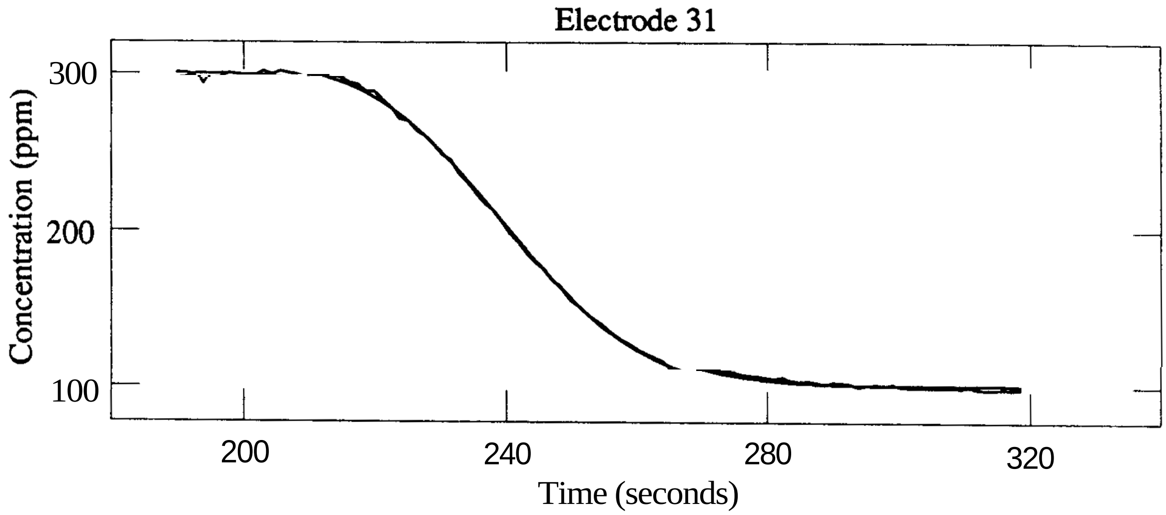

Electrode 61
300
200
100
160
200
240
280
320
Time (seconds)

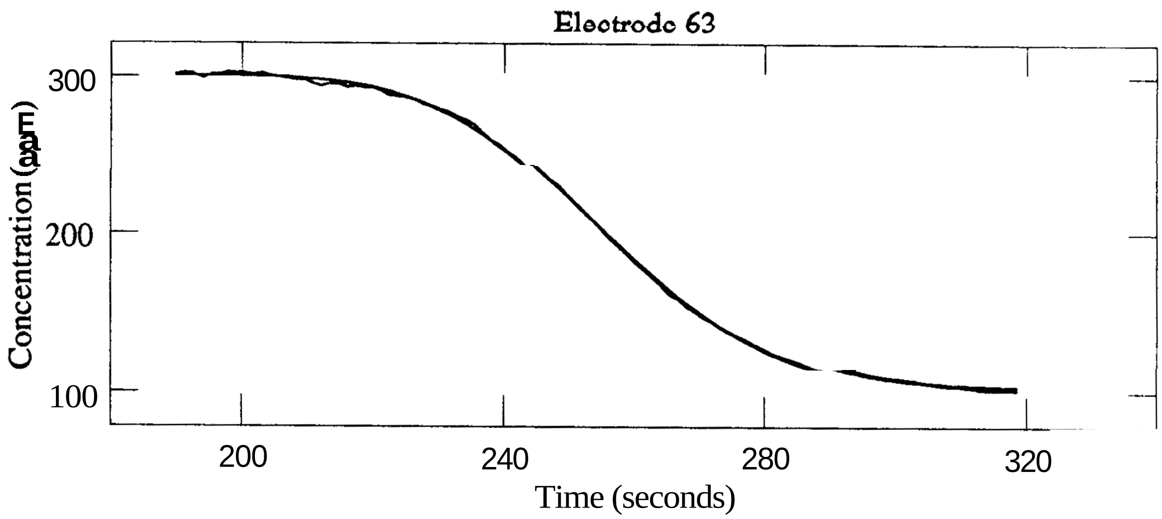

Electrode 93
300
200
100
160
200
240
280
320
Time (seconds)

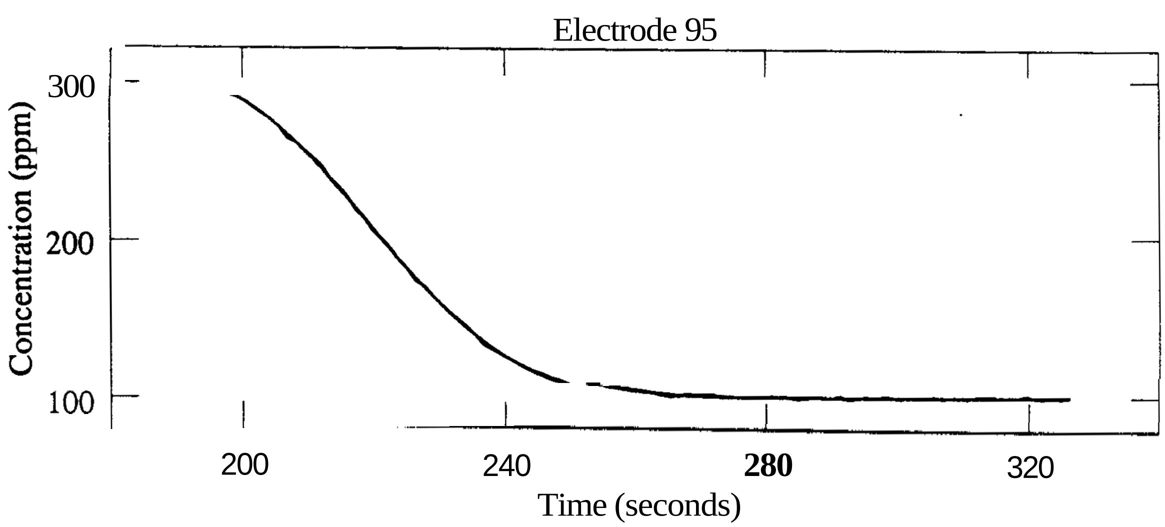

Electrode 30
Concentration (ppm)
300
200
100
160
200
240
280
320
Time (seconds)

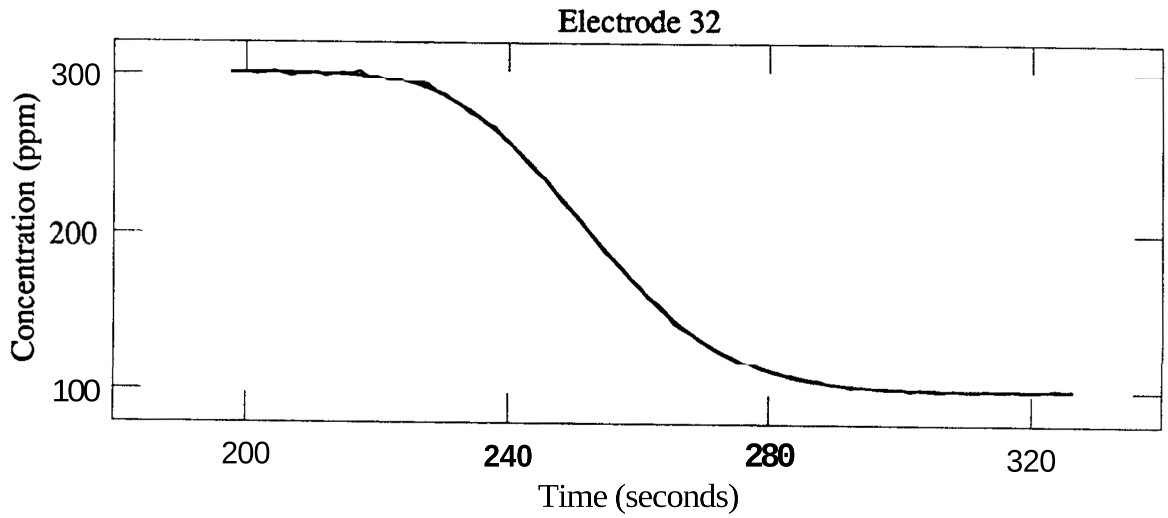

Electrode 62
Concentration (ppm)
300
200
100
160
200
240
280
320
Time (seconds)

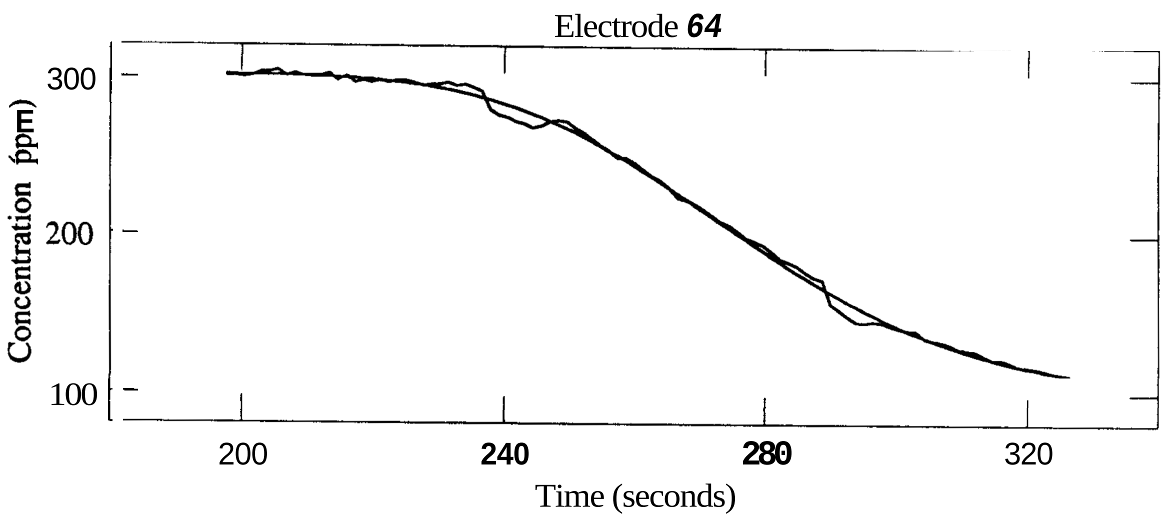

Electrode 94
Concentration (ppm)
300
200
100
160
200
240
280
320
Time (seconds)

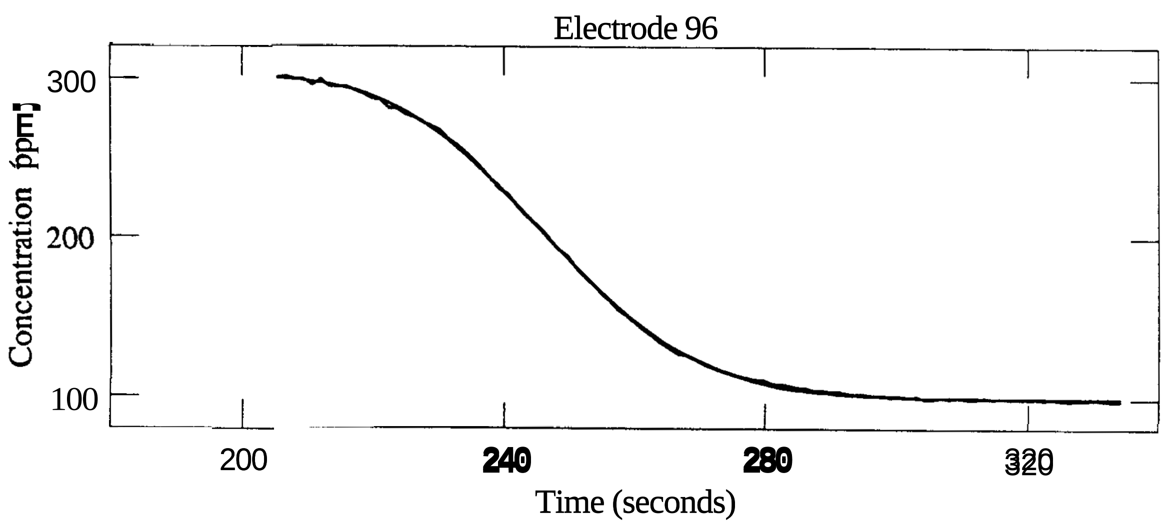

Electrode 31
300
200
100
Concentration (ppm)
200
240
280
320
Time (seconds)


Electrode 63
300
200
100
Concentration (ppm)
200
240
280
320
Time (seconds)


Electrode 95
300
200
100
Concentration (ppm)
200
240
280
320
Time (seconds)

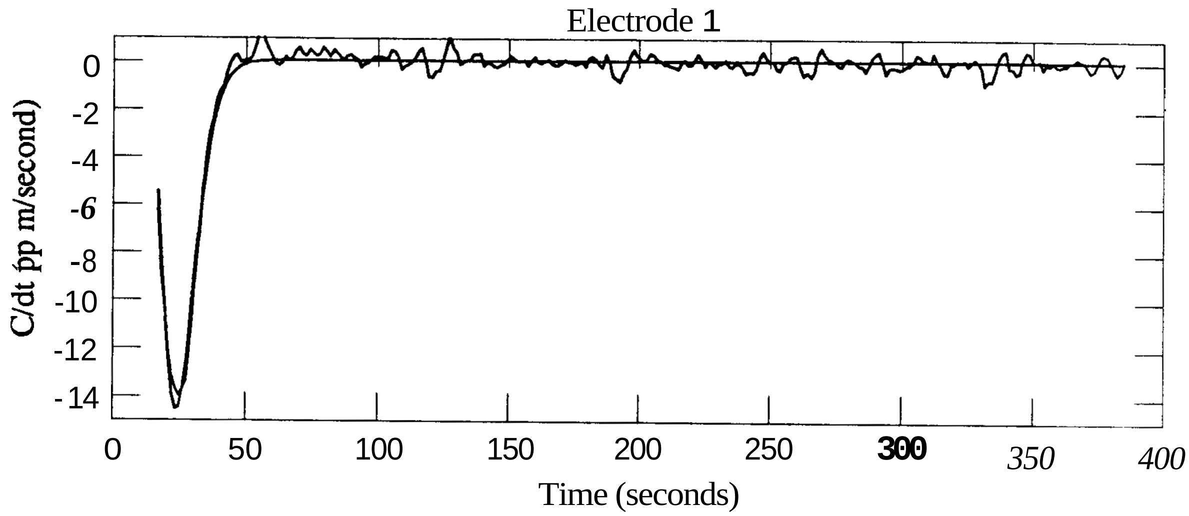

Electrode 32
Concentration (ppm)
300
200
100
200
240
280
320
Time (seconds)

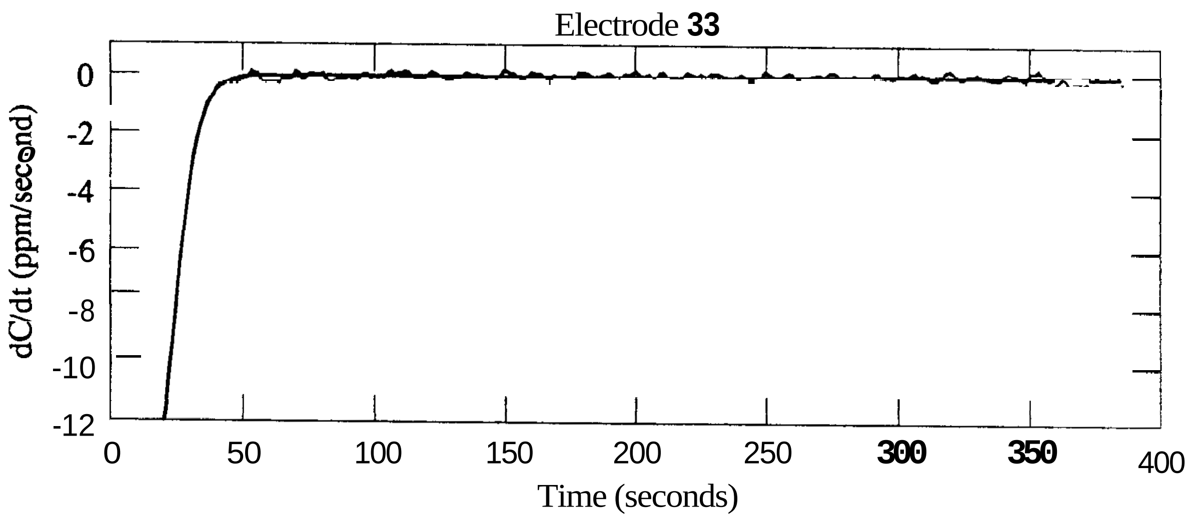

Electrode 64
Concentration (ppm)
300
200
100
200
240
280
320
Time (seconds)

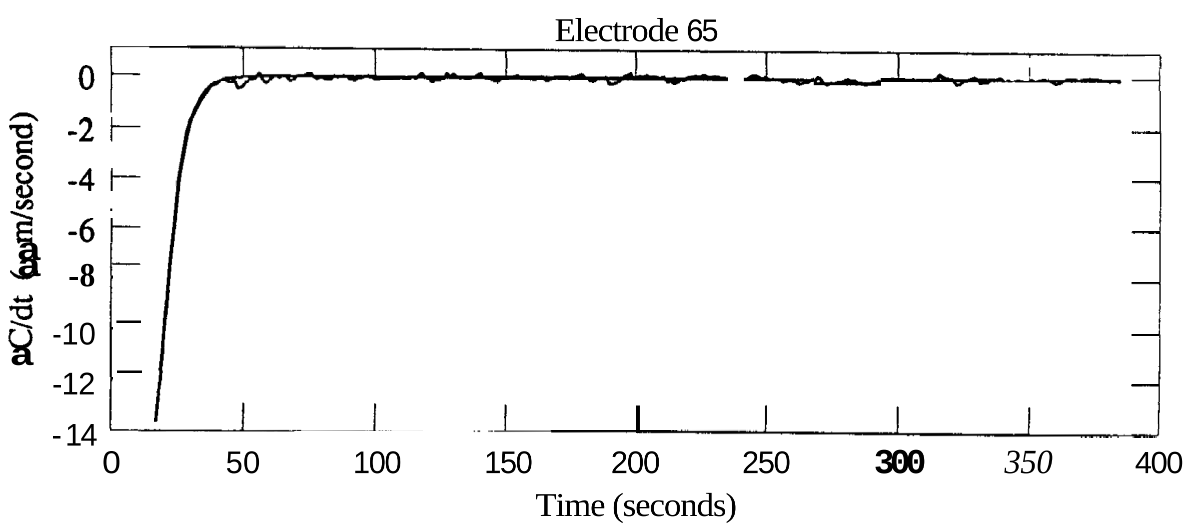

Electrode 96
Concentration (ppm)
300
200
100
200
240
280
320
Time (seconds)

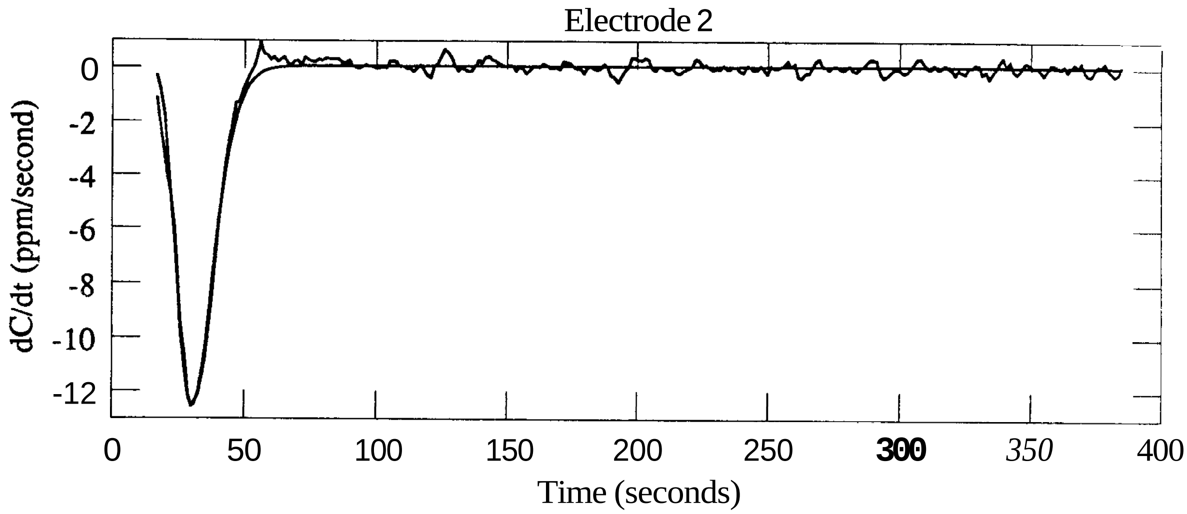
Appendix B: GRAPHS OF RUN 16 DERIVATIVE FITS

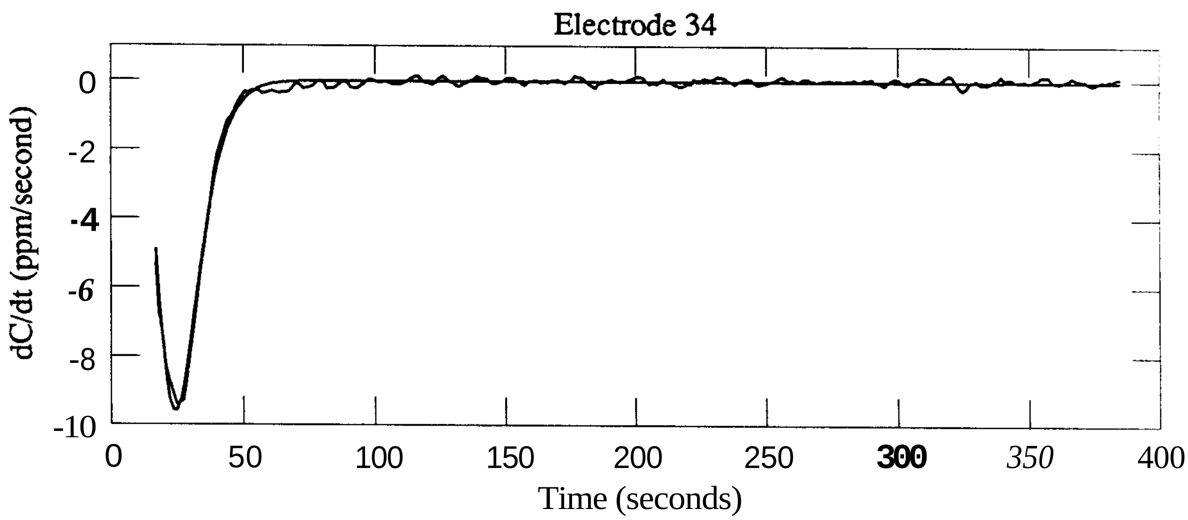
This appendix comprises a complete set of graphs of the curvefit matches of the derivative of Eq. 2 for Run 16. Each page is in effect a cross-section of the Hele-Shaw cell at a particular distance along the cell from the inlet valve. The electrodes are in three groups numbered 1-32, 33-64, and 65-96 along the cell from the valve. Recall that Table 4 lists the electrodes which were included in the final calculations for this run. Any electrode not included in Table 4 was eliminated from consideration. From the graphs in this appendix it is usually apparent why a particular electrode was not included.

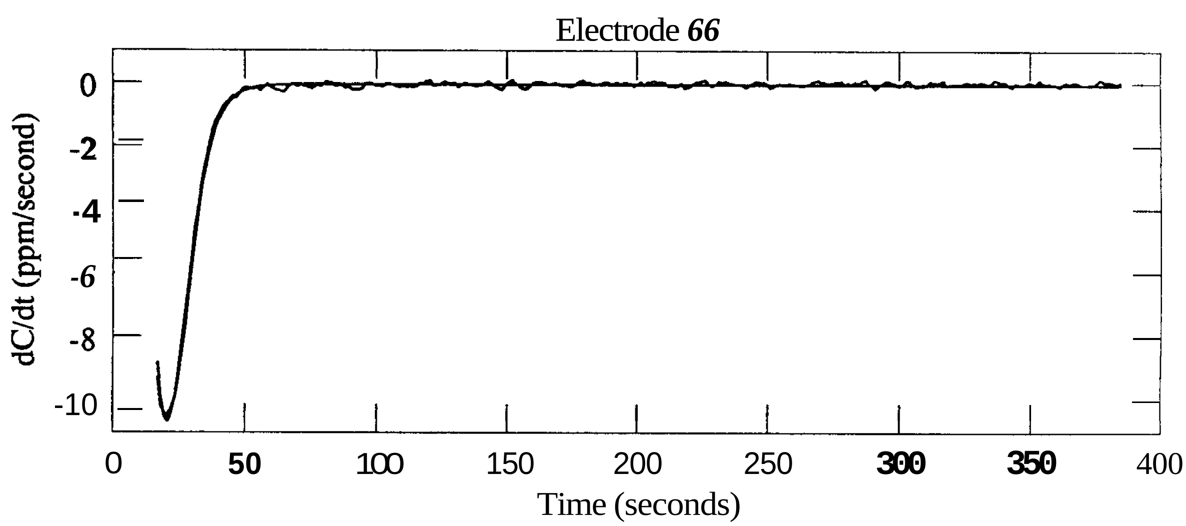

Electrode 1
C/dt (pp m/second)
0
-2
-4
-6
-8
-10
-12
-14
0
50
100
150
200
250
300
350
400
Time (seconds)

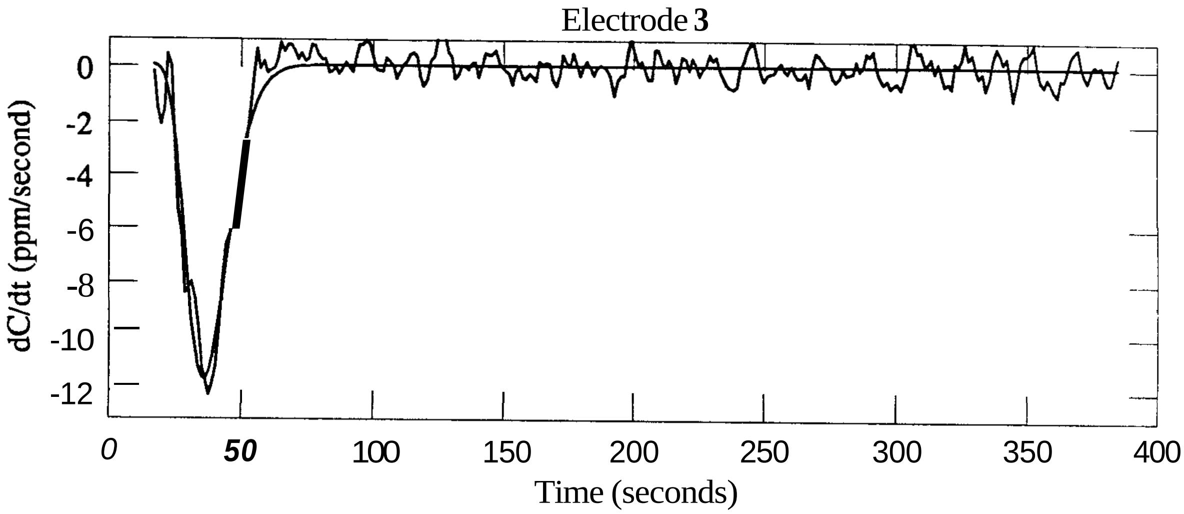

Electrode 33
dC/dt (ppm/second)
0
-2
-4
-6
-8
-10
-12
0
50
100
150
200
250
300
350
400
Time (seconds)

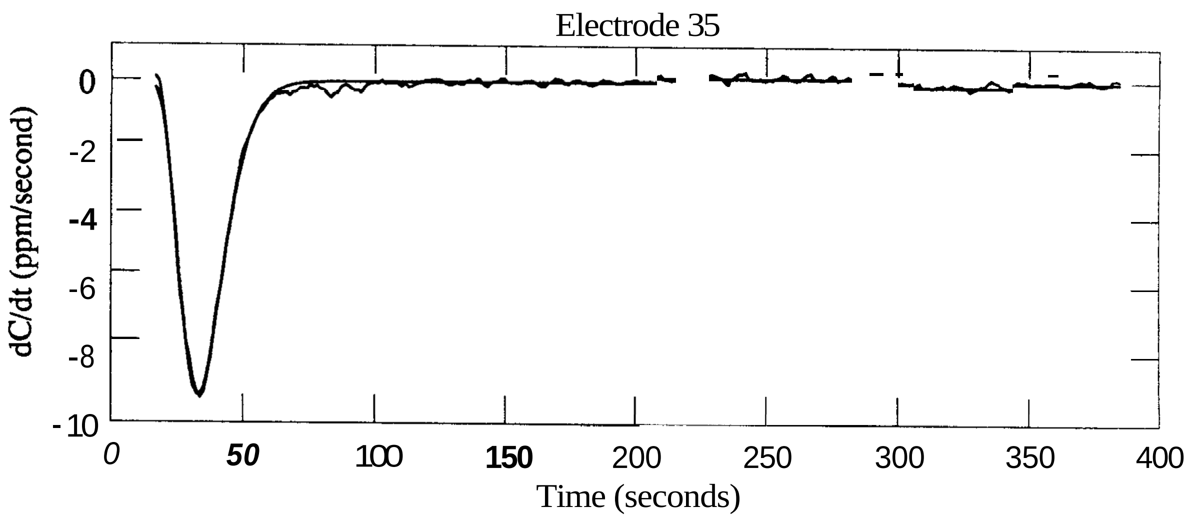

Electrode 65
dC/dt (ppm/second)
0
-2
-4
-6
-8
-10
-12
-14
0
50
100
150
200
250
300
350
400
Time (seconds)

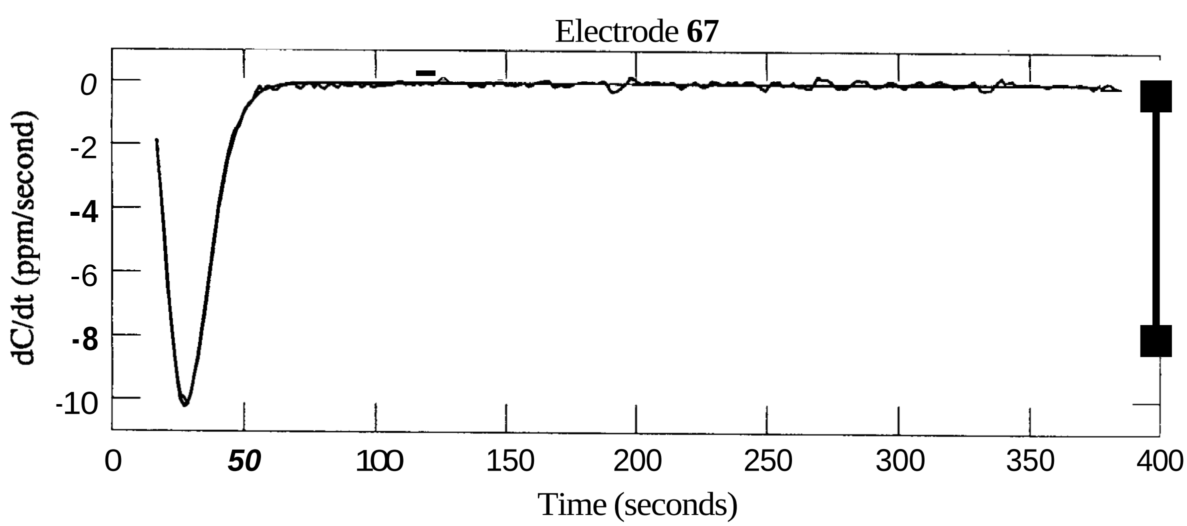

Electrode 2
dC/dt (ppm/second)
0
-2
-4
-6
-8
-10
-12
0
50
100
150
200
250
300
350
400
Time (seconds)

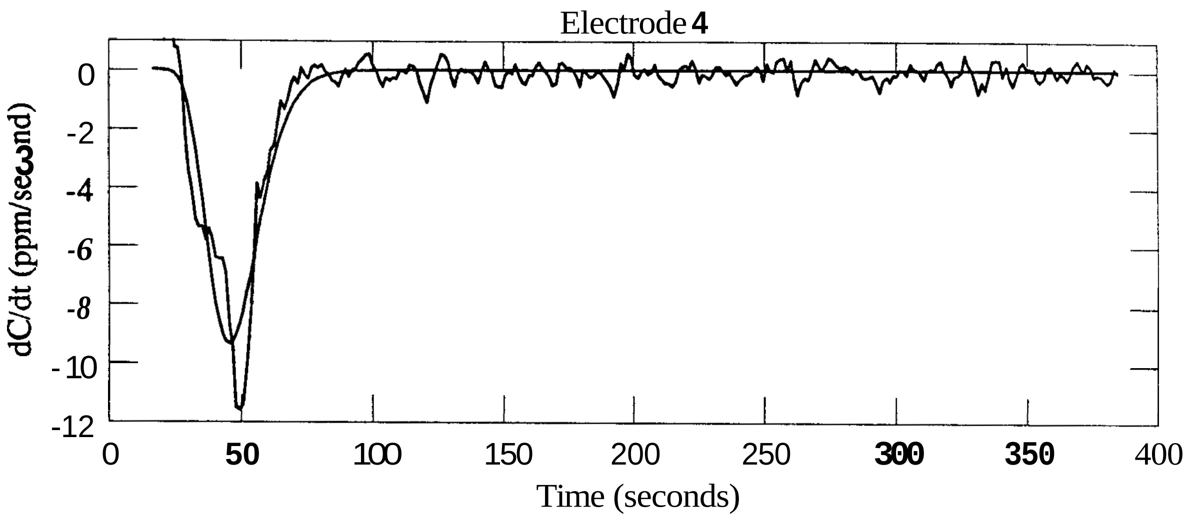

Electrode 34
dC/dt (ppm/second)
0
-2
-4
-6
-8
-10
0
50
100
150
200
250
300
350
400
Time (seconds)

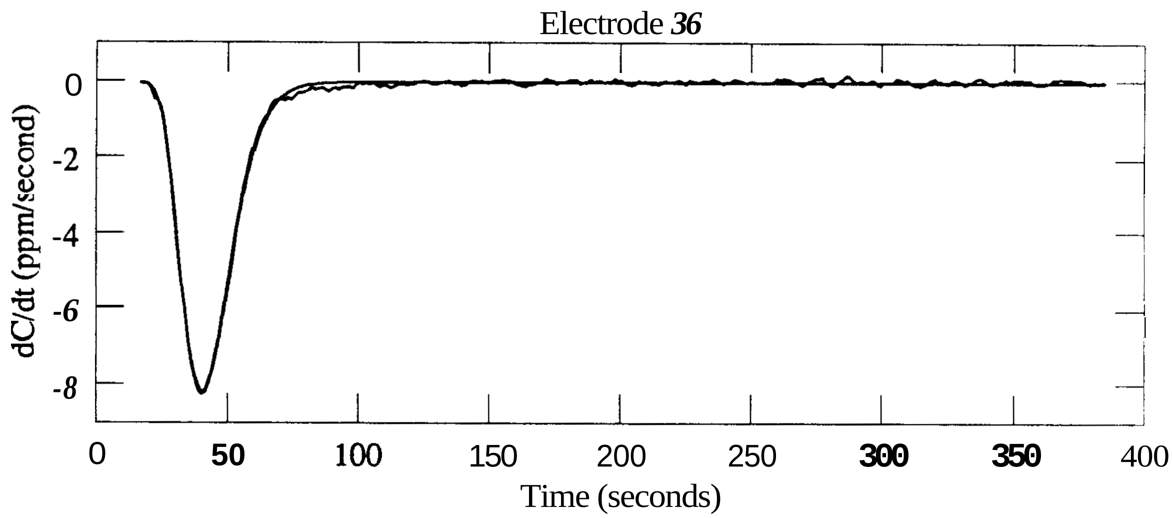

Electrode 66
dC/dt (ppm/second)
0
-2
-4
-6
-8
-10
0
50
100
150
200
250
300
350
400
Time (seconds)

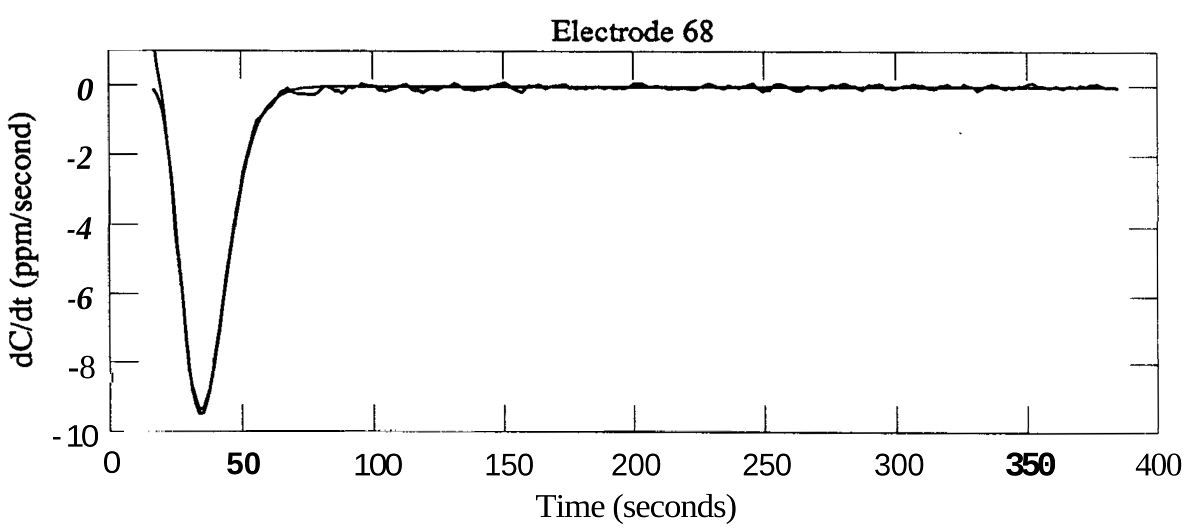

Electrode 3
dC/dt (ppm/second)
0
-2
-4
-6
-8
-10
-12
0
50
100
150
200
250
300
350
400
Time (seconds)

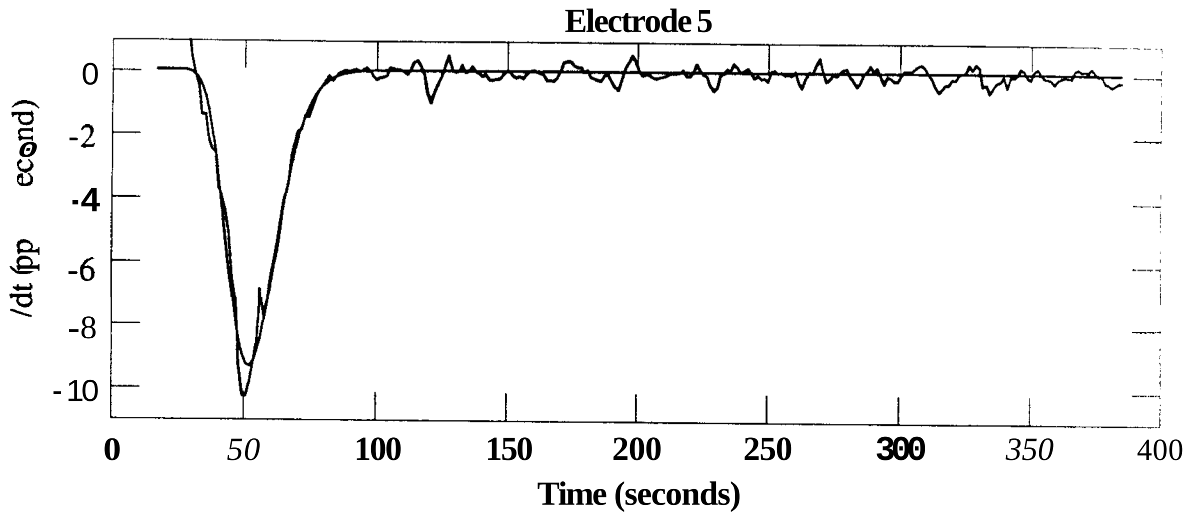

Electrode 35
dC/dt (ppm/second)
0
-2
-4
-6
-8
-10
0
50
100
150
200
250
300
350
400
Time (seconds)

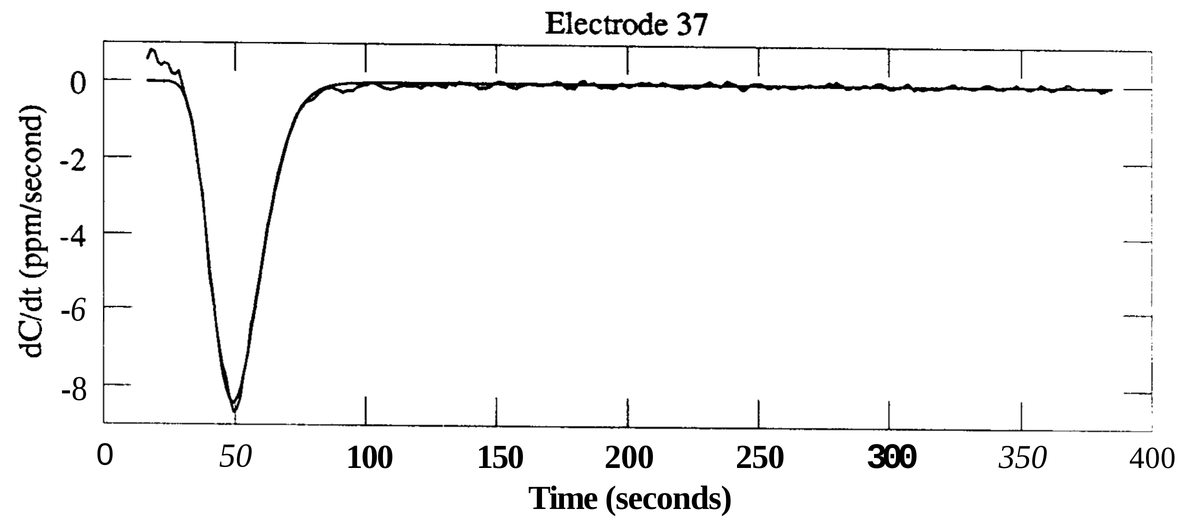

Electrode 67
dC/dt (ppm/second)
0
-2
-4
-6
-8
-10
0
50
100
150
200
250
300
350
400
Time (seconds)

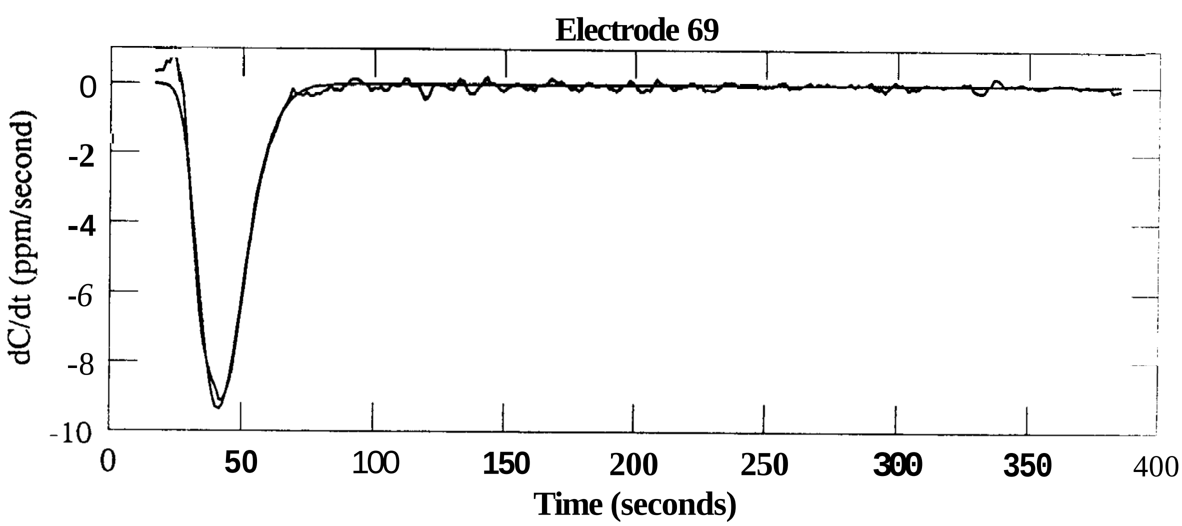

Electrode 4
dC/dt (ppm/second)
0
-2
-4
-6
-8
-10
-12
0
50
100
150
200
250
300
350
400
Time (seconds)

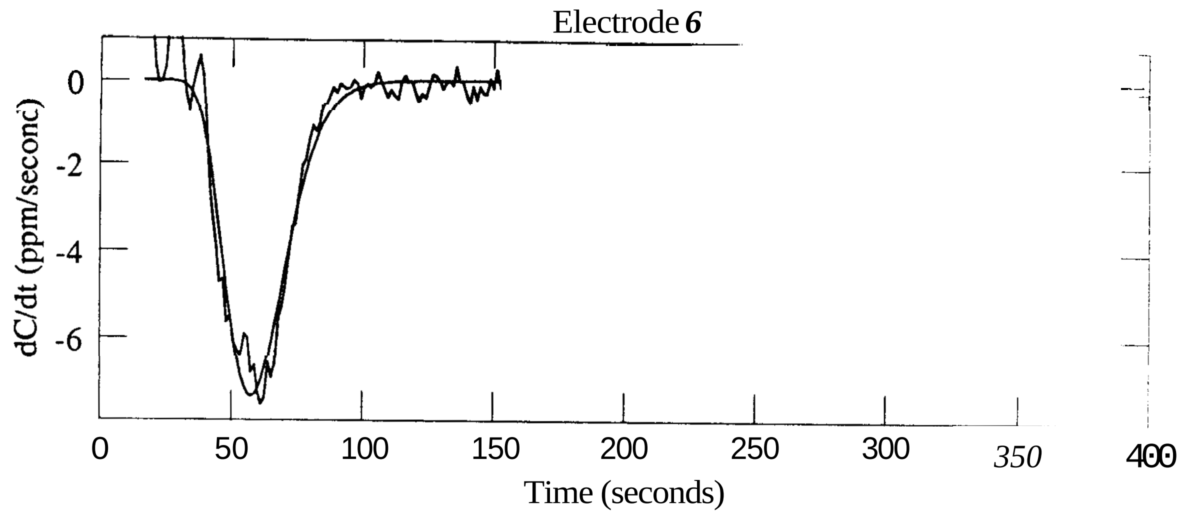

Electrode 36
dC/dt (ppm/second)
0
-2
-4
-6
-8
0
50
100
150
200
250
300
350
400
Time (seconds)

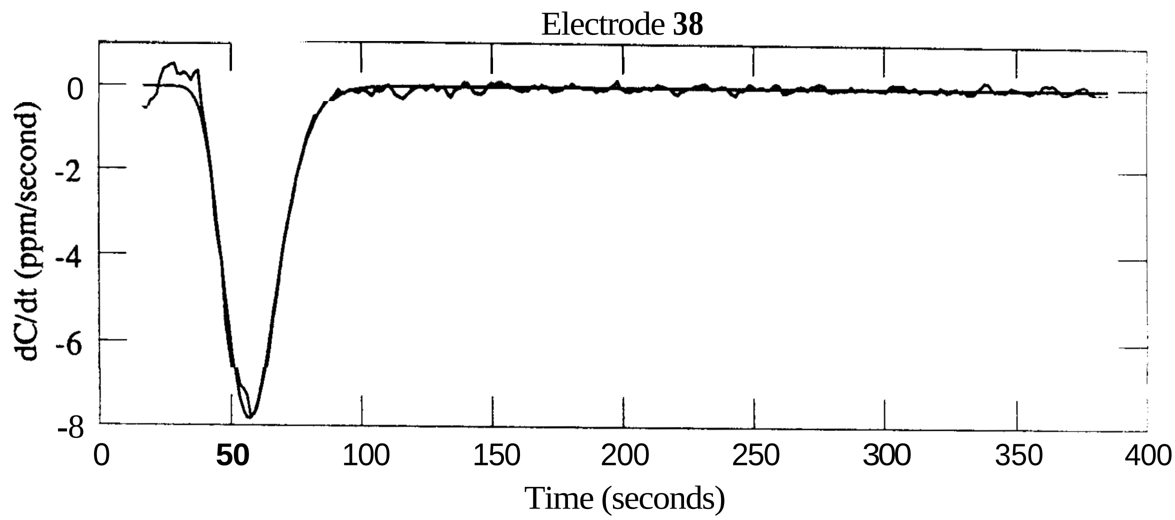

Electrode 68
dC/dt (ppm/second)
0
-2
-4
-6
-8
-10
0
50
100
150
200
250
300
350
400
Time (seconds)

