PARAMETER ESTIMATION USING MICRODIELECTROMETRY WITH APPLICATION TO TRANSFORMER MONITORING

Vol 1

by

MARK CAMERON ZARETSKY

S.B., Massachusetts Institute of Technology
(1980)

S.M., Massachusetts Institute of Technology
(1982)

SUBMITTED TO THE DEPARTMENT OF
ELECTRICAL ENGINEERING AND COMPUTER SCIENCE
IN PARTIAL FULFILLMENT OF THE REQUIREMENTS
FOR THE DEGREE OF

DOCTOR OF SCIENCE
at the
MASSACHUSETTS INSTITUTE OF TECHNOLOGY
November, 1987

Signature of Author .....
Department of Electrical Engineering
and Computer Science

Certified by .....
Stratton Professor, EECS
Thesis Supervisor

Accepted by .....
Chairman, Departmental Committee
on Graduate Students

Vol 1

PARAMETER ESTIMATION USING MICRODIELECTROMETRY WITH APPLICATION TO TRANSFORMER MONITORING

by

MARK CAMERON ZARETSKY

Submitted to the Department of Electrical Engineering and Computer Science on November 13, 1987 in partial fulfillment of the requirement for the Degree of Doctor of Science

ABSTRACT

Two related objectives of this work are to develop the capability of inferring fundamental continuum parameters of semi-insulating heterogeneous media using an interdigital electrode microdielectrometer and to utilize this capability to provide new sensors useful for transformer monitoring. Using a modal approach, a continuum model is derived for describing the electrostatic field distribution associated with an interdigital electrode structure. In this "imposed $\omega - k$" technique, the medium is excited at the temporal (angular) frequency ω by means of an interdigital structure having a spatial periodicity length $\lambda = 2\omega/k$ and hence a dominant wavenumber k. Given one fourier component of the normal displacement field in response to an applied potential, expressed as the surface capacitance density $C(\omega, k)$, of any linear system having property gradients perpendicular to the plane of the electrodes, the model predicts the resulting complex gain taking into account the properties and geometry of the interdigital structure. Root searching techniques incorporating the continuum model, using either a secant or Newton's method, are then developed for solving inverse problems and hence estimating continuum parameters from experimental data. Specifically illustrated are *absolute* measurements of the thickness of deposited polymer films (0–20 μm), the dispersive complex bulk permittivities of plasma deposited bromobenzene, parylene, paper insulation and transformer oil, and the dispersive complex surface permittivities of a tin oxide thin film (350 Å) and moisture adsorbed on the surface of SiO_2. These measurements are made using electrode structures of 50 μm (microchip) and 1 mm (macrochip) spatial wavelength. In addition, an algorithm has been developed for estimating spatial distributions in heterogeneous media utilizing data obtained by varying the spatial wavelength, rather than temporal frequency, imposed by the interdigital electrode structure.

These capabilities are then applied towards developing sensors for both oil complex bulk permittivity and oil moisture content. Passivation of the microchip and macrochip electrodes with a 5–7 μm layer of parylene is necessary for prevention of the adsorption of moisture from the oil. Consequently, a prerequisite for accurate measurement of the oil complex bulk permittivity is the ability to take into account the finite thickness parylene layer and its dispersive complex bulk permittivity. This capability is provided by the previously mentioned continuum model.

Sensitization of the microchip to moisture content in the oil is accomplished by plasma depositing a thin (2–7 μm) film of bromobenzene. This film undergoes changes

in its dielectric properties that reflect the adsorption of moisture. The observed frequency response can be normalized to external parameters such as moisture content and temperature. This allows the coating sensitivity to be characterized by one parameter, chosen to be the frequency at which the phase peak occurs, f_p. It is shown that the coating behaves as a semipermeable membrane, serving as a barrier to the transformer oil but absorbing moisture proportional to the absolute moisture content in the oil. Sensitivity to moisture contents of less than 5 to roughly 50 ppm is demonstrated. The transient response of the coated microchips is on the order of a minute. Preliminary on–line monitoring results provide a real time demonstration of the change in moisture content with load cycling of a 50 kVA test transformer. Moisture permeation and electrical conduction within the coating are examined in terms of the heterogeneous structure of the coating. The aging of the films on a monthly time scale, resulting in decreased moisture sensitivity, is characterized and discussed.

A preliminary effort is made to develop a hydrogen gas sensor using a thin film (500 Å) palladium resistor. Changes in bulk resistance in response to both 100 and 1000 ppm H_2 in nitrogen at temperatures of 100° and 225°F are observed.

Thesis Supervisor: James R. Melcher
Title: Stratton Professor of Electrical Engineering and Computer Science

ACKNOWLEDGEMENTS

For many years now, I have had the pleasure of both working for and learning with Professor Melcher. It has been a long and slippery path, beginning with corn oil, passing through ferrofluids, diesel soot (with motor oil), and ending in transformer oil. The experience has been very stimulating and rewarding (not to mention dirty). His impression on both my academic and personal pursuits will be a positive and lasting one.

This work was supported by a consortium of several electric utilities. I would like to thank them for the opportunity to perform scientific research and apply the knowledge gained to the practical application of transformer monitoring. Thanks to Phil Gattens for providing the moisture content data presented in Section 6.2.2.

The constructive criticisms and support of the thesis readers, Professors Stephen Senturia and Alan Grodzinsky, and Doctor Chathan Cooke, was greatly appreciated. Each reader provided their own unique insights, resulting in a much improved document.

Bridge measurements of transformer oil samples, as well as insights into dielectric behaviour, were provided by Bill Westphal. Avi Benatar was very helpful in learning to operate the plasma deposition apparatus, obtained on loan from Professor Gutowski. Huan Lee and Dave Day, both of Micromet Instruments, were very generous with their time – helping in the installation of the microdielectrometer and answering questions about its operation. The thin film palladium resistors were fabricated by Professor Mark Wrighton's chemistry lab and Tracy Jones was instrumental in this effort.

Eliot Frank has consistently been a life, I mean computer, saver. He figured out how to talk to the microdielectrometer in Xenix and his software provided most of the plots in this thesis. The data taken in the test facility would not have been possible without the herculean efforts of Wayne Hagman and Dan Flagg. Testing of the thin film palladium resistors was performed by Paul Warren, another person I have known and enjoyed working with for many years. Many friends in the lab have been helpful over the years, both on an intellectual and social level: Lama Mouayad, Philip Von Guggenberg, Kevin Delin, Paul Grimshaw, Patrick Li, Silvano Brewster, Ric Carreras, Steve Bart and Karen Kirmse. Thanks for bearing with me, folks. A special mention goes to ABC basketball and Out-of-Towners softball.

My parents, Carole and Herb, who have made it too easy for me, have provided the role models from which I draw my strength upon. This accomplishment is as much theirs as mine. Finally, to my girlfriend Judy, I gratefully acknowledge your patience, understanding, support and love.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

1.1 Background – Engineering and Science

As with many research endeavors, this work reflects the interplay of two driving forces – scientific interests and engineering needs. The scientific interest is understanding observed phenomena at a fundamental (molecular) level and correlating these observations with self consistent models. The engineering need is developing electrochemical sensors to monitor various chemical constituents within transformer oil. The trick is to recognize that one interest does not preclude the other and, in fact, to be successful a project must satisfy both interests. In the following sections the scientific and engineering aspects of this work are discussed along with the goals of this thesis.

1.2 Scientific Motivations

The work presented here applies the disciplines of system identification and analytical and numerical field theory to the problem of interpreting information obtained using interdigital electrode structures. Using the fringing fields of a planar electrode structure, the response to an applied potential at the electrode–medium interface can be measured noninvasively. Estimates of the dielectric properties of the material can be obtained from these measurements via parameter estimation routines incorporating a continuum model of the electric field distribution within the material. An increased

understanding of the physical structure of a material can be obtained with such estimations.

A specific application of the above approach is to measure the dielectric properties of thin semi–insulating films. Using the insight gained from these measurements regarding the physical structure of the films, a strategy can be evolved for developing electrochemical sensors. Of particular interest here is the development of both a moisture and a hydrogen gas sensor for transformer oil.

The instrument used to make these dielectric measurements is a microdielectrometer – a device utilizing a set of microfabricated interdigital electrodes for electrostatically probing a material. A continuum model for the electric field distribution of this interdigitated electrode structure has been formulated. This model has an emphasis on generality with respect to the types of media with which the electrode structure may interact. By embedding this continuum model in simple single or multi parameter estimation routines, it is possible to hypothesize and estimate parameters, from experimental data, describing heterogeneities or interactions within a medium under investigation. Prior to this work the only models for interpreting experimental data estimated the dielectric properties assuming one of two limits: 1) a uniform medium or 2) a thin film ($< 1\mu$m) surrounded by a uniform dielectric. Thus, the work presented here greatly expands the applicability of the microdielectrometer.

The following two sections discuss the advantages of using planar, interdigital electrode structures and the implications of probing a medium using electroquasistatic fields.

1.2.1 Imposed ω–k Sensing

Classical dielectrometry obtains information concerning the physical properties and state of a medium from the electrical frequency response. The modal approach to dielectrometry, presented in Chapter 2, not only extracts its information from the control of the temporal frequency, but also from the imposition of a spatial periodicity as well. In this "imposed ω–k" technique, the medium is excited at the temporal (angular) frequency ω by means of an interdigital electrode structure having a spatial

periodicity length $\lambda = 2\pi/k$ and hence a dominant wavenumber k.

In Fig. 1.1, the medium is above and the structure used to make the measurements is below. Because the electrodes introduce higher spatial harmonics, field quantities, such as the potential, are expressed as

$$\Phi(x,y,t) = Re\hat{\Phi}(x,y)e^{j\omega t} \tag{1.1}$$

where

$$\hat{\Phi}(x,y) = \sum_{n=0}^{\infty} \hat{\Phi}_n(x)\cos(k_n y); k_n = 2n\pi/\lambda$$

representing a superposition of standing waves of dominant wavenumber $k = k_1$ in the plane of the electrodes. The choice of the cosinusoidal representation exploits the symmetry of both the electrode structure and the situation above the electrodes (for example, there is no motion).

The advantages of the ω–k approach are many. First, there is the coupling into the medium from a single surface, allowing measurements to be more readily made noninvasive, if desired. For example, dielectric measurements of thin films can be performed without having to vapor deposit an upper metal electrode for use with a capacitance bridge, and in many cases, can be taken at the site of the coating process without changing the ambient environment.

Secondly, because the fields generated by the electrodes in the medium (above in Fig. 1.1) are quasistatic, they tend to decay into the material exponentially with a characteristic length that is at most a fraction of the spatial wavelength. Consequently, the electrodes are sensitive to dielectric materials located within a region roughly $\lambda/3$ or less from the electrode plane. Thus, the spatial sensitivity of the device can be tailored to individual needs.

A third advantage comes with modern microfabrication techniques that not only make micrometer scale electrodes possible, but make the integration with the signal processing electronics feasible as well. Electrodes with a very short spatial wavelength, λ, can be deposited. This refines the spatial resolution. For example, given a spatial

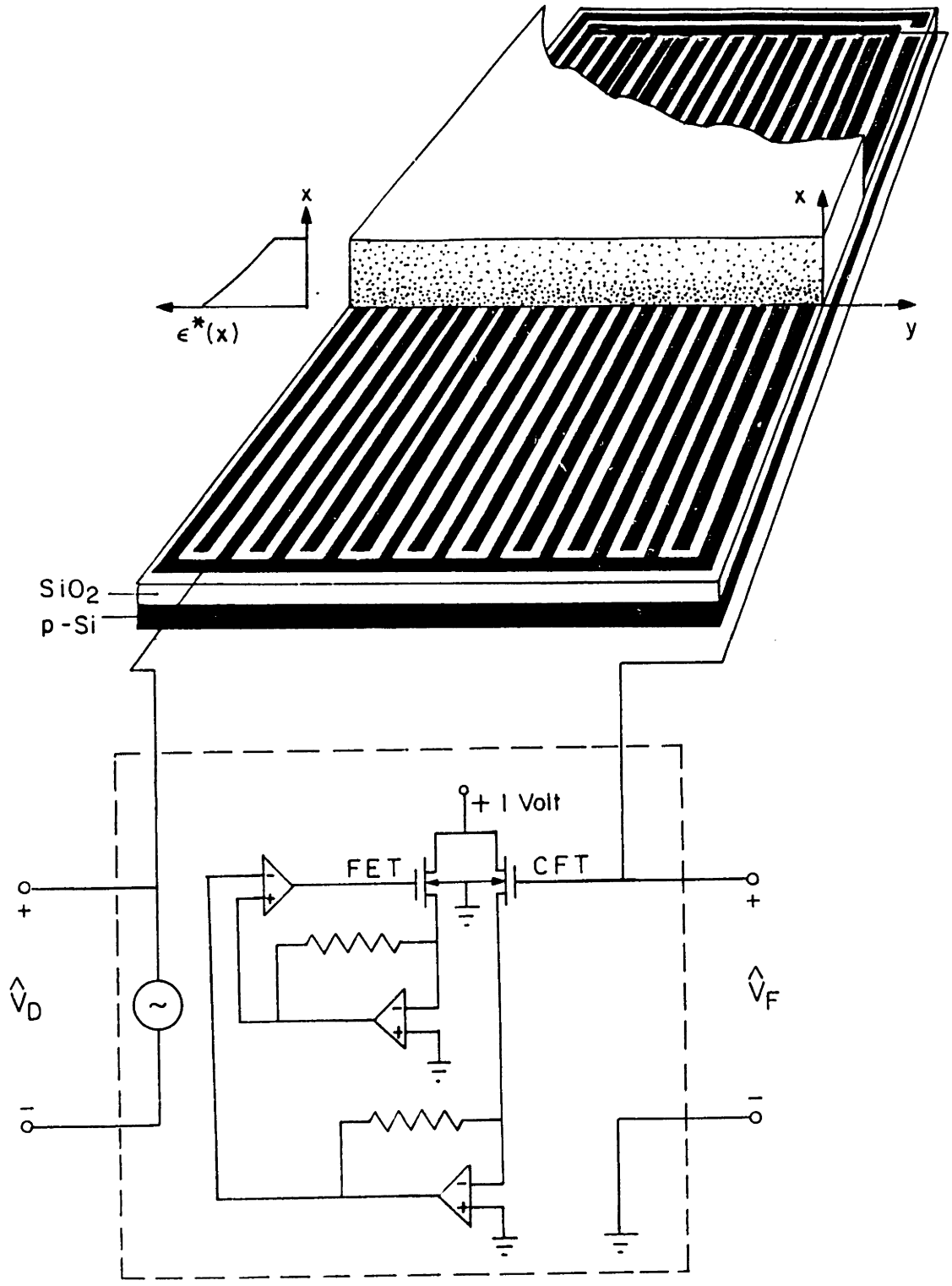


Figure 1.1: Representation of Interdigital Electrode Structure and Associated Circuitry Interacting with a Heterogeneous Media.

wavelength of 50 micrometers, thin films of 20 micrometers or less (perhaps coated on the electrodes) can be distinguished and their dielectric dispersions examined.

Having short wavelengths also improves the sensitivity to measurements of surface conductivity, σ_s. Sensing conduction on the surface of an insulator without making electrical contact with the surface requires capacitive coupling. Thus, the frequency must be high enough to make the capacitive reactance on the order of the resistance of the surface. Roughly, this requires that to measure a surface conductivity σ_s, at the frequency $f \equiv \omega/2\pi$, the ratio $(\sigma_s/f\epsilon\lambda)$ must be of the order of unity. There is a practical lower limit on the frequency. Therefore the smaller the wavelength λ, the smaller the surface conductivity that can be measured. Having the electronics integrated with the electrode structure makes it possible to use frequencies as low as 0.005 Hz using the "chip" described in the next section.

Although the emphasis here will be on dielectrometry, the imposed ω–k approach can be used in systems where the electrical response reflects electromechanical and electrochemical phenomena. This is illustrated by earlier studies where interdigital electrodes were used to study the electromechanics of double layers [1], electrohydrodynamic surface waves and instabilities [2] and sensing of electrification effects in insulating fluids [3,4].

1.2.2 Intimate Sensing

By contrast with remote sensing, where electromagnetic fields described by the wave equation are often used to resolve properties and states of materials at great distances, the imposed ω–k approach exploits fields that are quasistatic from the electromagnetic point of view. In the electroquasistatic embodiment described here, the dynamics result from charge migration and diffusion, and polarization relaxation. In the magnetoquasistatic analogue, the dynamics would involve current diffusion and magnetization hysteresis. In either of these latter cases, the fields are quasistatic and tend to be confined to the immediate neighborhood of the interdigital electrode structure. Because measurements reflect the properties and state within a short distance, the application of such structures to parameter estimation might be dubbed "intimate sensing".

Strategies exploiting the wavelength controlled resolution inherent to having fields that are confined to the proximity of the electrode can utilize the model developed here.

With the combination of microfabrication technology and the "imposed ω–k" technique, a determination of electromechanical or electrochemical structures and dynamics on a microscale becomes feasible. Fluid boundary layers, electrochemical double layers and thin film structures all fall into this microscale realm. Up to now, the behaviour of these phenomena as a function of temporal frequency has been used to identify various structures. To the extent that this requires the assumption of frequency independent parameters to effectively extract information, this approach to parameter estimation gives little opportunity to see the resulting model as unique.

With the usage of the approach presented here, a continuum model coupled with a parameter estimation routine, it becomes feasible to extract information at a particular temporal frequency by varying the spatial frequency of the applied potential. As mentioned before, the field decay normal to electrodes is directly proportional to the spatial wavelength of the electrodes. Intuitively, it makes more sense to determine the structure of heterogeneous media by probing spatially, rather than temporally. This method does not have the restriction regarding the frequency dependence of parameters such as the complex permittivity. Practically, this effort amounts to having multiple electrode structures, each having a different spatial wavelength, monitoring the same medium, or perhaps one array of electrodes with switchable terminal connections to produce the various spatial wavelengths.

1.3 Engineering Motivations – On–Line Transformer Monitoring

Large scale power transformers (1 MVA and up) are devices that represent a large capital investment, provide an essential service requiring minimal interruption, are subject to catastrophic failures and are costly to repair. The potential benefits from improving the decision making process with regards to maintenance and servicing make it highly desirable to monitor these transformers. The following section reviews the

field experience gained from monitoring changes in various properties of transformer oil using off–line techniques.

1.3.1 Background

It is not the intention to extensively review all quantities that might be of interest. The focus here is the monitoring of the oil circulating within the transformer. Just as a blood analysis reveals significant information about the condition of the body, it is felt that an oil analysis will be important in determining the state of a transformer. There are two major functions served by the oil, heat transfer and insulation. Traditionally, mineral oils consisting mainly of paraffins and cyclo–paraffins have been employed. Other insulating fluids are continually being examined due to the vagaries of world oil supply and in the search for less flammable or otherwise more technically advantageous fluids. However, at present, economics greatly favor using refined petroleum oil. An excellent review of insulating fluids, their properties and uses has been done by Wilson [5].

There are a wide variety of ASTM tests used to characterize the electrical, physical and chemical properties of transformer oil [6]. The oil must meet certain specifications both prior to going into a transformer and while in service. Quantities of direct interest are: dielectric breakdown strength, power factor, dielectric loss, moisture content, dissolved gas content, acidity, interfacial tension, copper and iron content, sludge and particulate count. Reprinted in Table 1.1 is a summary of screening specifications for refined mineral oil without additives and property requirements for new oil to be used in electrical apparatus. As a result of many years of accumulated experience, both in the field and laboratory, changes in the above quantities over time (with the exclusion of the first three) can be associated with particular degradation processes occurring within the transformer [7]. [1]

An important example is dissolved gas analysis (DGA). Though there are several schemes for calculating and interpreting gas ratios [8,9] it is generally agreed

[1]Note – many of the quantities discussed are highly temperature dependent, thus monitoring or estimating the oil temperature is assumed.

Table 1.1: Reprinted Summary of ASTM Screening Specifications and Functional Property Requirements (from [6]).

TABLE 1 Screening Specifications

Property[a]	Specified Value	ASTM Method[a]
Physical:		
Color, max	1	D 1500
Flash point, min	146 C (295 F)[b]	D 92
Interfacial tension, min, dynes/cm	40	D 971
Pour point, max	−40 C (−40 F)[c]	D 97
Specific gravity at 15.56/15.56 C (60/60 F)	0.84 to 0.91	D 1298
Viscosity, max, SUS:		
at 37.78 C (100 F)	65	D 88
at 0 C (32 F)	320	D 445 and D 2161
Chemical:		
Additives	nil	
Combined corrosive sulfur	noncorrosive	D 1275
Inorganic chlorides and sulfates	nil	D 878
Neutralization number, max, mg KOH/g oil	0.05	D 974 or D 664
Sludge by oxidation test,[d] max, percent	0.06	D 1313
Sludge after oxidation stability:		
72 h, max, percent	0.3	D 2440
164 h, max, percent	0.7	D 2440
Neutralization number after oxidation:		
72 h, max, mg KOH/g oil	2.4	D 2440
164 h, max, mg KOH/g oil	2.6	D 2440
Water, max, ppm	35	D 1533
Electrical.		
Dielectric breakdown voltage, min, kV	30.0	D 877
Power factor at 100 C (212 F), 60 Hz, max	0.005 (0.5 percent)	D 924

D 3487

TABLE 1 Functional Property Requirements

Property	Limit		ASTM Test Method
	Type I	Type II	
Physical:			
Aniline point, °C	(63–78[a])	(63–78)[a]	D 611
Color, max	0.5	0.5	D 1500
Flash point, min, °C	145	145	D 92
Interfacial tension at 25°C, min, dynes/cm	40	40	D 971
Pour point, max, °C	−40[b]	−40[b]	D 97
Specific gravity, 15°C/15°C max	0.91	0.91	D 1298
Viscosity, max, cSt (SUS) at:			
100°C	3.0 (36)[c]	3.0 (36)[c]	D 445 or D 88
40°C	12.0 (66)[c]	12.0 (66)[c]	
0°C	76.0 (350)	76.0 (350)	
Visual examination	clear and bright	clear and bright	D 1524
Electrical.			
Dielectric breakdown voltage at 60 Hz			
Disk electrodes, min, kV	30	30	D 877
VDE electrodes, min, kV either at 0.040-in. (1.02-mm) gap, or at 0.080-in. (2.04-mm) gap	28[d]	28[d]	D 1816
	56[d]	56[d]	
Dielectric breakdown voltage, impulse conditions			D 3300
25°C, min, kV. Needle negative to sphere grounded, 1-in. (25.4-mm) gap	145[e,g]	145[e,g]	
Power factor at 60 Hz max, %			D 924
25°C	0.05	0.05	
100°C	0.30	0.30	
Chemical.			
Oxidation stability (acid-sludge test)			D 2440
72 h:			
% sludge, max, by mass	0.15	0.1[e]	
Total acid number, max, mg KOH/g	0.5	0.3[e]	
164 h:			
% sludge, max, by mass	0.3	0.2[e]	
Total acid number, max, mg KOH/g	0.6	0.4[e]	
Oxidation stability (rotating bomb test), min, minutes	—	195	D 2112
Oxidation inhibitor content, max, % by mass	0.08	0.3	D 1473 or D 2668[f]
Corrosive sulfur	noncorrosive		D 1275
Water, max, ppm	35	35	D 1315 or D 1533
Neutralization number, total acid number, max, mg KOH/g	0.03	0.03	D 974

that mechanisms such as arcing, partial discharge and thermal degradation can be distinguished using DGA. For example, large quantities of hydrogen indicate partial discharges, whereas acetylene indicates arcing is occurring [10,11]. Carbon monoxide and dioxide indicate overheating of the cellulose insulation.

Moisture content is used as an indication of the condition of the paper insulation, existence of free water and leaks to the atmosphere [12]. Typically, paper is dried to roughly 0.5–1% moisture by weight prior to filling a transformer with oil. The solubility of moisture in the paper, relative to the oil, will vary from 3000 times more at 20°C to 400 times at 60°C [13]. Thus, during the regular load cycling of the transformer, the accompanying temperature variations will drive moisture into and out of the paper. At the higher temperatures the oil can absorb more moisture. This excess moisture should go back into the paper upon cooling. However, if the oil cools faster than the time required for equilibrium to occur, then the oil can become supersaturated and free water will form. Free water can lead to breakdown if it appears in high field stress locations either in the oil or absorbed into the paper. Thus it is important to monitor the relative saturation of the oil.

Assuming an equilibrium exists, the moisture content in oil can be directly related to the moisture content in the paper [14] (this is a point of contention and will be discussed further in Section 6.1). Paper with a high moisture content ($> 2\%$) will suffer a significant loss in electrical breakdown strength [15]. Degradation of the oil can result in higher moisture saturation levels in the oil, increasing the moisture content at a given temperature [16]. Thermal degradation of the cellulose insulation will result in the formation of carbon and water. Moisture content can also affect static electrification, by altering the charging tendency of the oil [17].

Oxidation of the oil is indicated by changes in oil resistivity, acidity, sludge formation or interfacial tension [18]–[21]. Analysis of the particulate matter in oil with respect to composition, number, size and shape is also important [22]. Copper and iron particulate are good catalysts for oxidation of the oil while excessive amounts of cellulose fiber might indicate excessive wear of the insulation. A review of electrical loss mechanisms in transformer oil, and how oxidation effects its dielectric properties

can be found in [23, Section 1.2].

Oil sampling and analysis, under normal operating circumstances, is currently performed on a semi–annual basis. There are documented cases of the dissolved gas analysis alerting operators to internal problems (usually in conjunction with other tests) [24]–[27] and even preventing failure [28]. The utility of this analysis has been extended to other types of power equipment such as bushings [29]. It is clear that monitoring dissolved gas content is important and it follows that similar benefits may be gained through monitoring of quantities such as moisture content and acidity.

However, the above tests all involve removing a sample of oil from the transformer and introducing the oil into another apparatus. For example, moisture analysis involves titrating the oil with a Karl Fischer reagent and measuring the resultant current. The method is based on the reduction of iodine by sulfur dioxide in the presence of water. Dissolved gas analysis requires stripping the gas from an oil sample and passing the gas through a gas chromatograph. It would be very costly to frequently obtain oil samples from transformers in the field and analyze them in a lab. Typically, sampling is performed annually, or semi–annually, unless there is suspicion of a malfunction. In that case, more frequent monitoring of the oil is performed. This mode of sampling is haphazard, it will catch some transformers before they fail – ones slowly deteriorating or rapidly failing at the time of sampling – and completely miss others. There is at least one documented case of the time to failure being on the order of minutes [30]. In addition, the measurements may be questionable due to factors such as poor or variable sampling technique or sample transport. Thus, there is an established need for on–line monitoring of various physico–chemical properties of transformer oil.

1.3.2 State of the Art Review

This section reviews existing sensor and analyzer technology as either directly applied to transformer oil monitoring or more broadly applied to selective gas detection. Regarding the latter category, the focus is narrowed to solid state devices for gas detection, avoiding liquid electrolyte detectors. The main reason for this focus is the emphasis (perhaps unnecessarily) on developing detectors immersible in the oil. The

sealing problems involved with liquid electrolyte detectors made investigation of this area unattractive. Of course, operation of detectors in the vapor phase would greatly alter this viewpoint. Another factor skewing the focus of this search is the desire to find techniques adaptable to the microchip structure.

Functionally, there are at least two distinct processes involved in the analysis of either moisture or dissolved gases in oil – extraction and detection. With gas detection a third process, gas separation, may be necessary. This is dependent on the selectivity of the gas sensor used. All of the sensors discussed below operate in the vapor phase. Therefore, gas extraction is performed using a reinforced diaphragm, separating the oil from a sampling chamber. This diaphragm is designed to withstand the internal oil pressure and allow the pertinent gases to diffuse through. When equilibrium has been reached across the membrane, the gas detection process takes place. The time until equilibrium is reached will depend greatly on the membrane permeability to each gas.

Moisture Sensors

Only one type of device for on–line monitoring of moisture in oil is available at present [12]. This device allows a small volume (< 1 cc) of initially dry air to come into equilibrium with the oil across a polymer membrane. Thus, the device essentially is a dew point hygrometer – the collected vapor is cooled through contact with a mirror surface whose temperature is monitored. An optical system detects the occurrence of condensation, thus measuring the dew point (Fig. 1.2). This dew point also indicates the relative humidity of the air. At equilibrium the relative humidity of the air equals the relative saturation of the oil (see Section 6.1). In this manner, a measurement of the relative saturation of the oil is obtained. As discussed earlier, the relative saturation of the oil can indicate the possibility of free water formation. In addition, it can also be used to help determine the moisture content of the paper insulation as shown in Chapter 6. No information on field testing of such devices has been reported as of this writing.

Gas Detectors

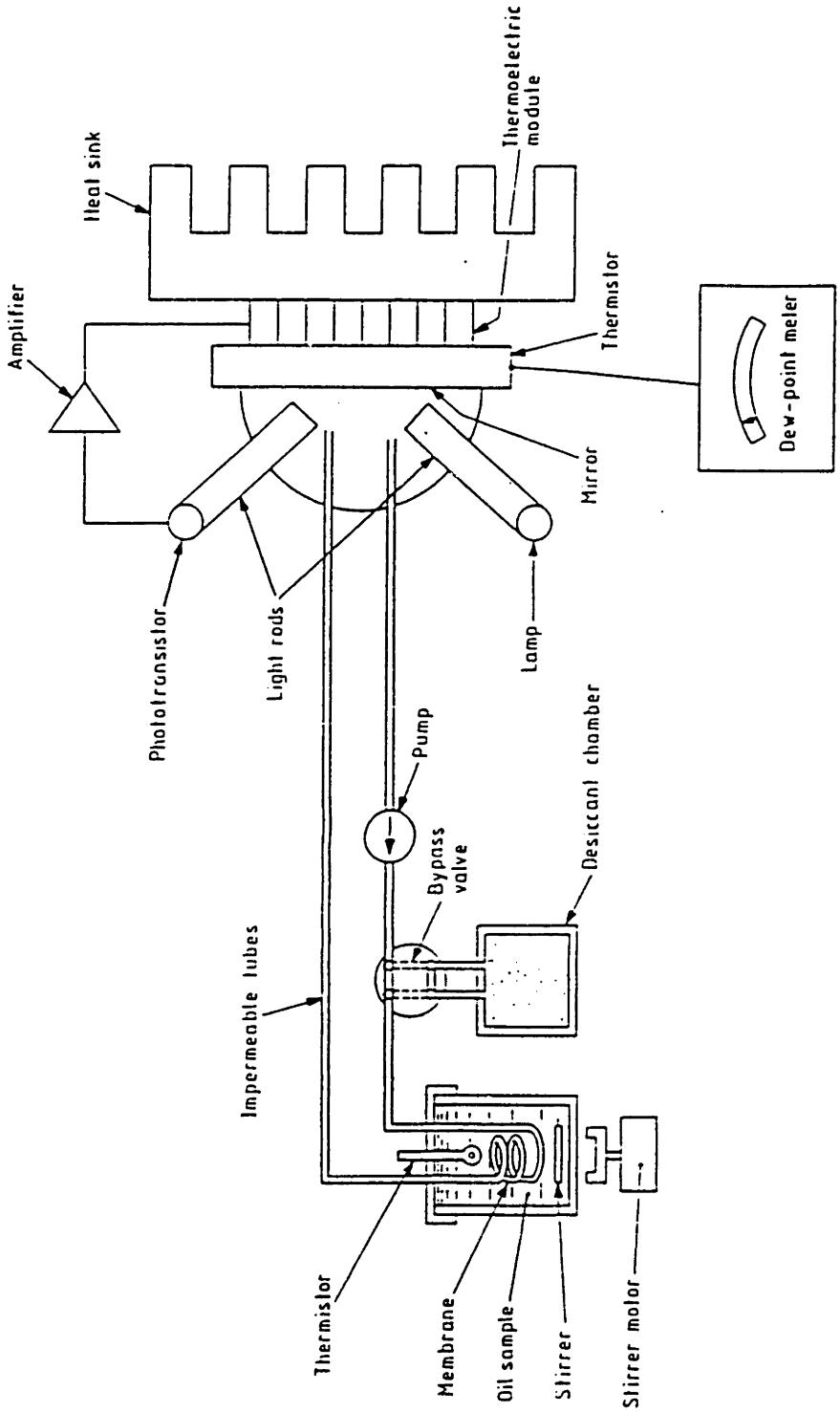


Figure 1.2: Apparatus for Measurement of Relative Saturation of Oil (from [12]).

FID and TCD – Flame ionization and thermal conductivity detectors are standard, nonselective gas detectors employed in gas chromatographs. In flame ionization the gas is consumed in the flame, generating a signal proportional to the amount of gas present. Thermal conductivity detectors measure the change in thermal conductivity of the carrier gas due to the presence of another gaseous species. Due to the lack of selectivity, both detectors require gas separation, usually accomplished by passing the gas mixture through one or more gas chromatographic columns.

Electrochemical Detectors – An electrochemical fuel cell for detecting hydrogen gas is commercially available [31]. The fuel cell is composed of platinum electrodes and an aqueous solution of H_2SO_4 (Fig. 1.3). In operation, hydrogen is oxidized at one electrode with a corresponding reduction of oxygen at the other electrode. The resulting current flow is monitored and related to the ambient hydrogen concentration. Industrial tests of this apparatus on current transformers [32] and HVDC valve damping resistors [33] are ongoing. In addition, this detector is incorporated into a transformer monitoring system being developed at Westinghouse [34]. As of this writing, this detector is not used with a gas separation section although it is not completely selective. The problem is the lack of selectivity in distinguishing between several combustible gases. The detector will also respond, to some extent, to carbon monoxide, acetylene and ethylene but expresses the result as an equivalent concentration of hydrogen.

Another type of electrochemical device for detecting hydrogen gas is a diffusion cell [35]. A proton conducting polymer is sandwiched between two noble metal electrodes (Fig. 1.4). The potential developed across the polymer film is related to the partial pressure of hydrogen on either side of the detector. For the specific device shown in Fig. 1.4, the Pd electrode has been saturated with hydrogen forming a palladium hydride reference electrode. Thus, the voltage can be directly related to the ambient hydrogen concentration. A major problem with this device is the stability of the reference electrode. It is very difficult to prevent hydrogen diffusion from the reference over an extended period of time. This renders the device impractical at the present time for on-line hydrogen gas monitoring.

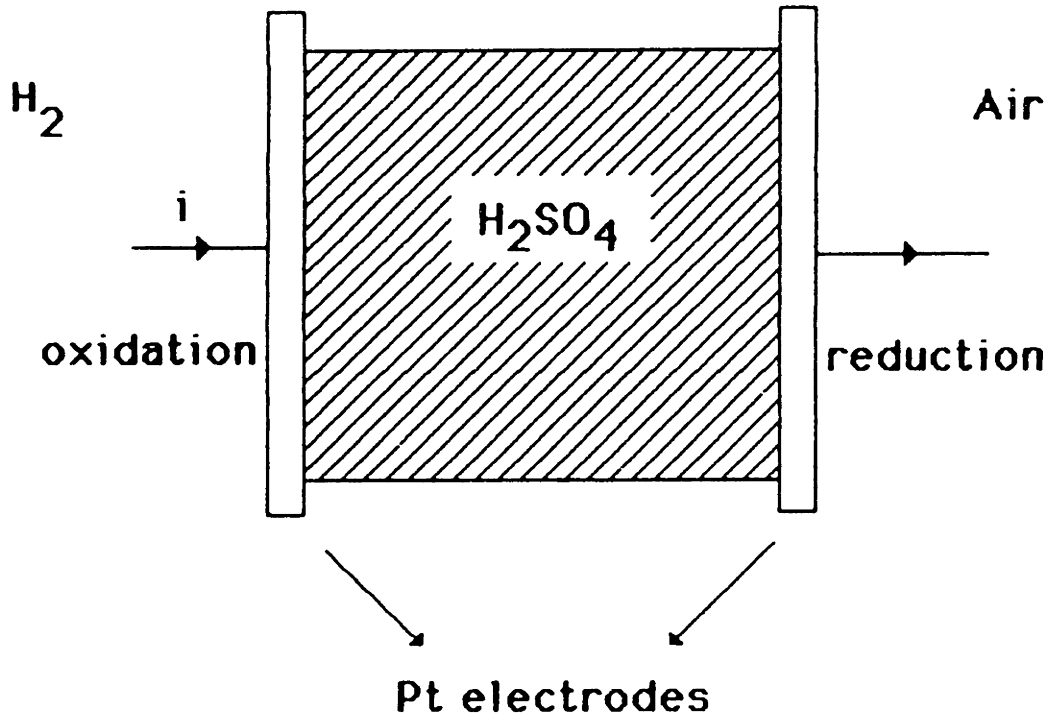


Figure 1.3: Electrochemical Fuel Cell for Detecting Hydrogen Gas (from [31]).

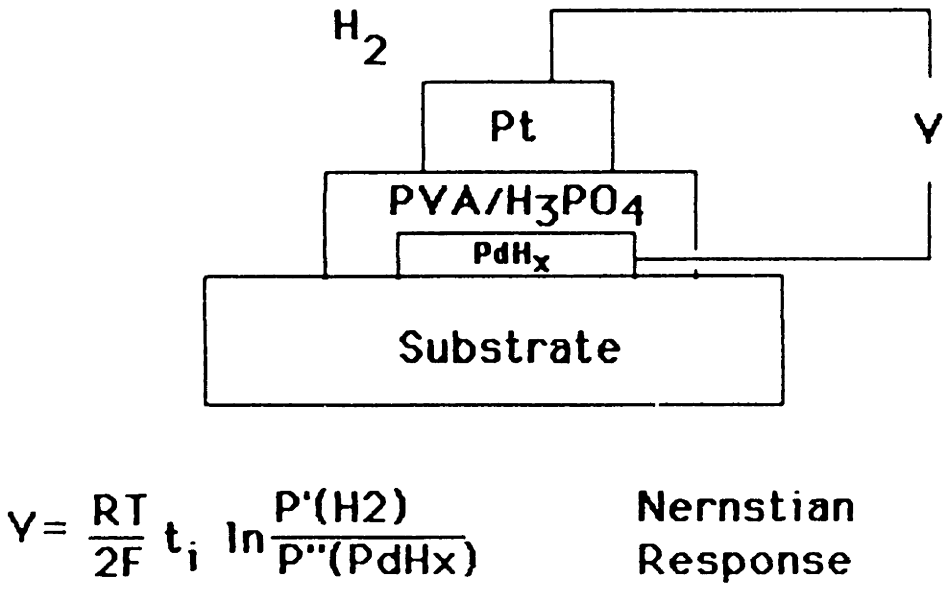


Figure 1.4: Diffusion Cell for Detecting Hydrogen Gas (from [35]).

Solid State Detectors – This category can be divided between structured and unstructured devices.

Unstructured devices are doped metal oxides and come in many varieties: 1) pellets or thick films of sintered tin oxide doped with a noble metal such as palladium or rhodium and 2) thin films of sputtered tin oxide. The most popular commercially available device is the Taguchi Gas Sensor (TGS) which is of the first variety. The TGS, and other devices of this kind, is prepared from a paste of sintered tin oxide powder doped with palladium [36]. A wealth of empirical information on preparation of and experimentation with these devices can be found in [37]. An attempt at quantitatively characterizing the operation and response of these devices is presented in work by Clifford and Tuma [38],[39]. Work has been done using sputtered thin films of tin oxide (< 5000 Å) for sensing hydrogen and hydrogen sulfide [40], ethanol [41], nitrogen oxide [42] and carbon monoxide [43].

In operation, the DC conductivity of tin oxide gas detector is altered by the presence of reducing gases such as hydrogen, carbon monoxide or hydrocarbons. The physical mechanism most commonly used to explain this result is the reducing gas removes oxygen adsorbed on the grain surfaces (Fig. 1.5). This removal lowers the barrier potential existing across grain boundaries, freeing up electrons for conduction. As a result, the conductivity of the device is increased. The presence of the noble metal dopant serves as a catalyst to enhance the surface reactions.

For proper operation there are several provisos. Oxygen must be present to ensure reversibility. The devices are always operated at elevated temperatures ($> 150°C$) to eliminate problems with moisture adsorption and to improve the adsorption/desorption kinetics for the gases of interest. Any reducing gas present will affect the measured conductivity to some extent. Therefore, these detectors are inherently nonselective and are usually used in situations where only one gaseous component present is required to be monitored in a relatively inert background. The selectivity can be improved by altering the doping concentration, materials and the preparation process but it is a highly empirical approach. An alternative is to use the operating device temperature to improve selectivity. The temperature dependence of the adsorption/desorption ki-

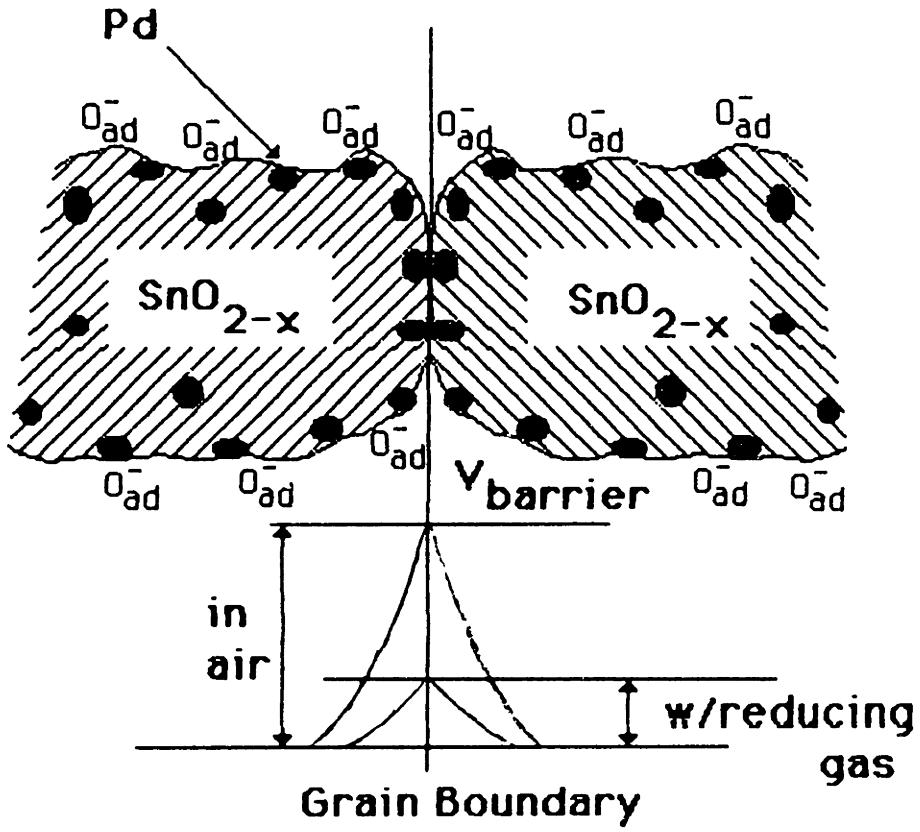


Figure 1.5: Effect of Reducing Gas on Work Function ($V_{barrier}$) at the Grain Boundary of Pd Doped SnO_{2-x} Having Surface Adsorbed Oxygen.

netics will be different for each gaseous species. By ramping the temperature of the device, each gas will be driven off at a different temperature, providing a gas separation analogous to that obtained in gas chromatography. The individual effect of each gas upon the conductivity is then monitored, providing (ideally) individual gas concentrations.

Structured devices involve microfabricated structures such as diodes, capacitors and transistors as shown in Fig. 1.6. In these devices the gas sensitivity is due to interfacial phenomena occurring at either the Pd–oxide or the oxide–doped semiconductor interface and manifests itself in changes in i–V, C–V or gate threshold voltage characteristics. Most commonly, it is the sensitivity to hydrogen that is studied [44], [45] although carbon monoxide [46], [47] and hydrocarbon gas sensitivity [48], [49] have also been reported.

The net effect of the pertinent gas is attributed to an alteration of the electrochemical work function (barrier potential) existing across the appropriate interface. On a device level, this translates into a change in the voltage drop across the device that shows up directly in the C–V and threshold voltage measurements [48], [50]. For the diodes, the diode current is an exponential function of this voltage, thus altering the entire i–V curve shape [51].

The oxide layer thickness plays a very important role in determining the location of the interface at which the gas sensitivity occurs. Lundström and Svensson [52] maintain that the change in work function, for devices having oxide layer thicknesses greater than roughly 100 Å, is due to the formation of atomic hydrogen dipoles at the Pd–oxide interface. For thin oxide layers, Keramati and Zemel propose that the hydrogen diffuses through the Pd and the oxide layer and cause changes in the interfacial trap density at the oxide layer–doped silicon interface [44], [53]. The dominant physics is very dependent on the structure under examination with regards to materials, purity and perhaps most relevant here, oxide layer thickness. The mechanism of interfacial trapping is important for thin (< 100 Å) oxide layers [54].

Alteration of the Pd layer thickness is also very important. For Pd layers thicker than 100Å, forming a continuous film, only small molecules such as hydrogen can dif-

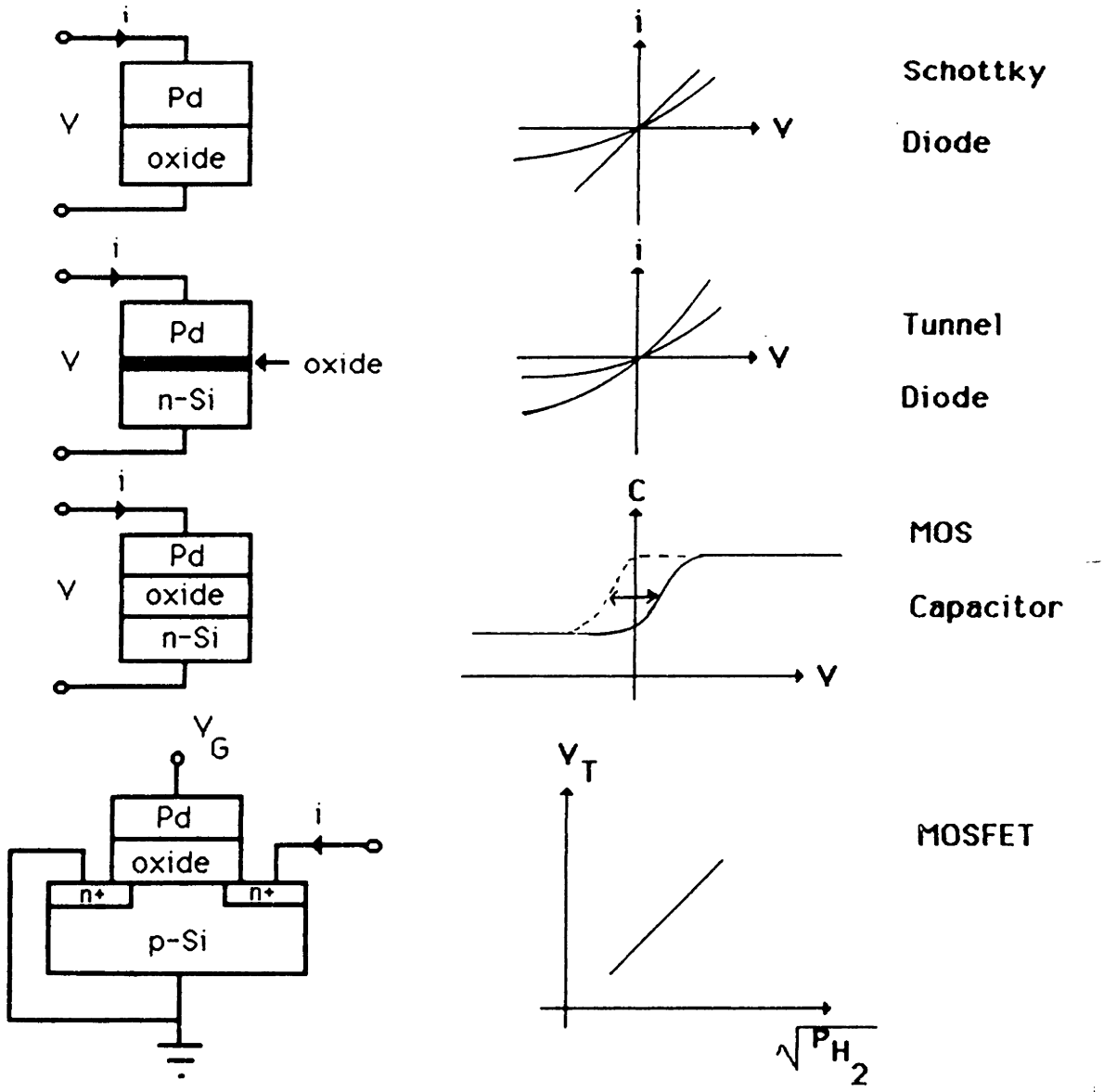


Figure 1.6: Microfabricated Gas Sensing Diodes, Capacitor and Transistor Utilizing Pd-Oxide Interface.

fuse through to reach the appropriate interface. This enhances the device selectivity to hydrogen or gases composed of hydrogen. Sensitizing the device to larger molecular gases such as carbon monoxide has been demonstrated by using either ultrathin, discontinuous Pd films [47] or masking the Pd deposition process so as to produce a porous film with holes (channels) to the oxide layer interface [46].

Typically these devices are also operated at elevated temperatures ($> 150°C$) both to enhance adsorption/desorption kinetics and to eliminate the effects of moisture. The presence of oxygen is again of great importance. With the strong emphasis on surface reactions at either the Pd or oxide interface, any other species that either takes up adsorption sites or reacts with surface adsorbed molecules can significantly alter device performance. For example, in an inert atmosphere such as argon, introduction of hydrogen produces a large, rapid response but the subsequent removal of hydrogen yields a slower change with an irreversible component. In the presence of oxygen, the devices are reversible but the gas sensitivity is reduced due to the reaction of hydrogen and oxygen at the Pd surface. These class of devices have not been made commercially available yet. There are problems with long term stability due to sulfur poisoning [45] and hydrogen drift into the oxide layer [55].

Gas Analyzers

As pointed out several times above, most of the gas detectors are not very selective. Thus, the additional step of gas separation must be performed prior to detection. A logical approach, taken by a group in Japan, has been to automate the standard gas extraction/chromatography process [56]. An oil sample is extracted and degassed by pulling a vacuum. The gas mixture is then sent through chromatographic separation columns and then to the FID and TCD detectors. Accompanying this dissolved gas analysis unit was a total combustible gas (TCG) unit. This unit used the same gas extraction apparatus but only measured total combustible gas content using a catalytic combustion (TGS) type gas sensor. Various monitoring strategies using the two units were proposed and some field testing performed. Though the units performed satisfactorily in the field (two samples per day) they are rather large, contain much

automated valving, require maintenance of the gas separation columns and periodic replenishing of the carrier gas cylinders, take one or two hours per sample and must store the oil samples removed from the transformer. If it is required to monitor on a timescale of minutes or even perform several samples per day then the units will consume their resources rapidly and remove a significant quantity of oil.

Most of the effort has been towards developing new techniques for gas extraction and detection. The common alternative in gas extraction is to expose a polymer membrane, usually Teflon or a similar fluoro–polymer, to the circulating oil. The gases diffuse out of the oil, through the membrane, are collected in a chamber and then, if DGA is required, separated using a conventional gas chromatography column. Correlation between gas–in–oil concentrations and detector responses require sampling after time is allowed for the partial pressure of a gas to equilibrate on both sides of the membrane. This equilibration time is a function of the solubility in oil and membrane permeability of the particular gas to be detected. Further, both of these quantities are temperature dependent. Equilibration times typically can range from 10 hours for hydrogen to 50 hours for ethane (depending on the actual physical dimensions of the membrane and collection chamber) [57]. This criteria, then, imposes an upper limit on sampling frequency that may or may not be acceptable.

The major point of divergence occurs at the gas detection end. A group in the UK has used a flame ionization detector, common in gas chromatography, combined with a silicon rubber membrane used for gas extraction [30]. Samples are taken every 40 minutes – no mention was made of equilibration times. Along similar lines, a group at Westinghouse has used a thermal conductivity detector, also common in gas chromatography [58,59]. This device samples once every 24 hours. Based on the field experience of [56] the thermal conductivity detector is more stable than the flame ionization detector. Regardless, both techniques have many of the same drawbacks mentioned above with the exception of those associated with oil removal.

Another type of detector is a combustion semiconductor gas sensor similar to those used for monitoring emissions from automobiles or industrial processes for pollution control, fire safety, or health reasons. Though not well described in papers on gas

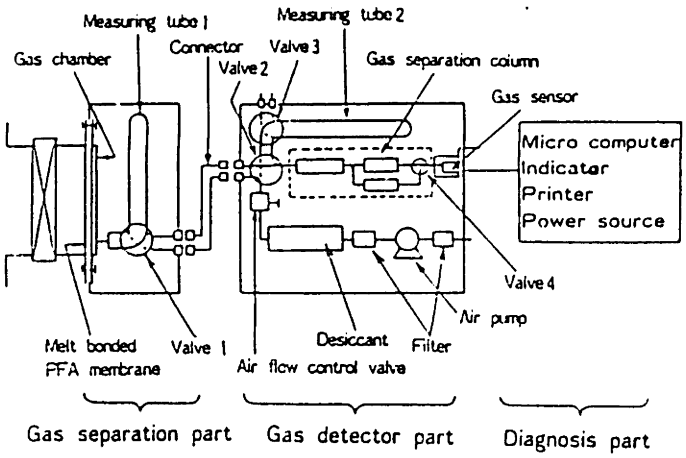


Figure 1.7: Hitachi On-Line Gas Sampling and Monitoring System (from [62]).

monitoring systems for transformers developed at the Hitachi Labs [60,61] the detector used is believed to be similar in operation to the TGS devices previously discussed. In the Hitachi system, gas separation is performed using chromatographic columns and a carrier gas of filtered air [62] (see Fig. 1.7). This necessitates much valving, an air pump, and accompanying air filters to clean and dry the air. Demonstration of the system on field transformers compared well to oil samples analyzed off-line over a period of a year.

An alternative approach to gas analysis is Fourier Transform Infrared (FTIR) spectroscopy. In this optical approach to gas sensing, information about a gaseous mixture is obtained in the infrared portion of the spectrum where rotational-vibrational modes are excited. This technique can be used to resolve CO, CO_2, and hydrocarbon gases with an accuracy on the order of parts per million without the need for physically separating the individual gases from the mixture. It will not detect symmetric gas molecules such as H_2, N_2 or O_2. Additionally, the measurement can be performed in less than a second allowing very rapid monitoring. Gas extraction from the oil would then become the rate limiting step in performing gas analysis. There has been experience in monitoring air quality in process plants, sampling from several different locations, using an FTIR spectrometer [63].

1.4 Microdielectrometry

1.4.1 Description and Background

Microdielectrometry, as developed by Senturia and co-workers [64,65], is a commercially available technique for measuring complex permittivity utilizing microfabrication technology to incorporate both the sensing electrodes and associated circuitry on the same microchip (Fig. 1.1) [66]. A set of interdigital, planar electrodes are deposited on an insulating oxide layer. In addition to the interdigital electrodes, two FET transistors are also fabricated on the microchip. The electrical excitation and response is obtained using a microdielectrometer, specifically a Eumetric SYSTEM II Microdielectrometer [67]. One set of electrodes is driven by a variable frequency (.005 – 10,000 Hz) AC voltage. Digital processing is used for both the synthesis of the driven voltage and the measurement of the response. There are 58 discrete frequencies available for the driven voltage. The response is the magnitude and phase of the complex ratio of the potential developed at the second electrode to the driven voltage. This electrode is allowed to float by connecting it to the gate of one of the transistors. The other transistor serves as a reference. Using the unity gain feedback circuit shown in Fig. 1.1, in conjunction with the transistors, the floating gate potential is reproduced at the gate of the reference FET. Thus, a very high impedance measurement of the floating gate voltage can be obtained, even at very low frequencies. Having two transistors on the same chip automatically compensates for variations in transistor characteristics due to changes in the local environment such as temperature or pressure. The output of the device, defined as the complex gain $\hat{G}$, is the complex ratio of the floating to driven gate voltage, $(\hat{V}_F/\hat{V}_D)$, and is obtained using a correlation analysis of the two voltages. This analysis also provides the value of any DC component of the signal, referred to as the DC offset. DC offset can arise from: 1) a mismatch in characteristics of the two FETs and 2) static charge buildup either on the electrodes or in the region sensed by the electrode array. An in-depth analysis of the system design and performance of low frequency dielectric measurement systems, with the microdielectrometer as a specific example, can be found in a dissertation by Coln [68].

Note that the upper limit on frequency for the microdielectrometer of 10 KHz is not a fundamental one regarding the electroquasistatic field approximation. There is no reason why the range could not be extended upwards into the megahertz, and possibly even gigahertz region, as long as the properties of the insulating oxide layer are appropriately taken into consideration. The electroquasistatic approximation will be valid for frequencies $f < (c/\lambda)$ where $c = \sqrt{1/\epsilon_o \mu_o}$, the speed of light in free space. For the microchip, the critical dimension is the length of the electrodes, on the order of 2 mm. The electrodes must be regarded as equipotentials along this length, implying a minimum electric field wavelength of 10–100 times greater. This yields a cutoff frequency on the order of 10^9 Hz. However, this cutoff can be lowered considerably if the inductance of the particular electrode structure used resonates with the electrode capacitance at a lower frequency.

Two specialized models have been developed for interpreting data obtained with the microdielectrometer. One model is for the application of the microdielectrometer to the monitoring of epoxy resin cures [69,70]. A microchip is embedded in a sample of epoxy and the change in its dielectric properties with curing is monitored. This model, incorporated in the form of a lookup table supplied with the microdielectrometer system [67], makes it possible to relate the measured complex gain to the complex permittivity of a uniform infinite half space. A relationship between the dielectric properties and the curing process was established. For this case, a finite difference simulation of Laplace's equation was used to determine the electric field distribution, treating the region above the electrode structure as a semi–infinite, isotropic medium [70, Chapter 3]. The second model is for the application of developing a humidity sensor. In this case a thin film (< 1 μm) was deposited onto the electrode structure and the change in its sheet resistance as a function of the relative humidity in air was monitored [71,72]. For modeling purposes, the thin film's influence was taken to be predominantly in the interelectrode region. Thus, the thin film–oxide layer was represented as an R–C transmission line shunted by lumped capacitors [71]. This made it possible to represent the gate voltage as the transmission line response to a driving voltage. Comparisons with both of these models will be made in Section 3.3.

1.4.2 Micro and Macro Sensors

Presently, the interdigitated electrode structures are available in two different sizes. The microchip has an aluminum electrode structure of spatial wavelength (λ) of 50 μm deposited on a thin SiO_2 layer roughly 1.5 μm thick and having a permittivity of $3.9\epsilon_o$. A ceramic macrochip, developed for high temperature applications, is also available [74]. This ceramic macrochip, produced using thick film technology, has PD–silver alloy electrodes of 1 mm spatial wavelength deposited on an $A\ell_2O_3$ layer of permittivity $9.8\epsilon_o$ and 250 μm thick. There are no on–board electronics with the macrochip. The longer spatial wavelength allows dielectric measurements to be made on a larger length scale (up to 330 μm). A detailed discussion of the macrochip, including its design, operation and experimental results, can be found in a master's thesis by Li [23]. References to Li's work will be frequently made throughout this thesis and is viewed as a companion document to this one. An important contrast between the micro and macrochip is the decreased sensitivity of the macrochip to changes in the dielectric medium above the electrodes. This is due to both the higher permittivity and smaller ratio of spatial wavelength to layer thickness of the oxide layer for the ceramic sensor. These parameters will be discussed further in Section 3.1.5. Also, fringing fields are a greater problem with the macrochip due to the smaller aspect ratio of the electrode fingers.

For an uncoated microchip, the gain varies from roughly –40 dB in air to almost 0 dB when the electrodes are shorted together. Due to a slight mismatch in transistor characteristics the latter measurement is usually 0 ± 0.2 dB. The phase varies between 0° and roughly $-90°$, except for certain situations described in Sections 3.1.2 and 3.1.3 where the phase can go more negative. Typically, DC offsets are negative and on the order of a couple of hundred millivolts. There are various instances under which the DC offset will change significantly and these are described in Sections 5.4 and 7.1.8 for uncoated and coated microchips in oil, respectively. Generally, large changes in DC offset occurred when low moisture content oil was flowing by a microchip. Increased oil flow rates and temperature exacerbated this problem. Uncoated microchips were much more susceptible to this problem than coated ones. For DC offset information

concerning the macrochip, consult Li's thesis [23, Section 2.1.2].

1.4.3 Preliminary Work

The early work on developing a continuum model representation of the microchip and in gaining practical experience of using an uncoated microchip in transformer oil was performed by Mouayad [73].

Experimentally, it was observed that the response of the microchip was sensitive to adsorption of moisture at the surface of the silicon dioxide insulating layer, particularly if this surface was dirty. Adsorption of moisture at the surface created a surface conductance between the electrodes, thus altering the response. This effect occurred both in air and in moderately wet oil (20–30 ppm). Therefore an uncoated microchip would not be well suited as a sensor unless the surface adsorption and resulting conductance was well controlled. This will be discussed further in Section 5.4.

Evidence for polarization at the microchip-liquid interface was observed in experiments where the conductivity of the oil was varied through the addition of an antistatic agent. Transformer oil is highly insulating, typically having a conductivity of 10^{-13} to 10^{-12} S/m. Double layer thicknesses resulting from charge separation at solid/liquid interfaces vary with $1/\sqrt{\sigma}$ and can be significant in size with respect to the spatial wavelength of the microchip. Stirring of the oil altered the measured response. It was believed that the structure of the diffuse double layer was affected by the fluid motion. These results will be presented in Appendix D.

1.5 Goals

1.5.1 Analytical and Numerical Tools

The first task is to develop a generalized continuum model of the electrostatic field distribution about an interdigital electrode structure. This will allow simulation of the wide variety of physical situations in which the microchip might be used. In practice, not all the parameters required to completely specify the model are known. A key

parameter, such as the complex permittivity of a medium under investigation, will be unknown. By incorporating the continuum model into various parameter estimation routines, interpretation of experimental data will be possible. Both single and multi–parameter estimation routines will be developed for estimating quantities such as layer thickness, complex bulk and surface permittivities and spatial distributions of these properties.

1.5.2 Experimental Verification of Model

Experimental verification of the continuum model and parameter estimation routines must be performed to gain confidence in the ability to make absolute measurements. Measurements of thickness and complex bulk permittivity, as obtained with the interdigital electrode structures, and the continuum model, will be compared to independent measurements using standardized techniques. The development of the continuum model allows the microchip to be used for several new applications. One fairly straightforward development is to use the microchip as a thickness gauge for thin films formed by deposition, evaporation or sedimentation of material.

1.5.3 Electrochemical Sensors For Transformer Monitoring

The primary task is to adapt the microchip and macrochip structures for transformer oil monitoring. The motivation is the potential advantage of monitoring the oil almost instantaneously with something as simple and cheap as a microchip. The simplest measurement, that of the complex bulk permittivity of the oil, is not a straightforward measurement because of problems encountered with moisture adsorption by the electrode structures. Passivation of the microchip and macrochip is a critical step in making this simple measurement. This passivation procedure, coating the electrodes with a moisture impermeable membrane, makes necessary the continuum model to extract the correct complex bulk permittivity from the experimental data. While there has been no conclusive links made between specific degradation processes and dielectric data, the frequency information available with microdielectrometry may shed new

light on the situation.

Next, it is desirable to be able to monitor the moisture and dissolved gas content of the oil. The key is to sensitize the microchip to a particular contaminant. This will be achieved by coating the microchip with a thin film that preferentially ad- or absorbs the contaminant of interest and exhibits a change in its dielectric properties proportional to the contaminant concentration. Again, the continuum model is required to correctly extract the dielectric properties when using a coated microchip. In addition, parameter estimation of the dielectric properties over the entire frequency range can help in forming a physical model of the coating structure and in understanding the basis for the sensitivity to the specific contaminant.

Efforts will be concentrated on developing a moisture sensor and preliminary development of a hydrogen gas sensor, as there is previous evidence in the literature of materials sensitive to these two constituents. Passivation of the microchip surface is an essential step in developing gas sensors and is also important for developing a moisture sensor.

1.6 Overview

The first part of the thesis concerns itself with developing and verifying the continuum model, and parameter estimation routines, necessary for applying the microdielectrometer to transformer monitoring. A derivation of the continuum model is presented in Chapter 2. In Section 3.1, a rapport is developed for the typical responses of the microdielectrometer when used to measure homogeneous media, thin films, surface conductivities and/or complex bulk permittivities. Some general working theorems for interpreting gain–phase responses are elaborated in Section 3.2. A brief comparison with other models, for various physical limits, is presented in Section 3.3. Development of the parameter estimation routines is covered in Chapter 4. Section 4.2 deals with single parameter estimates of thickness or complex permittivity. Multiparameter estimation of spatial distributions of dielectric properties, as obtained by varying the spatial, rather than the temporal frequency, is examined in Section 4.3. Two param-

eter estimation of layer thickness and permittivity by varying the dielectric medium above the layer is discussed in Section 4.4. Experimental verification of the continuum model and parameter estimation routines is presented in Chapter 5. Calibration of the microchip and macrochip is described in Section 5.1. The specific application of using the interdigital electrode structure as a thickness gauge, when configured in both the low and high impedance modes, is discussed in Sections 5.2.1 and 5.2.2. The effort towards developing a sensor to monitor the complex bulk permittivity of the oil are outlined in Section 5.4, including results obtained using sensors passivated with parylene. Estimation of complex surface conductivities from experimental observations is performed in Section 5.5.

The specific application of adapting microdielectrometry to the monitoring of moisture in transformer oil is the subject of Chapters 6, 7 and 8. Chapter 6 outlines the potential increase in understanding of the condition of the paper insulation and oil degradation via on–line moisture monitoring. A design strategy for sensitizing the microchip to the moisture content in oil is given in Section 6.3. This sensitization is accomplished through the plasma deposition of a thin film of bromobenzene onto the microchip electrodes. Experimental development and testing of these microchip moisture sensors is presented in Chapter 7. Section 7.1 gives the experimental results, obtained in controlled oil samples, used to characterize the sensitivity of the moisture sensors. Results from testing of the moisture sensor in the oil circulation loop of an instrumented pole type (50KVA) transformer are given in Section 7.2. An investigation of the electrical and physical properties of the plasma deposited bromobenzene films is conducted in Chapter 8. The biphasic structure of the film, particles embedded in a rigid film, is examined in Section 8.1. This structure is examined with respect to the moisture permeability of the films and is related to evidence in the literature on similar film structures. In Section 8.2, the dispersive nature of the dielectric properties of plasma deposited bromobenzene is revealed and shown to follow a power law dispersion, common to many amorphous structures. Experience from the literature on film aging, and the measures that can be taken to increase film stability, are summarized in Section 8.3.

Chapter 9 summarizes preliminary attempts developing a hydrogen gas sensor. After motivating this effort in Section 9.1, inconclusive results using microchips coated with tin oxide and palladium are given in Section 9.2. Encouraging results using thin film palladium resistors are shown in Section 9.3.

Finally, conclusions and an outline of possible future directions are contained in Chapter 10.

In Appendices A and B, mathematical derivations are presented in greater detail. Appendix C documents the experience with the plasma deposition apparatus. Experimental results examining double layer effects on the microchip response in doped oil are presented in Appendix D. Data acquisition software used to take frequency and time scans with the microdielectrometer is documented in Appendix E. All the parameter estimation software, along with the continuum model, is presented in Appendix F.

Chapter 2

Continuum Model

2.1 Outline of Approach

The objective here is to develop a general and adaptible model for determining the output of the interdigital electrode dielectrometer. In Fig. 2.1 is shown a schematic view summarizing the structure of the continuum model developed in the next section. One set of inputs to the model, the lower box on the left in the diagram, describes the electrode array structure. Parameters such as interelectrode spacing (a) and the spatial wavelength (λ) are specified here. In addition, for the specific electrode structure considered here, interdigital electrodes deposited on an oxide layer with ground plane underneath, the electric field distribution below the electrodes is completely specified by the insulating oxide layer thickness (h) and permittivity (ϵ_{ox}). The other input, the upper box on the left, describes the half space of media above the electrodes. This description is in the form of a complex surface capacitance density $\hat{C}_n^a$, representing the response of the half space to one fourier component of a potential applied at the electrodes. As shown in the upper part of the figure, this response is the complex ratio of the normal displacement field ($\hat{D}_n^a$) measured at the interface to the interfacial potential ($\hat{\Phi}_n^a$). The $\hat{}$ signifies a complex quantity, the superscript indicates that the function is to be evaluated just above the interface and the subscript indicates the nth fourier component. The quantity $\hat{C}_n^a$ will be dependent on the particular medium under investigation. All the heterogeneity and structure of the medium is encapsulated

in $\hat{C}_n^a$. Once these inputs have been defined the fields are completely specified.

The '$\hat{V}_j$' box represents the step of solving the electrostatic field problem. Field quantities in the oxide layer and in the medium are represented in terms of their fourier components. Boundary conditions at the electrode interface are used to match up the field solutions for the regions above and below the electrodes. A mixed boundary value problem occurs at this interface, as the potential is constrained along that part occupied by the electrodes while conservation of charge and Gauss' law combine to put a constraint on the potential and on the jump in its normal gradient along the interelectrode part of the surface. The solution involves discretizing the interelectrode surface by introducing a grid of unknown voltages $\hat{V}_j$ at k collocation points. The potential is assumed to vary linearly between these collocation points. A system of equations is generated by maintaining a discretized conservation of charge along surface segments bracketing these collocation points. These equations are written in the form $A \bullet V = X$ where the elements of A are the coefficients of the unknown voltages $\hat{V}_j$ and the elements of X represent the known voltage, $\hat{V}_D$. Inverting the matrix A yields the voltage distribution along the interelectrode surface, and thus, the electric field distribution.

In the final box, the evaluation of the Y_{ij}'s is tantamount to a determination of the response with any terminal configuration. Due to the symmetry of the electrode structure, $Y_{11} = Y_{22}$. All of the admittances representing the electrode structure are determined by finding the electrical terminal currents, $\hat{i}_D$ and $\hat{i}_F$, with the floating gate electrode grounded. These currents are obtained by integrating the current density over the surfaces of the electrodes. Using these admittances, the response of the microdielectrometer with an arbitrary load is the complex gain, the ratio of the floating to driven gate voltage

$$\hat{G} = \frac{\hat{V}_F}{\hat{V}_D} = \frac{Y_{12}}{Y_{11} + Y_{12} + Y_\ell} \tag{2.1}$$

where Y_ℓ is the load admittance of the floating gate FET.

In the medium above the electrodes and in the oxide substrate below, the potential

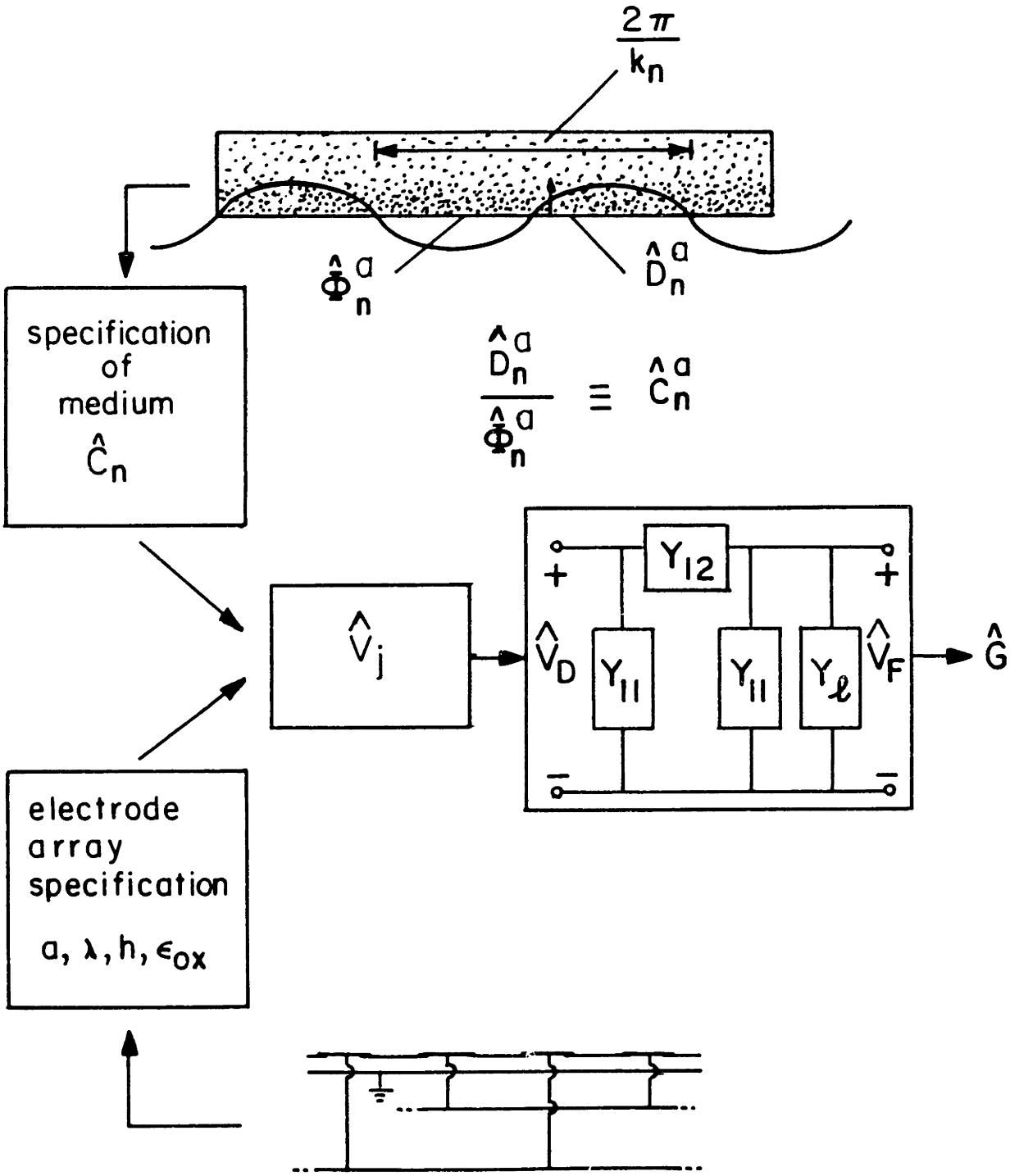


Figure 2.1: Schematic Representation of Continuum Model – Required Inputs, Solution for Collocation Potentials, Two Port Network Equivalent, and Complex Gain Output.

is represented by (1.1) and the fields are taken as electroquasistatic, the electric field intensity is irrotational and hence represented by the potential Φ.

$$\vec{E} = -\nabla\Phi \tag{2.2}$$

For completeness, the oxide layer will be given a complex permittivity, ϵ_b^*.

2.2 Electrode Structure Boundary Conditions

The electrode structure of Fig. 1.1 is modeled as shown in Fig. 2.2. The electrodes are treated as being one dimensional, one set driven sinusoidally with a peak voltage of $\hat{V}_D$ and frequency ω, the other set grounded. It is possible to include the finite thickness of the electrodes using the techniques outlined here but the effort is not judged to be worthwhile at present. For generality, an interelectrode spacing of arbitrary width a is allowed. Thus, in a half period, the endpoint of one electrode is at $y = y_0$ and the beginning of the other electrode is at $y = y_0 + a \equiv y_{k+1}$.

At the electrodes, the potential is constrained, whereas along the surface between the electrodes there is only continuity of the potential

$$\begin{aligned}
&\Phi^b(y,t) = Re\left\{\hat{V}_D e^{j\omega t}\right\} && \text{for } 0 \le y \le y_0 && \text{(a)}\\
&\Phi^b(y,t) = 0 && \text{for } y_{k+1} \le y \le \tfrac{\lambda}{2} && \text{(b)}\\
&\Phi^b(y,t) = \Phi^a(y,t) && \text{for } 0 \le y \le \tfrac{\lambda}{2} && \text{(c)}
\end{aligned} \tag{2.3}$$

On the interelectrode surfaces, Gauss' Law relates the discontinuity in the normal displacement field to the amount of free surface charge density, σ_{su}, ($\vec{n}$ is the unit normal to the interface)

$$\vec{n} \bullet [\![\vec{D}]\!] + \nabla_\Sigma \bullet \vec{D}_s = \sigma_{su} \tag{2.4}$$

where $[\![\ ]\!]$ implies the difference in the function within the brackets evaluated just above and just below the interface and $\nabla_\Sigma \bullet$ is the surface divergence. The surface electric displacement vector, $\vec{D}_s$, represents the possibility of polarization phenomena concentrated within the interface. Conservation of charge accounts for the accumulation

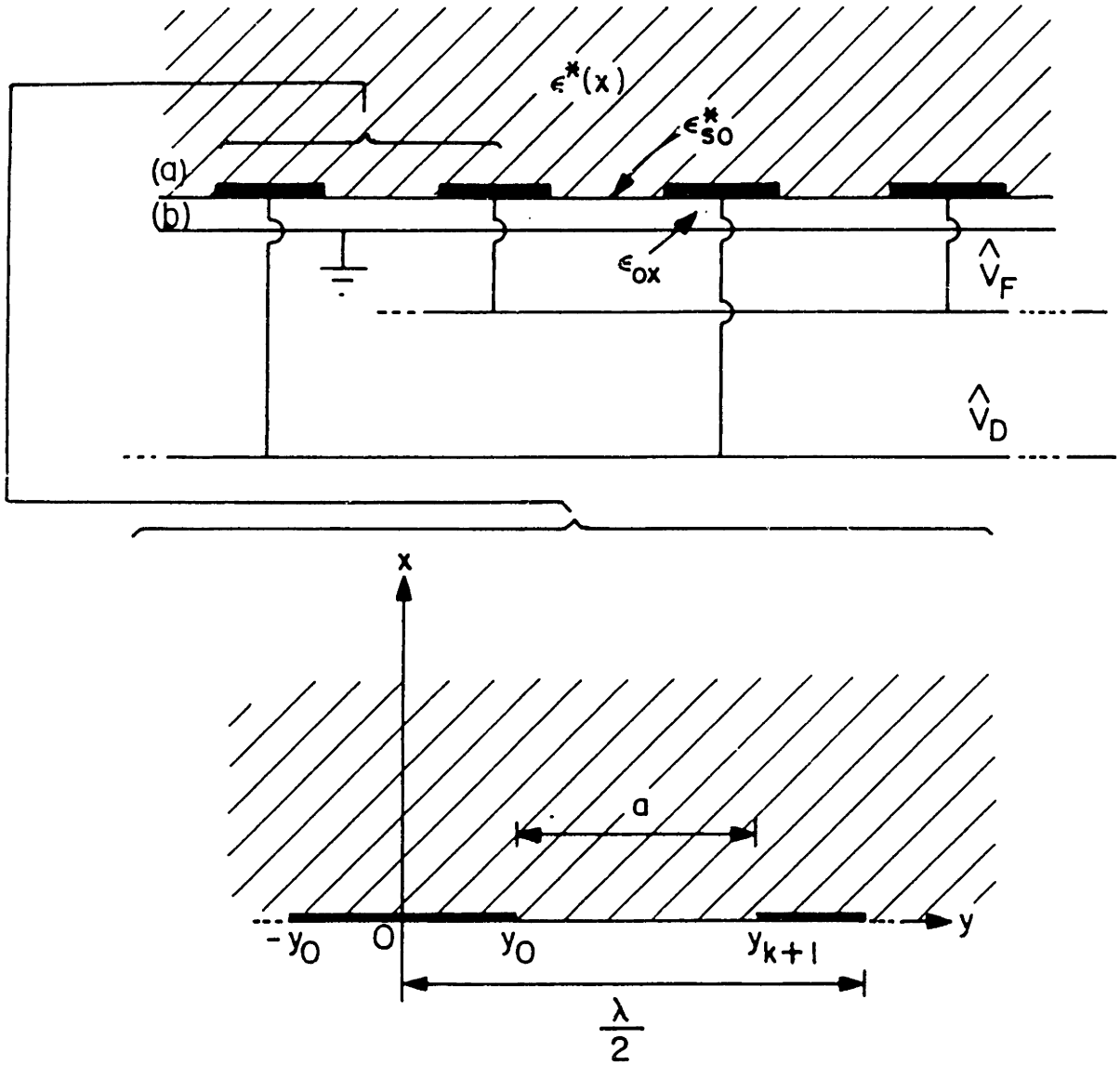


Figure 2.2: Coordinate System Used for Electrode Structure of Fig. 1.1.

of σ_{su} via the discontinuity in normal conduction current ($\vec{J}$), including conductivity within the interface represented by the surface current density $\vec{J}_s$,

$$\vec{n} \bullet [\![\vec{J}]\!] + \nabla_\Sigma \bullet \vec{J}_s + \frac{\partial \sigma_{su}}{\partial t} = 0 \tag{2.5}$$

Substituting (2.4) into (2.5), using the complex notation introduced by (1.1) and assuming sinusoidal steady state ($\frac{\partial}{\partial t} \rightarrow j\omega$) yields the following boundary condition

$$\vec{n} \bullet [\![\epsilon^* \hat{\vec{E}}]\!] + \nabla_\Sigma \bullet (\epsilon_s^* \hat{\vec{E}}_s) = 0 \tag{2.6}$$

where complex bulk and surface permittivities have been introduced, $\epsilon^* = \epsilon' - j\epsilon''$ and $\epsilon_s^* = \epsilon_s' - j\epsilon_s''$, and the conduction and displacement volume and surface current densities have been combined ($\vec{E}_s$ is the component of $\vec{E}$ in the interfacial plane),

$$j\omega\epsilon^* \hat{\vec{E}} = \hat{\vec{J}} + j\omega \hat{\vec{D}} \quad ; \quad j\omega\epsilon_s^* \hat{\vec{E}}_s = \hat{\vec{J}}_s + j\omega \hat{\vec{D}}_s; \tag{2.7}$$

For an ohmic conductor ϵ' and ϵ_s' are constant, and $\epsilon'' = (\sigma/\omega)$ and $\epsilon_s'' = (\sigma_s/\omega)$, where σ and σ_s are constant.

For uniform surface properties (2.6) yields the second interelectrode region jump condition,

$$\left[\epsilon_a^* \hat{E}_x^a(y) - \epsilon_b^* \hat{E}_x^b(y)\right] + \epsilon_{so}^* \frac{\partial \hat{E}_y^a(y)}{\partial y} = 0 \text{ for } y_0 \leq y \leq y_{k+1} \tag{2.8}$$

2.3 Bulk Relations

As discussed earlier, the field distribution in the half space of material either above or below the electrodes can be represented by the appropriate surface capacitance density. This quantity is the ratio of the the complex fourier amplitudes of normal electric displacement to potential, both evaluated on the (a) or (b) side of the interface.

$$\frac{\hat{D}_n^{a,b}}{\hat{\Phi}_n^{a,b}} = \frac{\epsilon_{a,b}^* \hat{E}_n^{a,b}}{\hat{\Phi}_n^{a,b}} \equiv \hat{C}_n^{a,b} \tag{2.9}$$

Here, the subscript n refers to the order of the fourier modes. Given the distribution of a complex permittivity $\epsilon_a^*(x)$ above the electrodes, with the coordinate x perpendicular to the interface (including surface singularities in that distribution), the complex surface capacitance density $\hat{C}_n^a$ is found by determining the Fourier amplitude of the normal flux density at the surface (a) caused by the nth term of the imposed potential having the form

$$\hat{\Phi}(x,y) = \sum_{n=0}^{\infty} \hat{\Phi}_n(x)\cos(k_n y) \quad ; \quad k_n = 2n\pi/\lambda$$

Examples of surface capacitance densities for various types of homogeneous and heterogeneous media are given in Section 3.1.

2.4 Potentials at Collocation Points

The Fourier amplitudes of (1.1) are

$$\begin{aligned} \hat{\Phi}_n^a &= \tfrac{2}{\lambda}\int_0^{\lambda} \Phi^a(y)\cos(k_n y)\,dy \quad \text{for} \quad n = 1,2\ldots \\ \hat{\Phi}_0^a &= \tfrac{1}{\lambda}\int_0^{\lambda} \Phi^a(y)\,dy \qquad\qquad \text{for} \quad n = 0 \end{aligned} \tag{2.10}$$

The potential, $\hat{\Phi}^a$, is known only at the electrodes. On the surface, in the interelectrode region, the potential assumes a distribution such that it is continuous, (2.3c), and conservation of charge, (2.5), is satisfied. The strategy taken here is to represent the interface potential by piece–wise continuous linear segments having the collocation potential $\hat{V}_j$ at each collocation end–point y_j. Illustrated in Fig. 2.3 are four collocation points, introduced in the interelectrode region between y_0 and y_{k+1}. An initial effort divided the region into equal length line segments [73, p. 33]. A cosinusoidal distribution has been utilized here to give finer resolution in the regions of higher electric stress (adjacent to the electrodes). Thus, the collocation points are at

$$y_j = (y_0 + \frac{a}{2}) - \left(\frac{a}{2}\right)\cos\left(\frac{\pi j}{k+1}\right) \quad \text{for} \quad 0 \le j \le k+1 \tag{2.11}$$

and the potential distribution is

$$\hat{\Phi}^a(y) = \begin{cases} \hat{V}_D & \text{for } 0 \le y \le y_0 \\ \dfrac{\hat{V}_j(y - y_{j-1}) - \hat{V}_{j-1}(y - y_j)}{(y_j - y_{j-1})} & \text{for } y_{j-1} \le y \le y_j \\ & j = 1, 2, \ldots, k+1 \\ 0 & \text{for } y_{k+1} \le y \le \frac{\lambda}{2} \end{cases} \tag{2.12}$$

Substituting for $\hat{\Phi}^a(y)$ into (2.10) yields the Fourier series coefficients

$$\begin{aligned} \hat{\Phi}^a_n &= \frac{\lambda \hat{V}_D}{(n\pi)^2} \frac{1}{(y_1 - y_0)} [\cos(k_n y_0) - \cos(k_n y_1)] \\ &\quad + \sum_{j=1}^{k} \frac{\lambda \hat{V}_j}{(n\pi)^2} \left[\left(\frac{1}{y_{j+1} - y_j} + \frac{1}{y_j - y_{j-1}} \right) \cos(k_n y_j) - \frac{\cos(k_n y_{j+1})}{y_{j+1} - y_j} \right. \\ &\quad \left. - \frac{\cos(k_n y_{j-1})}{y_j - y_{j-1}} \right] \\ \hat{\Phi}^a_0 &= \frac{\hat{V}_D}{\lambda}(y_0 + y_1) + \frac{1}{\lambda} \sum_{j=1}^{k} \hat{V}_j (y_{j+1} - y_{j-1}) \end{aligned} \tag{2.13}$$

The problem has been transformed to one of solving for k unknowns, the $\hat{V}_j$'s, with the continuity of potential satisfied by (2.3c). The second continuity condition at the interface, (2.8), remains to be satisfied. By dividing the interelectrode surface into k intervals, centered on the collocation points, and satisfying conservation of charge, (2.8), within every interval, k equations for the k $\hat{V}_j$'s are obtained. The intervals are demarcated by points y_j^* located at the edges of the electrodes and at the midpoints of the line segments formed by consecutive collocation points, y_j and y_{j+1} (Fig. 2.3).

$$y_j^* = \begin{cases} y_0 & \text{for } j = 1 \\ \frac{1}{2}(y_j + y_{j-1}) & \text{for } 2 \le j \le k \\ y_{k+1} & \text{for } j = k+1 \end{cases} \tag{2.14}$$

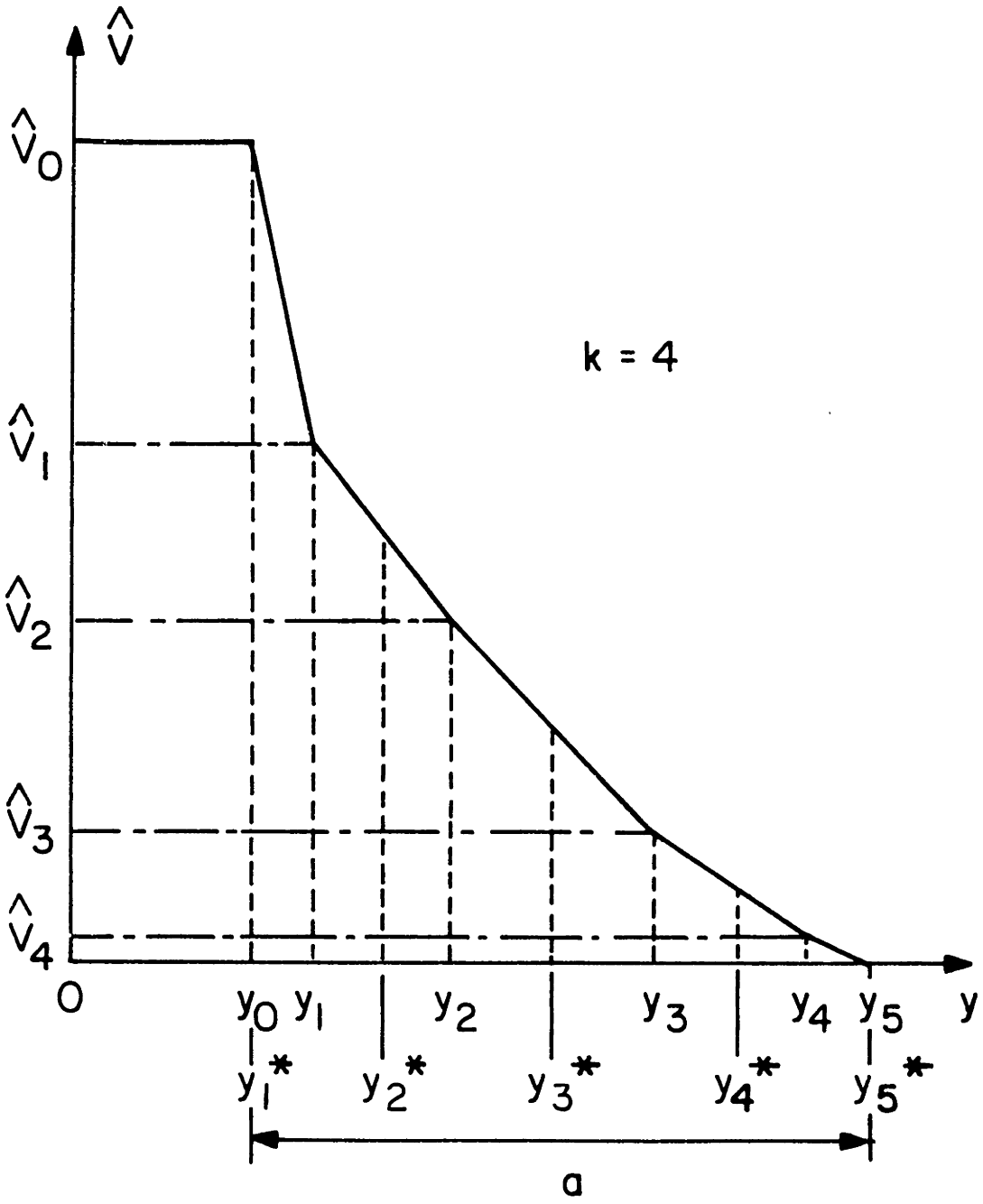


Figure 2.3: Voltage Distribution Along Discretized Interelectrode Surface for k=4 Collocation Points. Unknown Voltages V_j are Introduced at the Points y_j. Conservation of Charge is Maintained for Line Segments Defined by the y_j^*.

With this choice of y_j^*, the ambiguity in the value of the tangential electric field at the collocation points y_j is avoided. Using (2.2) and (2.12) yields

$$\hat{E}_y^a(y_j^*) = -\left(\frac{\hat{V}_j - \hat{V}_{j-1}}{y_j - y_{j-1}}\right) \tag{2.15}$$

Note that the tangential field is antisymmetric about the x-axis. As shown in Fig. 2.4, $\hat{E}_y^a(y)$ is a piecewise discontinuous function that is not well defined at the collocation points y_j. The intervals over which charge is conserved (between broken lines in Fig. 2.4) extend half a segment in either direction about a collocation point y_j except for the first and last intervals, which respectively have end points at the electrode edges. In addition, the values of the tangential field used for the endpoints y_1^* and y_{k+1}^* are not averaged though they are at a discontinuity. It is more realistic to assume that these values, for their respective segments, extend to the electrode as shown in Fig. 2.4.

Integrating (2.8) over each interval, from y_j^* to y_{j+1}^*, and setting the result to zero yields k equations for k unknowns,

$$\int_{y_j^*}^{y_{j+1}^*} \left[\epsilon_a^* \hat{E}_x^a(y) - \epsilon_b^* \hat{E}_x^b(y) + \epsilon_{so}^* \frac{\partial \hat{E}_y^a(y)}{\partial y}\right] dy = 0 \text{ for } j = 1, 2 \ldots k \tag{2.16}$$

The third term, reflecting contributions due to surface effects, integrates to

$$\epsilon_{so}^* \left[\hat{E}_y^a(y_{j+1}^*) - \hat{E}_y^a(y_j^*)\right] \tag{2.17}$$

and using (2.15) yields,

$$\epsilon_{so}^* \left[\left(\frac{1}{y_{j+1} - y_j} + \frac{1}{y_j - y_{j-1}}\right)\hat{V}_j - \frac{\hat{V}_{j+1}}{y_{j+1} - y_j} - \frac{\hat{V}_{j-1}}{y_j - y_{j-1}}\right] \tag{2.18}$$

Integrating the remaining two terms of (2.16), using the cosinusoidal dependence of the normal electric field upon x implied by (1.1) and (2.2), yields the set of equations

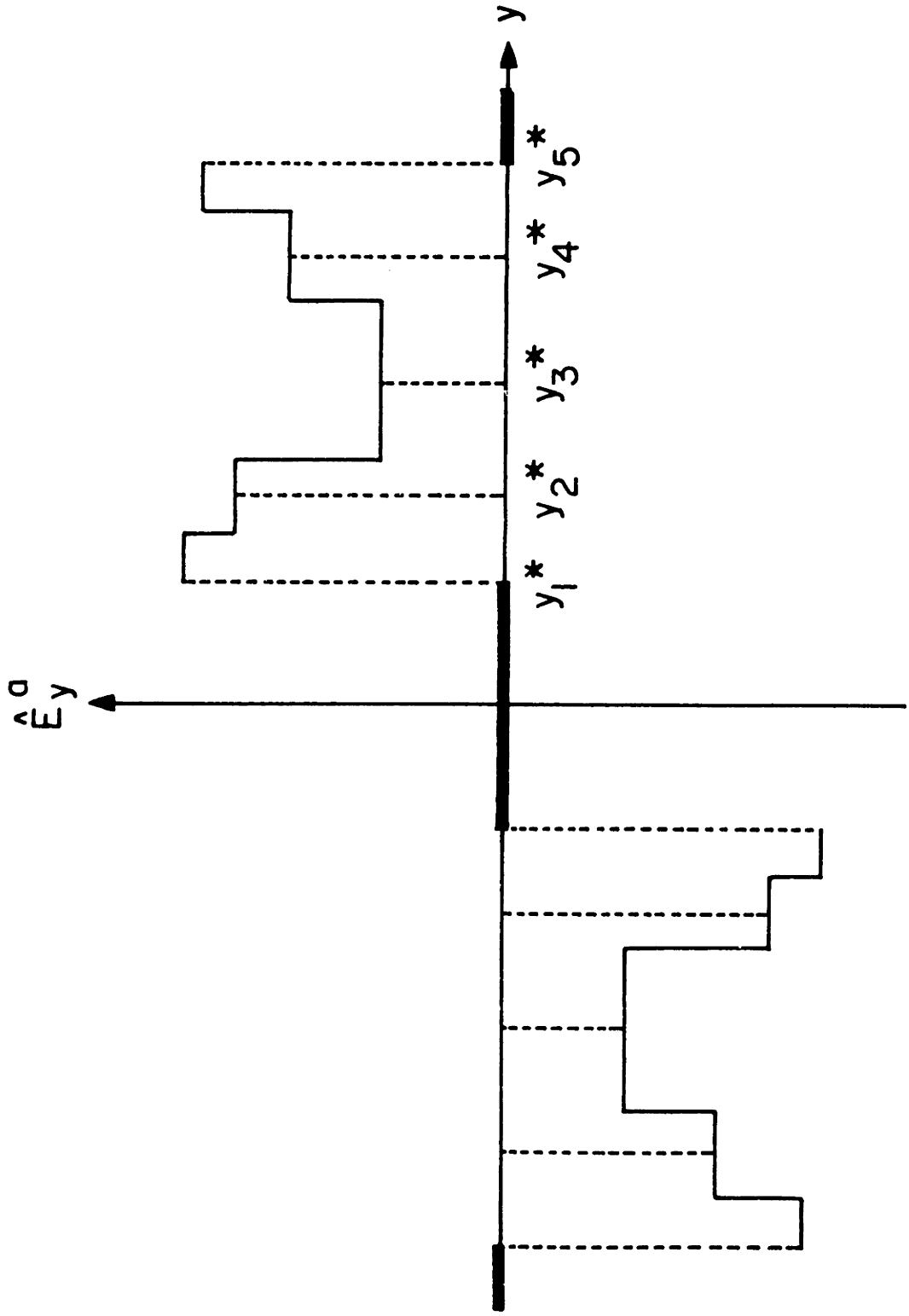


Figure 2.4: Distribution of Tangential Electric Field, $\hat{E}_y^a(y)$, Along Electrode–Medium Interface.

$$\sum_{n=1}^{\infty}\frac{1}{k_n}\left(\epsilon_a^*\hat{\mathrm{E}}_n^a-\epsilon_b^*\hat{\mathrm{E}}_n^b\right)\left[\sin(k_n y_{j+1}^*)-\sin(k_n y_j^*)\right]+\left[\epsilon_b^*\hat{\mathrm{E}}_0^a-\epsilon_a^*\hat{\mathrm{E}}_0^b\right](y_{j+1}^*-y_j^*)$$

$$+\epsilon_{so}^*\left[\left(\frac{1}{y_{j+1}-y_j}+\frac{1}{y_j-y_{j-1}}\right)\hat{\mathrm{V}}_j-\frac{\hat{\mathrm{V}}_{j+1}}{y_{j+1}-y_j}-\frac{\hat{\mathrm{V}}_{j-1}}{y_j-y_{j-1}}\right]=0 \text{ for } j=1,2\ldots k \tag{2.19}$$

Again substituting for the electric fields, this time using (2.9) (remembering continuity of the potential across an interface) into (2.19) yields

$$\left\{\sum_{n=1}^{\infty}\frac{1}{k_n}\hat{\Phi}_n^a\left(\hat{\mathrm{C}}_n^a-\hat{\mathrm{C}}_n^b\right)\left[\sin(k_n y_{j+1}^*)-\sin(k_n y_j^*)\right]+\hat{\Phi}_0^a\left(\hat{\mathrm{C}}_0^a-\hat{\mathrm{C}}_0^b\right)(y_{j+1}^*-y_j^*)\right.$$

$$\left.+\epsilon_{so}^*\left[\left(\frac{1}{y_{j+1}-y_j}+\frac{1}{y_j-y_{j-1}}\right)\hat{\mathrm{V}}_j-\frac{\hat{\mathrm{V}}_{j+1}}{y_{j+1}-y_j}-\frac{\hat{\mathrm{V}}_{j-1}}{y_j-y_{j-1}}\right]\right\}=0 \text{ for } j=1,2\ldots k \tag{2.20}$$

The set of equations described by (2.20) can be expressed in matrix form, A•$\mathcal{V}$=X, with the elements of A representing the coefficients of the unknown normalized voltages $\hat{\mathcal{V}}_j \equiv (\hat{\mathrm{V}}_j/\hat{\mathrm{V}}_{\mathrm{D}})$, and the elements of X representing the coefficient of the known drive. In order to generate the matrix elements, (2.13) is substituted into (2.20) and terms are rearranged such that all terms proportional to the drive are moved to the right hand side. Any expression involving $\hat{\mathrm{C}}_0^a$ also goes on the right hand side as it can only arise from an externally applied field and thus, is also a driven signal. This yields

$$\mathrm{A}(i,j)=\mathrm{A}_1(i,j)+\mathrm{A}_2(i,j)+\mathrm{A}_3(i,j) \tag{2.21}$$

where

$$
\begin{aligned}
A_1(i,j) &= \sum_{n=1}^{\infty} \frac{\lambda}{k_n (n\pi)^2} \left(\hat{C}_n^a - \hat{C}_n^b \right) \left[\sin(k_n y_{i+1}^*) - \sin(k_n y_i^*) \right] \\
&\quad \bullet \left[\left(\frac{1}{y_{j+1} - y_j} + \frac{1}{y_j - y_{j-1}} \right) \cos(k_n y_j) - \frac{\cos(k_n y_{j+1})}{y_{j+1} - y_j} \right. \\
&\quad \left. - \frac{\cos(k_n y_{j-1})}{y_j - y_{j-1}} \right] \qquad \text{(a)} \\
A_2(i,j) &= -\hat{C}_0^b (y_{i+1}^* - y_i^*)(y_{j+1} - y_{j-1}) \qquad \text{(b)} \\
A_3(i,j) &= \epsilon_{so}^* \left[\left(\frac{1}{y_{i+1} - y_i} + \frac{1}{y_i - y_{i-1}} \right) \delta(i-j) - \frac{\delta(i-j+1)}{y_{i+1} - y_i} \right. \qquad \text{(c)} \\
&\quad \left. - \frac{\delta(i-j-1)}{y_i - y_{i-1}} \right]
\end{aligned}
\tag{2.22}
$$

and

$$
X(i) = X_1(i) + X_2(i) + X_3(i) \tag{2.23}
$$

where

$$
\begin{aligned}
X_1(i) &= -\sum_{n=1}^{\infty} \frac{\lambda}{k_n (n\pi)^2} \left(\hat{C}_n^a - \hat{C}_n^b \right) \left[\sin(k_n y_{i+1}^* - \sin(k_n y_i^*) \right] \\
&\quad \bullet \frac{1}{(y_1 - y_0)} \left[\cos(k_n y_0) - \cos(k_n y_1) \right] \qquad \text{(a)} \\
X_2(i) &= -\left[\hat{C}_0^a - \hat{C}_0^b \left(\frac{y_0 + y_1}{\lambda} \right) \right] (y_{i+1}^* - y_i^*) \qquad \text{(b)} \\
X_3(i) &= \epsilon_{so}^* \frac{\delta(i-1)}{y_1 - y_0} \qquad \text{(c)}
\end{aligned}
\tag{2.24}
$$

where y_j and y_j^* have been previously defined and $\delta(i)$ is defined as unity if the argument is zero and zero if it is nonzero. Using this representation, the set of k equations in k unknowns can be numerically inverted to determine the $\hat{\nu}_j$'s. A more rapidly convergent series for (2.22a) and (2.24a) is derived in Appendix A.