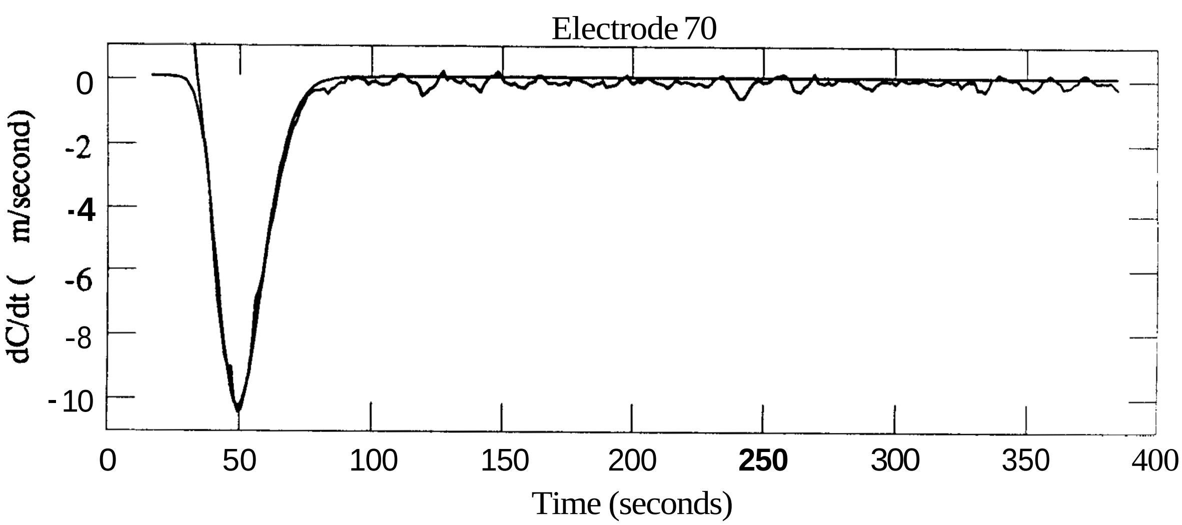

Electrode 5
0
-2
-4
-6
-8
-10
/dt (pp ec⊖nd)
0
50
100
150
200
250
300
350
400
Time (seconds)


Electrode 37
0
-2
-4
-6
-8
dC/dt (ppm/second)
0
50
100
150
200
250
300
350
400
Time (seconds)

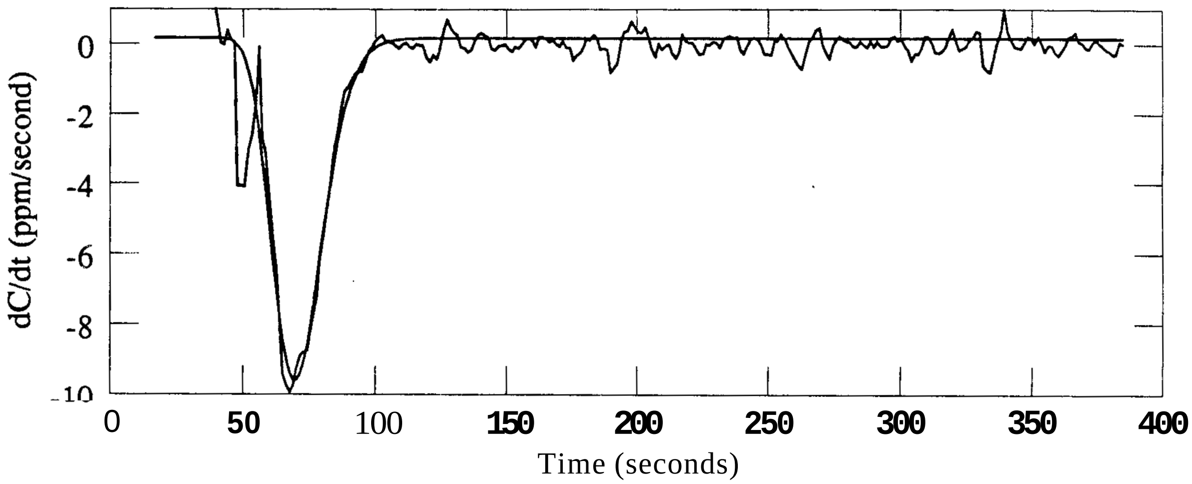

Electrode 69
0
-2
-4
-6
-8
-10
dC/dt (ppm/second)
0
50
100
150
200
250
300
350
400
Time (seconds)


Electrode 6
0
-2
-4
-6
dC/dt (ppm/seconc)
0
50
100
150
200
250
300
350
400
Time (seconds)

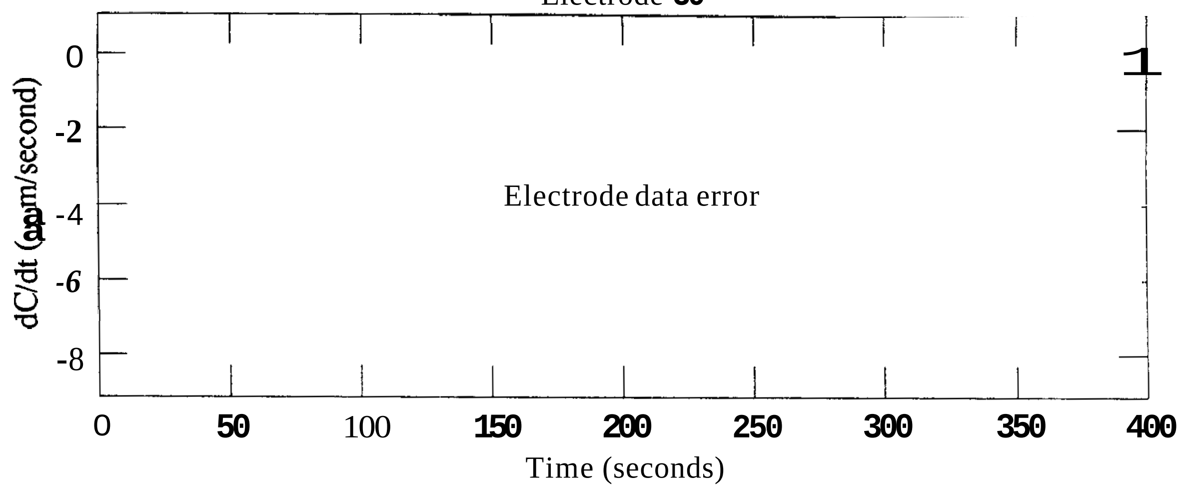

Electrode 38
0
-2
-4
-6
-8
dC/dt (ppm/second)
0
50
100
150
200
250
300
350
400
Time (seconds)


Electrode 70
0
-2
-4
-6
-8
-10
dC/dt (m/second)
0
50
100
150
200
250
300
350
400
Time (seconds)

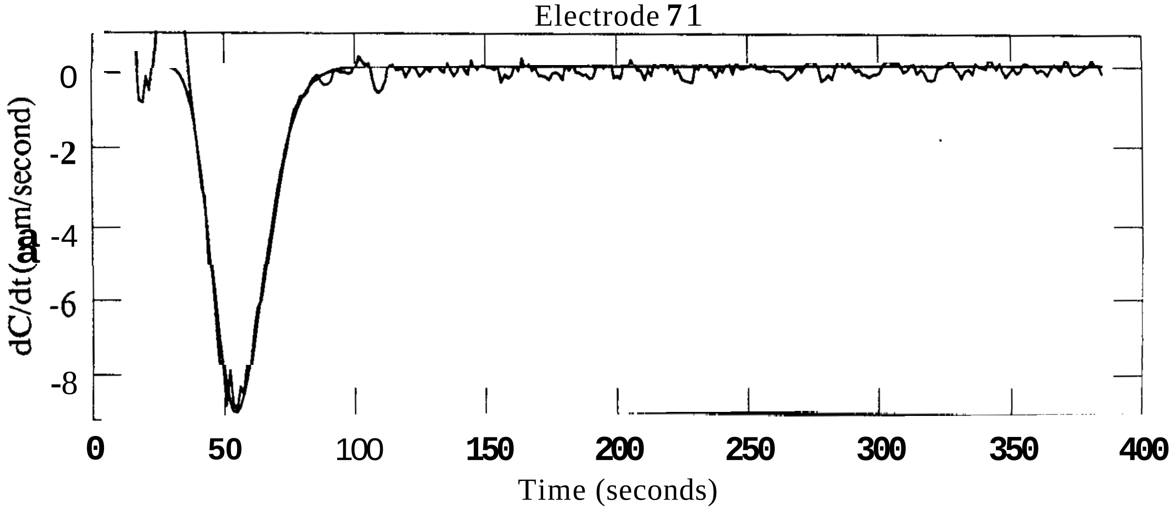
Electrode 7

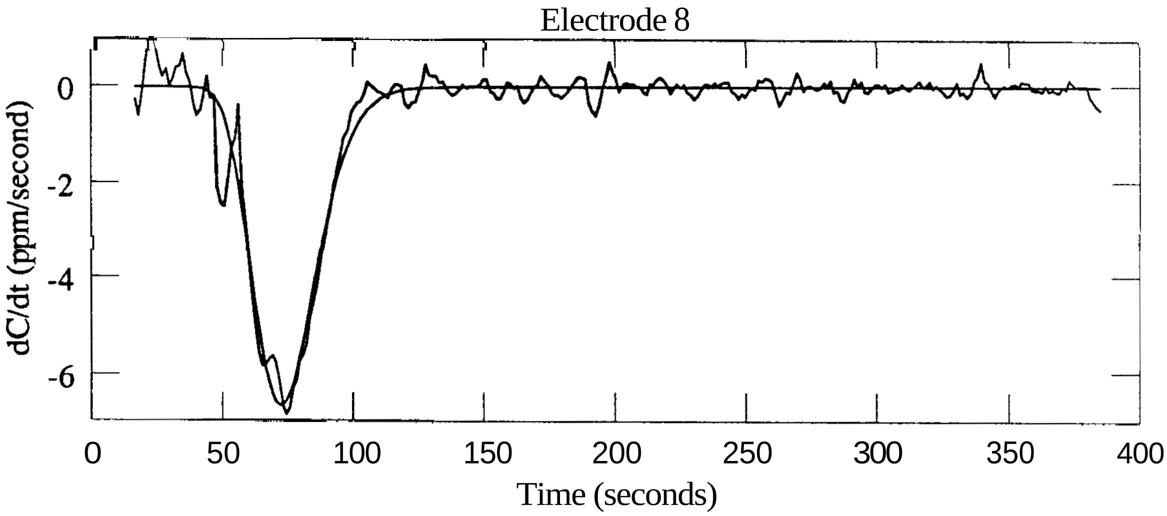


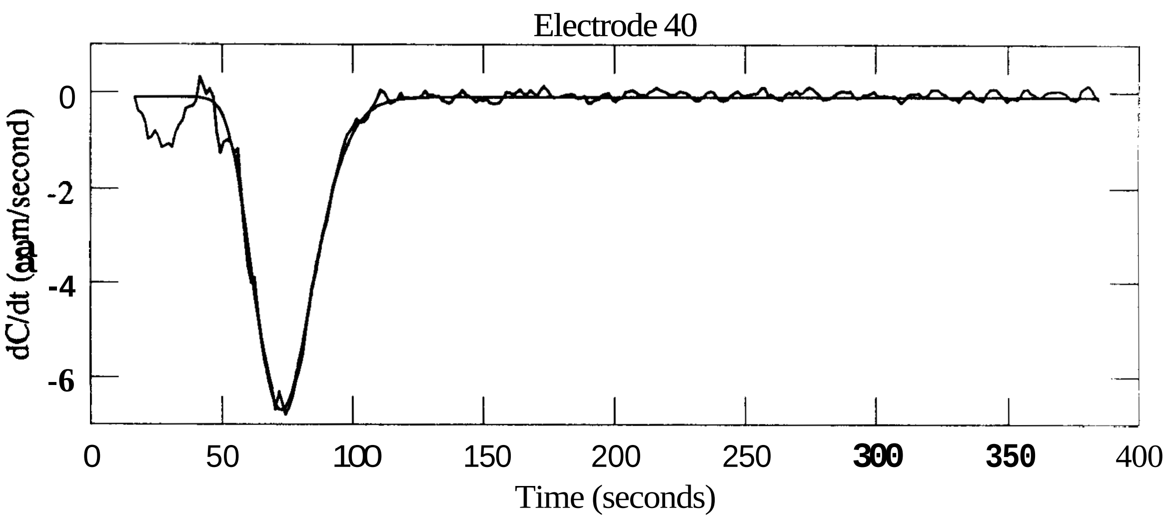
Electrode 39

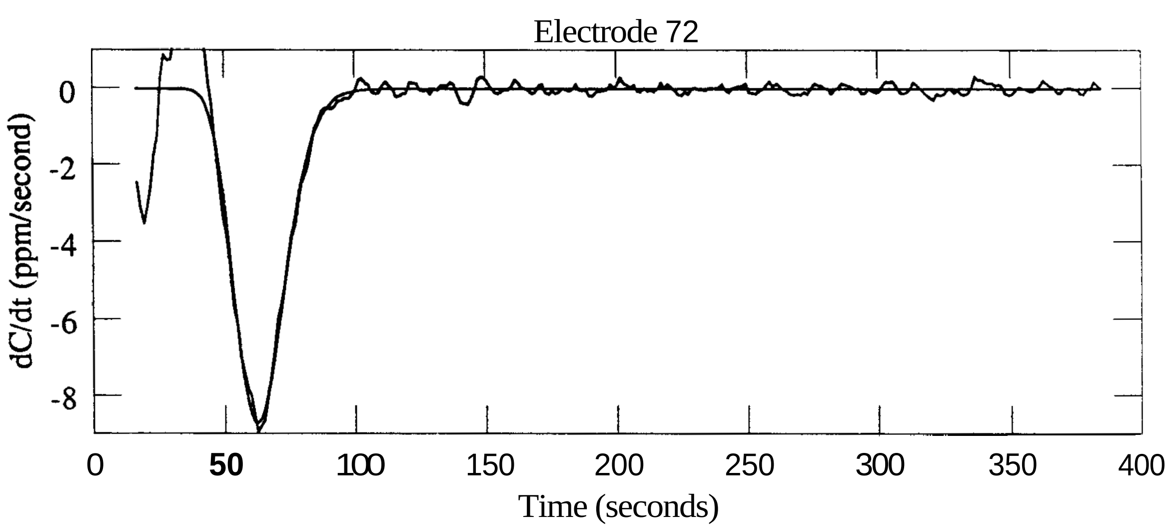


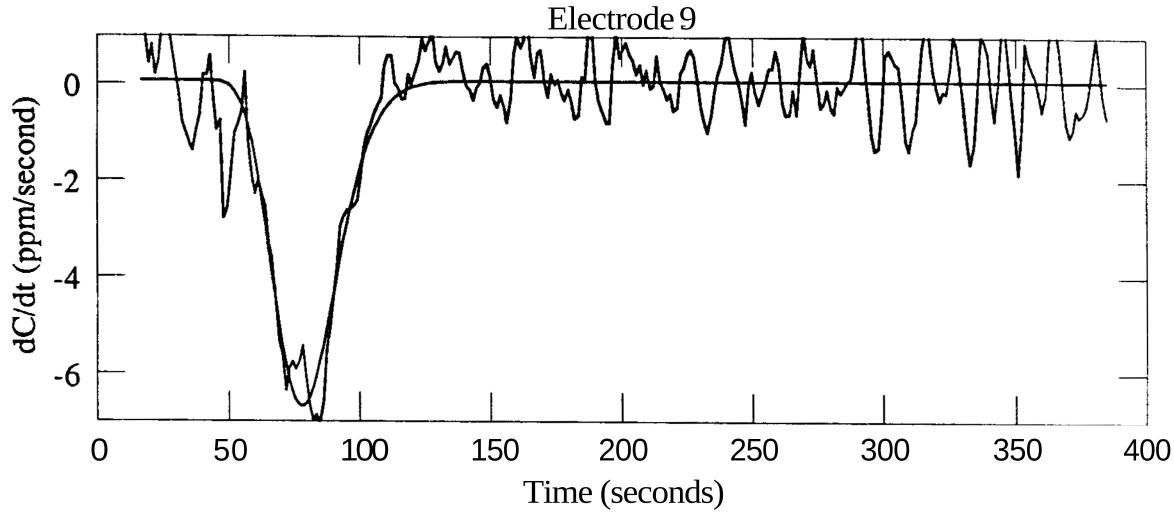
Electrode 71

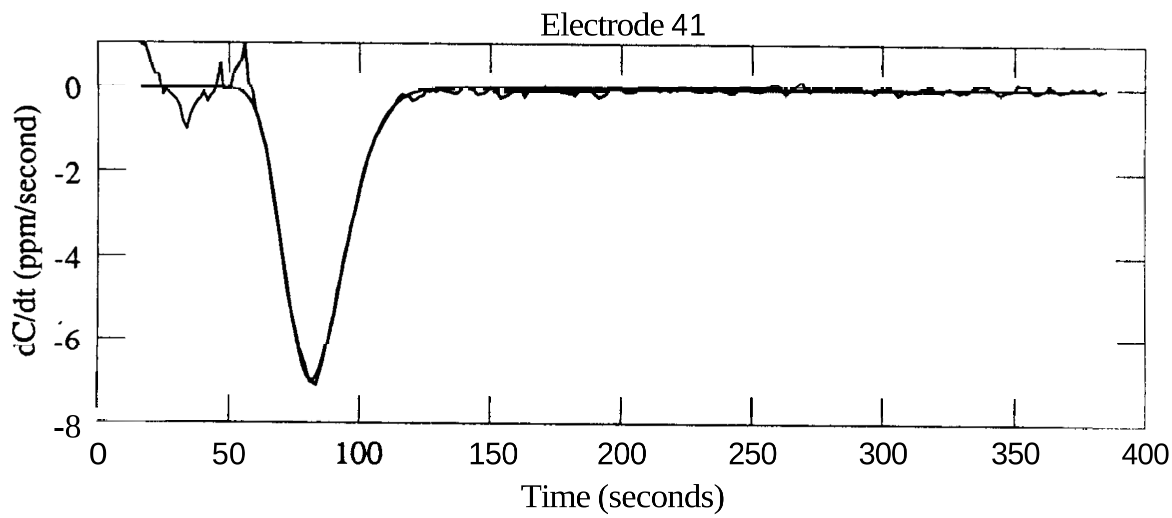


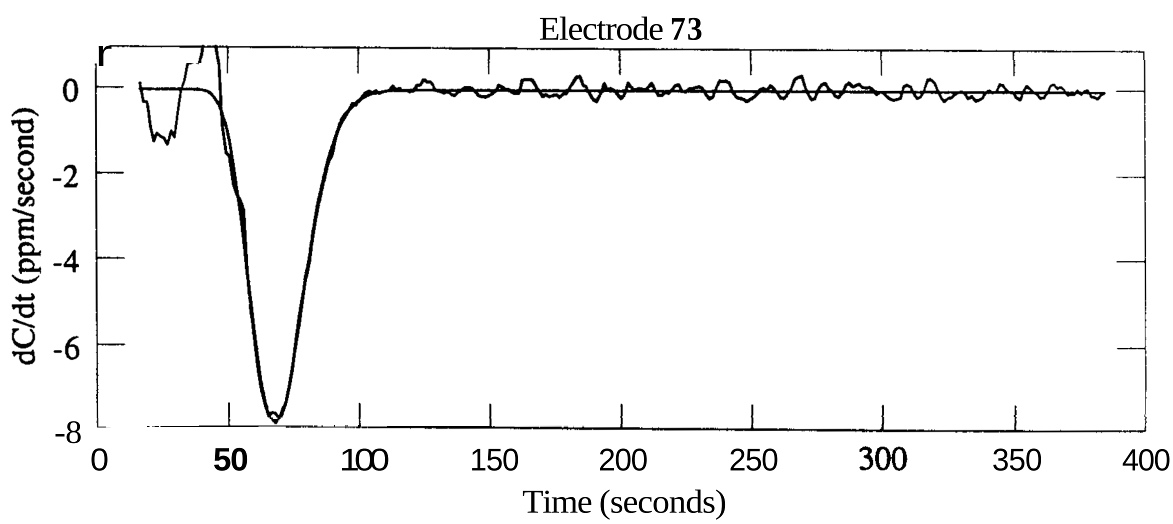

Electrode 8
dC/dt (ppm/second)
0
-2
-4
-6
0
50
100
150
200
250
300
350
400
Time (seconds)

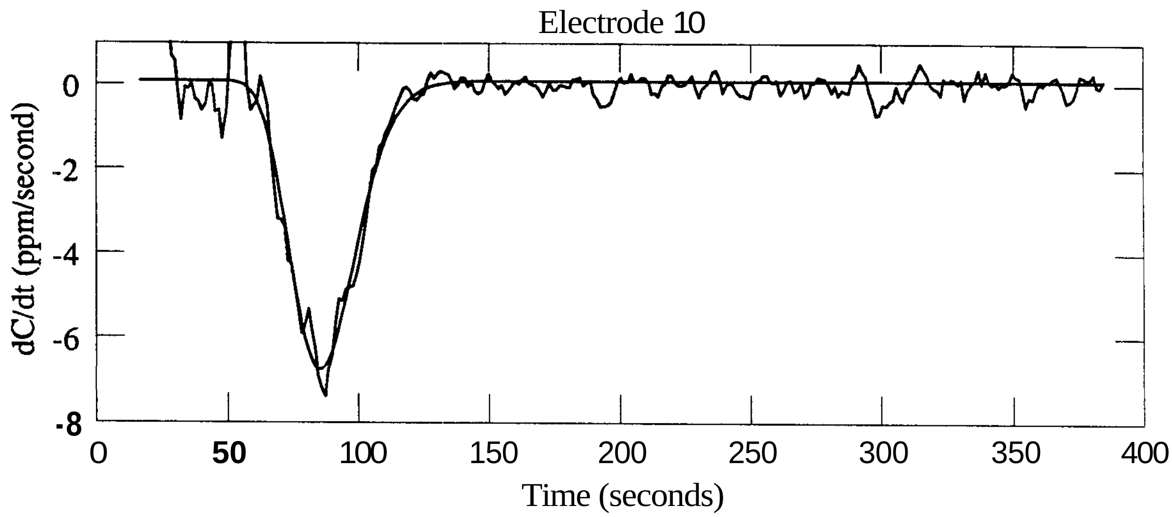

Electrode 40
dC/dt (ppm/second)
0
-2
-4
-6
0
50
100
150
200
250
300
350
400
Time (seconds)

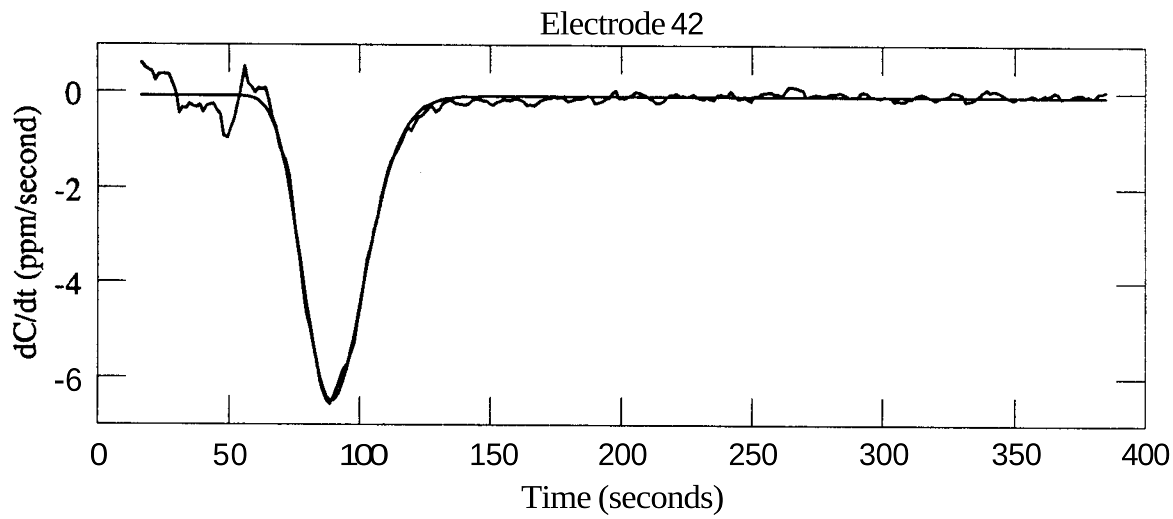

Electrode 72
dC/dt (ppm/second)
0
-2
-4
-6
-8
0
50
100
150
200
250
300
350
400
Time (seconds)

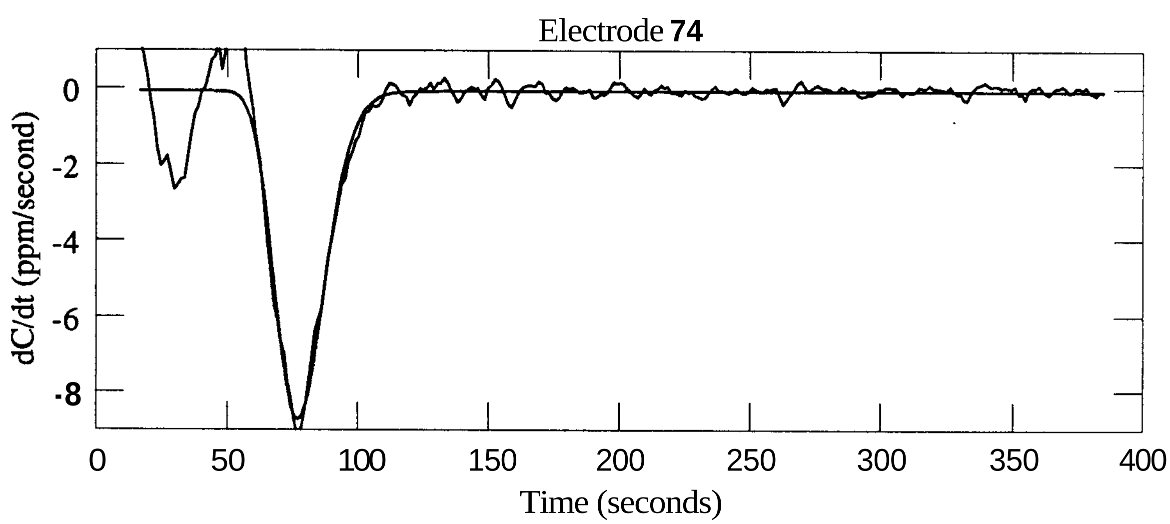

Electrode 9
dC/dt (ppm/second)
0
-2
-4
-6
0
50
100
150
200
250
300
350
400
Time (seconds)

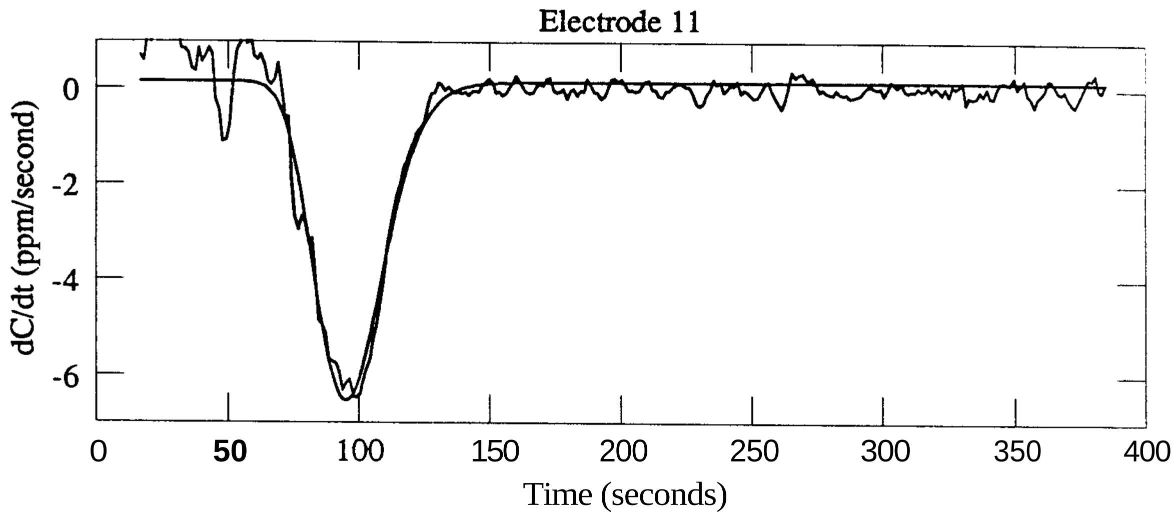

Electrode 41
dC/dt (ppm/second)
0
-2
-4
-6
-8
0
50
100
150
200
250
300
350
400
Time (seconds)

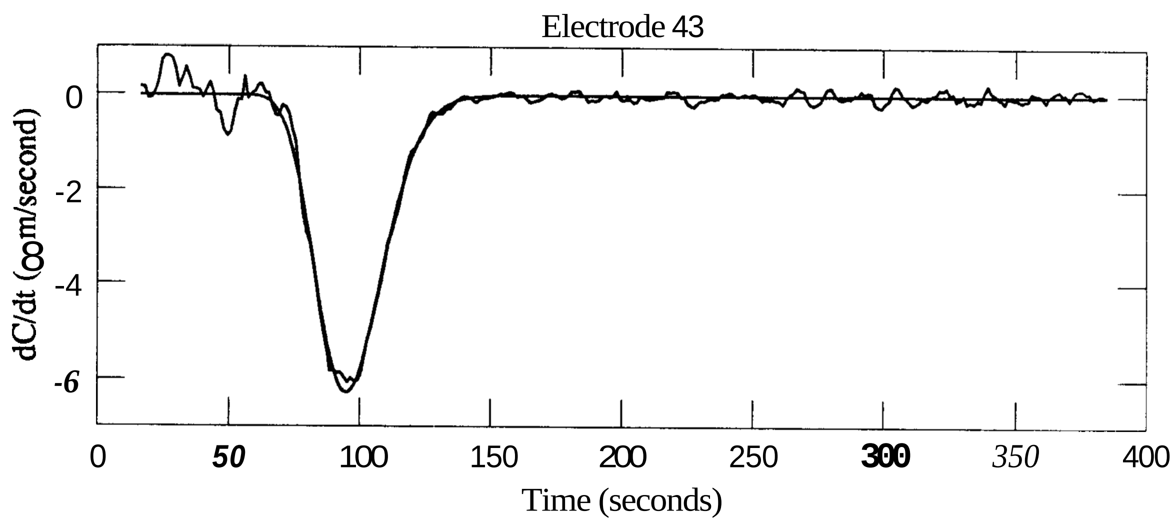

Electrode 73
dC/dt (ppm/second)
0
-2
-4
-6
-8
0
50
100
150
200
250
300
350
400
Time (seconds)

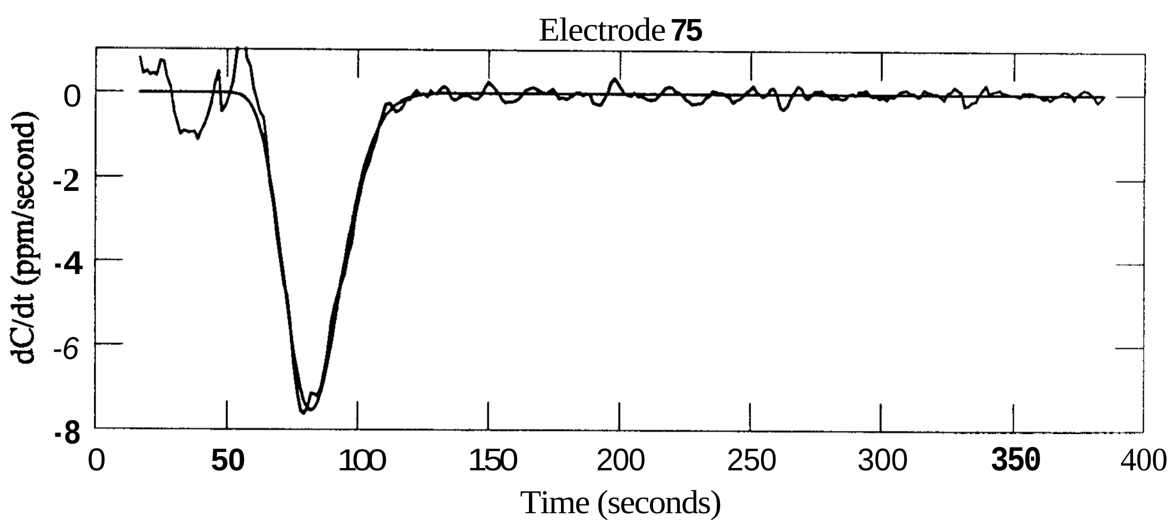

Electrode 10
0
-2
-4
-6
-8
dC/dt (ppm/second)
0
50
100
150
200
250
300
350
400
Time (seconds)

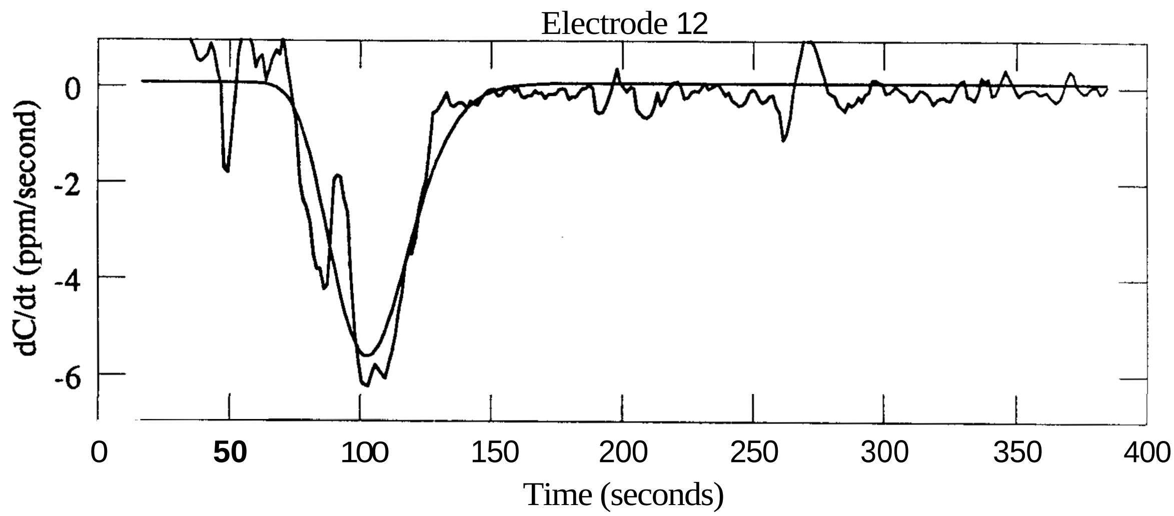

Electrode 42
0
-2
-4
-6
dC/dt (ppm/second)
0
50
100
150
200
250
300
350
400
Time (seconds)

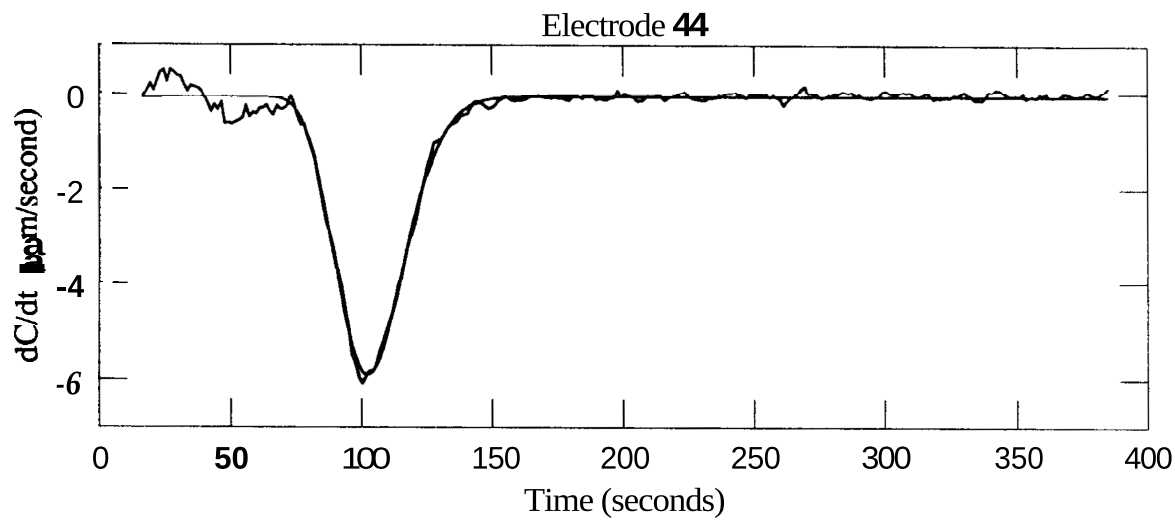

Electrode 74
0
-2
-4
-6
-8
dC/dt (ppm/second)
0
50
100
150
200
250
300
350
400
Time (seconds)

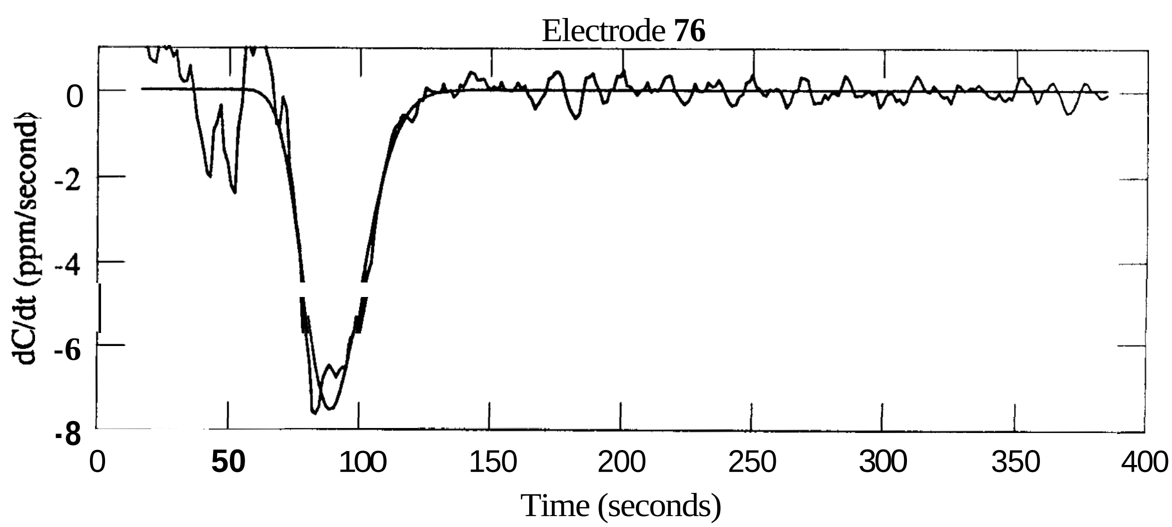

Electrode 11
dC/dt (ppm/second)
0
-2
-4
-6
0 50 100 150 200 250 300 350 400
Time (seconds)

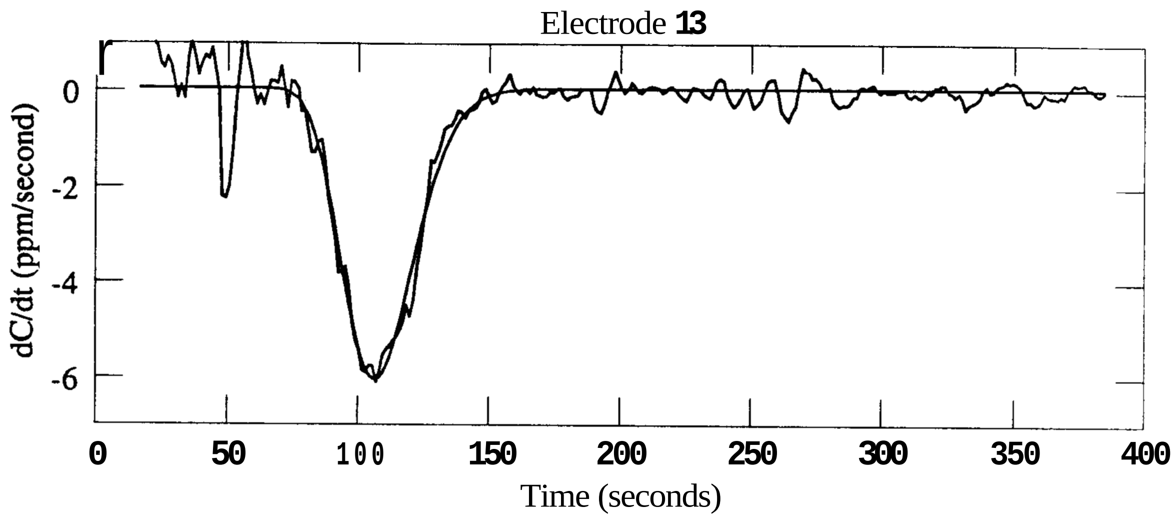

Electrode 43
0
-2
-4
-6
0 50 100 150 200 250 300 350 400
Time (seconds)

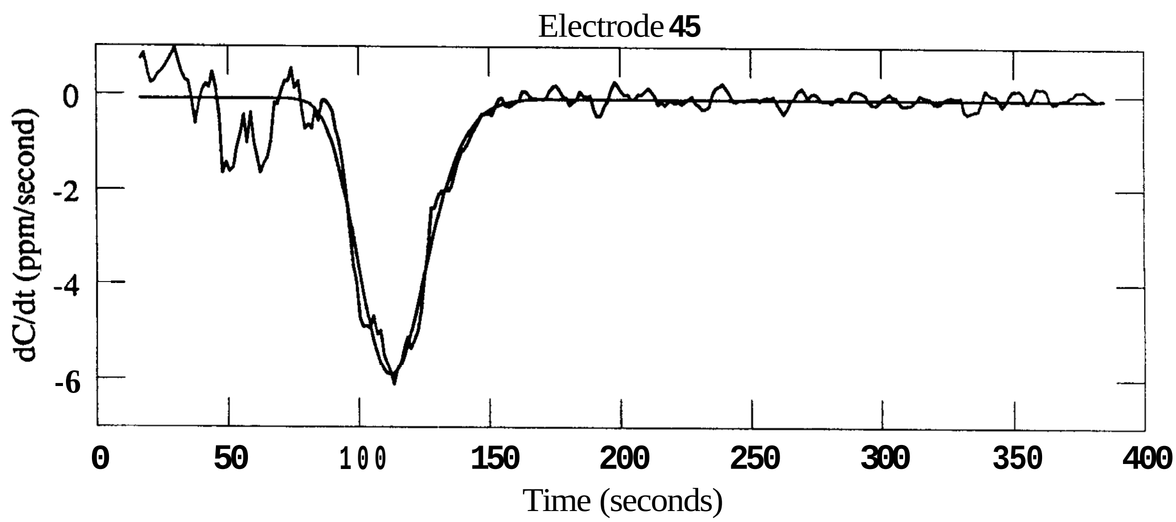

Electrode 75
dC/dt (ppm/second)
0
-2
-4
-6
-8
0 50 100 150 200 250 300 350 400
Time (seconds)

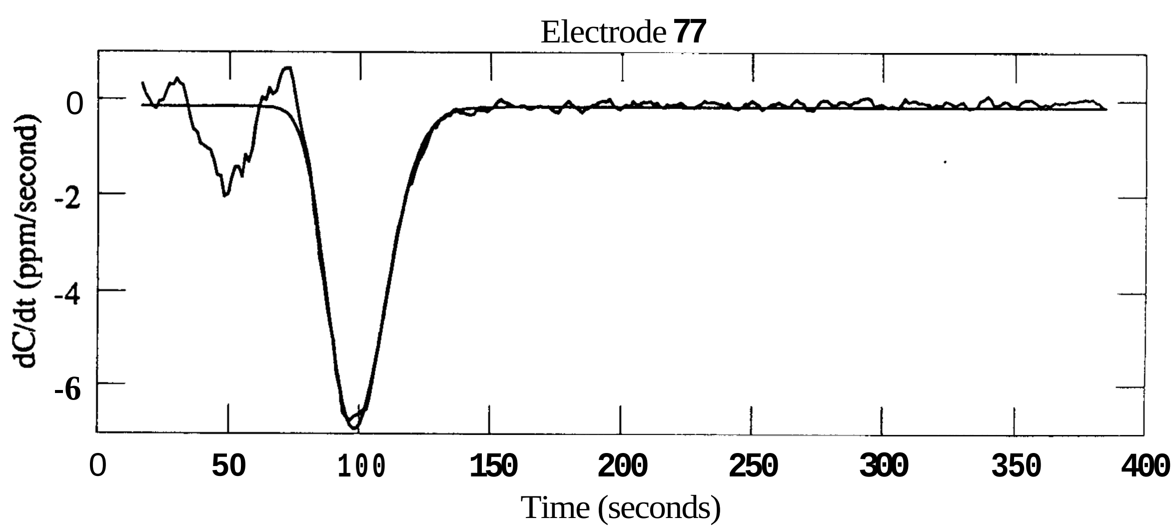

Electrode 12
dC/dt (ppm/second)
0
-2
-4
-6
0
50
100
150
200
250
300
350
400
Time (seconds)

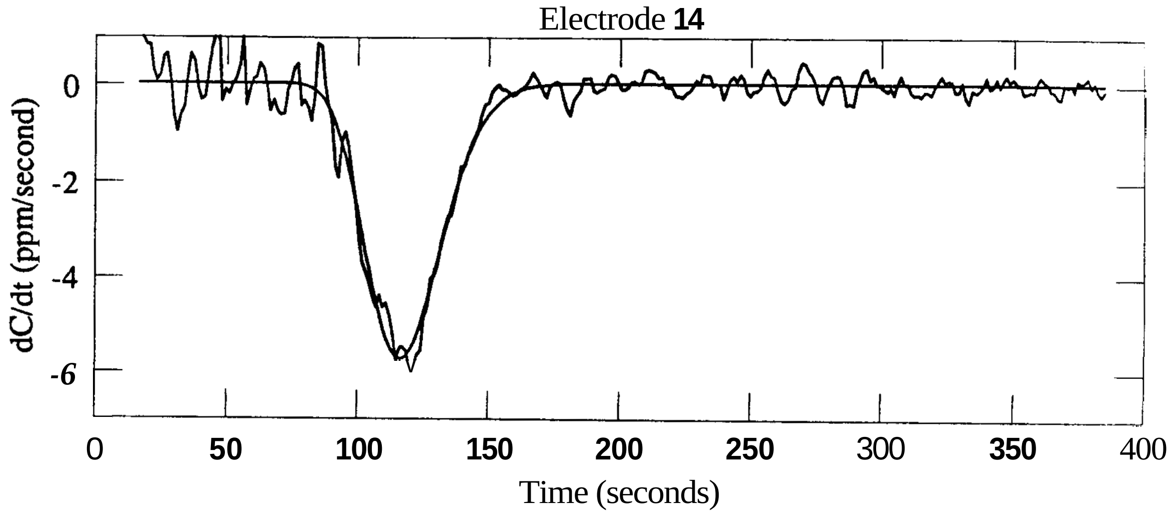

Electrode 44
dC/dt (ppm/second)
0
-2
-4
-6
0
50
100
150
200
250
300
350
400
Time (seconds)

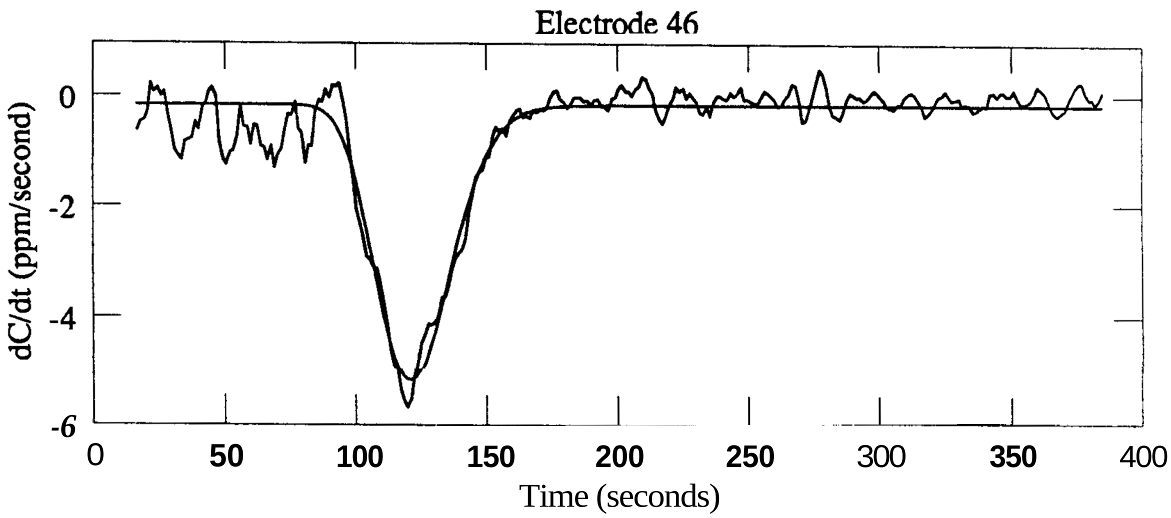

Electrode 76
dC/dt (ppm/second)
0
-2
-4
-6
-8
0
50
100
150
200
250
300
350
400
Time (seconds)

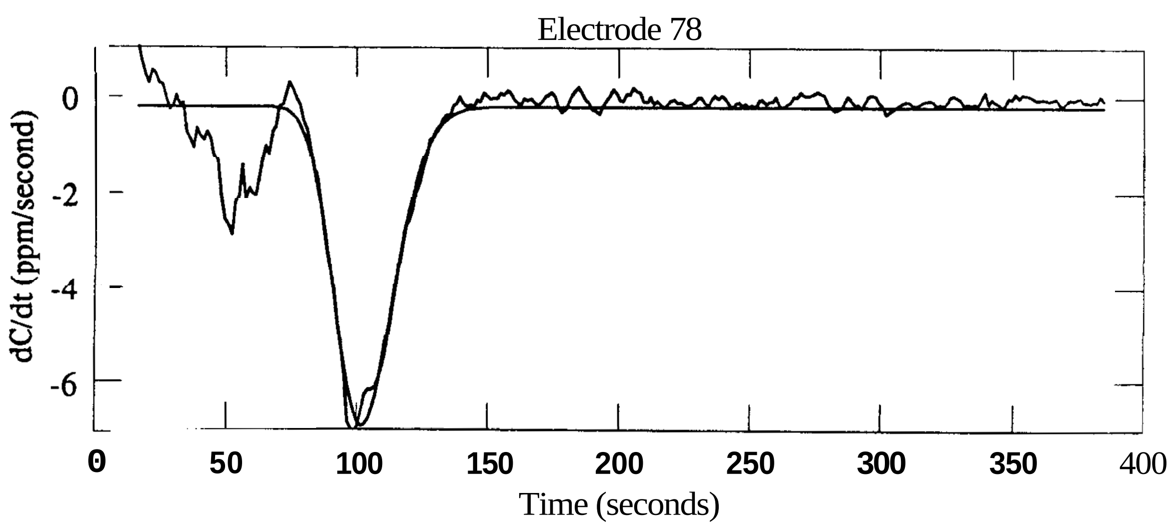

Electrode 13
dC/dt (ppm/second)
0
-2
-4
-6
0
50
100
150
200
250
300
350
400
Time (seconds)

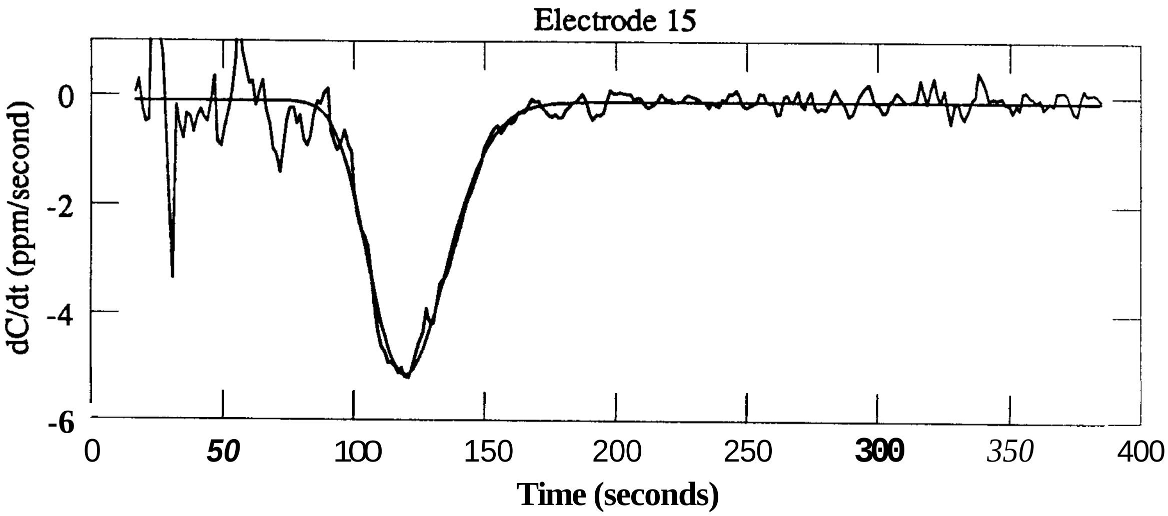

Electrode 45
dC/dt (ppm/second)
0
-2
-4
-6
0
50
100
150
200
250
300
350
400
Time (seconds)

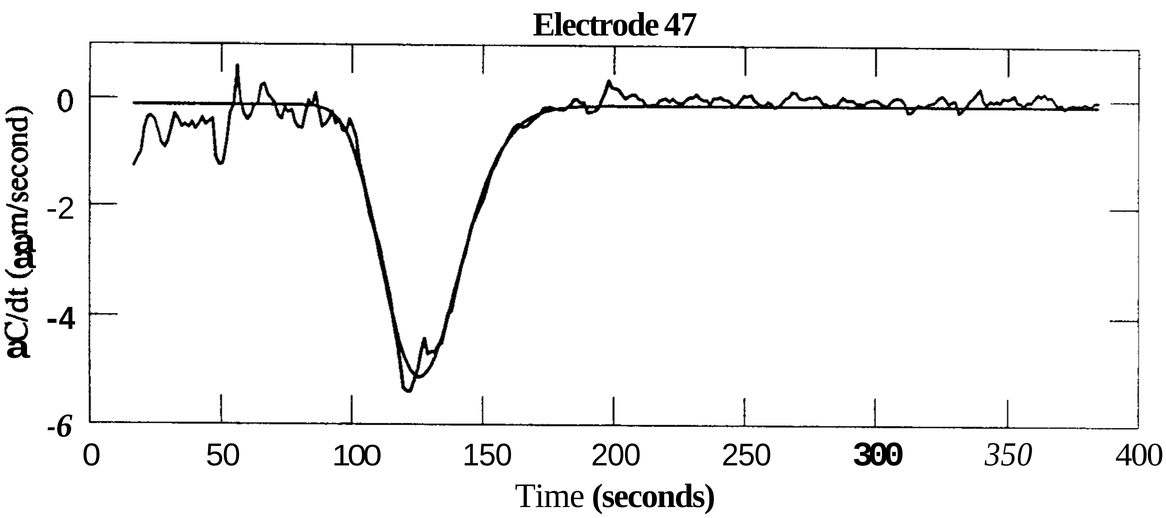

Electrode 77
dC/dt (ppm/second)
0
-2
-4
-6
0
50
100
150
200
250
300
350
400
Time (seconds)

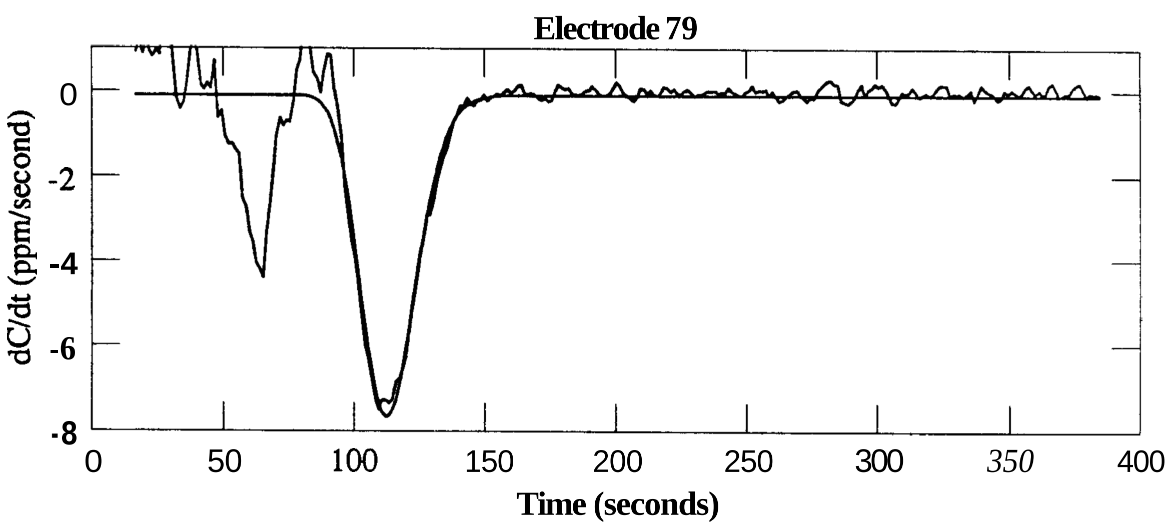

Electrode 14
dC/dt (ppm/second)
0
-2
-4
-6
0
50
100
150
200
250
300
350
400
Time (seconds)

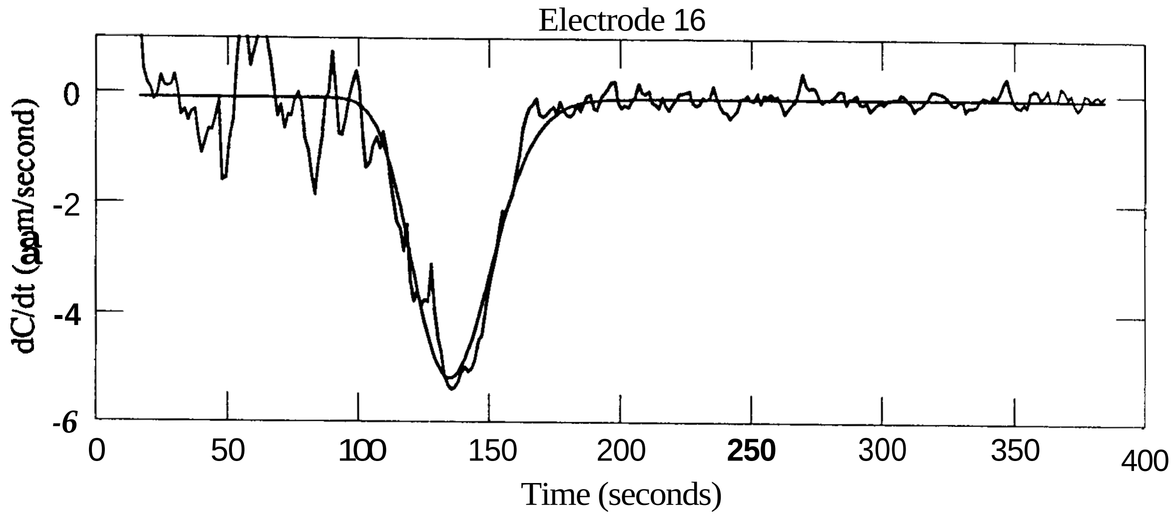

Electrode 46
dC/dt (ppm/second)
0
-2
-4
-6
0
50
100
150
200
250
300
350
400
Time (seconds)

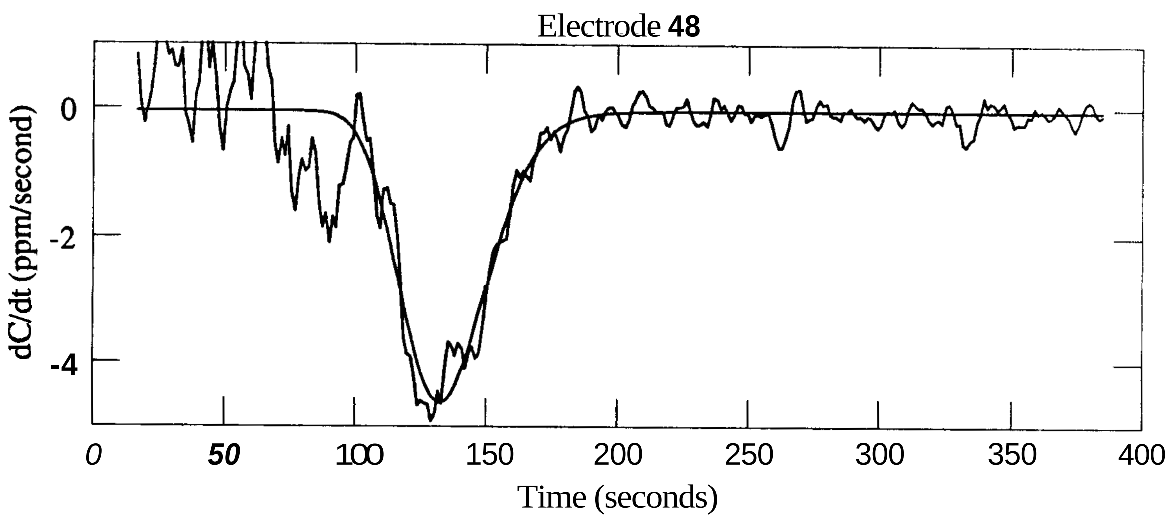

Electrode 78
dC/dt (ppm/second)
0
-2
-4
-6
0
50
100
150
200
250
300
350
400
Time (seconds)

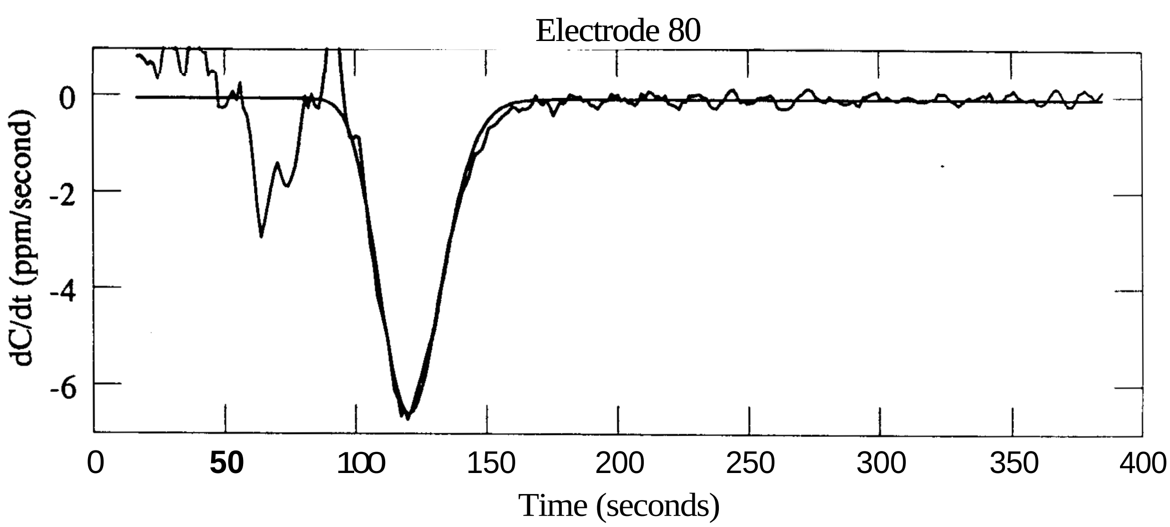

Electrode 15
dC/dt (ppm/second)
0
-2
-4
-6
0
50
100
150
200
250
300
350
400
Time (seconds)

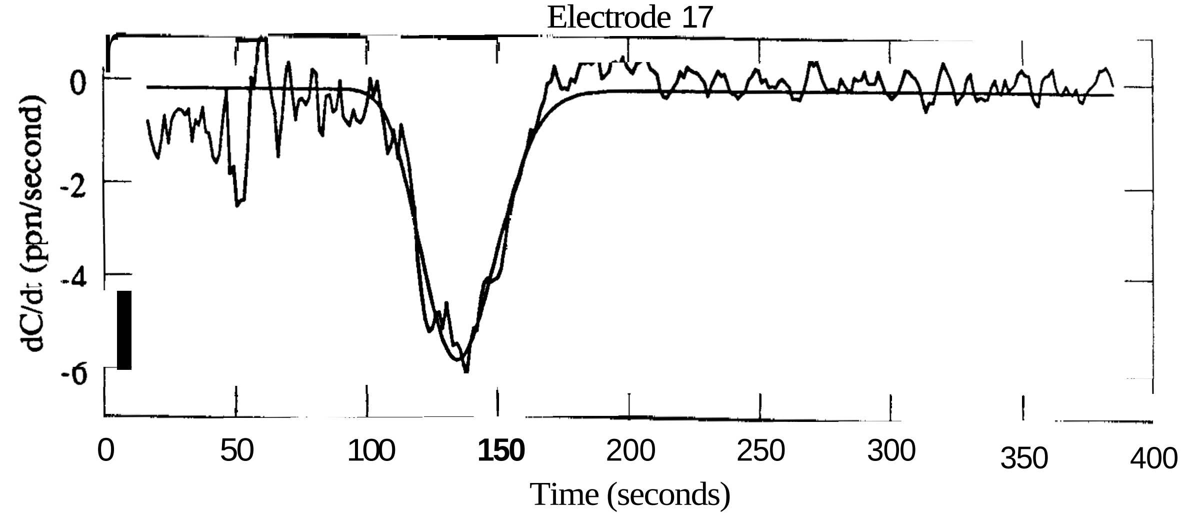

Electrode 47
dC/dt (ppm/second)
0
-2
-4
-6
0
50
100
150
200
250
300
350
400
Time (seconds)

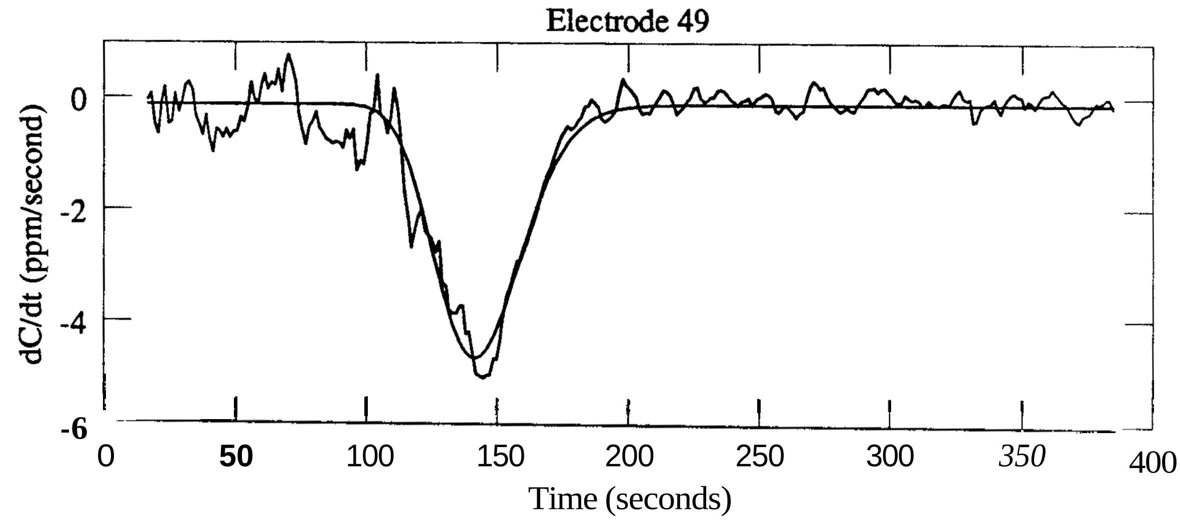

Electrode 79
dC/dt (ppm/second)
0
-2
-4
-6
-8
0
50
100
150
200
250
300
350
400
Time (seconds)

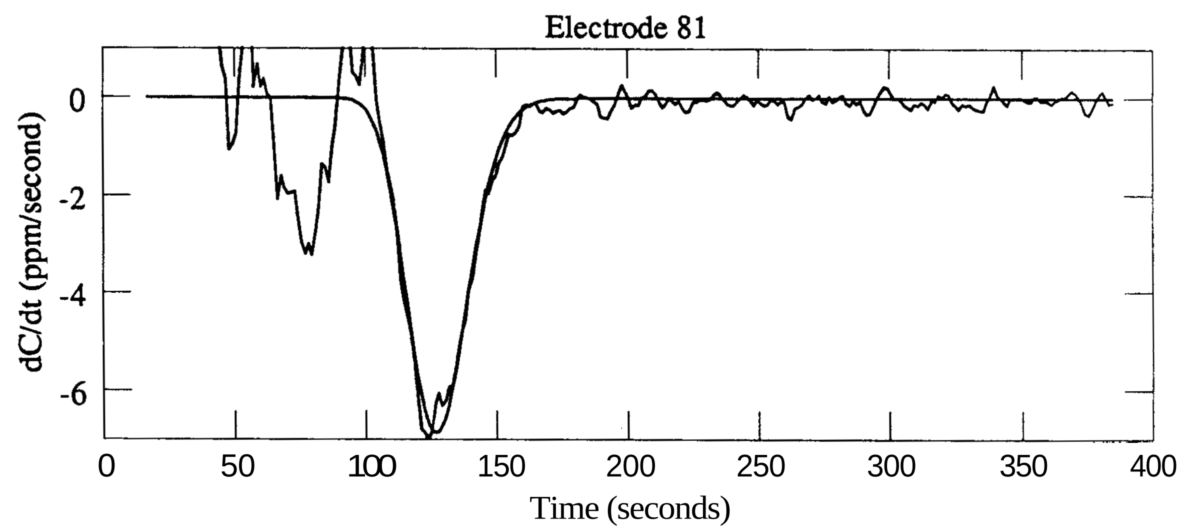

Electrode 16
dC/dt (ppm/second)
0
-2
-4
-6
0
50
100
150
200
250
300
350
400
Time (seconds)

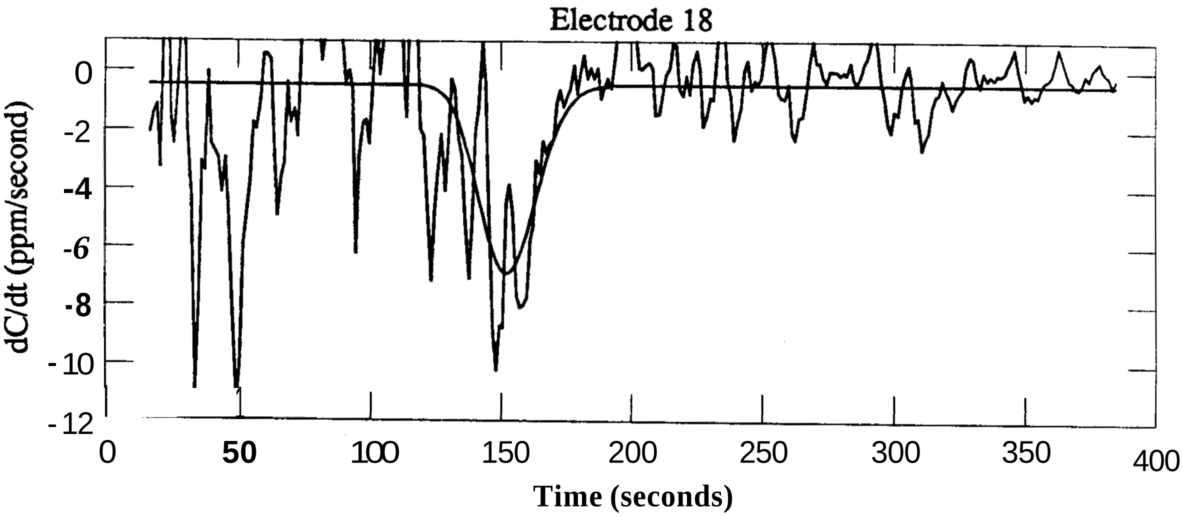

Electrode 48
dC/dt (ppm/second)
0
-2
-4
0
50
100
150
200
250
300
350
400
Time (seconds)

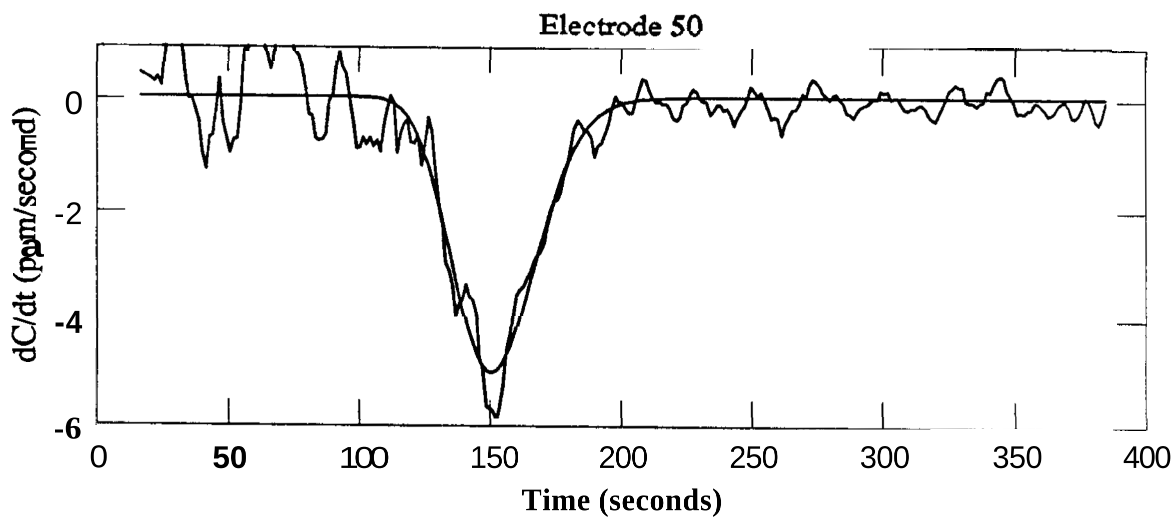

Electrode 80
dC/dt (ppm/second)
0
-2
-4
-6
0
50
100
150
200
250
300
350
400
Time (seconds)

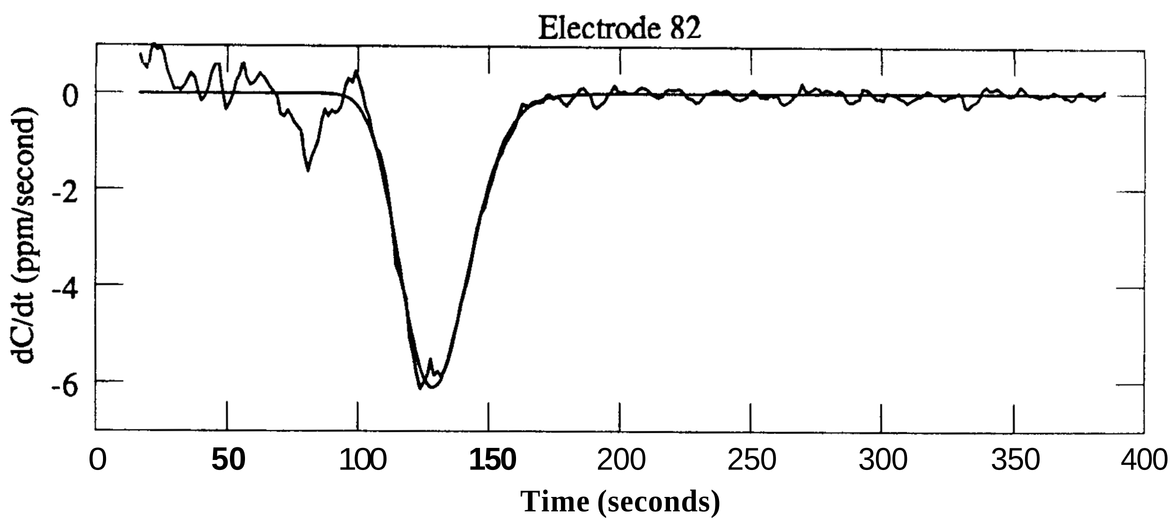

Electrode 17
dC/d: (ppn/second)
0
-2
-4
-6
0
50
100
150
200
250
300
350
400
Time (seconds)

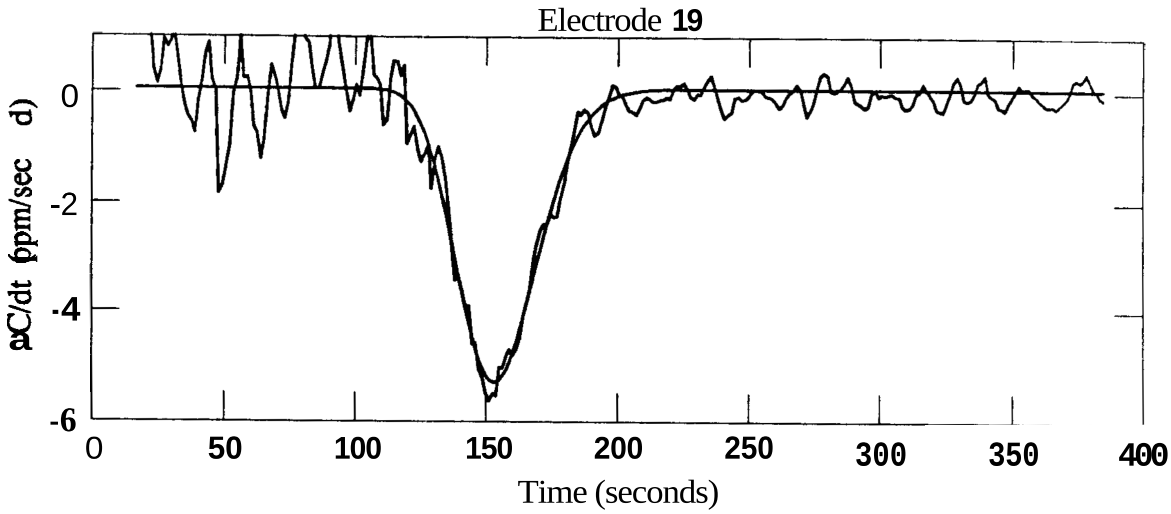

Electrode 49
dC/dt (ppm/second)
0
-2
-4
-6
0
50
100
150
200
250
300
350
400
Time (seconds)

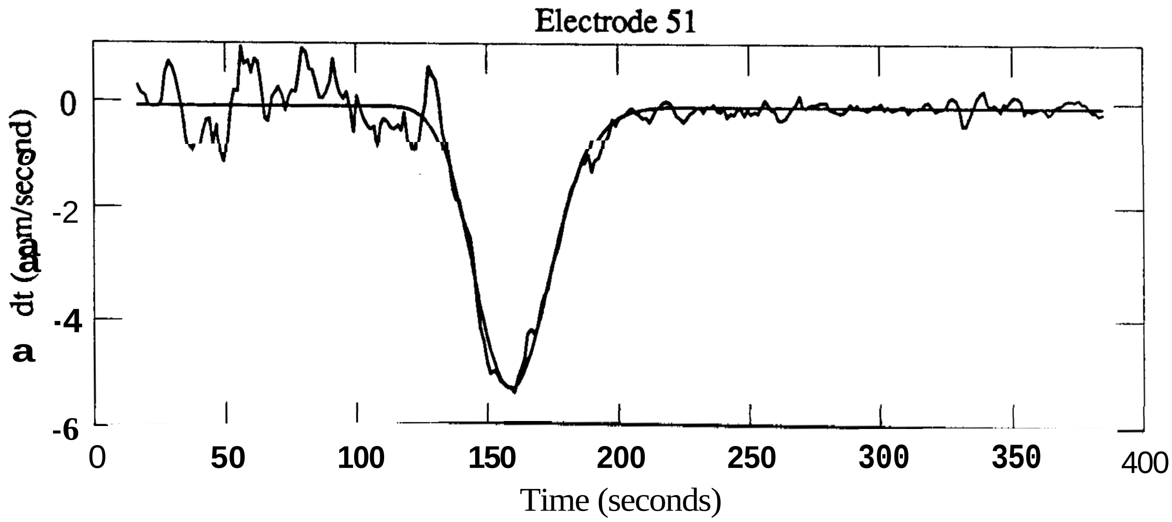

Electrode 81
dC/dt (ppm/second)
0
-2
-4
-6
0
50
100
150
200
250
300
350
400
Time (seconds)

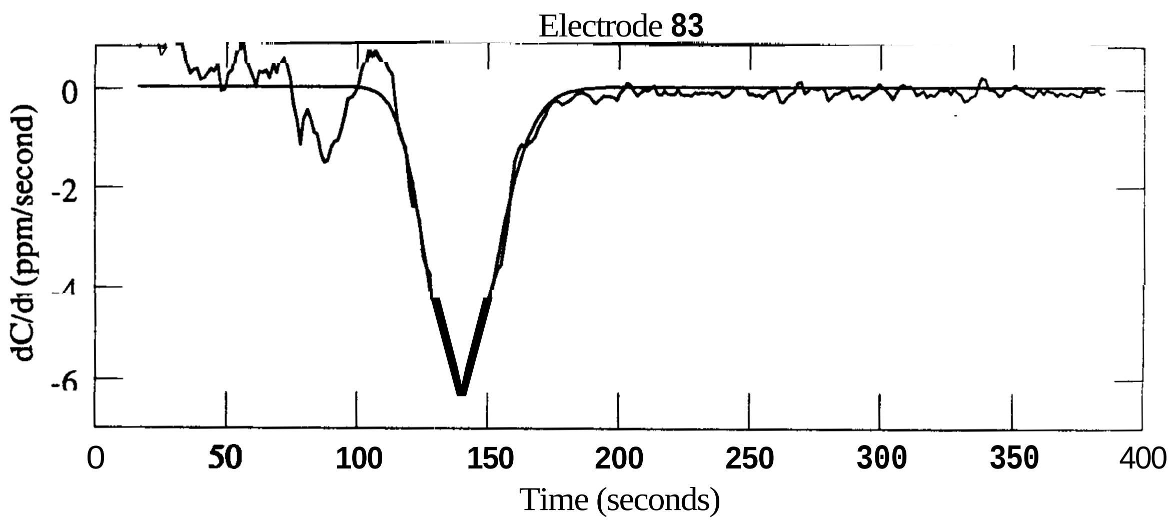

Electrode 18
dC/dt (ppm/second)
0
-2
-4
-6
-8
-10
-12
0
50
100
150
200
250
300
350
400
Time (seconds)

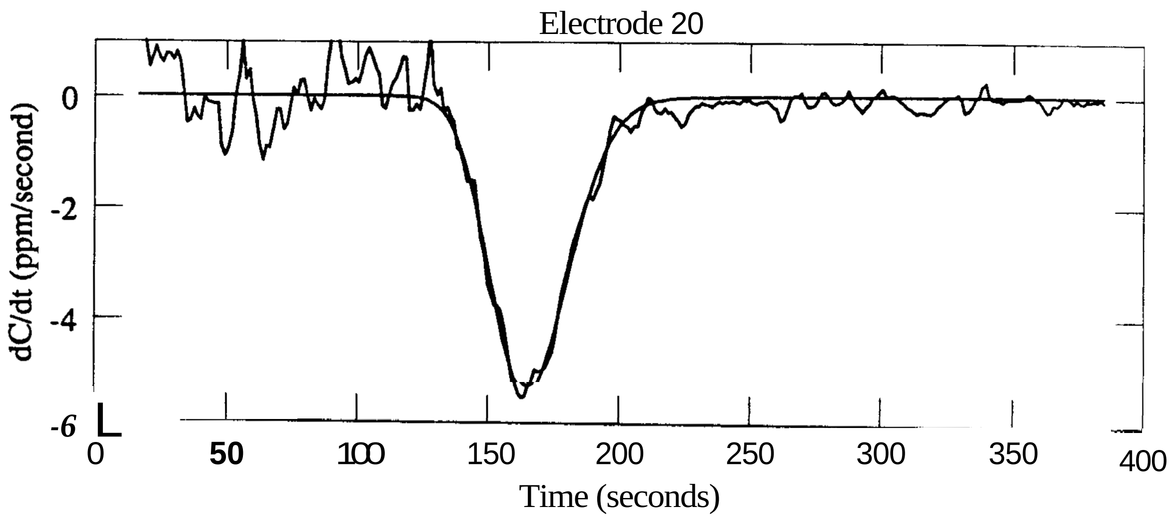

Electrode 50
dC/dt (ppm/second)
0
-2
-4
-6
0
50
100
150
200
250
300
350
400
Time (seconds)

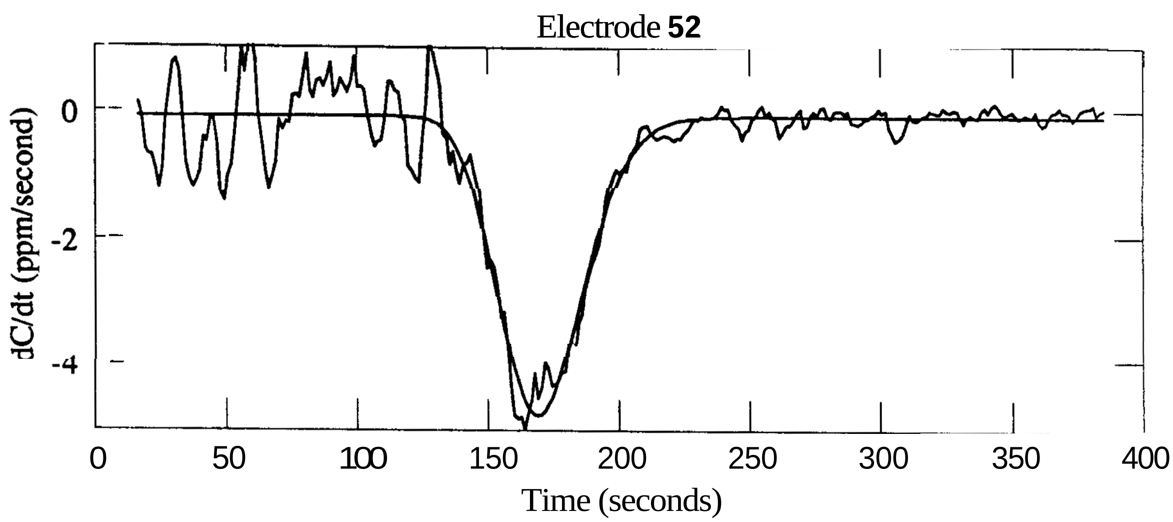

Electrode 82
dC/dt (ppm/second)
0
-2
-4
-6
0
50
100
150
200
250
300
350
400
Time (seconds)

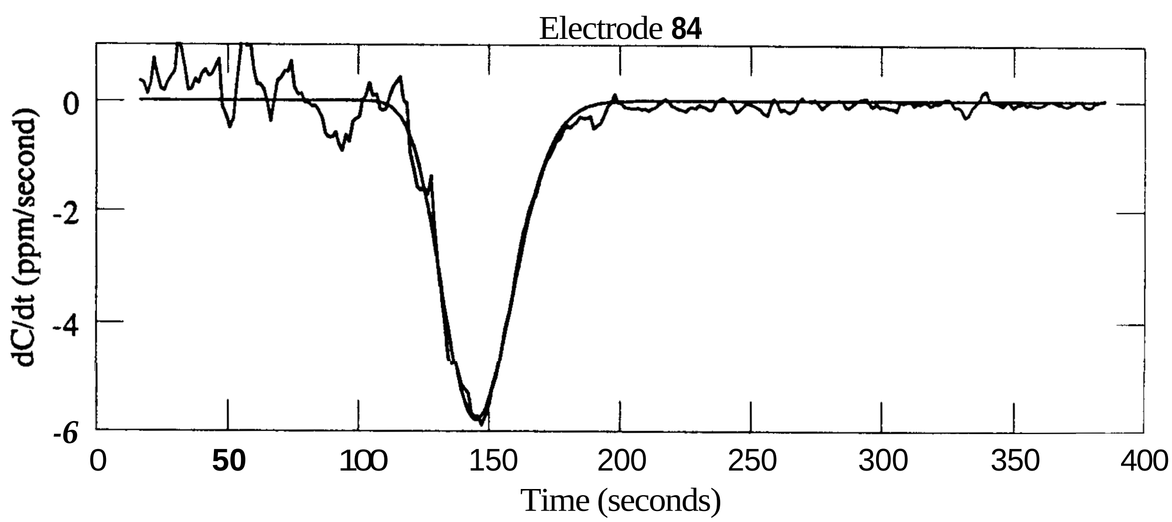

Electrode 19
0
-2
-4
-6
0
50
100
150
200
250
300
350
400
Time (seconds)

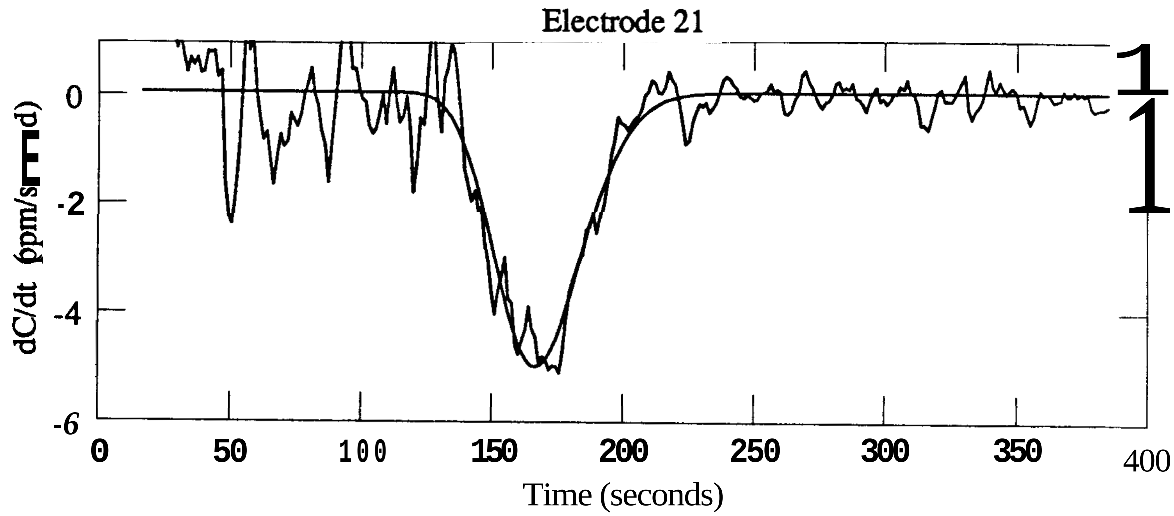

Electrode 51
0
-2
-4
-6
0
50
100
150
200
250
300
350
400
Time (seconds)

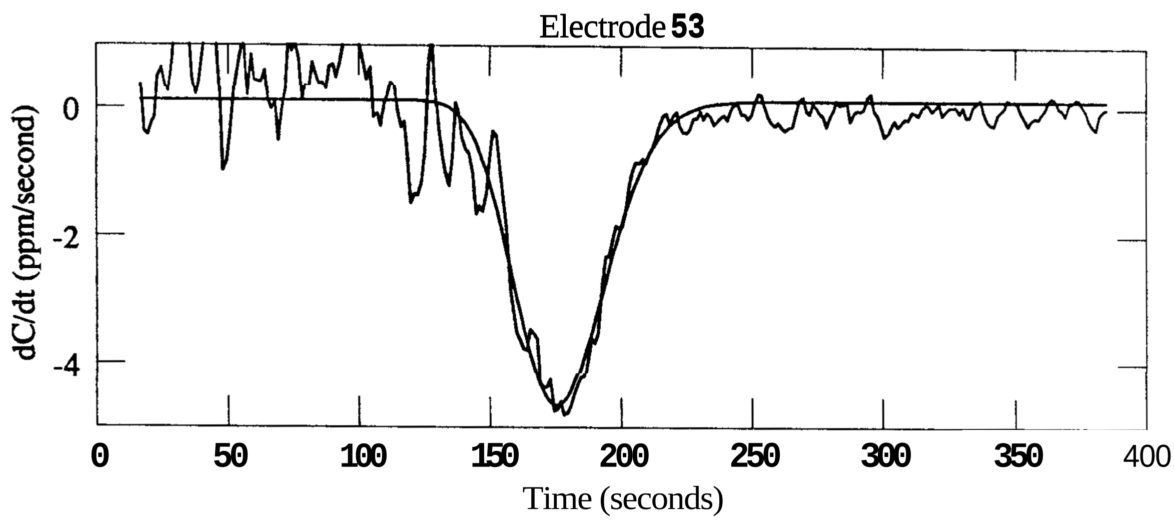

Electrode 83
dC/dt (ppm/second)
0
-2
-4
-6
0
50
100
150
200
250
300
350
400
Time (seconds)

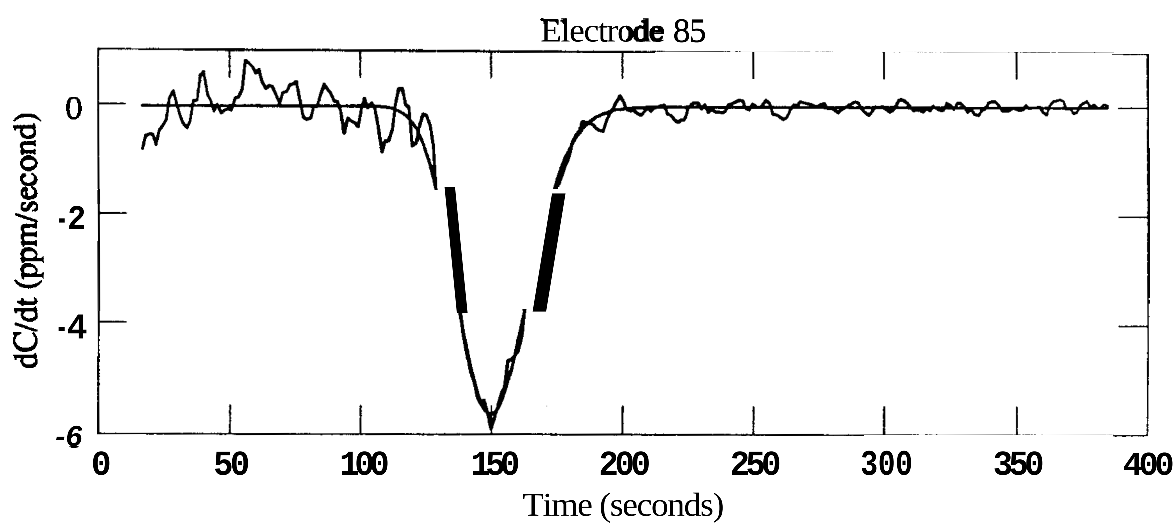

Electrode 20
dC/dt (ppm/second)
0
-2
-4
-6
0
50
100
150
200
250
300
350
400
Time (seconds)

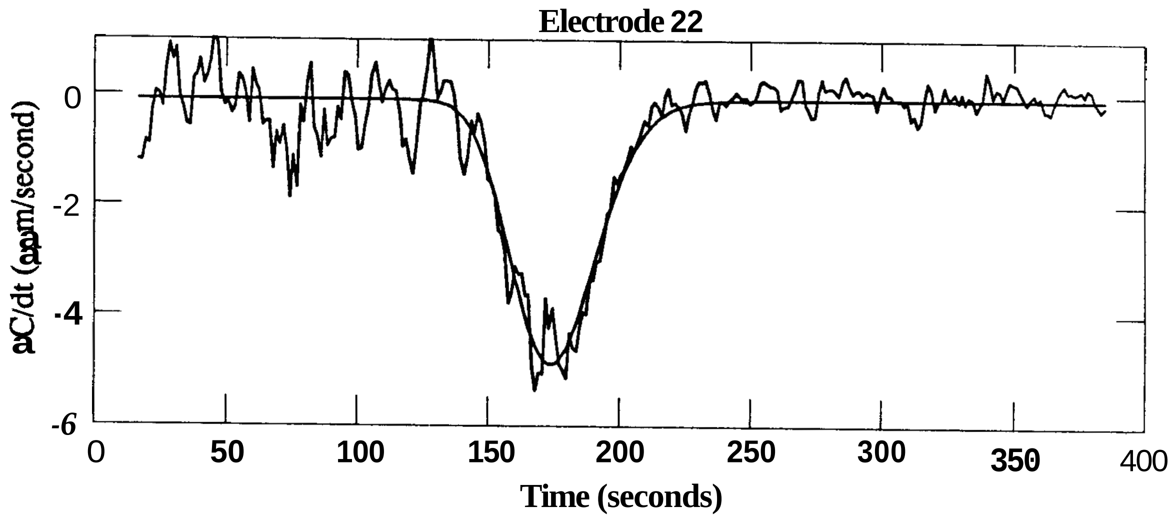

Electrode 52
dC/dt (ppm/second)
0
-2
-4
0
50
100
150
200
250
300
350
400
Time (seconds)