2.5 Equivalent Circuit Parameters

The pi network of Fig. 2.1 is helpful for formalizing the transition from the field analysis just undertaken to the general situation where the detection electrodes are not grounded but rather, are free to assume a potential consistent with the attached circuit. With the driven gate at potential $\hat{V}_D$ and the floating gate grounded, Y_{11} and Y_{12} are directly related to the electrode currents, and hence calculable from the electric fields just determined with the floating gate grounded.

$$\begin{aligned} \hat{i}_F &= -Y_{12}\hat{V}_D \\ \hat{i}_D &= (Y_{11} + Y_{12})\hat{V}_D \end{aligned} \tag{2.25}$$

The electrode currents are found by integrating (2.5), conservation of charge, throughout a volume enclosing each electrode. Assuming a total electrode meander length of M_{el}, these currents are

$$\begin{aligned} \hat{i}_D &= j\omega M_{el}\left\{2\epsilon_{so}^*\hat{E}_y^a(y_1^*) + 2\int_0^{y_0}\left[\epsilon_a^*\hat{E}_x^a(y) - \epsilon_b^*\hat{E}_x^b(y)\right]\,dy\right\} \\ \hat{i}_F &= j\omega M_{el}\left\{-2\epsilon_{so}^*\hat{E}_y^a(y_{k+1}^*) + 2\int_{y_{k+1}}^{\frac{\lambda}{2}}\left[\epsilon_a^*\hat{E}_x^a(y) - \epsilon_b^*\hat{E}_x^b(y)\right]\,dy\right\} \end{aligned} \tag{2.26}$$

Solving (2.25) for Y_{11} and Y_{12} in terms of $\hat{i}_D$ and $\hat{i}_F$ and incorporating (1.1), (2.2), (2.9) and (2.15) yields

$$
\begin{aligned}
Y_{11} &= 2j\omega M_{el}\left\{\epsilon^*_{so}\left(\frac{1-\hat{\mathcal{V}}_1}{y_1-y_0}-\frac{\hat{\mathcal{V}}_k}{y_{k+1}-y_k}\right)\right. \\
&\quad +\sum_n \frac{1}{k_n}\hat{\Phi}^a_n\left[\hat{C}^a_n-\hat{C}^b_n\right](\sin(k_n y_0)-\sin(k_n y_{k+1})) \\
&\quad \left.+\hat{\Phi}^a_0\left[\hat{C}^a_0-\hat{C}^b_0\right]\left(y_0+\frac{\lambda}{2}-y_{k+1}\right)\right\} \\
Y_{12} &= 2j\omega M_{el}\left\{\epsilon^*_{so}\left(\frac{\hat{\mathcal{V}}_k}{y_{k+1}-y_k}\right)\right. \\
&\quad +\sum_n \frac{1}{k_n}\hat{\Phi}^a_n\left[\hat{C}^a_n-\hat{C}^b_n\right]\sin(k_n y_{k+1}) \\
&\quad \left.-\hat{\Phi}^a_0\left[\hat{C}^a_0-\hat{C}^b_0\right]\left(\frac{\lambda}{2}-y_{k+1}\right)\right\}
\end{aligned}
\tag{2.27}
$$

where $\hat{\mathcal{V}}_j \equiv (\hat{V}_j/\hat{V}_D)$. Once (2.27) is programmed, it is only the surface capacitance densities, $\hat{C}^a_n$, that are altered when the medium above is changed.

2.6 Determination of Gain

At this point it is instructive to recap the major steps for calculating the Y_{ij}'s and hence, the gain. Solution of a system of equations of the form A•$\mathcal{V}$=X with the matrix elements defined by (2.21)–(2.24) determines the $\hat{\mathcal{V}}_j$'s and the electric field distribution. With these $\hat{\mathcal{V}}_j$'s, the fourier coefficients for the potential at the interface, $\hat{\Phi}^a_n$ and $\hat{\Phi}^a_0$, using (2.13), and hence, the circuit parameters Y_{11} and Y_{12}, using (2.27), are evaluated. Finally, given a load admittance Y_l, the gain is then calculated according to (2.1). The software routine for computing the gain–phase response is called **mz1.for** and may be found in Appendix F.

2.7 Collocation Points, Fourier Modes and Convergence

Several tradeoffs must be considered for deciding the optimum number of collocation points and fourier modes. Increasing the number of collocation points improves the representation of the actual potential distribution – at the cost of greatly increasing the computing time required to calculate each element of the matrix A and to compute the matrix inverse. Increasing the number of fourier modes used to compute the voltage distribution increases the accuracy – of representing this distribution with a piecewise linear function. Beyond a certain number of modes the piecewise representation may differ significantly from the actual one. However, as previously mentioned and elaborated in Appendix A, a more rapidly convergent series is used to sum the fourier modes for the elements of A matrix. This series effectively sums up roughly 1,000 modes. In this case, it is quicker to compute either a small or a large number of modes than to compute an intermediate number. Using the large number of modes, the number of fourier modes summed in (2.27) when computing Y_{11} and Y_{12} can be altered. Increasing the number of terms used here has a much smaller, though still significant, effect on the computing time.

Determination of convergence requires a specification against which various results can be compared. For the case of the microdielectrometer, where comparisons are made with an actual output, the accuracy of the gain is given as 0.1 dB [67]. This accuracy will be used as a tolerance for measuring convergence. Modeling an uncoated microchip in air, the high frequency coupling capacitances per unit length (normalized to the oxide layer permittivity) and resulting gains are tabulated for a matrix of values for the number of collocation points and fourier modes (Table 2.1). The normalized load capacitance used was 1.46. It was observed that using 25 collocation points and 100 summation terms provides sufficient convergence when compared to the maximum values of 35 and 1000, respectively (as well as taking a reasonable amount of computing time). For the rest of this thesis, unless otherwise noted, 25 collocation points and 100 summation terms will be the standard numbers used.

Table 2.1: Predicted Gain Output vs. k Collocation Points and N Summation Terms.

Model Parameters		**Coupling Capacitances**		**Gain (dB)**
k	N	$C_{11}/\epsilon_{ox}M_{el}$	$C_{12}/\epsilon_{ox}M_{el}$	G_p
10	50	8.966	.2055	-34.275
	100	9.054	.2090	-34.203
	150	9.027	.2107	-34.114
	1000	9.023	.2096	-34.153
25	50	8.822	.2042	-34.210
	100	8.939	.2095	-34.089
	150	9.007	.2112	-34.074
	1000	9.110	.2150	-34.006
35	50	8.809	.2044	-34.194
	100	8.914	.2090	-34.089
	150	8.970	.2107	-34.064
	1000	9.132	.2159	-33.989

An additional note on the implementation of the model - there is a symmetry in the matrix $A_1(i,j)$ of (2.22) that is exploited to reduce the number of computations. It turns out that for k collocation points, producing a kxk matrix, $A_1(i,j) = A_1(k - i + 1, k - j + 1)$, thus halving the number of summations that need to be computed.

Chapter 3

Typical Responses Using Continuum Model

As a first step toward developing a scheme for identifying continuum parameters, Section 3.1 is devoted to examining predicted frequency responses for a variety of physical situations that would be commonly encountered. Insights formed from such an examination, combined with confidence gained from experimental evidence, result in the observations of Section 3.1.5. In addition, some working theorems are developed in Section 3.2 that may be useful in interpreting experimental gain–phase data of heterogeneous media. These theorems are exploited in Chapter 7 to interpret responses obtained using the moisture sensors. Comparisons of results using the continuum model with several limiting cases having analytical solutions are made in Section 3.3. The good agreement between models enhances the validity of the continuum model developed here.

3.1 Predicted Responses

For all of the following examples $\hat{E}_0^a = 0$ ($n = 0$ mode) thus $\hat{C}_0^a = 0$. Using the form expressed in (1.1), solution of Laplace's equation for the potential in the insulating oxide layer below the electrodes gives

$$\hat{C}_n^b = -\epsilon_{cx} k_n \coth(k_n h) \quad ; \quad \hat{C}_0^b = -\frac{\epsilon_{ox}}{h} \tag{3.1}$$

where the layer has a thickness h and is bounded from below by a "perfectly conducting" substrate.

3.1.1 Uniform Media

This situation, a semi-infinite half space of uniform complex permittivity, is pictured in Fig. 3.1a. The potential satisfies Laplace's equation and decays to zero as x goes to infinity. Thus,

$$\hat{C}_n^a = \frac{\epsilon_a^* \hat{E}_n^a}{\hat{\Phi}_n^a} = \epsilon_1^* k_n \tag{3.2}$$

(These expressions illustrate how transfer relations naturally fit into the spectral field description [75, p. 2.33]).

Typical responses, shown in Fig. 3.2, have the features of a linear system with one time constant – the charge relaxation time, $\tau_e = (\epsilon_1/\sigma_1)$. At high frequencies, $\omega\tau_e \gg 1$, the effect of dissipation currents is small. The coupling between the two electrodes is purely capacitive, explaining the low gain at high frequencies. As the frequency is lowered, $\omega\tau_e$ approaches unity and conduction currents become appreciable. Thus the gain increases and a nonzero phase occurs. Specifically, there is a peak in phase and the slope of the gain is -20 dB/decade, both indicative of a uniform medium (which, for constant permittivity and conductivity, is characterized by a single time). At very low frequencies, $\tau \gg \tau_e$, the medium becomes an extremely good conductor and consequently the electrodes are shorted together (0 dB gain and zero phase). It is evident from the equations that the frequency enters only as a ratio, $(\omega\epsilon_1/\sigma_1) \equiv \omega\tau_e$. Thus, the frequency scales with the bulk conductivity – altering the conductivity merely shifts the frequency response without changing the curve shapes (Fig. 3.2). Changing the bulk permittivity alters both the breakpoint (due to a change in τ_e and the high frequency gain (due to a change in the capacitive coupling), as shown in Fig. 3.3.

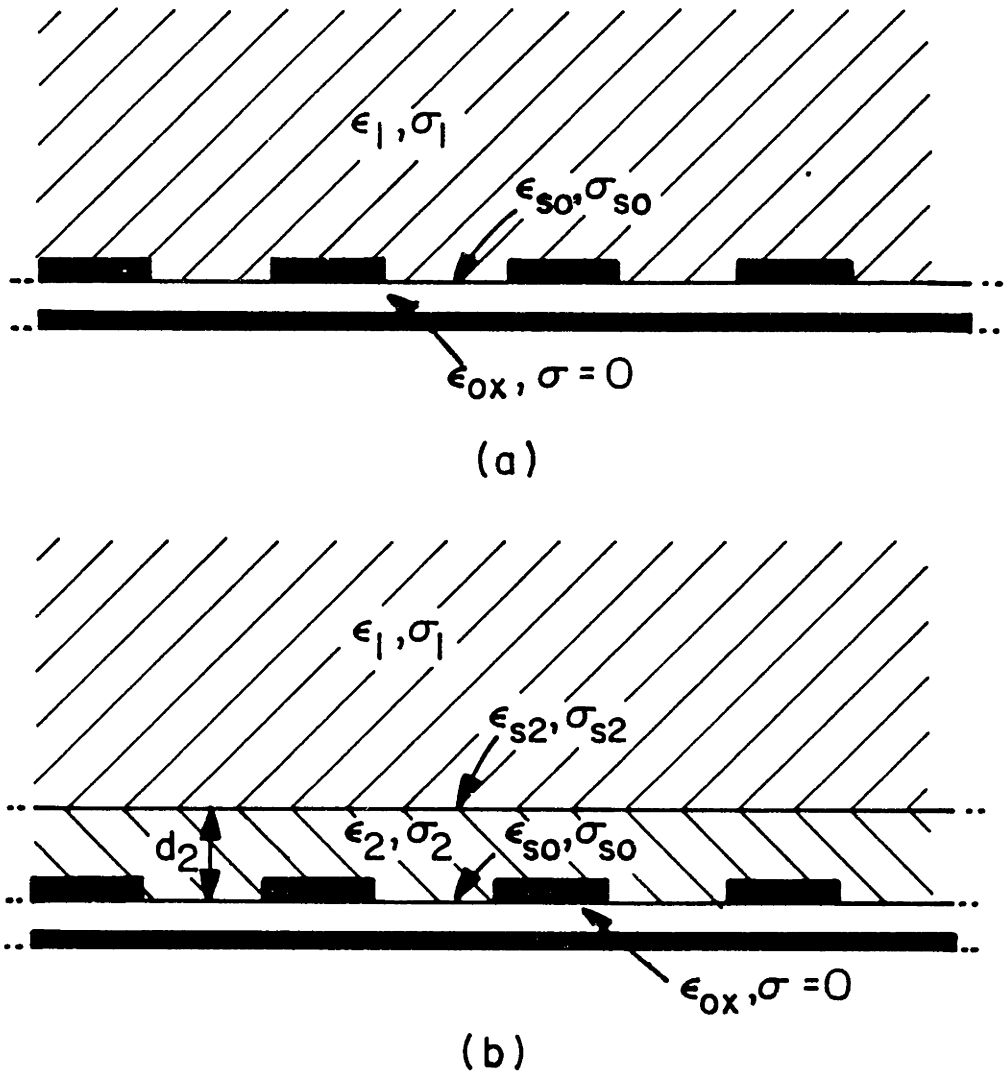


Figure 3.1: Electrode Structure Below a) Uniform Medium or b) Variable Thickness Homogeneous Layer and Uniform Medium.

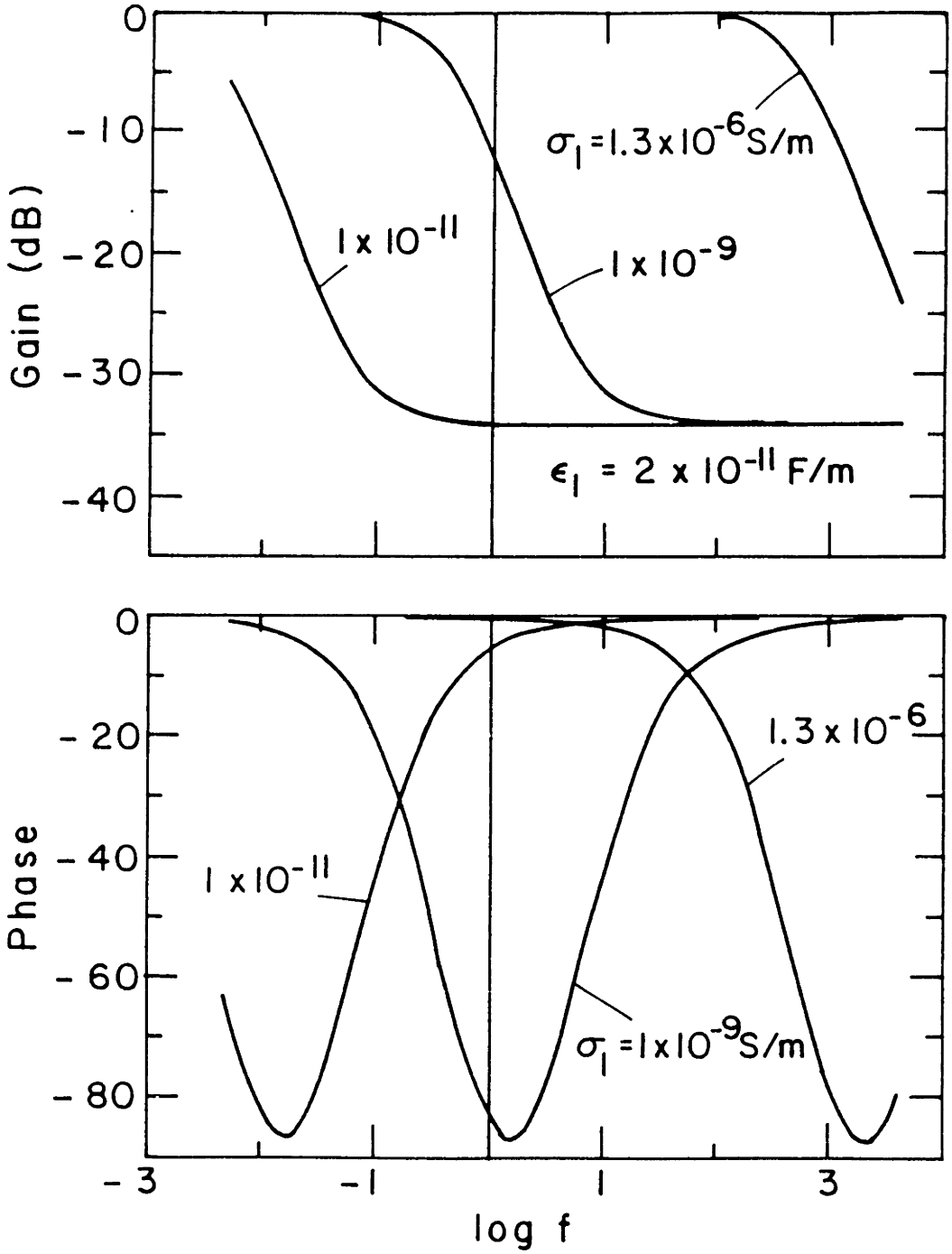


Figure 3.2: Predicted Responses for Microchip in Uniform Medium, Vary Conductivity σ_1 ($\epsilon_1 = 2x10^{-11}$ F/m).

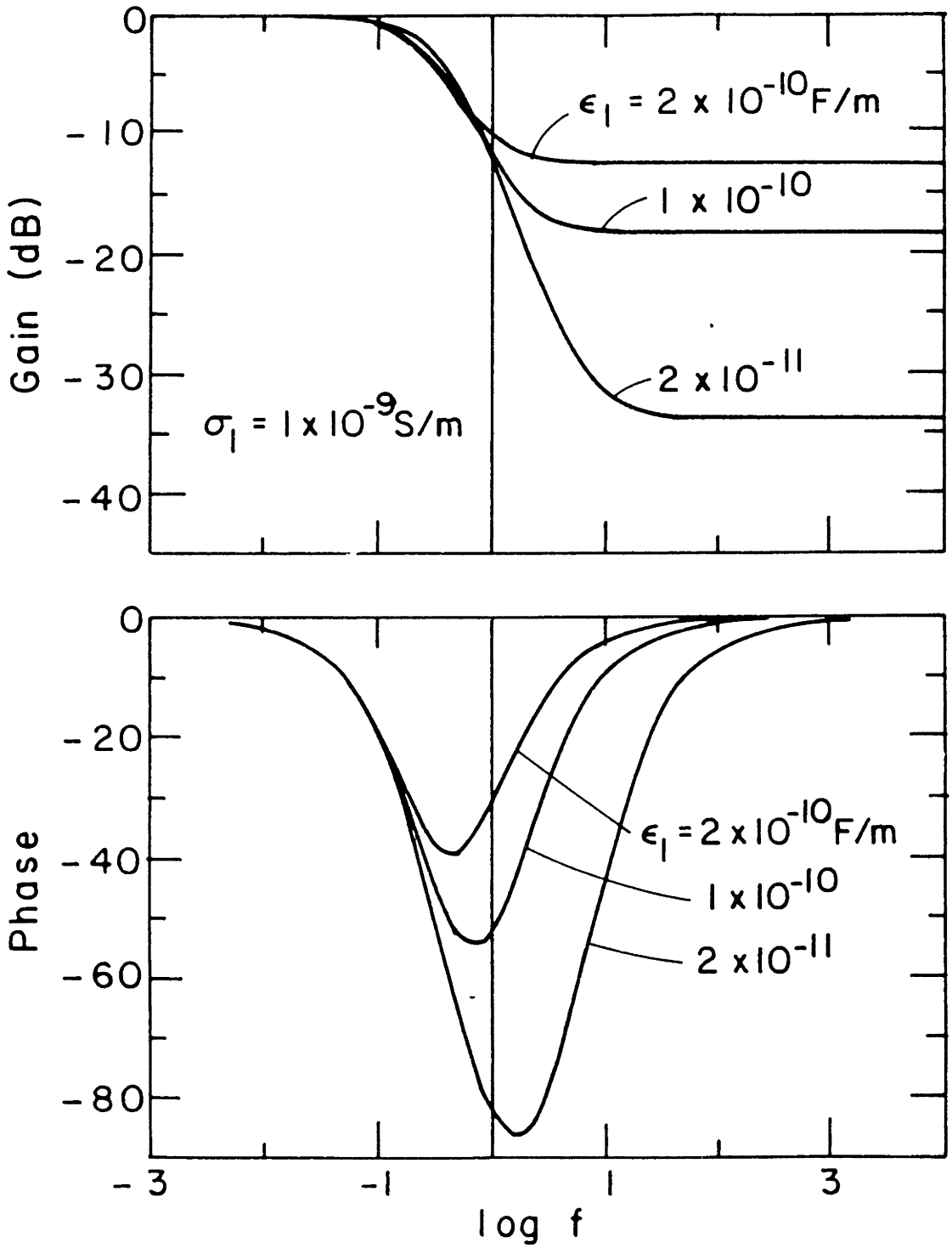


Figure 3.3: Predicted Responses for Microchip in Uniform Medium ($\sigma_1 = 10^{-9}$ S/m), Vary Permittivity ϵ_1.

One of the first applications of the microdielectrometer was to monitor the curing of epoxy resins. A finite difference simulation, employing grid elements and relaxation methods, to determine the electric field distribution was developed by Lee [70, Chapter 3]. Under the assumptions of dealing only with linear, homogeneous systems (no spatial variation in ϵ_1 or σ_1) and setting aside any surface conduction phenomena, the simulation calculates the gain and phase at a specified frequency for a material with a given ϵ' and ϵ''. A look-up table, compiled from many simulations, provided a unique conversion from gain, phase and frequency information to ϵ' and ϵ''. Dielectric relaxation information can then be extracted from plots of ϵ'' vs. ϵ' (Cole-Cole plots). When the phase is plotted versus gain for the curves shown in Figs. 3.2 and 3.3 (eliminating frequency as the independent parameter), the graph reproduces the lookup table generated by this finite difference simulation.

3.1.2 Variable Thickness Layer

A layer of thickness d_2 and characterized by a uniform complex bulk permittivity is placed immediately above the electrodes (Fig. 3.1b). Solution of Laplace's equation in a piece–wise fashion, perhaps using transfer relations (see Appendix B or [75, p. 2.33]), gives

$$\hat{C}_n^a = \frac{\epsilon_a^* \hat{E}_n^a}{\hat{\Phi}_n^a} = \epsilon_a^* k_n \left\{ \coth(k_n d_2) - \frac{1}{\sinh^2(k_n d_2)} \left[\left(\frac{\epsilon_1^* + k_n \epsilon_{s2}^*}{\epsilon_2^*} \right) + \coth(k_n d_2) \right]^{-1} \right\} \quad (3.3)$$

As background for inferring the layer thickness from a measurement of the gain, the effect on the frequency response of varying d_2 while holding the complex bulk permittivities of each region constant is illustrated by Fig. 3.4. In this case, the half space is taken to be more insulating than the layer and the surface properties of the layer–half space interface are taken to be zero.

For a thick layer, $d_2 > (\lambda/2\pi)$, the response is that of a uniform half–space, determined solely by the properties of the layer. The electrodes do not "see" beyond the thick layer for purely geometrical reasons. As the layer thickness is decreased the response is still a bulk one, but now it also reflects the fact that the fields have begun

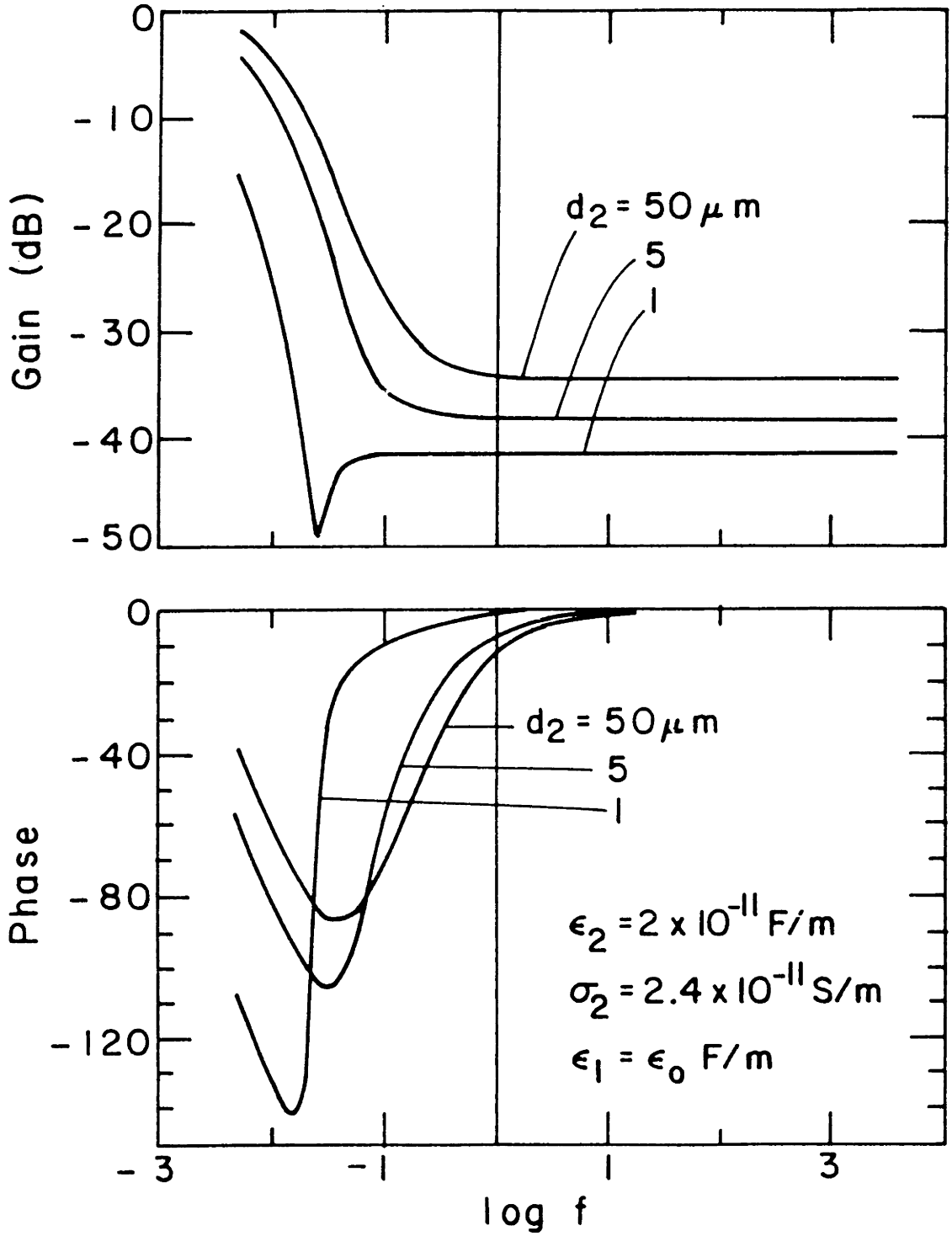


Figure 3.4: Predicted Responses for Microchip with Variable Thickness Layer in Air ($\epsilon_2 = 2x10^{-11}$ F/m, $\sigma_2 = 2.4x10^{-11}$ S/m).

to penetrate into the semi-infinite region, where the permittivity is less (accounting for the decrease in high frequency gain) and the conductivity is zero. At $d_2 \sim .1(\lambda/2\pi)$, the layer is so thin that it might well be described by a surface conductivity on the interelectrode interface. This case is discussed in the next section. Further decreases in the layer thickness reduces the effective interelectrode surface conductivity. This results in a shift of the interelectrode surface type response to lower frequencies. Generalization of the $\hat{C}_n$ to include multiple layers is presented in Appendix B.

3.1.3 Singular Properties at Substrate–Medium Interface

This is again the case of a uniform half-space (Fig. 3.1a) with $\hat{C}_n^a$ given by (3.2), but with a surface conductivity at the SiO_2–medium interface between the electrodes $\epsilon_{so}^* = -j(\sigma_{so}/\omega)$. A typical response is shown in Fig. 3.5 – clearly distinguishable from a bulk response. There is an additional characteristic time, $\tau_{se} = (\epsilon_1/k_n\sigma_{so})$, representing surface charge diffusion along the interface. Again the interplay between surface and bulk conduction and displacement currents accounts for the transition from low to high gain and a peak in phase. However, the slope of the gain is greater than –20 dB/decade and the gain has an overshoot. The phase curve is asymmetric and the phase peaks at a larger angle than previously seen. All these are characteristic of the surface charge diffusion from driven to floating electrodes made possible by the combination of the distributed surface conduction and the shunt capacitance between the interface and the highly conducting substrate.

As the surface conductivity decreases (τ_{se} increases), the curves pass through a regime where surface and bulk conduction are equally important to a point beyond which a purely bulk response is observed (Fig. 3.6).

The overshoot phenomena, a prediction of gains below the purely capacitive high frequency gain, is further evidence of the surface charge diffusion process. This can be explained by separating the coupling between the electrodes into two components, one through the bulk medium and one through a distributed transmission line composed of the interelectrode surface and the insulating oxide layer. The coupling through the

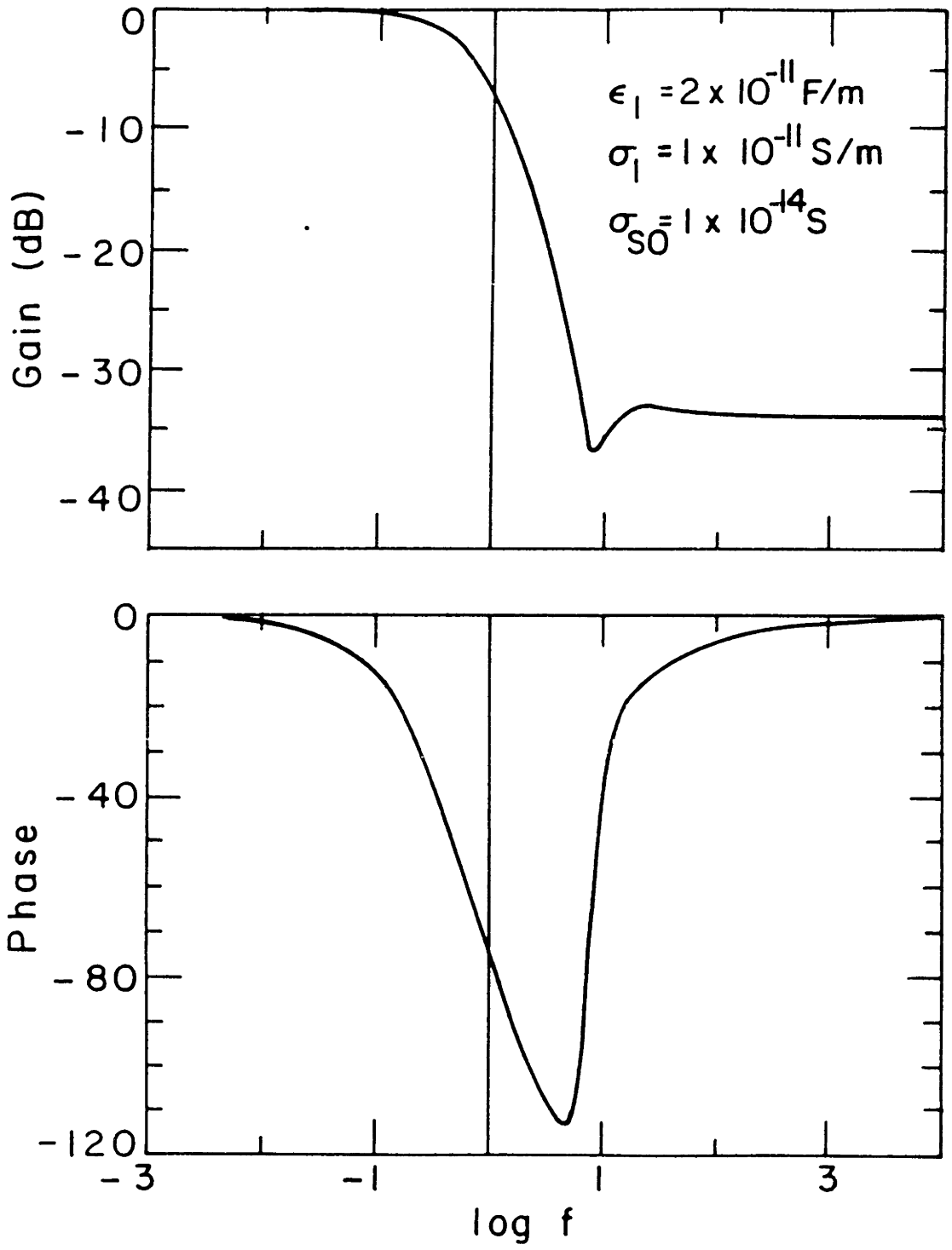


Figure 3.5: Predicted Response for Microchip in Uniform Medium with Surface Conductivity on Interelectrode Surface ($\epsilon_1 = 2x10^{-11}$ F/m, $\sigma_1 = 10^{-11}$ S/m, $\sigma_{so} = 10^{-14}$ S).

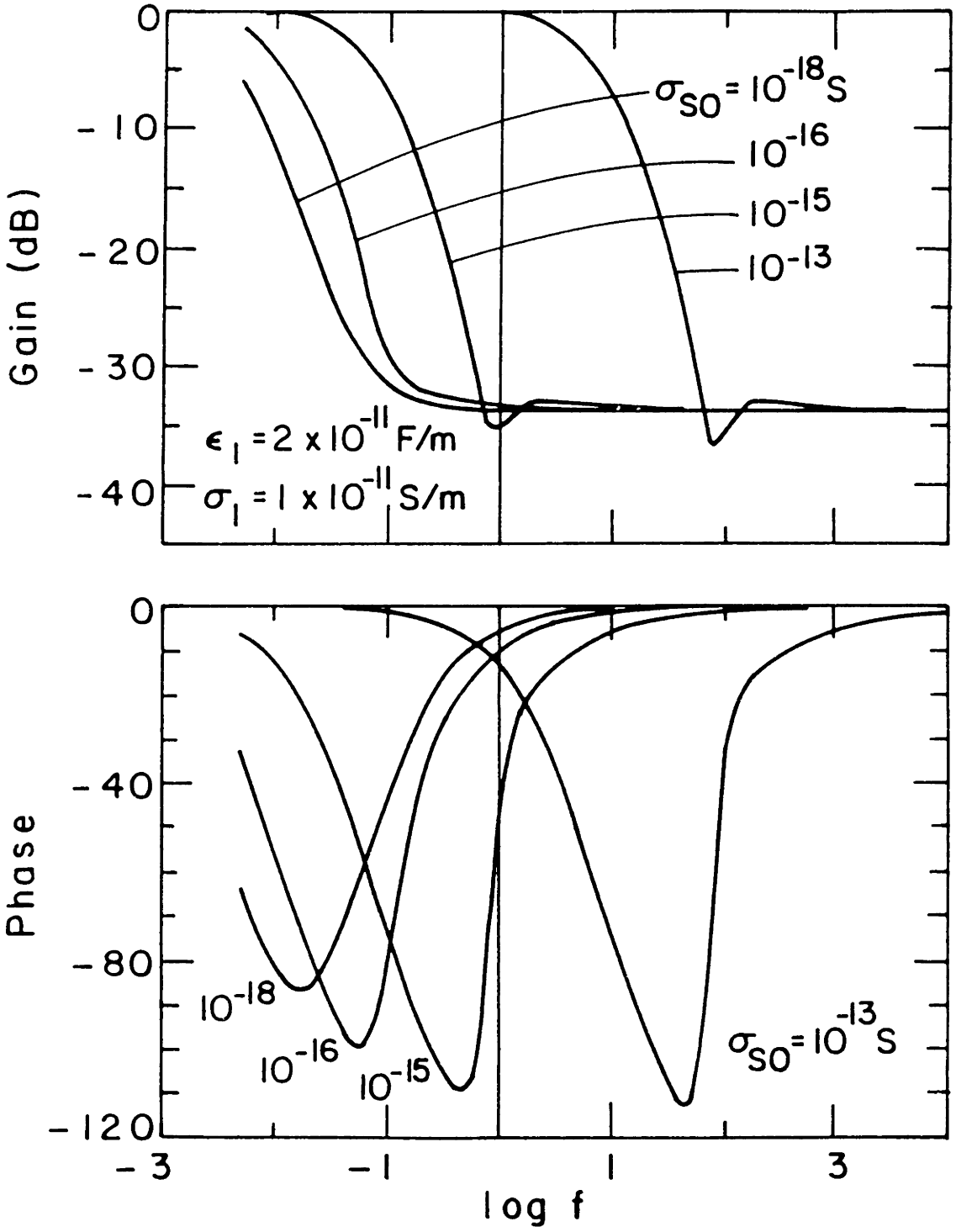


Figure 3.6: Predicted Responses for Microchip in Uniform Medium, Vary Surface Conductivity on Interelectrode Surface ($\epsilon_1 = 2x10^{-11}$ F/m, $\sigma_1 = 10^{-11}$ S/m, σ_{so}).

bulk makes a smooth transition from purely conductive to purely capacitive just as in the bulk response and therefore involves at most a 90° phase shift. However, the part of the signal resulting from transmission along the distributed RC transmission line comprised of the interface and the insulating oxide layer suffers larger phase shifts and hence a contribution that tends to cancel that due to the "direct" coupling. The result is a frequency range over which the gain is smaller than that at high frequency and the phase shift larger than 90°. The diminution of surface conductance coupled with an existing large bulk capacitance "snaps" the response from a surface conduction coupling to a purely capacitive coupling and accounts for the rapid changes exhibited in the frequency response.

An important parameter here is the silicon dioxide layer thickness, h. The thinner the oxide layer the greater the coupling between the electrodes and the substrate. This results in a reduced capacitive coupling between the electrodes through the oxide layer and necessitates the incorporation of a surface permittivity for the model to predict phases within the limits of 0° and −180° (see Fig. 3.15). Details on the effect of altering h are in Section 3.1.5 and on the requirement of a surface permittivity are in Section 3.3, case 4.

3.1.4 Layer With Surface Conductivity

Polymer films are often coated with conducting surface agents. With the objective of measuring the surface conductivity of these films without electrical contact, the frequency response of a layer (Fig. 3.1b) with surface conductivity attributed to the layer–medium interface is of interest. The surface capacitance density is given by (3.3) with ϵ_{s2}^* having only an imaginary component.

In Fig. 3.7 is shown the frequency responses for an electrode structure with a layer thickness $d_2 = .1\lambda$ in air. By increasing the layer's surface conductivity, as shown, two relaxation phenomena are observed. The higher frequency one results from the spectrum of times associated with diffusion of surface charge along the upper surface of the layer. The lower frequency one is from relaxation determined by the bulk properties of the layer. If $\sigma_{s2} \ll \sigma_2\lambda$, so that bulk conduction in the layer dominates

that of the surface, then relaxation through the bulk of the layer will come into play at a higher frequency and the relaxation due to the surface conductivity will not be so easily observed. Altering the layer thickness, with an observable surface conductivity, will affect the capacitance of the layer and thus, the mid–frequency gain plateau and the higher frequency phase peak.

3.1.5 Sensitivity to Electrode Structure Parameters

There are four parameters describing the electrode structure, each with its individual effect on the gain–phase response. The spatial wavelength, λ has been discussed previously with the application of intimate sensing. Thus, this value should be consistent with the characteristic length of the medium under measurement.

Two other parameters are the insulating layers' thickness, h, and permittivity, ϵ_{ox}. They are dependent on the material used (for ϵ_{ox}) and the fabrication method (for h). As mentioned earlier, varying h can significantly affect the response to a surface conductivity at the electrode boundary. This effect can be extended further to include the response to thin ($d_2 \sim 0.1(\lambda/2\pi)$) layers having either bulk or surface conductivities. An example is the response of the macrochip to surface conductivities. With its current, relatively thick oxide layer with high permittivity, there is no surface–like response with overshoot gains and asymmetric, large phase peaks when examining such thin layers [23, Section 2.3]. As shown by Li, this can be altered so as to optimize sensitivity to these type of phenomena by reducing h and ϵ_{ox}.

This lack of sensitivity with thick, high permittivity insulating layers actually extends to measurements of not only surface but bulk properties of thick and thin layers. Referring to (2.1), the gain changes with the material above due mainly to changes in Y_{12}. The admittance Y_{11} is basically capacitive and much larger than the capacitive part of Y_{12}. As shown by Li [23, Section 2.2.2], Y_{12} can be thought of as representing two parallel couplings between the floating and driven gates, one above the interface and through the medium, and the other below the interface and through the insulating layer. A contrast in this coupling above and below is shown in Fig. 3.8 for large and small values of h with the permittivity the same above and below the electrodes. With

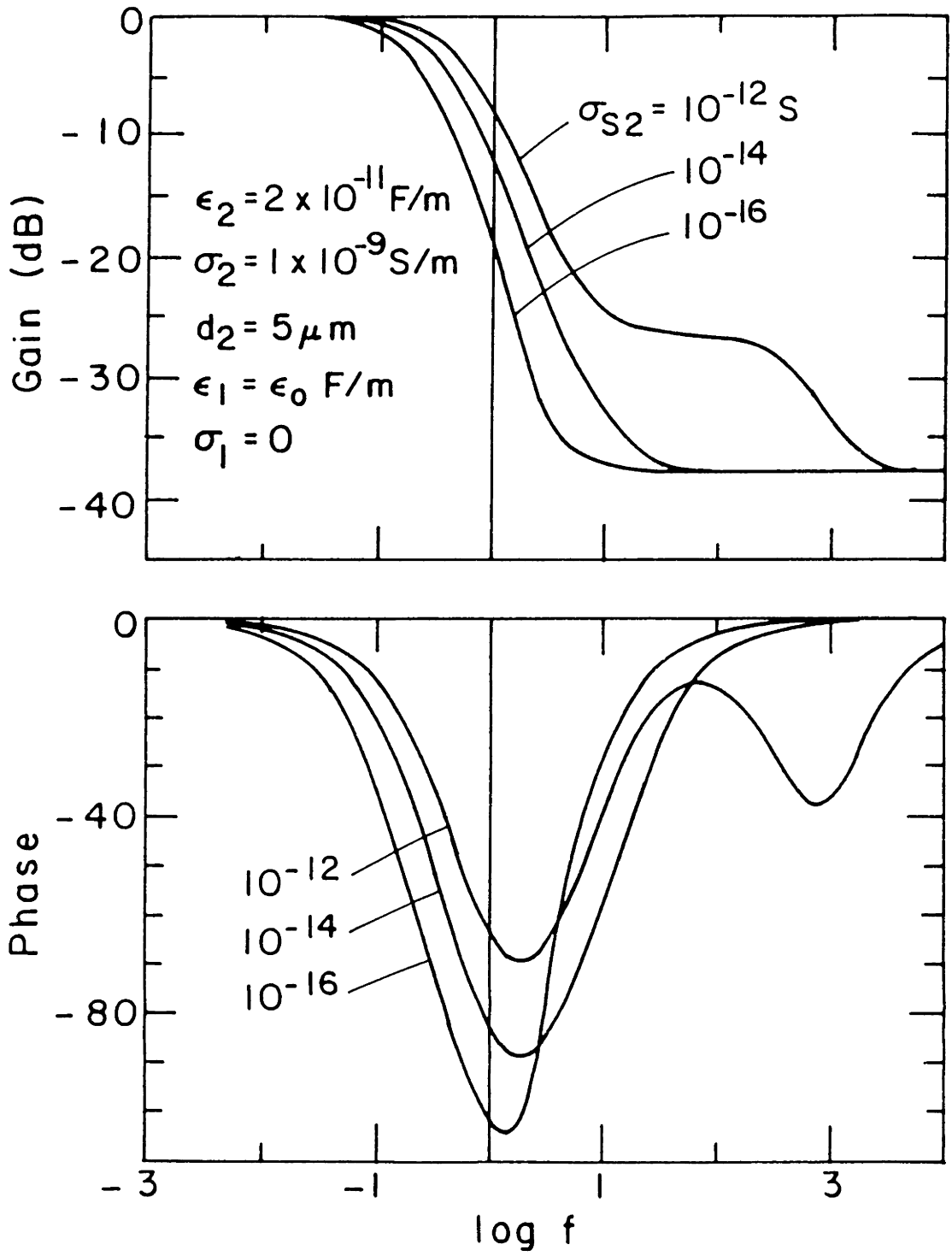


Figure 3.7: Predicted Responses for Microchip with a Layer Having a Variable Surface Conductivity in Air ($d_2 = 5\ \mu$m, $\epsilon_2 = 2x10^{-11}$ F/m, $\sigma_2 = 10^{-9}$ S/m, σ_{s2}) .

h large, the couplings above and below are equal. Changing the permittivity of the medium above will change the net coupling (the sum of the two couplings) by a much smaller percentage. If the permittivity below is much larger than above, this percentage change shrinks even further. With h small, the coupling below is much smaller than above. Most of the field below couples to the ground plane, thus the contribution to Y_{12} from below is greatly reduced. Now changes in the medium above are more fully reflected in changes in Y_{12} and the sensitivity is greatly enhanced. Thus, for maximum sensitivity, the oxide layer thickness and permittivity should be minimized (subject to fabrication constraints).

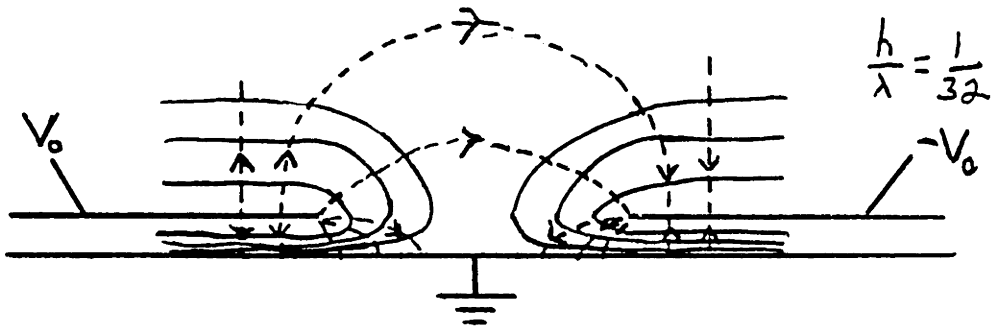
The fourth parameter is the interelectrode spacing, a. Using physical intuition, if a is decreased then the electrode–ground plane structure approaches that of a parallel plate capacitor. It is as if a is an "electric" shutter controlling the amount of electric field penetrating into the medium. Sensitivity to changes above will be reduced with small values of a. However, increasing a, while improving sensitivity, decreases the net coupling between the electrodes. This reduces the floating gate voltage and requires the electronics to accurately measure very small voltages with correspondingly small signal–to–noise ratios. Thus, it appears that the $(\lambda/4)$ width for a is a valid middle ground.

3.2 Working Theorems

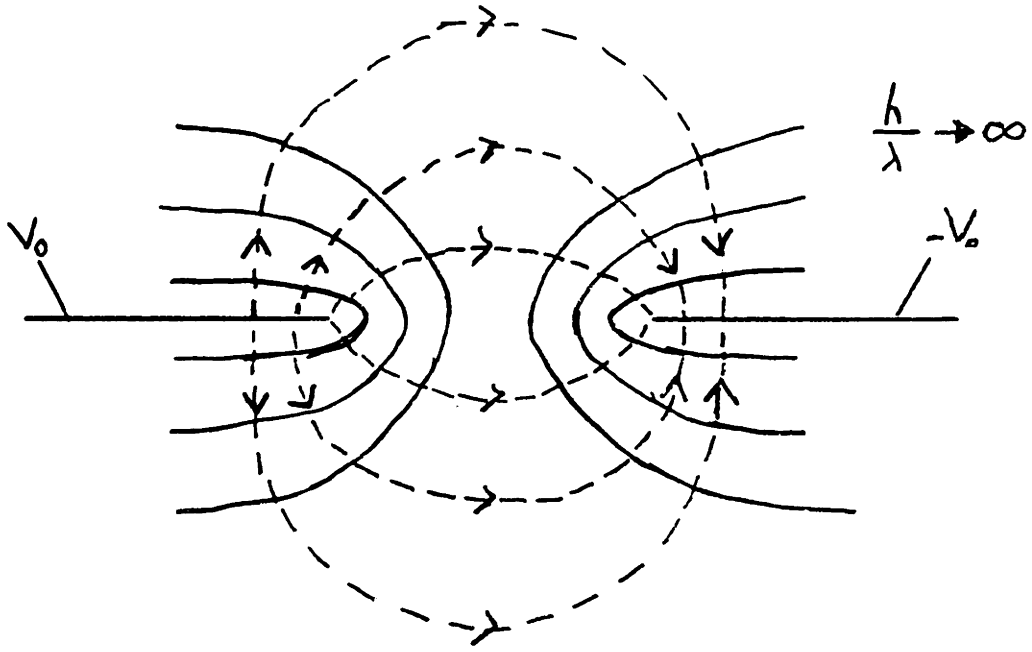
The following paragraphs represent a set of guidelines established while interpreting the gain–phase responses of a variety of media examined with the micro and macrochips. It is not meant to be a complete list, rather, it is just a set commonly encountered in the work performed here and in Li's [23] thesis.

3.2.1 Master Curves

Definition 3.1 *A master curve is the gain–phase response of a medium that displays shape invariance with respect to variations in an external, physical parameter.*



(a)



(b)

Figure 3.8: Electric Field Distribution as a Function of Oxide Layer Thickness.

Master curves are obtained by normalizing various frequency responses to the appropriate parameter. For example, a master curve can be drawn for the set of curves shown in Fig. 3.2 by normalizing the frequency to the phase peak frequency, f_p (Fig. 3.9). Once a master curve has been produced and the sensitivity of f_p to changes in the physical parameter of interest established, interpretation of experimental data is straightforward.

There are two types of media frequently encountered in this work that produce master curves – ohmic conductors and dispersive media. The special case of a semi-infinite ohmic conductor was just given as an example. This case will be shown to be a subset of dispersive media having a spatial distribution of complex permittivity.

Theorem 3.1 *If the complex permittivity of a medium having a spatial distribution of dispersive dielectric properties can be expressed as*

$$\epsilon^* = \epsilon'(\frac{\omega}{\omega_p}, x) - j\epsilon''(\frac{\omega}{\omega_p}, x) \tag{3.4}$$

where the dependence on external parameters can be lumped into

$$\omega_p = f(T, RH, c \ldots) \tag{3.5}$$

then a master curve can be produced.

Note that the real and imaginary parts can have different spatial dependencies. The frequency dependence is scaled by ω_p, a parameter containing the dependence on external variables such as temperature, humidity, or concentration. If it is possible to lump such dependencies into ω_p then the relevant parameter is the normalized frequency (ω/ω_p). Thus, the responses will be shape invariant and just shift in frequency, producing master curves. Examples of such dispersive phenomena are encountered in the moisture sensor experiments of Chapter 7.

For the special case of a nondispersive medium, ϵ' is independent of ω and ϵ'' varies inversely with (ω/ω_p). This yields an ohmic conductor with a spatial distribution of properties that remains constant with variations in external parameters. Again, the

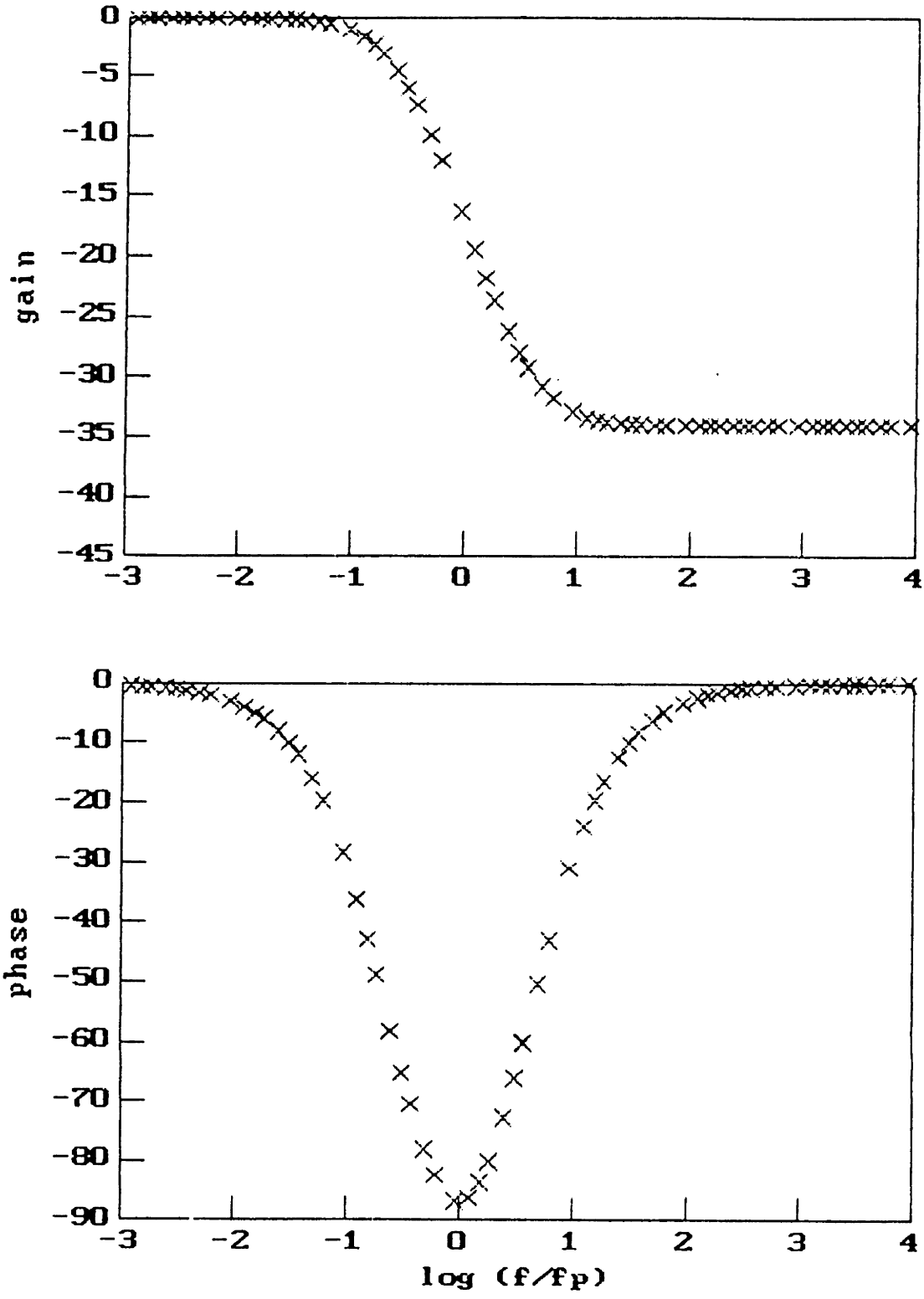


Figure 3.9: Predicted Responses of Fig. 3.2 Normalized to Frequency of Phase Peak, f_p.