Electrode 84
dC/dt (ppm/second)
0
-2
-4
-6
0
50
100
150
200
250
300
350
400
Time (seconds)

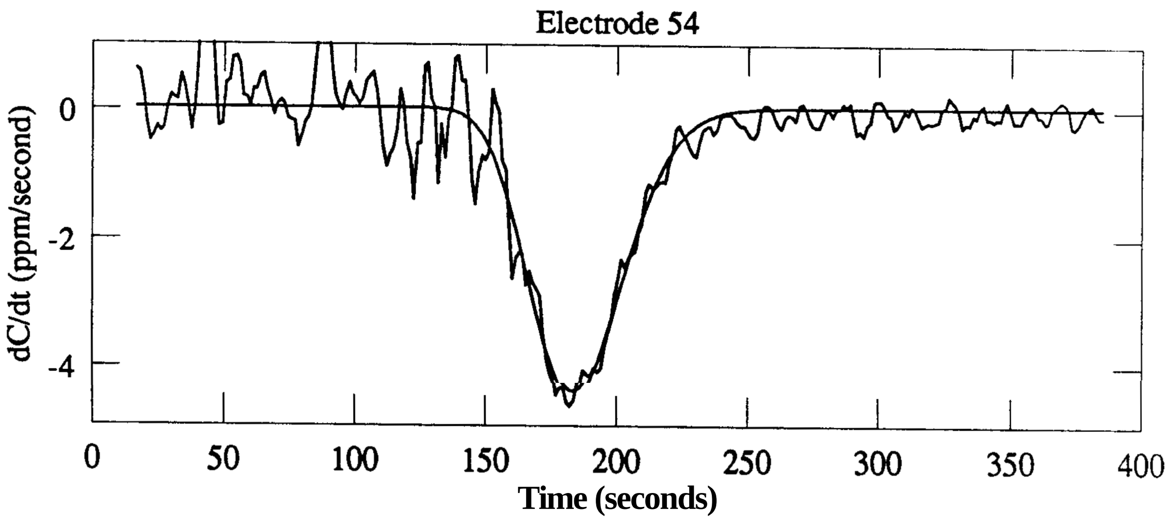

Electrode 21
dC/dt
0
-2
-4
-6
0
50
100
150
200
250
300
350
400
Time (seconds)


Electrode 53
dC/dt (ppm/second)
0
-2
-4
0
50
100
150
200
250
300
350
400
Time (seconds)

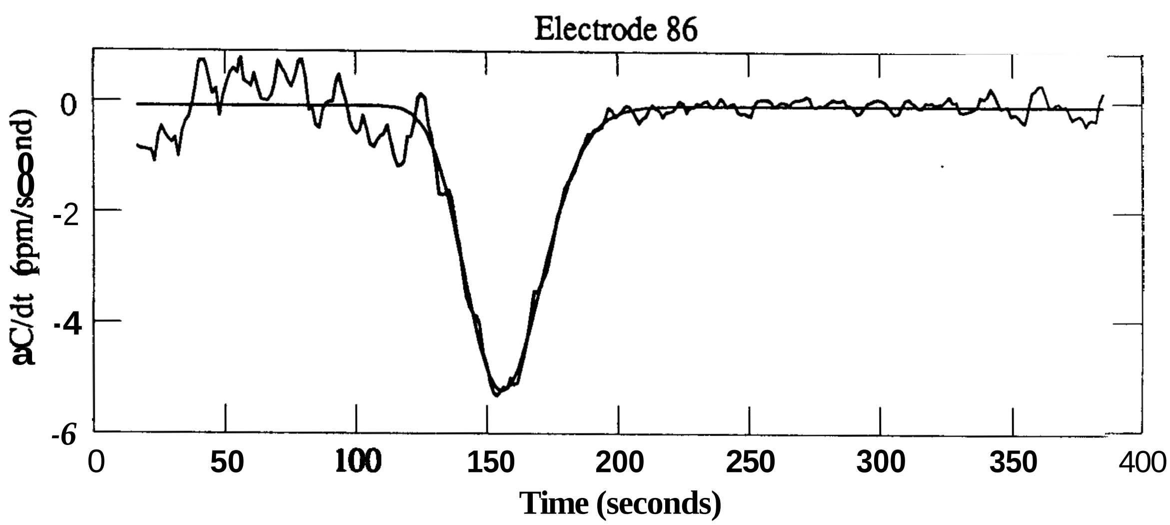

Electrode 85
dC/dt (ppm/second)
0
-2
-4
-6
0
50
100
150
200
250
300
350
400
Time (seconds)



Electrode 22
dC/dt (ppm/second)
0
-2
-4
-6
0
50
100
150
200
250
300
350
400
Time (seconds)

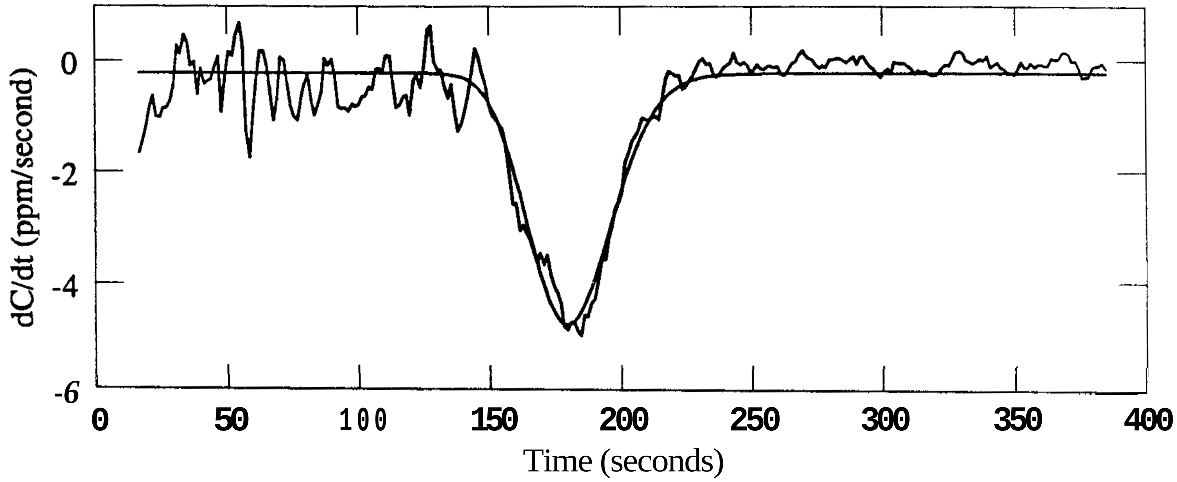


Electrode 54
dC/dt (ppm/second)
0
-2
-4
0
50
100
150
200
250
300
350
400
Time (seconds)

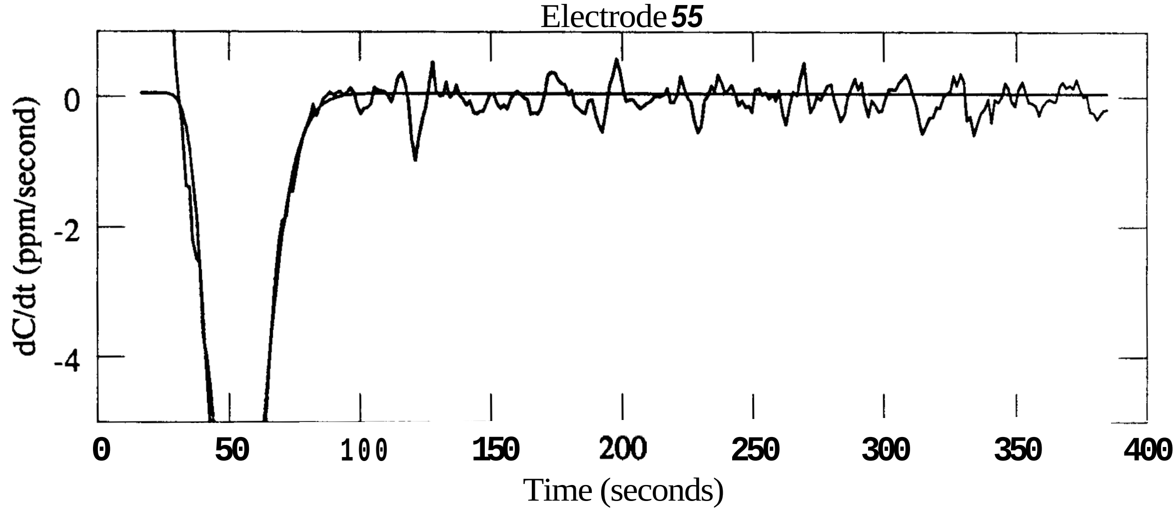


Electrode 86
dC/dt (ppm/second)
0
-2
-4
-6
0
50
100
150
200
250
300
350
400
Time (seconds)

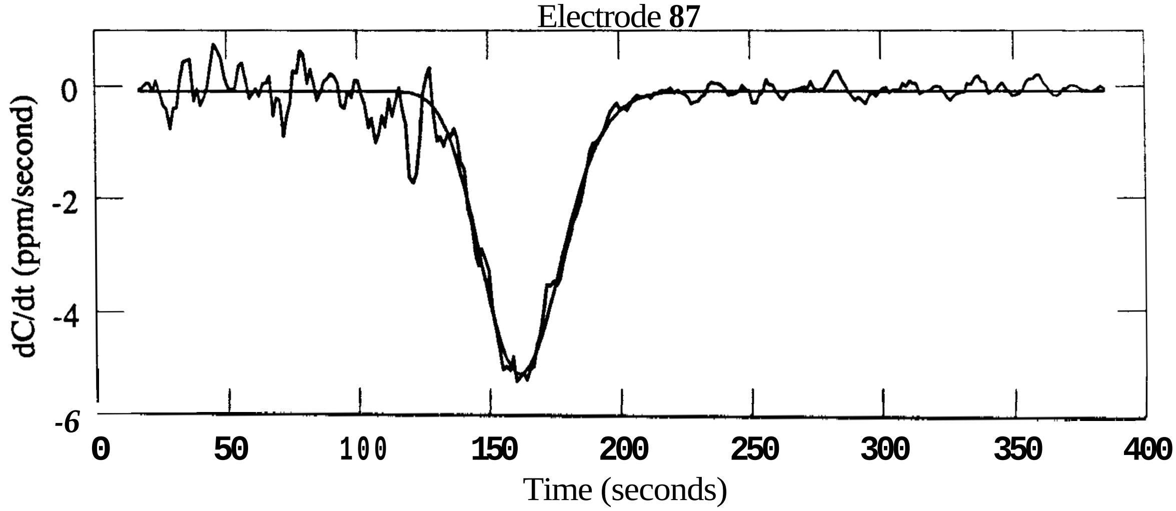
Electrode 23

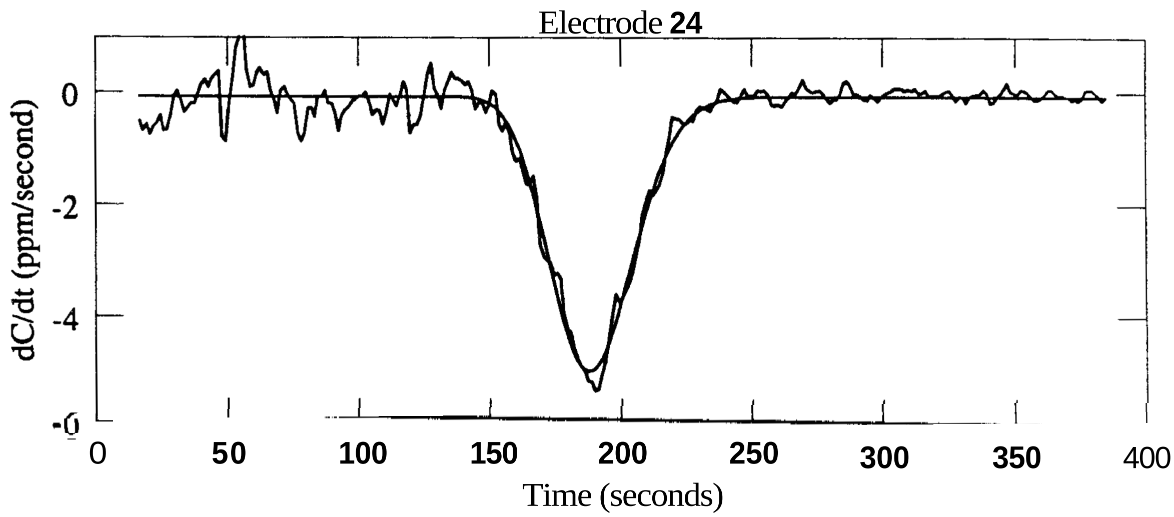


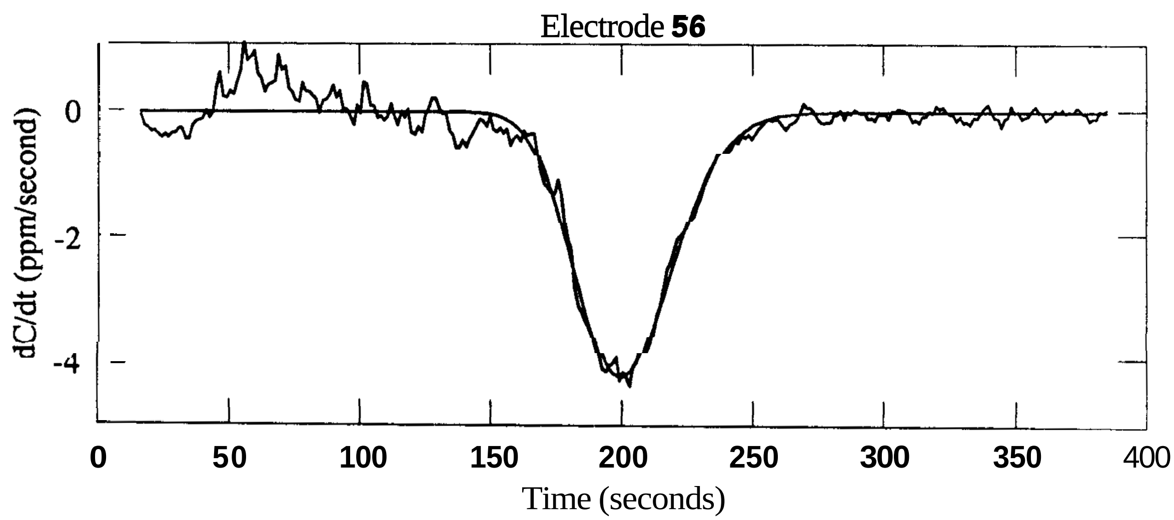
Electrode 55

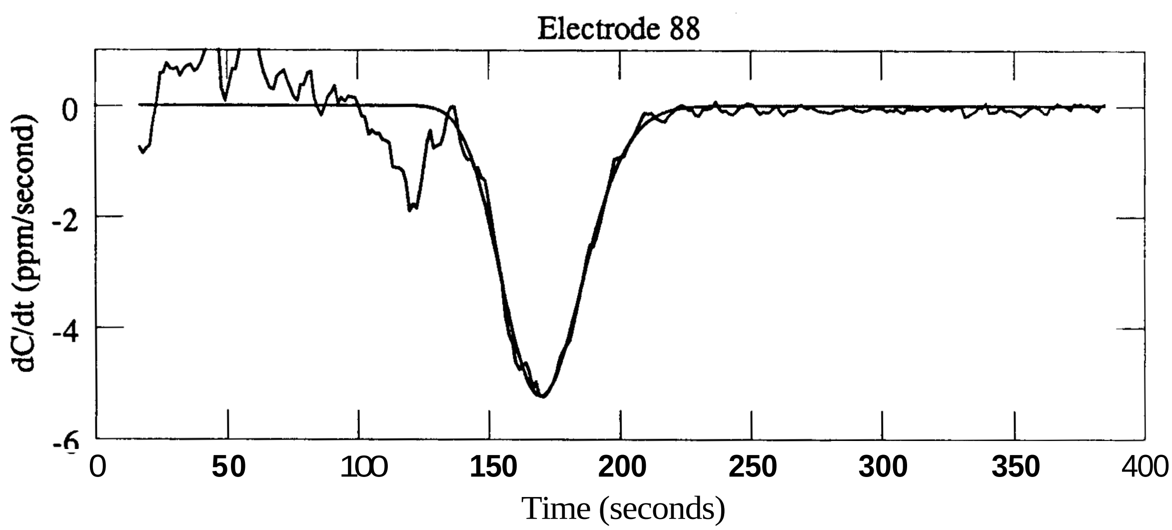


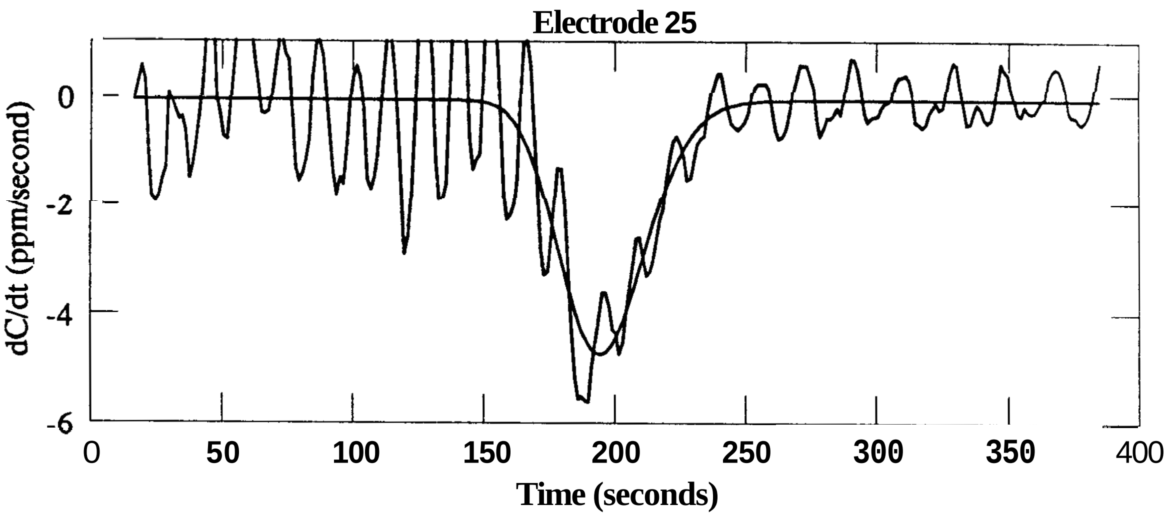
Electrode 87

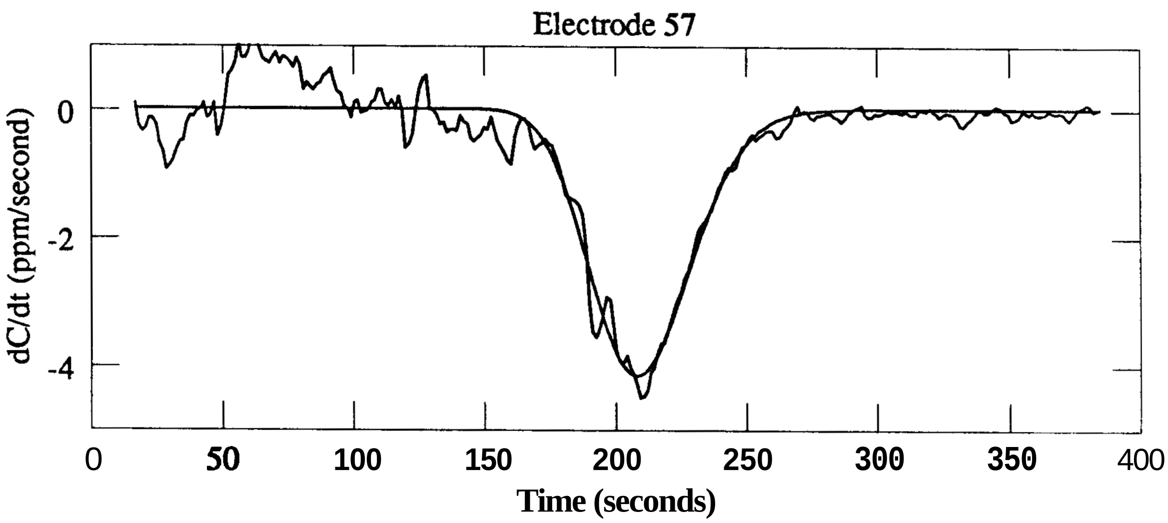


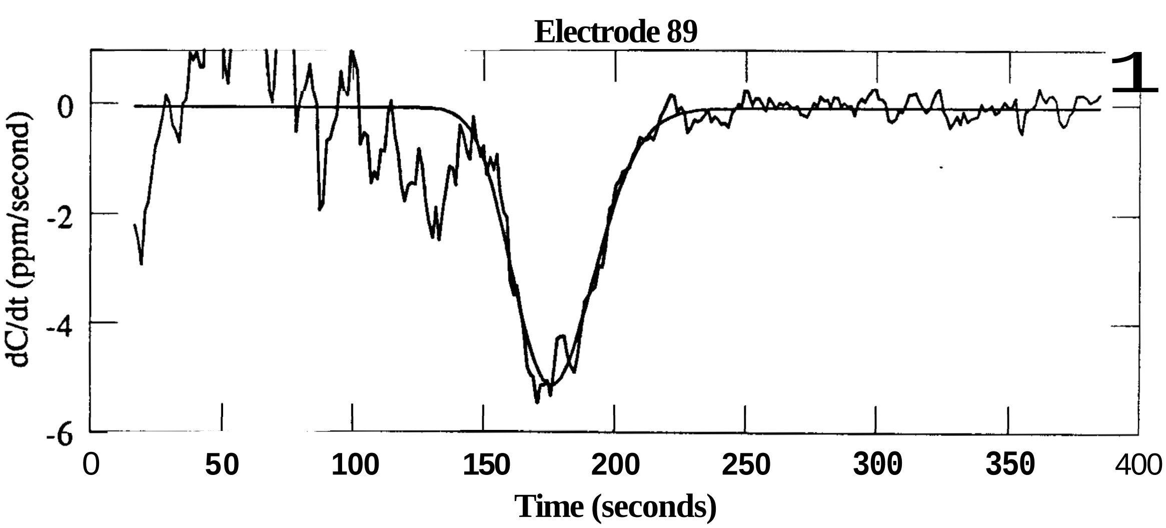

Electrode 24
dC/dt (ppm/second)
0
-2
-4
-6
0
50
100
150
200
250
300
350
400
Time (seconds)

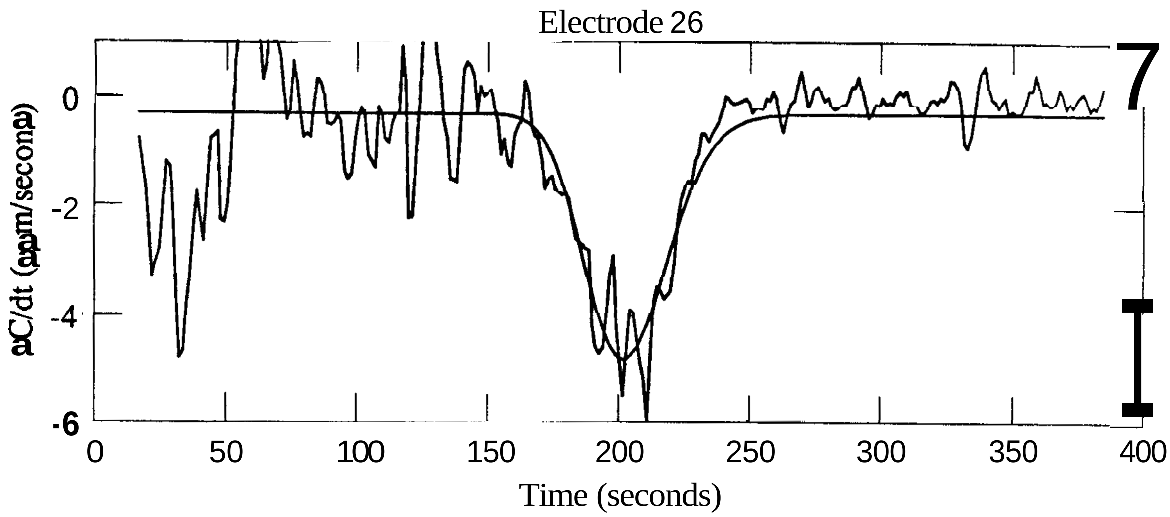

Electrode 56
dC/dt (ppm/second)
0
-2
-4
0
50
100
150
200
250
300
350
400
Time (seconds)

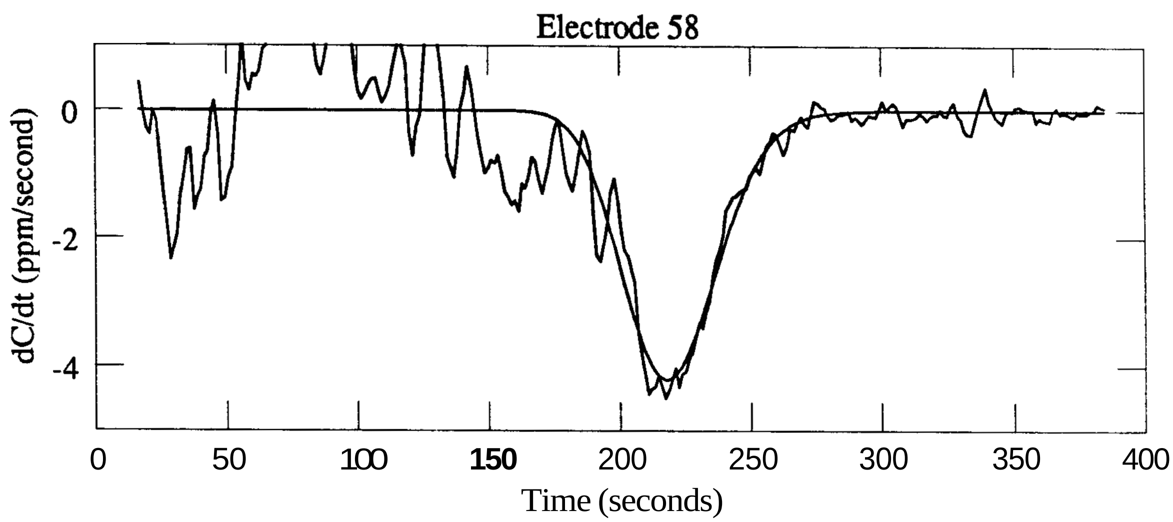

Electrode 88
dC/dt (ppm/second)
0
-2
-4
-6
0
50
100
150
200
250
300
350
400
Time (seconds)

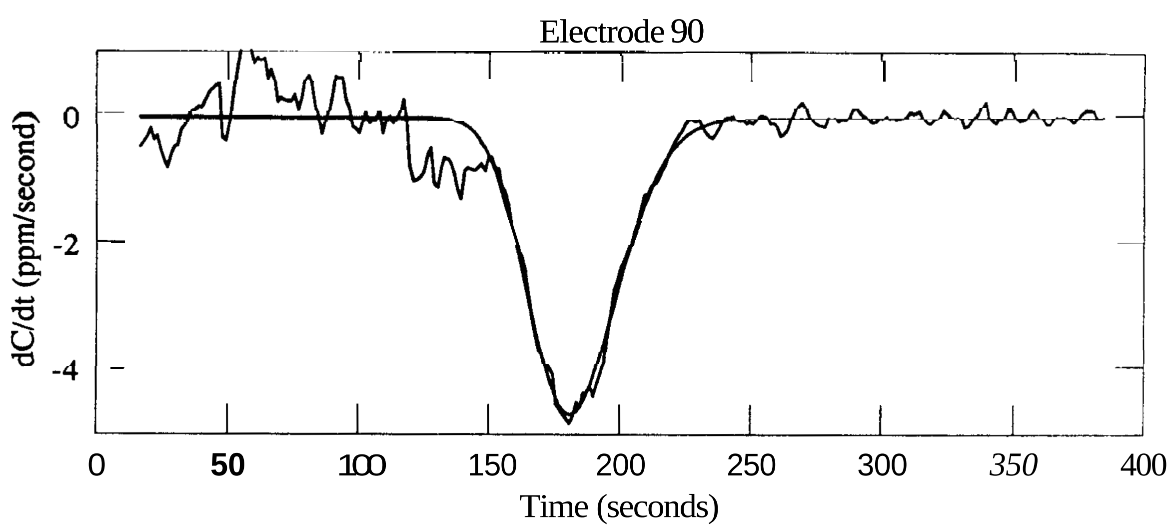

Electrode 25
dC/dt (ppm/second)
0
-2
-4
-6
0
50
100
150
200
250
300
350
400
Time (seconds)

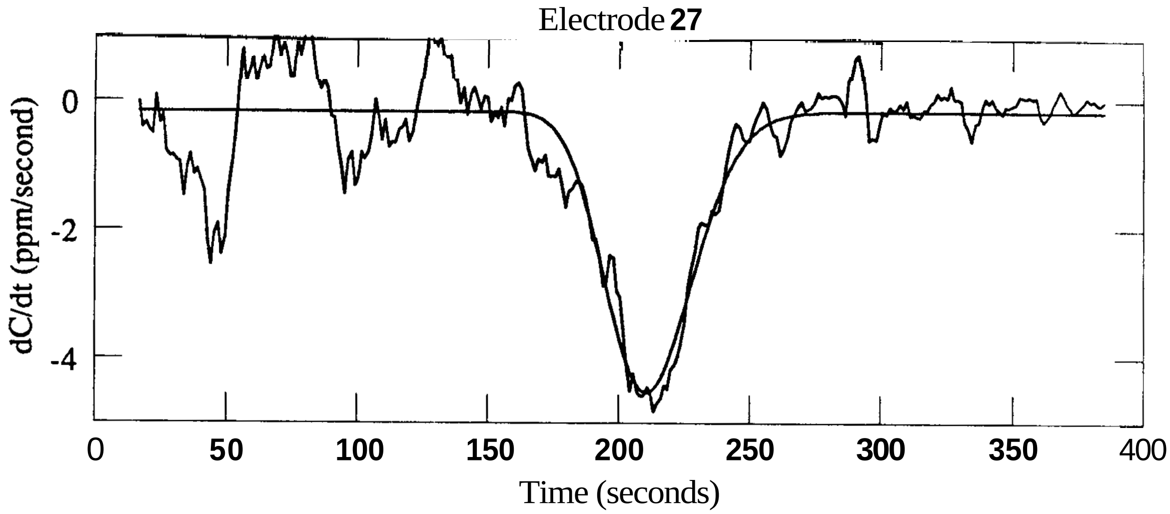

Electrode 57
dC/dt (ppm/second)
0
-2
-4
0
50
100
150
200
250
300
350
400
Time (seconds)

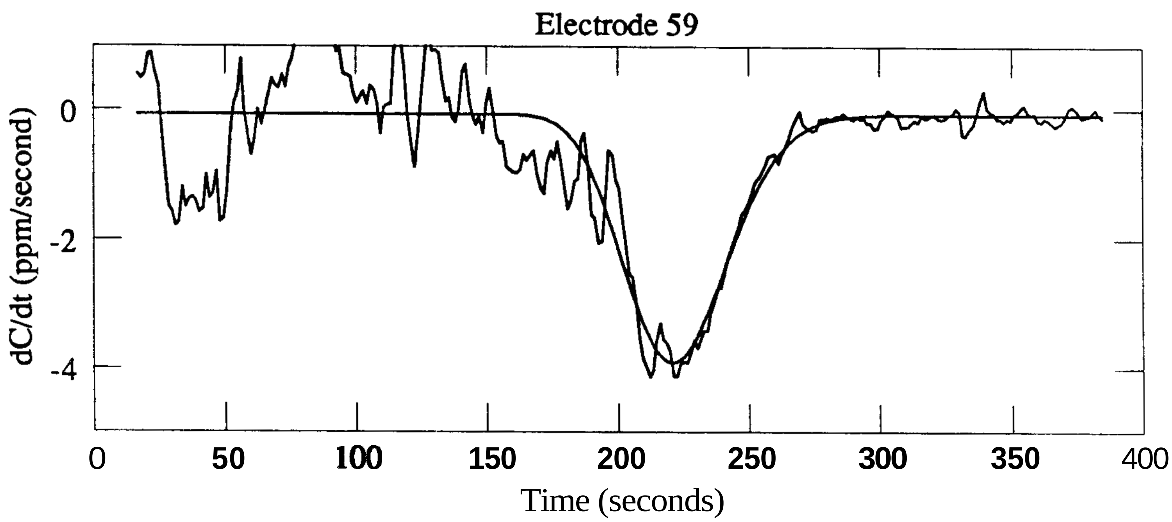

Electrode 89
dC/dt (ppm/second)
0
-2
-4
-6
0
50
100
150
200
250
300
350
400
Time (seconds)

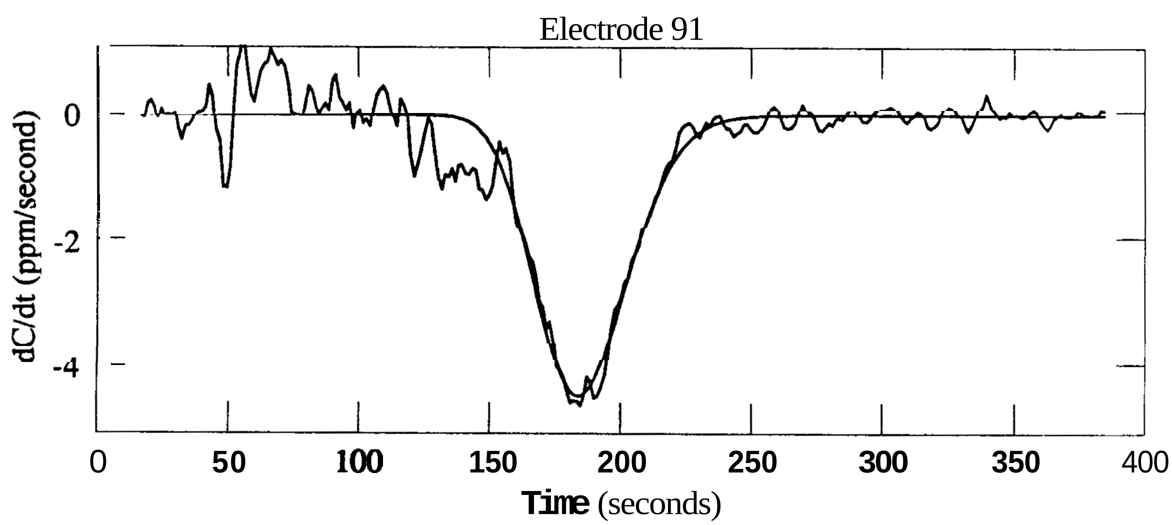

Electrode 26
dC/dt (ppm/second)
0
-2
-4
-6
0 50 100 150 200 250 300 350 400
Time (seconds)

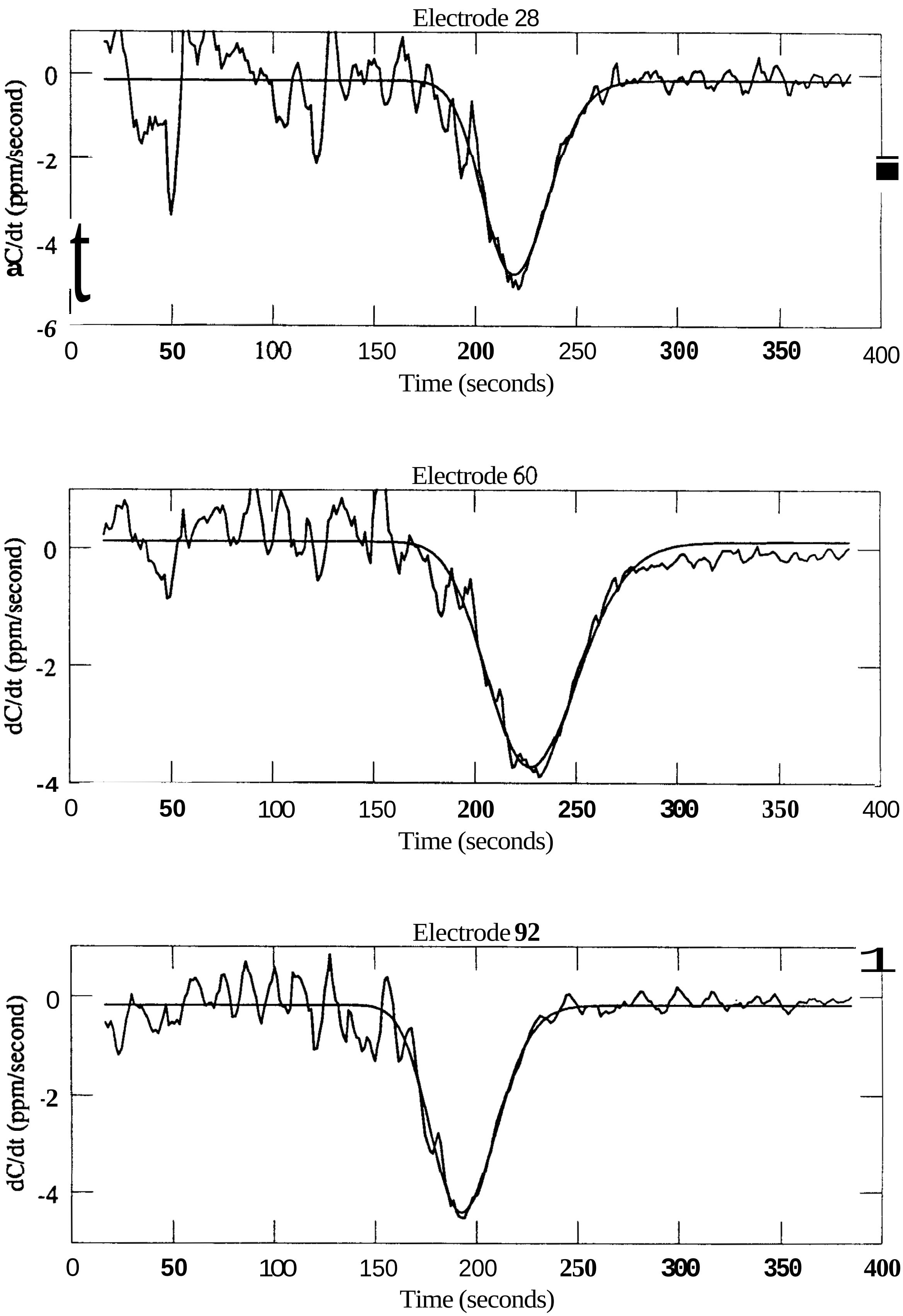

Electrode 58
dC/dt (ppm/second)
0
-2
-4
0 50 100 150 200 250 300 350 400
Time (seconds)

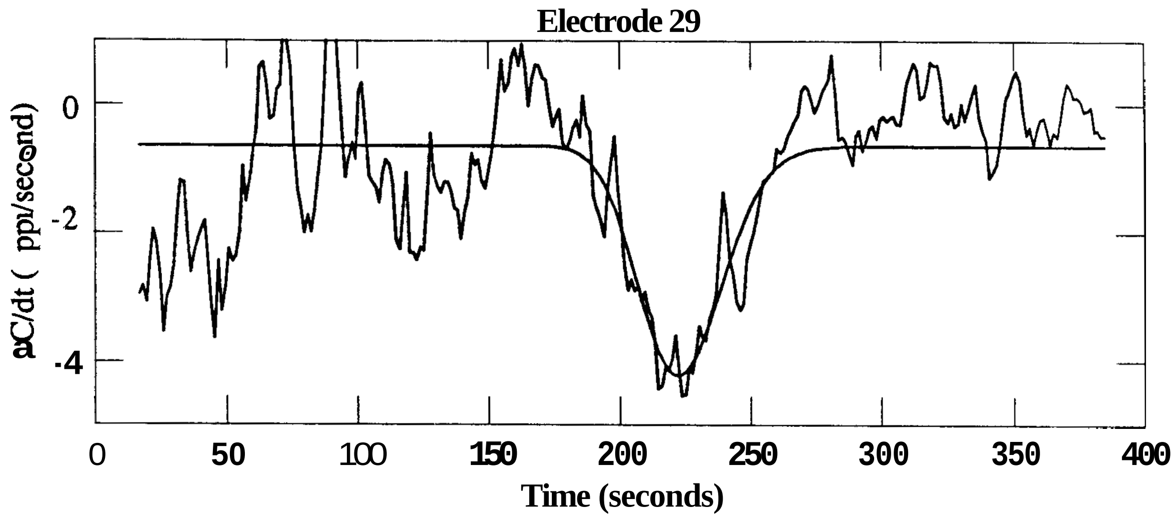

Electrode 90
dC/dt (ppm/second)
0
-2
-4
0 50 100 150 200 250 300 350 400
Time (seconds)

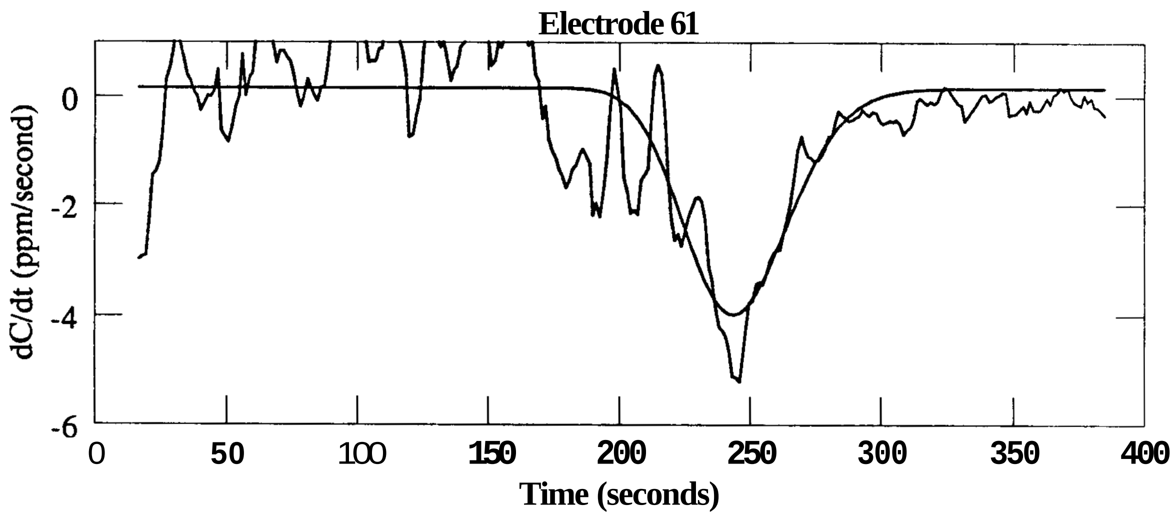

Electrode 27
dC/dt (ppm/second)
0
-2
-4
0
50
100
150
200
250
300
350
400
Time (seconds)

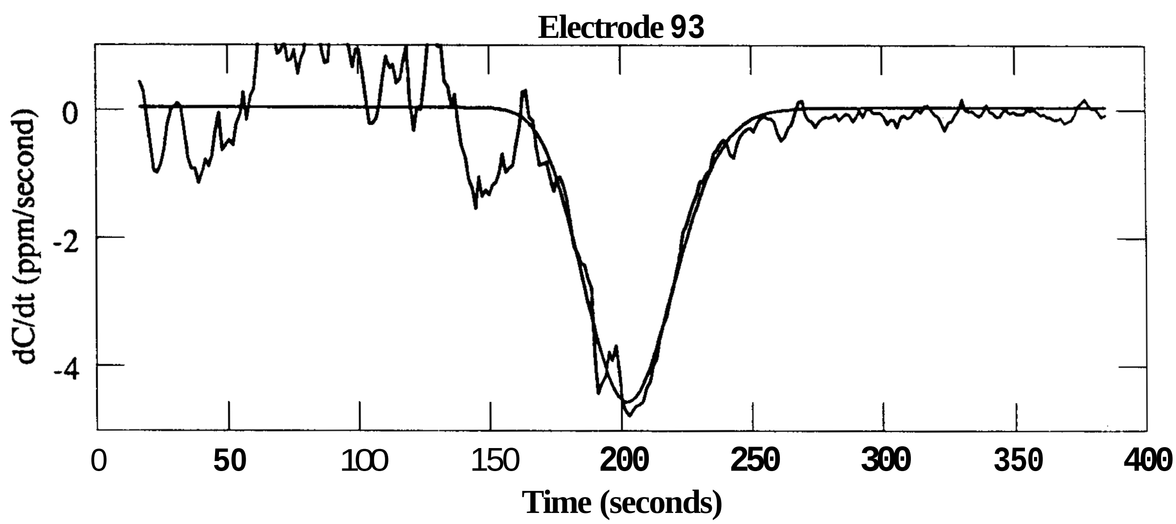

Electrode 59
dC/dt (ppm/second)
0
-2
-4
0
50
100
150
200
250
300
350
400
Time (seconds)

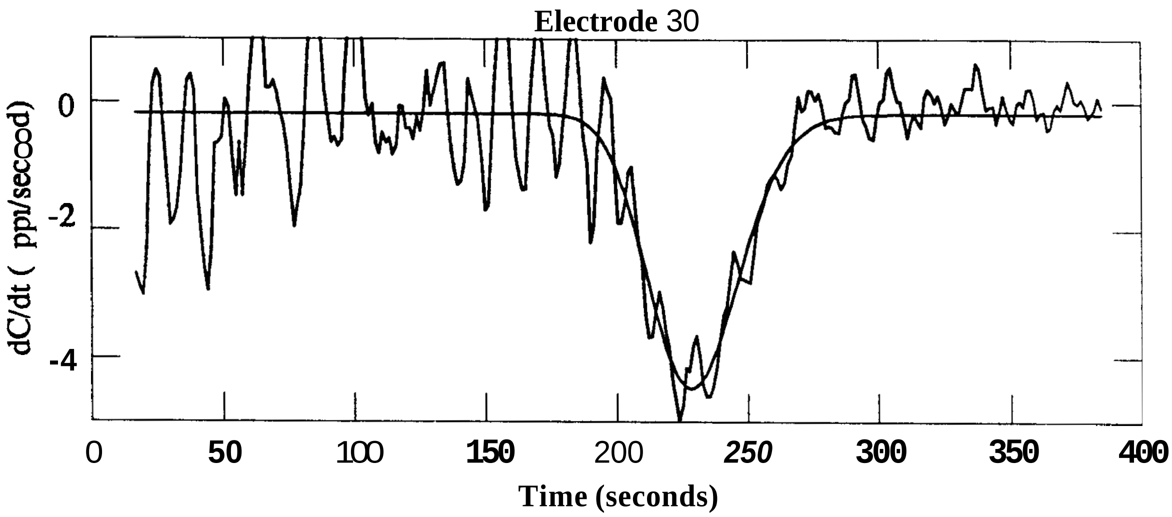

Electrode 91
dC/dt (ppm/second)
0
-2
-4
0
50
100
150
200
250
300
350
400
Time (seconds)