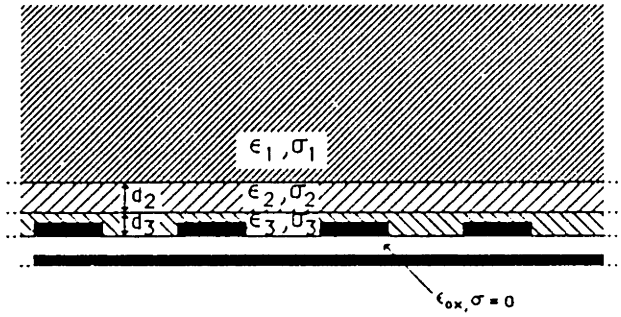
responses can be normalized into one master curve. Examples of responses produced by such spatial distributions may be found in Li's work [23, Section 5.3 and Appendix A].

3.2.2 Spatial Discontinuities

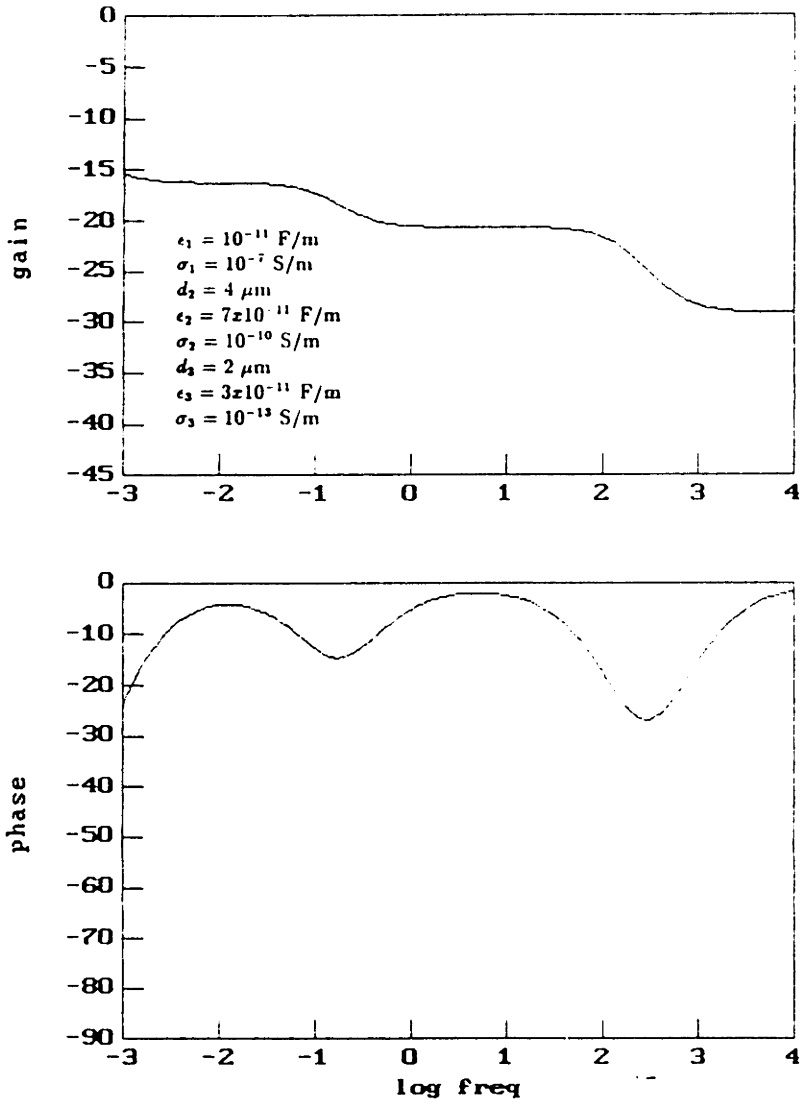
Very often, the medium under examination is a multilayered structure that can be modeled as a piecewise homogeneous structure (Fig. 3.10a). Further, the relaxation time constants associated with each layer may be quite different from each other as the layers vary from insulating to semi–insulating to semi–conducting as one moves away from the electrodes. Assuming the layers are within the sensing region of the particular spatial wavelength sensor used, the gain–phase response associated with such a structure will then exhibit the relaxation phenomena of each layer separately as shown in Fig. 3.10b. Each plateau in gain, accompanied by roughly zero phase, corresponds to the purely capacitive coupling between the electrodes through all the layers of the medium that remain insulating and, this is important, are below the first layer from the electrode plane that has become essentially perfectly conducting. This perfectly conducting layer forms an equipotential above which the fields do not penetrate. Note that if this perfectly conducting layer is the lowest layer, adjacent to the electrodes, it becomes impossible to see anything beyond this layer using one spatial wavelength and temporal information. It will be shown in Section 4.3 however, that distributions of conductivity that decrease away from the electrodes can be discerned using variable spatial wavelength information.

Thus, the plateaus in gain are determined by the permittivity and thickness of the layers below the first perfectly conducting layer. If all the permittivities are known and all but one thickness is known, then that thickness can be estimated from the experimental gain (see Section 4.2.1). For example, using this technique one can estimate the air or oil gap thickness for a material not in intimate contact with the macrochip [23, [p. 56 and p. 75].

Not only can layer thicknesses be estimated, but dielectric properties can be estimated and associated with the appropriate layer. The gain transitions between plateaus, and accompanying phase peaks, are the result of a layer going from capac-



(a)



(b)

Figure 3.10: a) Multilayered Structure with Spatial Distribution in Complex Bulk Permittivity and b) Predicted Response using Microchip.

itive to conductive as the frequency is decreased. Again the requirement is that the relaxation times are distinct enough so that these transitions and peaks do not significantly overlap one another. A good example of this is the estimation of the dispersive property of paper insulation within a multilayered structure [23, Section 5.3].

Theorem 3.2 *Given a multilayered structure of piecewise homogeneous dielectric properties, each layer, of known thickness, having an associated relaxation time. distinct enough from the others such that the gain has plateaus and the phase goes to zero at some time during those plateaus, and further, the relaxation times decrease with distance from the electrodes, then the dielectric properties and thickness of each layer can be uniquely determined from the frequency response.*

In addition to bulk dielectric properties, this approach can be extended to include surface properties. The proviso of distinct relaxation times must be satisfied by the surface properties as well, as demonstrated in Fig. 3.7. For $\sigma_{s2} < 10^{-16}$ S the response is dominated by the bulk relaxation within the layer. Only at $\sigma_{s2} \geq 10^{-12}$S does the surface coupled relaxation significantly stand out in the response. There is a question of uniqueness in ascertaining whether the response is truly due to a surface or bulk conductivity. When the surface conductivity is upon either a relatively thick layer, at some distance from the electrodes, or a relatively high permittivity layer, the response will lose the overshoot in gain and large, asymmetric phase, characteristic of the surface charge diffusion problem, and begin to resemble the response due to a bulk conductor (Figs. 3.11 and 3.12, respectively). It must be known *a priori* whether the conduction is bulk or surface in origin to perform the correct estimation using the temporal frequency response (see Li for estimation of surface conductivity of pink polyethylene [23, p. 56]).

It has been demonstrated that under certain conditions information on the spatial dependence of dielectric properties can be estimated from the frequency response. A more natural approach, as alluded to in the section on intimate sensing (Section 1.2.2) is to estimate spatial distributions using variable spatial wavelength data. This approach will be examined in Section 4.3.

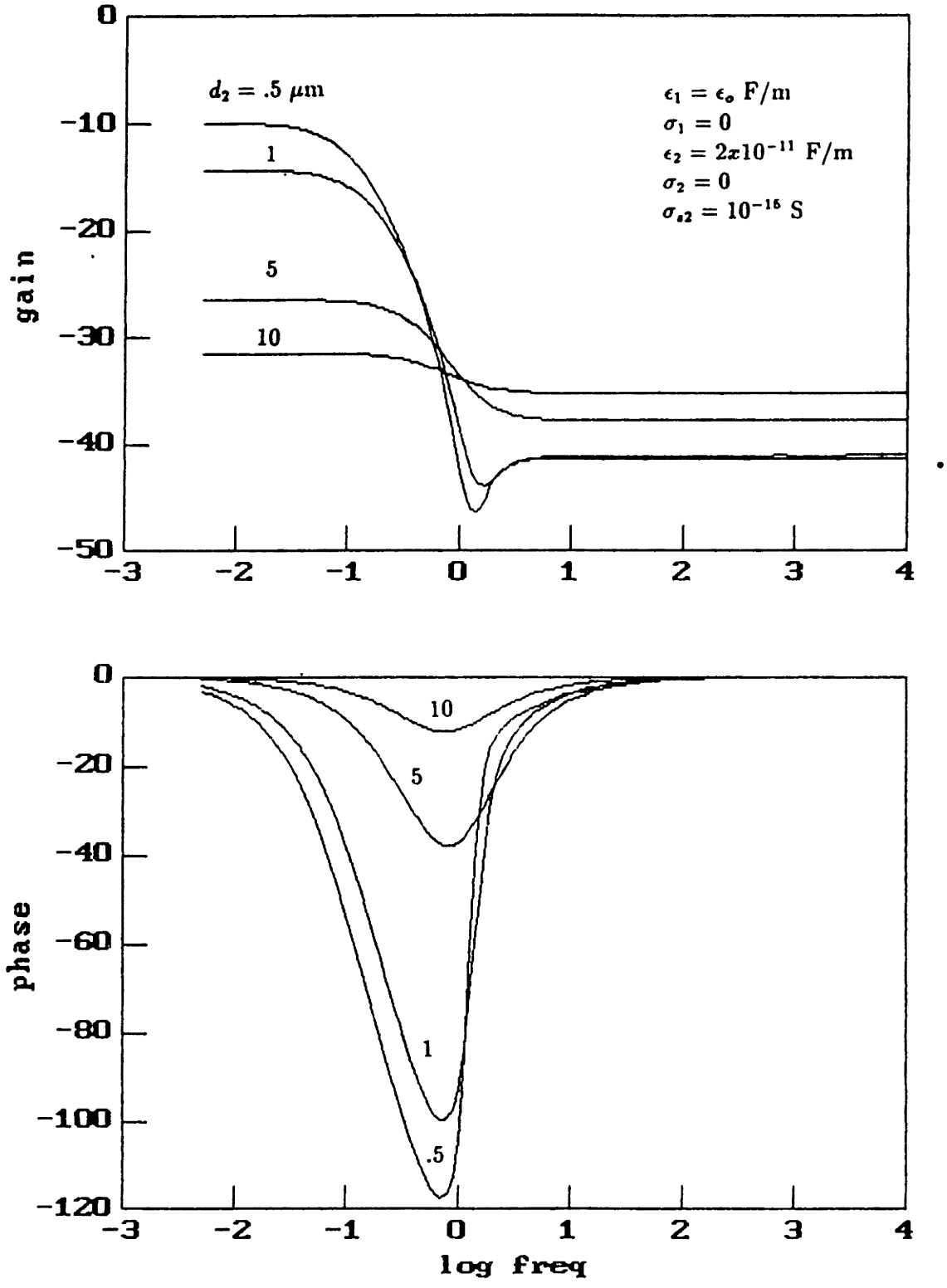


Figure 3.11: Predicted Responses for Microchip with an Insulating Layer of Variable Thickness (d_2, $\epsilon_2 = 2x10^{-11}$ F/m, $\sigma_2 = 0$ S/m) having a Surface Conductivity ($\sigma_{s2} = 10^{-15}$ S) in Air.

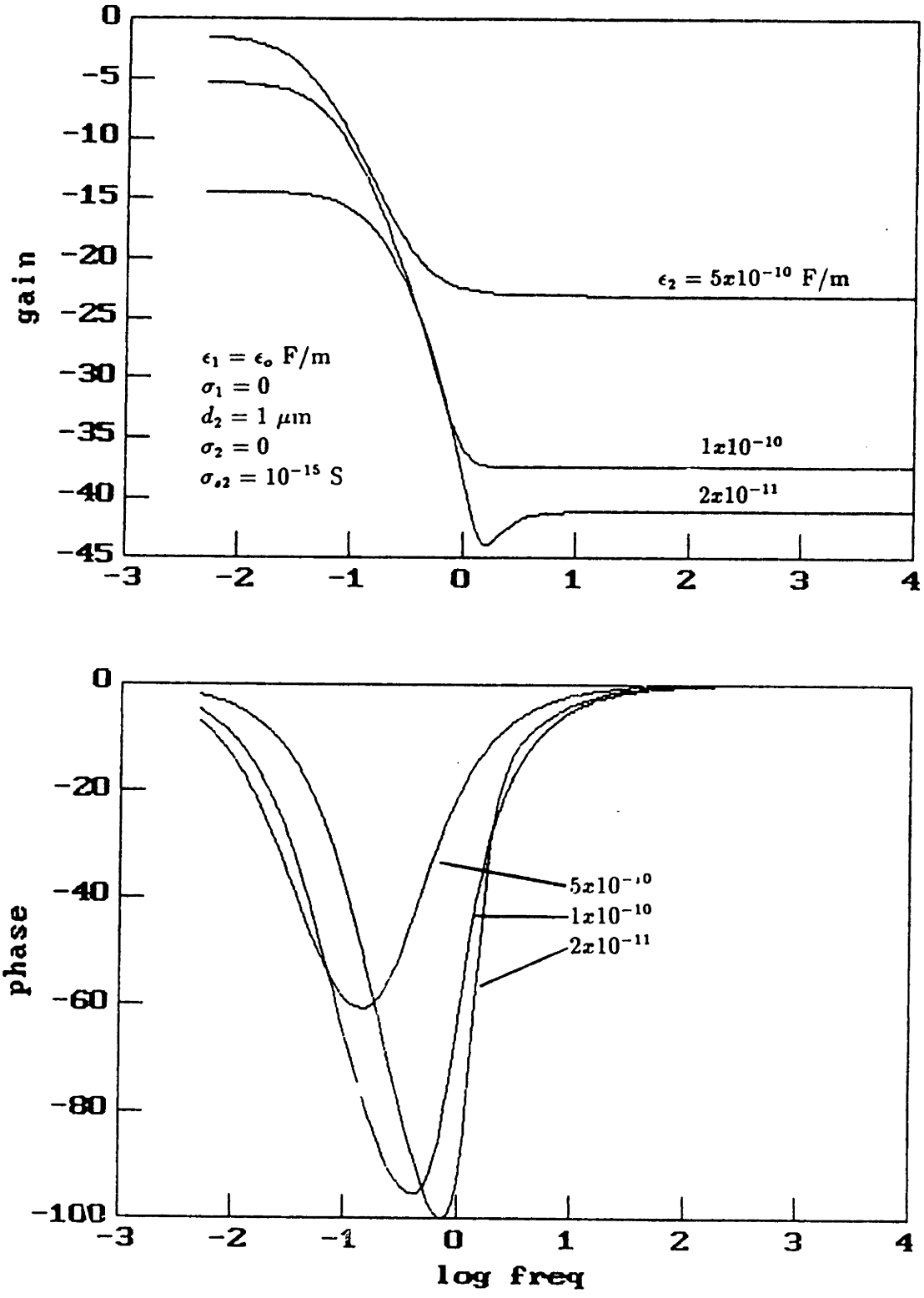


Figure 3.12: Predicted Responses for Microchip with an Insulating Layer of Variable Permittivity ($d_2 = 1$ μm, ϵ_2, $\sigma_2 = 0$ S/m) having a Surface Conductivity ($\sigma_{s2} = 10^{-15}$ S) in Air.

3.3 Comparison With Analytical Solutions

Analytical solutions for the coupling admittances between the electrodes can be found for various limits of (a/λ) and (h/λ). For simplicity, only pure dielectrics will be used for all the limits considered. Thus, only the normalized coupling capacitances per unit length $C_{11}/(\epsilon M_\ell)$ and $C_{12}/(\epsilon M_\ell)$ will be of interest. The four limiting cases are shown in Fig. 3.13.

Case 1 – In the limit of $(a/\lambda) \ll 1$ and $(h/\lambda) \gg 1$ an analytical approximation to the electric field distribution exists. The problem is drawn in Fig. 3.13a for one wavelength. The electric field in polar coordinates has the simple solution,

$$\vec{E} = \frac{V_0}{\pi r}\vec{i}_\theta \tag{3.6}$$

Integrating the surface charge on the grounded electrode, obtained using (2.4), and dividing by the voltage V_0 yields the capacitance per unit electrode length

$$C/M_{el} = \frac{1}{V_0}\int_{a/2}^{(\lambda-a)/2} (\epsilon_b E_\theta^b - \epsilon_a E_\theta^a)\, dr \tag{3.7}$$

Performing the integration yields

$$C/M_{el} = -\frac{\epsilon_a + \epsilon_b}{\pi} \ln\left(\frac{\lambda}{a} - 1\right) \tag{3.8}$$

With a long spatial wavelength, the coupling between the electrodes is concentrated in the adjacent halves of each electrode ($|y| < (\lambda/4)$). For a periodic structure, the total capacitance per electrode unit length is just multiplied by 2. Thus, the effective C_{12}/M_{el} for this analytical limit is

$$C_{12}/M_{el} = -2\frac{\epsilon_a + \epsilon_b}{\pi} \ln\left(\frac{\lambda}{a} - 1\right) \tag{3.9}$$

and can be compared to the value obtained from the continuum model. A tabulation of coupling capacitances for various values of a, k and N (number of collocation points and summation terms), with $(h/\lambda) \sim 10^7$ and $\epsilon_a = \epsilon_b$, is given in Table 3.1. It is seen that the two models are in agreement to better than 4% for $(a/\lambda) = .01$ and .001. The

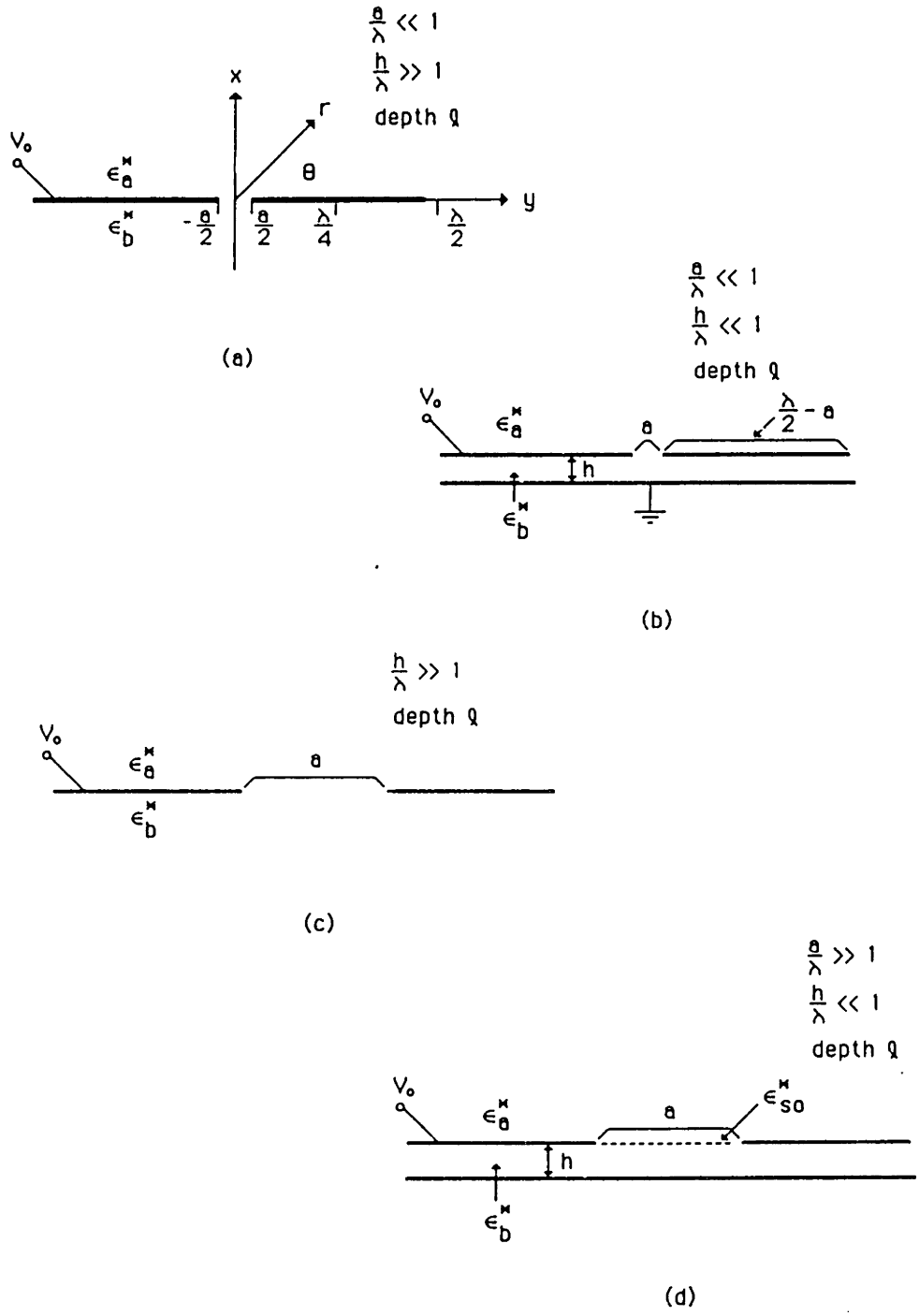


Figure 3.13: a) Case 1 – Small Interelectrode Spacing Limit, b) Case 2 – Parallel Plate Capacitor Limit, c) Case 3 – Microstrip Solution, and d) Case 4 – Long–Wavelength Limit.

Table 3.1: Case 1 – Closely Spaced Electrodes, No Ground Plane.

Model Parameters			Numerical	Analytical	Error
a/λ	k	N	$C_{12}/\epsilon_b M_{el}$	$C_{12}/\epsilon_b M_{el}$	%
10^{-1}	25	100	3.11		11.1
10^{-1}	25	1000	3.23	2.80	-15.4
10^{-1}	45	1000	3.20		-14.3
10^{-2}	25	100	5.74		1.7
10^{-2}	25	1000	6.05	5.85	-3.4
10^{-2}	45	1000	6.05		-3.4
10^{-3}	25	100	7.43		15.5
10^{-3}	25	1000	8.66	8.79	1.5
10^{-3}	45	1000	8.64		1.7
10^{-4}	25	100	7.51		36.0
10^{-4}	25	1000	10.39	11.73	11.42
10^{-4}	45	1000	10.20		13.0

upper bound on a results from the requirement that (3.6) be valid. The lower bound on a limits the intensity of the fringing field at the electrode edges, thus validating the analytic approximation and allowing a reasonable number of collocation points to approximate the field solution. For $(a/\lambda) = .1$ the analytical solution underestimates the electrode coupling. For $(a/\lambda) = .0001$, although there is convergence of $C_{12}/\epsilon_b M_{el}$ with N, 45 collocation points are not enough to effectively map the potential distribution. This is due to the intense field concentration at the electrode edges.

Case 2 – In the limit of $(a/\lambda) \ll 1$ and $(h/\lambda) \ll 1$ the problem approaches that of a parallel plate capacitor (Fig. 3.13b). Within the insulating layer the fields are uniform except for the intense fringing at the electrode gaps. The field is almost totally excluded from the region above the electrodes. Therefore, the coupling capacitance between the electrode and ground plane should be approximated by

Table 3.2: Case 2 – Parallel Plate Capacitor.

Model Parameters		Numerical	Analytical	Error
a/λ	h/λ	$C_{11}/\epsilon_b M_{el}$	$C_{11}/\epsilon_b M_{el}$	%
.01	10^{-2}	49.48	49.00	-.98
.01	10^{-1}	4.97	4.90	-1.4
.1	10^{-2}	41.57	40.00	-3.9
.1	10^{-1}	4.68	4.00	-17.0

$$C_{11}/\epsilon_b M_{el} = (\frac{\lambda}{2} - a)/h \tag{3.10}$$

A comparison between analytical and numerical results gives very good agreement (Table 3.2). For this case, k=25 and N=100, the standard number of collocation points and summation terms.

Case 3 – This case utilizes a solution to the fields problem as derived by Engan [76]. The particular application was for surface acoustic wave devices using interdigital electrodes. In this case, $(h/\lambda) \gg 1$ and (a/λ) is arbitrary (Fig. 3.13c). For a uniform, isotropic medium above the electrodes, the solution to the mixed boundary value problem is a Legendre function of the first kind. The resulting coupling capacitance is

$$C_{12}/\epsilon_b M_{el} = (\epsilon_a + \epsilon_b) \frac{K(\xi)}{K(\sqrt{1-\xi^2})} \tag{3.11}$$

with

$$\xi = \cos(\pi a/\lambda) \tag{3.12}$$

and $K(\xi)$ is a complete elliptic integral of the first kind. Values of $K(\xi)$ are tabulated as a function of $\sin^{-1}\xi$ which reduce, in this case, to

$$\sin^{-1}\xi = \pi(.5 + a/\lambda) \tag{3.13}$$

$$\sin^{-1}\sqrt{1-\xi^2} = \pi a/\lambda \tag{3.14}$$

Table 3.3: Case 3 – Microstrip Calculation.

Analytical Solution					
a/λ	$\sin^{-1}\xi$	$\sin^{-1}\sqrt{1-\xi^2}$	$K(\xi)$	$K(\sqrt{1-\xi^2})$	$C_{12}/\epsilon_b M_{el}$
.25	45°	45°	1.8541	1.8541	2.000
.1	72°	18°	2.5998	1.6105	3.229

Numerical Solution			Error
a/λ	N	$C_{12}/\epsilon_b M_{el}$	%
.25	100	1.93	3.5
.25	1000	1.995	.25
.1	100	3.11	3.7
.1	1000	3.226	.09

Again, with $(h/\lambda) \sim 1$ and $\epsilon_a = \epsilon_b$, very good agreement is obtained (Table 3.3).

Case 4 – This case is the long wave limit treated by Mouayad [73, Section 3.4] and shown to have an analytical solution corresponding to the transmission line model of Garverick [71]. Here $(a/h) \gg 1$, $(h/\lambda) \ll 1$, $\epsilon_b^* = \epsilon_{ox}$ and $|\epsilon_a^*/\epsilon_{so}^*| \ll 1$ (Fig. 3.13d). The first condition allows the approximate normal electric field distribution along the interface

$$\hat{E}_x^b(y) \approx -\frac{\hat{\Phi}^b(y)}{h} \tag{3.15}$$

The last condition implies the normal current into the boundary between the electrodes is negligible compared to the surface currents (see (2.8)), yielding a total differential equation for the potential at the interface

$$\frac{d^2\hat{\Phi}^b(y)}{dy^2} - \frac{\epsilon_{ox}}{\epsilon_{so}^* h}\hat{\Phi}^b(y) = 0 \tag{3.16}$$

With the boundary conditions of (2.3) the solution becomes

$$\hat{\Phi}^b(y) = V_0 \frac{\sinh\left[\alpha\left(y_{k+1} - y\right)\right]}{\sinh\left[\alpha\left(y_{k+1} - y_0\right)\right]} \tag{3.17}$$

with

$$\alpha^2 = \frac{\epsilon_{ox}}{\epsilon^*_{so} h} = \frac{1}{\frac{\epsilon_{so} h}{\epsilon_{ox}} - j\frac{\sigma_{so} h}{\omega \epsilon_{ox}}} \tag{3.18}$$

In support of the application of the device as a humidity sensor, a distributed parameter model was developed and implemented by Garverick [71]. In this situation, a thin film (<1 μm) was coated onto the electrode structure of the CFT and measurements were taken in a temperature and RH controlled environment. Garverick employed a quasi–one dimensional transmission line model. Instead of determining an electric field distribution first and then identifying circuit parameters, each electrode–to–electrode interaction (including the grounded substrate) was immediately represented by a lumped or distributed parameter (Fig. 3.14). In this case

$$\alpha^2 = \frac{1}{\frac{C_A}{C_T} - j\frac{G_A}{\omega C_T}} \tag{3.19}$$

It is interesting that the quasi–one dimensional lumped parameter approach to an inherently two dimensional problem works so well in predicting the measured response. In representing the driven-to-floating electrode impedance through the air as a simple lumped capacitance of constant value, the actual complex field distribution is being disregarded.

A point not well understood is the requirement of a sheet susceptance, C_A. In comparing the expressions for α^2, this susceptance corresponds to a surface permittivity, ϵ_{so}. Using the chip parameters given by Garverick [71] ($\lambda = 50$ μm, $h = 1$ μm, load capacitance of 4.3 pF, $\epsilon_{ox} = 3.9\epsilon_o$ F/m, and total interelectrode meander length of 10.15 mm) it is seen (Fig. 3.15) that a surface permittivity is required by the continuum model to prevent the phase from going positive (although the standard responses reported by Gaverick did not have positive phase, it has been observed [77]). The proposed physical explanation for this surface permittivity is the formation of a space charge layer throughout the thin film caused by the blocking of current flow at the aluminum electrodes. Typical sheet susceptances, when divided by the thickness of

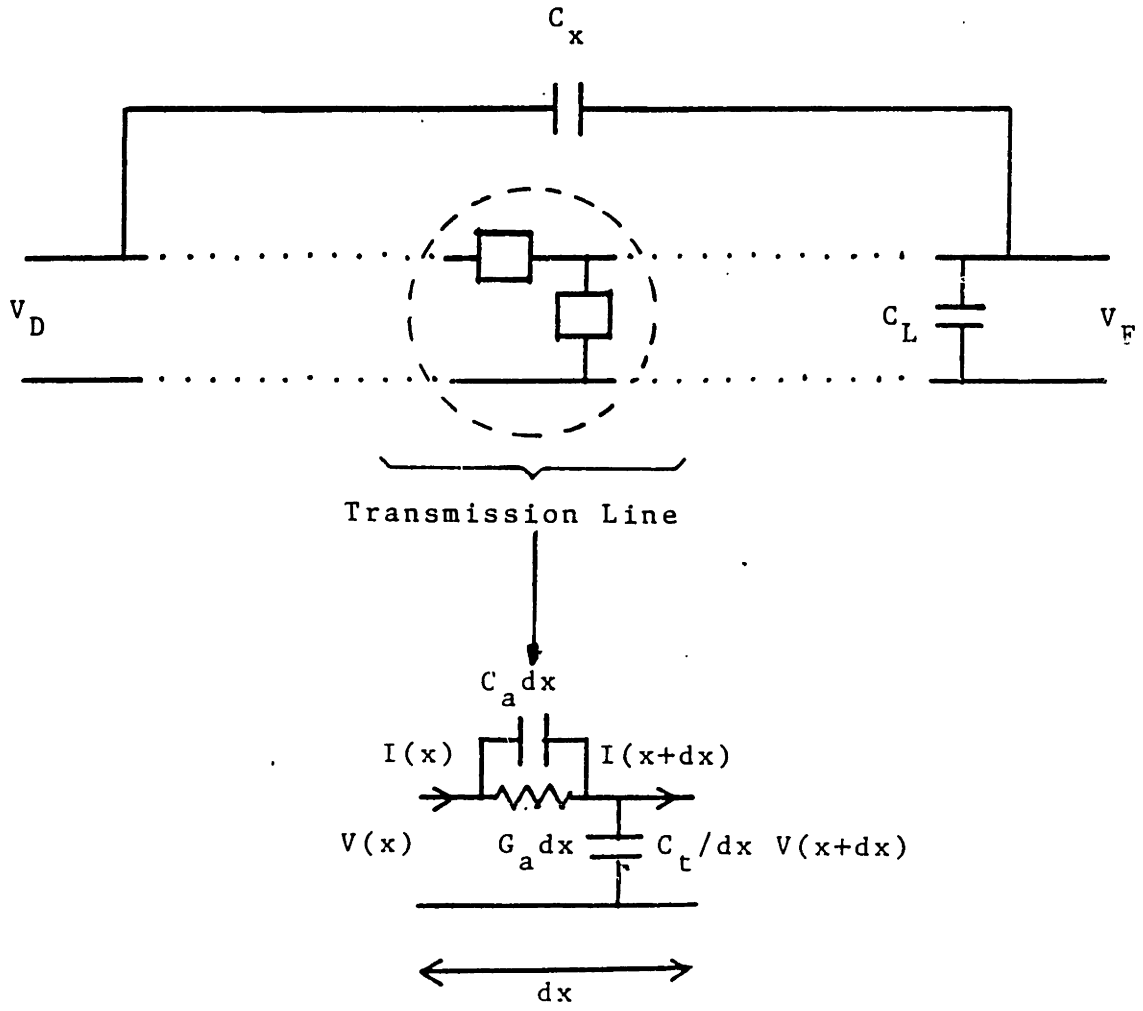


Figure 3.14: Distributed Transmission Line Approximation to Thin Film Coating Microchip in Air (from [71]).

the thin film, yield enormous bulk permittivities that are not physically justified.

A comparison between the continuum model and the transmission line model yielded similar frequency responses, when modeling the old version of the microchip, but values for the surface conductivity and permittivity that differed by ~30% (see Fig. 3.16). The transmission line response was generated using the chip parameters given in [71] producing a high frequency gain in air of -49 dB, a surface conductivity ($1/R_S$) of $1.56x10^{-14}$ S and a surface permittivity (C_S) of $1.6x10^{-16}$ F (corresponding to $C_T/C_A = 30$). However, with a normalized load capacitance of 24.56, the continuum model response was generated with a surface conductivity of $2.02x10^{-14}$ S and a surface permittivity of $1.35x10^{-16}$ F. The difference in parameters is significant as the response is very sensitive to the values of the surface parameters.

Note that the electrode structure parameters play an important role here. The chips used in these thin film studies [71,72] have a thinner oxide layer than those currently manufactured. Taken with the fact that the measurements were made in air (a low dielectric) the RC transmission line capacitive surface coupling provided by the thin film was essential for accurate interpretation of the data. If this situation is altered by having a thicker oxide layer, thicker moisture sensitive coating, or taking measurements in a higher permittivity medium, the resulting increase in the interelectrode capacitive coupling will greatly reduce the need for a surface permittivity. The effect of the thicker oxide layer of the present chip is shown in Fig. 3.17. No positive phase is observed here.

In summary, it is seen that the continuum model is applicable over a wide range of parameter values, encompassing many specialized models. A good contrast in implementation is provided by the finite difference approach. The simplicity of the continuum approach lies in the ease with which various multilayered structures can be represented by altering a subroutine that computes the surface capacitance density. Changing the finite difference model requires, at a minimum, restructuring the mesh pattern, not as easily automated. On the other hand, variations in the tangential (y) direction are easier to model using the finite difference routine, i.e. incorporating the thickness of the electrodes. The tradeoffs should be examined for the particular

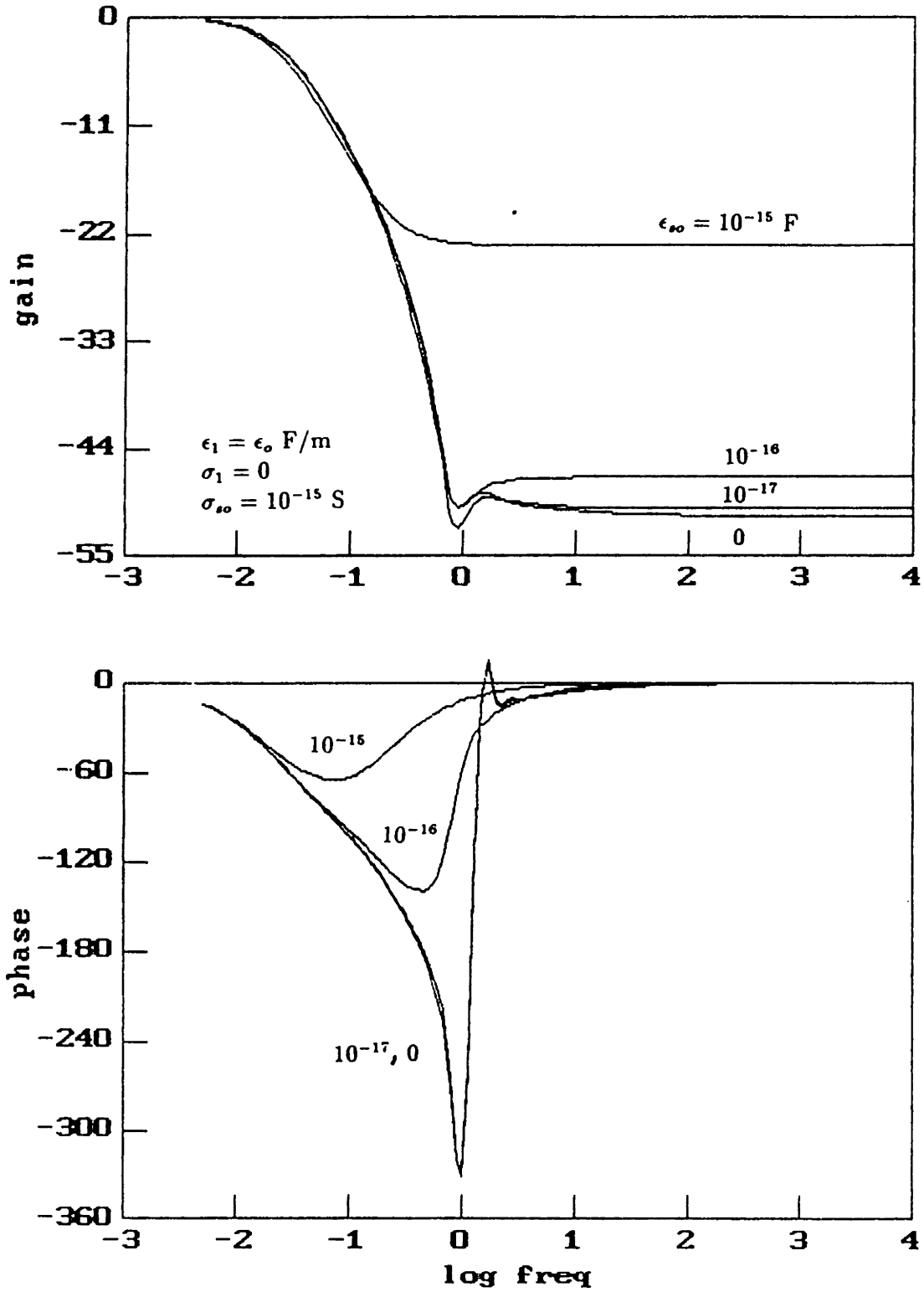


Figure 3.15: Predicted Responses using Old Microchip ($h = 1$ μm) having a Surface Conductivity ($\sigma_{so} = 10^{-15}$ S) and Variable Surface Permittivity (ϵ_{so}).

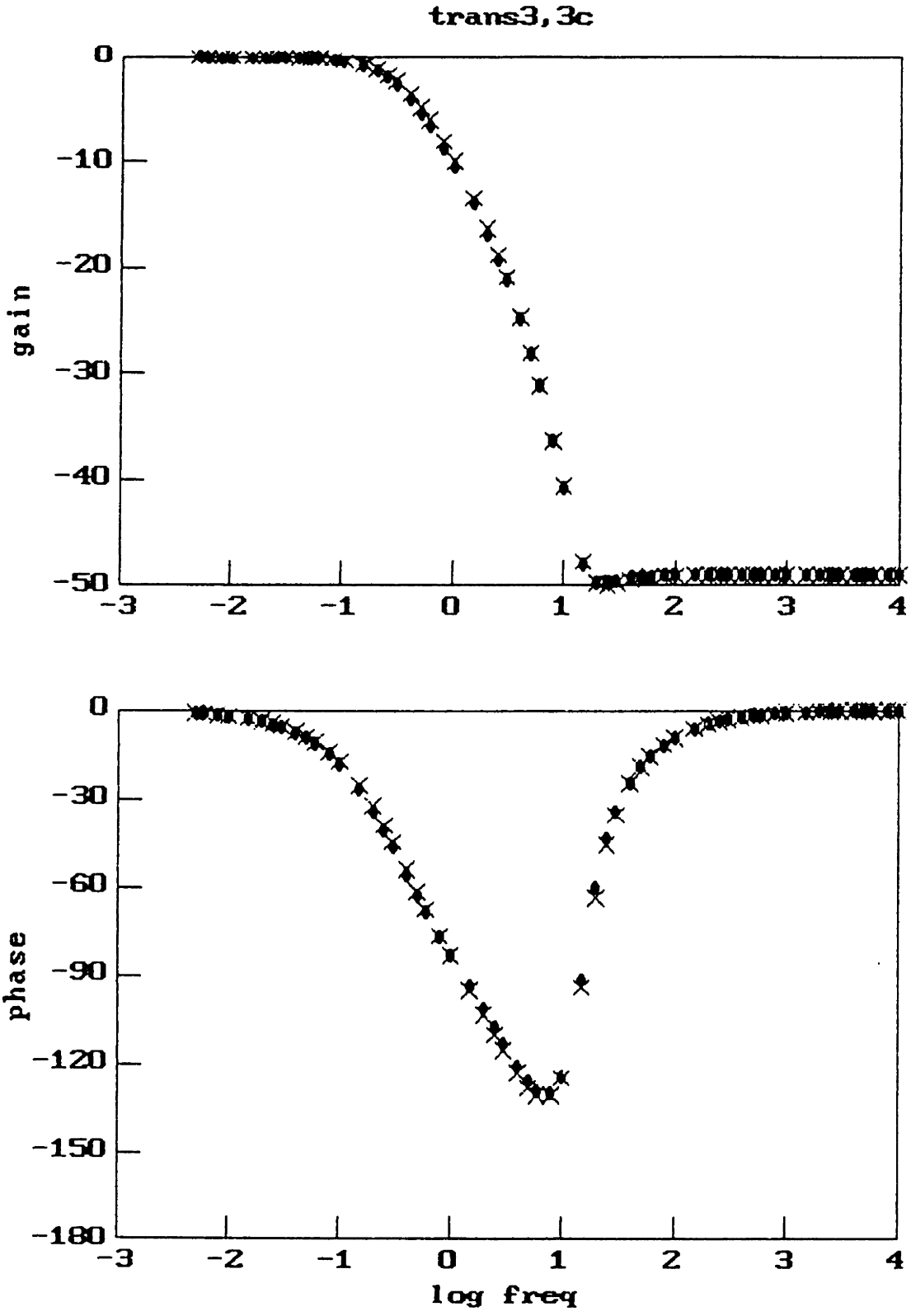


Figure 3.16: Comparison of Predicted Responses using Transmission Line Model (•) and Continuum Model (X) for a Thin Film Coating Old Microchip.

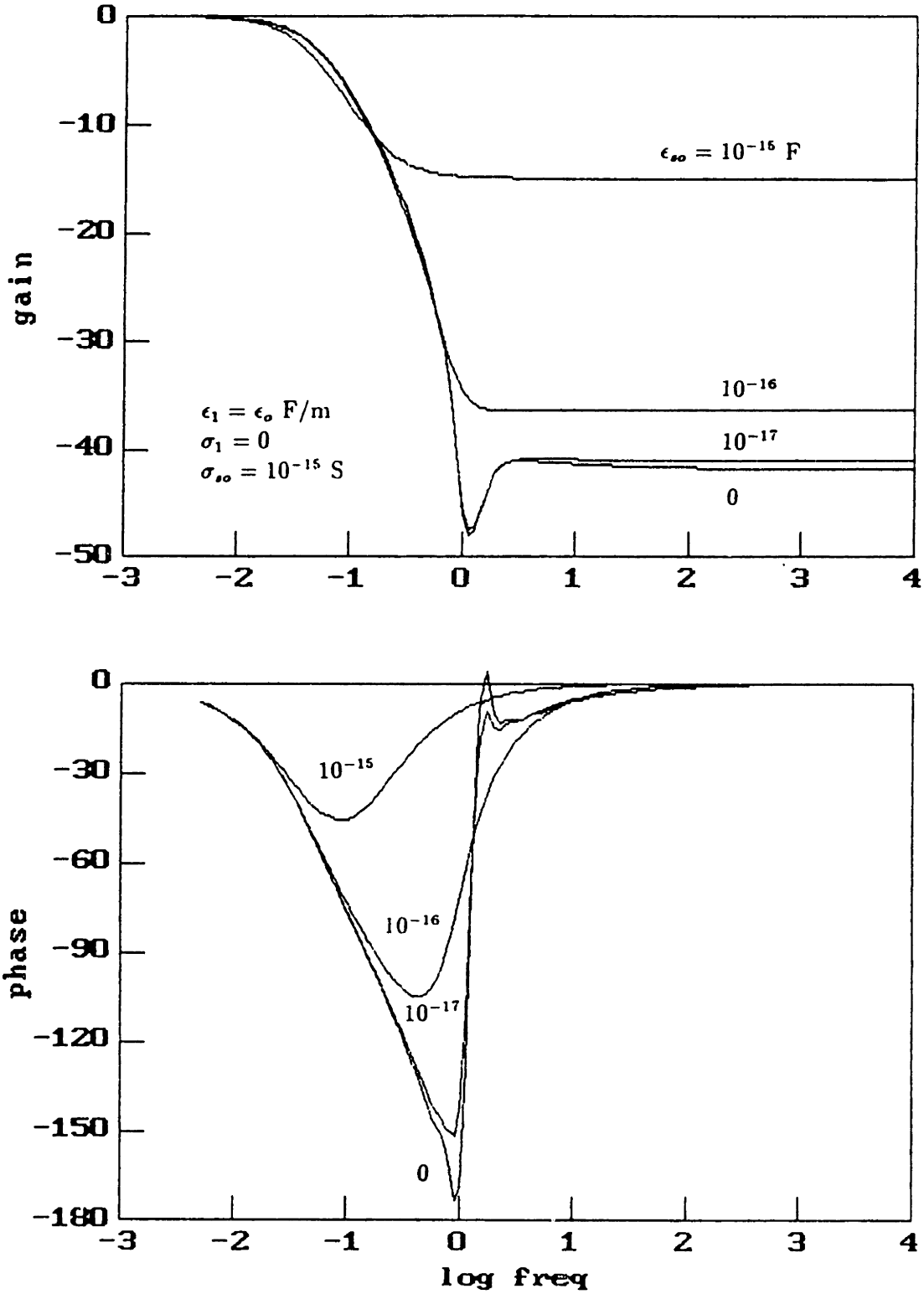


Figure 3.17: Predicted Responses using Present Microchip having a Surface Conductivity ($\sigma_{so} = 10^{-15}$ S) and Variable Surface Permittivity (ϵ_{so}).

application desired.

Chapter 4

Parameter Estimation Techniques

In the previous chapter, the problem was posed as one of determining a frequency response given all the pertinent complex permittivities, layer thicknesses and other relevant parameters describing the media. In practice, the situation is usually reversed. The unknown quantity is a material property such as complex permittivity (bulk or surface, distribution if inhomogeneous) or layer thickness. What is known is one or more experimentally measured responses, perhaps at different temporal or spatial frequencies. A parameter estimation scheme is required in which all *a priori* knowledge of the physical situation is utilized, i.e. number of layers, values of complex permittivities for each layer, layer thicknesses, and, based upon one or more measurements with respect to changes in a known parameter, the quantities of interest are deduced using a root searching routine. Noise or other stochastic processes affecting the experimental data call for more sophisticated techniques, not considered here. Thus, here the estimation routines may be viewed either as root searching or function minimization routines [78, Section 7.2].

General algorithms for single and multi- parameter estimation, as typically seen in optimization and numerical analysis references [78, Section 10.2][79, Chapters 4 and 5], are presented in Section 4.1. The motivation for using a secant method for single parameter estimation, and a derivation of the Newton search routine using a quadratic norm for multiparameter estimation, is given. A discussion of single parameter estimation routines for estimating layer thickness and complex bulk and surface

permittivities is presented in Section 4.2. The ability to uniquely ascertain spatial distributions in complex bulk permittivity using variable spatial wavelength data is described in Section 4.3. Several case studies are given, demonstrating the utility of extending the single parameter estimation routine to this multiparameter estimation problem. Finally, Section 4.4 describes a two parameter estimation routine that requires the use of a the Newton's search method.

For application to microdielectrometry, acceptable tolerances for convergence are determined by the experimental error in the response. The present version of the device, when using a microchip sensor, has an accuracy of 0.1 dB in gain and .1° in phase [67]. In the routines, the gain tolerance is tightened to 0.05 dB. This translates to a tolerance of 0.5% in gain at -40 dB and 0.1% in phase at $-90°$.

These routines were developed on an IBM PC/AT with an 8 Mhz clock and an 80286/287 microprocessor/math co-processor. The emphasis was on developing search routines that converged and were somewhat robust, not on generating the optimal search method. Most of the validation of these search algorithms was obtained using experimental data on semi-insulating materials. Consequently, there are areas of parameter space where the convergence and robustness of these routines have not been examined.

4.1 General Algorithms

4.1.1 Single Parameter – Secant Method

The search is for the root to the equation

$$\mathbf{e}(\theta) = 0 \tag{4.1}$$

where $\mathbf{e}(\theta)$ is a set of error functions (usually the difference between one or more measured and predicted gains) and θ is a list of parameters to be estimated (such as complex permittivities). Of course, the particular physical phenomenon associated with the parameters to be estimated must make a significant contribution to the gain,

and hence, the error.

There are many different methods used to conduct searches for roots. Some require only evaluation of a function, others, such as Newton methods, also use derivative information. In one dimension, the Newton–Raphson update to a guess is

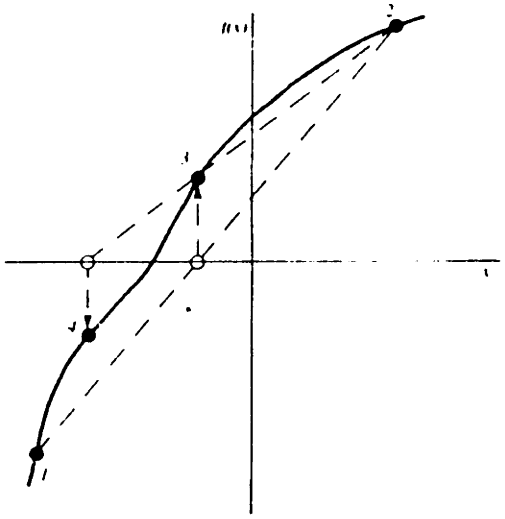
$$\Delta\theta_i = -\frac{e(\theta_i)}{\left(\frac{de(\theta_i)}{d\theta}\right)} \tag{4.2}$$

Along with the extra information provided by the derivative is an extra cost. If an analytic derivative is not available then it must be approximated numerically, greatly increasing the number of function evaluations. For a one dimensional search, if the method without derivatives converges in less than twice the number of iterations of a search using derivatives, then the advantage of the derivative is lost. This tradeoff becomes more important as the number of parameters increases and as the computation time for a function evaluation increases.

Given the complex nature of the continuum model, there is no simple analytic derivative with respect to parameters such as layer thickness or complex permittivity. The tradeoff described above applies here, especially given the time required for a function evaluation. A secant method of searching was employed for most of the parameter estimations because of this tradeoff [79, Section 5.1.1]. In this method, guesses, θ_i, are updated by $\Delta\theta_i$ using the secant formed by the two most recent guesses,

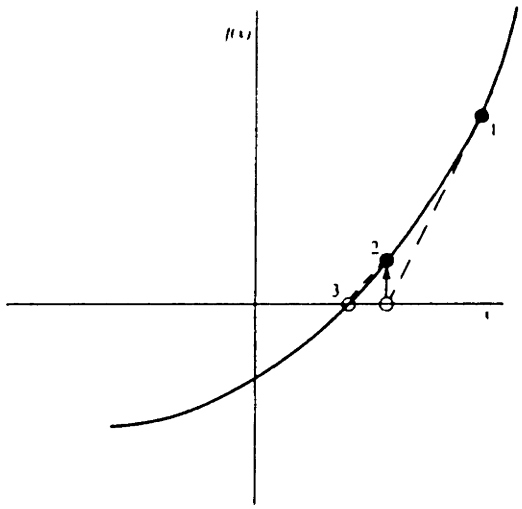
$$\Delta\theta_i = \frac{e(\theta_i)}{\left(\frac{e(\theta_i) - e(\theta_{i-1})}{\theta_i - \theta_{i-1}}\right)} \tag{4.3}$$

in the neighborhood of the root, the secant approaches the tangent, thus the secant method is also known as a quasi–Newton method. The danger that the root does not necessarily remain bracketed by the two guesses is decreased by providing bounds on the minimum and maximum values of the estimated parameter based upon physical constraints. The difference between the secant and Newton–Raphson methods is illustrated graphically in Fig. 4.1.