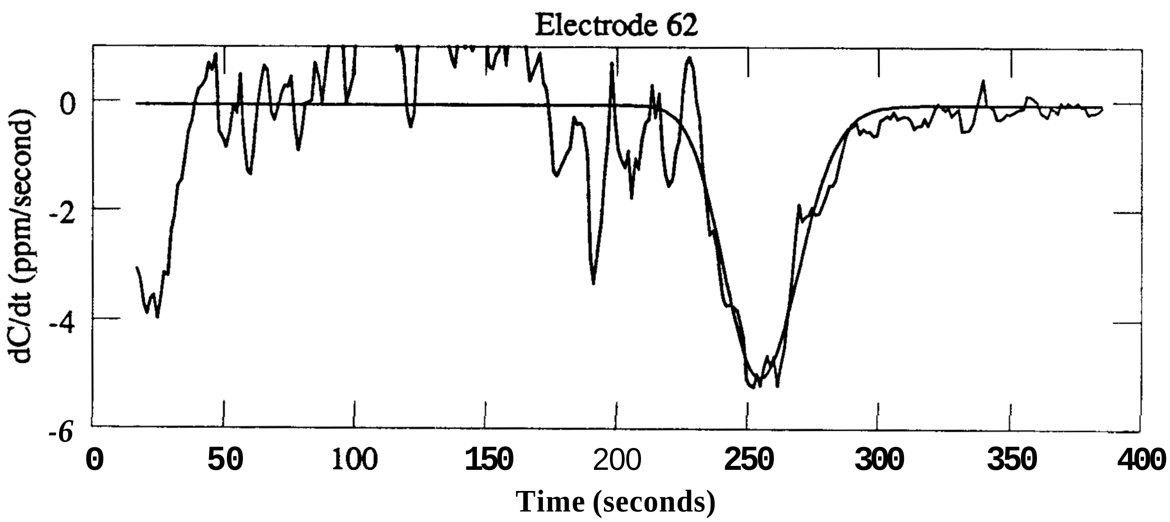

Electrode 28
dC/dt (ppm/second)
0
-2
-4
-6
0
50
100
150
200
250
300
350
400
Time (seconds)
Electrode 60
dC/dt (ppm/second)
0
-2
-4
0
50
100
150
200
250
300
350
400
Time (seconds)
Electrode 92
dC/dt (ppm/second)
0
-2
-4
0
50
100
150
200
250
300
350
400
Time (seconds)

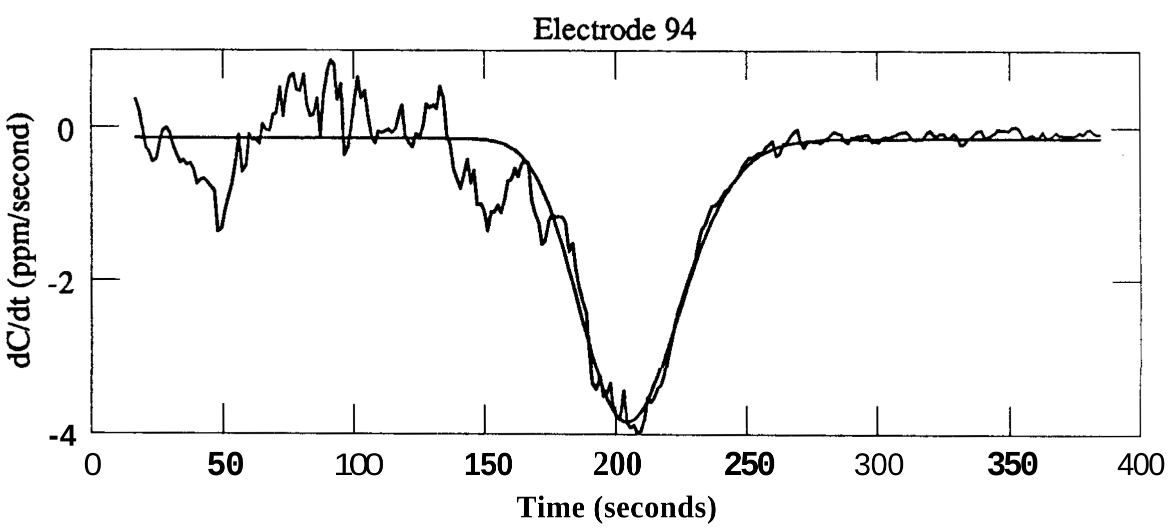

Electrode 29
dC/dt (ppm/second)
0
-2
-4
0
50
100
150
200
250
300
350
400
Time (seconds)

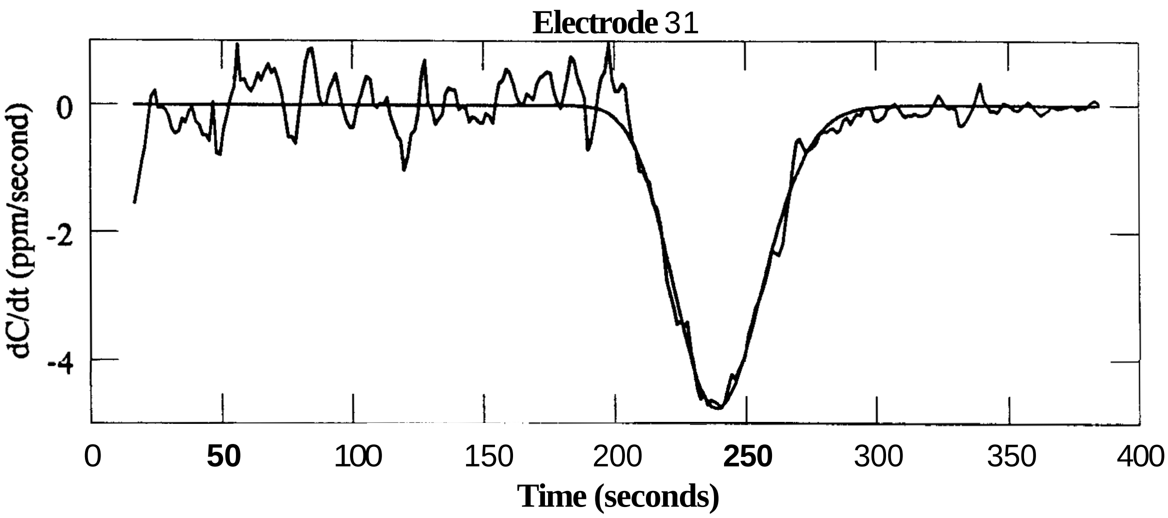

Electrode 61
dC/dt (ppm/second)
0
-2
-4
-6
0
50
100
150
200
250
300
350
400
Time (seconds)

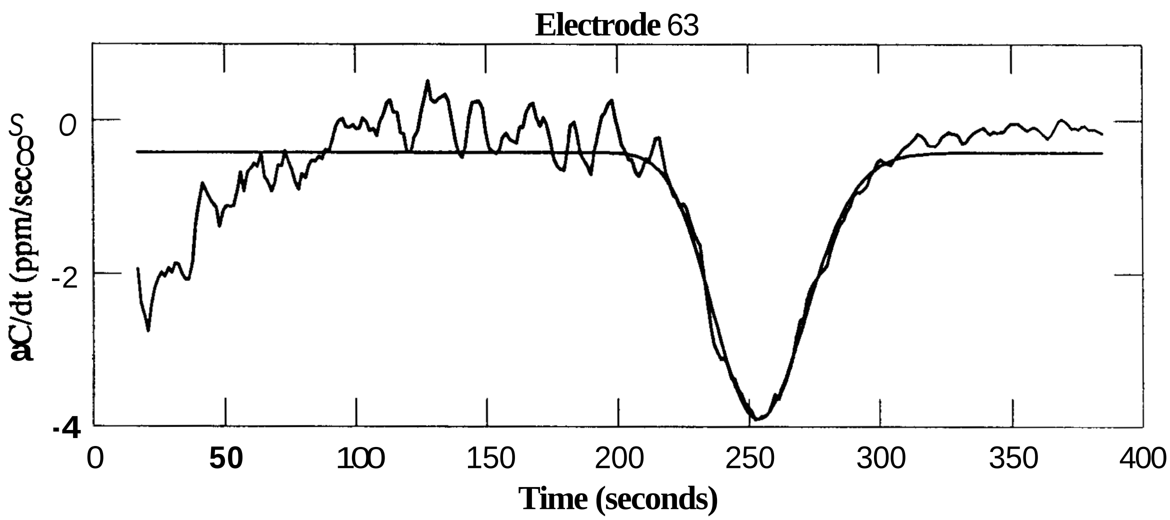

Electrode 93
dC/dt (ppm/second)
0
-2
-4
0
50
100
150
200
250
300
350
400
Time (seconds)

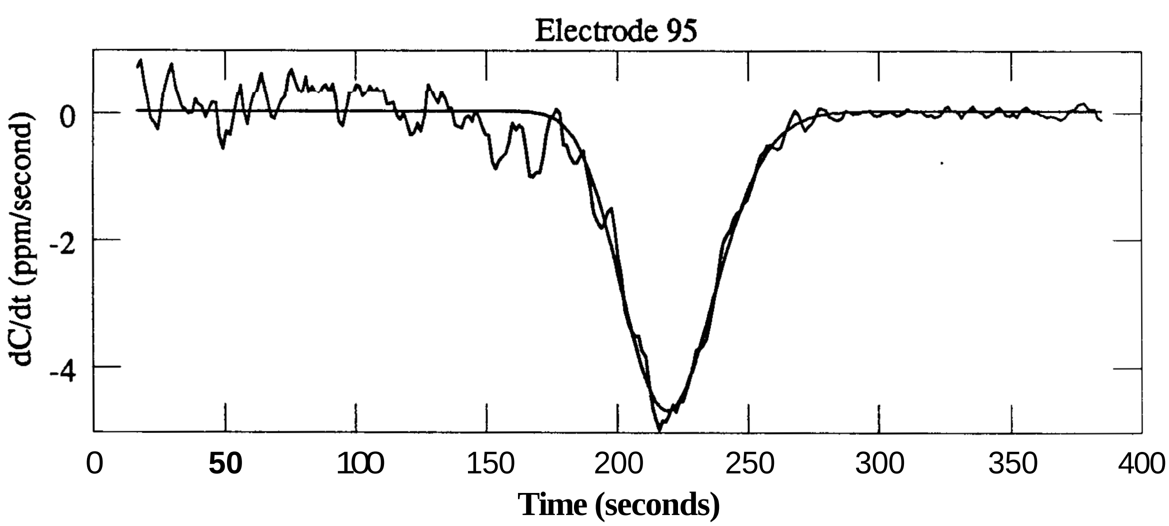

Electrode 30
dC/dt (ppi/secood)
0
-2
-4
0
50
100
150
200
250
300
350
400
Time (seconds)

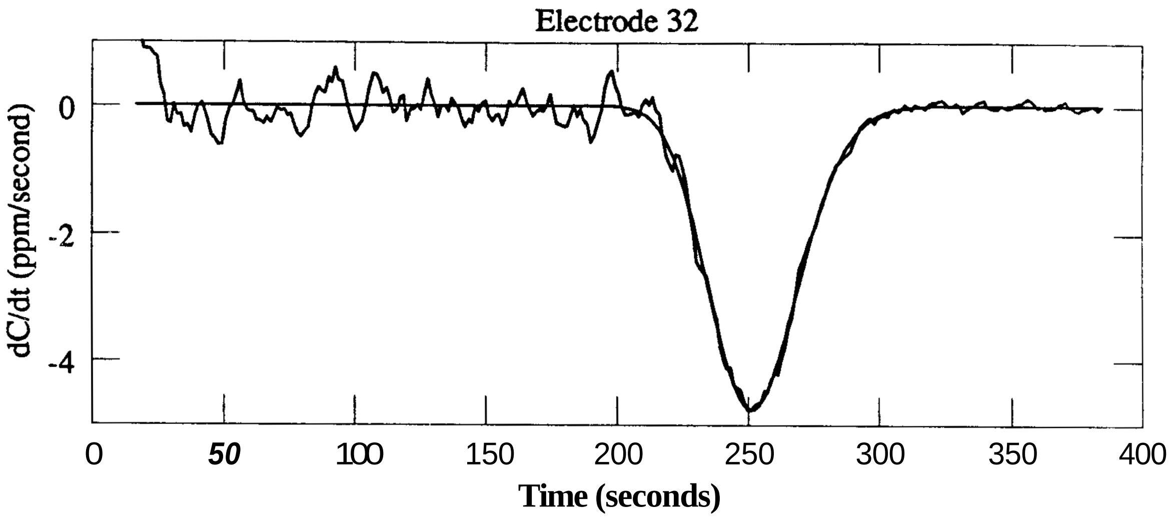

Electrode 62
dC/dt (ppm/second)
0
-2
-4
-6
0
50
100
150
200
250
300
350
400
Time (seconds)

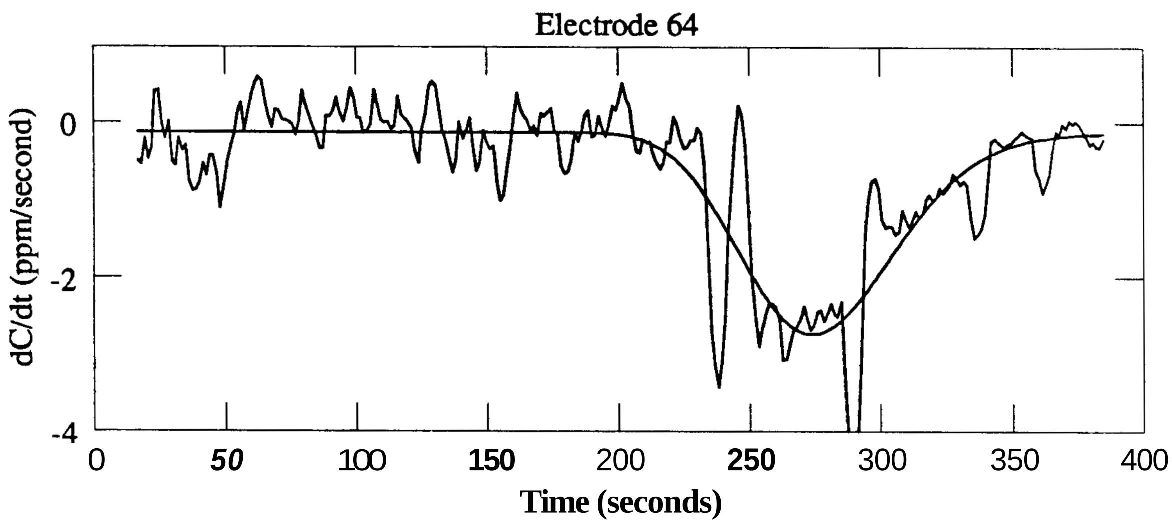

Electrode 94
dC/dt (ppm/second)
0
-2
-4
0
50
100
150
200
250
300
350
400
Time (seconds)

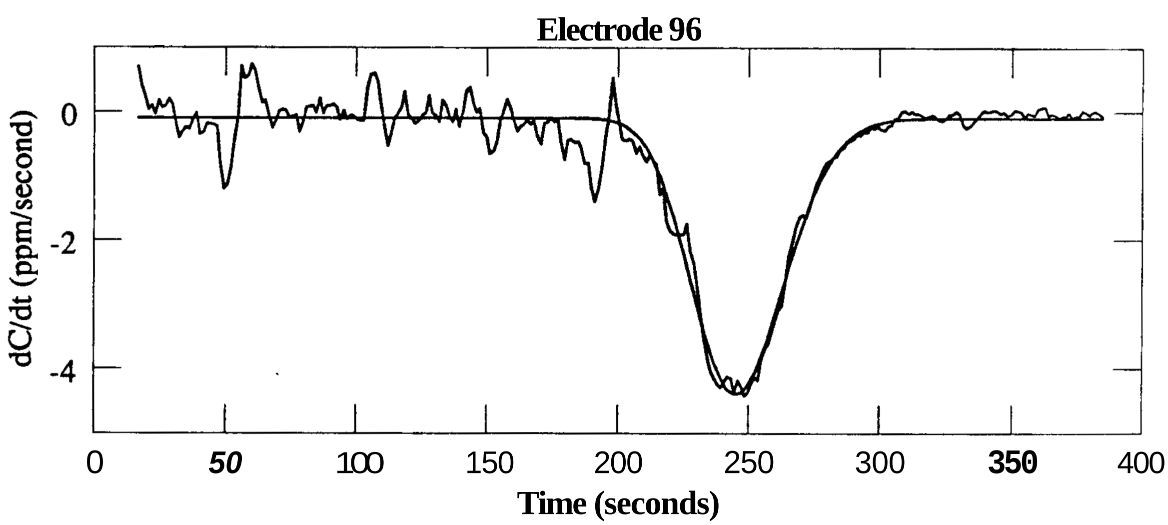

Electrode 31
dC/dt (ppm/second)
0
-2
-4
0
50
100
150
200
250
300
350
400
Time (seconds)


Electrode 63
dC/dt (ppm/second)
0
-2
-4
0
50
100
150
200
250
300
350
400
Time (seconds)


Electrode 95
dC/dt (ppm/second)
0
-2
-4
0
50
100
150
200
250
300
350
400
Time (seconds)


Electrode 32
dC/dt (ppm/second)
0
-2
-4
0
50
100
150
200
250
300
350
400
Time (seconds)


Electrode 64
dC/dt (ppm/second)
0
-2
-4
0
50
100
150
200
250
300
350
400
Time (seconds)


Electrode 96
dC/dt (ppm/second)
0
-2
-4
0
50
100
150
200
250
300
350
400
Time (seconds)

Appendix C: COMPUTER PROGRAM LISTINGS

This appendix comprises a complete set of program listings for the data acquisition programs listed **in** Table 1. They are:

```
1000 REM   *****************************************************************
1010 REM   *  GRAPHIC DISPLAY HEADER FOR SCAN500 DATA ACQUISITION DRIVER   *
1020 REM   *****************************************************************
1030 REM
1040 REM   *                 Program name:   SCAN5ØØ.BAS                   *
1050 REM   *
1060 REM   *                          Written for                          *
1070 REM   *      Stanford University Petroleum Engineering Department     *
1080 REM   *            by Lawrence W. Bouett, August 7, 1985              *
1090 REM
11ØØ REM   *****************************************************************
1110 :
1120 :
1130 REM     Thfs program prints a graphpic dtsplay header for the data
1140 REM   acquisition program package called SCAN5ØØ.  The program then
1150 REM   chains the program main menu called HEAD5ØØ.
1160 :
1170 REM   *****************************************************************
1180 REM   PRINT SCAN500 COMMERCIAL GRAPHIC DISPLAY HEADER
1190 REM   *****************************************************************
1200 :
121Ø WIDTH 8Ø:SCREEN Ø,Ø,Ø:LOCATE 1,1,Ø:CLS:KEY OFF
1220 GOSUB 9040                         :REM  Read the alphabet subroutine
1230 SCREEN 1
1240 :
1250 LOCATE 3,1,Ø
1260 PRINT "The"
127Ø :
128Ø DRAW "s4 cl bm11,5Ø"                :REM  Locate cursor
129Ø :
13ØØ DRAW S$                            :REM  Draw the ffrst part of "S"
131Ø DRAW SSUP$+SEP$                    :REM  Draw the second part of "S"
132Ø DRAW C$+SEP$                       :REM  Draw "C"
1330 DRAW CAS                           :REM  Fit "C" and "A" together
134Ø DRAW A$+SEP$                       :REM  Draw "A"
135Ø DRAW N$+SEP$                       :REM  Draw "N"
136Ø DRAW SEP$+SEP$                     :REM  Leave a space
137Ø DRAW FIVE$+SEP$                    :REM  Draw "5"
138Ø DRAW ZERO$+SEP$                    :REM  Draw "O"
139Ø DRAW ZEROS                         :REM  Draw "O"
14ØØ :
141Ø H=53                               :REM  Locate y-axis for painting
142Ø P=1                                :REM  Set paint color
143Ø :
1440 PAINT (3Ø,H),P                     :REM  Pafnt "S"
145Ø PAINT (73,H),P                     :REM  Pafnt "C"
146Ø PAINT (114,H),P                    :REM  Pafnt "A"
147Ø PAINT (146,H),P                    :REM  Paint "N"
148Ø PAINT (191,H),P                    :REM  Pafnt "5"
149Ø PAINT (251,H),P                    :REM  Palnt "Ø"
15ØØ PAINT (294,H),P                    :REM  Paint "Ø"
151Ø :
1520 LOCATE 14,5
1530 PRINT "Data Acquisitlon Software System"
1540 :
1550 DRAW "bmØ,35 d91 r319 u91 1319"    :REM  Draw a box around the message
1560 :
1570 FOR I=1 TO 1ØØØØ:NEXT I            :REM  Initiate time delay
1580 :
1590 CHAIN "HEAD5ØØ.EXE"                :REM  Chafn the program main menu
1600 :
1610 END
```

```
1000 REM ******************************************************************
1010 REM *              SCAN500 PROGRAM SUITE MAIN MENU                    *
1020 REM ******************************************************************
1030 REM *                                                                 *
1040 REM *                 Program name:   HEAD500.BAS                     *
1050 REM *                                                                 *
1060 REM *                        Written for                              *
1070 REM *     Stanford University Petroleum Engineering Department        *
1080 REM *          by Lawrence W. Bouett, March 25, 1986                  *
1090 REM *                                                                 *
1100 REM ******************************************************************
1110 :
1120 :
1130 WIDTH 80:SCREEN 0,0,0:LOCATE 1,1,0,0,7:CLS:KEY OFF
1140 :
1150 UL$=CHR$(201):UR$=CHR$(187):LR$=CHR$(188):LL$=CHR$(200)
1160 H$=CHR$(205):V$=CHR$(186):HS$=CHR$(196)
1170 :
1180 PRINT UL$H$H$H$H$H$H$H$H$H$H$H$H$H$H$H$H$H$H$H$H$H$H$H$H$H$H$H$H$H$;
1190 PRINT H$H$H$H$H$H$H$H$H$H$H$H$H$H$H$H$H$H$H$H$H$H$H$H$H$H$H$H$H$H$;
1200 PRINT H$H$H$H$H$H$H$H$H$H$H$H$H$H$H$H$H$H$UR$V$ TAB(80) VS;
1210 PRINT LL$H$H$H$H$H$H$H$H$H$H$H$H$H$H$H$H$H$H$H$H$H$H$H$H$H$H$H$H$H$;
1220 PRINT H$H$H$H$H$H$H$H$H$H$H$H$H$H$H$H$H$H$H$H$H$H$H$H$H$H$H$H$H$H$;
1230 PRINT H$H$H$H$H$H$H$H$H$H$H$H$H$H$H$H$H$H$LR$
1240 :
1250 LOCATE 2,4,0
1260 PRINT 'SCAN500 Data Acquisition System For KEITHLEY/das Series 500 Hardware'
1270 :
1280 LOCATE 6,1,0
1290 PRINT TAB(5) 'SCAN500 presents the following programs:'
1300 LOCATE 9,1,0
1310 PRINT TAB(10) "S     SCAN500 Data Acquisition Driver'
1320 PRINT TAB(10) 'F     FIX500 Data File Generator'
1330 PRINT TAB(10) "G     GRAPH500 Screen Graphic Display Driver'
1340 PRINT TAB(10) 'T     TEST500 Calibration And Test Program Suite"
1350 LOCATE 14,1,0
1360 PRINT TAB(10) "X     Terminate the program"
1370 LOCATE 17,1,1
1380 PRINT TAB(5) 'Enter your selection:  ";
1390 FOR 1-1 TO 4:SOUND RND*1000+37,2:NEXT I
1400 :
1410 X$=INKEY$
1428   IF X$="s" OR X$="S" THEN PROGRAM$="DATA500.EXE":GOTO 1530
1430   IF X$="f" OR X$="F" THEN PROGRAM$="FIX500.EXE":GOTO 1540
1440   IF X$="g" OR X$="G" THEN PROGRAM$="GRAPH500.EXE":GOTO 1550
1450   IF X$="t" OR X$="T" THEN PROGRAM$="TEST500.EXE":GOTO 1560
1460  IF X$="x" OR X$="X" THEN PRINT "X":GOTO 1490
1470 GOTO 1410
1480 :
1490 LOCATE 17,1,0
1500 PRINT TAB(5) 'Program terminated ..." TAB(80)
1510 FOR I=1 TO 4000:NEXT I:END
1520 :
1530 DISPLAY$="DATA500 Data Acquisition Driver":GOTO 1580
1540 DISPLAY$="FIX500 Data File Generator":GOTO 1580
1550 DISPLAY$="GRAPH500 Screen Graphic Display Driver":GOTO 1580
1560 DISPLAY$="TEST500 Calibration and Test Program Suite'
1570 :
1580 LOCATE 17,1,0:PRINT TAB(5) 'Loading " DISPLAYS " ..." TAB(80)
1590 FOR I=1 TO 4000:NEXT 1
1600 CHAIN PROGRAMS                    :REM  Chain the selected program
1610 :
1620 END
```

```
1000 REM   *****************************************************************
1010 REM   * DATA ACQUISITION DRIVER FOR KEITHLEY/DAS SERIES 500 HARDWARE *
1020 REM   *****************************************************************
1030 REM   *                                                               *
1040 REM   *              Program name:   DATA500.BAS                      *
1050 REM   *                                                               *
1060 REM   *                         Written for                           *
1070 REM   *    Stanford University Petroleum Engineering Department       *
1080 REM   *             by Lawrence W.  Bouett, July 2, 1985              *
1090 REM   *                                                               *
1100 REM   *****************************************************************
1110 :
1120 :
1130 REM   *****************************************************************
1140 REM   CONFIGURATION OF THE KEITHLEY/DAS SERIES 500 HARDWARE
1150 REM   *****************************************************************
1160 :
1170 REM     The Series 500 hardware comprises a box with a power supply
1180 REM   and six cards.  The cards are:
1190 :
1200 REM                       Card      Slot   Remarks
1210 REM                       ------    ----   -----------------
1220 REM                       AMM1        1    A/D Input voltage
1230 REM                       AOM1/2      2    D/A output voltage
1240 REM                       blank       3
1250 REM                       AIM3        4    Left 32 channels
1260 REM                       AIM3        5    Center 32 channels
1270 REM                       AIM3        6    Right 32 channels
1280 REM                       blank       7
1290 REM                       blank       8
1300 REM                       blank       9
1310 REM                       DIO1       10
1320 :
1330 REM     There is an IBM Interface card in slot 3 of a COMPAQ(TM) personal
1340 REM   computer which is used to drive the Series 500 hardware.  Output
1350 REM   voltage range from the AOM1/2 card is calibrated at plus or minus
1360 REM   2.5 volts.  Actual output voltage is software selected at plus or
1370 REM   minus one volt.  Output voltage is applied on software command in
1380 REM   parallel to all 96 electrodes.  A global gain of 1 is applied to
1390 REM   all input voltages.
1400 :
1410 REM   *****************************************************************
1420 REM   VARIABLE LIST AND DESCRIPTIONS
1430 REM   *****************************************************************
1440 :
1450 REM   Name          Type    Use
1460 REM   ------------  ------  ........................................
1470 REM   A             SNG     Dummy variable for disk file handling
1480 REM   A$            STR     Dummy variable for disk file handling
1490 REM   ADDRESS(I)    SNG     CMDA address for SLOT(I)
1500 REM   B             SNG     Dummy variable for disk file handling
1510 REM   B$            STR     Dummy variable for disk file handling
1520 REM   C             SNG     Dummy variable for disk file handling
1530 REM   C$            STR     Dummy variable for disk file handling
1540 REM   CCO           SNG     Current count counter one
1550 REM   CCZ           SNG     Current count counter zero
1560 REM   CHANNEL       INT     Analog input channel (SLOT dependent)
1570 REM   CO            SNG     Loaded count counter one
1580 REM   COLUMN        INT     Screen output column
1590 REM   COUNTER       INT     Number of datasets between screen displays
1600 REM   CURRENTCOUNT  DBL     Number of clock pulses from clock loading
1610 REM   CYCLE         INT     Polarity of analog output voltage
1620 REM   CZ            SNG     Loaded count counter zero
1630 REM   D             SNG     Dummy variable for disk file handling
```

```
1640 REM  DATASETS      INT   Total number of datasets scanned
1650 REM  DATES         STR   The date from the internal clock
1660 REM  DELAY         SNG   Tlmc delay in seconds between datasets
1670 REM  DEHOFLAG      INT   Flag to toggle a demonstratlon run
1680 REM  DHIGH(I)      INT   High byte result of A/D voltage conversion
1690 REM  DISPLAYFLAG   INT   Flag to toggle screen graphlc dlsplay
1700 REM  DLOW(I)       INT   Low byte result of A/D voltage converslon
1710 REM  DRES          SNG   Reconstituted raw datum from A/D conversion
1720 REM  DUMMY         INT   Twelve-blt voltage step for demonstration run
1730 REM  DVOLTS        SNG   Converslon of DRES to voltage
1740 REM  DVOLTS(I)     SNG   Voltage element corrected for global gain
1750 REM  E             SNG   Dummy varlable for disk flle handling
1760 REM  ELECTRODES    INT   Number of electrodes scanned for each dataset
1770 REM  ENDTIME       SNG   End tlme of dataset scan in seconds
1780 REM  ENDTIMEE      STR   Index end clock tlme of run
1790 REM  ETIME         SNG   Elapsed ttme of dataset scan in seconds
1800 REM  F             SNG   Dummy varlable for dlsk flle handling
1810 REM  FILENAMES     STR   Name of .LOG permanent log ftle
1820 REM  FINISH        INT   Origln of last graph for screen dlsplay
1830 REM  FIRST         INT   First slot used for analog lnput
1840 REM  FREQUENCY     1NT   Frequency of graphlc screen dlsplay updates
1850 REM  G             SNG   Dummy varlable for disk file handllng
1860 REM  HS            STR   Dummy variable for graphlc dlsplay
1870 REM  HAN           INT   High byte negative analog voltage output
1880 REM  HAP           INT   Hlgh byte positive analog voltage output
1890 REM  HO            INT   Hlgh byte counter one (loaded or read)
1900 REM  HSS           STR   Dummy varlable for graphic display
1910 REM  HZ            INT   Hlgh byte counter zero (loaded or read)
1920 REM  I             INT   Index for dataset element (1-32 or 1-96)
1930 REM  IDLETIME      SNG   Current delay between datasets in seconds
1940 REM  LAN           INT   Low byte negative analog voltage output
1950 REM  LAP           INT   Low byte positive analog voltage output
1960 REM  LAST          INT   Last slot used for analog tnput
1970 REM  LO            INT   Low byte counter one (loaded or read)
1980 REM  LLS           STR   Dummy varlable for graphlc dlsplay
1990 REM  LRS           STR   Dummy varlable for graphlc dlsplay
2000 REM  LZ            INT   Low byte counter zero (loaded or read)
2010 REM  N             INT   Seconds read from TIMES
2020 REM  NERROR        INT   Runtime error number
2030 REM  OLDTIME       SNG   Start time of run in seconds
2040 REM  PRESSURE      SNG   Barometric pressure during run
2050 REM  QUITFLAG      INT   Flag to direct termination of run
2060 REM  ROW           INT   Screen output row number
2070 REM  ROWS          INT   Number of rows of 32 electrodes used (1 or 3)
2080 REM  SLOT          INT   Index for hardware slot location
2090 REM  START         INT   Origin of flrst graph for screen dlsplay
2100 REM  STARTFLAG     INT   Flag to allow lndexlng start tlme of run
2110 REM  STARTTIME     SNG   Start tlme of dataset in seconds
2120 REM  STARTTIMES    STR   Index start clock ttme of run
2130 REM  STEPS         INT   Number of voltage steps for demonstration run
2140 REM  TEMPERATURE   INT   Ambient temperature durlng run
2150 REM  TIME          SNG   Absolute elapsed tlme on clock since loading
2160 REM  TIMES         STR   The ttme from the internal clock
2170 REM  TIMELIMIT     INT   Ttme limit of run in seconds
2180 REM  TIMEOUTFLAG   INT   Flag to direct status dlsplay after termlnatlon
2190 REM  TOTALTIME     SNG   Elapsed tlme of run in seconds
2200 REM  ULS           STR   Dummy varlable for graphlc dlsplay
2210 REM  URE           STR   Dummy variable for graphic dlsplay
2220 REM  VS            STR   Dummy variable for graphic dlsplay
2230 REM  X             SNG   Dummy argument for INT(x) converslon
2240 REM  XS            STR   Dummy variable for INKEYS function
2250 REM  XAXIS         SNG   X-coordinate for screen graphlc dlsplay
2260 REM  Y(I)          SNG   Y-coordinate for screen graphlc dlsplay
2270 REM  YBASE         INT   Location of X axis for each graph on screen
```

```
2280 REM  YOLD(I)        SNG      Previous Y-coordinate for screen graphic display
2290 :
2300 REM  ********************************************************************
2310 REM  PRINT THE SIGN-ON HEADER
2320 REM  ********************************************************************
2330 :
2340 GOSUB 7040                          :REM  Read string subroutine
2350 LOCATE 2,1,Ø                        :REM  Print the sign-on message
2360 PRINT UL$H$H$H$H$H$H$H$H$H$H$H$H$H$H$H$H$H$H$H$H$H$H$H$H$H$H$H$H$H$H$H$:
2370 PRINT H$H$H$H$H$H$H$H$H$H$H$H$H$H$H$H$H$H$H$H$H$H$H$H$H$H$H$H$H$H$H$H$; .
2380 PRINT HSHSHSHSHSHSHSHSHSHSHSHSHSHSHSURSVS TAB(8Ø) VSVS TAB(8Ø) V$;
2390 PRINT VS TAB(34) "S C A N  5 O Ø" TAB(8Ø) VSVS TAB(8Ø) V$;
2400 PRINT VS TAB(16) 'Data Acquisition Driver for Dispersion Experiments";
2410 PRINT TAB(8Ø) VSVS TAB(8Ø) VSVS TAB(8Ø) VSVS TAB(34) H$H$H$H$H$H$H$H$H$;
2420 PRINT HSHSHSHSHS TAB(8Ø) VSVS TAB(8Ø) V$;
2430 PRINT VS TAB(2Ø) 'Written 02 July 1985 by Lawrence W. Bouett" TAB(8Ø) VS;
2440 PRINT V$ TAB(8Ø) VS:
2450 PRINT V$ TAB(15) "Stanford University Petroleum Engineering Department";
2460 PRINT TAB(8Ø) VSVS TAB(8Ø) VSVS TAB(8Ø) VS:
2470 PRINT LL$H$H$H$H$H$H$H$H$H$H$H$H$H$H$H$H$H$H$H$H$H$H$H$H$H$H$H$H$H$H$H$;
2480 PRINT H$H$H$H$H$H$H$H$H$H$H$H$H$H$H$H$H$H$H$H$H$H$H$H$H$H$H$H$H$H$H$H$;
2490 PRINT H$H$H$H$H$H$H$H$H$H$H$H$H$H$H$LR$
2500 PRINT:PRINT
2510 PRINT "Press the RETURN kev to continue.  ":
2520 LOCATE ,,1,Ø,7                      :REM  Wait to start the program
253Ø FOR I=1 TO 4:SOUND RND*1ØØØ+37,2:NEXT I
254Ø X$=INKEY$
255Ø  IF X$=CHR$(13) GOTO 262Ø
256Ø GOTO 254Ø
257Ø :
258Ø REM  ********************************************************************
259Ø REM  PRINT THE PERMANENT LOG HEADER
26ØØ REM  ********************************************************************
261Ø :
2620 GOSUB 7130                          :REM  Print the permanent log header
2630 LOCATE 2,4,Ø
2640 PRINT "SCAN500 Data Acquisition Driver Permanent Log"
2650 :
2660 LOCATE 5,1,0                        :REM  Request permanent log information
2670 PRINT TAB(1Ø) 'Please enter the following information for the ";
2680 PRINT "permanent log:"
2690 LOCATE 8,1,1
2700 PRINT TAB(2Ø) 'Is this a demonstration run?  ";
2710 X$=INKEY$
2720  IF X$="n" OR X$="N" THEN PRINT "No.":DEMOFLAG=Ø:GOTO 2760
2730  IF X$="y" OR X$="Y" THEN PRINT "Yes.":DEMOFLAG=1:GOTO 2760
2740 GOTO 2710
2750 :
2760 LOCATE 1Ø,1,1
2770 PRINT TAB(2Ø) "Ambient temperature (F.)    :  ";
2780 INPUT; "", TEMPERATURE
2790 PRINT " degrees F."
2800 PRINT TAB(2Ø) "Barometric pressure (in. Hg):  ";
2810 INPUT; "", PRESSURE
282Ø PRINT " inches Hg"
2830 :
2840 IF DEMOFLAG=Ø THEN FILENAME$="B:SCAN5ØØ.LOG":GOTO 291Ø
2850 FILENAME$="B:SCANDEMO.LOG"
2860 :
2870 REM  ********************************************************************
2880 REM  PRINT THE PROGRAM MAIN MENU
2890 REM  ********************************************************************
2900 :
2910 GOSUB 713Ø                          :REM  Print the program main menu
```

```
2920 LOCATE 2,4,Ø
2930 PRINT 'SCAN500 Data Acquisition Driver Menu'
2940 :
2950 LOCATE 5,1,Ø
2960 PRINT TAB(1Ø) "Given the following configurations:":PRINT
2970 PRINT TAB(2Ø) 1      Center row only " HSSHSS " 32 electrodes'
2980 PRINT TAB(2Ø) "3     All 3 rows " HSSHSS " 96 electrodes'
2990 LOCATE 1Ø,1,Ø
3000 PRINT TAB(1Ø) 'Please enter your selection:  ";
3010 LOCATE ,,1
3020 X$=INKEY$
3030  IF X$="1" THEN PRINT 'Center 32 electrodes only.";:GOTO 3070
3040  IF X$="3" THEN PRINT 'All 96 electrodes.";:GOTO 3080
3050 GOTO 3020
3060 :
3070 ROWS=1:ELECTRODES=32:GOTO 3100     :REM  Set variables for 32 or 96
3080 ROWS=3:ELECTRODES=96               :REM    electrodes
3090 :
3100 LOCATE 12,1,1
3110 PRINT TAB(1Ø) 'Enter the tlme limit for the run in minutes:  ";
3120 INPUT: "", TIMELIMIT               :REM  User tnput value for tfme limit
3130 IF TIMELIMIT>Ø GOTO 3160
3140 TIMELIMIT=1Ø                       :REM  Default value for tlme limit
3150 PRINT "1Ø"
316Ø TIMELIMIT=TIMELIMIT*6Ø             rREM  Convert ttme ltmlt to seconds
3170 :
318Ø LOCATE 14,1,1
319Ø PRINT TAB(1Ø) "Enter the time delay between datasets in seconds:  ";
32ØØ CPOS=POS(X)
3210 INPUT: "",DELAY                    :REM  User input value for tfme delay
3220 IF DELAY>Ø GOTO 3270
3230 LOCATE ,CPOS,Ø                     :REM  Default values for tfme delay
3240 DELAY=Ø:PRINT 'No delay.'  TAB(8Ø):GOTO 3270
3250 DELAY=2:PRINT '2'
3260 :
3270 LOCATE 16,1,1                      :REM  Toggle graphtc dlsplay of data
3280 PRINT TAB(1Ø) 'Do you wish the results displayed graphically?  ";
3290 X$=INKEY$
3300  IF X$="y" OR X$="Y" THEN DISPLAYFLAG=1:PRINT "Yes.":GOTO 3340
3310  IF X$="n" OR X$="N" THEN DISPLAYFLAG=Ø:PRINT "No.":PRINT:GOTO 3550
3320 GOTO 3290
3330 :
3340 ROW=1:COLUMN=7Ø                    :REM  Coordlnates for ttme dlsplay
3350 :
3360 PRINT TAB(1Ø) 'Enter the frequency for graphing the results:  ";
3370 INPUT; "",X                        :REM  User lnput value for frequency
3380 FREQUENCY=INT(X)                   rREM  Convert frequency to an integer
3390 IF FREQUENCY>Ø THEN PRINT:PRINT:GOTO 3570
3400                                    :REM  Calculate default value for
3410 IF DELAY<5 GOTO 3440               :REM    frequency of graphic dfsplay
3420 FREQUENCY-1                        :REM    so that the screen is refreshed
3430 PRINT 'Graph all datasets.":PRINT:GOTO 3570
3440 IF ELECTRODES=96 GOTO 3480         :REM    at least once each five seconds
3450 IF DELAY+1.556>5 GOTO 3420
3460 FREQUENCY=INT(5/(DELAY+1.556))
3470 PRINT FREQUENCY:PRINT:GOTO 357Ø
3480 IF DELAY+4.368>5 GOTO 342Ø
3490 FREQUENCY=INT(5/(DELAY+4.368)):GOTO 347Ø
3500 :
3510 REM  ****************************************************************
3520 REM  PERFORM PRELIMINARY SETUP
3530 REM  ****************************************************************
3540 :
3550 ROW=8:COLUMN=35                    rREM  Coordlnates for tlme dlsplay
```

```
3560 :
3570 ON ERROR 60TO 9210                    rREM  Set up error trapping
3580 :
3590 ON KEY(1) GOSUB 7500                  :REM  Set up F1 to halt executton
3600 KEY(1) ON                             :REM  Acttvate F1 key capture
3610 KEY(1) STOP                           :REM  Retaln F1 for next KEY(1) ON
3620 ON KEY(2) GOSUB 9450                  rREM  Set up F2 to change dtsplay
3630 KEY(2) ON                             :REM  Activate F2 key capture
3640 KEY(2) STOP                           :REM  Retaln F2 for next KEY(2) ON
3650
3660 OPEN "C:SCAN5ØØ.DAT" FOR OUTPUT AS #1
3670                                       :REM  Open Insta-Drive for data output
3680 :
3690 REM *******************************************************************
3700 REM  SET THE INTERVAL TIMER
3710 REM *******************************************************************
3720 :
3730 DEF SEG=&HAFFØ                        :REM  Set current segment address
3740 CO=5ØØØØ!                             :REM  Set pulses for counter one
3750 CZ=5ØØØØ!                             :REM  Set pulses for counter zero
3760 :
3770 HZ=INT(CZ/256)                        :REM  High byte counter zero
378Ø LZ=CZ-(256*HZ)                        :REM  Low byte counter zero
3790 HO=INT(CO/256)                        :REM  HIgh byte counter one
3800 LO=CO-(256*HO)                        :REM  Low byte counter one
381Ø :
3820 POKE &HEØ,1                           :REM  Timer global command
3830
384Ø POKE &HC3,54                          :REM  Counter zero control (ttmer)
3850 POKE &HCØ,LZ                          :REM  Write low byte
3860 POKE &HCØ,HZ                          :REM  Write high byte
3870 .
388Ø POKE &HC3,116                         :REM  Counter one control (carry)
389Ø POKE &HC1,LO                          :REM  Write low byte
3908 POKE &HC1,HO                          :REM  Wrfte high byte
3910
3920 REM *******************************************************************
3930 REM  INITIALIZE THE PROGRAM FOR MENU-SELECTED HARDWARE CONFIGURATION
3940 REM *******************************************************************
3950
3960 OPTION BASE 1                         :REM  Set subscrtpt low value at 1
3970 DIM ADDRESS(6)
3980
3990 IF ROWS=1 THEN FIRST=5:LAST=5:GOTO 4020
4000 FIRST=4:LAST=6
4010
4020 ADDRESS(4)=&H86                       :REM  Set CMDA address for slot 4
4030 ADDRESS(5)=&H88                       :REM  Set CMDA address for slot 5
4040 ADDRESS(6)=&H8A                       :REM  Set CMDA address for slot 6
4050
4060 REM *******************************************************************
4070 REM  WAIT FOR START COMMAND TO BEGIN DATA ACQUISITION
4088 REM *******************************************************************
4090 :
4100 PRINT TAB(1Ø) "The software is ready for data acquisition."
4110 PRINT TAB(1Ø) "Functton key FI terminates program execution.'
4120 IF DISPLAYFLAG=Ø THEN PRINT:GOTO 4150
4130 PRINT TAB(1Ø) "Functton key F2 allows the graphic display frequency ";
4140 PRINT "to be changed.":PRINT
4150 LOCATE ,,1
4160 PRINT TAB(1Ø) "Press the RETURN key to begin or " CHR$(34) "X" CHR$(34);
4170 PRINT " to restart.  ";
4180 FOR I=1 TO 16:SOUND RND*1ØØØ+37,2:NEXT I
4190 X$=INKEY$
```

```
4200  IF X$=CHR$(13) THEN CLS:GOSUB 7040:GOTO 4280
4210  IF X$="x" OR X$="X" THEN CLOSE:CLEAR:GOSUB 7040:GOTO 2620
4220 GOTO 4190
4230 :
4240 REM  *******************************************************************
4250 REM  SET UP SCREEN ACCORDING TO GRAPHIC OUTPUT TOGGLE
4260 REM  *******************************************************************
4270 :
4280 DIM Y(100),YOLD(100),DVOLTS(100),DHIGH(100),DLOW(100)
4290 STARTTIME$=TIME$                       :REM  Index start tlme of run
4300 IF DISPLAYFLAG=1 GOTO 4400             :REM  Jump ahead if dlsplay desired
4310 :
4320 GOSUB 7130                             rREM  Prlnt startup screen header
4330 LOCATE 2,4,0
4340 PRINT 'SCAN500  Data Acquisition Active ..."
4350 :
4360 TIMEOUTFLAG=0                          :REM  Toggle start of run
4370 GOSUB 7280                             :REM  Prtnt startup status screen
4380 GOTO 4610                              :REM  Junp ahead since no dtsplay
4390 :
4400 SCREEN 2
4410 :
4420 IF ROWS=1 THEN START=129:FINISH=129:REM  Set up for 32 electrodes
4430 IF ROWS=3 THEN START=59:FINISH=199 :REM  Set up for 96 electrodes
4440 :
4450 FOR YBASE=START TO FINISH STEP 70  rREM  Loop to draw box(es)
4460  LINE (0,YBASE-59)-(0,YBASE)       :REM  )
4470  LINE (1,YBASE-59)-(1,YBASE)       rREM  )
4480  LINE (0,YBASE)-(639,YBASE)        :REM  )-Draw box for one row
4490  LINE (639,YBASE)-(639,YBASE-59)   :REM  )   of 32 electrodes
4500  LINE (638,YBASE)-(638,YBASE-59)   :REM  )
4510  LINE (639,YBASE-59)-(0,YBASE-59)  :REM  )
4520 NEXT YBASE                         :REM  Loop to draw next box
4530 :
4540 LOCATE 1,2:PRINT " "+DATE$+" "     :REM  Prlnt date in ULH corner
4550 IF F2FLAG=1 GOTO 5870
4560 :
4570 REM  *******************************************************************
4580 REM  BEGIN DATA ACQUISITION
4590 REM  *******************************************************************
4600 :
4610 DATASETS=0                         :REM  Initialize dataset counter
4620 COUNTER=0                          :REM  Initialize graphic dlsplay counter
4630 :
4640 HAP=11                             :REM  High byte positive applied voltage
4650 LAP=51                             :REM  Low byte positive applied voltage
4660 HAN=4                              :REM  High byte negative applied voltage
4670 LAN=205                            :REM  Low byte negative applied voltage
4680 :
4690 :
4700 POKE &H9A,3                        :REM  Set global galn (X 10)
4710 POKE &H9D,64                       rREM  Strobe enable command
4720 :
4730 PRINT #1, USING '##";ELECTRODES    rREM  Write number of electrodes to disk
4740 :
4750 STARTFLAG=1                        :REM  Allow Indexlng start ttme of run
4760 :
4770 FOR CYCLE=1 TO 2                   :REM  Alternate + and - applied voltages
4780 :
4790  IF CYCLE=2 GOTO 4880
4800 :
4810  DATASETS=DATASETS+1               rREM  Increment the dataset counter
4820  COUNTER=COUNTER+1                 :REM  Increment graphic display counter
4830  POKE &H82,0                       :REM  D/A control CMDA low byte
```

```
4840  POKE &H83,LAP                       :REM  Wrlte CMDB positive low byte
4850  POKE &H82,1                         :REM  D/A control CMDA hfgh byte
4860  POKE &H83,HAP:GOTO 4930             :REM  Write CMDB posftfve high byte
4870 :
4880  POKE &H82,Ø                         rREM  D/A control CMDA low byte
4890  POKE &H83,LAN                       :REM  Write CMDB negative low byte
4900  POKE &H82,1                         :REM  D/A control CMDA high byte
4910  POKE &H83,HAN                       :REM  Write CMDB negatfve high byte
4920
4930 POKE &H9D,1                          rREM  Strobe send data
4940 GOSUB 8970                          rREM  Read time
4950 IF STARTFLAG=Ø GOTO 4980
4960 OLDTIME=TIME                         :REM  Calculate start tfme of run
4970 STARTFLAG=Ø                          :REM    then set STARTFLAG to zero
4980 STARTTIME=TIME-OLDTIME               :REM  Calculate start time of dataset
499Ø .
5000 FOR SLOT=FIRST TO LAST               :REM  Start slot selectfon loop
5010  FOR CHANNEL=1 TO 32                 :REM  Start channel selectfon loop
5020   I=(SLOT-FIRST)*32+CHANNEL          :REM  Calculate fndex for array element
5030   POKE ADDRESS(SLOT),CHANNEL-1       :REM  Select channel (Ø-31)
5040   POKE &H81,SLOT                     :REM  Select slot
5050   POKE &H9B,255                      :REM  Start A/D conversfon
5060   IF PEEK(&H9B)<>127 GOTO 5068       :REM  Loop ff A/D converslon incomplete
5070   DLOW(I)=PEEK(&H8Ø)                 :REM  Read low byte
5080   DHIGH(I)=PEEK(&H81)                :REM  Read high byte
5090  NEXT CHANNEL                        :REM  Loop for next channel
5100 NEXT SLOT                            :REM  Loop for next slot
5110 :
5120 IF CYCLE-2 GOTO 5900
5130 GOSUB 8970                          :REM  Read the current tfme
5140 ENDTIME=TIME-OLDTIME                 :REM  Calculate stop time of dataset
5150 ETIME=ENDTIME-STARTTIME              :REM  Calculate elapsed time of dataset
5160
517Ø LOCATE ROW,COLUMN:PRINT " "+TIMES$+" ";
5180                                      :REM  Prtnt real time on the screen
5190 :
5200 PRINT #1, USING "###.###";STARTTIME
5210                                      :REM  Write ET to start of dataset
5220 PRINT #1, USING "#.###";ETIME        rREM  Write elapsed time of dataset
5230 :
5240 IF DEMOFLAG-0 GOTO 5280              :REM  Jump ahead if not a demonstration
5250 N=INT(VAL(MID$(TIME$,7,2)))          :REM  Read seconds from TIMES
5260 RANDOMIZE N                          :REM  Seed random number gemerator
5270 :
5280 FOR SLOT=FIRST TO LAST               :REM  Write high and low bytes for each
5290  FOR CHANNEL=1 TO 32                 :REM    electrode from the last dataset
5300   I=(SLOT-FIRST)*32+CHANNEL          :REM  Index the datum
5310   IF DEMOFLAG=Ø GOTO 5360            :REM  Jump ahead if not a demonstratfon
5320   STEPS=INT(RND*4Ø96/2Ø)+2Ø48        :REM  ***** DUMMY DEMO DATUM *****
5330   DUMMY=INT(STEPS/256)               :REM  ***** DUMMY DEMO DATUM *****
5340   DHIGH(I)=DUMMY+24Ø                 :REM  ***** DUMMY DEMO DATUM *****
5350   DLOW(I)=STEPS-(DUMMY*256)          :REM  ***** DUMMY DEMO DATUM *****
5360   PRINT #1, USING "### ###";DLOW(I),DHIGH(I)
5370  NEXT CHANNEL
5380 NEXT SLOT
5390 :
5400 TOTALTIME=ENDTIME                    :REM  [redacted] elapsed time of run
5410 :
5420 IF DISPLAYFLAG=Ø GOTO 5730           :REM  Jump ahead ff no display desired
5430 IF COUNTER=FREQUENCY THEN COUNTER=Ø:GOTO 5470
5440                                      :REM  Graph results at FREQUENCY
5450 GOTO 57ØØ                            :REM  Jump ahead; no dfsplay required
5460 :
5470 FOR YBASE=START TO FINISH STEP 70    :REM  Loop to draw graph(s) in box(es)
```

```
5480  LINE (Ø,YBASE)-(Ø,YBASE),Ø          :REM  Locate origin of box
5490  FOR XAXIS=9 TO 639 STEP 20          :REM  Loop for each electrode
5500   I=INT((YBASE-START)/7Ø)*32+INT(XAXIS/2Ø)+1
5510                                      :REM  Calculate index for array element
5520   DRES=DLOW(I)+(DHIGH(I)-24Ø)*256    :REM  Reconstitute datum
5530   DVOLTS=ABS(DRES*(2Ø/4Ø95)-1Ø)      :REM  Convert to volts
5540   IF DVOLTS>1 THEN DVOLTS=.99        :REM  Display a maximum value only
5550   Y(I)=YBASE-INT(DVOLTS*58)-1        :REM  Locate Y within the box
5560   IF I=1 OR I=33 OR I=65 GOTO 5630   :REM  No segment erasure for first point
5570   IF DATASETS=1 GOTO 5610            :REM  No graph erasure for first graph
5580   IF DATASETS<=FREQUENCY GOTO 5610
5590   LINE (XAXIS-2Ø,YOLD(I-1))-(XAXIS,YOLD(I)),Ø
5600                                      :REM  Erase segment from previous graph
5610   LINE (XAXIS-2Ø,Y(I-1))-(XAXIS,Y(I))
5620                                      :REM  Draw a line from the last point
5630   LOCATE 1,71:PRINT TIMES            :REM  Print time in URH corner
5640   IF I=1 OR I=33 OR I=65 GOTO 5660   :REM  No erasure needed for first point
5650   YOLD(I-1)=Y(I-1)                   :REM  Store data for erasing old graph
5660  NEXT XAXIS                          :REM  Loop to next electrode
5670  YOLD(I)=Y(I)                        :REM  Store last datum for erasing
5680 NEXT YBASE                           :REM  Loop to draw next graph
5690 :
5700 LOCATE 25,2,Ø:PRINT DATASETS;        :REM  Print graph number in LLH corner
5710 LOCATE 25,71,Ø:PRINT USING " ###.### ";TOTALTIME;:GOTO 5780
5720                                      :REM  Print elapsed time in LRH corner
5730 LOCATE 8,36,Ø:PRINT TIMES            :REM  Print status update on the screen
5740 LOCATE 9,36,Ø:PRINT USING "###.###";TOTALTIME
5750 LOCATE 1Ø,36,Ø:PRINT USING "###.###";ETIME
5760 LOCATE 11,36,Ø:PRINT USING "####";DATASETS
5770 :
5780 IF TOTALTIME>TIMELIMIT THEN TIMEOUTFLAG=1:GOSUB 7500
5790                                      :REM  Stop if time limit is exceeded
5800 :
5810 KEY(1) ON                            :REM  Check for program termination
5820 KEY(1) STOP                          :REM  Retain F1 for next KEY(1) ON
5830 IF DISPLAYFLAG=Ø GOTO 5870
5840 KEY(2) ON                            :REM  Check for change of display
5850 KEY(2) STOP                          :REM  Retain F2 for next KEY(2) ON
5860 :
5870 GOSUB 8970                           :REM  Read the current time
5880 IDLETIME=TIME-OLDTIME-ENDTIME        :REM  Calculate time of inactivity
5890 IF IDLETIME>DELAY GOTO 5940          :REM  Implement time delay
5900 LOCATE ROW,COLUMN:PRINT " "+TIMES+" "
5910 IF CYCLE=2 GOTO 5940                 :REM  Print real time on the screen
5920 GOTO 5870                            :REM  Loop back to check time delay
5930 :
5940 NEXT CYCLE                           :REM  Reverse applied voltage polarity
5950 GOTO 4770                            :REM  Return for another dataset
5960 :
5970 END                                  :REM  Program end
5980 :
7ØØØ REM  ****************************************************************
7Ø1Ø REM  SUBROUTINE:  Read Graphic Display Characters
7Ø2Ø REM  ****************************************************************
7Ø3Ø :
7Ø4Ø WIDTH 8Ø:SCREEN Ø,Ø,Ø:LOCATE 1,1,Ø:CLS:KEY OFF
7Ø5Ø UL$=CHR$(2Ø1):UR$=CHR$(187):LR$=CHR$(188):LL$=CHR$(2ØØ)
7Ø6Ø H$=CHR$(2Ø5):V$=CHR$(186):HS$=CHR$(196)
7Ø7Ø RETURN
7Ø8Ø :
7Ø9Ø REM  ****************************************************************
71ØØ REM  SUBROUTINE:  Print The Menu Header Box
711Ø REM  ****************************************************************
712Ø :
```

```
7130 WIDTH 8Ø:SCREEN Ø,Ø,Ø:LOCATE 1,1,Ø:CLS:KEY OFF
7140 PRINT UL$H$H$H$H$H$H$H$H$H$H$H$H$H$H$H$H$H$H$H$H$H$H$H$H$H$H$H$H$H$H$H$H$H$;
7150 PRINT H$H$H$H$H$H$H$H$H$H$H$H$H$H$H$H$H$H$H$H$H$H$H$H$H$H$H$H$H$H$H$H$H$H$;
7160 PRINT H$H$H$H$H$H$H$H$H$H$H$H$H$H$H$H$H$H$H$H$UR$V$ TAB(8Ø) V$;
7170 PRINT LL$H$H$H$H$H$H$H$H$H$H$H$H$H$H$H$H$H$H$H$H$H$H$H$H$H$H$H$H$H$H$H$H$H$;
7180 PRINT H$H$H$H$H$H$H$H$H$H$H$H$H$H$H$H$H$H$H$H$H$H$H$H$H$H$H$H$H$H$H$H$H$H$;
7190 PRINT H$H$H$H$H$H$H$H$H$H$H$H$H$H$H$H$H$H$H$H$LR$
7200 .
7210 RETURN
7220 :
7230 REM ******************************************************************
7240 REM  SUBROUTINE:  Set up Status Screen
7250 REM ******************************************************************
7260
7270 IF DISPLAYFLAG=Ø AND TIMEOUTFLAG=1 THEN LOCATE 8,1,Ø:GOTO 7320
7280 LOCATE 6,1,Ø                          :REM  Print status screen variables
7290 PRINT TAB(1Ø) 'Date of run          :  ":DATES
7300 PRINT TAB(1Ø) "Start time of run     :  ";STARTTIME$
7310 IF TIMEOUTFLAG=Ø GOTO 7330
7320 PRINT TAB(1Ø) "End time of run       :  ";ENDTIME$:GOTO 7340
7330 PRINT TAB(1Ø) "Current time          :  ";TIMES
7340 IF DISPLAY=Ø AND TIMEOUTFLAG=1 THEN RETURN
7350 PRINT TAB(1Ø) "Elapsed time of run   .  ",
7360 PRINT USING "###.###";TIME;
7370 PRINT " seconds"
7380 PRINT TAB(1Ø) "Elapsed time of dataset:  ";
7390 PRINT USING "###.###";ETIME;
7400 PRINT " seconds"
7410 PRINT TAB(1Ø) "Number of datasets    :  ";
7420 PRINT USING "###";DATASETS
7430 :
7440 RETURN
7450 :
746Ø REM ******************************************************************
7470 REM  SUBROUTINE:  Execute Functton Key F1
748Ø REM ******************************************************************
7490 :
7500 CLOSE #1                              :REM  Close the output file
7510 ENDTIME$=TIME$                        :REM  Index the end time of the run
7520 IF DISPLAYFLAG=Ø GOTO 7570            :REM  Continue if no display desired
7530 GOSUB 7130                            :REM  Clear graphics screen and print
7540 LOCATE 2,4,Ø                          :REM    status header for shutdown
7550 PRINT "SCAN500 Data Acquisition Terminated ...":GOTO 7600
7560 :
7570 LOCATE 2.29
7580 PRINT "Terminated ..."
7590 :
7600 GOSUB 7270                            :REM  Print shutdown status screen
7610 :
7620 LOCATE 13.1.1                         :REM  PrInt shutdown message
7630 FOR I=1 TO 16:SOUND RND*1ØØØ+37,2:NEXT I
7640 PRINT "Proqram execution has been halted."
7650 PRINT "Elapsed time-of the run is ·,,
7660 IF TOTALTIME>1ØØ THEN F$="###,###" ELSE F$="##,###"
7670 IF TOTALTIME<1Ø THEN F$="#,###"
7680 PRINT USING FS; TOTALTIME:
7690 PRINT " seconds after";DATASETS;
7700 IF DATASETS>1 THEN PRINT "datasets.":GOTO 7730
7710 PRINT "dataset."
7720 :
7730 LOCATE 16.1.1                         :REM  Request to continue program
7740 PRINT "Do you wtsh to organize the data at this tlme?  ";
7750 X$=INKEY$
7760  IF X$="y" OR X$="Y" THEN PRINT "Yes.":QUITFLAG=Ø:GOTO 7800
```

```
7770  IF X$="n" OR X$="N" THEN PRINT "No.":QUITFLAG=1:GOTO 7800
7780 GOTO 7750
7790 :
7800 LOCATE 18,1,1                          :REM  Request floppy disks for backup
7810 PRINT "Place  a formatted disk  in drive A  and the  .LOG disk in drive B"
7820 PRINT "     then press RETURN.    "; :REM    of  log and header information
7830 X$=INKEY$                              :REM    in drive B: even  if raw data
7840  IF X$=CHR$(13) GOTO 7880              :REM    organization is  not performed
7850  IF X$="x" OR X$="X" THEN DEMOFLAG=2:GOTO 8140
7860 GOTO 7830                              :REM    at this time.
7870 :
7880 LOCATE 21,1,0
7890 PRINT "Working ...  ";
7900 :
7910 OPEN "C:SCAN500.DAT" FOR INPUT AS #1
7920 OPEN "A:SCAN500.DAT" FOR OUTPUT AS #2
7930 :
7940 INPUT #1, A
7950 PRINT #2, USING "##"; A                :REM  Copy number of electrodes
7960 :
7970 IF EOF(1) GOTO 8080
7980 INPUT #1, B, C
7990 PRINT #2, USING "###.###"; B           :REM  Copy ET to start of dataset
8000 PRINT #2, USING "#.###"; C             :REM  Copy ET of dataset
8010 :
8020 FOR I=1 TO ELECTRODES
8030  INPUT #1, D, E
8040  PRINT #2, USING "### ###"; D, E  :REM  Copy raw data for each electrode
8050 NEXT I
8060 GOTO 7970
8070 :
8080 CLOSE #1:CLOSE #2
8090 :
8100 LOCATE 18,1,0                          :REM  Print backup completion message
8110 FOR I=1 TO 8:SOUND RND*1000+37,2:NEXT I
8120 PRINT "Data backup has been written to drive A"  TAB(80)
8130 :
8140 LOCATE 18,1,0:PRINT TAB(80)
8150 LOCATE 19,1,0:PRINT TAB(80)
8160 LOCATE 21,1,0:PRINT TAB(80):IF DEMOFLAG=2 GOTO 8350
8170 :
8180 LOCATE 20,1,0                          :REM  Add information on current run to
8190 PRINT "Working ...  ";                 :REM    permanent log
8200 OPEN FILENAME$ FOR INPUT AS #1
8210 OPEN "B:COPY.TMP" FOR OUTPUT AS #2
8220 :
8230 A=0
8240 :
8250 IF EOF(1) GOTO 8330                    :REM  Copy permanent log to temporary
8260 INPUT #1,A,B,A$,B$,C$,C,D,E,F,G        :REM    file
8270 A$=CHR$(34)+A$+CHR$(34)
8280 B$=CHR$(34)+B$+CHR$(34)
8290 C$=CHR$(34)+C$+CHR$(34)
8300 PRINT #2, A;B;A$;B$;C$;C;D;E;F;G
8310 GOTO 8250
8320 :
8330 CLOSE #1
8340 :
8350 A=A+1
8360 A$=CHR$(34)+DATE$+CHR$(34)
8370 B$=CHR$(34)+STARTTIME$+CHR$(34)
8380 C$=CHR$(34)+ENDTIME$+CHR$(34)
8390 B=DEMOFLAG
8400 C=TOTALTIME
```

```
8410 D=ELECTRODES
8420 E=DATASETS
8430 F=TEMPERATURE
8440 G=PRESSURE:IF DEMOFLAG=2 GOTO 8660
8450 :
8460 OPEN "B:SCAN5ØØ.HDR" FOR OUTPUT AS #1
8470 OPEN "A:SCAN5ØØ.HDR" FOR OUTPUT AS #3
8480 :
8490 PRINT #1, A;B;A$;B$;C$;C;D;E;F;G   :REM  Write header for current run
8500 PRINT #2, A;B;A$;B$;C$;C;D;E;F;G   :REM  Add current run to permanent log
8510 PRINT #3, A;B;A$;B$;C$;C;D;E;F;G   :REM  Wrlte header to backup dlsk
8520 :
8530 CLOSE #1:CLOSE #2:CLOSE #3
8540 KILL FILENAME$
8550 NAME "B:COPY.TMP" AS FILENAMES
8560 :
8570 LOCATE 19,1,Ø                       :REM  Print disk file completion message
8580 PRINT 'Header backup and .LOG files have been wrltten to drlves A and B.
8590 :
8600 LOCATE 2Ø,1,1                       :REM  Request to continue program
8610 PRINT 'Place the SCAN500 system disk in drive A then press RETURN.  ";
8620 X$=INKEY$
8630  IF X$=CHR$(13) GOTO 8660
8640 GOTO 8620
8650 :
8660 LOCATE 22,1,Ø
8670 PRINT "Working ...  ";
8680 :
8690 OPEN "A:SCAN5ØØ.HDR" FOR OUTPUT AS #1
8700 PRINT #1, A;B;A$;B$;C$;C;D;E;F;G
8710 CLOSE #1
8720 :
8730 LOCATE 2Ø,1,Ø
874% PRINT 'The run header flle has been written to drtve A." TAB(8Ø)
8750 LOCATE 21,1,Ø
8760 PRINT TAB(8Ø)
8770 :
8780 FOR I=1 TO 4:SOUND RND*1ØØØ+37,2:NEXT I
8790 IF QUITFLAG=Ø GOTO 8850            :REM  End at thls time if quit selected
8800 :
8810 LOCATE 22,1,Ø                       :REM  Chaln back to system menu
8820 PRINT "Program termlnated ..."
8830 FOR I=1 TO 4ØØØ:NEXT I:CHAIN "HEAD5ØØ.EXE"
8840 :
8856 LOCATE 22,1,Ø                       :REM  Chain next program if selected
8860 PRINT "Loading FIX500 data organizer ...";
8870 FOR I=1 TO 4ØØØ:NEXT I:CHAIN "FIX5ØØ.EXE"
8880 :
8890 RETURN
8900 :
891Ø REM ******************************************************************
8926 REM  SUBROUTINE:  Read the Interval Tlmer
893Ø REM ******************************************************************
8940 :
8950 REM  Latch the Interval Counters
8960 :
8970 DEF SEG=&HAFFØ                      :REM  Set current segment address
8980 :
8990 POKE &HC3,Ø                         :REM  Latches counter zero
9000 LZ=PEEK(&HCØ)                       :REM  Reads low byte
9010 HZ=PEEK(&HCØ)                       :REM  Reads hfgh byte
9020 CCZ=LZ+(HZ*256)                     :REM  Reconstitutes counter zero
9030 :
9040 POKE &HC3,64                        :REM  Latches counter one
```

```
9050 LO=PEEK(&HC1)                          :REM  Reads low byte
9060 HO=PEEK(&HC1)                          :REM  Reads high byte
9070 CCO=LO+(HO*256)                        :REM  Reconstitutes counter one
9080 :
9090 REM  Convert Current Count to Elapsed Ttme
9100 :
9110 CURRENTCOUNT#=(((CZ-CCZ)*CO)+(CO-CCO))/2
9120                                        :REM  Current count in clock pulses
9130 TIME=CURRENTCOUNT#*.ØØØØØ1Ø46#         :REM  Elapsed tlme in seconds
9140 :
9150 RETURN
9160 :
917Ø REM  *****************************************************************
9180 REM  SUBROUTINE:  Error Trapplng
919Ø REM  *****************************************************************
9200 :
9210 IF ERR<>61 GOTO 9270                  :REM  Prtnt 'Disk full' message