Secant method. Extrapolation or interpolation lines (dashed) are drawn through the two most recently evaluated points, whether or not they bracket the function. The points are numbered in the order that they are used.

(a)



Newton's method extrapolates the local derivative to find the next estimate of the root. In this example it works well and converges quadratically.

(b)

Figure 4.1: Comparison of a) Secant Method and b) Newton's Method for Root Searching (from [80, p. 249 and 255]).

4.1.2 Multiparameter – Minimization of Quadratic Error Function

The general approach for multiparameter estimation using derivative information is outlined below (adapted from [78, p. 282]. This formalism will be used for the multiparameter estimation of layer thickness and complex bulk permittivity and complex bulk permittivity and surface conductivity. For these estimation problems the objective is to minimize a quadratic norm defined as

$$V(\theta) = \frac{1}{2} e^{T}(\theta) W e(\theta) \tag{4.4}$$

where V is a nonnegative scalar, θ is a list of one or more parameters to be estimated, $e(\theta)$ is a column vector of errors and W is a symmetric, positive semidefinite weighting matrix. Although these quantities are usually treated as real functions of real variables, there appears to be no reason why the concept cannot be extended to include complex functions of complex variables, as is done in Sections 4.3.6 and 4.3.7. When $\theta = \theta_o$, the correct values, the quadratic norm will be at a minimum and its gradient, $V'(\theta_o)$ will be zero. Expanding in a Taylor series in the neighborhood of θ_o yields

$$V'(\theta_o) = V'(\theta) + V''(\theta)\Delta\theta = 0 \tag{4.5}$$

or

$$\Delta\theta = -[V''(\theta)]^{-1} V'(\theta) \tag{4.6}$$

This form of the increment to update guesses is a common one for many search routines utilizing the derivative of a function. For methods other than the Newton search, alterations can made to $V''(\theta)$ to improve convergence and robustness. Using the quadratic norm defined above (4.4)

$$\begin{aligned} V'(\theta) &= e'^{T}(\theta) W e(\theta) \\ V''(\theta) &= e'^{T}(\theta) W e'(\theta) \end{aligned} \tag{4.7}$$

where $e'(\theta)$ is the jacobian of $e(\theta)$. The form of the jacobian is

$$e'(\theta) = -\begin{bmatrix} \frac{\partial e_1}{\partial \theta_1} & \cdots & \frac{\partial e_1}{\partial \theta_m} \\ \vdots & & \vdots \\ \frac{\partial e_n}{\partial \theta_1} & \cdots & \frac{\partial e_n}{\partial \theta_m} \end{bmatrix} \tag{4.8}$$

where there are n experimental data points, m parameters to be estimated and $n \geq m$. The increment to the old guess becomes

$$\Delta\theta = -\left[e'^{\mathrm{T}}(\theta)\mathrm{W}e'(\theta)\right]^{-1}\left[e'^{\mathrm{T}}(\theta)\mathrm{W}e(\theta)\right] \tag{4.9}$$

which, if $e'(\theta)$ is square and both $e'(\theta)$ and W are nonsingular, reduces to

$$\Delta\theta = -e'^{-1}(\theta)e(\theta) \tag{4.10}$$

It is easily verified that (4.10) reduces to (4.2) for one dimensional searches.

4.2 Single Parameter Estimation

For these estimations, the error is defined as the difference between the measured and the predicted complex gain,

$$\hat{e}(\theta) = \hat{\mathcal{G}}_m - \hat{\mathcal{G}}_p(\theta) \tag{4.11}$$

where θ is the complex parameter to be estimated. Here, $\hat{\mathcal{G}}_m$ and $\hat{\mathcal{G}}_p$ are formed by evaluating the complex logarithm of the complex voltage ratio,

$$\hat{\mathcal{G}} \equiv \log \hat{\mathrm{G}} = \log\left|\frac{\hat{\mathrm{V}}_\mathrm{F}}{\hat{\mathrm{V}}_\mathrm{D}}\right| + j \log e \, \angle(\hat{\mathrm{V}}_\mathrm{F}/\hat{\mathrm{V}}_\mathrm{D}) \tag{4.12}$$

where the real part is the gain in decibels divided by 20 and the imaginary part is the phase in radians (closely related to the output of the microdielectrometer). If $\hat{\mathrm{G}}$, as defined in (2.1), is a complex analytic function of the complex variable θ, then $\hat{\mathcal{G}}_p$ and hence $\hat{e}$ are also. Thus, for the class of constitutive laws used here, it is meaningful to

directly apply one–dimensional root finding techniques to the estimation of a complex parameter.

Note that although one parameter is being estimated, this does not constrain the medium to be uniform. Using the surface capacitance density, $\hat{C}_n^a$, heterogeneous media can be represented as long as all but one of the parameters necessary to characterize the medium's dielectric properties is known.

4.2.1 Layer Thickness

This is the case of a layer and a surrounding uniform medium, as modeled in Section 3.1.2. At high frequencies, when the electrodes are purely capacitively coupled through the material, the phase will be zero and the gain will be a real function of the layer thickness and the permittivities of the layer and the surrounding medium. Given any three of these four real quantities, it is possible to estimate the fourth. Here the layer thickness (d_2) is unknown thus $\theta = d_2$ and the search is for a purely real root.

For the case of air as the surrounding medium, the curves of gain versus layer thickness for various layer permittivities are shown in Fig. 4.2. Finding the root of these functions is relatively straightforward. For robustness, the routine first checks whether the experimental data makes sense by calculating the gain of a uniform medium having the lesser of the two permittivities. This gain should be less than the experimental gain if a layer thickness is to be estimated. After passing this check the routine uses the secant method to find the root. The search is conducted using the finest grid of 25 points (k) and 100 summation terms (n). Due to the well behaved nature of the curves, the thickness is uniquely estimated with two place accuracy in the associated gain within 3 to 5 iterations. For a 50 micrometer wavelength electrode structure this implies a hundredth of a micrometer accuracy in the estimate for thicknesses in the range of 1 to roughly 15 micrometers. The lower bound may be set by features of the electrode that are not modeled such as electrode thickness or by the experimental error in the measurement. The upper bound reflects the decrease in sensitivity due to the exponential decay of the electrostatic fields.

The software for this thickness estimation routine, **thest.for**, may be found in Ap-

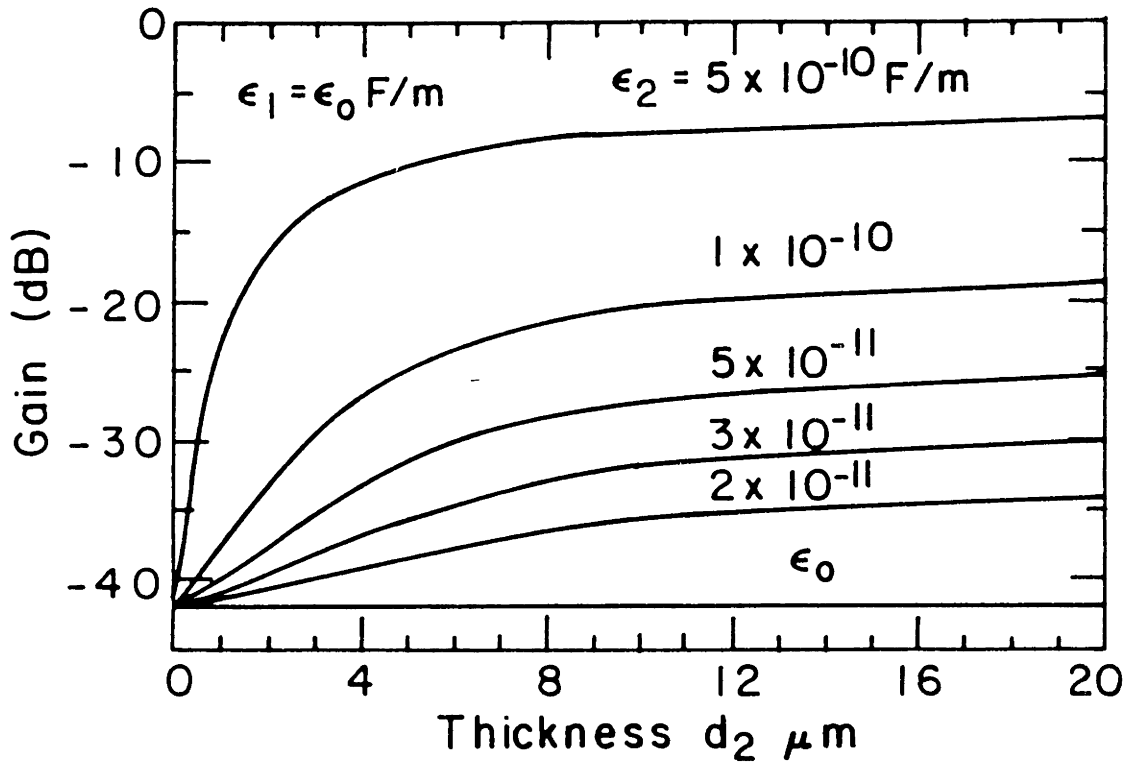


Figure 4.2: Predicted High Frequency (10 kHz) Gain for Microchip with Variable Thickness Layer (d_2), Vary Layer Permittivity (ϵ_2), in Air.

pendix F. This routine was used to interpret experimental results regarding deposition rate for the plasma deposition process and evaporation of thin oil films (Section 5.2.1), and sedimentation of monodisperse particles [23, Chapter 6].

4.2.2 Complex Permittivity (Bulk or Surface)

As long as $\hat{\mathcal{G}}_p(\theta)$ is an analytic function of a complex bulk or surface permittivity, the search is only slightly more complicated. A complex bulk or surface permittivity is estimated using a complex data point (gain and phase) and specifying all other relevant parameters such as the complex bulk permittivities of the other regions, the complex surface permittivities at the interfaces and the layer thicknesses. Now, in (4.3), $\theta = \epsilon' - j\epsilon''$ or $\epsilon_s' - j\epsilon_s''$ and $\hat{\mathrm{e}}(\theta)$, in (4.11), is complex.

The economy of using the secant, instead of the Newton method, is demonstrated here. The Newton method would require a function evaluation in the neighborhood of the most recent guess so as to compute a local difference approximation to the first derivative. When guesses are far from the root this second function evaluation will usually have no chance of producing the root. However, with the secant method, each evaluation has a reasonable chance of producing the root.

The search routine can be started either with a user input guess or with its own "guess" as to the complex permittivity. For bulk properties this "guess" corresponds to that of an insulating, nonpolar medium. The first update only is calculated using the local derivative. For the secant method the order of convergence is the 'golden ratio' of 1.618... [80, p. 248]. The key here is to get in the neighborhood of the root where the convergence is rapid. A substantial reduction in the time to convergence is obtained by letting the routine start with its own "guess" and using a technique of varying the number of collocation points used for the function evaluation. This technique performs sequential searches beginning with a coarse discretization of 2 grid points and ending with 25 grid points. After a root is found at 2 grid points, that root is used to begin a new search using 10 grid points. Again, once convergence has been reached, this new root is used to begin a search at 25 grid points. Generally anywhere from 20–40 iterations are required for convergence with 2 grid points resulting in only

1–10 iterations at 25 grid points. Without this technique, convergence can easily take 20 or more iterations when using only 25 grid points. It takes roughly 12 times longer to compute a gain–phase response using 25 collocation points versus 2 collocation points. Thus, a substantial savings in computing time can be realized.

A two-dimensional view of the space over which the routine searches when estimating the complex bulk permittivity of a layer 5 micrometer thick surrounded by a medium with a bulk permittivity of $\epsilon_1' = 2x10^{-11}$ F/m and a loss factor of $\epsilon_1'' = 1x10^{-8}$ F/m is shown in Fig. 4.3. This space is constructed by computing the gain–phase response for a matrix of complex bulk permittivities of the layer at a constant frequency. The contours are lines of constant ϵ_2' and ϵ_2'' and are orthogonal because $\hat{\mathcal{G}}$ is an analytic function of ϵ^*. Reiterating, this orthogonality justifies the computation of the complex slope used by the one dimensional secant search routine. As required by (4.12), the gain is in decibels divided by 20 and the phase in radians multiplied by log e. This figure is analogous to the lookup table given by Senturia, et al. [69] for the case of a uniform medium. With the gain defined by (4.12) the contours are orthogonal for the uniform medium case too (Fig. 4.4).

In Fig. 4.3, the values of ϵ_2'' that make the layer more lossy than the upper medium are concentrated near the origin. The dot in Fig. 4.3 is where the layer and surrounding medium have equal complex permittivities. By contrast, in Fig. 4.5 the upper medium is sufficiently insulating that the point of equal complex permittivities (the dot) is near the zero phase axis. Again, with a 5 micrometer thick layer, the surrounding medium now has a bulk permittivity of $\epsilon_1' = 2x10^{-11}$ F/m and a loss factor of $\epsilon_1'' = 1x10^{-12}$ F/m. The kink that appears in the left side of the plot is a result of charge diffusion along the layer, as discussed earlier in regards to the thin film humidity sensor. This kink is more evident when the layer is in a perfectly insulating medium such as air and/or the layer is thinner, say 2 μm (Fig. 4.6).

Given the well behaved nature of the spaces represented by Figs. 4.3 and 4.5, it is expected that estimation of unique complex permittivities is a straightforward process. The sensitivity of the estimation routine can be observed from the two figures. Poor estimates occur in regions of high contour density. For example, gain–phase data

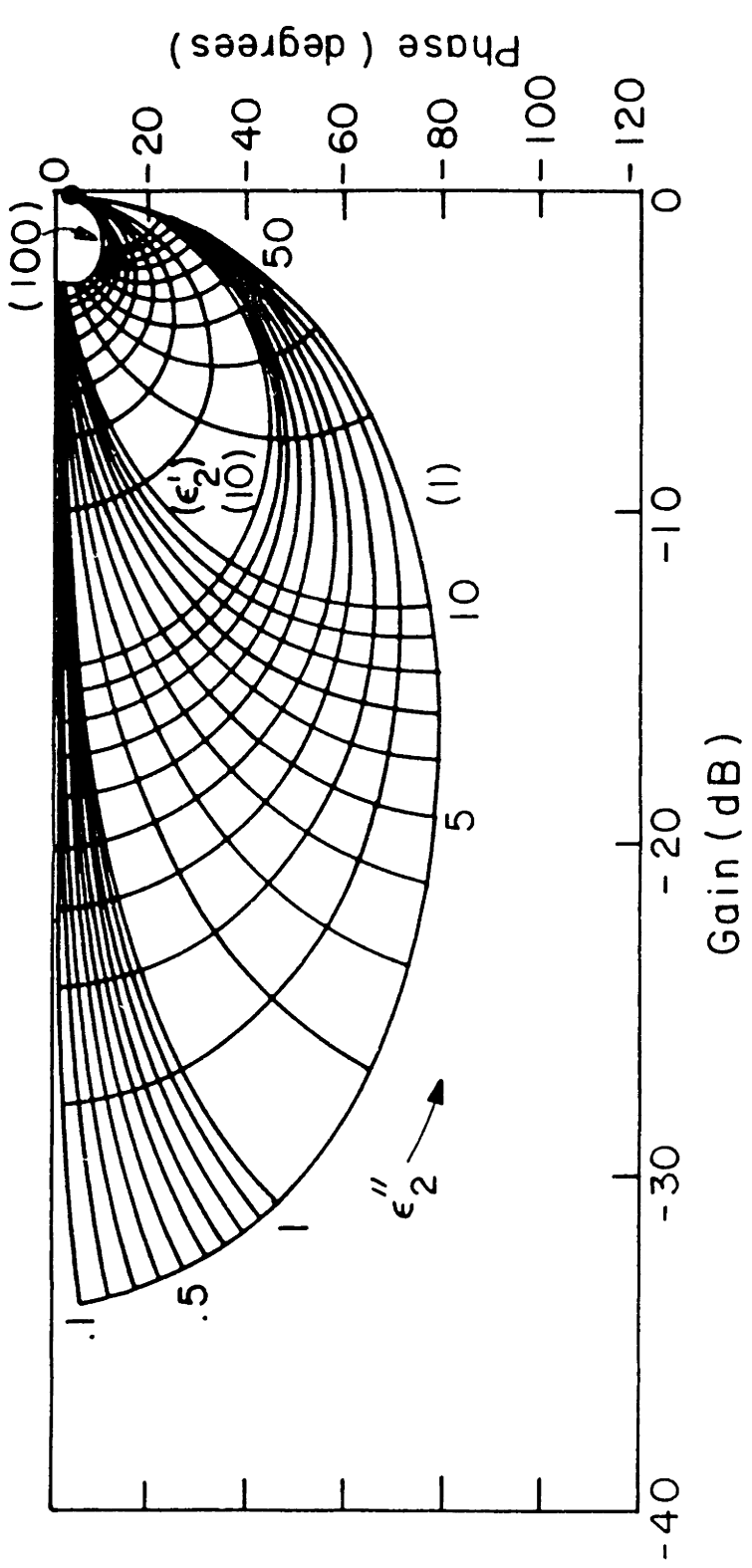


Figure 4.3: Parameter Space for Complex Bulk Permittivity Estimation of a 5 μm Layer in a Uniform Medium ($\epsilon_1' = 2x10^{-11}$ F/m and $\epsilon_1'' = 10^{-12}$ F/m) with ϵ_2' and ϵ_2'' in units of ϵ_o F/m.

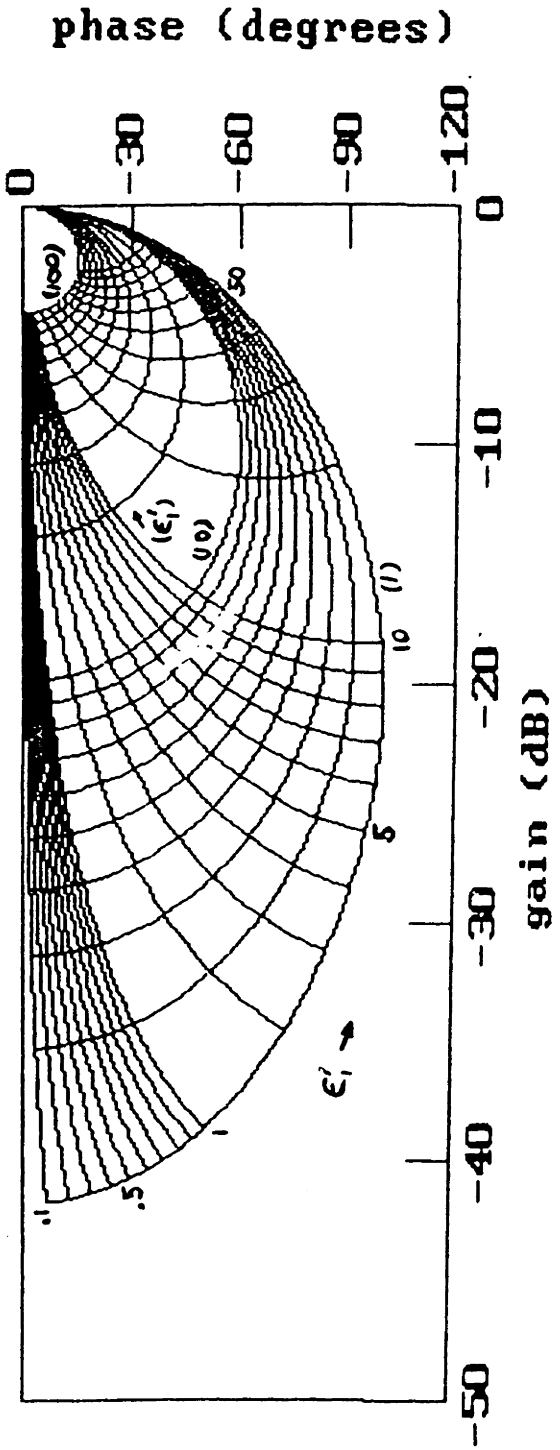


Figure 4.4: Parameter Space for Complex Bulk Permittivity Estimation of a Uniform Medium with ϵ_1' and ϵ_1'' in units of ϵ_o F/m.

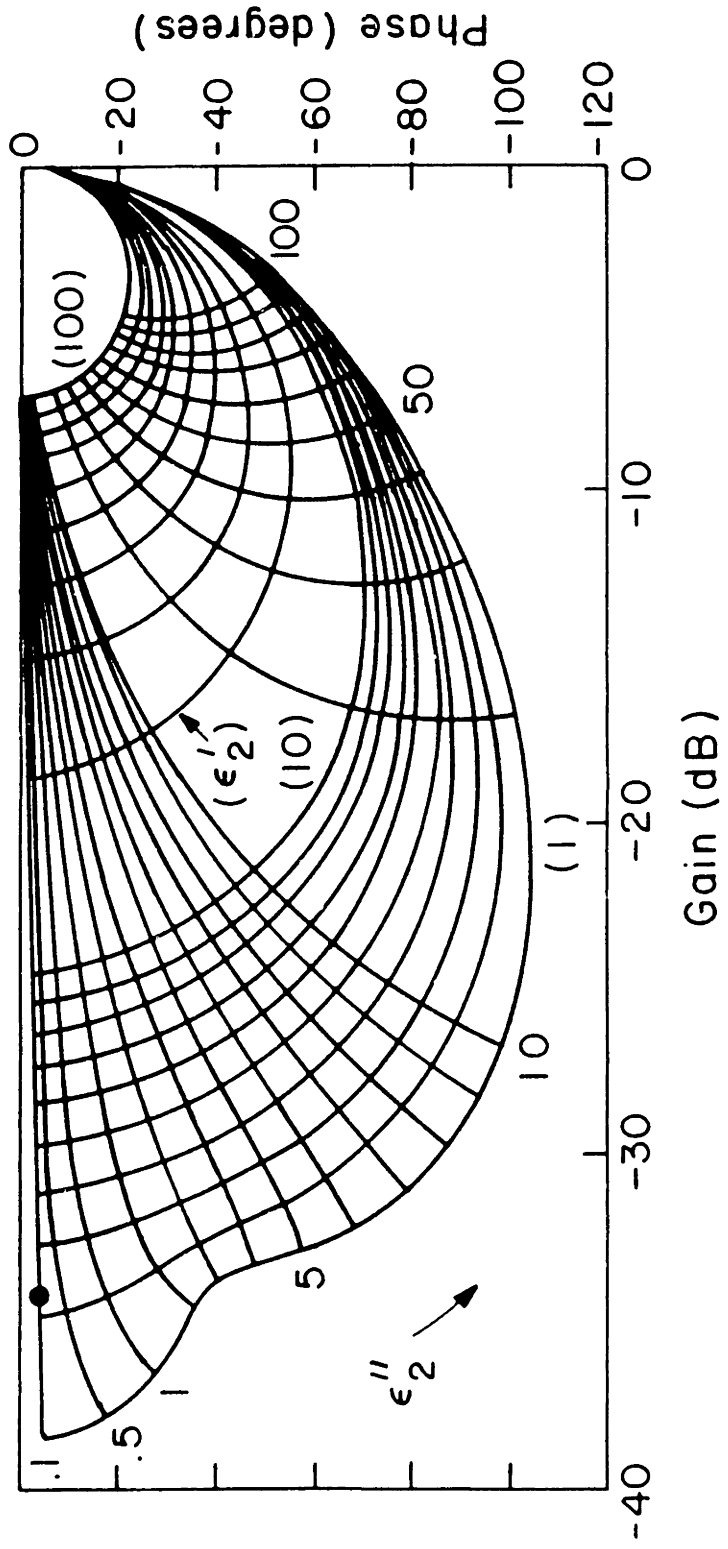


Figure 4.5: Parameter Space for Complex Bulk Permittivity Estimation of a 5 μm Layer in a Uniform Medium ($\epsilon_1' = 2x10^{-11}$ F/m and $\epsilon_1'' = 10^{-8}$ F/m) with ϵ_2' and ϵ_2'' in units of ϵ_o F/m.

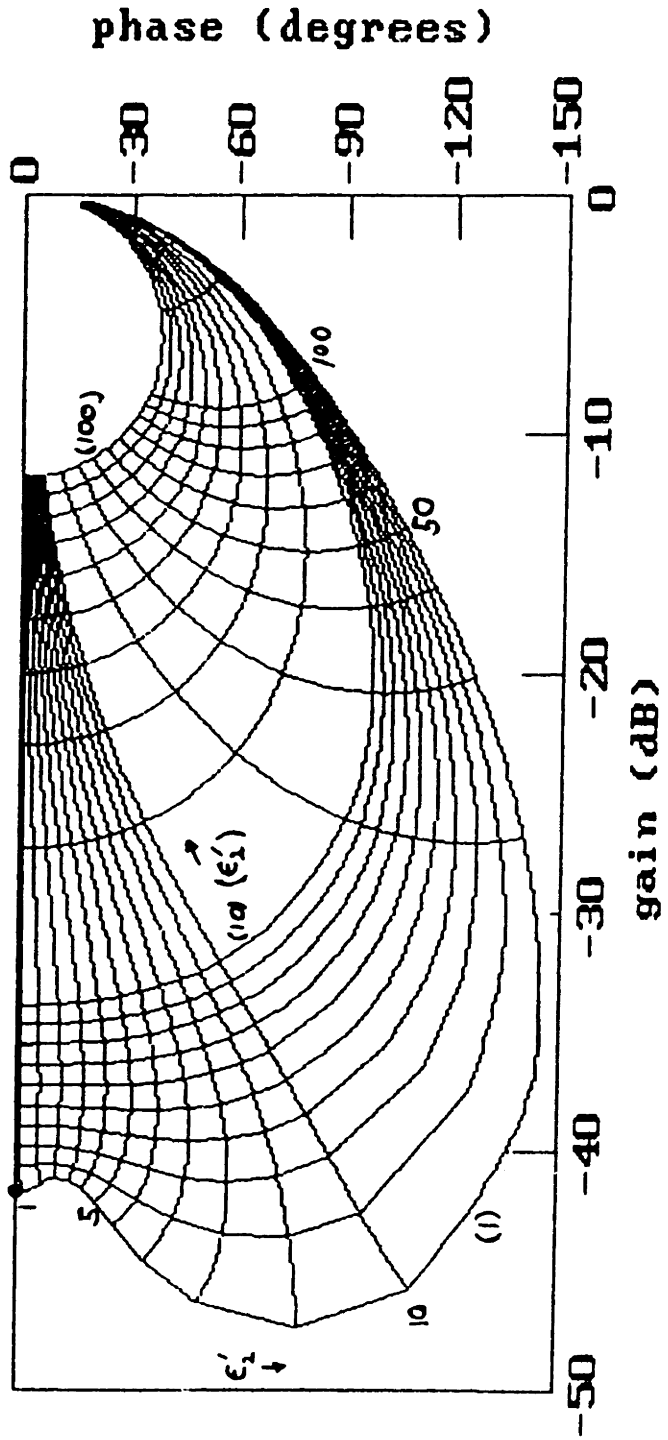


Figure 4.6: Parameter Space for Complex Bulk Permittivity Estimation of a 2 μm Layer in Air ($\epsilon_1' = \epsilon_o$ F/m and $\epsilon_1'' = 0$) with ϵ_2' and ϵ_2'' in units of ϵ_o F/m.

adjacent to the gain axis yield poor loss factor estimates while data with gains near zero yield poor permittivity estimates. The latter is a consequence of the high impedance mode of operation which utilizes the measurement of a floating voltage.

For the special case of ohmic media, $\epsilon'' = (\sigma/\omega)$. Thus, contours of constant ϵ' are also the gain–phase trajectories obtained if the temporal excitation frequency was varied. This may also be seen by eliminating the independent parameter of frequency from the plots in Fig. 3.2.

For the gain–phase spaces of Figs. 4.3–4.6, valid data must lie in the region having a lower bound given by the $\epsilon_i' = \epsilon_o$ curve. Upper and lower bounds can be placed on the parameter estimates to prevent the search from wandering too far off. These bounds are physically motivated – no bulk permittivities less than ϵ_o and no conductivities less than zero. Upper bounds are chosen based on reasonable guesses of the order of magnitude of the dielectric properties of the materials to be measured. Another common technique for preventing endless searches is to limit the maximum number of iterations a routine can search. Based on experience in using the routines documented here an upper limit of 50 iterations for a given grid point estimation is established.

In addition to estimating bulk parameters, this routine works just as well for estimating complex surface parameters.

The software for this complex permittivity estimation routine, **parest.for**, may be found in Appendix F. This routine was used to estimate the complex bulk permittivities of oil (Section 5.4 and [23, Chapter 4]), the complex surface permittivities of adsorbed layers (Section 5.4.1 and 5.5) and the dispersive dielectric properties of the plasma deposited bromobenzene (Section 8.2.3).

4.3 Multiparameter Estimation Using Variable Spatial Wavelengths

4.3.1 Motivation

Using the temporal frequency response, it is not possible to uniquely determine the spatial distribution in dielectric properties of a heterogeneous medium. However, this spatial distribution is uniquely observable through extraction of information using variable spatial wavelength electrodes [82].

The first claim is substantiated with the following example. Let a sample having a spatial distribution in complex permittivity, $\hat{\epsilon}(x)$ be placed between a pair of parallel plate electrodes as in Fig. 4.7. An electric field is imposed on the sample via a voltage source and the charge on the electrodes is sampled with an electrometer. With the charge on the electrodes defined as q such that the current $i = (\mathrm{d}q/\mathrm{d}t)$, the response to the applied voltage having the frequency ω is defined as

$$\hat{C}(\omega) \equiv \frac{\hat{q}}{\hat{v}} \tag{4.13}$$

In the bulk of the material the electroquasistatic fields are described by Gauss' Law

$$\nabla \bullet \left(\hat{\epsilon}(x)\hat{\vec{E}}\right) = 0 \tag{4.14}$$

with the requirement that the electric field be irrotational and therefore related to the complex potential $\hat{v}$ by

$$\hat{\vec{E}} = -\nabla\hat{\Phi} \tag{4.15}$$

The purpose of the guard electrode is to eliminate any fringing fields, producing an electric field having only a normal (x) component and uniform within the bulk. However, due to the spatial variation in complex permittivity with the normal component, the electric fields will be nonuniform in x. The effect of this nonuniformity in

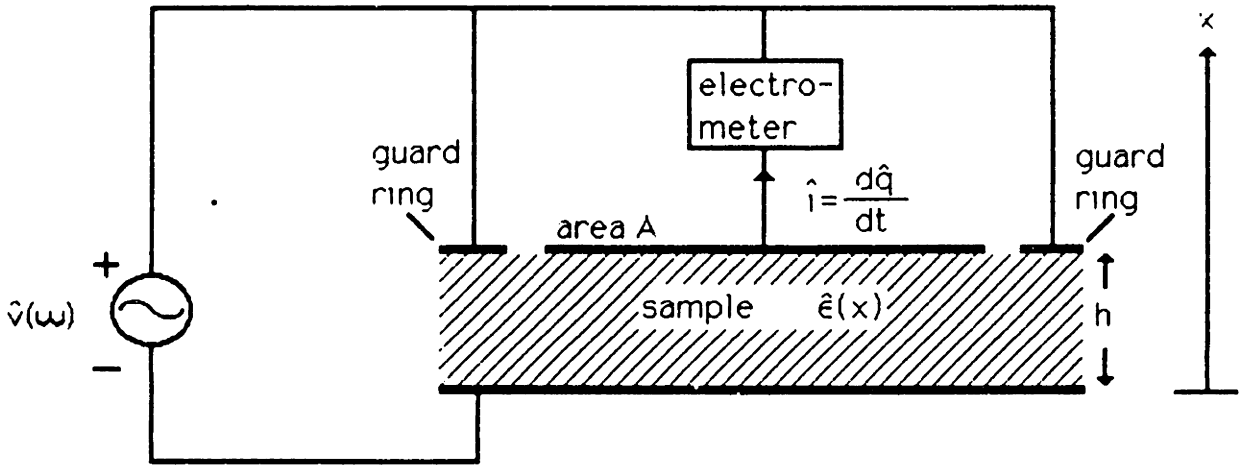


Figure 4.7: Complex Permittivity Measurement of Heterogeneous Medium Using Parallel Plates with Guard Ring.

the measured temporal frequency response can be seen by writing (4.14) and (4.15) as, respectively

$$\frac{\mathrm{d}\left(\hat{\epsilon}\hat{\mathrm{E}}_x\right)}{\mathrm{d}x} = 0 \tag{4.16}$$

$$\hat{\mathrm{E}}_x = -\frac{\mathrm{d}\hat{\Phi}}{\mathrm{d}x} \tag{4.17}$$

Integration of (4.16) yields

$$\hat{\epsilon}\hat{\mathrm{E}}_x = \hat{\sigma}_{su} \tag{4.18}$$

where $\hat{\sigma}_{su}$ is the magnitude of the free surface charge density on either electrode. Integration of (4.17) from the lower electrode, $x = 0$, where the potential is zero, to the upper electrode, $x = h$, where the potential is the applied voltage $\hat{v}$, and substituting for $\hat{\mathrm{E}}_x$ using (4.18) yields

$$\hat{v} = \hat{\sigma}_{su}\int_0^h \frac{\mathrm{d}x}{\hat{\epsilon}(x)} \tag{4.19}$$

Using $\hat{q} = A\hat{\sigma}_{su}$, where A is the surface area of the center electrode, and substituting (4.19) into (4.13) yields

$$\hat{C}(\omega) = A\int_0^h \frac{dx}{\hat{\epsilon}(x)} \tag{4.20}$$

Measurement of $\hat{C}(\omega)$ at a single temporal frequency therefore gives, at best, a spatial average of the complex permittivity distribution. Thus, there can be no unique inference of the spatial distribution from the temporal frequency response alone. Using an interdigital electrode structure, one that utilizes the fringing fields between electrodes, there are certain conditions under which a definitive spatial distribution can be estimated. This has been discussed in Section 3.2.2. However, in general, these conditions will not be satisfied and an alternative approach must be taken.

It is helpful to re–evaluate $\hat{C}(\omega)$ when probing with the fringing field of an interdigital electrode structure. Now there will be fields above and below the electrodes. The fields below are readily calculable because $\hat{C}^b$ is known. Thus, we are concerned with the calculation of $\hat{C}^a$, defined in (2.9). Using the field representation of Chapter 1, in a more general form,

$$\Phi(x,y,t) = Re\hat{\Phi}(x,y)e^{j\omega t} \tag{4.21}$$

where

$$\hat{\Phi}(x,y) = \sum_{n=-\infty}^{\infty} \hat{\Phi}_n(x)e^{-jk_n y}; k_n = 2n\pi/\lambda$$

the electric field has both an x and y dependence, thus (4.16) is replaced by

$$\frac{\partial(\hat{\epsilon}\hat{E}_x)}{\partial x} + \hat{\epsilon}\frac{\partial \hat{E}_y}{\partial y} = 0 \tag{4.22}$$

From (4.15) we get

$$\hat{E}_y = -\frac{\partial\hat{\Phi}}{\partial y} = j\sum_{n=-\infty}^{\infty} k_n\hat{\Phi}_n(x)e^{-jk_n y} \tag{4.23}$$

Substituting this result into (4.22) and integrating from $x = 0$, where $\hat{\epsilon}\hat{E}_x = \hat{\epsilon}_a\hat{E}_x^a$, to $x = \infty$, where $\hat{\epsilon}\hat{E}_x = 0$, yields

$$\hat{\epsilon}^a \hat{E}_x^a = -\int_0^\infty \hat{\epsilon}(x) \sum_{n=-\infty}^{\infty} k_n^2 \hat{\Phi}_n(x) e^{-jk_n y}\, dx \tag{4.24}$$

and using the definition of the surface capacitance density, (2.9)

$$\hat{C}_n^a = \frac{\hat{\epsilon}^a \hat{E}_n^a}{\hat{\Phi}_n^a} = -\frac{1}{\hat{\Phi}_n^a} \int_0^\infty \hat{\epsilon}(x) k_n^2 \hat{\Phi}_n(x)\, dx \tag{4.25}$$

In general, the potential $\hat{\Phi}_n(x)$ will decay exponentially with x, and the higher spatial harmonics (larger n), decaying even more rapidly. The spatial distribution in complex permittivity $\hat{\epsilon}(x)$ can be viewed as a weighting function for the potential distribution. Thus, unlike (4.20), the effect of the specific distribution in complex permittivity will, in general be unique.

There are several advantages to probing a medium using the variable spatial wavelength approach. The first is the ability to estimate a unique spatial distribution from the measured responses. As discussed in Section 1.2.1, the electroquasistatic fields generated by interdigital electrode structures decay exponentially in the direction normal to the electrode plane. Furthermore, the characteristic decay length is proportional to the spatial wavelength, roughly ($\lambda/3$). The response to a short wavelength will reflect the dielectric properties of the material in the immediate vicinity of the electrodes. The longer wavelength electrodes will sample both the material adjacent to the electrodes and material further out. As the wavelength is further increased, the dielectric properties at an increasing range from the sensing surface will influence the measurement. Unlike the temporal frequency response, the measurements here uniquely reflect the spatial distribution of dielectric properties.

Another advantage is the potential for simplification of the estimation problem. One could take the n measurements at n different spatial wavelengths and estimate m parameters of a function describing the spatial distribution all at once ($n \geq m$). This would entail computing and inverting an nxm jacobian (see (4.8)) for updating each guess. This estimation process is likely to involve a large amount of computer time for its implementation. However, by exploiting the exponentially decaying nature of the applied fields, the estimation procedure can be reduced to a sequence of single

parameter (or two parameter) estimation. This will be shown in Section 4.3.3.

The third advantage of this approach lies in the fact that variable spatial wavelength measurements can be taken at the same temporal frequency. Thus the spatial distribution can be deduced at each frequency without any restrictions or assumptions about the frequency dependence of the dielectric properties. This additional freedom allows an unconstrained evaluation of the possible physical mechanisms underlying the dielectric properties (such as molecular or ionic dynamics).

4.3.2 Apparatus

One straightforward implementation of this approach is to fabricate several interdigital electrode structures, each having a different spatial wavelength. If the response must be measured with all the electrodes contacting the material simultaneously, then there is a requirement that the spatial distribution within the material be the same at each separate location probed by the individual electrode structures. This constraint can be relaxed somewhat if the measurements can be made consecutively, thereby allowing each array to make contact with the material at roughly the same location.

It would be more convenient if just one set of electrodes could be used and the spatial wavelength of this array could be varied through changes in the electrical terminations of the electrodes. Such an array is shown in Fig. 4.8. The electrodes may be connected to either a driven voltage input or a floating gate output or may be totally disconnected. The shortest spatial wavelength can be imposed by connecting every other electrode together, one set driven and one floating. The next longer wavelength may be obtained by connecting pairs of electrodes together, again half the pairs driven and half floating. Beyond this, some of the electrodes between driven and floating electrodes must remain disconnected. They will assume a potential somewhere between the driven and floating voltage, as dictated by the constraint that they carry no net current. These electrodes are used to prevent the higher spatial harmonics (arising from the coupling between the edges of adjacent electrodes) from dominating the response. An array of switched electrodes based on this concept has been constructed for the purpose of measuring the thickness of a layer composed of ice and seawater

[81]. In this case, the measurement is of the impedance between the driven and floating electrodes, obtained by connecting the floating electrode to a virtual ground and measuring the current flow.

4.3.3 Estimation Algorithm

Typical distributions of complex permittivity are shown in Fig. 4.9 as a function of the normal coordinate (x) to the surface of the electrode array. In Fig. 4.9a the distribution is smooth. Those shown in Figs. 4.9b and c showcase the situation where there is a discrete discontinuity in the medium. The layer of thickness d is typical of those where there is a step discontinuity at the interface between layers. In Fig. 4.9c there is the addition of a complex surface permittivity associated with the interface. An example of this is a layer of material having a thickness d with adsorbed material on its interface giving rise to conduction in the plane of the interface (the dimensions of which are smaller than can be resolved by the measurements).

Representations of the complex permittivity distribution that can be used are shown in Fig. 4.10. These are general enough to represent any of the cases shown in Fig. 4.9. In Fig. 4.10a the distribution is represented by layers, each having a uniform complex permittivity. In Fig. 4.10b each layer has an exponential distribution. Included in each representation is the possibility of having complex surface permittivities at one or more of the interfaces. These are the impulse functions in Fig. 4.10. Without these complex surface permittivities, the representations of Figs. 4.10a and b approximate the distribution by stair–step and piece–wise continuous functions, respectively. Other functions can be used to represent the spatial distribution of properties, resulting in estimations of different parameters relevant to the individual function chosen.

The following example demonstrates the simplification in the estimation problem available with this approach. Suppose that the spatial distribution is smooth and the representation chosen is a stair–step one. Using the response at the shortest wavelength and a uniform medium model, an estimate is made of the complex permittivity

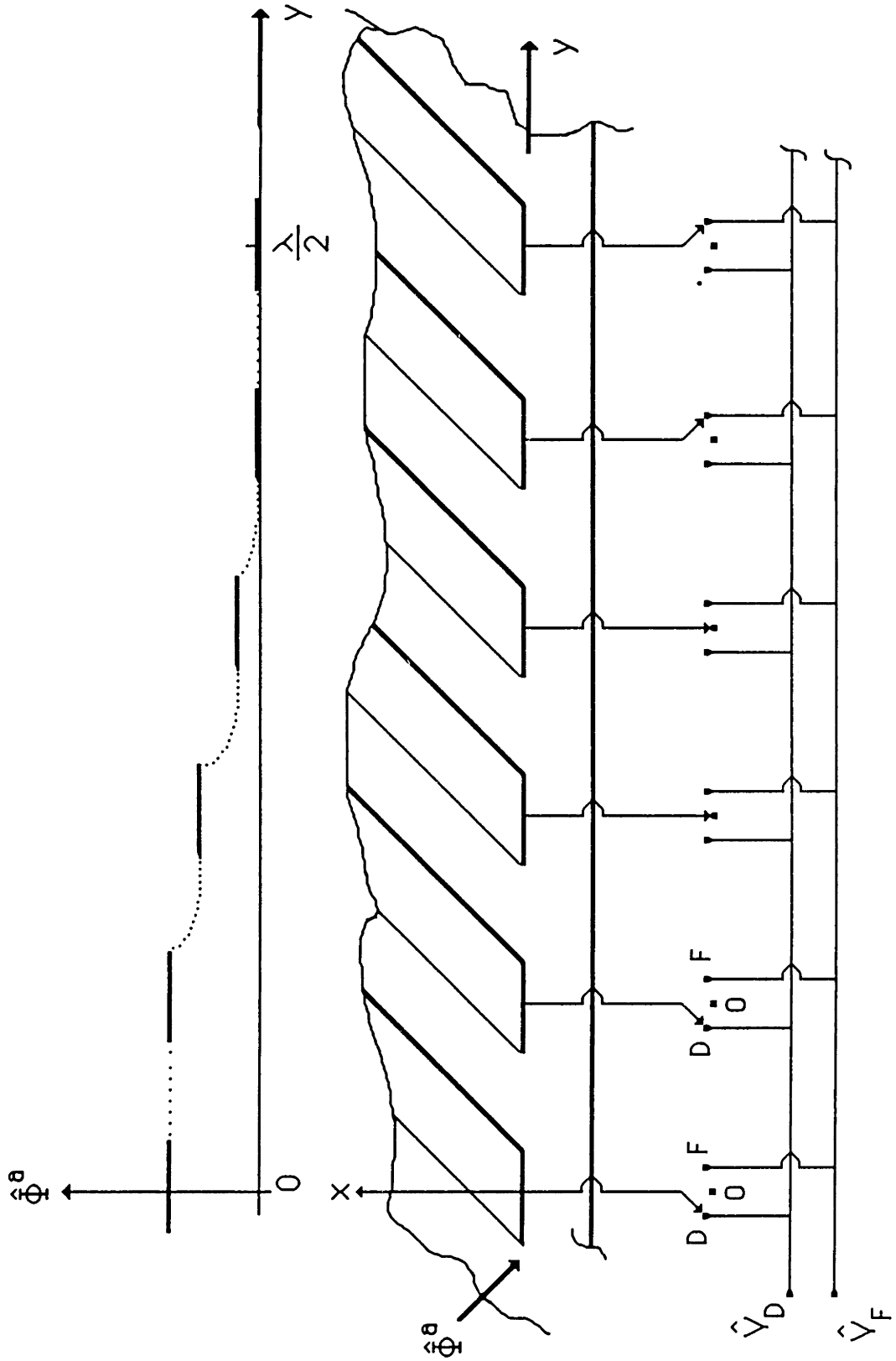


Figure 4.8: Apparatus to Provide Variable Spatial Wavelength in Potential Using a Set of Switched Electrodes.

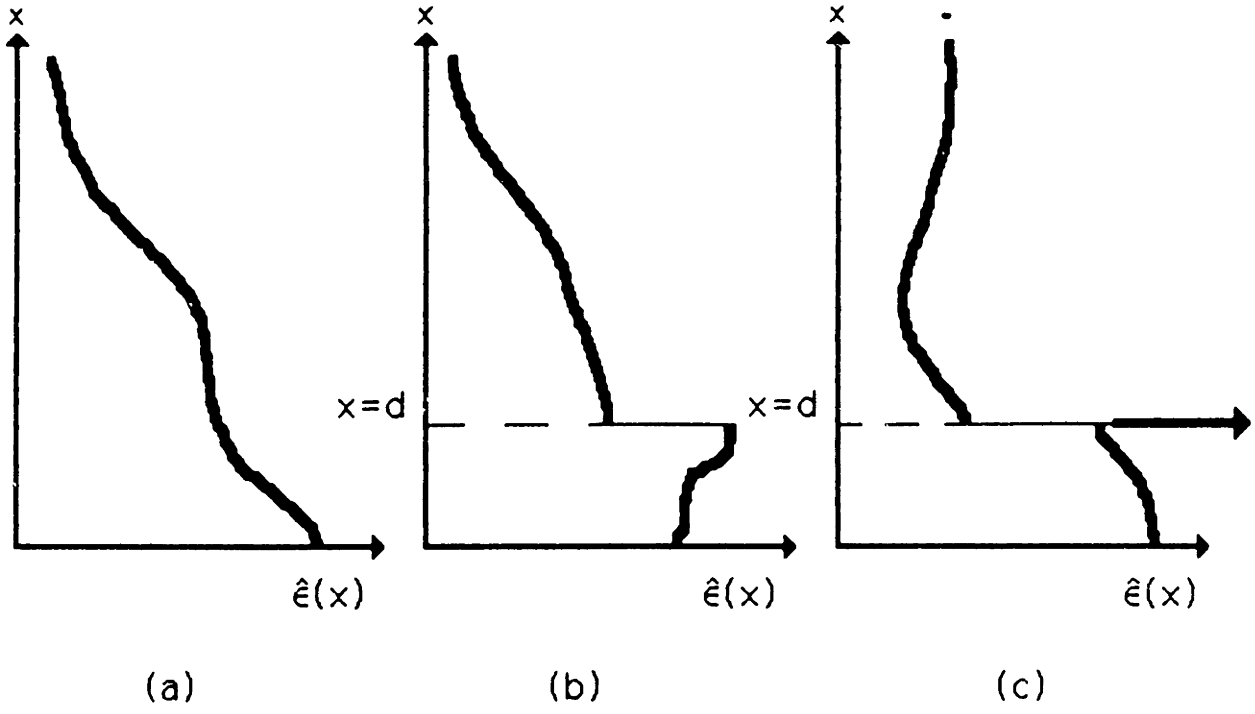


Figure 4.9: Typical Spatial Distributions in Complex Bulk Permittivity: a) Smoothly Varying, b) with Step Discontinuity, and c) with Complex Surface Permittivity.

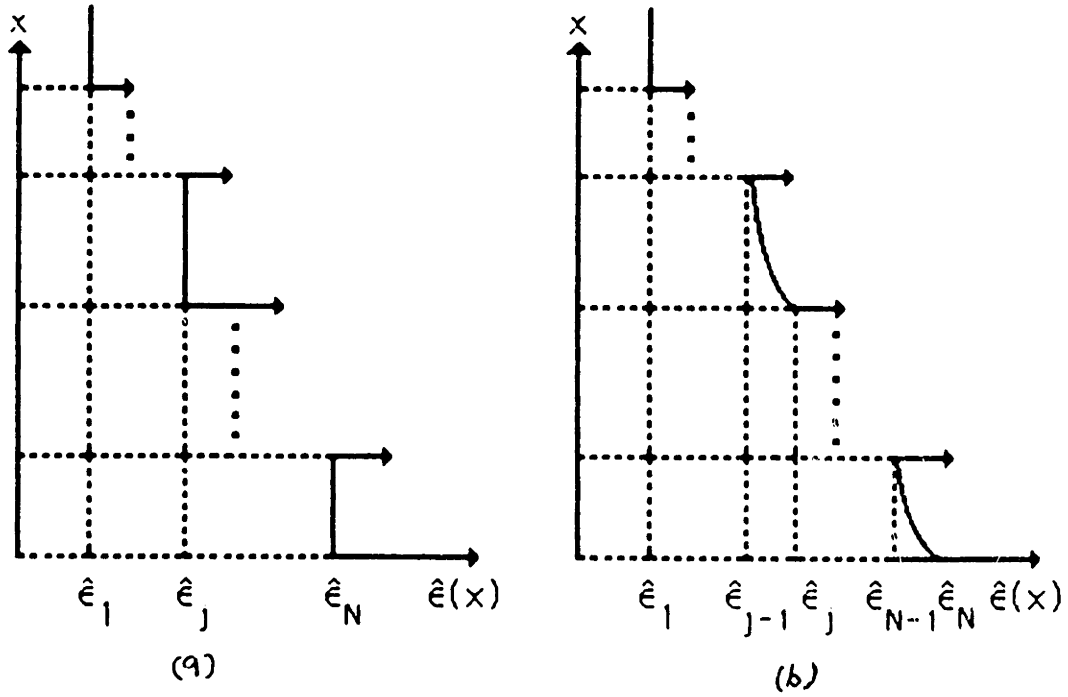


Figure 4.10: Representations that can be Used in Variable Spatial Wavelength Estimation Routine: a) Stair-Step and b) Exponential.

of the material adjacent to the electrodes. This is represented in the upper left portion of Fig. 4.11. Then, using the response at the next longer wavelength and a two region model (layer plus uniform medium), an estimate is made of the complex permittivity of the material a little further out (middle left in Fig. 4.11). Typically, a measurement at spatial wavelength λ_i will reflect the material properties out to about $(\lambda_i/4)$, determining the thickness of the previously estimated layer. Following this procedure, successive estimations are performed using previously estimated parameters and layer thicknesses determined by the spatial wavelengths associated with previous data. Once the first pass has been made through the data, working from short to long wavelengths (lower left in Fig. 4.11), the parameters can be refined by passing through the data again. In these subsequent iterations, one method of refinement would be to set the parameters of layers other than the one being estimated to their previously estimated values (upper right in Fig. 4.11). Other refinement schemes, involving smoothing the initial parameter estimates, may be more appropriate for certain cases, such as when the distribution is known to be smoothly varying.

At each spatial wavelength, the sensing depth into the material is a fraction of the wavelength, $\alpha\lambda_i$. With the stair–step or exponential representations of Fig. 4.10, the thickness of each layer d_i is determined by two spatial wavelengths, λ_i and λ_{i-1}. Proceeding from the shortest $(i = 1)$ to the longest spatial wavelength $(i = N)$, the accompanying layer thickness is

$$d_i = \begin{cases} \infty & \text{for } i = 1 \\ \alpha(\lambda_i - \lambda_{i-1}) & \text{for } i = 2, 3, \ldots, N-1 \\ \alpha\lambda_N & \text{for } i = N \end{cases} \tag{4.26}$$

For the cases considered here $\alpha = .25$, allowing some overlap between spatial wavelengths. In general, the oxide layer thickness and permittivity will affect, to some degree, the penetration depth of the fields above the electrodes. However, when using $\epsilon_{ox} = 3.9\epsilon_o$ and varying λ/h from 2 to 100, this effect is negligible.