9220 LOCATE 14,1,Ø
9230 FOR I=1 TO 4:SOUND RND*1ØØØ+37,2:NEXT I
9240 PRINT "Data acquisition has stopped.  ";
9250 PRINT 'Internal drive C: is full.'
9260 :
9270 IF ERR<>53 GOTO 9310                  :REM  Open .LOG ftle if nonexistent
9280 OPEN FILENAMES FOR OUTPUT AS #1
9290 CLOSE #1:RESUME                       :REM  Resume writing to .LOG file
9300 :
9310 LOCATE 2Ø,1,Ø
9320 NERROR-ERR
9330 FOR I=1 TO 4:SOUND RND*1ØØØ+37,2:NEXT I
9340 PRINT "Program execution has been halted because of runtime error ";
9350 PRINT USING "##";NERROR;
9360 PRINT "."
9370 :
9380 FOR I=1 TO 4ØØØ:NEXT I:CHAIN "HEAD5ØØ.EXE"
9390 :
9400 END
941Ø REM  *****************************************************************
9420 REM  SUBROUTINE:  Execute Function Key F2
943Ø REM  *****************************************************************
944Ø :
9450 WIDTH 8Ø:SCREEN 0.0.0
9460 GOSUB 7130
9470 LOCATE 2,4,Ø
9480 PRINT "Modify Graphic Dlsplay Frequency'
9490 :
9500 LOCATE 17.1.1
9510 PRINT TAB(1Ø) 'Enter the frequency for graphfng the results:  ";
9520 INPUT; "",X
953Ø FREQUENCY=INT(X)
9540 F2FLAG=1:GOTO 4400
9550 :
9560 RETURN
```

```
1000 REM   ******************************************************************
1010 REM   *  DATA FILE GENERATER FROM SCAN500-GENERATED OUT500.DAT FILE    *
1020 REM   ******************************************************************
1030 REM   *                                                                *
1040 REM   *                    Program name:   FIX500                       *
1050 REM   *                                                                *
1060 REM   *                          Written for                           *
1070 REM   *       Stanford University Petroleum Engineering Department     *
1080 REM   *             by Lawrence W. Bouett, July 29, 1985               *
1090 REM   *                                                                *
1100 REM   ******************************************************************
1110 :
1120 :
1130 REM  *******************************************************************
1140 REM  PRINT THE PROGRAM HEADER
1150 REM  *******************************************************************
1160 :
1170 WIDTH 80:SCREEN 0,0,0:LOCATE 1,1,0:CLS:KEY OFF
1180 UL$=CHR$(201):UR$=CHR$(187):LR$=CHR$(188):LL$=CHR$(200)
1190 H$=CHR$(205):V$=CHR$(186):HS$=CHR$(196)
1200
1210 PRINT UL$H$H$H$H$H$H$H$H$H$H$H$H$H$H$H$H$H$H$H$H$H$H$H$H$H$H$H$H$H$;
1220 PRINT H$H$H$H$H$H$H$H$H$H$H$H$H$H$H$H$H$H$H$H$H$H$H$H$H$H$H$H$H$H$;
1230 PRINT H$H$H$H$H$H$H$H$H$H$H$H$H$H$H$H$H$H$UR$;
1240 PRINT V$ "  FIX500 Data Organization Program" TAB(80) V$;
1250 PRINT LL$H$H$H$H$H$H$H$H$H$H$H$H$H$H$H$H$H$H$H$H$H$H$H$H$H$H$H$H$H$;
1260 PRINT H$H$H$H$H$H$H$H$H$H$H$H$H$H$H$H$H$H$H$H$H$H$H$H$H$H$H$H$H$H$;
1270 PRINT H$H$H$H$H$H$H$H$H$H$H$H$H$H$H$H$H$H$LR$
1280 :
1290 ON ERROR GOTO 3420                   :REM  Set up error trapping
1300 ON KEY(1) GOSUB 3830                 :REM  Set up function key F1
1310 KEY(1) ON                            :REM    to terminate program
1320 KEY(1) STOP                          :REM    execution
1330
1340 LOCATE 6,1,0,0,7                     :REM  Print startup message to screen
1350 PRINT TAB(10) "This program organizes the datafile SCAN500.DAT created"
1360 PRINT TAB(10) "by SCAN500 into a useable format for graphic display or"
1370 PRINT TAB(10) "curve-fitting.'
1380 :
1390 LOCATE 10,1,0
1400 PRINT TAB(10) "Enter the drlve designation for the output file:  ";
1410 LINE INPUT;OUTDRIVE$
1420 LOCATE 10,1,0
1430 PRINT TAB(10) "Enter the drlve designation for the temporary files:  ";
1440 LINE INPUT;TEMPDRIVE$
1450 LOCATE 10,1,0:PRINT TAB(80):LOCATE 10.1,1
1460 IF OUTDRIVE$=TEMPDRIVE$ GOTO 1490
1470 PRINT TAB(10) "Place formatted dfsks in drives " OUTDRIVES " and ";
1480 PRINT TEMPDRIVE$;:GOTO 1500
1430 PRINT TAB(10) "Place a formatted disk is drlve " OUTDRIVES;
1500 PRINT " then press RETURN.  ";
1510 :
1520 X$= INKEYS
1530  IF X$=CHR$(13) THEN LOCATE 10,10,0:PRINT TAB(80):GOTO 1560
1540 GOTO 1510
1550
1560 FILEFLAG=1
1570 DRIVE$="A"
1580 OPEN DRIVE$+":SCAN500.DAT" FOR INPUT AS #1
1590 CLOSE #1
1600 FILENAME$=DRIVE$+":SCAN500.DAT":DATDRIVE$=DRIVE$
1610 :
1620 LOCATE 10,10,0                       :REM  Confirm location of file
1630 PRINT "SCAN500.DAT is resident in drive " DRIVES "." TAB(80)
```

```
1640 :
1650 FILEFLAG=2
1660 DRIVE$="A"
1670 OPEN DRIVE$+":SCAN500.HDR" FOR INPUT AS #1
1680 INPUT #1, A,B,A$,B$,C$,C,D,E,F,G    :REM  Read header to establish variables
1690 CLOSE #1:HDRDRIVE$=DRIVE$
1700 :
1710 ELECTRODES=D
1720 DATASETS=E
1730 :
1740 DIM DVOLTS(96,120),TIME(1000)
1750 PASSES=FIX((DATASETS-1)/120)+1
1760 :
1770 LOCATE 12,1,0                          :REM  Set UP program status on screen
1780 PRINT TAB(20) "Start time of program run:  ";TIME$
1790 PRINT TAB(20) "Current time              :  ";TIME$
1800 :
1810 LOCATE 15,5,0
1820 PRINT "This is pass";:POSPASS=POS(X):PRINT " 1 out of a total of" PASSES;
1830 IF PASSES=1 THEN PRINT "pass."  ELSE PRINT "passes."
1840 :
1850 LOCATE 17,5,0
1860 PRINT "Input datasets total    :  " USING "###"; DATASETS
1870 :
1880 LOCATE 18,5,0
1890 PRINT "Input datasets processed:"
1900 :
1910 LOCATE 17,40,0
1920 PRINT "Output datasets total    :  " USING "##"; ELECTRODES
1930 :
1940 LOCATE 18,40,0
1950 PRINT "Output datasets processed:"
1960 :
1970 LOCATE 20,5,0
1980 FOR I=1 TO 4:SOUND RND*1000+37,2:NEXT I
1990 PRINT "Working ..."
2000 :
2010 IF DATDRIVE$=HDRDRIVE$ GOTO 2090
2020 OPEN OUTDRIVE$+":SCAN500.HDR" FOR OUTPUT AS #1
2030 A$=CHR$(34)+A$+CHR$(34)              :REM  Write the .HDR file to the
2040 B$=CHR$(34)+B$+CHR$(34)              :REM    disk on which FIX500.DAT
2050 C$=CHR$(34)+C$+CHR$(34)              :REM    will reside
2060 PRINT #1, A;B;A$;B$;C$;C;D;E;F;G
2070 CLOSE #1
2080 :
2090 OPEN OUTDRIVE$+":FIX500.DAT" FOR OUTPUT AS #2
2100 :
2110 PRINT #2, USING "###";DATASETS
2120 :
2130 IF DATASETS>120 THEN MAXDATUM=120 ELSE MAXDATUM=DATASETS
2140 PASS=1
2150 :
2160 OPEN FILENAMES FOR INPUT AS #1
2170 INPUT #1, ELECTRODES
2180 :
2190 FOR K=1 TO DATASETS
2200  IF EOF(1) GOTO 2600
2210  INPUT #1, STARTTIME, ETIME
2220  TIME(K)=STARTTIME
2230  J=K-(PASS-1)*120
2240 :
2250  FOR I=1 TO ELECTRODES
2260   INPUT #1, DLOW, DHIGH
2270   DRES=DLOW+(DHIGH-240)*256
```

```
2280   DVOLTS=ABS(DRES*(5/4Ø96))
2290   DVOLTS(I,J)=DVOLTS/5
2300  NEXT I
2310 :
2320  KEY(1) ON
2330  KEY(1) STOP
2340 :
2350  LOCATE 13,48,Ø:PRINT TIMES
2360  LOCATE 18,32,Ø:PRINT USING "###":K
2370 :
2380  IF K<MAXDATUM GOTO 2570
2390  IF PASSES=1 GOTO 2580
2400  IF PASS=PASSES GOTO 2580
2410 :
2420  TEMPFILE$=TEMPDRIVE$+":TEMP"+MID$(STR$(PASS),2,1)+".TMP"
2430  OPEN TEMPFILES FOR OUTPUT AS #3
2440  FOR I=1 TO ELECTRODES
2450   FOR J=1 TO 120
2460    PRINT #3, USING "#.####";DVOLTS(I,J)
2470   NEXT J
2480   LOCATE 13,48,Ø:PRINT TIMES
2490   LOCATE 18,67,Ø:PRINT USING "###";I
2500  NEXT I
2510  CLOSE #3
2520 :
2530 PASS=PASS+1
2540 LOCATE 15,POSPASS,Ø:PRINT PASS
2550 MAXDATUM=12Ø*PASS:IF MAXDATUM>DATASETS THEN MAXDATUM=DATASETS
2560 :
2570 NEXT K
2580 CLOSE #1
2590
2600 JFINAL=J
2610 :
2620 FOR K=1 TO DATASETS
2630  PRINT #2, USING "###.###":TIME(K)
2640 NEXT K
2650
2660 IF PASSES=1 GOTO 2970
2670 :
2680 FOR PASS=1 TO PASSES-1
2690  TEMPFILE$=TEMPDRIVE$+":TEMP"+MID$(STR$(PASS),2,1)+".TMP"
2700  FILENUMBER=PASS+2
2710  OPEN TEMPFILES FOR INPUT AS #FILENUMBER
2720 NEXT PASS
2730
2740 FOR I=1 TO ELECTRODES
2750  FOR PASS=1 TO PASSES-1
2760  FILENUMBER=PASS+2
2770   FOR J=1 TO 120
2780    INPUT #FILENUMBER, VOLTAGE
2790    PRINT #2, USING "#.####";VOLTAGE
2800   NEXT J
2810   LOCATE 13,48,Ø:PRINT TIMES
2820  NEXT PASS
2830  FOR J=1 TO JFINAL
2840   PRINT #2, USING "#.####";DVOLTS(I,J)
2850  NEXT J
2860   LOCATE 13,48,Ø:PRINT TIMES
2870  LOCATE 18,67,Ø:PRINT USING "###";I
2880 NEXT I
2890 :
2900 CLOSE #2
2910 FOR PASS=1 TO PASSES-1
```

```
2920  TEMPFILE$=TEMPDRIVE$+":TEMP"+MID$(STR$(PASS),2,1)+".TMP"
2930  FILENUMBER=PASS+2
2940  CLOSE #FILENUMBER:KILL TEMPFILE$
2950 NEXT PASS:GOTO 3100
2960 :
2970 FOR I=1 TO ELECTRODES
2980  FOR J-1 TO DATASETS
2990   PRINT #2, USING "#.####";DVOLTS(I,J)
3000  NEXT J
3010   LOCATE 13,48,Ø:PRINT TIME$
3020   LOCATE 18,67,Ø:PRINT USING "###";I
3030 NEXT I
3040 CLOSE #2
3050 :
3Ø6Ø REM  ****************************************************************
3070 REM  Shutdown The Program
3Ø8Ø REM  ****************************************************************
3090 :
3100 LOCATE 2Ø,5,Ø                         :REM  Prtnt status message to screen
3110 PRINT 'Job  complete ...          "
3120 x
3130 LOCATE 21,5,Ø                         :REM  Print output filename to screen
3140 PRINT "The output filename is FIX5ØØ.DAT."
3150 :
3160 LOCATE 22,5,1                         :REM  Request continuation of program
3170 FOR I=1 TO 4:SOUND RND*1ØØØ+37,2:NEXT I
3180 PRINT 'Do you wtsh the results displayed graphically?  ";
3190 X$=INKEY$
3200  IF X$="y" OR X$="Y" THEN PRINT "Yes.":GOTO 3290
3210  IF X$="n" OR X$="N" THEN PRINT "No.":GOTO 3240
3220 GOTO 3190
3230 :
3240 LOCATE 24,5,Ø                         rREM  Prlnt shutdown message
3250 FOR I=1 TO 4:SOUND RND*1ØØØ+37,2:NEXT I
3260 PRINT "Program terminated ...  ";
3270 FOR I=1 TO 4ØØØ:NEXT I:CHAIN "HEAD5ØØ.EXE"
3280 :
3290 LOCATE 24,5,Ø                         :REM  Prlnt continuation message
3300 FOR I=1 TO 4:SOUND RND*1ØØØ+37,2:NEXT I
3310 PRINT 'Loadlng GRAPH500 graphic dlsplay driver ...";
3320 FOR I=1 TO 4ØØØ:NEXT I                :REM  Initiate tlme delay
3330 :
3340 CHAIN "GRAPH5ØØ.EXE"                  :REM  Load graphics dtsplay program
3350 :
3360 END
3370 :
338Ø REM  ****************************************************************
3390 REM  SUBROUTINE:  Error Trapping
34ØØ REM  ****************************************************************
3410 :
3420 N=ERR
3430 IF ERR<>53 GOTO 3630
3440 :
3450 IF DRIVES-'A"  THEN DRIVE$="B":RESUME
3460 IF DRIVE$="B"  THEN DRIVE$="C":RESUME
3470 IF DRIVE$="C"  THEN LOCATE 16.1.1
3480 IF FILEFLAG=2 THEN FILE$="SCAN5ØØ.HDR" ELSE FILE$="SCAN5ØØ.DAT"
3490 FILEFLAG=Ø
3500 PRINT TAB(1Ø) FILES " is not resident in the computer."
3510 PRINT TAB(1Ø) "Please lnstall " FILES " then press RETURN.  ";
3520 FOR I=1 TO 4:SOUND RND*1ØØØ+37,2:NEXT I
3530 :
3540 X$=INKEY$
3550  IF X$=CHR$(13) GOTO 3590
```

```
3560  IF X$="x" OR X$="X" THEN CHAIN "HEAD5ØØ.EXE"
3570 GOTO 3540
3580 :
3590 LOCATE 16,1Ø,Ø:PRINT TAB(8Ø)
3600 LOCATE 17,1Ø,Ø:PRINT TAB(8Ø)
3610 DRIVE$="A":RESUME
3620 :
3630 IF ERR<>71 GOTO 3670
3640 IF DRIVE$="A" THEN DRIVE$="B":RESUME
3650 IF DRIVE$="B" THEN DRIVE$="C":RESUME
3660 :
3670 LOCATE 2Ø,1Ø,1
3680 FOR I=1 TO 4:SOUND RND*1ØØØ+37,2:NEXT I
3690 PRINT "Program executfon has been halted because of runtime error ";
3700 PRINT USING "##";N;
3710 PRINT " in lIne ";:PRINT USING "####";ERL;
3720 PRINT "."
3730 :
3740 FOR I=1 TO 4:SOUND RND*1ØØØ+37,2:NEXT I
3750 FOR I=1 TO 4ØØØ:NEXT I:CHAIN "HEAD5ØØ.EXE"
3760 :
3770 END
3780 :
379Ø REM  ***********************************************************************
3800 REM  SUBROUTINE:  Functlon Key F1 To Termfnate Program Execution
381Ø REM  ***********************************************************************
3820 :
3830 CLOSE #1:CLOSE #2:KILL "B:FIX5ØØ.DAT"
3840 FOR I=1 TO 4:SOUND RND*1ØØØ+37,2:NEXT I
3850 LOCATE 17,1,Ø:PRINT TAB(8Ø)
3860 LOCATE 18,1,Ø:PRINT TAB(8Ø)
3870 LOCATE 2Ø,5,Ø:PRINT "Program executfon has been halted.  Orfgfnal data ";
3880 PRINT "has been preserved:"
3890 PRINT TAB(5) "however the fIle " CHR$(34) "FIX5ØØ.DAT" CHR$(34);
3900 PRINT " has been erased."
3910 :
3920 FOR I=1 TO 4:SOUND RND*1ØØØ+37,2:NEXT I
3930 FOR I=1 TO 4ØØØ:NEXT I:CHAIN "HEAD5ØØ.EXE"
3940 :
3950 END
```

```
1000 REM    ****************************************************************
1010 REM    *SCREEN GRAPHIC GENERATER FROM FIX500-GENERATED FIX500.DAT FILE*
1020 REM    ****************************************************************
1030 REM    *                                                              *
1040 REM    *                 Program name:  GRAPH500.BAS                  *
1050 REM    *                                                              *
1060 REM    *                          Written for                         *
1070 REM    *      Stanford University Petroleum Engineering Department    *
1080 REM    *            by Lawrence W. Bouett, August 1, 1985             *
1090 REM    *                                                              *
1100 REM    ****************************************************************
1110 :
1120 :
1130 REM  ****************************************************************
1140 REM  PRINT THE PROGRAM HEADER
1150 REM  ****************************************************************
1160 :
1170 GOSUB 3380
1180 :
1190 LOCATE 6,1,0,0,7                    :REM  Print sign-on message
1200 PRINT TAB(10) "This program graphs voltage versus time for each of the"
1210 PRINT TAB(10) "electrodes from data in the current FIX500.DAT file."
1220 :
1230 ON ERROR GOTO 5340                  :REM  Initialize error trapping
1240 :                                   :REM    subroutine
1250 CHANGEFLAG=0                        :REM  Set internal flags
1260 DEMOFLAG=0
1270 STARTFLAG=0
1280 :
1290 DRIVE$="A"                          :REM  Locate file FIX500.DAT
1300 OPEN DRIVE$+":FIX500.DAT" FOR INPUT AS #1
1310 CLOSE #1
1320 :
1330 LOCATE 9,1,0
1340 PRINT TAB(10) "FIX500.DAT is resident on drive " DRIVE$ "."
1350 :
1360 LOCATE 11,1,0
1370 PRINT TAB(5) "From the following options:"
1380 LOCATE 13,1,0
1390 PRINT TAB(10) "1     Consecutive display with user-set time delay"
1400 PRINT TAB(10) "2     Consecutive display with user-toggled time delay"
1410 PRINT TAB(10) "3     Display a particular electrode"
1420 LOCATE 17,1,1
1430 PRINT TAB(5) "Please enter your selection:  ";
1440
1450 X$=INKEY$
1460  IF X$="1" THEN PRINT X$:CHOICE=1:GOTO 1510
1470  IF X$="2" THEN PRINT X$:CHOICE=2:GOTO 1520
1480  IF X$="3" THEN PRINT X$:CHOICE=3:GOTO 1530
1490 GOTO 1450
1500 :
1510 SELECTION$="consecutive display with user-set time delay.":GOTO 1550
1520 SELECTION$="consecutive display with user-toggled time delay.":GOTO 1550
1530 SELECTION$="single electrode display."
1540 :
1550 LOCATE 11,1,0:PRINT TAB(80)
1560 LOCATE 13,1,0:PRINT TAB(80)
1570 LOCATE 14,1,0:PRINT TAB(80)
1580 LOCATE 15,1,0:PRINT TAB(80)
1590 LOCATE 17,1,0:PRINT TAB(80)
1600 :
1610 LOCATE 10,1,0
1620 PRINT TAB(10) "Option is " SELECTION$:PRINT
1630 :
```

```
1640 FILENAME1$=DRIVE$+":SCAN500.HDR"
1650 FILENAME2$=DRIVE$+":FIX500.DAT"
1660 :
1670 IF CHOICE<>1 GOTO 1750
1680 PRINT TAB(10) "Time delay between graphs in seconds:  ";
1690 INPUT: "",DELAY
1700 LOCATE ,,0
1710 IF DELAY>0 THEN PRINT:PRINT:GOTO 1730
1720 DELAY=2:PRINT "2":PRINT
1730 DURATION=DELAY*2000
1740
1750 FILENAME$="SCAN500.HDR"
1760 OPEN FILENAME1$ FOR INPUT AS #1      :REM  Initialize variables for message
1770 INPUT #1, A,B,A$,B$,C$,C,D,E,F,G    :REM    written to the screen
1780 CLOSE #1
1790
1800 RUNNUMBER=A
1810 DEMOFLAG=B
1820 IF DEMOFLAG=0 THEN DEMOFLAG$="" ELSE DEMOFLAG$="demonstration "
1830 ELECTRODES=D
1840 DATASETS=E
1850
1860 IF RUNNUMBER>100 THEN F1$="###" ELSE F1$="##"
1870 IF RUNNUMBER<10 THEN F1$="#"
1880
1890 IF DATASETS>100 THEN F2$="###" ELSE F2$="##"
1900 IF DATASETS<10 THEN F2$="#"
1910
1920 PRINT TAB(10) "This dataset is " DEMOFLAGS "run number ";
1930 PRINT USING F1$; RUNNUMBER;
1940 PRINT ".  There are ";
1950 PRINT USING "##"; ELECTRODES;
1960 PRINT " electrodes,"
1970 PRINT TAB(10) "each with ";
1980 PRINT USING F2$; DATASETS;
1990 PRINT " timesteps.":PRINT
2000
2010 REM  ******************************************************************
2020 REM  SETUP FUNCTION KEYS
2030 REM  ******************************************************************
2040
2050 ON KEY(1) GOSUB 3570                 :REM  Set up F1 to halt execution
2060 KEY(1) ON                            :REM  Activate F1 key capture
2070 KEY(1) STOP                          :REM  Retain F1 for next KEY(1) ON
2080
2090 ON KEY(2) GOSUB 3810                 :REM  Set up F2 to change window
2100 KEY(2) ON                            :REM  Activate F2 key capture
2110 KEY(2) STOP                          :REM  Retain F2 for next KEY(2) ON
2120
2130 ON KEY(3) GOSUB 4090                 :REM  Set up F3 to maximize data
2140 KEY(3) ON                            :REM  Activate F3 key capture
2150 KEY(3) STOP                          :REM  Retain F3 for next KEY(3) ON
2160 :
2170 KEY(4) OFF:KEY(5) OFF:KEY(6) OFF:KEY(7) OFF:KEY(8) OFF
2180 ON KEY(9) GOSUB 5730                 :REM  Set up F9 to generate PLOT file
2190 KEY(9) ON                            :REM  Activate F9 key capture
2200 KEY(9) STOP                          :REM  Retain F9 for next KEY(9) ON
2210
2220 ON KEY(10) GOSUB 6040                :REM  Set up F10 to display HELP menu
2230 KEY(10) ON                           :REM  Activate F10 key capture
2240 KEY(10) STOP                         :REM  Retain F10 for next KEY(10) ON
2250
2260 REM  ******************************************************************
2270 REM  INITIALIZE THE SCREEN FOR GRAPHIC DISPLAY
```

```
2280 REM ********************************************************************
2290 :
2300 DIM TIME(1000),YVALUE(1000),XAXIS(1000),YAXIS(1000)
2310 :
2320 MATCH=0:IF CHOICE<>3 GOTO 2380
2330 PRINT TAB(10) 'Enter the electrode to be dlsplayed: ";
2348 INPUT: ".MATCH
2350 IF MATCH>0 AND MATCH<-ELECTRODES THEN PRINT:PRINT:GOTO 2380
2360 FOR I=1 TO 4:SOUND RND*1000+37,2:NEXT I
2370 LOCATE ,1,0:PRINT TAB(80):LOCATE ,1,0:GOTO 2330
2380 PRINT TAB(10) 'The program is ready for graphlc dtsplay.'
2390 PRINT TAB(10) 'Function key F1 terminates program executton.'
2400 IF CHOICE=1 THEN PRINT:GOTO 2440
2410 PRINT TAB(10) 'Functlon key F2 changes the dtsplay wfndow.'
2420 PRINT TAB(10) "The RETURN key toggles the next display.":PRINT
2430 :
2440 LOCATE -,I
2450 PRINT TAB(10) 'Press the RETURN key to begin. ";
2460 X$=INKEY$
2470  IF X$=CHR$(13) GOTO 2500
2480 GOTO 2460
2490 :
2500 LOCATE ,1,0:PRINT TAB(80)
2510 LOCATE ,1,0:PRINT TAB(10) "Working ...";
2520 FILENAME$="FIX500.DAT"
2530 OPEN FILENAME2$ FOR INPUT AS #1
2540 INPUT #1, DATASETS
2550 :
2560 FOR I=1 TO DATASETS
2570  INPUT #1, TIME(I)
2580 NEXT I
2590 XINC=592/TIME(DATASETS)
2600 XMIN=0:XMAX=TIME(DATASETS)
2610 XSTART=1:XFINISH=DATASETS
2620 :
2630 ELECTRODE=0
2640 :
2650 IF EOF(1) THEN GOSUB 3570
2660 :
2670 YMIN=9.999999E+37:YMAX=-9.999999E+37
2680 FOR I=1 TO DATASETS
2690  INPUT #1, YVALUE(I)
2700  IF I<5 GOTO 2740
2710  IF I<XSTART OR I>XFINISH GOTO 2740
2720  IF YVALUE(I)>YMAX THEN YMAX=YVALUE(I)
2730  IF YVALUE(I)<YMIN THEN YMIN=YVALUE(I)
2740 NEXT I
2750 IF ABS(YMAX-YMIN)<.0001 THEN YMIN=YMAX-.1
2760 OFFSET=(YMAX-YMIN)/20
2770 YMIN=YMIN-OFFSET:YMAX=YMAX+OFFSET:YINC=160/(YMAX-YMIN)
2780 :
2790 ELECTRODE=ELECTRODE+1
2800 IF MATCH>ELECTRODE GOTO 2670
2810 :
2820 GOSUB 4690
2830 :
2840 LOCATE 1,44
2850 PRINT USING "##";ELECTRODE
2860 :
2870 IF XSTART<5 THEN START=5 ELSE START=XSTART
2880 FOR I-START TO XFINISH
2890  XAXIS(I)=49+INT(XINC*(TIME(I)-XMIN))
2900  YAXIS(I)=170-INT(YINC*(YVALUE(I)-YMIN))
2910  IF YAXIS(I)<10 THEN YAXIS(I)=10
```

```
2920  IF YAXIS(I)>17Ø THEN YAXIS(I)=17Ø
2930  IF I<=START GOTO 2950
2940  LINE (XAXIS(I-1),YAXIS(I-1))-(XAXIS(I),YAXIS(I))
2950 NEXT ▮
2960 :
2970 KEY(1) ON                              :REM   Check for program termination
2980 KEY(1) STOP                            :REM   Retain F1 for next KEY(1) ON
2990 :
3000 IF CHOICE=1 GOTO 3110
3010 STARTFLAG=Ø
3020 KEY(1) ON
3030 KEY(2) ON
3040 KEY(3) ON
3050 KEY(9) ON
3060 KEY(1Ø) ON
3070 X$=INKEY$
3080  IF X$=CHR$(13) 60TO 3138
3090 GOTO 3070
3100 :
3110 FOR I=1 TO DURATION:NEXT ▮
3120 :
3130 KEY(1) STOP:KEY(2) STOP:KEY(3) STOP
3140 KEY(9) STOP:KEY(1Ø) STOP
3150 IF YMAXFLAG=1 AND CHOICE03 GOTO 265Ø
3160 FOR I=START TO XFINISH
3170  IF I<=START GOTO 3190
318Ø  LINE (XAXIS(I-1),YAXIS(I-1))-(XAXIS(I),YAXIS(I)),Ø
3190 NEXT ▮
3200 :
3210 IF CHOICE<>3 GOTO 2650
3220 LOCATE 1Ø,25,1
3230 PRINT "Enter the next electrode:  ";
3240 INPUT: "", NEWMATCH
3250 LOCATE 1Ø,25,Ø:PRINT TAB(8Ø)
3260 IF NEWMATCH=>Ø AND NEWMATCH<=ELECTRODES GOTO 3281
3270 FOR I=1 TO 4:SOUND RND*1ØØØ+37,2:NEXT I:GOTO 322Ø
3280 IF NEWMATCH=Ø THEN NEWMATCH=ELECTRODE+1
3290 IF NEWMATCH>MATCH THEN MATCH=NEWMATCH:GOTO 267Ø
3300 CLOSE #1:MATCH=NEWMATCH:GOTO 253Ø
3310 :
3320 END
3330 :
3340 REM  ******************************************************************
3350 REM  SUBROUTINE:  Print the Program Header
3360 REM  ******************************************************************
3370 :
3380 WIDTH 8Ø:SCREEN Ø,Ø,Ø:LOCATE 1,1,Ø:CLS:KEY OFF
3390 :
3400 UL$=CHR$(2Ø1):UR$=CHR$(187):LR$=CHR$(188):LL$=CHR$(2ØØ)
3410 H$=CHR$(2Ø5):V$=CHR$(186):HS$=CHR$(196)
3420 :
3430 PRINT UL$H$H$H$H$H$H$H$H$H$H$H$H$H$H$H$H$H$H$H$H$H$H$H$H$H$H$H$H$;
3440 PRINT H$H$H$H$H$H$H$H$H$H$H$H$H$H$H$H$H$H$H$H$H$H$H$H$H$H$H$H$H$;
3450 PRINT H$H$H$H$H$H$H$H$H$H$H$H$H$H$H$H$H$H$H$H$UR$;
3460 PRINT V$ "  GRAPH5ØØ Screen Graphic Display Program" TAB(8Ø) V$;
3470 PRINT LL$H$H$H$H$H$H$H$H$H$H$H$H$H$H$H$H$H$H$H$H$H$H$H$H$H$H$H$H$;
3480 PRINT H$H$H$H$H$H$H$H$H$H$H$H$H$H$H$H$H$H$H$H$H$H$H$H$H$H$H$H$H$;
3490 PRINT H$H$H$H$H$H$H$H$H$H$H$H$H$H$H$H$H$H$H$H$LR$
35ØØ :
3510 RETURN
3520 :
3530 REM  ******************************************************************
3540 REM  SUBROUTINE:  Execute Function Key F1 - Terminate Program
3550 REM  ******************************************************************
```

```
3560 :
3570 CLOSE #1
3580 SCREEN Ø,Ø,Ø
3590 GOSUB 3430
3600 :
3610 LOCATE 5,1,Ø
3620 FOR I=1 TO 4:SOUND RND*1ØØØ+37,2:NEXT I
3630 PRINT TAB(1Ø) 'Program executton has been halted after displaying ";
3640 PRINT USING "##";ELECTRODE
3650 IF ELECTRODE>1 THEN PRINT TAB(1Ø) "graphs ";:GOTO 3670
3660 PRINT TAB(1Ø) "graph ";
3670 PRINT 'out of a total of ";
3680 PRINT USING '##';ELECTRODES;
3690 PRINT " in the dataset.'
3700 x
3710 LOCATE 9,1,1
3720 PRINT TAB(1Ø) 'Program terminated .... ",
3730 FOR I=1 TO 4ØØØ:NEXT I:CHAIN "HEAD5ØØ.EXE"
3740 x
3750 RETURN
3760 :
3770 REM ******************************************************************
3780 REM  SUBROUTINE:  Execute Functfon Keys F2-F4 -- Change Dfsplay WIndow
3790 REM ******************************************************************
3800 :
3810 KEY(1) STOP
3820 KEY(2) STOP
3830 KEY(3) STOP
3840 KEY(4) STOP
3850 KEY(9) STOP
3860 KEY(1Ø) STOP
3870 WINDOWFLAG=Ø:GOSUB 3380
3880 LOCATE 6.1.0
3890 PRINT TAB(1Ø) "From the followfng options:"
3900 LOCATE 8,1,Ø
3910 PRINT TAB(15) "X      Change the scale of the x-8x1s"
3920 PRINT TAB(15) "Y      Change the scale of the y-axis"
3930 PRINT TAB(15) "2      Change the scales of both axes.
3940 LOCATE 12,1,Ø
3950 PRINT TAB(15) 'D      Return to minimum scale on both axes (default)'
3960 LOCATE 14,1,1
3970 PRINT TAB(1Ø) 'Please enter your selection:  ";
3980 :
3990 X$=INKEY$
4000  IF X$="x" OR X$="X" THEN PRINT 'Change x-axis":GOTO 4060
4010  IF X$="y" OR X$="Y" THEN PRINT "Change y-axis":GOTO 4070
4020  IF X$="2" THEN PRINT 'Change both axes":GOTO 4080
4030  IF X$="d" OR X$="D" THEN PRINT "Maximize all data":GOTO 4090
4040 GOTO 3990
4050
4Ø6Ø PRINT:GOSUB 417Ø:GOSUB 428Ø:GOSUB 452Ø:GOSUB 469Ø:GOTO 2840
4070 PRINT:GOSUB 442Ø:GOSUB 463Ø:GOSUB 469Ø:GOTO 2840
4080 PRINT:GOSUB 417Ø:GOSUB 428Ø:GOSUB 442Ø:GOSUB 463Ø:GOSUB 469Ø:GOTO 2840
4090 KEY(1) STOP:KEY(2) STOP:KEY(3) STOP:KEY(9) STOP:KEY(1Ø) STOP
4100 XMIN=Ø:XMAX=TIME(DATASETS):YMIN=9.999999E+37:YMAX=-9.999999E+37
4110 GOSUB 428Ø:GOSUB 452Ø:GOSUB 469Ø:GOTO 2840
4120 :
4130 REM ****************************************************************
4140 REM  SUBROUTINE:  Change The X-axis Limits
4150 REM ****************************************************************
4160 :
4170 LOCATE ,1,Ø:PRINT TAB(8Ø);:LOCATE ,1,Ø
4180 PRINT TAB(1Ø) "Enter the limits of the x-axis (min, max):  ";
4190 INPUT; "",XMIN,XMAX
```

```
4200 IF XMAX>TIME(DATASETS) THEN XMAX=TIME(DATASETS)
4210 IF XMIN<XMAX THEN RETURN
4220 FOR I=1 TO 4:SOUND RND*1ØØØ+37.2:NEXT I:GOTO 4170
423Ø :
424Ø REM  *****************************************************************
425Ø REM  SUBROUTINE:  Index The X-axis Values And Calulate XINC
426Ø REM  *****************************************************************
427Ø :
428Ø XMINTRIP=Ø
429Ø FOR I=1 TO DATASETS
43ØØ  IF XMINTRIP=1 GOTO 432Ø
431Ø  IF TIME(I)<XMIN GOTO 433Ø ELSE XSTART=I:XMINTRIP=1
432Ø  IF TIME(I)>XMAX GOTO 435Ø
433Ø NEXT I
434Ø :
435Ø IF I>DATASETS THEN XFINISH=DATASETS ELSE XFINISH=I
436Ø XINC=592/(TIME(XFINISH)-TIME(XSTART)):RETURN
437Ø :
438Ø REM  *****************************************************************
439Ø REM  SUBROUTINE:  Change The Y-axis Limits
44ØØ REM  *****************************************************************
441Ø :
442Ø LOCATE ,1,Ø:PRINT TAB(8Ø);:LOCATE ,1,Ø
443Ø PRINT TAB(1Ø) "Enter the limits of the y-axis (min, max):  ";
4440 INPUT; "",YMAX,YMIN
445Ø IF YMIN>YMAX THEN RETURN
446Ø FOR I=1 TO 4:SOUND RND*1ØØØ+37,2:NEXT I:GOTO 442Ø
447Ø :
448Ø REM  *****************************************************************
449Ø REM  SUBROUTINE:  Index The Y-axis Values And Maximize Them
45ØØ REM  *****************************************************************
451Ø :
452Ø YMIN=9.999999E+37:YMAX=-9.999999E+37
4530 IF XSTART<5 THEN START=5 ELSE START=XSTART
4540 FOR I=START TO XFINISH
4550  IF YVALUE(I)>YMAX THEN YMAX=YVALUE(I)
4560  IF YVALUE(I)<YMIN THEN YMIN=YVALUE(I)
4570 NEXT I
4580 :
4590 IF ABS(YMAX-YMIN)<.ØØØ1 THEN YMIN=YMAX-.1
4600 OFFSET=(YMAX-YMIN)/2Ø
4610 IF YFLAG=Ø THEN YMAX=YMAX+OFFSET:YMIN=YMIN-OFFSET
4620 :
4630 YINC=16Ø/(YMAX-YMIN):YFLAG=Ø:RETURN
4640 :
465Ø REM  *****************************************************************
4660 REM  SUBROUTINE:  Draw Coordinate Axes And Scale Axes
467Ø REM  *****************************************************************
4680 :
4690 CLS:KEY OFF:SCREEN 2
4700 :
4710 LOCATE 1Ø,1:PRINT "V"
4720 LOCATE 11,1:PRINT "o"
4730 LOCATE 12,1:PRINT "1"
4740 LOCATE 13,1:PRINT "t"
4750 LOCATE 14,1:PRINT "s"
4760 :
4770 LINE (47,8)-(47,174)
4780 LINE (48,8)-(48,174)
4790 LINE (42,171)-(639,171)
4800 :
4810 LOCATE 24,35:PRINT "Ttme (seconds)";
4820 :
4830 LOCATE 1,34
```

```
4840 PRINT "Electrode";
4850 :
4860 REM  Scale The X-axis And Print Tics
4870
4880 EXPONENT=Ø
4890 SCALER=TIME(XFINISH)-TIME(XSTART)
4900 IF SCALER<=1Ø GOTO 4930
4910 SCALER=SCALER/1Ø
4920 EXPONENT=EXPONENT+1:GOTO 4900
4930 SCALER=INT(SCALER)
4940 :
4950 FOR I=Ø TO SCALER
4960  X1=47+XINC*I*1Ø^EXPONENT
4970  X2=X1+1
4980  LINE (X1,171)-(X1,174)
4990  LINE (X2,171)-(X2,174)
5000  COLUMN=INT(X1/8)
5010  IF COLUMN>77 THEN COLUMN-78
5020  XSCALE=I*1Ø^EXPONENT+XMIN
5030  LOCATE 23,COLUMN
5040  PRINT USING "###";XSCALE;
5050 NEXT I
5060 :
5070 REM  Scale The Y-axis And Print Tics
5080 :
5090 FOR YTIC=11 TO 155 STEP 16
5100  LINE (42,YTIC)-(47,YTIC)
5110 NEXT YTIC
5120 :
5130 YSTEP=(YMAX-YMIN)/1Ø
5140 ROW=Ø
5150 FOR YSCALE=YMAX TO VMIN STEP -YSTEP
5160  IF YSCALE=YMAX THEN FORMATS="#.##" ELSE FORMATS='.##"
5170  IF YSCALE=YMAX THEN COLUMN=2 ELSE COLUMN=3
5180  ROW=ROW+2
5190  LOCATE ROW,COLUMN,Ø
5200  PRINT USING FORMATS;YSCALE
5210  IF YSTEP<.Ø1 GOTO 5250
5220  IF YSCALE-YSTEP<YMIN GOTO 525Ø
5230 NEXT VSCALE
5240 :
5250 LOCATE 22,2,Ø
5260 PRINT USING "#.##";YMIN
5270 :
5280 RETURN
5290 :
5300 REM  *****************************************************************
5310 REM  SUBROUTINE:  Error Trapping
5320 REM  *****************************************************************
5330 :
5340 N=ERR
5350 IF ERR<>53 GOTO 5550
5360
5370 IF DRIVES="A" THEN DRIVE$="B":RESUME
5380 IF DRIVE$="B" THEN DRIVE$="C":RESUME
5390 IF DRIVE$="C" THEN LOCATE 16.1.1
5400 IF FILEFLAG=2 THEN FILE$="SCAN5ØØ.HDR" ELSE FILE$="FIX5ØØ.DAT"
5410 FILEFLAG-0
5420 PRINT TAB(1Ø) FILES " is not resident in the computer.'
5430 PRINT TAB(1Ø) "Please install " FILES " then press RETURN.  ";
5440 FOR I=1 TO 4:SOUND RND*1ØØØ+37,2:NEXT I
5450 :
5460 X$=INKEY$
5470  IF X$=CHR$(13) GOTO 5510
```

```
5480  IF X$="x" OR X$="X" THEN CHAIN "HEAD5ØØ.EXE"
5490 GOTO 5460
5500 :
5510 LOCATE 16,1Ø,Ø:PRINT TAB(8Ø)
5520 LOCATE 17,1Ø,Ø:PRINT TAB(8Ø)
5530 DRIVE$="A":RESUME
5540 :
5550 IF ERR<>71 GOTO 5590
5560 IF DRIVES-"A" THEN DRIVE$="B":RESUME
5570 IF DRIVE$="B" THEN DRIVE$="C":RESUME
5580 :
5590 LOCATE 2Ø,1Ø,1
5600 FOR I=1 TO 4:SOUND RND*1ØØØ+37,2:NEXT I
5610 PRINT 'Program executfon has been halted because of runtlme error ";
5620 PRINT USING "##";N;
5630 PRINT "."
5640 :
5650 FOR I=1 TO 4ØØØ:NEXT I:CHAIN "HEAD5ØØ.EXE"
5660 :
5670 END
5680 :
569Ø REM ******************************************************************
5700 REM  SUBROUTINE:  Execute Function Key F9 - Generate PLOT FIle
571Ø REM ******************************************************************
5720 :
5730 KEY(1) STOP:KEY(2) STOP:KEY(3) STOP:KEY(4) STOP
5740 KEY(9) STOP:KEY(1Ø) STOP
5750 LOCATE 25,1,1
5760 PRINT 'Enter the drive desfgnatfon:  ";
5770 X$=INKEY$
5780  IF X$="a" OR X$="A" THEN DRIVE$="A":GOTO 5830
5790  IF X$="b" OR X$="B" THEN DRIVE$="B":GOTO 5830
5800  IF X$="c" OR X$="C" THEN DRIVE$="C":GOTO 5830
5810 GOTO 5770
5820 :
5830 PRINT DRIVES;
5840 RUNNUMBER$=MID$(STR$(RUNNUMBER),2)
585Ø ELECTRODE$=MID$(STR$(ELECTRODE),2)
5860 PLOTFILE$=DRIVE$+":RUN"+RUNNUMBER$+"-"+ELECTRODE$+".DAT"
587Ø :
5880 OPEN PLOTFILES FOR OUTPUT AS #2
5690 FOR I=START TO XFINISH
5900  PRINT +2, USING "###.###  #.####";TIME(I),YVALUE(I)
5918 NEXT I
5920 CLOSE #2
5930 :
5940 LOCATE 25,1,1
5950 PRINT 'FIle " PLOTFILES " created.  Press RETURN to contlnue.  ";
5960 X$=INKEY$
597Ø  IF X$=CHR$(13) THEN LOCATE 25,1,Ø:PRINT TAB(8Ø);:GOTO 3Ø2Ø
5980 GOTO 5960
5990 :
6OOO REM ******************************************************************
6010 REM  SUBROUTINE:  Execute Functfon Key F1Ø - Dfsplay HELP Menu
6020 REM ******************************************************************
6030 :
6040 KEY(1) STOP:KEY(2) STOP:KEY(3) STOP:KEY(4) STOP
6050 KEY(9) STOP:KEY(1Ø) STOP
6060 GOSUB 338Ø
6070 LOCATE 2,44,Ø:PRINT "HELP Menu"
6080 :
6090 LOCATE 6,1,Ø
6100 PRINT TAB(5) "Functfon keys F1 - F1Ø perform the followfng functfons:"
61IO LOCATE 8,1,Ø
```

```
6120 PRINT TAB(1Ø) 'Key       Functton'
6130 PRINT TAB(1Ø);:FOR I=1 TO 6Ø:PRINT HS$;:NEXT I
6140 PRINT TAB(1Ø) "F1        Termfnate program executfon"
6150 PRINT TAB(1Ø) "F2        Change graphtc dtsplay wfndow"
6160 PRINT TAB(1Ø) "F3        Maximize graphlc dlsplay of the data on both axes'
6170 PRINT TAB(1Ø) "F4"
618Ø PRINT TAB(1Ø) "F5"
6190 PRINT TAB(1Ø) "F6"
6200 PRINT TAB(1Ø) "F7"
6210 PRINT TAB(1Ø) "F8"
6220 PRINT TAB(1Ø) "F9        Generate a PLOT flle of the data on the screen.
6230 PRINT TAB(1Ø) "F1Ø       Prtnt the HELP menu (thfs screen)'
6240 PRINT:PRINT:PRINT TAB(5) 'Press the RETURN key to contfnue program ";
6250 PRINT 'executlon.   ";
6260 LOCATE ,,1
6270 X$=INKEY$
6280  IF X$=CHR$(13) GOTO 2820
6290 GOTO 6270
6300 :
6310 RETURN
```

```
1ØØØ REM ************************************************************************
1010 REM *      Electrode Charge Time/Tracer Concentration Calibration          *
1020 REM *            Program For KEITHLEY/das Series 500 AOMI Card              *
1030 REM *      Using a Digital Multi-Meter Or Internal AMMI Or AIM3 Card        *
1040 REM *                                                                       *
1050 REM *                     Program Name:   TEST5ØØ.BAS                       *
1060 REM *                                                                       *
1070 REM *             With Options For Alternative Voltage Inputs               *
1080 REM *          And A Linearity Program For Any Given Voltage Range          *
1090 REM *                                                                       *
1100 REM *             Written Ø2 March 1986 by Lawrence W. Bouett               *
1110 REM *                  (C) Copyright 1986 by LaBoDAS, Inc.                  *
112Ø REM ************************************************************************
1130 :
1140 DEF SEG=&HAFFØ                          :REM   Define segment address
1150 POKE &H9A,Ø                             :REM   Set global gain at x1
1160 DMM=Ø:AMM=Ø:SPACEBAR=Ø                  :REM   Set internal flags to zero
1170 CHARGE=Ø:SELECTION=Ø
1180 :
1190 GOSUB 9040                              :REM   Print program header
1200 :
1210 STROBE=&H9D                             :REM   Define strobe address
1220 POKE STROBE,64                          :REM   Enable strobe
1230 :
1240 LOCATE 5,1,Ø,Ø,7                        :REM   Select program
1250 PRINT TAB(1Ø) 'Choose  a program option from the list below:'
1260 LOCATE 7,1,Ø
1270 PRINT TAB(15) "1     General test electrode program"
1280 PRINT TAB(15) "2     Special effluent concentration program"
1290 PRINT TAB(15) "3     General calibration program"
1300 PRINT:PRINT TAB(15) 'X      Exit the program"
1310 :
1320 LOCATE 14,1,1
1330 PRINT TAB(1Ø) "Enter your selection:   ";
1340 X$=INKEY$
1350   IF X$="1" THEN SELECTION=1:GOTO 1450
1360   IF X$="2" THEN SELECTION=2:GOTO 1340
1370   IF X$="3" THEN SELECTION=3:GOTO 1730
1380   IF X$="x" OR X$="X" GOTO 7250
1390 GOTO 1340
1400 :
141Ø REM ********************************************************************
1420 REM  PROGRAM 1:  Set Up To Run The General Test Electrode Program
143Ø REM ********************************************************************
1440 :
1450 CMDA=&H82:CMDB=&H83:AIMSLOT=1:AOMSLOT=2
1460 CHANNELOUT=Ø:CHANNELIN=Ø                :REM   Define program parameters
1470 DIVISOR=5/4Ø96:OFFSET=2.5
1480 AMM=1:SPACEBAR=1
1490 :
1500 GOSUB 9040
1510 LOCATE 6,1,Ø                            :REM   Check options
1520 PRINT TAB(5) 'You are performing a voltage tnput check using ";
1530 PRINT "the test electrode."
1540 PRINT TAB(5) "If this is NOT what you want then press ESC; otherwlse,';
1550 PRINT "immerse"
1560 PRINT TAB(5) "the test electrode in the solution to be tested";
1570 PRINT "then press RETURN.   ";
1580 LOCATE ,,1
1590 :
1600 X$=INKEY$
1610  IF X$=CHR$(13) GOTO 1650
1620  IF X$=CHR$(27) THEN CLS:GOTO 1140
1630 GOTO 1600
```

```
1640 :
1650 LOCATE 7,1,Ø:PRINT TAB(8Ø)          :REM  Screen cleanup
1660 LOCATE 8,1,Ø:PRINT TAB(8Ø):GOSUB 96ØØ:GOTO 6860
1670 :
168Ø :
1690 REM  *******************************************************************
1700 REM  PROGRAM 3:  Set Up To Run The General Calibration Program
1710 REM  *******************************************************************
1720 :
1730 LOCATE 5,1,Ø:PRINT TAB(8Ø)          :REM  Screen cleanup
1740 LOCATE 7,1,Ø:PRINT TAB(8Ø)
1750 LOCATE 8,1,Ø:PRINT TAB(8Ø)
1760 LOCATE 9,1,Ø:PRINT TAB(8Ø)
1770 LOCATE 11,1,Ø:PRINT TAB(8Ø)
1786 LOCATE 12,1,Ø:PRINT TAB(8Ø)
1790 LOCATE 14,1,Ø:PRINT TAB(8Ø)
1800 :
1810 LOCATE 5,1,Ø                        :REM  Get AOMI card type
1820 PRINT TAB(1Ø) 'Enter the followlng information:"
1830 LOCATE 7,1,Ø
1840 PRINT TAB(15) 'AOMI card type:  ";
1850 PRINT TAB(17) '2     AOM1/2"
1860 PRINT TAB(17) '5   . AOM1/5"
1870 :
1880 LOCATE 7,32,1
1890 X$=INKEY$
1900   IF X$="2" GOTO 1940
1910   IF X$="5" GOTO 1940
1920   IF X$="x" OR X$="X" COTO 7250
1930 GOTO 1890
1940 PRINT X$:CARD=VAL(X$):TYPE$=X$      :REM  Set varlables
1950 :
1960 LOCATE 11.1.1                       :REM  Get AOMI slot number
1970 PRINT TAB(15) "AOM1/" TYPES " slot number (2-1Ø):  ";
1980 GOSUB 9260
1990 PRINT X$:AOMSLOT=VAL(X$)
2000 :
2010 LOCATE 13.1.1                       :REM  Get AOMI channel number
2020 IF CARD=2 THEN A$="(Ø-1)" ELSE A$="(Ø-4)"
2030 PRINT TAB(15) 'Output channel number of AOM1/" TYPES " " AS ":  ";
2040 X$=INKEY$
2050   IF X$="Ø" GOTO 2130
2060   IF X$="1" GOTO 213Ø
2070   IF CARD=2 GOTO 2040
2080   IF X$="2" GOTO 2130
2090   IF X$="3" GOTO 2130
2100   IF X$="4" GOTO 2130
2110   IF X$="x" OR X$="X" GOTO 7250
2120 GOTO 2040
2130 PRINT X$:CHANNELOUT=VAL(X$)
2140 :
2150 LOCATE 15,1,Ø                       :REM  Get output range of AOMI
2160 PRINT TAB(15) 'Output range of AOM1/" TYPES ":  ";
2170 PRINT TAB(17) "1     0 to +5V"
2180 PRINT TAB(17) '2     0 to +1ØV"
2190 PRINT TAB(17) '3     +/- 2.5V"
2200 PRINT TAB(17) "4     +/- 5V"
2210 PRINT TAB(17) "5     +/- 1ØV"
2220 :
2230 LOCATE 15,4Ø,1
2240 X$=INKEYS
2250   IF X$="1" THEN GOSUB 945Ø:GOTO 2330
2260   IF X$="2" THEN GOSUB 947Ø:GOTO 2330
2270   IF X$="3" THEN GOSUB 949Ø:GOTO 2330
```

```
2280   IF X$="4" THEN GOSUB 951Ø:GOTO 2330
2290   IF X$="5" THEN GOSUB 953Ø:GOTO 2330
2300   IF X$="x" OR X$="X" GOTO 7250
2310 GOTO 2240
2320 '
2330 PRINT X$:VOLTS$=LOW$
2340 :
2350 LOCATE 22,1,Ø                          :REM  Toggle DMM or AMMI card for Input
2360 PRINT TAB(15) "Method of output displays'
2370 PRINT TAB(17) "1     Dlgltal Multi-meter'
2380 PRINT TAB(17) "2     AMMI or AIM3 card":
2390 '
2400 LOCATE 22,42,1
2410 X$=INKEYS
2420   IF X$="1" THEN METHOD$="a digital multi-meter":DMM=1:GOTO 4660
2430   IF X$="2" THEN DMM=Ø:GOTO 2510
2448   IF X$="x" OR X$="X" GOTO 7250
2450 GOTO 2410
2460
2470 REM  ****************************************************************
2480 REM  ANALOG INPUT OPTION FOR PROGRAM 3
2490 REM  ****************************************************************
2500 '
2510 GOSUB 9040                             :REM  Prlnt program header
2520 LOCATE 6,1,Ø                           :REM  Get tnput card type
2530 PRINT TAB(1Ø) 'Enter the input card type:"
2540 PRINT TAB(17) "1     AMMI card"
2550 PRINT TAB(17) '2     AIM3 card"
2560 LOCATE 6,38,1
2570 X$=INKEY$
2580   IF X$="1" THEN METHOD$="the AMMI card":A$="(Ø-7)":AMM=1:ROW=1Ø:GOTO 267Ø
2590   IF X$="2" THEN METHOD$="the AIM3 card":A$="(Ø-31)":ROW=12:GOTO 2620
2600   IF X$="x" OR X$="X" GOTO 7250
2610 GOTO 2570
2620 PRINT X$
2630 :
2640 LOCATE 1Ø,1,1                          :REM  Get AIM3 slot number
2650 PRINT TAB(1Ø) 'Enter the slot number of the AIM3 card (2-1Ø):  ";
2660 GOSUB 9260
2670 PRINT X$:AIMSLOT=VAL(X$)
2680 '
2690 LOCATE ROW,1,1                         :REM  Get AOMI channel number
2700 PRINT TAB(1Ø) 'Enter the Input channel on " METHODS " " AS;
2710 IF AMM=Ø THEN PRINT " then press RETURN:  ";:GOTO 2860
2720 PRINT ":  ";
2730 X$=INKEY$
2740   IF X$="Ø" GOTO 2840
2750   IF X$="1" GOTO 2840
2760   IF X$="2" GOTO 2840
2770   IF X$="3" GOTO 2840
2780   IF X$="4" GOTO 2840
2790   IF X$="5" GOTO 2840
2800   IF X$="6" GOTO 2840
2810   IF X$="7" GOTO 2840
2820   IF X$="x" OR X$="X" GOTO 7250
2830 GOTO 2730
2840 PRINT X$:CHANNELIN=VAL(X$):GOTO 2900
2850 :
2860 INPUT "",CHANNELIN                     :REM  Get AIM3 channel number
2870 IF CHANNELIN=>Ø AND CHANNELIN <=31 GOTO 2900
2880 LOCATE ROW,1,1:BEEP:PRINT TAB(8Ø):GOTO 2690
2890 :
2900 LOCATE ROW+2,1,Ø                       :REM  Get callbration option
2910 PRINT TAB(1Ø) "Enter the type of callbration:  ";
```

```
2920 POSITION=POS(X)
2930 PRINT TAB(17) 1      channel' CHANNELOUT 'of  the AOM1/" TYPES " card ";
2940 PRINT 'in  slot" AOMSLOT
2950 PRINT TAB(17) '2      Electrode charge time/tracer concentration"
2960 :
2970 LOCATE ROW+2,POSITION,1
2980 X$=INKEYS
2990   IF X$="1" GOTO 4660
3000   IF X$="2" THEN PRINT X$:GOTO 3080
3010   IF X$="x" OR X$="X" COTO 7250
3020 GOTO 2980
3030 :
3040 REM  *******************************************************************
3050 REM  CHARGE TIME/CONCENTRATION TEST OPTION FOR ANALOG INPUT OPTION
3060 REM  *******************************************************************
3070 :
3080 LOCATE ROW+6,1,1                       :REM  Get tracer concentration
3090 PRINT TAB(1Ø) "Enter the concentration (ppm) then press RETURN:  ";
3100 POSITION=POS(X)
3110 INPUT '",CONCENTRATION
3120 IF CONCENTRATION=Ø THEN LOCATE ROW+6,POSITION,Ø:PRINT "distilled water"
3130 IF CONCENTRATION<Ø THEN LOCATE ROW+6,1,Ø:PRINT TAB(8Ø):BEEP:GOTO 3Ø8Ø
3140 :
3150 LOCATE ROW+8,1,Ø                       :REM  Get type of test
3160 PRINT TAB(1Ø) 'Enter  the type of test:  ";
3170 POSITION=POS(X)
3180 PRINT TAB(17) 1      Electrode charge time"
3190 PRINT TAB(17) 2      Input voltage ";
3200 IF CONCENTRATION>Ø GOTO 3220
3210 PRINT 'using  distilled water":GOTO 3240
3220 PRINT "at' CONCENTRATION "ppm tracer concentration"
3230 :
3240 LOCATE ROW+8,POSITION,1
3250 X$=INKEY$
3260   IF X$="1" THEN TEST$="an electrode charge time":CHARGE=1:GOTO 3310
3270   IF X$="2" THEN TEST$="a concentration voltage":GOTO 3310
3280   IF X$="x" OR X$="X" GOTO 7250
3290 COTO 3250
3380 :
3310 GOSUB 9040
3320 LOCATE 6,1,Ø                           :REM  Check options
3330 PRINT TAB(5) 'You  are performing " TESTS " test using channel';
3340 PRINT CHANNELOUT 'output'
3350 PRINT TAB(5) "from the AOM1/" TYPES " card in slot" AOMSLOT "at " RANGES;
3360 PRINT " volts DC and channel" CHANNELIN 'input"
3370 PRINT TAB(5) "of " METHODS;
3380 IF DMM=Ø THEN PRINT " in slot ";
3390 IF DMM=Ø AND AMM=1 THEN PRINT "1";:GOTO 3420
3400 IF AIMSLOT-10 THEN FORMATS="##'  ELSE FORMATS='#'
3410 IF DMM=Ø THEN PRINT USING FORMAT$;AIMSLOT;
3420 PRINT ".  The tracer ";
3430 IF CONCENTRATION>Ø GOTO 3450
3440 PRINT "is distilled water.  ";:POSITION=POS(X):GOTO 3460
3450 PRINT 'concentration is" CONCENTRATION 'ppm.'
3460 PRINT TAB(5) "If this is NOT what you want then press ESC; otherwise, ";
3470 PRINT "connect channel'
3480 PRINT TAB(4) CHANNELOUT 'output of the AOM1/" TYPES " card to test ";
3490 PRINT 'point  one (TP1) of the electrode"
3500 PRINT TAB(5) 'test  stand and channel" CHANNELIN 'input of " METHODS;
3510 PRINT " to test point two (TP2)"
3520 PRINT TAB(5) 'then  press RETURN.  ";
3530 :
3540 LOCATE ,,1:BEEP
3550 XS=INKEYS
```

```
3560   IF X$=CHR$(27) GOTO 1140
3570   IF X$=CHR$(13) GOTO 3600
3580 GOTO 3550
3590
3600 LOCATE 9,1,Ø:PRINT TAB(8Ø)           :REM   Screen cleanup
3610 LOCATE 1Ø,1,Ø:PRINT TAB(8Ø)
3620 LOCATE 11,1,Ø:PRINT TAB(8Ø)
3630 LOCATE 12,1,Ø:PRINT TAB(8Ø)
3640
3650 REM  ******************************************************************
3660 REM  SET THE INTERVAL TIMER
3670 REM  ******************************************************************
3680
3690 CO=5ØØØØ!                                :REM   Set pulses for counter one
3700 CZ=5ØØØØ!                                :REM   Set pulses for counter zero
3710
3720 HZ=INT(CZ/256)                           :REM   High byte counter zero
3730 LZ=CZ-(HZ*256)                           :REM   Low byte counter zero
3740 HO=INT(CO/256)                           :REM   High byte counter one
3750 LO=CO-(HO*256)                           :REM   Low byte counter one
3760 :
3770 POKE &HEØ,1                              :REM   Timer global command