If α was much larger, the spatial wavelengths would be completely uncoupled. Parameter estimates due to a particular wavelength would have no effect on the gain–

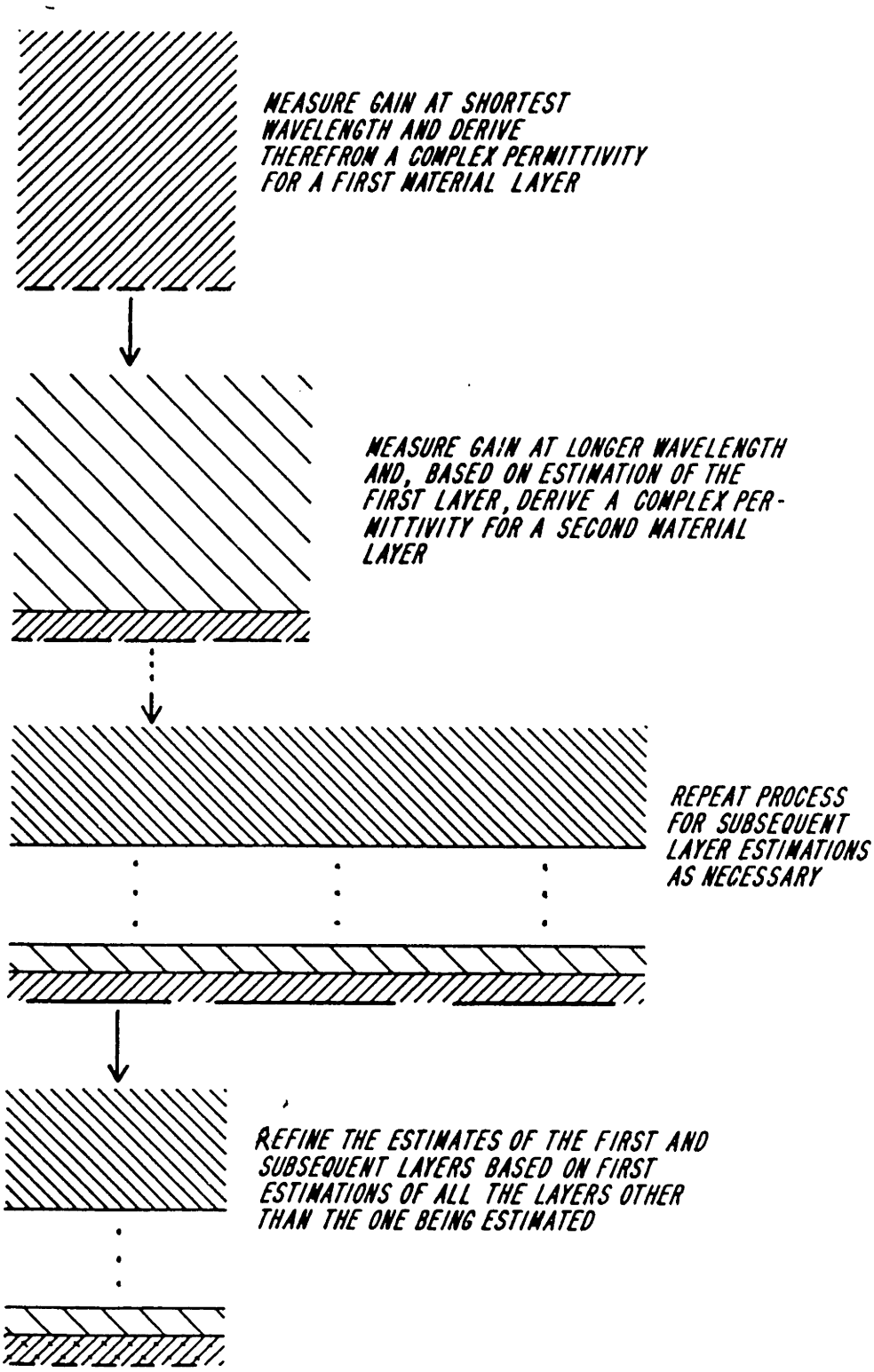


Figure 4.11: Schematic View of Variable Spatial Wavelength Estimation Algorithm Using Stair-Step Representation.

phase response as measured by any of the shorter spatial wavelengths. No refinement of the estimates could be accomplished by subsequent passes. If this length was much smaller, the coupling between the spatial wavelengths would be too intimate. Any slight change in a parameter would ripple through much of the estimation, greatly increasing the time to convergence. The value chosen, ($\alpha = .25$), represents a compromise given the above considerations and also is physically based, as discussed in the intimate sensing section (Section 1.2.2).

The following three sections describe implementations of the estimation algorithm for examples involving smoothly varying distributions in complex permittivity and step discontinuities. Conduction in the materials is assumed to be ohmic, thus the complex permittivity is specified in terms of the permittivity and conductivity. Experimental data at a constant frequency is simulated using the continuum model. For these simulations, a layer in which there is a smoothly varying distribution is approximated by breaking it into 90 discrete sublayers having uniform properties. The gain–phase response is computed for several different spatial wavelengths at the same temporal frequency, here it is 0.1 Hz. A stair–step representation (Fig 4.10a) is used for all the examples. The sequence of single parameter estimations is performed using the estimation routine described in Section 4.2.2. This includes the technique of using variable grid points to rapidly find the neighborhood of the root. Convergence is defined using the same tolerances on gain (.05 dB) and phase (.1°) as for previous estimations using the microchip, unless otherwise denoted. The software for this estimation routine is called **hetest.for** and may be found in Appendix F.

For these case studies, an apparatus such as the one shown in Fig. 4.8 is envisioned. Therefore the oxide layer thickness is the same for all spatial wavelengths and is chosen to be 10 μm, having a permittivity of $3.45x10^{-11}$ F/m. The wavelengths used, and corresponding normalized load capacitances, are compiled in Table 4.1. The normalized load capacitance is adjusted so as to produce the same dynamic range for all spatial wavelengths when operated in air. This capacitance is a strong function of the oxide layer thickness. Here, with the stated thickness of 10 μm, the dynamic range chosen was 0 to –60 dB. Although this may be an unreasonable dynamic range in practice, it

Table 4.1: Spatial Wavelength and Normalized Load Capacitance.

$h = 10\ \mu$m, $\epsilon_{ox} = 3.45x10^{-11}$ F/m, $a = .25$

λ (μm)	$(C_\ell/\epsilon_{ox}M_{el})$
20	768.78
50	370.87
100	165.84
200	90.50
300	72.34
400	63.11
500	56.73
600	51.48
700	47.39
800	43.45
900	39.86
1000	36.62

was felt to be important that each spatial wavelength operate with the same dynamic range. Any smaller range would have required negative load capacitances at the longer spatial wavelengths.

4.3.4 Smooth Distribution of Complex Permittivity

Case 1 - The first case considered is a medium whose complex permittivity varies exponentially away from the electrodes. The permittivity decreases exponentially from $3x10^{-11}$ F/m at the electrodes to $2x10^{-11}$ F/m at a distance of 50 μm, at which point the permittivity remains constant. However, the conductivity increases exponentially from 10^{-12} S/m at the electrodes to 10^{-10} S/m at a distance of 100 μm , also remaining constant beyond this distance. This distribution is represented by the thick line in

Table 4.2: Simulated Data for Case 1.

λ (μm)	gain (dB)	phase (deg.)
20	–55.18	-2.13
100	–49.24	-10.20
200	–45.68	-25.20
400	–41.10	-39.04
800	–37.13	-51.96
1000	–36.18	-56.47

Fig. 4.12. The simulated data, tabulated in Table 4.2, is generated using six different spatial wavelengths using the oxide layer thickness and normalized load capacitances mentioned above and a temporal frequency of 0.1 Hz. Note that the spatial wavelengths vary from 20 to 1000 μm, effectively probing out to a distance of roughly 250 μm.

Also shown in Fig. 4.12 are the resulting estimates of 10 "marches", or passes, through the data. Using the stair–step representation, at each spatial wavelength there is an associated layer thickness having its own estimated complex bulk permittivity. The estimated distribution after the first pass through the data is denoted by the triangles, and the final estimate, after the tenth pass, by the crosses. The intervening iterations are denoted numerically, where possible, to indicate convergence or divergence. Convergence to a reasonable stair–step approximation of the exponential distribution is observed. The nomenclature established here regarding the thick line and symbols is used for all the case studies in this section.

Case 2 – The second case considered also utilizes an exponential distribution in complex permittivity over a distance of 50 μm. The permittivity varies the same as before but the conductivity varies oppositely, decreasing exponentially from 10^{-10} S/m at the electrodes to 10^{-12} S/m at a distance of 50 μm and constant thereafter. The thick line in Fig. 4.13 again represents the actual distribution and the simulated data (at 0.1 Hz) for this case can also be found in Table 4.3.

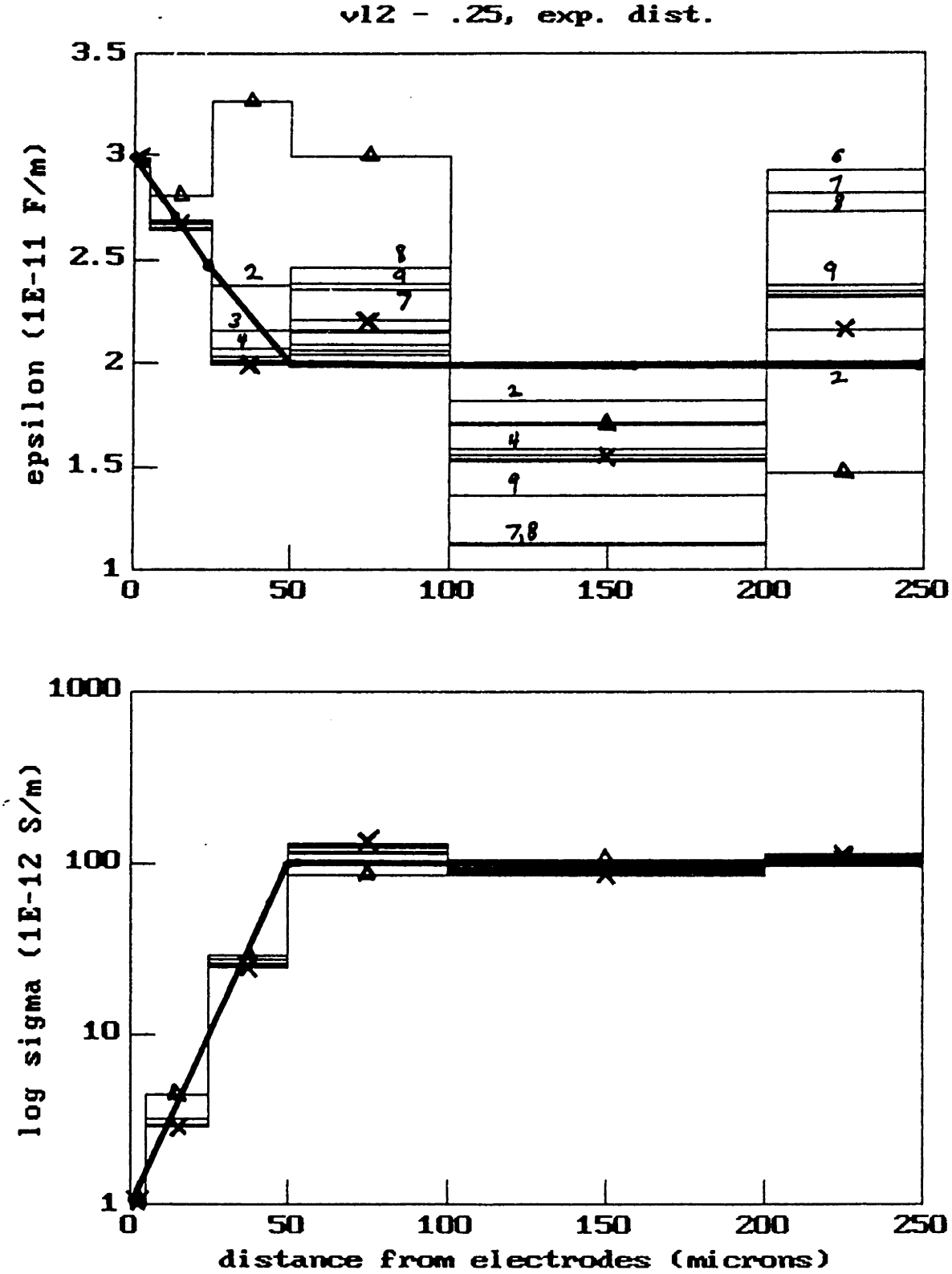


Figure 4.12: Estimated Spatial Distribution Using 6 Spatial Wavelengths (Case 1). Simulated Data in Table 4.2. (△) – first estimation, (X) – final estimation.

Using the same 6 spatial wavelengths as in the previous example, the estimation procedure ran into problems with stability. During the first pass (triangles in Fig. 4.13), there was a problem estimating the conductivity for the fourth layer (extending 50–100 μm). The result, indicated as $1x10^{-14}$ S/m, is actually a default value used for plotting the data, as the estimate actually wanted to go to negative conductivities and was set to zero. After the final pass (crosses), this problem shows up on the other side of the discontinuity. Furthermore, it is observed that the permittivity estimates are diverging from the actual values and from the estimates obtained on the first pass. This same result is obtained if the same spatial distribution is probed using all 12 spatial wavelengths for finer resolution (Fig. 4.14). Here, the first pass produces a reasonable permittivity profile (triangles) but again has a problem with the conductivity following the discontinuity. After the tenth pass, the permittivity and conductivity estimates are diverging, and the conductivity estimates at 800 and 900 μm have been set to zero.

It appears that the iteration process is unstable. Even the result of Fig. 4.12 appears to begin diverging slightly at the end of the 10 passes, it's just that the first pass produced such a poor estimated profile that the following iterations had to begin converging first, before diverging. Altering the tolerance limits for convergence, or the parameter determining the layer thickness (α), did not improve convergence. Moving the discontinuity out to 500 μm while retaining the exponential distribution, did not eliminate the divergence problem. The simulated data for this estimation problem is also tabulated in Table 4.3. The results using 6 and 12 spatial wavelengths are shown in Figs. 4.15 and 4.16 respectively. In both cases, the best estimate is provided by the first pass through (triangles) although the only significant divergence using 6 wavelengths was for the last layer ($>$ 200 μm). Probing the medium with 4 or 8 spatial wavelengths did not improve the situation.

Finally, the case of a uniform medium having no spatial distribution was examined. The permittivity of the medium was $2x10^{-11}$ F/m and the conductivity was 10^{-10} S/m. The simulated data, generated at both 0.1 and 1 Hz, is tabulated in Table 4.4. Again the first pass is close to the actual distribution with subsequent iterations resulting in a diverging solution (Figs. 4.17 and 4.18). This divergence occurs for the search at both

Table 4.3: Simulated Data for Case 2.

	$d_2 = 50\ \mu$m		$d_2 = 500\ \mu$m	
λ (μm)	gain (dB)	phase (deg.)	gain (dB)	phase (deg.)
20	-45.30	-71.25	-44.42	-73.03
50	-41.04	-75.54	-38.59	-79.17
100	-37.53	-80.66	-32.51	-85.49
200	-39.77	-87.56	-29.46	-94.46
300	-45.46	-81.38	-29.99	-100.93
400	-49.65	-57.92	-31.41	-105.07
500	-50.64	-36.99	-33.13	-107.15
600	-50.69	-27.00	-34.94	-107.38
700	-50.83	-22.23	-36.81	-105.98
800	-51.02	-19.12	-38.59	-103.16
900	-51.21	-17.05	-40.27	-99.10
1000	-51.40	-15.41	-41.81	-94.04

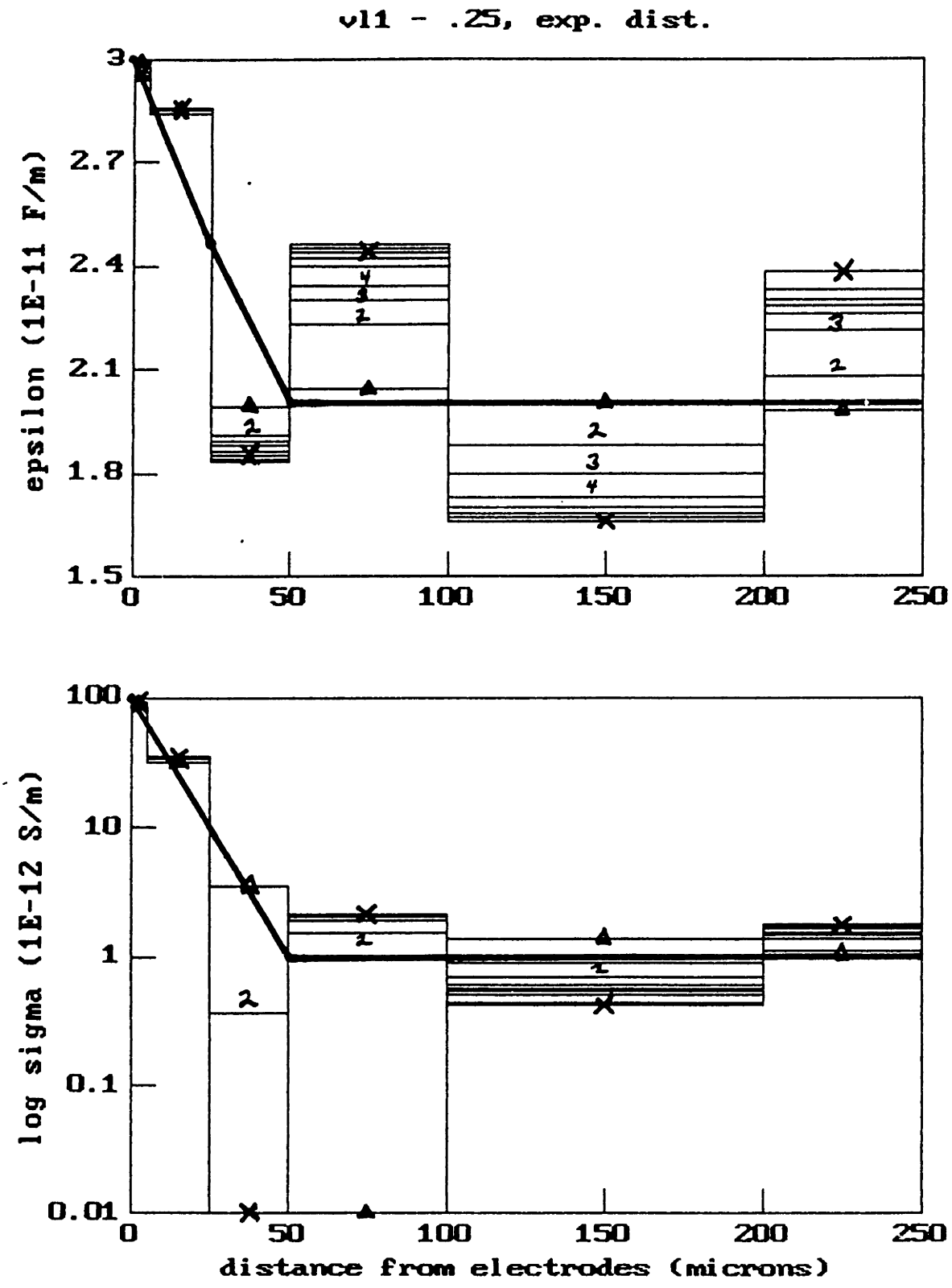


Figure 4.13: Estimated Spatial Distribution Using 6 Spatial Wavelengths (Case 2). Simulated Data in Table 4.3, $d_2 = 50$ μm. ($\triangle$) – first estimation, (X) – final estimation.

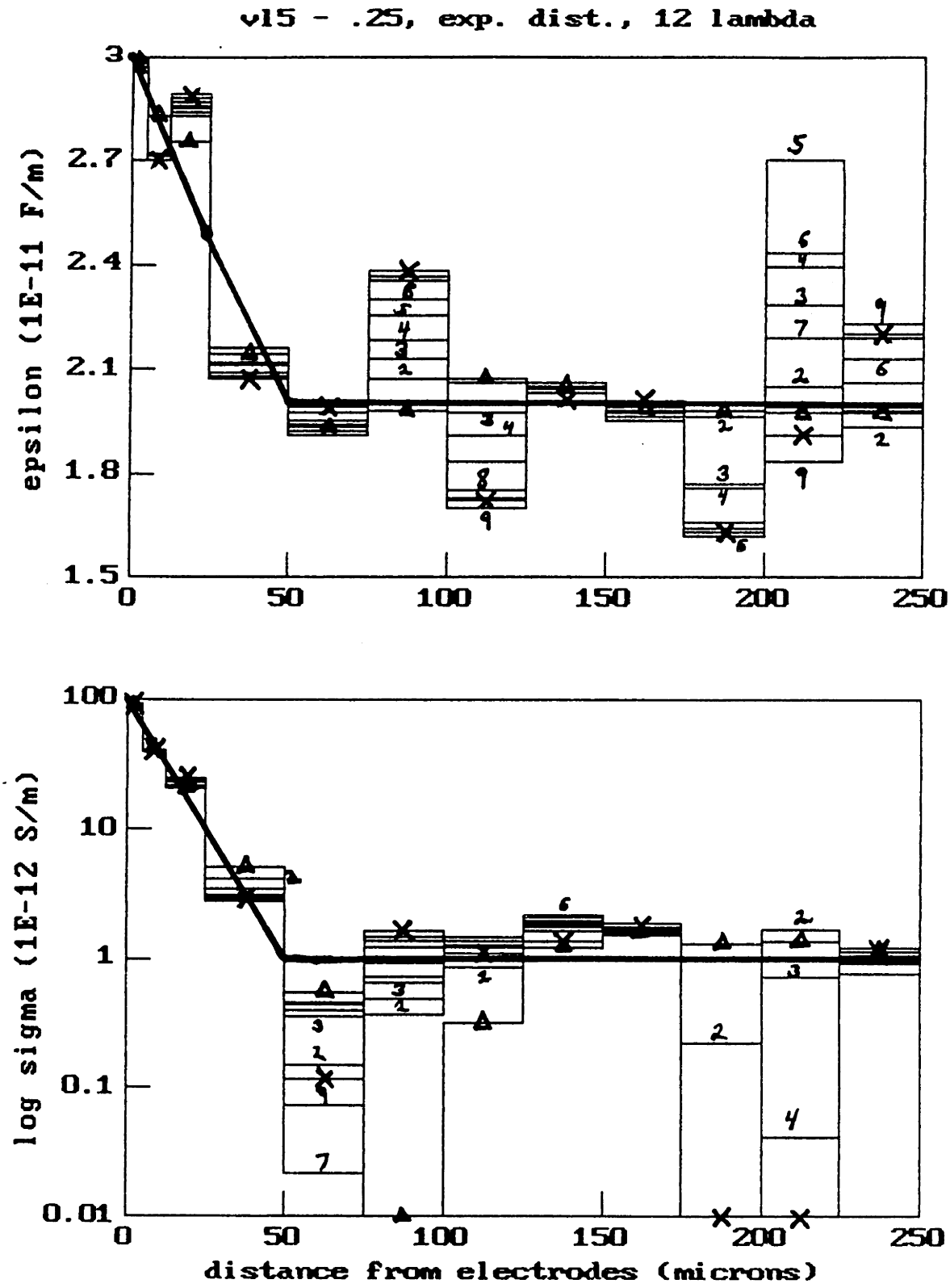


Figure 4.14: Estimated Spatial Distribution Using 12 Spatial Wavelengths (Case 2). Simulated Data in Table 4.3, $d_2 = 50$ μm. ($\triangle$) – first estimation, (X) – final estimation.

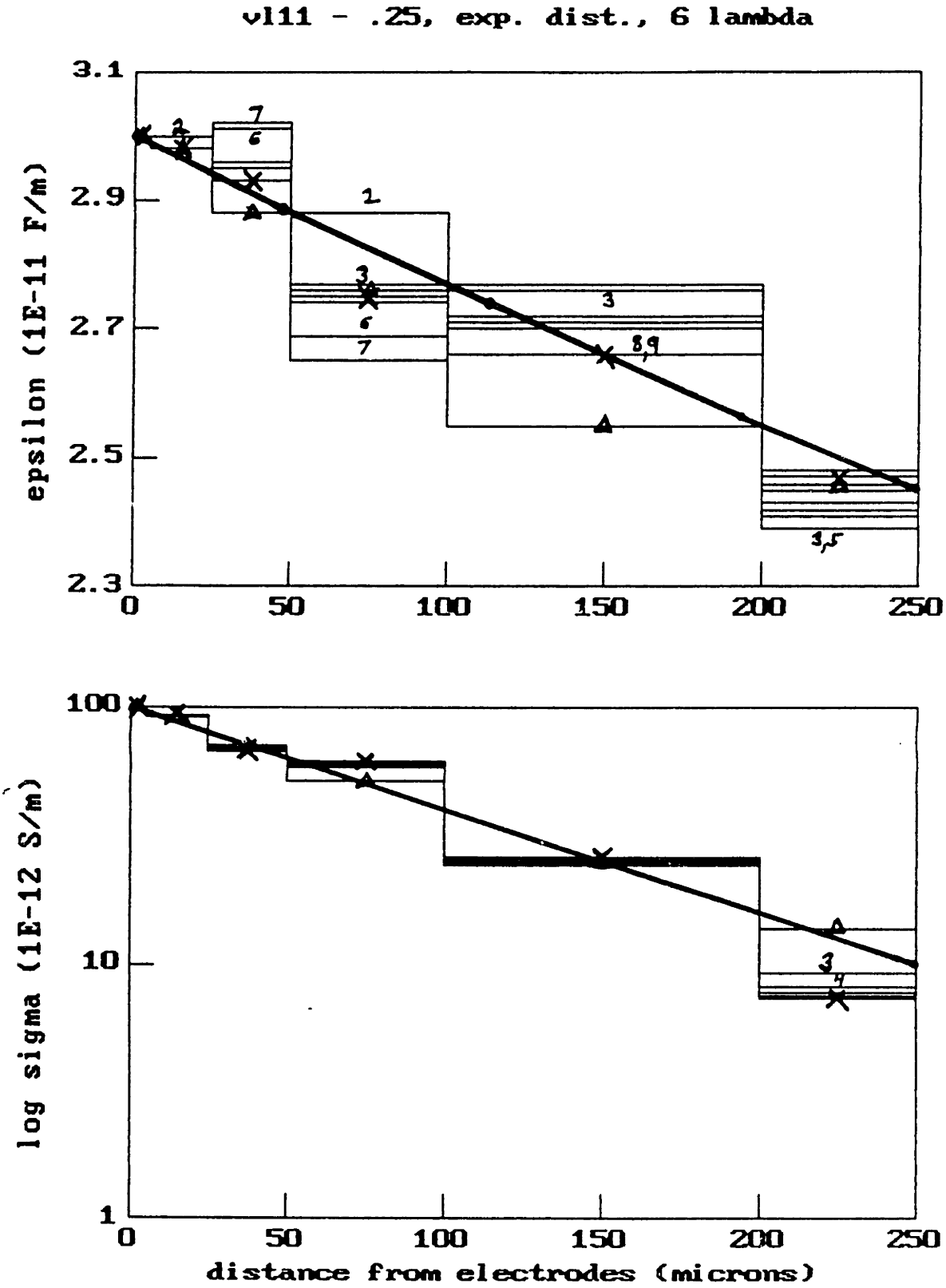


Figure 4.15: Estimated Spatial Distribution Using 6 Spatial Wavelengths (Case 2). Simulated Data in Table 4.3, $d_2 = 500$ μm. ($\triangle$) - first estimation, (X) - final estimation.

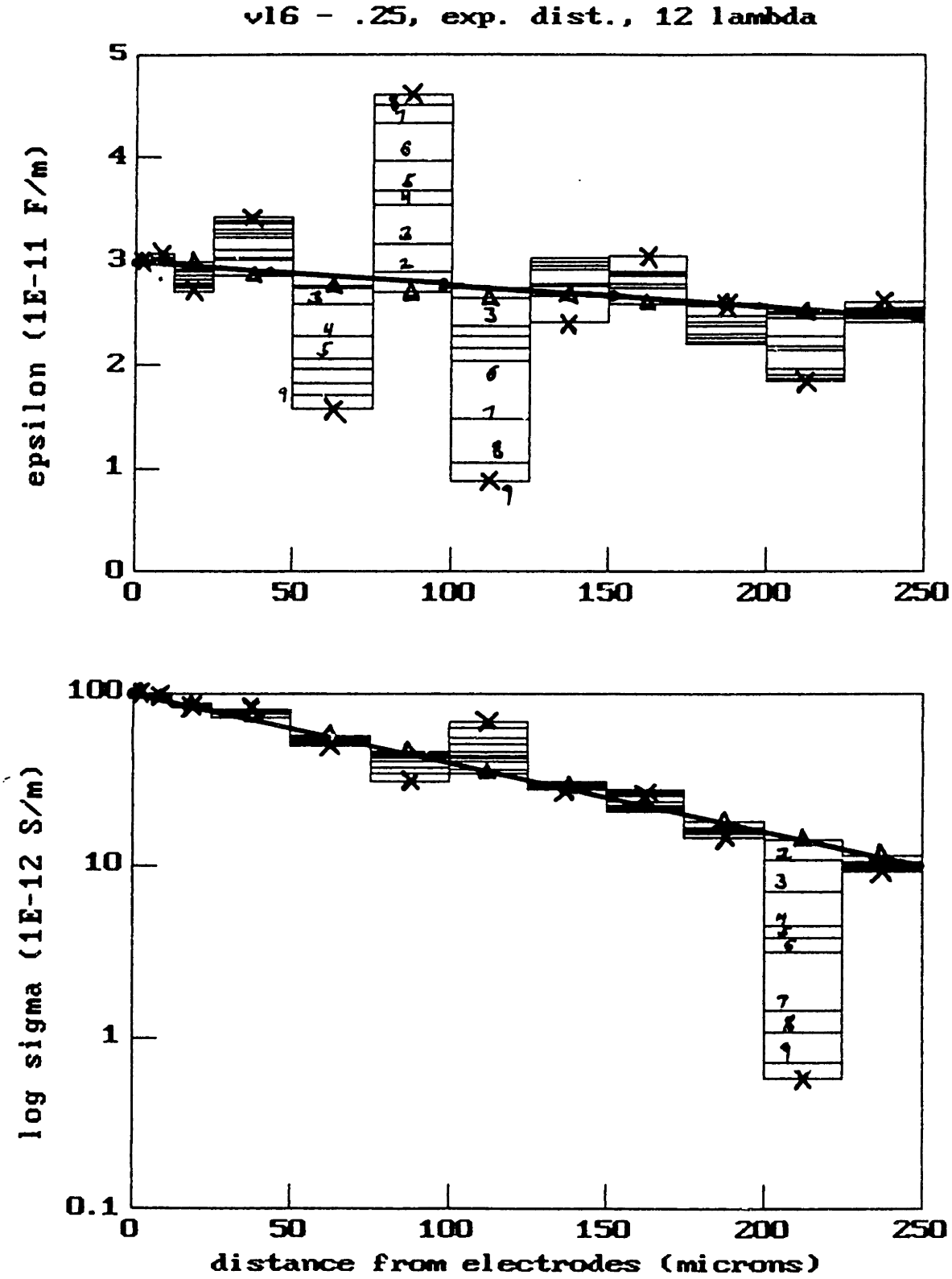


Figure 4.16: Estimated Spatial Distribution Using 12 Spatial Wavelengths (Case 2). Simulated Data in Table 4.3, $d_2 = 500$ μm. ($\triangle$) - first estimation, (X) - final estimation.

Table 4.4: Simulated Data for Uniform Medium.

λ (μm)	$f = 0.1$ Hz gain (dB)	$f = 0.1$ Hz phase (deg.)	$f = 1$ Hz gain (dB)	$f = 1$ Hz phase (deg.)
20	-44.54	-76.42	-56.46	-22.16
100	-32.11	-89.50	-50.74	-42.31
200	-28.28	-98.35	-49.42	-46.99
400	-28.33	-108.03	-49.73	-45.80
800	-30.73	-109.78	-50.12	-43.19
1000	-31.62	-107.48	-50.24	-42.47

frequencies, the difference lying in the estimated parameter that has the most significant divergence. At 0.1 Hz the permittivity estimate has the larger deviation whereas a 1 Hz it is the conductivity estimate that has the larger deviation. This behaviour indicates the importance of knowing where in the gain–phase mapping (see Figs. 4.3–4.6) the estimation is being performed. At higher frequencies, the estimation routine is less sensitive to conductivity (closer to gain axis) whereas at lower frequencies, the routine is less sensitive to permittivity (closer to origin).

One problem may be that on the subsequent marches through data, in the effort to refine the estimates, the errors from the longer spatial wavelength estimations may be unduly influencing the estimates using the shorter spatial wavelengths. For example, if the estimate ϵ_2^*, resulting from λ_2, perturbs in one direction the predicted response, as seen by the shorter wavelength λ_1, on the following iteration, then the refined estimate ϵ_1^* will try to compensate by perturbing in the other direction. Unfortunately, this can be seen as a type of positive feedback situation, resulting in an unstable process producing diverging complex permittivity profiles. This might be the cause of the divergence in the examples using a uniform medium (Figs. 4.17 and 4.18). If the refinement process is unstable, any perturbation errors resulting from the first pass may cause growing, rather than decaying, errors in the subsequent iterations.

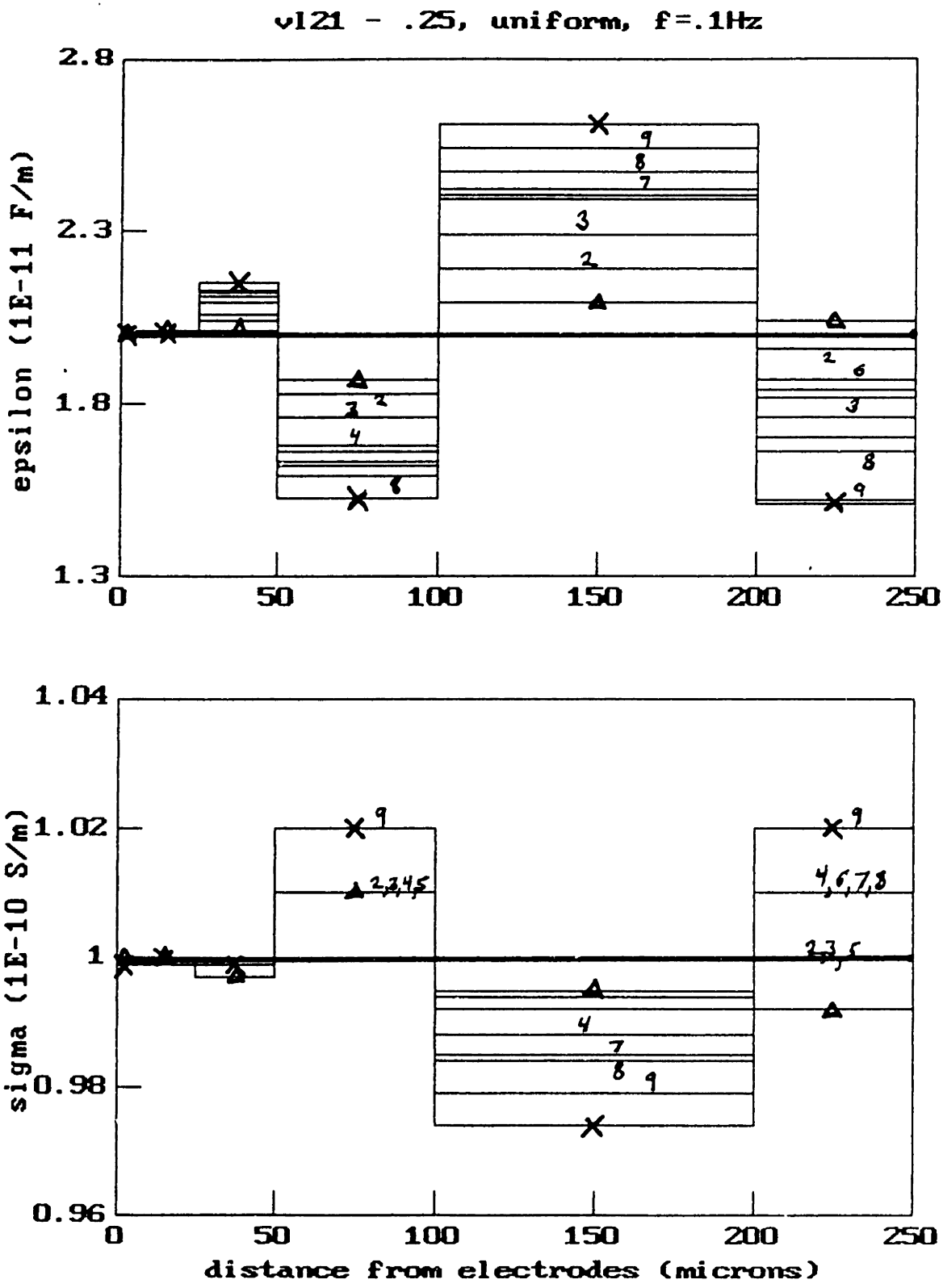


Figure 4.17: Estimated Spatial Distribution Using 6 Spatial Wavelengths (Uniform Medium). Simulated Data in Table 4.4, $f = 0.1$ Hz. ($\triangle$) - first estimation, (X) - final estimation.

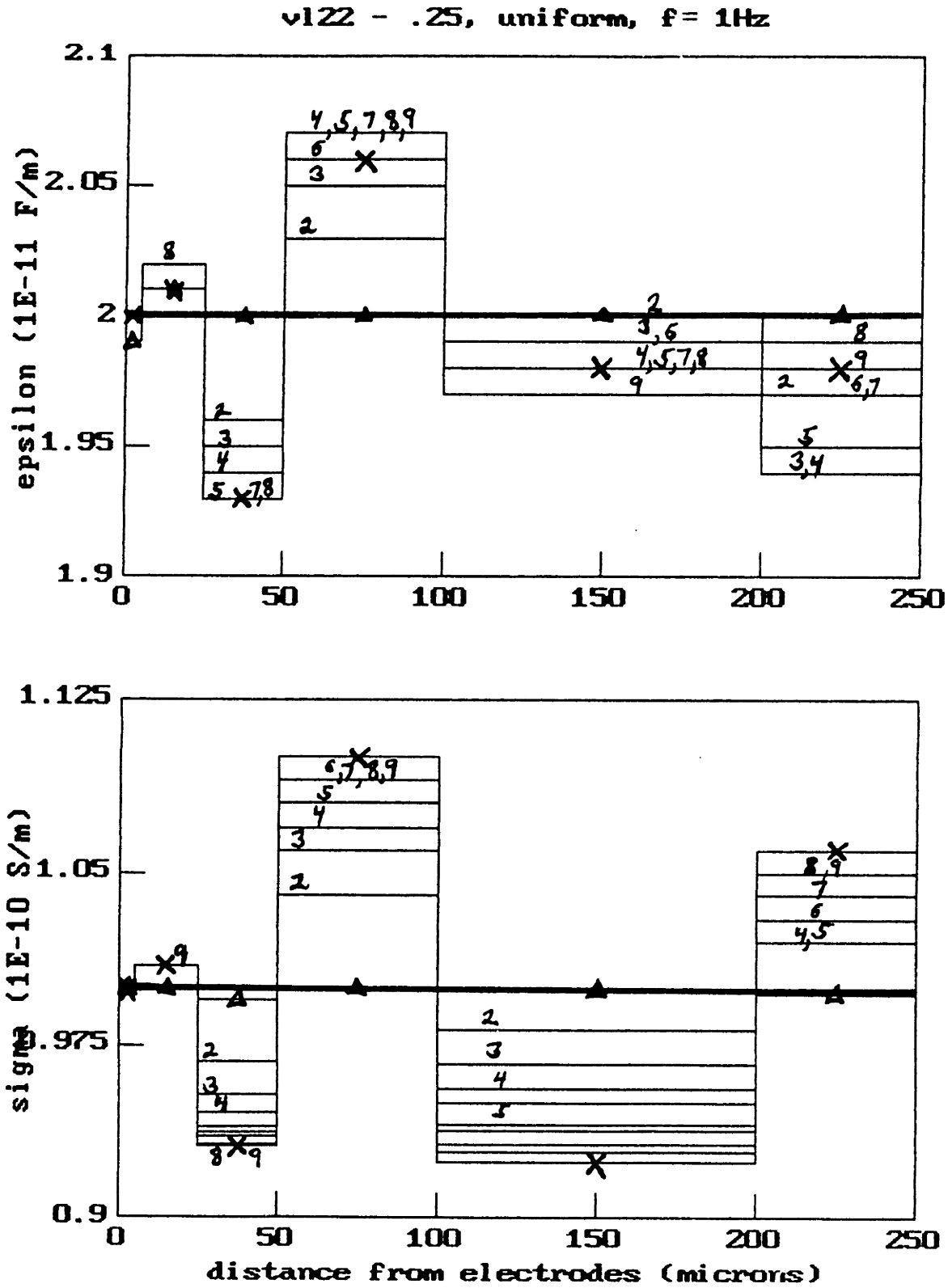


Figure 4.18: Estimated Spatial Distribution Using 6 Spatial Wavelengths (Uniform Medium). Simulated Data in Table 4.4, $f = 1$ Hz. ($\triangle$) - first estimation, (X) - final estimation.

Another problem may lie in the usage of the stair–step representation for approximating a smooth distribution. This amounts to representing distributed charges within the bulk using impulse charge distributions at the interfaces. The discontinuities introduced by such a representation may be placing an unfair demand on the estimation routine.

4.3.5 Step Discontinuities

Case 3 – In many instances, the distribution may actually consist of two different, homogeneous materials, forming a step discontinuity. A stair–step representation is particularly appropriate for the first example presented here. The permittivity of the two materials is the same, $2x10^{-11}$ F/m, but there is a step change in conductivity at a distance of 50 μm from the electrodes, decreasing from 10^{-10} to 10^{-12} S/m away from the electrodes. The simulated data used is in Table 4.5. It is seen in Fig. 4.19 that the estimation routine converges almost exactly to the discontinuity. This convergence is very dependent on the location of the discontinuity relative to the stair–step layer thicknesses, as established by the spatial wavelengths used. For example, with the discontinuity moved to 70 μm, and using the same spatial wavelengths, the estimation does a very poor job of finding the distribution (Fig. 4.20). It cannot compensate for the fact that the discontinuity occurs in the middle of one of the layers. A more natural approach to this problem, when it is known that a discontinuity exists, is to use variable wavelength data to estimate both the layer thickness and complex bulk permittivity, as shown in the following section.

4.3.6 Layer Thickness and Complex Bulk Permittivity

Case 4 – The previous example of a step discontinuity is treated in a different manner here. It is assumed at the outset that the complex bulk permittivity of the layer adjacent to the electrodes has been estimated using a very short spatial wavelength measurement, one that probes only this layer. Then, a two parameter estimation is performed to obtain the layer thickness and complex bulk permittivity of the material

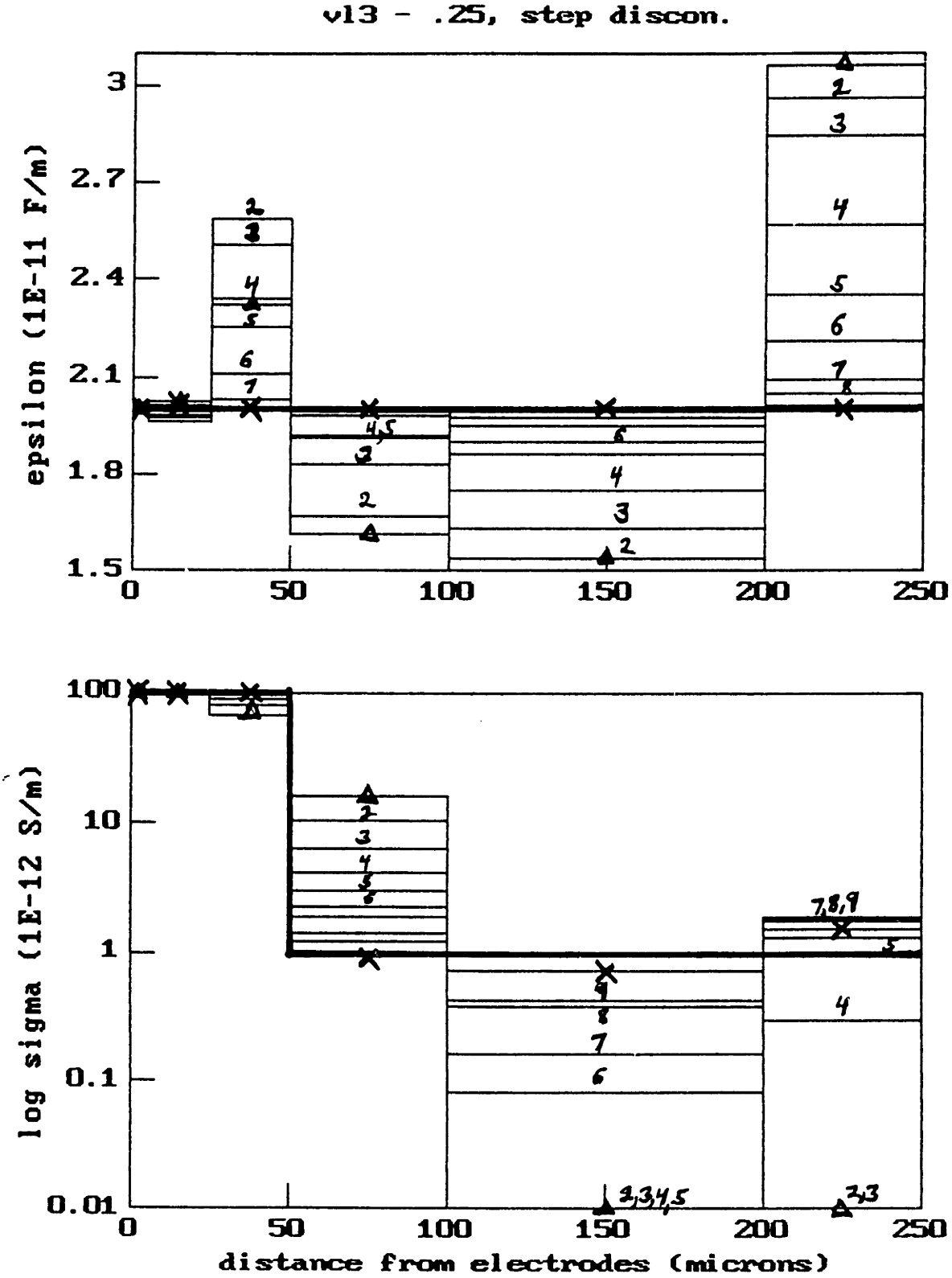


Figure 4.19: Estimated Step Discontinuity Using 6 Spatial Wavelengths (Case 3). Simulated Data in Table 4.5, $d_2 = 50$ μm. ($\triangle$) – first estimation, (X) – final estimation.

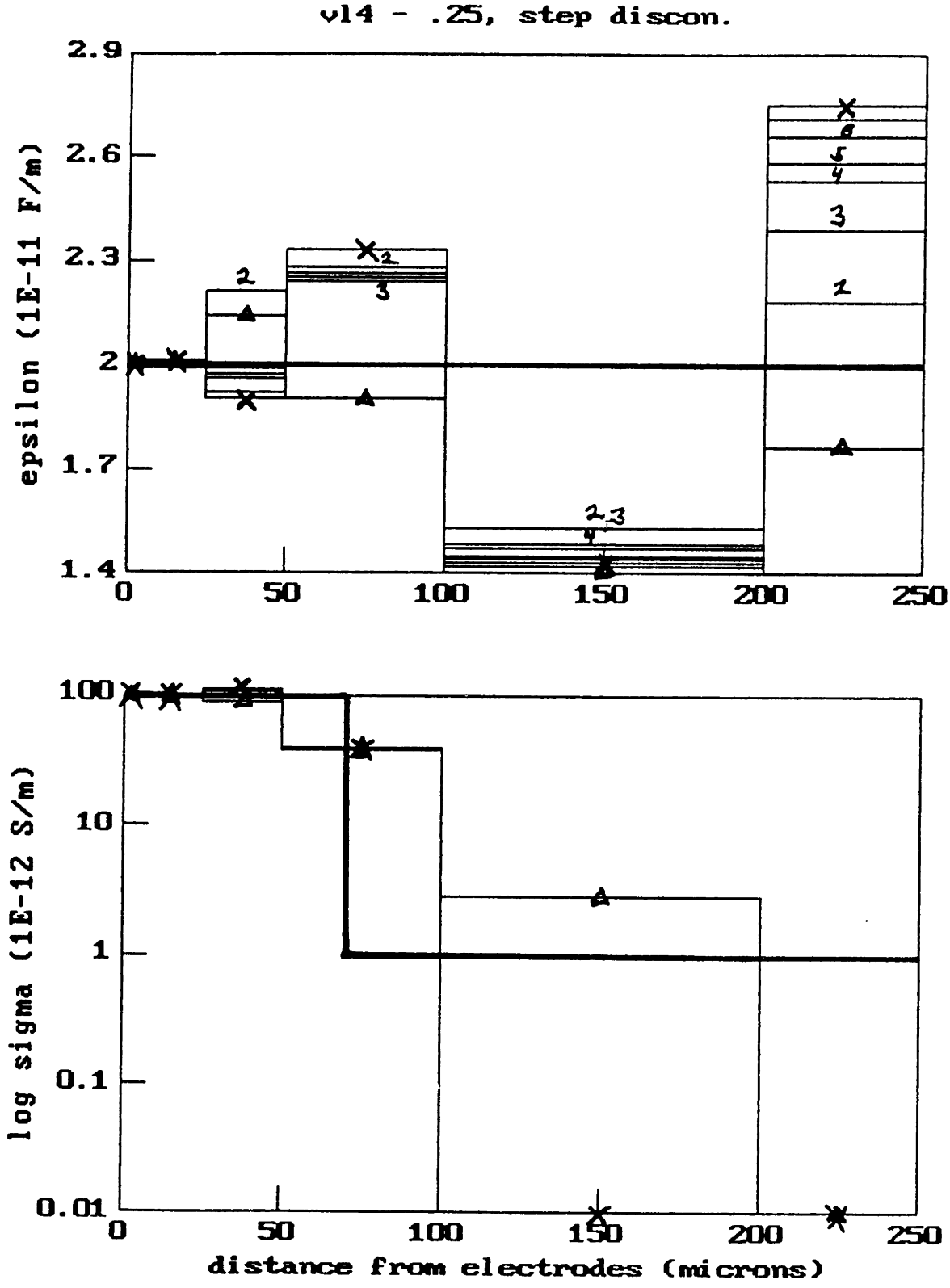


Figure 4.20: Estimated Step Discontinuity Using 6 Spatial Wavelengths (Case 3). Simulated Data in Table 4.5, $d_2 = 70$ μm. ($\triangle$) – first estimation, (X) – final estimation.

Table 4.5: Simulated Data for Case 3.

λ (μm)	$d_2 = 50$ μm gain (dB)	$d_2 = 50$ μm phase (deg.)	$d_2 = 70$ μm gain (dB)	$d_2 = 70$ μm phase (deg.)
20	-44.54	-76.42	-44.54	-76.42
100	-32.14	-89.48	-32.11	-89.50
200	-29.06	-98.79	-28.50	-98.48
400	-33.32	-116.12	-30.87	-112.24
800	-52.94	-121.35	-43.93	-128.72
1000	-56.65	-32.27	-53.70	-111.09

beyond this layer using the data from two or more longer spatial wavelengths, those capable of probing beyond the first layer. This two parameter estimation is obtained using the formalism described in Section 4.1.2. The errors, and thus the elements of the jacobian, are complex because the form of the gain used is (2.1). The complex jacobian is formed by taking numerical differences to approximate the local derivatives. Here, the optional initial guess that is programmed into the routine begins with a layer thickness of ($d_2 = \lambda_1/8$), where λ_1 is the shortest spatial wavelength used for this estimation, and an insulating, nonpolar material above this layer ($\epsilon_1 = 9.07x10^{-12}$ F/m and $\sigma_1 = 3x10^{-12}$ S/m). As is typical of Newton search routines, these programs are not very robust when the guesses are very far from the root. This is the reason for beginning the routine with a programmed guess for the layer thickness in the middle of the sensitivity range of the shortest wavelength. The update to the thickness (Δd_2) utilizes only the real part of the complex update. The software routine is **discon1.for** and may be found in Appendix F.

The first set of examples reexamines the discontinuities presented in Case 3. The complex bulk permittivity of the lower layer is known from a short spatial wavelength estimation, say a 50 μm measurement ($\epsilon_2 = 2x10^{-11}$ F/m and $\sigma_2 = 10^{-10}$ S/m). Using only the data at 400 and 1000 μm from Table 4.5, the two parameter estimation of the

Table 4.6: Simulated Data for Case 4.

λ (μm)	$d_2 = 50$ μm gain (dB)	$d_2 = 50$ μm phase (deg.)	$d_2 = 70$ μm gain (dB)	$d_2 = 70$ μm phase (deg.)
400	-50.94	-15.79	-51.41	-9.43
1000	-50.26	-32.11	-50.48	-27.67

lower layer thickness and the upper layer complex bulk permittivity was performed. Convergence to the correct profile was obtained in 6 iterations for the 50 μm discontinuity (Fig. 4.21) and in 4 iterations for the 70 μm discontinuity (Fig. 4.22). Again the triangles indicate the initial estimate and the crosses, the final estimate. The layer thickness estimates are indicated at each iteration, as well as the permittivity and conductivity estimates.

A second set of examples involves estimating the same discontinuities with the layer properties reversed, namely the lower conductivity layer is now adjacent to the electrodes. The simulated data, again using 400 and 1000 μm spatial wavelength electrodes but now at 1 Hz, is given in Table 4.6. Convergence to the correct profile was obtained in 6 iterations for the 50 μm discontinuity (Fig. 4.23) and in 5 iterations for the 70 μm discontinuity (Fig. 4.24). As with all parameter estimations, the accuracy of the parameter estimation will depend on the specific region of the gain–phase plot (of the type shown in Fig. 4.3) in which a search is being conducted. This, in turn, will depend on the temporal frequency at which the data was obtained. For example, repeating this estimation problem using data at 0.01 Hz results in a poor estimate of the conductivity of the upper layer.

4.3.7 Complex Surface and Bulk Permittivity

Case 5 – Unlike the previous case, the layer thickness is known here. Instead, there is an unknown surface conductivity on the surface of this layer in addition to the unknown bulk conductivity beyond the layer. Again, it is assumed that the complex bulk

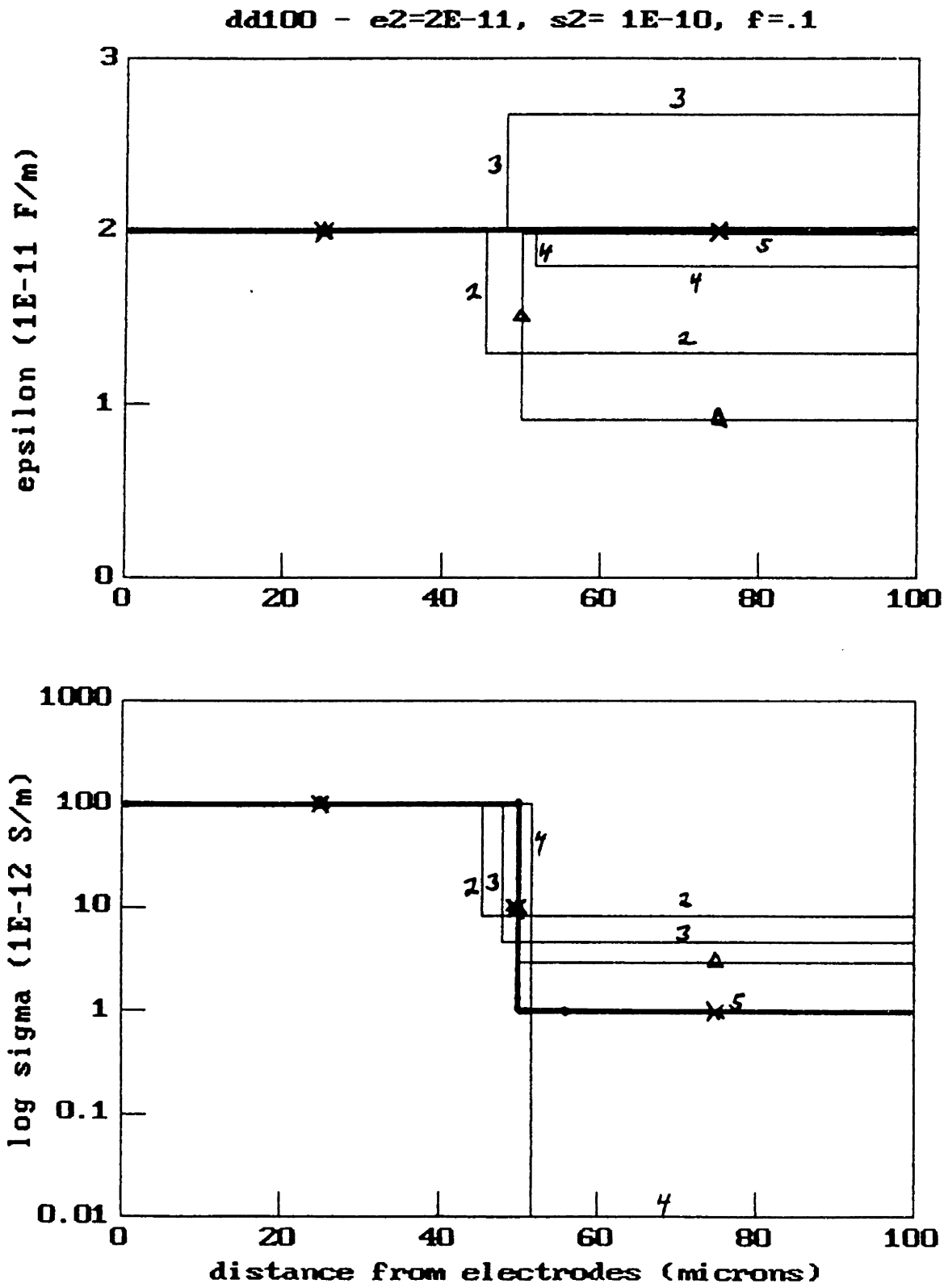


Figure 4.21: Estimated Layer Thickness and Upper Medium Complex Bulk Permittivity Using 400 and 1000 μm Spatial Wavelength Data from Table 4.5 (Case 3, $d_2 = 50$ μm).

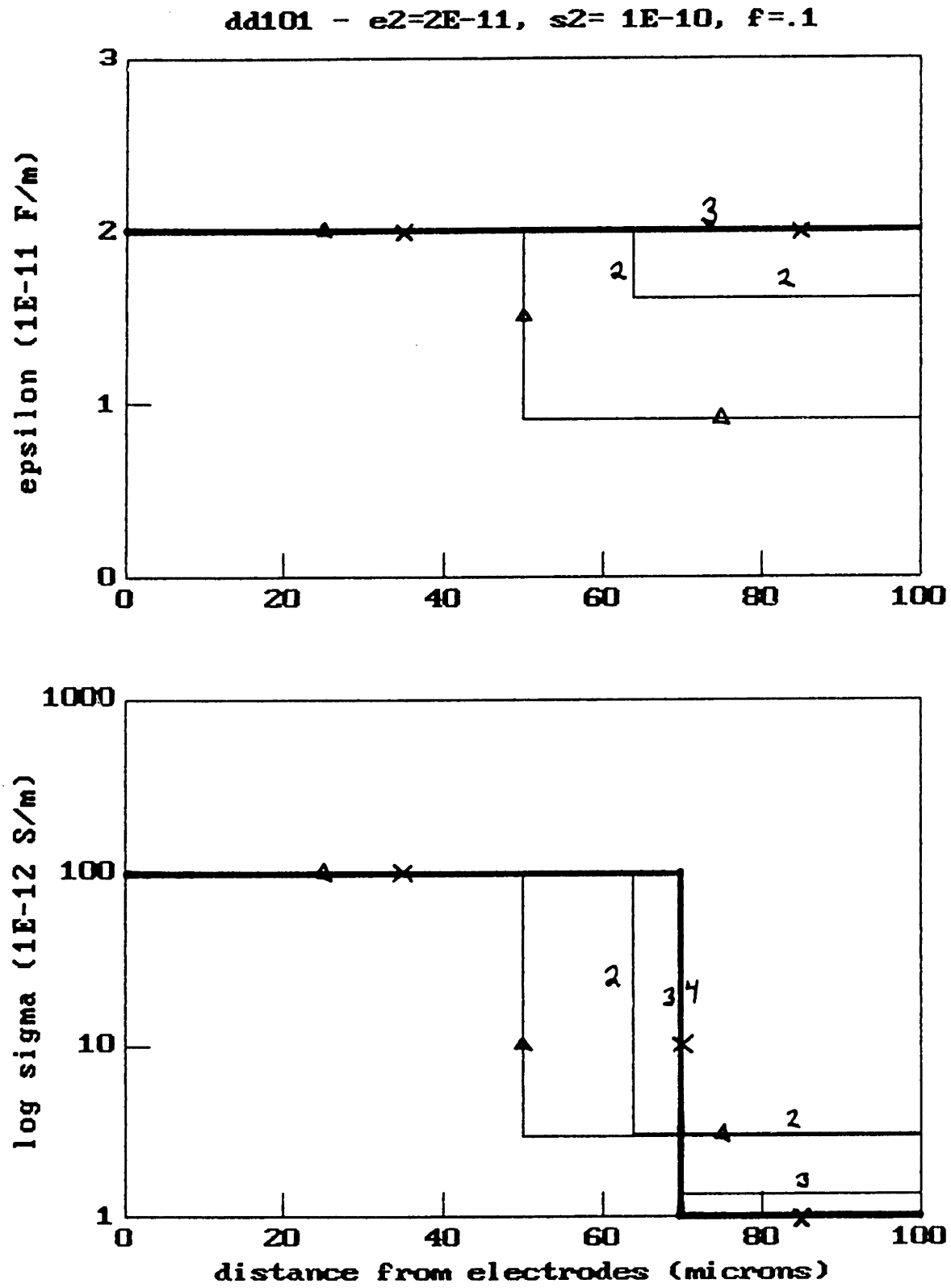


Figure 4.22: Estimated Layer Thickness and Upper Medium Complex Bulk Permittivity Using 400 and 1000 μm Spatial Wavelength Data from Table 4.5 (Case 3, $d_2 = 70$ μm).

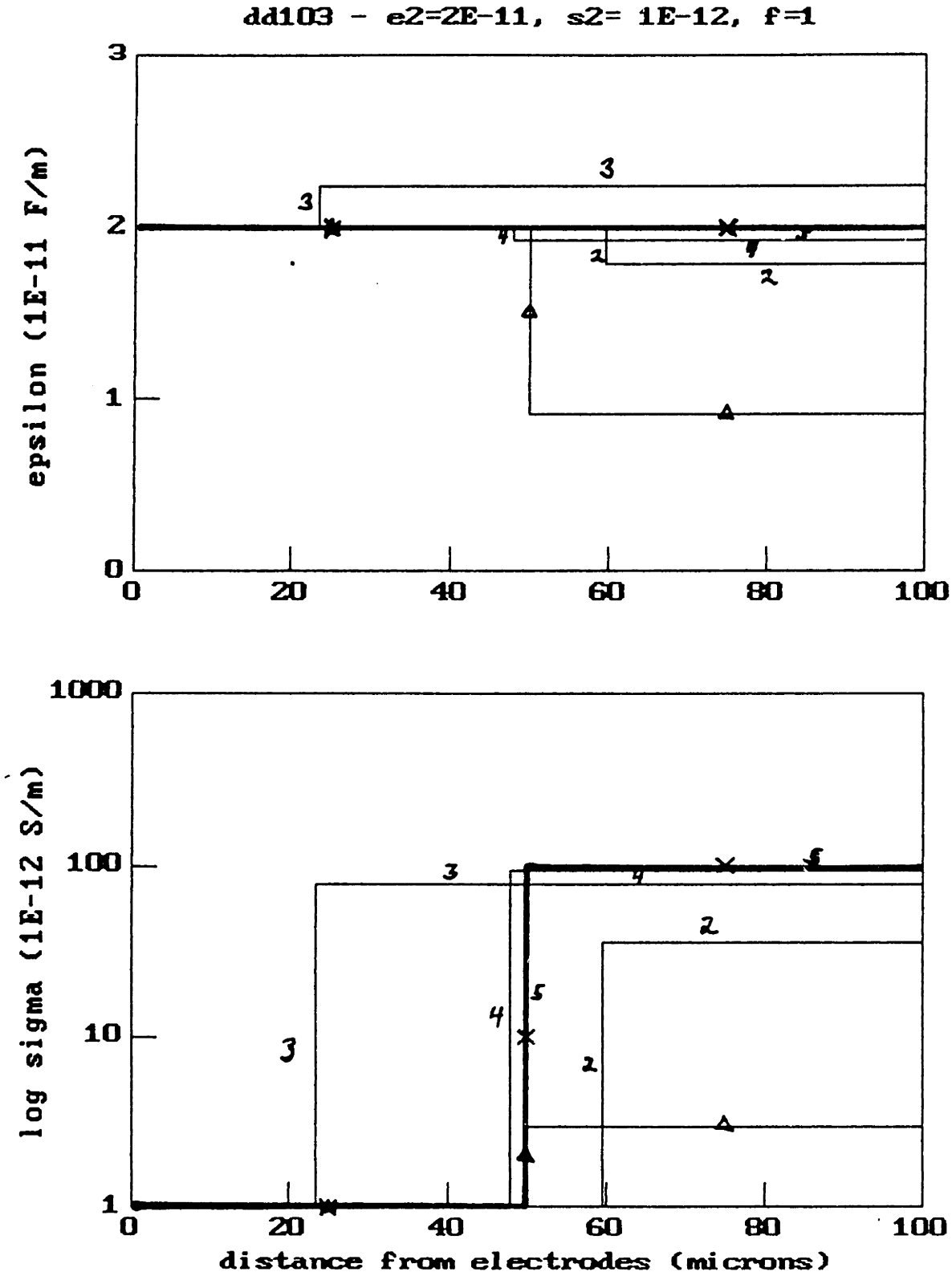


Figure 4.23: Estimated Layer Thickness and Upper Medium Complex Bulk Permittivity Using 400 and 1000 μm Spatial Wavelength Data from Table 4.6 (Case 4, $d_2 = 50$ μm).

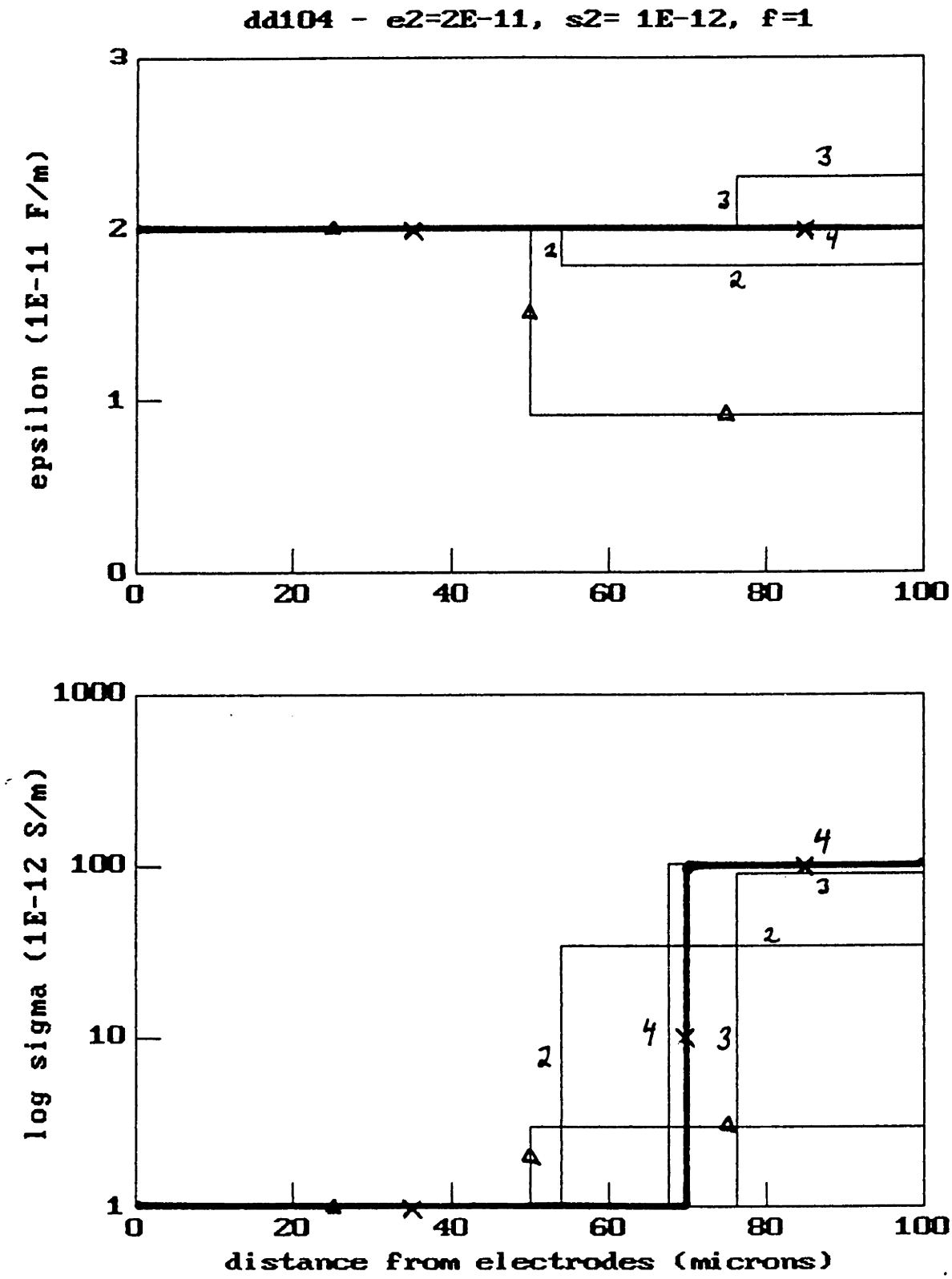


Figure 4.24: Estimated Layer Thickness and Upper Medium Complex Bulk Permittivity Using 400 and 1000 μm Spatial Wavelength Data from Table 4.6 (Case 4, $d_2 = 70$ μm).

permittivity of the layer adjacent to the electrodes has been estimated using a very short spatial wavelength measurement. Now, a two parameter estimation is performed to obtain the surface conductivity and complex bulk permittivity of the material beyond this layer using the data from two or more longer spatial wavelengths. This two parameter estimation is again obtained using the formalism described in Section 4.1.2. A complex jacobian is formed by taking numerical differences to approximate the local derivatives. Here, the optional initial guess that is programmed into the routine begins with a complex surface permittivity that is purely imaginary, with $\sigma_{s2} = (\omega\epsilon_o\lambda_1/4)$ S, where λ_1 is the shortest spatial wavelength used for this estimation and ω is the angular temporal frequency at which the response was measured, and a semi-insulating, nonpolar material above this layer ($\epsilon_1 = 9.07x10^{-12}$ F/m and $\sigma_1 = \omega\epsilon_o$ S/m). Again, to improve robustness, the routine begins with a programmed guess for the surface conductivity in a regime where $((\sigma_{s2}4/\omega\epsilon_o\lambda) \sim 1)$. It is only in the regime where the surface conductivity contributes to the loss that the parameter may be estimated. The routine allows suppression of the real part of the complex surface permittivity if only a surface conductivity is to be estimated. The software routine is **discon2.for** and may be found in Appendix F.

The first example involves a layer 50 μm thick, having a bulk permittivity of $2x10^{-11}$ F/m, a bulk conductivity of 10^{-10} S/m and a surface conductivity of 10^{-16} S, below a medium of unknown complex bulk permittivity. The correct parameter values, along with the estimated values at each iteration, are shown in Fig. 4.25. In the plot of surface conductivities, the square denotes the actual value while the triangle and cross indicate the first and last iteration, as before. This search was conducted using simulated data obtained from 400 and 1000 μm electrode structures at 0.01 Hz. Convergence occurs in 4 iterations. This search is repeated using simulated data obtained from 600 and 1000 μm electrode structures at 0.01 Hz (Fig. 4.26). Convergence again occurs in 4 iterations but the accuracy of the parameter estimates is not as good. The closer spatial wavelength pair, here 600 and 1000 μm, are less sensitive to the exact parameter values and consequently, are less able to discern the exact parameter values. The simulated data used is tabulated in Table 4.7.

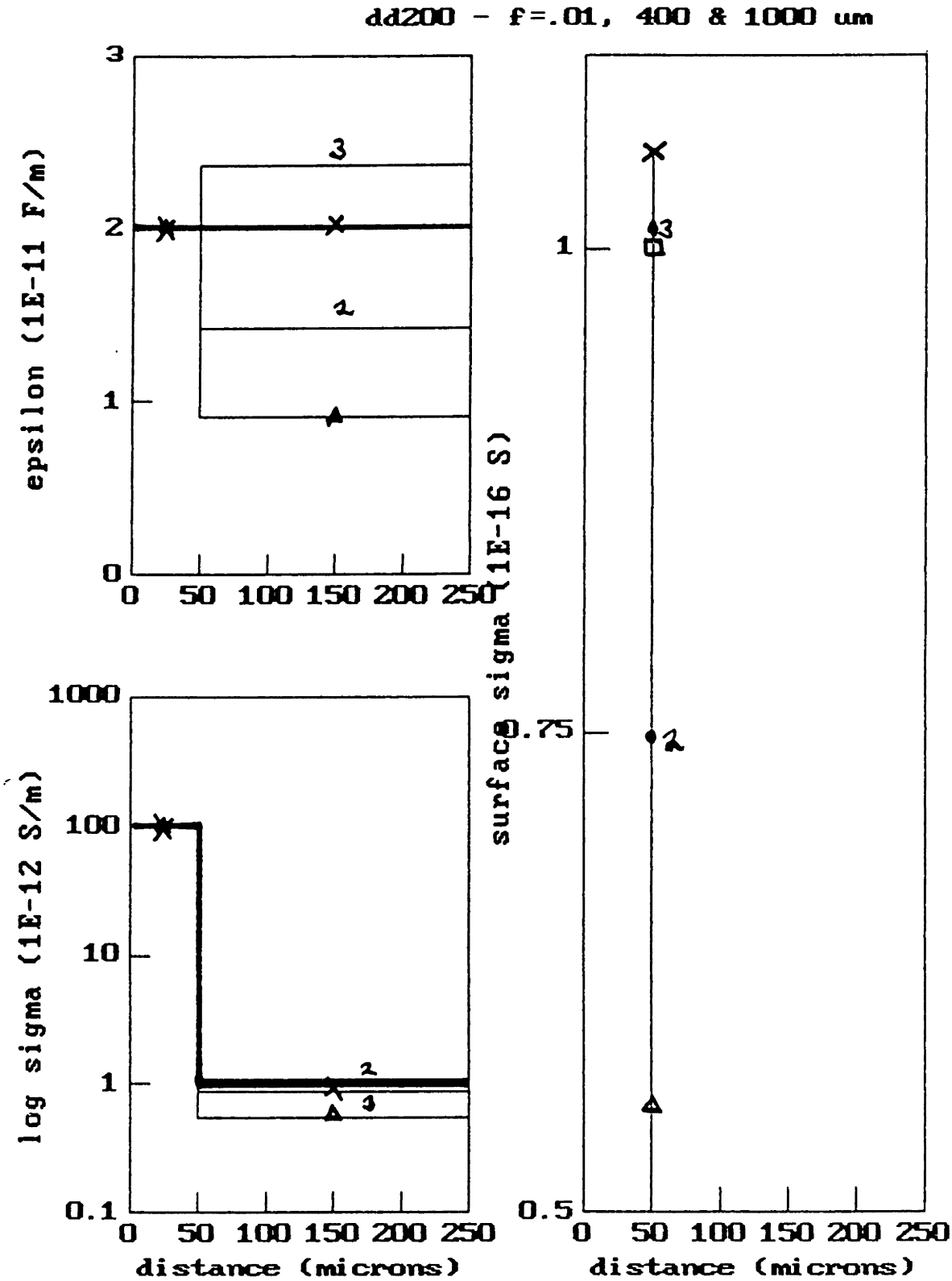


Figure 4.25: Estimated Layer Surface Conductivity and Upper Medium Complex Bulk Permittivity Using 400 and 1000 μm Spatial Wavelength Data from Table 4.7 (Case 5, $\sigma_{s2} = 10^{-16}$ S).

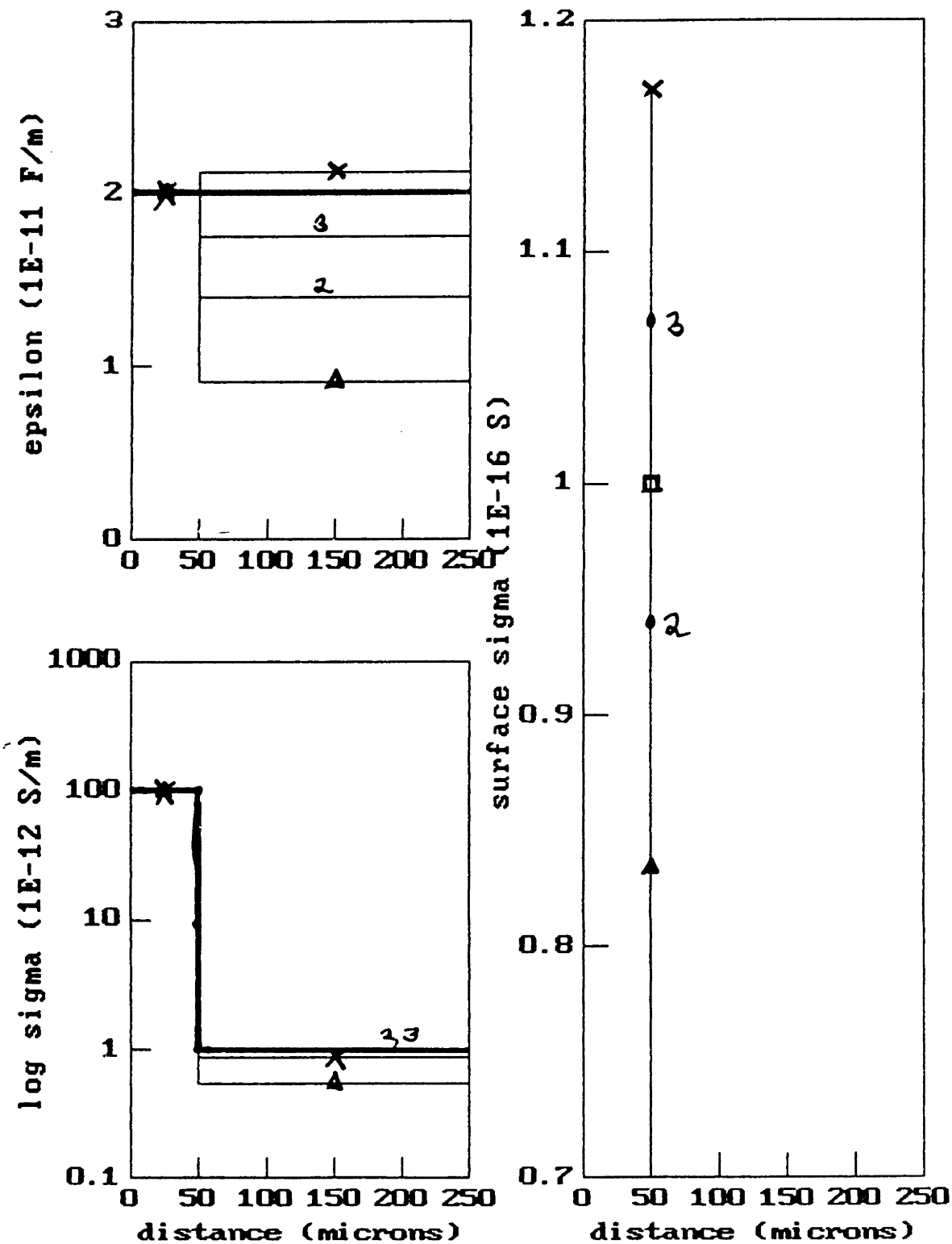


Figure 4.26: Estimated Layer Surface Conductivity and Upper Medium Complex Bulk Permittivity Using 600 and 1000 μm Spatial Wavelength Data from Table 4.7 (Case 5, $\sigma_{s2} = 10^{-16}$ S).

Table 4.7: Simulated Data for Case 5.

	$\sigma_{s2} = 10^{-16}$ S		$\sigma_{s2} = 10^{-17}$ S	
λ (μm)	gain (dB)	phase (deg.)	gain (dB)	phase (deg.)
400	-8.96	-74.18	-50.69	-34.12
600	-10.91	-83.67	-	-
1000	-15.02	-104.40	-51.95	-13.03

A second example, having a layer closer in conductivity to the upper medium ($\epsilon_2 = 2x10^{-11}$ F/m, $\sigma_2 = 10^{-12}$ S/m), is shown in Fig 4.27. Here, the search used simulated data from 400 and 1000 μm electrode structures again at 0.01 Hz. Convergence occurred in 4 iterations and yielded very good estimates of the actual parameter values.

4.4 Two Parameter Estimation Using Variable Dielectric Media

Very often, both the layer thickness and permittivity are unknown, requiring two data points for the estimation problem. One parameter that can be easily varied, in many situations, to obtain the two data points is the permittivity of the surrounding medium. As with the layer thickness estimation of Section 4.2.1, the search is in the high frequency, purely capacitively coupled regime. Thus, the gain used in (4.11) is purely real.

Unlike the complex permittivity estimation, this two dimensional search cannot be converted to a one dimensional search of a complex parameter. Therefore, the minimization technique presented in Section 4.1.2 is utilized here. The guesses are updated using a $2x2$ jacobian matrix (4.8). Here, with $\theta_1 = d_2$ (layer thickness), $\theta_2 = \epsilon_2$ (layer permittivity) and G_p the real part of the complex gain, the jacobian is

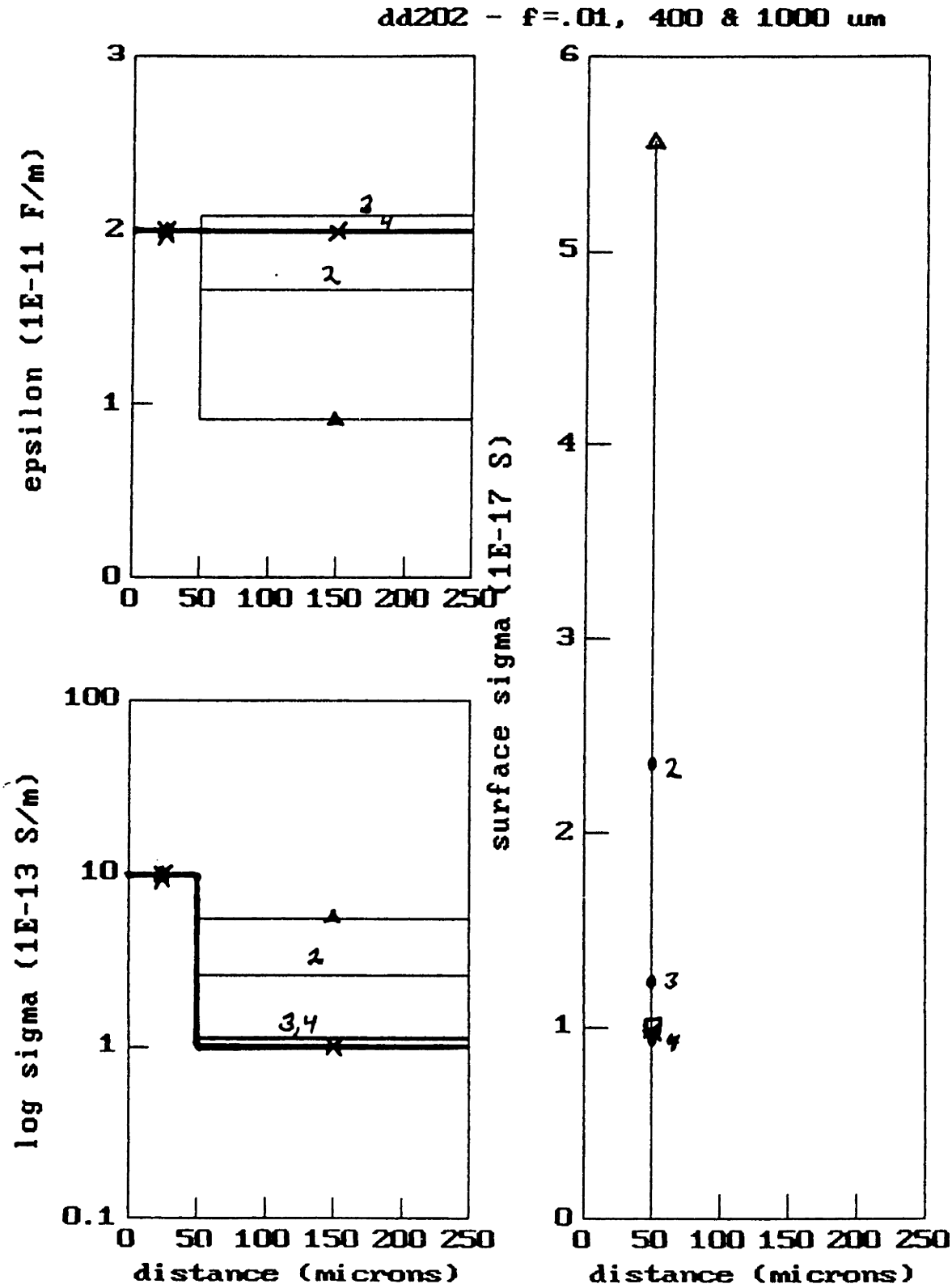


Figure 4.27: Estimated Layer Surface Conductivity and Upper Medium Complex Bulk Permittivity Using 400 and 1000 μm Spatial Wavelength Data from Table 4.7 (Case 5, $\sigma_{s2} = 10^{-17}$ S).

$$\mathbf{e}'(\theta) = -\begin{bmatrix} \frac{\partial G_{p1}}{\partial d_2} & \frac{\partial G_{p1}}{\partial \epsilon_2} \\ \cdot & \cdot \\ \cdot & \cdot \\ \cdot & \cdot \\ \frac{\partial G_{p2}}{\partial d_2} & \frac{\partial G_{p2}}{\partial \epsilon_2} \end{bmatrix} \tag{4.27}$$

For each guess, 4 additional function evaluations are required to obtain the partial derivatives. The time required for a search was reduced somewhat by again using the variable collocation point technique, searching first with 2, then 10 and finally 25 collocation points. Convergence usually takes anywhere from 5–20 iterations. No check on the validity of the data is made in this routine. For robustness, guesses are again bounded using physically motivated limits. There is also a maximum limit on the number of iterations.

The software for this two parameter estimation routine, **th4est.for**, may be found in Appendix F. This routine was used to estimate the thickness and permittivity of the plasma deposited bromobenzene layers (Table 5.1).

In summary, all of the estimation routines described here demonstrated reasonable convergence, with the exception of the variable spatial wavelength estimations of Section 4.3. In the following chapter, examples will be presented demonstrating the utility of the layer thickness and complex bulk and surface estimation routines.

Chapter 5

Parameter Estimation Examples & Oil Complex Permittivity Measurements

There are several topics to be covered in this chapter. First, it is important to obtain experimental verification of the absolute measurement capability of the interdigital electrode structures when used in conjunction with the continuum model, and the parameter estimation techniques previously described. Second, the test cases chosen for the verification were motivated by possible applications to transformer monitoring and involve measurements of the dielectric properties of oil, paper insulation and moisture sensitive thin film coatings. Third, some of the experience, when using the microchip and macrochip for various monitoring applications, is related. This includes situations when problems occur, such as the complex bulk permittivity measurement of insulating fluids with a microchip.

After outlining the calibration techniques for the microchip and macrochip in Section 5.1, several cases are cited in which parameters are both estimated from experimental data and measured independently. These cases consist of both the layer thickness of the thin film coatings (Sections 5.2 and 5.3) and the bulk complex permittivity of oil (Section 5.4). Additional cases are presented where the experimental response can be predicted using the continuum model, though no independent measurement

was performed. These cases include monitoring the sedimentation of small particles (Section 5.2.1) and the surface adsorption of moisture (Section 5.4.1), when it is difficult to envision simple, alternative techniques for these situations. The sedimentation example arose from the idea of monitoring the particulate concentration in transformers while the surface adsorption example was the result of a problem encountered when using an uncoated microchip in wet oil.

For some cases, where the medium is not ohmic but dispersive in complex permittivity, the dispersion can be estimated, allowing either a better understanding of the loss mechanisms or an interpretation of the response when using the medium in other physical situations. Examples of the former include the study of the surface conduction between electrodes due to moisture adsorption or the behaviour of submicron films deposited on the microchip (Section 5.5). The submicron films were vacuum deposited palladium and/or tin oxide and were motivated by the idea of adapting the microchip to a hydrogen gas sensor. An example of the latter is using a passivated electrode structure to estimate the complex bulk permittivity of oil (Section 5.4.2). Examples of both instances include measurements of the complex bulk permittivity of paper insulation (Section 5.6) and moisture sensitive thin film coatings (Section 8.2.3).

5.1 Calibration

5.1.1 Microchip

For the case of the microchip, there is significant variability in the oxide layer thickness due to the fabrication process used. Therefore, before any parameter estimation can be made, the response of an uncoated microchip must be calibrated by estimation of the oxide layer thickness. This is usually performed using the high frequency gain of the bare microchip in air. Given this gain, and the load admittance (Y_ℓ), the oxide layer thickness can be estimated using a secant search similar to the one outlined in Section 4.2.1 for layer thickness. The error tolerance for convergence is 0.03 dB. This routine is called **hest.for** and may be found in Appendix F.

Calibration of the microchips, as performed by Micromet, is slightly different. The expected range of high frequency gain in air, given the variability in oxide layer thickness, is from −39.9 to −43.0 dB. This range is divided into 9 intervals of 0.3 dB width and 1 interval of 0.5 dB width, with a chip identification number assigned to each interval. This ID number, when input into the microdielectrometer, indicates the particular lookup table to be used for mapping the gain–phase space into the complex bulk permittivity space (using a uniform medium model). The 0.3 dB width intervals correspond to a variation of $0.02\epsilon_o$ F/m in permittivity.

5.1.2 Macrochip

For the case of the macrochip, as mentioned in Section 1.4.2, there was a problem associated with fringing fields and ascribed to the smaller aspect ratio of the electrodes presently available [74]. The deduction was that fringing fields resulted in a measurable reduction in sensitivity to the medium above the electrodes than predicted by the continuum model. A compensation technique for calibrating the macrochip was developed by Li [23, p. 40]. In this technique, drawing on the discussion of Section 3.1.5, the coupling admittance Y_{12} is separated into two parallel admittances representing the coupling through the medium above and below the electrodes. The effect of fringing is modeled as contributing a purely capacitive component that is independent of the medium above. Calibration is performed using a high frequency measurement in two different dielectric media and solving for the load capacitance (C_ℓ) and fringing field capacitance (C_x) using the experimental gains and satisfying the relation between the gain and the admittance parameters, (2.1). With a redesign of the macrochip electrodes, specifically an increase in the electrode aspect ratio, it is expected that the fringing field component will be negligible.

5.2 Thickness Gauge

A simple application of these interdigital electrode structures, in conjunction with the parameter estimation routines discussed in Chapter 4, is as a measure of the thickness

of dielectric or conducting layers. If measurements are taken at a frequency such that the coupling between the electrodes is either purely capacitive (or conductive), the response will vary only with the permittivity (or conductivity) and thickness of the medium above the electrodes. This allows measurements of layer thicknesses that would be difficult to obtain, particularly in a dynamic mode, where the layer thickness evolves with time. With the availability of microfabricated structures, this technique projects the thickness measurement into the micrometer and submicrometer range.

The microdielectrometer provides the ability for performing a high impedance measurement (capacitive limit). Several examples of this capability are given in the next section. An examination of the virtual ground measurement (conductive limit) is also presented although no experimental measurements have been made for this case.

5.2.1 High Impedance Measurement

Gain to Thickness Mapping

When configured in the high impedance mode using a floating gate electrode, the device output at relatively high frequencies will reflect the purely capacitive coupling between the electrodes. There will be an attenuated response, in phase with the drive, varying only with the permittivity and thickness of the medium above the electrodes. This output is inherently dependent on the parallel capacitances of the media above and below the electrodes. Thus, the output gain is specific to the particular oxide layer permittivity (ϵ_{ox}), thickness (h) and load admittance (Y_ℓ) describing the microchip. The ratios of interest that completely describe the problem are (ϵ_1/ϵ_{ox}), (ϵ_2/ϵ_{ox}), (d_2/λ), and (h/λ). Given the variety of parameter combinations, no single two-dimensional plot can be made to represent all the possibilities.

One limit, shown in Fig. 4.2, is the high frequency gain as a function of thickness for a dielectric layer, with air above, on top of a commercially available microchip [66]. Again, knowing the permittivity of both the layer and the semi-infinte region above, and given an experimentally measured high frequency gain, a unique layer thickness can be estimated. Observe that the sensitivity to layer thickness is practically

nonexistent beyond roughly $\lambda/3$. For a practical implementation, the mapping must take into account the specific oxide leyr thickness of the microchip, a value that can vary by roughly 20%.

Another interesting limit is that of a dielectric layer with an equipotential at its upper surface. This equipotential may be produced by either a highly conducting medium above the layer, or a high surface conductivity on the layer. The ratio of $(\epsilon_1/\epsilon_{ox})$ is unimportant here. For this case, a two dimensional plot can describe all the possible combinations of layer thickness and permittivity when using a microchip. The high frequency gain as a function of thickness for various ratios of $(\epsilon_2/\epsilon_{ox})$, again using a microchip, is shown in Fig. 5.1. Here, the region of high sensitivity (large slope) occurs at higher signal levels (gain closer to unity). Thus, this mode of thickness measurement may provide better accuracy in practice.

Experimental Results

The first demonstration of applying the microchip to thickness monitoring is a qualitative one, involving a mass transfer process using evaporation. A small volume of transformer oil ($\sim$ 1 μL) was placed on the microchip electrodes and the initial frequency response was measured (boxes of Fig. 5.2). The microchip, with the oil sample, was then placed in a chamber and evacuated to remove the oil via evaporation. After a time, the chamber was repressurized to atmospheric and a second response was taken (dots). The third response (crosses) was taken after further evacuation, resulting in complete removal of the oil. The transition of the response from bulk–like to surface–like clearly demonstrates the effect of thinning down the oil layer.

Jumping ahead slightly, the bulk properties of the oil layer prior to evacuation were estimated from the first response, assuming the oil layer was thick enough so that the field never penetrates the air above. The predicted response using these estimated parameters ($\epsilon = 2x10^{-11}$ F/m, $\sigma = 2.87x10^{-11}$ S/m), yielded a good fit to the data (boxes and solid line). The estimated value for the oil permittivity was then used to estimate the oil layer thickness associated with the second response (after some thinning of the layer had occurred). Using this estimated oil layer thickness of 1.09 μm, the bulk properties of the oil layer were then re–estimated ($\epsilon = 2.35x10^{-11}$ F/m,

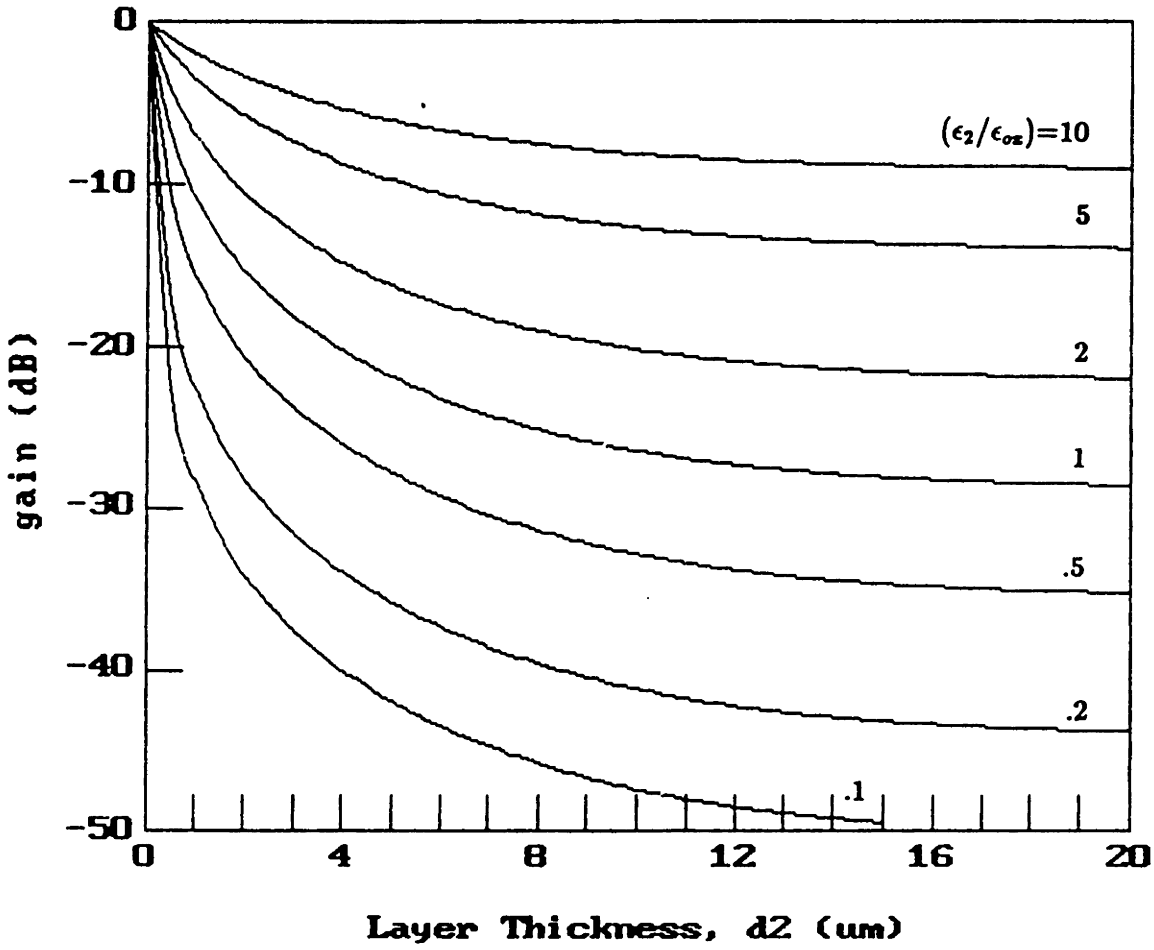


Figure 5.1: Predicted High Frequency (10 kHz) Gains for Microchip with Variable Thickness Layer (d_2), Vary Layer Permittivity (ϵ_2), with Perfect Conductor Above Layer.

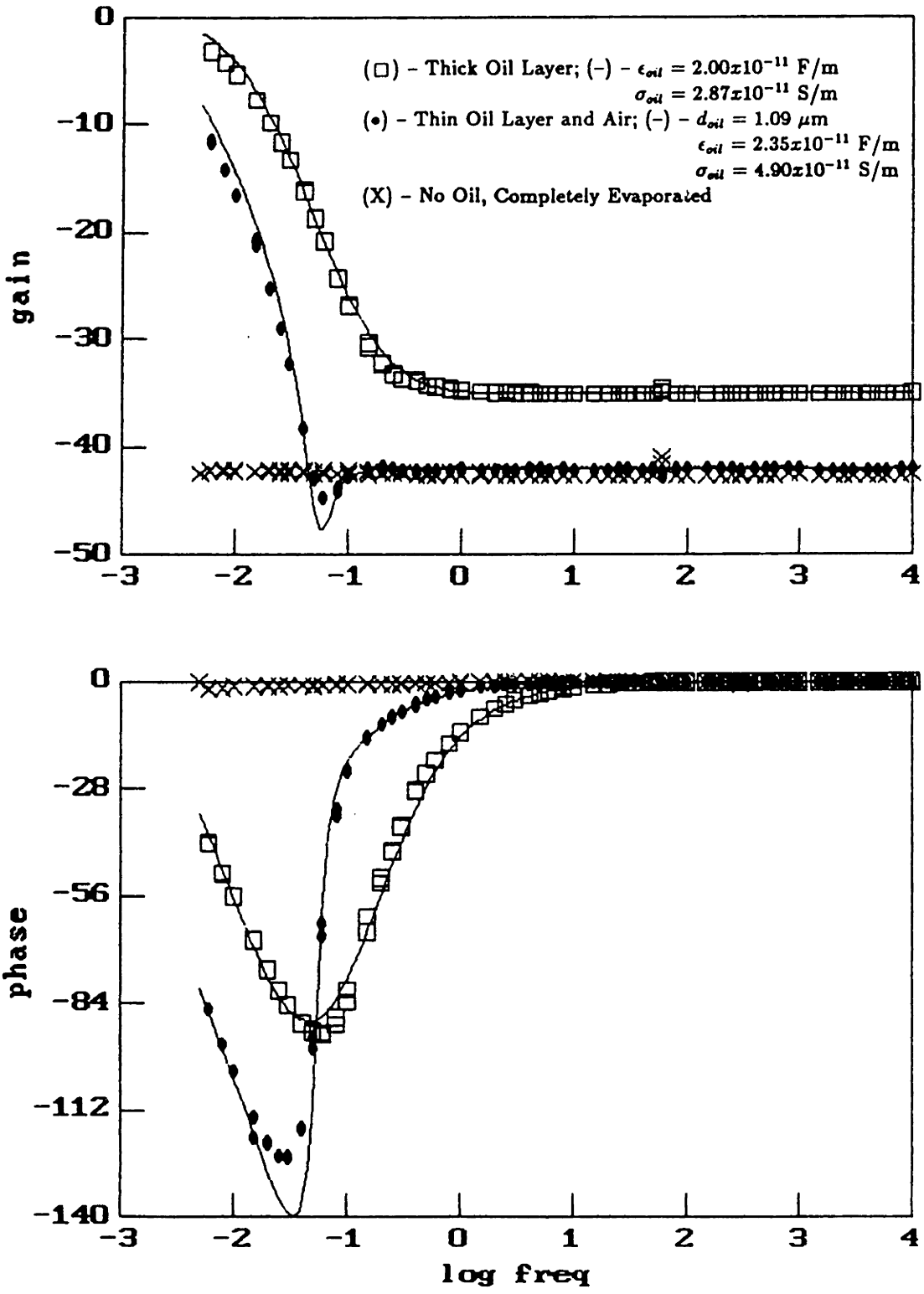


Figure 5.2: Measured and Predicted (-) Frequency Responses of Microchip with Oil Layer of Variable Thickness in Air.

$\sigma = 4.90x10^{-11}$ S/m). Again, the predicted response using these estimated parameters yielded a good fit to the data (dots and solid line). The loop, however, is not a closed one for several reasons. Neither the oil layer thickness nor the complex bulk permittivity were measured independently. Furthermore, the oil permittivity used to estimate the oil layer thickness did not match the re–estimated oil layer permittivity. It is possible that there is a significant change in both the permittivity and conductivity of the thinned down oil layer, possibly due to the removal of lower molecular weight material or changes in ion concentration. A more consistent procedure for estimating the oil layer thickness and complex bulk permittivity (assuming an ohmic model is valid for this thin oil layer) would be to simultaneously estimate both quantities from two or more frequency data points.

The opposite process, that of mass transfer by deposition, provides a quantitative example of a thickness gauge. Using the plasma deposition apparatus described in Appendix C, and used to coat microchips with moisture sensitive thin films, an experiment was conducted whereby a microchip was coated with a very thick film ($\sim$ 33 μm). By taking a measurement at several times during the deposition process, with the microchip in the evacuated deposition chamber but with the RF field turned off, a plot of high frequency gain versus time was obtained (Fig. 5.3). After a long deposition time, the gain saturates due to the thickness of the coating, as also seen in Fig. 4.2. At this thickness, the permittivity can be estimated using a semi–infinite layer thickness. Assuming the permittivity is constant throughout the deposition process, the thickness at each measurement can then be estimated, yielding the linear deposition rate of 165 Å/min. Given the particular configuration used for deposition, this rate compares reasonably well with those reported elsewhere [83, Section 8.3.3.2 and p. 324].