3780
3790 POKE &HC3,54                             :REM   Counter zero control (timer)
3800 POKE &HCØ,LZ                             :REM   Wrlte low byte
3810 POKE &HCØ,HZ                             :REM   Wrlte high byte
3820
3830 POKE &HC3,116                            :REM   Counter one control (carry)
3840 POKE &HC1,LO                             :REM   Wrlte low byte
3850 POKE &HC1,HO                             :REM   Wrlte high byte
3860 :
3870 LOCATE 12,1,Ø                            :REM   Instructions to begin test
3880 BEEP
3890 PRINT TAB(5) 'Immerse the test electrode in the tracer ";
3900 PRINT "then press RETURN.  ";
3910
3928 LOCATE ,,1
3930 X$=INKEY$
3940   IF X$=CHR$(13) THEN LOCATE 12,1,Ø:PRINT TAB(8Ø):GOTO 397Ø
3950 GOTO 3930
3960
3970 OLDDLOW=5ØØ
3980 LOCATE 12,1,Ø                            :REM   Get output voltage
3990 BEEP
4080 PRINT TAB(5) "Enter an output voltage between " RANGES;
4010 PRINT "VDC then press RETURN:  ";:POSITION=POS(X)
4020
4030 INPUT "",VOLTAGE
4040 IF VOLTAGE=Ø THEN LOCATE 12,POSITION,Ø:PRINT 'O.OOO"
4050 IF VOLTAGE=>VAL(LOW$) AND VOLTAGE<=VAL(HIGH$) GOTO 4Ø8Ø
4060 LOCATE 12,1,Ø:PRINT TAB(8Ø):GOTO 3970
4070 :
4080 IF RANGE$="Ø to +5" OR RANGE$="Ø to +1Ø" GOTO 4100
4090 DVOLTS=FIX(VOLTAGE/DIVISOR)+2Ø48:GOTO 4110
4100 DVOLTS=FIX(VOLTAGE/DIVISOR)
4110 DHIGH=FIX(DVOLTS/256)
4120 DLOW=DVOLTS-(DHIGH*256)
4130 DRES=DLOW+DHIGH*256
4140 IF RANGE$="Ø to +5" OR RANGE$="Ø to +1Ø" GOTO 4160
4150 CVOLTAGE=(DRES-2Ø48)*DIVISOR:GOTO 4170
4160 CVOLTAGE=DRES*DIVISOR
4178 POKE CMDA,2*CHANNELOUT:POKE CMDB,DLOW
4180 POKE CMDA,2*CHANNELOUT+1:POKE CMDB,DHIGH
4190 POKE STROBE.l
```

```
4200 GOSUB 9820:STARTCOUNT#=CURRENTCOUNT#
4210
4220 POKE &H81,1                          :REM  Select slot 1 for AMMI card
4230 POKE &H80,CHANNELIN                  :REM  Select channel on AMMI card
4240 POKE &H9B,255                        :REM  Start A/D converslon
4250 IF PEEK(&H9B)<>127 GOTO 4250         :REM  Loop !f converslon incomplete
4260 DLOW=PEEK(&H80)                      :REM  Read low byte
4270 DHIGH=PEEK(&H81)                     :REM  Read high byte
4280 GOSUB 9820:ENDCOUNT#=CURRENTCOUNT#
4290 IF OLDDLOW=DLOW GOTO 4320
4300 OLDDLOW=DLOW:OLDENDCOUNT#=ENDCOUNT#:GOTO 4240
4310 :
4320 DRES=DLOW+(DHIGH-240)*256            :REM  Calculate resolut!on of datum
4330 DVOLTS=DRES*(20/4096)-10             :REM    and convert to volts
4340 :
4350 TIME=(OLDENDCOUNT#-STARTCOUNT#)*.000001046#
4360                                      :REM  Calculate elapsed time in seconds
4370 :
4380 LOCATE 14,1,0                        :REM  Prtnt resolut!on output voltage
4390 PRINT TAB(5) "Output voltage:  ";
4400 PRINT USING "###.#####";CVOLTAGE
4410 :
4420 LOCATE 15,1,0                        :REM  PrInt Input voltage
4430 PRINT TAB(5) "Input voltage :  ";    :REM    and charge tfme
4440 PRINT USING "###.#####";DVOLTS;
4450 IF CHARGE=0 GOTO 4490
4460 PRINT TAB(40) "Charge tlme:  ";
4470 PRINT USING "##.######";TIME;:PRINT " seconds"
4480 :
4490 LOCATE 21,1,1                        :REM  PrInt options to proceed
4500 PRINT TAB(5) 'Press RETURN to contInue or ESC to quit the program.  ";
4510 BEEP
4520 X$=INKEY$
4530   IF X$=CHR$(13) GOTO 4570
4540   IF X$=CHR$(27) THEN LOCATE 21,1,0:PRINT TAB(80):GOTO 7250
4550 GOTO 4520
4560 :
4570 LOCATE 21,1,0:PRINT TAB(80)          :REM  Screen cleanup
4580 LOCATE 15,1,0:PRINT TAB(80)
4590 LOCATE 14,1,0:PRINT TAB(80)
4600 LOCATE 12,1,0:PRINT TAB(80):GOTO 3870
4610 :
4620 REM  ******************************************************************
4630 REM  ANALOG OUTPUT VOLTAGE CALIBRATION OPTION
4640 REM  ******************************************************************
4650 :
4660 GOSUB 9040                           :REM  Print program header
4670 LOCATE 6,1,1                         :REM  Confirm all options
4680 BEEP
4690 PRINT TAB(5) "You are calibrat!ng channel" CHANNELOUT "of an AOM1/";
4700 PRINT TYPES " card fn slot" AOMSLOT "at " RANGES
4710 PRINT TAB(5) 'volts DC usIng ";
4720 IF DMM=0 THEN PRINT "channel" CHANNELIN 'of ";
4730 PRINT METHODS;
4740 IF DMM=0 THEN PRINT " in slot ";
4750 IF DMM=0 AND AMM=1 THEN PRINT "1";:GOTO 4780
4760 IF AIMSLOT=>10 THEN FORMAT$="##" ELSE FORMAT$="#"
4770 IF DMM=0 THEN PRINT USING FORMAT$;AIMSLOT;
4780 PRINT ".  ";:POSITION=POS(X):PRINT "If this !s NOT'
4790 PRINT TAB(5) "what you want then press ESC; otherwlse, connect channel';
4800 PRINT CHANNELOUT "of the AOM1/" TYPES;
4810 PRINT TAB(5) "card to ";
4820 IF DMM=0 THEN PRINT 'channel" CHANNELIN 'of ";
4830 PRINT METHODS " then press RETURN.  ";
```

```
4840 :
4850 X$=INKEY$
4860   IF X$=CHR$(13) GOTO 4900          :REM  Proceed if optlons are correct
4870   IF X$=CHR$(27) GOTO 1190          :REM  Return If corrections necessary
4880 GOTO 4850
4890
4900 LOCATE 7,POSITION,Ø:PRINT TAB(8Ø)  :REM  Screen cleanup
4910 LOCATE 8,1,Ø:PRINT TAB(8Ø)
4920 LOCATE 9,1,Ø:PRINT TAB(8Ø)
4930 :
4940 LOCATE 11,1,Ø
4950 BEEP
4960 PRINT TAB(5) 'Enter  your selection from the following optlons:  ";
4970 POSITION=POS(X)
4980 LOCATE 13,1,Ø                        :REM  Get calibration options
4990 PRINT TAB(1Ø) '1     Proceed wlth the calibration"
5000 PRINT TAB(1Ø) "2     Test a dlffercnt output voltage'
5010 IF DMM=1 GOTO 5030
5020 PRINT TAB(1Ø) "3     Run a linearltv test over a given voltage range"
5030 PRINT:PRINT TAB(1Ø) "X     Exit the-program"
5040 LOCATE 11,POSITION,1
505% X$=INKEY$
5060   IF X$="1" THEN GOSUB 517Ø:GOTO 6070
5070   IF X$="2" THEN SPACEBAR=1:VOLTS$=HIGH$:GOSUB 517Ø:GOSUB 96ØØ:GOTO 6860
5080   IF DMM=1 GOTO 5100
5090   IF X$="3" THEN GOSUB 517Ø:GOTO 5280
5100   IF X$="x" OR X$="X" GOTO 7260
5110 GOTO 5050
5120 :
5130 REM  ******************************************************************
5140 REM  INTERNAL SUBROUTINE:  SCREEN CLEANUP
5150 REM  ******************************************************************
5160
5170 LOCATE 11,1,Ø:PRINT TAB(8Ø)        :REM  Screen erase subroutine
5180 LOCATE 13,1,Ø:PRINT TAB(8Ø)        :REM    used with the INKEYS
5190 LOCATE 14,1,Ø:PRINT TAB(8Ø)        :REM    function above
5200 LOCATE 15,1,Ø:PRINT TAB(8Ø)
5210 LOCATE 16,1,Ø:PRINT TAB(8Ø)
5220 LOCATE 17,1,Ø:PRINT TAB(8Ø):RETURN
5230 :
5240 REM  ********************************************************************
5250 REM  OPTION 3:  Run A Linearity Test On The System Over A Given Range
5260 REM  ********************************************************************
5270
5280 LOCATE 11,1,Ø
5290 PRINT TAB(5) "Enter the followlns for the linearity test:"
5300 LOCATE 13,1,Ø
5310 PRINT TAB(1Ø) "Minimum voltage                    :  ";
5320 INPUT "",MINVOLTS
5330 LOCATE 14,1,Ø
5340 PRINT TAB(1Ø) "Maximum voltage                    :  ";
5350 INPUT "",MAXVOLTS
5360 LOCATE 15,1,Ø
5370 PRINT TAB(1Ø) "Voltage increment                  :  ";
5380 INPUT "",INCREMENT
5390 LOCATE 16,1.0
5400 PRINT TAB(1Ø) "Effective system resistance (k" CHR$(234) "):  ";
5410 INPUT "".RESISTANCE
5420 :
5430 LOCATE 18.1.1                        :REM  Set up for data storage
5440 PRINT TAB(5) "Insert the data disk in drtve B then press RETURN.  ";
5450 X$=INKEYS
5460   IF X$=CHR$(13) GOTO 5490
5470 GOTO 5450
```

```
5480 :
5490 LOCATE 11,5,Ø:PRINT "Given":GOTO 5500
5500 LOCATE 18,1,Ø:PRINT TAB(8Ø)        :REM  Set up for screen output of data
5510 LOCATE 18,1,Ø
5520 PRINT TAB(5) 'Output voltage:  ";:POSOUT=POS(X)
5530 PRINT TAB(53) "Input voltage:   ";:POSIN=POS(X)
5540 LOCATE 2Ø,1,Ø:PRINT TAB(5) "Working ...  "
5550 :
5560 OPEN "B:CALVOLT.DAT" FOR OUTPUT AS #1
5570 OPEN "B:CALVOLT.ERR" FOR OUTPUT AS #2
5580 :
5590 PRINT #1, USING "####';RESISTANCE
5600 PRINT #1,RANGE$
5610 PRINT #2, USING "####";RESISTANCE
5620 PRINT #2,RANGE$
5630
5640 MAXERROR=Ø:MINERROR=Ø:OLDDLOW=5ØØ:CVOLTAGE=MINVOLTS-1
5650
5660 FOR VOLTAGE=MINVOLTS TO MAXVOLTS STEP INCREMENT
5670   IF VOLTAGE<1Ø AND VOLTAGE =>-1Ø THEN FORMATS="##.####'
5680   IF VOLTAGE<=CVOLTAGE GOTO 5960
5690   DVOLTS=FIX(VOLTAGE/DIVISOR)+2Ø48
5700   DHIGH=FIX(DVOLTS/256)
5710   DLOW=DVOLTS-(DHIGH*256)
5720   IF DLOW=OLDDLOW GOTO 5960
5730   DRES=DLOW+DHIGH*256
5740   IF RANGE$="Ø to +5" OR RANGE$="Ø to +1Ø" 60TO 5760
5750   CVOLTAGE=(DRES-2Ø48)*DIVISOR:GOTO 5770
5760   CVOLTAGE=DRES*DIVISOR
5770   LOCATE 18,POSOUT.Ø:PRINT USING FORMAT$:CVOLTAGE
578Ø   OLDDLOW=DLOW
5790   POKE CMDA,2*CHANNELOUT:POKE CMDB,DLOW
5800   POKE CMDA,2*CHANNELOUT+1:POKE CMDB,DHIGH
5810   POKE STROBE.l
5820   POKE &H81,1
5830   POKE &H8Ø,CHANNELIN
5840   POKE &H9B,255
5850   IF PEEK(&H9B)<>127 60TO 5850
5860   DLOW=PEEK(&H8Ø)
5870   DHIGH=PEEK(&H81)
5880   DRES=DLOW+(DHIGH-24Ø)*256
5890   DVOLTS=DRES*DIVISOR-OFFSET
5900   VALERROR=DVOLTS-CVOLTAGE
5910   IF VALERROR>MAXERROR THEN MAXERROR=VALERROR
5920   IF VALERROR<MINERROR THEN MINERROR-VALERROR
5930   PRINT #1, USING "###.#### ###.####";CVOLTAGE,DVOLTS
5940   PRINT #2, USING "###.#### ##.#####";CVOLTAGE,VALERROR
5950   LOCATE 18,POSIN,Ø:PRINT USING FORMAT$;DVOLTS
5960 NEXT VOLTAGE
5970 :
5980 PRINT #2, USING "##.####X   ##.#####";MAXERROR,MINERROR
5990 CLOSE #1:CLOSE #2
6000 GOSUB 517Ø:LOCATE 18,1,Ø:PRINT TAB(8Ø)
6010 LOCATE 2Ø,1,Ø:PRINT TAB(8Ø):GOTO 4940
6Ø2Ø :
6Ø3Ø REM  ***********************************************************************
6Ø4Ø REM  OPTION 1:  Proceed With Calibrating The Designated Card/Channel
6Ø5Ø REM  ***********************************************************************
6Ø6Ø :
6Ø7Ø SPACEBAR=Ø
6Ø8Ø VOLTS$=LOW$
6Ø9Ø :
61ØØ GOSUB 96ØØ
611Ø :
```

```
6120 POKE CMDA,2*CHANNELOUT:POKE CMDB,Ø :REM  Write minimum voltage limit
6130 POKE CMDA,2*CHANNELOUT+1:POKE CMDB,Ø
6140 POKE STROBE,1                        :REM  Update output voltage on AOMI
6150 :
616Ø ADJUST$="OFFSET"
6170 IF DMM=1 THEN DISPLAY$="DMM" ELSE DISPLAY$="screen"
6180 :
6190 LOCATE 23,1,Ø
6200 PRINT TAB(5) "Press the SPACE bar to test a different input voltage.  ";
6210 ROW=23:POSITION=POS(X)
6220 IF DMM=1 THEN LOCATE ,,1
6230 IF SPACEBAR=1 GOTO 6340
6240 :
6250 IF VOLTS$=LOW$ THEN LOCATE 24,1,Ø:PRINT TAB(8Ø);:GOTO 6290
6260 LOCATE 24,1,Ø
6270 PRINT TAB(5) "Press ESC to restart or repeat the calibration. ";
6280 :
6290 LOCATE 9,1,1                         :REM  Print calibration message on
6300 BEEP                                 :REM    screen
6310 PRINT TAB(5) "Adjust channel" CHANNEL ADJUST$ " for " VOLTS$;
6320 PRINT " volts on the " DISPLAY$ " then press RETURN.  ";
6330 ROW=9:POSITION=POS(X)
6340 X$=INKEY$
6350   IF X$=CHR$(13) THEN LOCATE 9,1,Ø:PRINT TAB(8Ø):IF CHOICE=3 GOTO 6610
6360   IF X$=CHR$(27) THEN IF SPACEBAR=Ø AND VOLTS$=HIGH$ THEN GOTO 6070
6370   IF X$=CHR$(32) THEN SPACEBAR=1:VOLTS$=HIGH$:GOTO 6860
6380   IF DMM=1 GOTO 6340
6390   POKE &H8Ø,CHANNELIN                :REM  Select channel on AMMI card
6400   POKE &H81,1                        :REM  Select slot 1 for AMMI card
6410   POKE &H9B,255                      :REM  Start A/D conversion
6420   IF PEEK(&H9B)<>127 GOTO 6420       :REM  Loop if conversion incomplete
6430   DLOW=PEEK(&H8Ø)                    :REM  Read low byte
6440   DHIGH=PEEK(&H81)                   :REM  Read high byte
6450   DRES=DLOW+(DHIGH-24Ø)*256          :REM  Calculate resolution of datum
6460   DVOLTS=DRES*(2Ø/4Ø96)-1Ø           :REM    and convert to volts
6470   LOCATE 11,69,Ø                     :REM  Update input voltage on screen
6480   IF DVOLTS<1Ø THEN FORMAT$="##.####" ELSE FORMAT$="###.####"
6490   PRINT USING FORMAT$;DVOLTS
6500   IF DVOLTS>Ø THEN LOCATE 11,69,Ø:PRINT "+"
6510   LOCATE 13,POSINHIGH,Ø:PRINT USING "###";DHIGH-24Ø
6520   LOCATE 14,POSINLOW,Ø:PRINT USING "###";DLOW
6530   LOCATE ROW,POSITION,1
654Ø   FOR I=1 TO 1ØØØ:NEXT I
655Ø GOTO 634Ø
656Ø :
657Ø REM  ****************************************************************
658Ø REM  INTERNAL SUBROUTINE:  Screen Cleanup
659Ø REM  ****************************************************************
66ØØ :
661Ø IF VOLTS$=LOW$ GOTO 672Ø
662Ø LOCATE 11,1,Ø:PRINT TAB(8Ø)
663Ø LOCATE 13,1,Ø:PRINT TAB(8Ø)
664Ø LOCATE 14,1,Ø:PRINT TAB(8Ø)
665Ø LOCATE 23,1,Ø:PRINT TAB(8Ø)
666Ø LOCATE 24,1,Ø:PRINT TAB(8Ø);:GOTO 717Ø
667Ø :
668Ø REM  ****************************************************************
669Ø REM  CONTINUE WITH HIGH SIDE AOM1 VOLTAGE CALIBRATION
67ØØ REM  ****************************************************************
671Ø :
672Ø VOLTS$=HIGH$:ADJUST$="GAIN"         :REM  Change variables for next
673Ø HIGHBYTE=15:LOWBYTE=255
674Ø POKE CMDA,2*CHANNEL:POKE CMDB,255   :REM    adjustment and write maximum
675Ø POKE CMDA,2*CHANNEL+1:POKE CMDB,15  :REM    voltage
```

```
6760 POKE STROBE,1                     :REM  Update output voltage on AOM1
6770 LOCATE 11,22,0:PRINT VOLTS$       :REM  Update required adjustment voltage
6780 IF DMM=1 GOTO 6260
6790 LOCATE 13,POSOUTHIGH,0:PRINT USING "###";HIGHBYTE
6800 LOCATE 14,POSOUTLOW,0:PRINT USING "###";LOWBYTE:GOTO 6260
6810 :
6820 REM  ********************************************************************
6830 REM  OPTION 2:  Test An Output Voltage From A Keyboard Voltage Input
6840 REM  ********************************************************************
6850 :
6860 LOCATE 9,1,0:PRINT TAB(80)         :REM  Clear boilerplate for new data
6870 LOCATE 11,22,0:PRINT "             "
6880 LOCATE 11,69,0:PRINT TAB(80)
6890 LOCATE 13,POSOUTHIGH,0:PRINT "   "
6900 LOCATE 13,POSINHIGH,0:PRINT "   "
6910 LOCATE 14,POSOUTLOW,0:PRINT "   "
6920 LOCATE 14,POSINLOW,0:PRINT "   "
6930 LOCATE 16,POSOUTLOW,0:PRINT TAB(80)
6940 LOCATE 23,1,0:PRINT TAB(80)
6950 LOCATE 24,1,0:PRINT TAB(80);
6960
6970 LOCATE 11,22,0                     :REM  Get output voltage
6980 INPUT "",VOLTAGE
6990 IF RANGE$="0 to +5" OR RANGE$="0 to +10" GOTO 7010
7000 DVOLTS=FIX(VOLTAGE/DIVISOR)+2048:GOTO 7020
7010 DVOLTS=FIX(VOLTAGE/DIVISOR)
7020 DHIGH=FIX(DVOLTS/256)
7030 DLOW=DVOLTS-(DHIGH*256)
7040 CVOLTAGE=((DHIGH*256)+DLOW)*DIVISOR-OFFSET
7050 POKE CMDA,2*CHANNELOUT:POKE CMDB,DLOW
7060 POKE CMDA,2*CHANNELOUT+1:POKE CMDB,DHIGH
7070 POKE STROBE,1
7080 LOCATE 13,POSOUTHIGH,0:PRINT USING "###";DHIGH
7090 LOCATE 14,POSOUTLOW,0:PRINT USING "###";DLOW
7100 IF SELECTION<>1 GOTO 7140
7110 IF CVOLTAGE<10 THEN FORMAT$="##.####" ELSE FORMAT$="###.####"
7120 LOCATE 16,POSOUTLOW,0
7130 PRINT USING FORMAT$;CVOLTAGE
7140 LOCATE 24,1,0
7150 PRINT TAB(5) "Press RETURN to continue or ESC to quit.";:GOTO 6190
7160 :
7170 LOCATE 21,1,1                      :REM  Option to adjust another card
7180 BEEP                               :REM     or channel
7190 PRINT TAB(5) "Do you wish to calibrate another card/channel?  ";
7200 X$=INKEY$
7210   IF X$="y" OR X$="Y" GOTO 1190
7220   IF X$="n" OR X$="N" THEN LOCATE 21,1,0:PRINT TAB(80):GOTO 4940
7230 GOTO 7200
7240 :
7250 LOCATE 21,1,0:PRINT TAB(80)
7260 LOCATE 21,1,0                      :REM  Terminate program
7270 PRINT TAB(5) "Program terminated ..."
7280 FOR I=1 TO 4000:NEXT I:CHAIN "HEAD500.EXE"
7290 :
7300 END                                :REM  Program end
7310 :
9000 REM  ****************************************************************
9010 REM  SUBROUTINE:   Print The Program Header
9020 REM  ****************************************************************
9030 :
9040 WIDTH 80:SCREEN 0,0,0:LOCATE 1,1,0:CLS:KEY OFF
9050 :
9060 UL$=CHR$(201):UR$=CHR$(187):LR$=CHR$(188):LL$=CHR$(200)
9070 H$=CHR$(205):V$=CHR$(186):HS$=CHR$(196)
```

```
9080 :
9090 PRINT UL$H$H$H$H$H$H$H$H$H$H$H$H$H$H$H$H$H$H$H$H$H$H$H$H$H$H$H$H$;
9100 PRINT H$H$H$H$H$H$H$H$H$H$H$H$H$H$H$H$H$H$H$H$H$H$H$H$H$H$H$H$H$;
9110 PRINT H$H$H$H$H$H$H$H$H$H$H$H$H$H$H$H$H$H$H$H$UR$V$ TAB(8Ø) V$;
9120 PRINT LL$H$H$H$H$H$H$H$H$H$H$H$H$H$H$H$H$H$H$H$H$H$H$H$H$H$H$H$H$;
9130 PRINT H$H$H$H$H$H$H$H$H$H$H$H$H$H$H$H$H$H$H$H$H$H$H$H$H$H$H$H$H$;
9140 PRINT H$H$H$H$H$H$H$H$H$H$H$H$H$H$H$H$H$H$H$H$LR$
9150 :
9160 LOCATE 2,4,Ø
9170 PRINT "TEST5ØØ Calibration And Test Programs For KEITHLEY/das ";
9180 PRINT "Series 5ØØ Hardware"
9190 :
9200 RETURN
9210 :
9220 REM  ********************************************************************
9230 REM  SUBROUTINE:  Inkey Function For Slots 2-1Ø
9240 REM  ********************************************************************
9250 :
9260 X$=INKEY$
9270   IF X$="2" THEN CMDA=&H82:CMDB=&H83:RETURN
9280   IF X$="3" THEN CMDA=&H84:CMDB=&H85:RETURN
9290   IF X$="4" THEN CMDA=&H86:CMDB=&H87:RETURN
9300   IF X$="5" THEN CMDA=&H88:CMDB=&H89:RETURN
9310   IF X$="6" THEN CMDA=&H8A:CMDB=&H8B:RETURN
9320   IF X$="7" THEN CMDA=&H8C:CMDB=&H8D:RETURN
9330   IF X$="8" THEN CMDA=&H8E:CMDB=&H8F:RETURN
9340   IF X$="9" THEN CMDA=&H9Ø:CMDB=&H91:RETURN
9350   IF X$="1" THEN X$="1Ø":CMDA=&H92:CMDB=&H93:RETURN
9360   IF X$="x" OR X$="X" GOTO 725Ø
9370 GOTO 926Ø
9380 :
9390 RETURN
9400 :
9410 REM  ****************************************************************
9420 REM  SUBROUTINES:  Voltage Range Parameters
9430 REM  ****************************************************************
9440 :
9450 LOWS="Ø.ØØØ":HIGH$="+4.999":RANGE$="Ø to +5":DIVISOR=1.22Ø7Ø3E-Ø3
9460   OFFSET=Ø:RETURN
9471 LOW$="Ø.ØØØ":HIGH$="+9.996":RANGE$="Ø to +1Ø":DIVISOR=2.4414Ø6E-Ø3
9480   OFFSET=Ø:RETURN
9490 LOW$="-2.5ØØ":HIGH$="+2.499":RANGE$="+/- 2.5":DIVISOR=1.22Ø7Ø3E-Ø3
9500   OFFSET=2.5:RETURN
9510 LOW$="-5.ØØØ":HIGH$="+4.996":RANGE$="+/- 5":DIVISOR=2.4414Ø6E-Ø3
9520   OFFSET=5:RETURN
9530 LOW$="-1Ø.ØØØ":HIGH$="+9.992":RANGE$="+/- 1Ø":DIVISOR=4.882813E-Ø3
9540   OFFSET=1Ø:RETURN
9550 :
9560 REM  ********************************************************************
9570 REM  SUBROUTINE:  Print Output Boilerplate On The Screen
9580 REM  ********************************************************************
9590 .
9600 LOCATE 11,1,Ø                       :REM  Update Input voltage on screen
9610 PRINT TAB(5) "Output voltage:  " VOLTS$;
9620 IF DMM=1 THEN RETURN
9630 PRINT TAB(53) "Input voltage:" TAB(8Ø)
9640 LOCATE 13,1,Ø
9650 PRINT TAB(1Ø) "high byte:  ";:POSOUTHIGH=POS(X):PRINT "  O";
9660 PRINT TAB(57) "high byte:  ";:POSINHIGH=POS(X)
9670 LOCATE 14,1,Ø
9680 PRINT TAB(11) "low byte:  ";:POSOUTLOW=POS(X):PRINT "  O";
9690 PRINT TAB(58) "low byte:  ";:POSINLOW=POS(X)
9700 .
9710 LOCATE 16,1,Ø
```

```
9720 PRINT "Resolution voltage:"
9730 :
9740 RETURN
9750 :
9760 REM  ********************************************************************
9770 REM  SUBROUTINE:  Read the Interval Timer
9780 REM  ********************************************************************
9790 :
9800 REM  Latch the Interval Counters
9810 :
9820 POKE &HC3,Ø                           :REM  Latches counter zero
9830 POKE &HC3,64                          :REM  Latches counter one
9848 :
9850 LZ=PEEK(&HCØ)                         :REM  Reads low byte
9860 HZ=PEEK(&HCØ)                         :REM  Reads high byte
9870 CCZ=LZ+(HZ*256)                       :REM  Reconstitutes counter zero
9880 :
9890 LO=PEEK(&HC1)                         :REM  Reads low byte
9900 HO=PEEK(&HC1)                         :REM  Reads high byte
9910 CCO=LO+(HO*256)                       :REM  Reconstitutes counter one
9920 :
9930 REM  Convert Current Count to Elapsed Time
9940 :
9950 CURRENTCOUNT#=(((CZ-CCZ)*CO)+(CO-CCO))/2
9960                                       :REM  Current count in clock pulses
9970 RETURN
```

```
C ***********************************************************************
C
C       PROGRAM TITLE:   FIT500
C               Data Analysis Program for FIX5ØØ.DAT Data Output Files
C               Created by SCAN500 Data Acquisition System
C
C       Written by Roland N. Horne
C       Modified for the KEITHLEY/das System by L. W. Bouett
C               Stanford University, 22 April 1986
C
C ***********************************************************************
C ***********************************************************************
C
C
      IMPLICIT REAL*8(A-B,D-H,O-Z)
C
C
C       SET DIMENSIONS FOR VARPRO. BE CAREFUL WHEN SETTING THE
C       DIMENSIONS FOR THE INCIDENCE MATRIX INC. SEE NOTE.
C
      DIMENSION Y(5ØØ),T(5ØØ),TIN(5ØØ),ALF(3),BETA(2),W(5ØØ),A(5ØØ,7),
     *INC(12,8),NELTRY(96),NGO(96)
      COMMON XPOS,TZERO
      COMMON /DUMMY/C(5ØØ,6)
      EXTERNAL ADA
      CHARACTER*6 CFILE(96)
      DATA CFILE/'1.dat','2.dat','3.dat','4.dat','5.dat','6.dat',
     @ '7.dat','8.dat','9.dat','1Ø.dat','11.dat','12.dat','13.dat',
     1 '14.dat','15.dat','16.dat','17.dat','18.dat','19.dat','2Ø.dat',
     @ '21.dat','22.dat','23.dat','24.dat','25.dat','26.dat','27.dat',
     2 '28.dat','29.dat','3Ø.dat','31.dat','32.dat','33.dat','34.dat',
     @ '35.dat','36.dat','37.dat','38.dat','39.dat','4Ø.dat','41.dat',
     3 '42.dat','43.dat','44.dat','45.dat','46.dat','47.dat','48.dat',
     @ '49.dat','5Ø.dat','51.dat','52.dat','53.dat','54.dat','55.dat',
     4 '56.dat','57.dat','58.dat','59.dat','6Ø.dat','61.dat','62.dat',
     @ '63.dat','64.dat','65.dat','66.dat','67.dat','68.dat','69.dat',
     5 '7Ø.dat','71.dat','72.dat','73.dat','74.dat','75.dat','76.dat',
     @ '77.dat','78.dat','79.dat','8Ø.dat','81.dat','82.dat','83.dat',
     6 '84.dat','85.dat','86.dat','87.dat','88.dat','89.dat','9Ø.dat',
     @ '91.dat','92.dat','93.dat','94.dat','95.dat','96.dat'/
C
C          SET PARAMETERS FOR VARPRO.
C
C
      IPLOT-0
      NMAX = 500
      IPRINT=5Ø
C
C
C          READ DATA SEQUENTIAL ORDERING AND
C          PROPER FORMATTING ARE IMPORTANT.
C
C
C
C        NL IS THE NUMBER OF NONLINEAR PARAMETERS
C
C
      READ (5,311) NL
 311  FORMAT (I1)
      WRITE(6,12) NL
 12   FORMAT (1HØ,1ØX,'NUMBER OF NONLINEAR PARAMETERS'//(I3))
C
C
C        L IS THE NUMBER OF LINEAR PARAMETERS
```

```
C
C
      L=2
C
C
C       ESTIMATES OF THE NONLINEAR PARAMETERS
C
C
      POSØ=14.9225
      READ (5,31Ø) VELOC,DIFFS,TZERO,DCØ,DC1
 310  FORMAT (F1Ø.4)
      READ (5,312) NTRIES,(NELTRY(K),K=1,NTRIES)
 312  FORMAT (4ØI2)
      DO 314 K=1,96
 314  NGO(K)=1.
      DO 313 K=1,NTRIES
 313  NGO(NELTRY(K))=Ø.
      WRITE (6,2Ø) VELOC,DIFFS,TZERO
 20   FORMAT (/,'Ø Mean Velocity(cm/sec)   Diffusivity (cm2/sec)'
     # '  Time zero (sec)',/,(5X,F9.5,1ØX,F13.3,1ØX,F1Ø.3))
C
C
      LPP2=L+NL+2
C
C
C
C       N IS THE NUMBER OF OBSERVATIONS
C       IV IS THE NUMBER OF INDEPENDENT VARIABLES T
C
C
      IV=1
C
C
C
C
C     T IS THE INDEPENDENT VARIABLE
C     Y IS THE N-VECTOR OF OBSERVATIONS
C
C
c
       OPEN(1,FILE='fix5ØØ.dat',STATUS='OLD',ACCESS='SEQUENTIAL')
       open(7,file='results',status='new',access='sequential')
       open(8,file='eta.dat',status='new',access='sequential')
       open(9,file='u.dat',status='new',access='sequential')
       open(1Ø,file='etacalc.dat',status='new',access='sequential')
c
      WRITE (7,323)
 323  FORMAT(2X,'Elec       CO         Cinj        Velocity  ',
     1 '    Disp         To',/,'                 ppm          ppm     ',
     2 '   cm/sec      cm2/sec       sec')
c
       READ(1,*) NIN
501    FORMAT(I3)
c
       write (8,5Ø1) ntries
       write (9,5Ø1) ntries
       write (1Ø,5Ø1) ntries
c
       DO 601  I=1,NIN
               READ(1,*) TIN(I)
601    CONTINUE
502    FORMAT(F7.3)
      ALF(1)=VELOC
      ALF(2)=DIFFS
```

```
      ALF(3)=TZERO
      DO 200 NELEC=1, 96
       DO 602 I=1, NIN
              READ(1,*) Y(I)
C         WRITE (6,*) T(I),Y(I)
602    CONTINUE
503    FORMAT(F6.4)
      IF (NGO(NELEC).GT.Ø) GO TO 200
      WRITE (6,32Ø) NELEC
 320  FORMAT (//,2X,'Electrode number ',I3,'******************')
      XPOS=POSØ+5.Ø8*(NELEC-1)
       if (nelec.gt.32.and.nelec.lt.65) xpos=posØ+5.Ø8*(nelec-33)
       if (nelec.gt.64) xpos=posØ+5.Ø8*(nelec-65)
      ALF(1)=VELOC
      ALF(2)=DIFFS
      ALF(3)=TZERO
C
C     W(I) ARE THE WEIGHTING PARAMETERS
C
C
        DO 1 I=1,NIN
 1       W(I)=1.Ø
C
          IMINP=1
          IMAXP=NIN
         IF(NIN.LT.200) THEN
              IMIN=5
              GOTO 702
         ENDIF
         TMINP=(TZERO+XPOS/VELOC)-12Ø.Ø
         TMAXP=TMINP+13Ø.Ø
         DO 70 I=1,NIN
         IF (TIN(I).LT.TMINP) IMINP=I
         IF (TIN(I).LT.TMAXP) IMAXP=I
 70      CONTINUE
         IMIN=IMINP
 702     N=IMAXP-IMIN
         IF (N.GT.NIN) N=NIN
         DO 201 I=1,N
         T(I)=TIN(IMIN+I)
         Y(I)=Y(IMIN+I)
 201     CONTINUE
 701     CALL VARPRO(L,NL,N,NMAX,LPP2,IV,T,Y,W,ADA,A,
     *IPRINT,ALF,BETA,IERR,1Ø)
C
      VØ=BETA(1)
      V1=BETA(1)+2.*BETA(2)
      BPAR=(V1-VØ)/DLOG(DCØ/DC1)
      APAR=V1+BPAR*DLOG(DC1)
      DO 211 I=1,N
 211  Y(I)=DEXP((APAR-Y(I))/BPAR)
         CALL VARPRO(L,NL,N,NMAX,LPP2,IV,T,Y,W,ADA,A,
     *IPRINT,ALF,BETA,IERR,1ØØ)
C
      LP1=L+1
      CALL ADA (LP1,NL,N,NMAX,LPP2,IV,A,INC,T,ALF,1)
      DO 8 I=1,N
      C(I,LP1)=Ø.
      DO 9 J=1,L
      C(I,J)=BETA(J)*A(I,J)
 9    C(I,LP1)=C(I,LP1)+C(I,J)
 8    CONTINUE
      OPEN (2,FILE=CFILE(NELEC),STATUS='NEW',ACCESS='SEQUENTIAL')
      WRITE (2,3Ø5) N
```

```
      DO 2Ø5 I=1,N
 205  WRITE (2,14) T(I),Y(I)
      WRITE (2,3Ø5) N
      DO 206 I=1,N
 206  WRITE (2,14) T(I),C(I,LP1)
 14   FORMAT (1X,8F1Ø.4)
 305  FORMAT (I6)
C
      CLOSE (2)
      DCMID=BETA(1)+BETA(2)
      DO 400 II=1,N
      IF (DCØ.LT.DC1) THEN
          IF (C(II,LP1).GT.SNGL(DCMID)) GO TO 401
          ELSE
          IF (C(II,LP1).LT.SNGL(DCMID)) GOTO 401
      ENDIF
      TMID=T(II)
 4ØØ  CONTINUE
 401  TØCALC=TMID-XPOS/ALF(1)
       ETACALC=.935695Ø33dØ*ALF(1)**2
      WRITE (7,3Ø2) NELEC,BETA(1),BETA(1)+2.*BETA(2),ALF(1),ALF(2)
     1 ,TØCALC
       write (8,3Ø1) NELEC,ALF(2)
       write (9,3Ø1) NELEC,ALF(1)
       write (1Ø,3Ø1) NELEC,ETACALC
301    format (2X,I4,F12.4)
 302  FORMAT (2X,I4,5F12.4)
 200  CONTINUE
       STOP
       END
C
C
C *********************************************************************
C *********************************************************************
C
C                      SUBROUTINES
C
C *********************************************************************
C *********************************************************************
C
C
C
C
      SUBROUTINE ADA (LP,NL,N,NMAX,LPP2,IV,A,INC,T,ALF,ISEL)
      IMPLICIT REAL*8(A-H,O-Z)
      DIMENSION ALF(NL),A(NMAX,LPP2),T(NMAX),INC(12,8)
      COMMON XPOS,TZERO
C
      PI=4.*ATAN(1.)
      TPI=2./DSQRT(PI)
      L=LP-1
      IF (NL.LT.3) ALF(3)=TZERO
C
C
C      THE INCIDENCE MATRIX INC(NL,L+1) IS FORMED BY SETTING
C      INC(K,J)=1 IF THE NONLINEAR PARAMETER ALF(K) APPEARS
C      IN THE J-TH FUNCTION PHI(J). (THE PROGRAM SETS ALL OTHER
C      INC(K,J) TO ZERO.)
C
C
       IF(ISEL.EQ.2)  GO TO 90
       IF(ISEL.EQ.3)  GO TO 165
C
       INC(1,1)=Ø
```

```
      INC(2,1)=Ø
      INC(1,2)=1
      INC(2,2)=1
      IF (NL.EQ.3) INC(3,2)=1
C
C
C
C       THE VECTOR-SAMPLED FUNCTIONS PHI(J) ARE STORED IN
C       THE FIRST N ROWS AND FIRST L COLUMNS OF THE MATRIX
C       B(I,J). B(I,J) CONTAINS PHI(J,ALF;T(I),I,...N;
C       J=1,L. THE CONSTANT FUNCTIONS PHI WHICH DO NOT
C       DEPEND UPON ANY NONLINEAR PARAMETERS ALF MUST
C       APPEAR FIRST.
C
C
C
C
C
C
 90         alf(2)=dabs(alf(2))
            DO 81 I=1,N
            IF (ISEL.EQ.1) A(I,1)=1.Ø
            A(I,2)=Ø.
            TTI=T(I)-ALF(3)
            IF (TTI.LE.Ø.) GO TO 81
            TFUNC=DSQRT(ALF(2)*TTI)
            A(I,2)=DERFC((XPOS-ALF(1)*TTI)/(2.*TFUNC))
      IF (XPOS*ALF(1)/ALF(2).LT.2Ø.) A(I,2)=A(I,2)+DEXP(ALF
     1     (1)*XPOS/ALF(2))*DERFC((XPOS+ALF(1)*TTI)/(2.*TFUNC))
C
  81        CONTINUE
C
C
C
       IF (ISEL.EQ.2) GO TO 2ØØ
C
C
C
C
C
C
C
 165      DO 170 I=1,N
C
          A(I,4)=Ø.
          A(I,5)=Ø.
          A(I,6)=Ø.
          TTI=T(I)-ALF(3)
          IF (TTI.LE.Ø.) GO TO 170
          E1=DEXP(-((XPOS-ALF(1)*TTI)**2)/(4.*ALF(2)*TTI))
          E2=DEXP(-((XPOS+ALF(1)*TTI)**2)/(4.*ALF(2)*TTI))
          E3=Ø.
         IF (ALF(1)*XPOS/ALF(2).LT.2Ø.) E3=DEXP(ALF(1)*XPOS/ALF(2))
          ERF1=DERFC((XPOS+ALF(1)*TTI)/(2.*DSQRT(ALF(2)*TTI)))
          FUNC=2.*DSQRT(TTI*ALF(2)**3)
          A(I,4)=-TPI*E1*(-TTI/(2.*DSQRT(ALF(2)*TTI)))
          IF (E3.GT.Ø.) A(I,4)=A(I,4)+(XPOS/ALF
     1      (2))*E3*ERF1+E3*(-TPI*E2)*TTI/(2.*DSQRT(ALF(2)*TTI))
          A(I,5)=TPI*E1*(XPOS-ALF(1)*TTI)/FUNC
          IF (E3.GT.Ø.) A(I,5)=A(I,5)-E3*ERF1*ALF(1)*XPOS/
     1     ALF(2)**2+E3*TPI*E2*(XPOS+ALF(1)*TTI)/FUNC
          IF (NL.LT.3) GO TO 170
          A(I,6)=-TPI*E1*(XPOS/(2.*DSQRT(ALF(2)*TTI**3))+ALF(1)/
     1      FUNC)
```

```
      IF (E3.GT.Ø.) A(I,6)=A(I,6)+E3*(-TPI*E2)
     2   *(XPOS/(2.*DSQRT(ALF(1)*TTI**3))-ALF(1)/FUNC)
C
C
 170          CONTINUE
C
 200      CONTINUE
C
      RETURN
      END
```

```
C ***********************************************************************
C
C       PROGRAM TITLE:  FIT600
C               Data Analysis Program for FIX5ØØ.DAT Data Output Files
C               Created by SCAN500 Data Acquisition System
C
C       Written by Roland N. Horne
C       Modified for the KEITHLEY/das System by L. W. Bouett
C               Stanford University, 22 April 1986
C
C       Modified again by RNH to analyse derivatives wrt time 5/23/86
C
C
C ***********************************************************************
C ***********************************************************************
C
C
      IMPLICIT REAL*8(A-B,D-H,O-Z)
C
C
C       SET DIMENSIONS FOR VARPRO. BE CAREFUL WHEN SETTING THE
C       DIMENSIONS FOR THE INCIDENCE MATRIX INC. SEE NOTE.
C
      DIMENSION Y(5ØØ),T(5ØØ),TIN(5ØØ),ALF(3),BETA(2),W(5ØØ),A(5ØØ,7),
     *INC(12,8),NELTRY(96),NGO(96),YIN(5ØØ)
      COMMON XPOS,TZERO
      COMMON /DUMMY/C(5ØØ,6)
      EXTERNAL ADA,ADADRV
      CHARACTER*6 CFILE(96)
      DATA CFILE/'1.dat','2.dat','3.dat','4.dat','5.dat','6.dat',
     @ '7.dat','8.dat','9.dat','1Ø.dat','11.dat','12.dat','13.dat',
     1 '14.dat','15.dat','16.dat','17.dat','18.dat','19.dat','2Ø.dat',
     @ '21.dat','22.dat','23.dat','24.dat','25.dat','26.dat','27.dat',
     2 '28.dat','29.dat','3Ø.dat','31.dat','32.dat','33.dat','34.dat',
     @ '35.dat','36.dat','37.dat','38.dat','39.dat','4Ø.dat','41.dat',
     3 '42.dat','43.dat','44.dat','45.dat','46.dat','47.dat','48.dat',
     @ '49.dat','5Ø.dat','51.dat','52.dat','53.dat','54.dat','55.dat',
     4 '56.dat','57.dat','58.dat','59.dat','6Ø.dat','61.dat','62.dat',
     @ '63.dat','64.dat','65.dat','66.dat','67.dat','68.dat','69.dat',
     5 '7Ø.dat','71.dat','72.dat','73.dat','74.dat','75.dat','76.dat',
     @ '77.dat','78.dat','79.dat','8Ø.dat','81.dat','82.dat','83.dat',
     6 '84.dat','85.dat','86.dat','87.dat','88.dat','89.dat','9Ø.dat',
     @ '91.dat','92.dat','93.dat','94.dat','95.dat','96.dat'/
C
C          SET PARAMETERS FOR VARPRO.
C
C
      IPLOT=Ø
      NMAX = 500
      IPRINT=5Ø
C
C
C          READ DATA SEQUENTIAL ORDERING AND
C          PROPER FORMATTING ARE IMPORTANT.
C
C
C
C       NL IS THE NUMBER OF NONLINEAR PARAMETERS
C
C
      READ (5,311) NL
 311  FORMAT (I1)
      WRITE(6,12) NL
 12   FORMAT (1HØ,1ØX,'NUMBER OF NONLINEAR PARAMETERS'//(I3))
```

```
C
C
C        L IS THE NUMBER OF LINEAR PARAMETERS
C
C
      L=2
C
C
C        ESTIMATES OF THE NONLINEAR PARAMETERS
C
C
      POSØ=14.9225
      READ (5,31Ø) VELOC,DIFFS,TZERO,DCØ,DC1
 31Ø  FORMAT (F1Ø.4)
      READ (5,312) NTRIES,(NELTRY(K),K=1,NTRIES)
 312  FORMAT (4ØI2)
      DO 314 K=1,96
 314  NGO(K)=1.
      DO 313 K=1,NTRIES
 313  NGO(NELTRY(K))=Ø.
      WRITE (6,2Ø) VELOC,DIFFS,TZERO
 2Ø   FORMAT (/,'Ø Mean Velocity(cm/sec)   Diffusivlty (cm2/sec)'
     # ,'  Time zero (sec)',/,(5X,F9.5,1ØX,F13.3,1ØX,F1Ø.3))
C
C
      LPP2=L+NL+2
C
C
C
C        N IS THE NUMBER OF OBSERVATIONS
C        IV IS THE NUMBER OF INDEPENDENT VARIABLES T
C
C
      IV=1
C
C
C
C
C       T IS THE INDEPENDENT VARIABLE
C       Y IS THE N-VECTOR OF OBSERVATIONS
C
C
       OPEN(1,FILE='fix5ØØ.dat',STATUS='OLD',ACCESS='SEQUENTIAL')
       OPEN(7,FILE='results',STATUS='new',ACCESS='SEQUENTIAL')
       OPEN(8,FILE='eta.dat',STATUS='new',ACCESS='SEQUENTIAL')
       OPEN(9,FILE='u.dat',STATUS='new',ACCESS='SEQUENTIAL')
C
      WRITE (7,323)
 323  FORMAT(2X,'Elec       co        Cin.1      Velocity  '.
     1 '     Disp        To',/,'              ppm          ppm    ',
     2 '  cm/sec      cm2/sec        sec')
c
       READ(1,*) NIN
501    FORMAT(I3)
c
       write (8,5Ø1) ntrles
       write (9,5Ø1) ntries
c
       DO 601  I=1,NIN
               READ(1,*) TIN(I)
601    CONTINUE
502    FORMAT(F7.3)
      ALF(1)=VELOC
      ALF(2)=DIFFS
```

```
       ALF(3)=TZERO
       DO 2ØØ NELEC=1, 96
        DO 602 I=1, NIN
               READ(1,*) YIN(I)
               Y(I)=YIN(I)
C         WRITE (6,*) T(I),Y(I)
602     CONTINUE
503     FORMAT(F6.4)
       IF (NGO(NELEC).GT.Ø) GO TO 200
       WRITE (6,32Ø) NELEC
 328   FORMAT (//,2X,'Electrode number ',I3,'*****************')
       XPOS=POSØ+5.Ø8*(NELEC-1)
        if (nelec.gt.32.and.nelec.lt.65) xpos=posØ+5.Ø8*(nelec-33)
        if (nelec.gt.64) xpos=posØ+5.Ø8*(nelec-65)
       ALF(1)=VELOC
       ALF(2)=DIFFS
       ALF(3)=TZERO
C
C      W(I) ARE THE WEIGHTING PARAMETERS
C
C
          DO 1 I=1,NIN
 1         W(I)=1.Ø
C
            IMINP=5
            IMAXP=NIN
          IF(NIN.LT.2ØØ) THEN
               IMIN=5
               GOTO 702
          ENDIF
          TMINP=(TZERO+XPOS/VELOC)-12Ø.Ø
          TMAXP=TMINP+13Ø.Ø
          DO 70 I=1,NIN
          IF (TIN(I).LT.TMINP) IMINP-I
          IF (TIN(I).LT.TMAXP) IMAXP=I
 70       CONTINUE
          IMIN=IMINP
          IF (IMIN.LT.5) IMIN-5
 702      N=IMAXP-IMIN
          IF (N.GT.NIN) N=NIN
c         write (*,*) n,nin,imin,iminp,imaxp,tminp,tmaxp
          DO 201 I=1,N
          T(I)=TIN(IMIN+I)
          Y(I)=Y(IMIN+I)
c          write (*,'(i2,2g1Ø.4)') i,t(i),y(i)
 201      CONTINUE
 701      CALL VARPRO(L,NL,N,NMAX,LPP2,IV,T,Y,W,ADA,A,
     *IPRINT,ALF,BETA,IERR,1Ø)
C
       VØ=BETA(1)
       V1=BETA(1)+2.*BETA(2)
       BPAR=(V1-VØ)/DLOG(DCØ/DC1)
       APAR=V1+BPAR*DLOG(DC1)
       DO 211 I=1,N
 211   Y(I)=DEXP((APAR-Y(I))/BPAR)
       DO 212 I=1,NIN
 212   YIN(I)=DEXP((APAR-YIN(I))/BPAR)
C
        NSKIP=3
        CALL FIXUP(Y,YIN,T,TIN,N,NIN,NSKIP)
C
C
           CALL VARPRO(L,NL,N,NMAX,LPP2,IV,T,Y,W,ADADRV,A,
     *IPRINT,ALF,BETA,IERR,1ØØ)
```

```
C
      LP1=L+1
      CALL ADADRV (LP1,NL,N,NMAX,LPP2,IV,A,INC,T,ALF,1)
      DO 8 I=1,N
      C(I,LP1)=Ø.
      DO 9 J=1,L
      C(I,J)=BETA(J)*A(I,J)
 9    C(I,LP1)=C(I,LP1)+C(I,J)
 8    CONTINUE
      OPEN (2,FILE=CFILE(NELEC),STATUS='unknown',ACCESS='SEQUENTIAL')
      WRITE (2,3Ø5) N
      DO 205 I=1,N
 205  WRITE (2.14) T(I),Y(I)
      WRITE (2,3Ø5) N
      DO 206 I=1,N
 206  WRITE (2,14) T(I),C(I,LP1)
 14   FORMAT (1X,8F1Ø.4)
 305  FORMAT (I6)
C
      CLOSE (2)
      DCMID=BETA(1)+BETA(2)
      cmax=c(1,lp1)
      DO 400 II=1,N
      if (dcØ.lt.dc1) then
          if (c(ii,lp1).ge.cmax) then
              cmax=c(ii,lp1)
              TMID=T(II)
          endif
      else
          if (c(ii,lp1).le.cmax) then
              cmax=c(ii,lp1)
              TMID=T(II)
          end if
      end if
 400  CONTINUE
 401  TØCALC=TMID-XPOS/ALF(1)
      WRITE (7,3Ø2) NELEC,BETA(1),BETA(1)+2.*BETA(2),ALF(1),ALF(2)
     1 ,TØCALC
       write (8,3Ø1) nelec,alf(2)
       write (9,3Ø1) nelec,alf(1)
 301   format (2x,i4,f12.4)
 302  FORMAT (2X,I4,5F12.4)
 200  CONTINUE
       STOP
       END
C
C
C *******************************************************************
C *******************************************************************
C
C                    SUBROUTINES
C
C *******************************************************************
C *******************************************************************
C
C
C
C
      SUBROUTINE ADA (LP,NL,N,NMAX,LPP2,IV,A,INC,T,ALF,ISEL)
      IMPLICIT REAL*8(A-H,O-Z)
      DIMENSION ALF(NL),A(NMAX,LPP2),T(NMAX),INC(12,8)
      COMMON XPOS,TZERO
C
      PI=4.*ATAN(1.)
```

```
      TPI=2./DSQRT(PI)
      L=LP-1
      IF (NL.LT.3) ALF(3)=TZERO
C
C
C       THE INCIDENCE MATRIX INC(NL,L+1) IS FORMED BY SETTING
C       INC(K,J)=1 IF THE NONLINEAR PARAMETER ALF(K) APPEARS
C       IN THE J-TH FUNCTION PHI(J). (THE PROGRAM SETS ALL OTHER
C       INC(K,J) TO ZERO.)
C
C
      IF(ISEL.EQ.2) 60 TO 90
      IF(ISEL.EQ.3) GO TO 165
C
      INC(1,1)=Ø
      INC(2,1)=Ø
      INC(1,2)=1
      INC(2,2)=1
      IF (NL.EQ.3) INC(3,2)=1
C
C
C
C       THE VECTOR-SAMPLED FUNCTIONS PHI(J) ARE STORED IN
C       THE FIRST N ROWS AND FIRST L COLUMNS OF THE MATRIX
C       B(I,J). B(I,J) CONTAINS PHI(J,ALF;T(I),I,...N;
C       J=1,L. THE CONSTANT FUNCTIONS PHI WHICH DO NOT
C       DEPEND UPON ANY NONLINEAR PARAMETERS ALF MUST
C       APPEAR FIRST.
C
C
C
C
C
C
 90         alf(2)=dabs(alf(2))
            DO 81 I=1,N
            IF (ISEL.EQ.1) A(I,1)=1.Ø
            A(I,2)=Ø.
            TTI=T(I)-ALF(3)
            IF (TTI.LE.0.) GO TO 81
            TFUNC=DSQRT(ALF(2)*TTI)
            A(I,2)=DERFC((XPOS-ALF(1)*TTI)/(2.*TFUNC))
      IF (XPOS*ALF(1)/ALF(2).LT.2Ø.) A(I,2)=A(I,2)+DEXP(ALF
     1      (1)*XPOS/ALF(2))*DERFC((XPOS+ALF(1)*TTI)/(2.*TFUNC))
C
  81        CONTINUE
C
C
C
       IF (ISEL.EQ.2) GO TO 200
C
C
C
C
C
C
C
 165       DO 170 I=1,N
C
           A(I,4)=Ø.
           A(I,5)=Ø.
           A(I,6)=Ø.
           TTI=T(I)-ALF(3)
           IF (TTI.LE.Ø.) GO TO 170
```

```
          E1=DEXP(-((XPOS-ALF(1)*TTI)**2)/(4.*ALF(2)*TTI))
          E2=DEXP(-((XPOS+ALF(1)*TTI)**2)/(4.*ALF(2)*TTI))
          E3=Ø.
         IF (ALF(1)*XPOS/ALF(2).LT.2Ø.) E3=DEXP(ALF(1)*XPOS/ALF(2))
          ERF1=DERFC((XPOS+ALF(1)*TTI)/(2.*DSQRT(ALF(2)*TTI)))
          FUNC=2.*DSQRT(TTI*ALF(2)**3)
          A(I,4)=-TPI*E1*(-TTI/(2.*DSQRT(ALF(2)*TTI)))
          IF (E3.GT.Ø.) A(I,4)=A(I,4)+(XPOS/ALF
     1      (2))*E3*ERF1+E3*(-TPI*E2)*TTI/(2.*DSQRT(ALF(2)*TTI))
          A(I,5)=TPI*E1*(XPOS-ALF(1)*TTI)/FUNC
          IF (E3.GT.Ø.) A(I,5)=A(I,5)-E3*ERF1*ALF(1)*XPOS/
     1     ALF(2)**2+E3*TPI*E2*(XPOS+ALF(1)*TTI)/FUNC
          IF (NL.LT.3) GO TO 170
          A(I,6)=-TPI*E1*(XPOS/(2.*DSQRT(ALF(2)*TTI**3))+ALF(1)/
     1      FUNC)
         IF (E3.GT.Ø.) A(I,6)=A(I,6)+E3*(-TPI*E2)
     2      *(XPOS/(2.*DSQRT(ALF(1)*TTI**3))-ALF(1)/FUNC)
C
C
 170              CONTINUE
C
  200        CONTINUE
C
        RETURN
        END
C
C
C
C
C
       SUBROUTINE ADADRV (LP,NL,N,NMAX,LPP2,IV,A,INC,T,ALF,ISEL)
       IMPLICIT REAL*8(A-H,O-Z)
       DIMENSION ALF(NL),A(NMAX,LPP2),T(NMAX),INC(12,8)
       COMMON XPOS,TZERO
C
       PI=4.*ATAN(1.)
       TPI=2./DSQRT(PI)
       L=LP-1
       IF (NL.LT.3) ALF(3)=TZERO
C
C
C       THE INCIDENCE MATRIX INC(NL,L+1) IS FORMED BY SETTING
C       INC(K,J)=1 IF THE NONLINEAR PARAMETER ALF(K) APPEARS
C       IN THE J-TH FUNCTION PHI(J). (THE PROGRAM SETS ALL OTHER
C       INC(K,J) TO ZERO.)
C
C
       IF(ISEL.EQ.2) GO TO 90
       IF(ISEL.EQ.3) GO TO 165
C
       INC(1,1)=Ø
       INC(2,1)=Ø
       INC(1,2)=1
       INC(2,2)=1
       IF (NL.EQ.3) INC(3,2)=1
C
C
C
C       THE VECTOR-SAMPLED FUNCTIONS PHI(J) ARE STORED IN
C       THE FIRST N ROWS AND FIRST L COLUMNS OF THE MATRIX
C       B(I,J). B(I,J) CONTAINS PHI(J,ALF;T(I),I,...N;
C       J=1,L. THE CONSTANT FUNCTIONS PHI WHICH DO NOT
C       DEPEND UPON ANY NONLINEAR PARAMETERS ALF MUST
C       APPEAR FIRST.
```

```
C
C
C
C
C
C
 90         alf(2)=dabs(alf(2))
            DO 81 I=1,N
            IF (ISEL.EQ.1) A(I,1)=1.Ø
            A(I,2)=Ø.
            TTI=T(I)-ALF(3)
            IF (TTI.LE.Ø.) GO TO 81
c
c     calculate derivatives
c
            tti=tti+1.d-5
            TFUNC=DSQRT(ALF(2)*TTI)
            ffunc=DERFC((XPOS-ALF(1)*TTI)/(2.*TFUNC))
      IF (XPOS*ALF(1)/ALF(2).LT.2Ø.) ffunc=ffunc+DEXP(ALF
     1     (1)*XPOS/ALF(2))*DERFC((XPOS+ALF(1)*TTI)/(2.*TFUNC))
            tti=tti-2.d-5
            TFUNC=DSQRT(ALF(2)*TTI)
            gfunc=DERFC((XPOS-ALF(1)*TTI)/(2.*TFUNC))
      IF (XPOS*ALF(1)/ALF(2).LT.2Ø.) gfunc=gfunc+DEXP(ALF
     1     (1)*XPOS/ALF(2))*DERFC((XPOS+ALF(1)*TTI)/(2.*TFUNC))
            a(i,2)=(ffunc-gfunc)/2.d-5
C
 81         CONTINUE
C
C
C
        IF (ISEL.EQ.2) GO TO 200
C
C
C
C
C
C
C
 165        DO 170 I=1,N
C
            A(I,4)=Ø.
            A(I,5)=Ø.
            A(I,6)=Ø.
            TTI=T(I)-ALF(3)
            IF (TTI.LE.Ø.) GO TO 170
            tti=tti+1.d-5
            E1=DEXP(-((XPOS-ALF(1)*TTI)**2)/(4.*ALF(2)*TTI))
            E2=DEXP(-((XPOS+ALF(1)*TTI)**2)/(4.*ALF(2)*TTI))
            E3=Ø.
          IF (ALF(1)*XPOS/ALF(2).LT.2Ø.) E3=DEXP(ALF(1)*XPOS/ALF(2))
            ERF1=DERFC((XPOS+ALF(1)*TTI)/(2.*DSQRT(ALF(2)*TTI)))
            FUNC=2.*DSQRT(TTI*ALF(2)**3)
            A4p=-TPI*E1*(-TTI/(2.*DSQRT(ALF(2)*TTI)))
            IF (E3.GT.Ø.) A4p=A4p+(XPOS/ALF
     1        (2))*E3*ERF1+E3*(-TPI*E2)*TTI/(2.*DSQRT(ALF(2)*TTI))
            A5p=TPI*E1*(XPOS-ALF(1)*TTI)/FUNC
            IF (E3.GT.Ø.) A5p=A5p-E3*ERF1*ALF(1)*XPOS/
     1       ALF(2)**2+E3*TPI*E2*(XPOS+ALF(1)*TTI)/FUNC
            IF (NL.LT.3) GO TO 2Ø2
            A6p=-TPI*E1*(XPOS/(2.*DSQRT(ALF(2)*TTI**3))+ALF(1)/
     1        FUNC)
          IF (E3.GT.Ø.) A6p=A6p+E3*(-TPI*E2)
     2        *(XPOS/(2.*DSQRT(ALF(1)*TTI**3))-ALF(1)/FUNC)
```

```
c
 202      tti=tti-2.d-5
          E1=DEXP(-((XPOS-ALF(1)*TTI)**2)/(4.*ALF(2)*TTI))
          E2=DEXP(-((XPOS+ALF(1)*TTI)**2)/(4.*ALF(2)*TTI))
          E3=Ø.
         IF (ALF(1)*XPOS/ALF(2).LT.2Ø.) E3=DEXP(ALF(1)*XPOS/ALF(2))
          ERF1=DERFC((XPOS+ALF(1)*TTI)/(2.*DSQRT(ALF(2)*TTI)))
          FUNC=2.*DSQRT(TTI*ALF(2)**3)
          A4m=-TPI*E1*(-TTI/(2.*DSQRT(ALF(2)*TTI)))
          IF (E3.GT.Ø.) A4m=A4m+(XPOS/ALF
     1      (2))*E3*ERF1+E3*(-TPI*E2)*TTI/(2.*DSQRT(ALF(2)*TTI))
          A5m=TPI*E1*(XPOS-ALF(1)*TTI)/FUNC
          IF (E3.GT.Ø.) A5m=A5m-E3*ERF1*ALF(1)*XPOS/
     1     ALF(2)**2+E3*TPI*E2*(XPOS+ALF(1)*TTI)/FUNC
          IF (NL.LT.3) GO TO 281
          A6m=-TPI*E1*(XPOS/(2.*DSQRT(ALF(2)*TTI**3))+ALF(1)/
     1      FUNC)
         IF (E3.GT.Ø.) A6m=A6m+E3*(-TPI*E2)
     2      *(XPOS/(2.*DSQRT(ALF(1)*TTI**3))-ALF(1)/FUNC)
c
 2Ø1     a(i,4)=(a4p-a4m)/2.d-5
         a(i,5)=(a5p-a5m)/2.d-5
         if (nl.lt.3) goto 170
         a(i,6)=(a6p-a6m)/2.d-5
C
C
 170             CONTINUE
C
  200        CONTINUE
C
         RETURN
         END
C
         subroutine fixup (yout,x,tout,t,n,np,nplus)
         implicit real*8 (a-h,o-z)
        dimension t(5ØØ),x(5ØØ),yout(5ØØ),tout(5ØØ)
        n3=2*nplus+1
        n=Ø
        istart=1Ø
        do 1 i=istart+nplus+1,np-nplus
        n=n+1
        sumx=Ø.Ø
        sumy=Ø.Ø
        sumxy=Ø.Ø
        sumx2=Ø.Ø
        do 2 j=i-nplus,i+nplus
        sumx=sumx+t(j)
        sumy=sumy+x(j)
        sumxy=sumxy+x(j)*t(j)
        sumx2=sumx2+t(j)*t(j)
 2      continue
        grad=(sumx*sumy-n3*sumxy)/(sumx*sumx-n3*sumx2)
        tout(n)=t(i)
        yout(n)=grad
 1      continue
        return
        end
```