A similar problem is one of monitoring the sedimentation of small particles in a fluid. Specifically, Li [23, Chapter 6] examined the sedimentation of monodisperse, semi–insulating glass beads in transformer oil onto a passivated macrochip sensor. A high frequency (10 kHz) gain versus time plot displayed the same trend as shown in Fig. 5.3 – beginning linearly and ending at a saturated value. Correlations were made between the observed settling velocity and rate of rise of the sedimented layer with

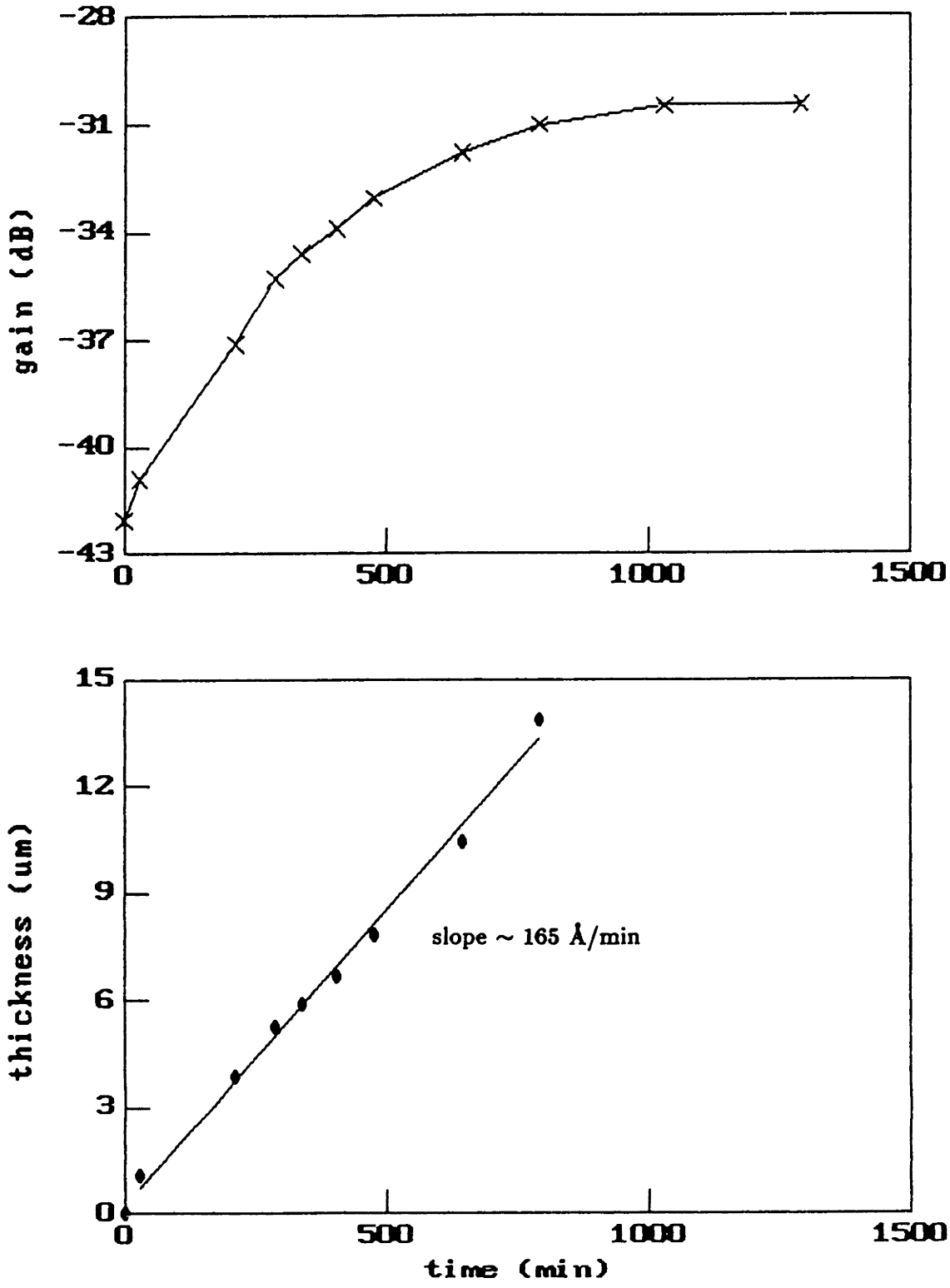


Figure 5.3: Measured High Frequency Gain of Microchip During Deposition of Plasma Deposited Bromobenzene and Estimated Layer Thickness using $\epsilon_2 = 3.0x10^{-11}$ F/m.

estimates of the effective layer permittivity and layer thickness. As far as the feasibility of a particulate monitor was concerned, it was determined to be impractical due to the polydispersity and wide range of dielectric properties of the particles present in transformer oil.

5.2.2 Virtual Ground Measurement

Conductance to Thickness Mapping

In this limit, the "floating" electrode would be terminated with a virtual ground, providing a direct measurement of the admittance Y_{12} (see Fig. 2.1). At a temporal frequency low enough such that $\omega\tau_e \ll 1$, where τ_e is the charge relaxation time within the medium of interest, the coupling between the electrodes would be purely conductive. The measured admittance would be purely conductive, G_{12}, and would depend only on the conductivities of the layer and medium above, and the layer thickness. Thus, given the two conductivities and a measured conductance, the layer thickness could be estimated, using the same estimation routine. Furthermore, assuming the oxide layer remains insulating, the only ratios of importance now are (σ_1/σ_2) and (d_2/λ). A two–dimensional plot can describe all the possible combinations of materials above the electrodes. This assumes the conductivities are ohmic or that the dispersive nature is well characterized in the frequency regime of interest.

A plot of the normalized conductance, $(G_{12}/\sigma_2 M_{el})$ as a function of normalized thickness, (d_2/λ), for various ratios of (σ_1/σ_2), is shown in Fig. 5.4. As expected, the measured conductance will be larger for greater ratios of (σ_1/σ_2) and approaches the conductance of the lower layer as the layer thickness increases. This limiting normalized conductance is 0.967, not unity. In a crude sense, the normalized conductance seen by the electrodes in this limit represents the ratio of the electrode width $(\lambda/4)$ to the effective "length" of the conductor. The fact that this ratio is less than unity implies that the path length of the conduction, in some average sense, is slightly longer than $(\lambda/4)$. This is not surprising as the fields are distributed throughout the medium and not just concentrated at the electrode edges.

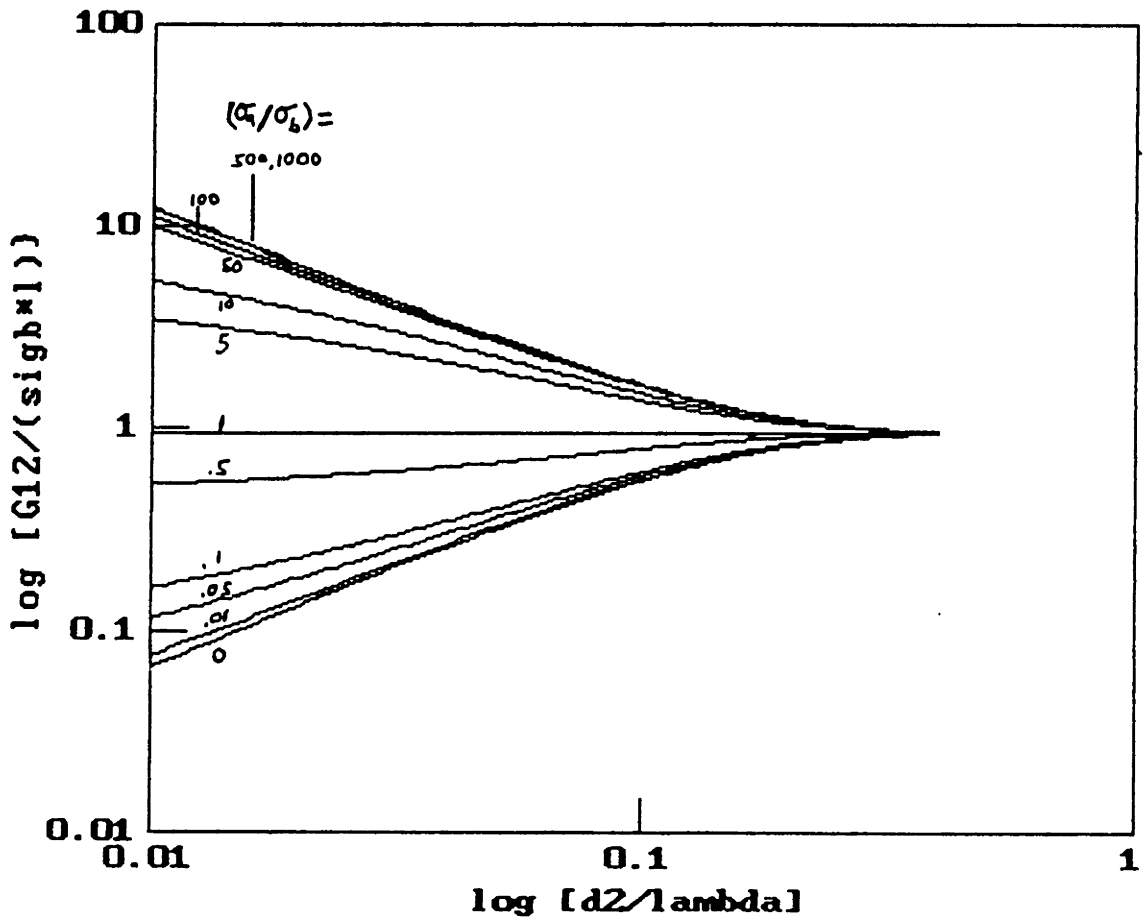


Figure 5.4: Predicted Normalized Conductance ($G_{12}/\sigma_b\lambda$) as a Function of Normalized Layer Thickness (d_2/λ) and Relative Conductivity of Upper Medium (σ_a/σ_b) for Microchip with Virtually Grounded Electrode.

A discussion of the practical issues involved in implementing a virtual ground design may be found in Coln's thesis [68, Section 2.3]. It was concluded that for the application of monitoring epoxy resin cures using relatively low frequencies, this approach was not the most practical one. However, the analysis should be reconsidered for applications in other areas and in other frequency regimes. Aside from the electronic considerations, some emphasis should be given to the potential problems that could arise with the occurrence of significant conduction currents. The electrochemistry occurring at the electrode–medium interface becomes even more important than in the high impedance configuration of the microdielectrometer, with the possibility of significant redox reactions and electrode polarization.

5.3 Layer Thickness and Permittivity

It is not always possible to either know the bulk permittivity of the deposited material beforehand, or deposit a layer thick enough so as to estimate the bulk permittivity independently of the layer thickness. The moisture sensors introduced in Chapter 6 are a good example of this problem. As discussed in Section 4.4, a multiparameter estimation of both the layer thickness and permittivity can be performed by significantly varying the permittivity of the medium above the deposited layer. In this particular case, coated chips were operated both in air and in transformer oil. This provided enough information so as to uniquely estimate both layer properties. A compilation of such estimates for a large set of plasma polymerized bromobenzene coated microchips is presented in Table 5.1. These coated microchips were the moisture sensors studied in Chapters 7 and 8. The high frequency gain of an uncoated microchip in air is used to compute the actual oxide layer thickness, thus calibrating the microchip. The two high frequency gains using a coated microchip, in air and in oil, provide the data necessary for the two parameter estimation. Witness strips composed of silicon were also coated. These strips were partially masked off, producing a step change in height. The coating thickness was measured using a Dektak profilometer, a device that measures step changes in height using a stylus that tracks across the sample. The correlation

Table 5.1: Two Parameter Estimation Using Coated Microchips.

ID #	HFG Uncoated (in Air) (dB)	HFG Coated (in Air) (dB)	HFG Coated (in Oil) (dB)	Thickness meas. (μm)	Thickness est. (μm)	Permittivity est. ($x10^{-11}$F/m)
Z1	-40.30	-39.60	- 32.30	0.7	0.6	3.01
Z3	-41.86	-40.55	-33.80	0.5	1.6	2.44
Z7	-40.87	-33.60	-31.39	7.4	6.8	2.60
Z8	-40.85	-33.74	-31.41	7.4	6.6	2.58
Z17	-42.56	-40.70	-34.30	2.0	2.1	2.63
Z20	-41.00	-34.67	-31.89	-	6.0	2.50
Z22	-41.01	-37.20	-32.30	$\sim$ 3.4	3.5	2.55
Z24	-42.93	-40.52	-34.30	2.4	2.2	2.95
Z25	-42.83	-41.72	-34.75	$\sim$ 1.1	1.29	2.69
C3	-42.63	-35.36	-32.95	6.4	6.7	2.7

between estimated and measured thicknesses is good to within 10%. Included in Table 5.1 is a result obtained with a microchip passivated with a parylene coating (C3). For the parylene sample, a witness strip was produced using a piece of aluminum. The film was peeled off the aluminum and measured with a micrometer.

5.4 Complex Bulk Permittivity of Oil

One of the early goals of this work was to use the microchip as a monitor of the complex bulk permittivity of the transformer oil. As the transformer and oil aged, degradation byproducts might introduce detectable changes in the complex bulk permittivity, changes that would permit early notification of the degradation process. As a natural extension of the epoxy resin application, an uncoated microchip was simply placed in the oil. Besides serving as a complex permittivity monitor, this experiment also provided a direct comparison between the continuum model, using parameter values

obtained with the estimation routine discussed in Section 4.2.2, the finite difference estimate, supplied by the microdielectrometer, and a standard bridge measurement.

One advantage of using the microdielectrometer is its capability to measure the dispersion at low frequencies, if any, in the complex bulk permittivity of the oil. A limitation on standard capacitance bridges, such as a Schering bridge, is the minimum measurable dissipation factor, or loss tangent (ϵ''/ϵ') and is determined by the dissipation of the capacitors within the bridge. As an example, to measure the loss tangent of oil having a bulk conductivity of 10^{-13} S/m at 0.01 Hz implies a capability of measuring a loss tangent of 0.08 at this low frequency, not a simple nor inexpensive requirement. The microdielectrometer, unlike the capacitance bridge, does not try to match the impedance but performs an absolute measurement of the coupling due to the effect of the impedance. Without the problem of having to provide low dissipation capacitors, low frequency measurements of semi-insulating material are now available.

The interest in low frequency measurements lies in the possibility of extracting additional information about the state of the transformer oil from any observed dispersion in the frequency response. Unlike the epoxy resins, dipolar dispersions that might indicate changes in the molecular structure of the oil cannot be observed at these low frequencies. Instead, the observable loss mechanisms at these frequencies are ionic conduction and space charge polarization. The focus of this section is losses due to ionic conduction. The uncoated microchip, having a spatial sensitivity of roughly 15–20 μm, is very sensitive to interfacial effects such as moisture adsorption and electrode polarization. These properties, in turn, are dependent on external parameters such as the moisture content and temperature of the oil. Very often, the frequency response did not reflect the condition of the bulk oil but the conditions existing at the microchip–oil interface. An attempt at improving reproducibility was made by passivating the microchip with a thin layer of parylene, an insulator known for its electrical stability and inertness, and used in biological applications [84]. The parylene coating displayed a dispersive complex bulk permittivity and increased the complexity of determining the oil properties. However, with the use of the continuum model, the dispersion could be estimated and taken into account when estimating the complex bulk permittivity

of the oil. This technique worked fine except under conditions of dry, circulating oil. Under these conditions, problems would arise due to the same interfacial effects, now removed to the coating–oil interface. Though these effects represented an opportunity for using the microchip to study electrokinetic processes, they stood in the way of the goal of measuring the complex bulk permittivity of the oil. By going to a parylene coated macrochip the deleterious interfacial effects were overcome. The longer spatial wavelength of the macrochip prevented the conditions at the interface from dominating the response. However, due to the less than optimum design of the present macrochip (see Section 1.4.2), it is very insensitivity compared to the microchip (gain accuracy of only 0.5 dB). This results in very poor accuracy in the estimates of the bulk permittivity. Thus, at present, the combination of an uncoated microchip (for bulk permittivity) and passivated macrochip (for loss) must be used to provide the complex bulk permittivity of transformer oil. There is no reason why a redesigned macrochip, when passivated, could not provide the complex bulk permittivity of the oil over the entire frequency range. The experiments upon which these conclusions were based are presented in the following two sections.

5.4.1 Uncoated Microchip

There were instances where an uncoated microchip appeared to be responding to the bulk properties of the transformer oil. The response of a microchip in unused oil, at room temperature and having a 70 ppm moisture content, is shown in Fig. 5.5 (dots), along with a predicted bulk response using a complex permittivity estimated from the experimental data at 0.5 Hz (solid line). A similar match is shown in Fig. 5.6 for a different microchip in unused oil, also at room temperature, with a moisture content of 53 ppm. These results are tabulated in Table 5.2 and compared to the complex bulk permittivity as estimated by the finite difference simulation model used by the microdielectrometer. Some of the small discrepancy between the values may lie in the coarser calibration method employed by the microdielectrometer (see Section 5.1).

However, there were more instances in which interfacial phenomena dominated

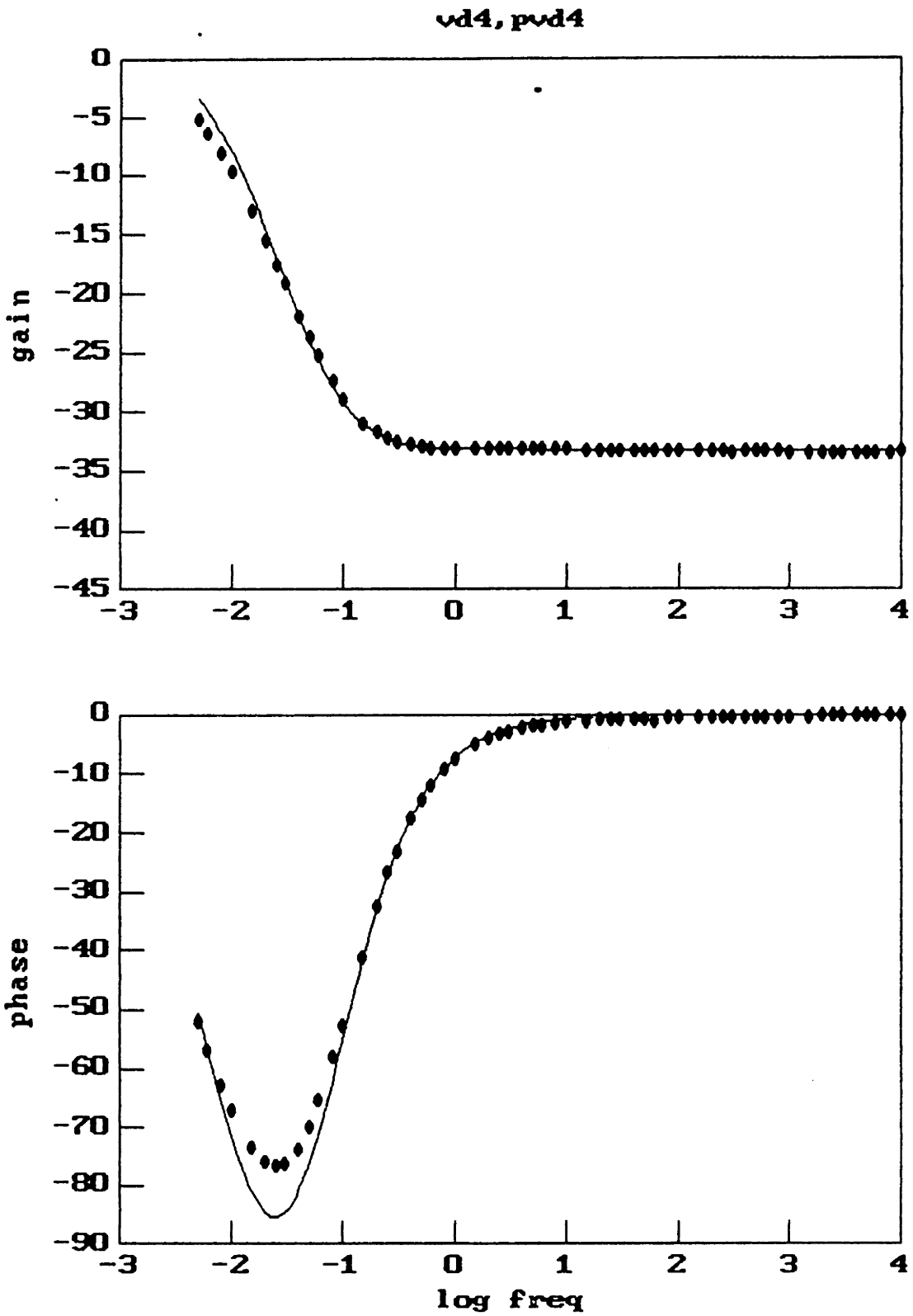


Figure 5.5: Measured (•) and Predicted (–) Frequency Response for Uncoated Microchip in Unstirred, Unused Oil with 70 ppm Water (use $\epsilon_1 = 2.03x10^{-11}$ F/m and $\sigma_1 = 1.46x10^{-11}$ S/m).

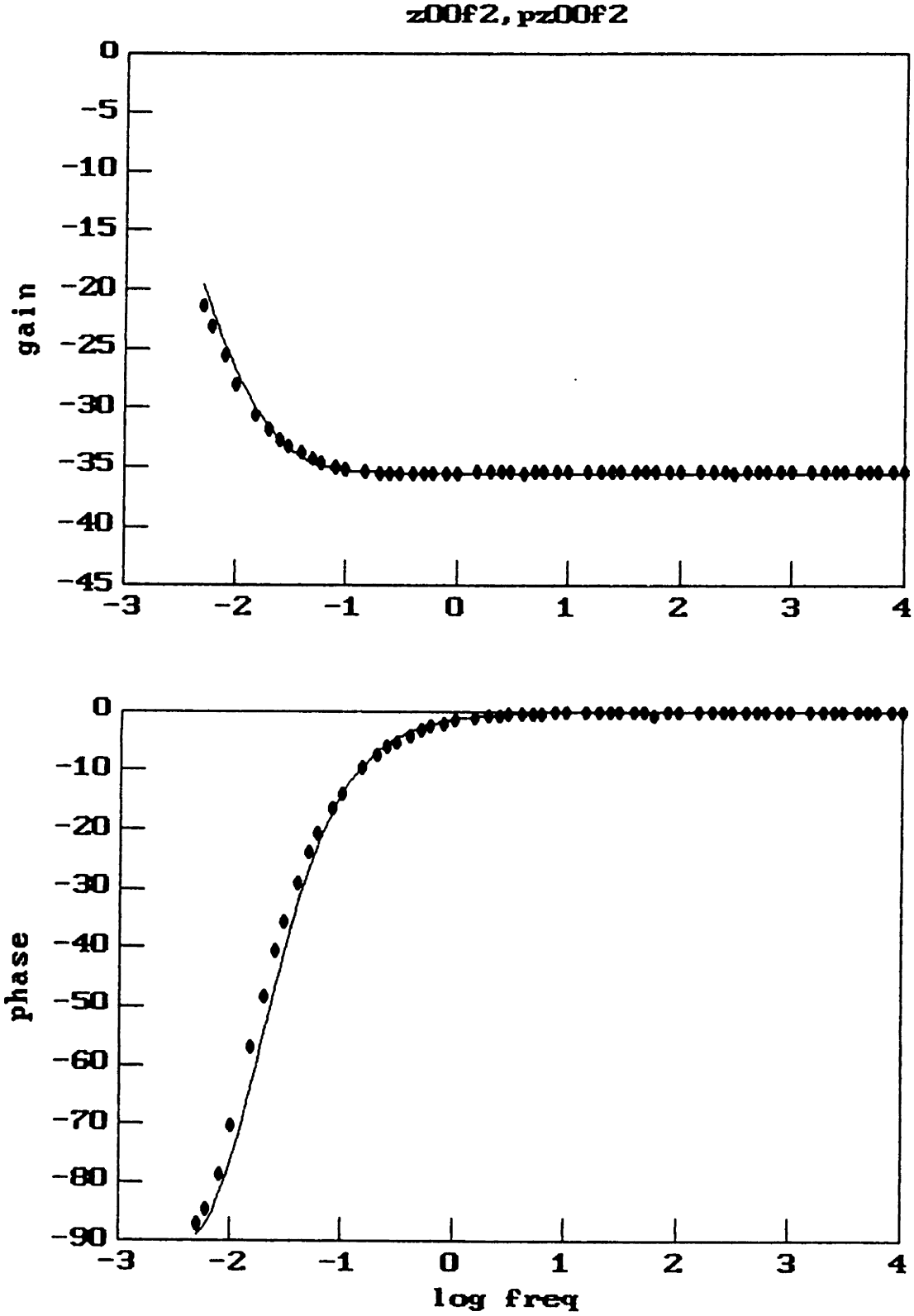


Figure 5.6: Measured (•) and Predicted (–) Frequency Response of Uncoated Microchip in Unstirred, Unused Oil with 53 ppm Water (use $\epsilon_1 = 1.96x10^{-11}$ F/m and $\sigma_1 = 2.77x10^{-12}$ S/m).

Table 5.2: Oil Complex Bulk Permittivity Measurement With Microchip.

Note – ϵ' and ϵ'' relative to ϵ_o F/m

ID #	Oil	HFG	Response in Oil			Micromet		Cont. Model	
	Cond.	(in Air)	freq.	gain	phase	ϵ'	ϵ''	ϵ'	ϵ''
88991.23	unused, 70 ppm	–41.03	0.5	–32.94	–14.22	2.25	0.5083	2.29	0.5249
88025.24	unused, 53 ppm	–42.90	0.1	–35.25	–13.98	2.12	0.4674	2.21	0.4979

the response. One such instance, shown in Fig. 5.7 (dots), displays a response for an uncoated microchip in oil from a failed transformer, having a moisture content of 50 ppm. This response was modeled using by giving the interface between the electrodes a surface conductivity $\sigma_{so} = 2.52x10^{-14}$ S and a surface permittivity $\epsilon_{so} = 0$ (solid line). Adsorption of moisture and other ionizable species along the silicon dioxide interface could account for such a surface conductivity.

Examples of other cases where the frequency response differed significantly from a purely bulk response are given in Figs. 5.8–5.10. Fig. 5.8 shows the response in unstirred, unused oil with a moisture content of 14 ppm. This response cannot be accounted for using a combination of surface and bulk conductivities, as can be seen by examining Fig. 3.6. It is believed that charge separation effects, such as double layer formation, are dominating the response in this frequency range, producing a relaxation higher in frequency than would be expected based on the bulk properties of the oil.

Operating a microchip while evacuating and stirring the oil gave rise to drastic and irreversible changes in the DC offset (defined in Section 1.4.1). The response shown in Fig. 5.9 was taken while the oil remained under evacuation (30 mT), having a moisture content of roughly 10 ppm, but now was unstirred. Large, negative DC offsets of roughly -2 V were observed in the response. It is difficult to guess what is occurring in Fig. 5.9 – the response looks somewhat like the surface conductivity

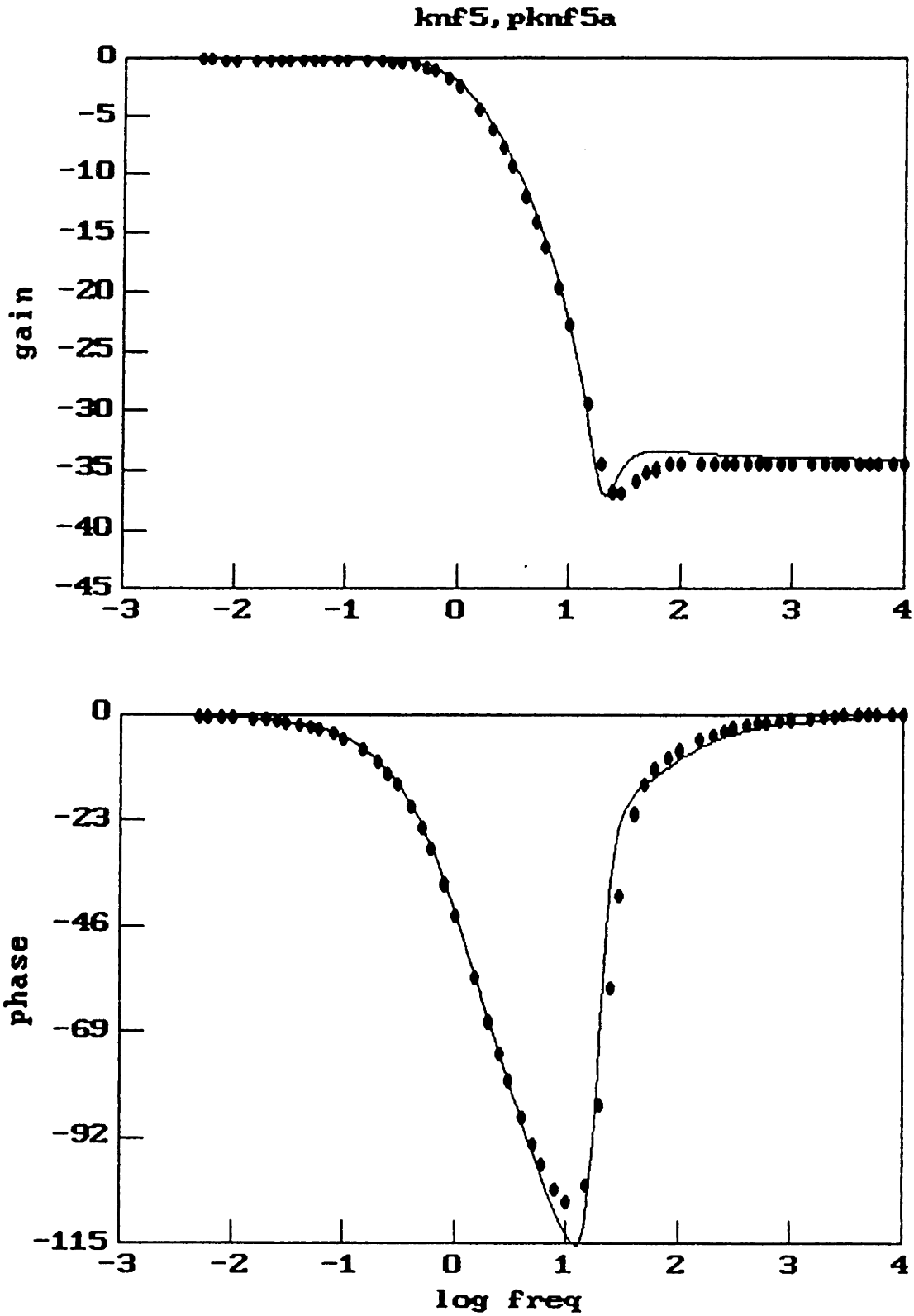


Figure 5.7: Measured (•) and Predicted (–) Frequency Response of Uncoated Microchip in Unstirred, Used Oil with 50 ppm Water (use $\epsilon_1 = 2.0x10^{-11}$ F/m and $\sigma_{so} = 2.52x10^{-14}$ S).

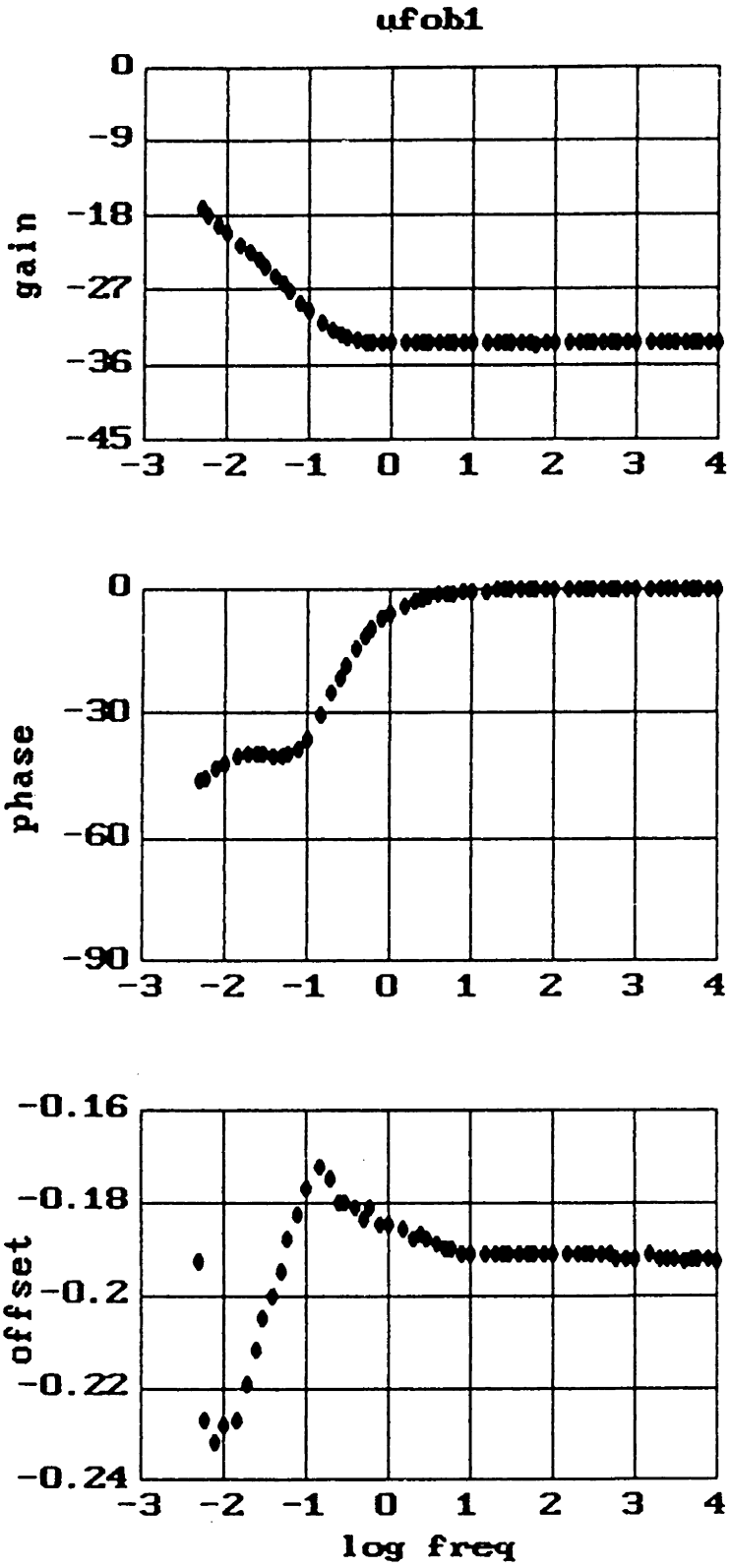


Figure 5.8: Measured Frequency Response of Uncoated Microchip in Unstirred, Unused Oil at 14 ppm Water.

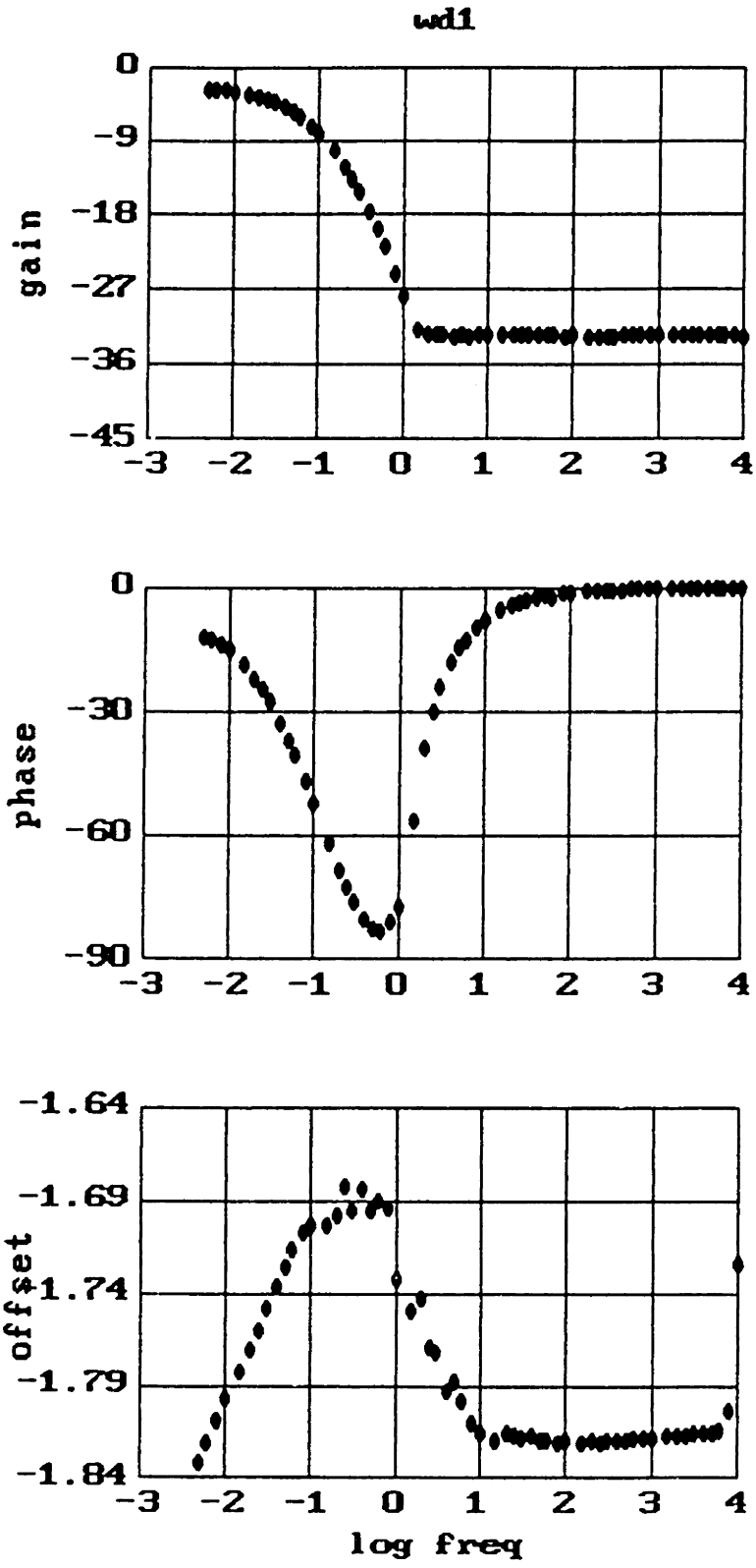


Figure 5.9: Measured Frequency Response of Uncoated Microchip in Unstirred, Evacuated, Unused Oil at roughly 10 ppm Water.

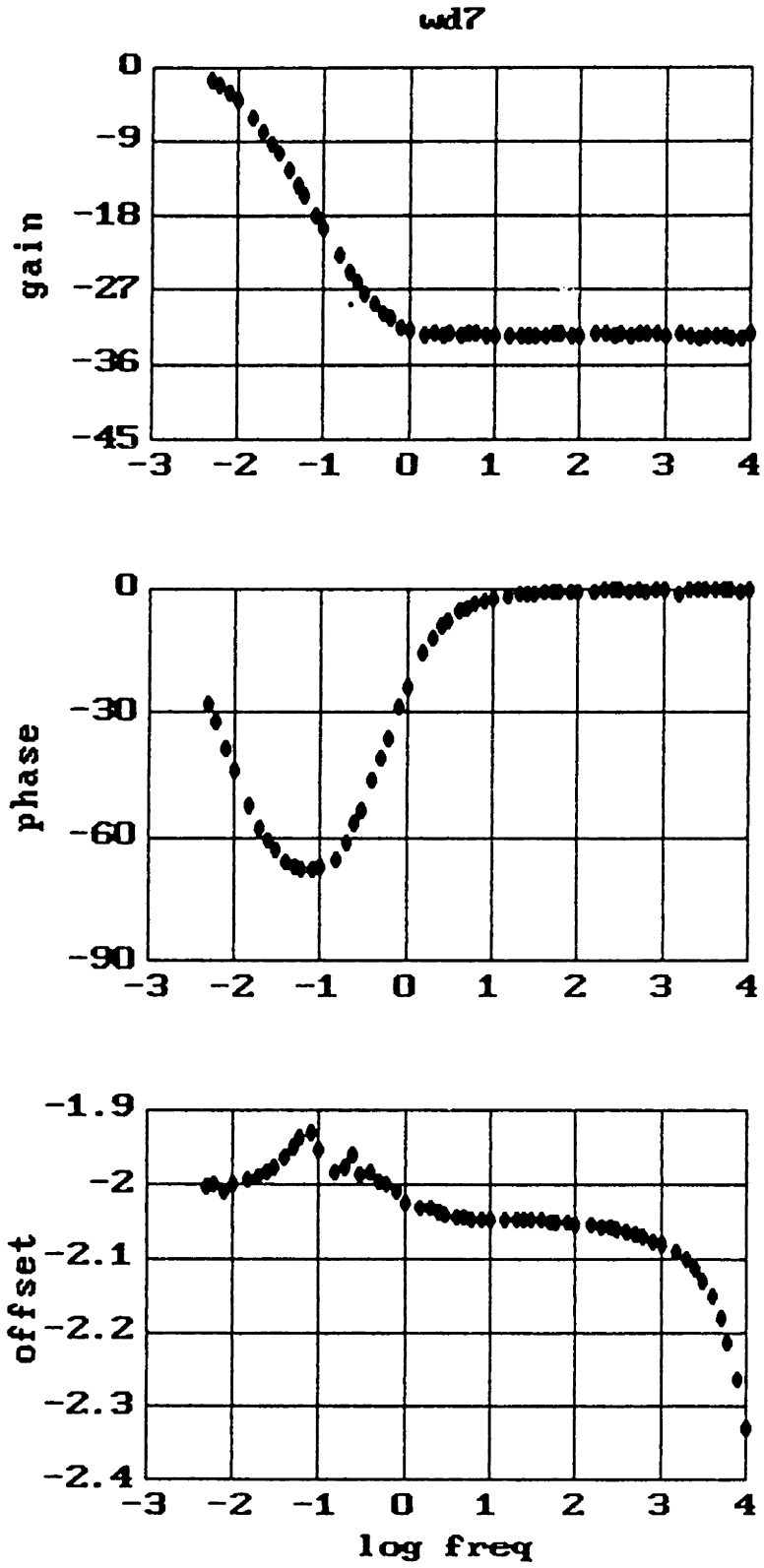


Figure 5.10: Measured Frequency Response of Uncoated Microchip in Unstirred, Unused Oil at 70 ppm Water.

response of Fig. 5.7 with the rapidly rising gain and asymmetric phase. However, the gain levels out below unity gain, indicating a considerable low frequency capacitance. The occurrence of both a surface conductivity and a low frequency capacitance is inexplicable as it would be expected that the surface conductivity would shunt the electrodes together at low frequencies.

After wetting the oil to 70 ppm, the response became that shown in Fig. 5.10. Here, the effect of the large DC offset is seen in the upward shift of the entire gain curve – resulting in low frequency gains greater than unity. The phase curve is significantly broader than those shown in Figs. 5.5 and 5.6. Again, the deviation from a purely bulk response is attributed to interfacial phenomena.

In another experiment, slight changes in the frequency response were observed when attenuating the driven signal by as much as 20 dB (Fig. 5.11). The gain curves of Fig. 5.11a are normalized to give the same gain at 1 Hz in Fig. 5.11b. It is believed that the change in gain and phase (Fig. 5.11c) with field strength indicate the possibility of field dependent conduction in the interfacial region. However, this behaviour may also be the result of the change in offset (Fig. 5.11d) at each attenuated voltage level.

From the above observations it was felt that, except in the case of large DC offsets, the high frequency gain of the uncoated microchip could give a reliable estimate of the bulk permittivity of the oil. Comparisons between estimations from the microdielectrometer, the continuum model and a bridge measurement for various oil samples, all at relatively high frequencies, are summarized in Table 5.3. The correlation between all the measurements is within the accuracy of the estimation procedures.

5.4.2 Passivated Electrodes

A microchip was coated by Paratronix, Inc. [85] with a thin film (6.4 μm) of parylene to remove the problem of moisture adsorption on the electrode interface. Parylene is a trademark encompassing a family of linear polymers produced by pyrolyzing Di–para–xylylene vapors and vacuum depositing the resulting monomer (Fig. 5.12). The actual film deposited was Parylene–C, consisting of a chloride atom on each benzene

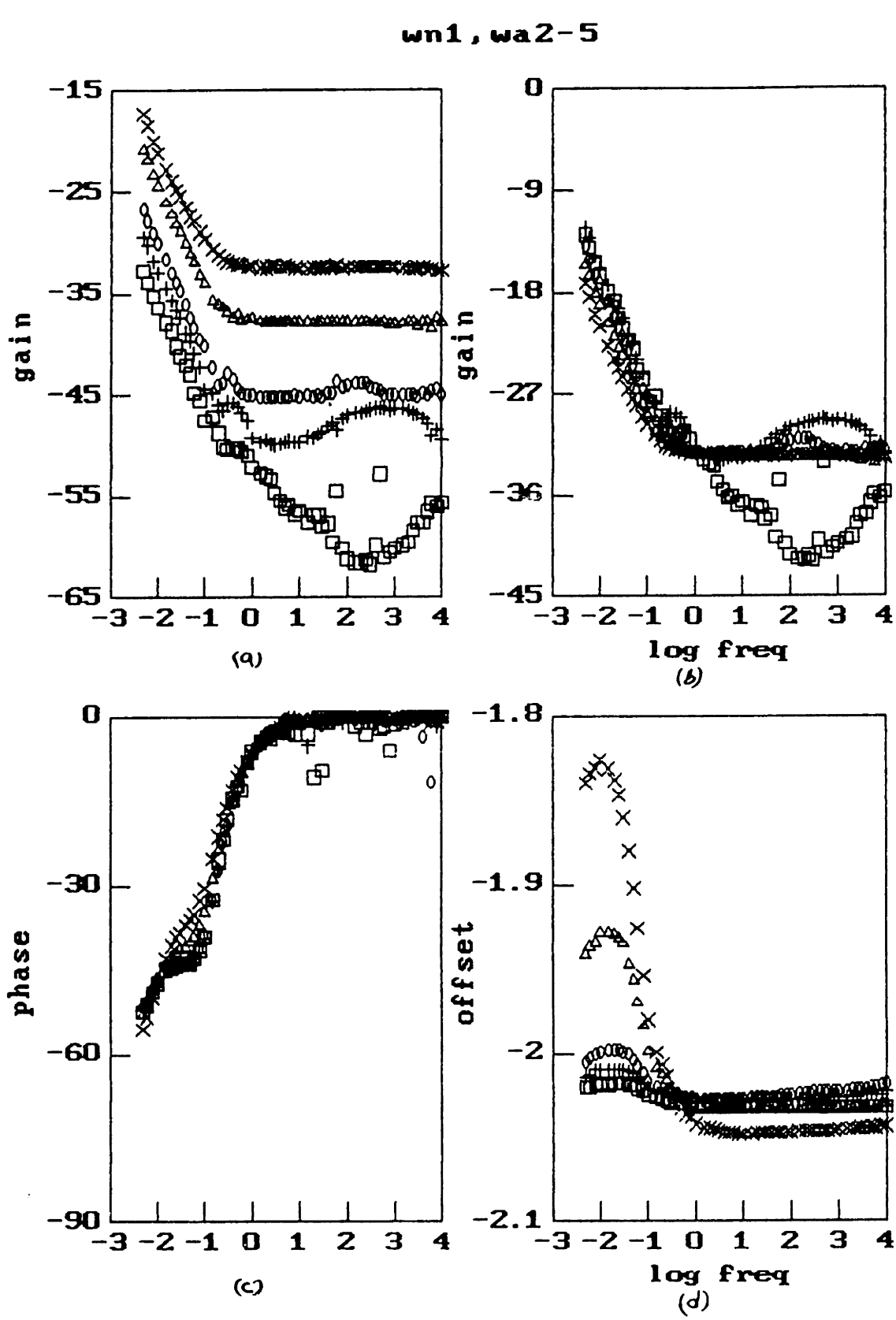


Figure 5.11: Effect of Attenuating Applied Potential on Measured Frequency Response of Uncoated Microchip in Unused Oil (X – full value, △ – 55%, o – 24.5%, + – 14.5%, and □ – 7.2%).

Table 5.3: Bulk Permittivity Measurement Using An Uncoated Microchip And A Bridge.

Notes – ID # = 88025.39, HFG in air = –42.60 dB

measurement with microdielectrometer at 1 kHz, with bridge at 100 Hz

ϵ' relative to ϵ_o F/m

Oil Condition	Response in Oil gain	Micromet ϵ'	Cont. Model ϵ'	Bridge ϵ'
unused	–35.35	2.16	2.16	2.17
"	–35.18	2.20	2.19	2.17
oxidized	–35.05	2.23	2.23	2.25
test facility	–35.20	2.20	2.19	2.17

ring. There are some similarities in the growth mechanisms of Parylene films and films produced by plasma polymerization [83, Section 5.4.2].

The response of a parylene coated microchip in room air is shown in Fig. 5.13. The film is very insulating but does exhibit some relaxation at the lower frequencies. The extended decrease in gain and slight phase with increasing frequency indicates a non–ohmic relaxation. For estimation of the complex bulk permittivity of the oil, the effect of the dispersion in the parylene must be taken into account. This will be done by first estimating the complex bulk permittivity of the parylene, from the response of Fig. 5.13, at the specific frequency of interest. Then, assuming this dispersion remains the same after immersion of the passivated microchip into the oil, the complex bulk permittivity of the oil will be estimated from data taken at a frequency at which the complex bulk permittivity of the parylene is known (via the previous estimation).

For example, after placing the parylene coated microchip in oxidized oil, the frequency response of Fig. 5.14 was measured (dots). First, an estimate was made of the complex bulk permittivity of the parylene bulk properties at 0.08 Hz from the response in Fig. 5.13 (–34.23 dB and –1.73°). The parylene layer thickness was obtained from

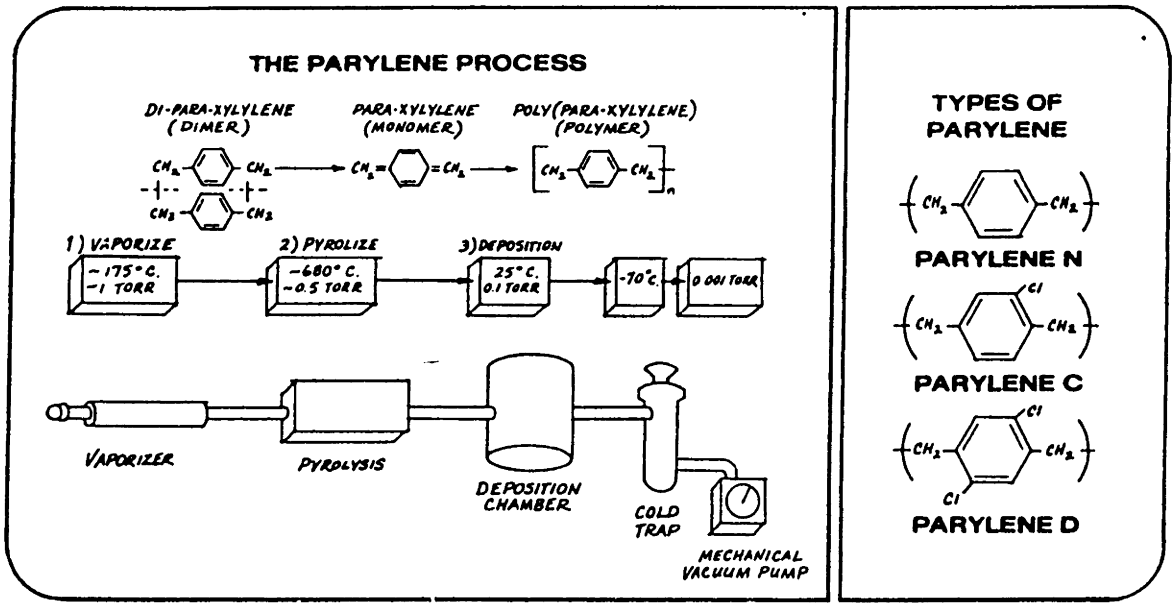


Figure 5.12: Description of Parylene Coating Process and Molecular Structure (from [85]).

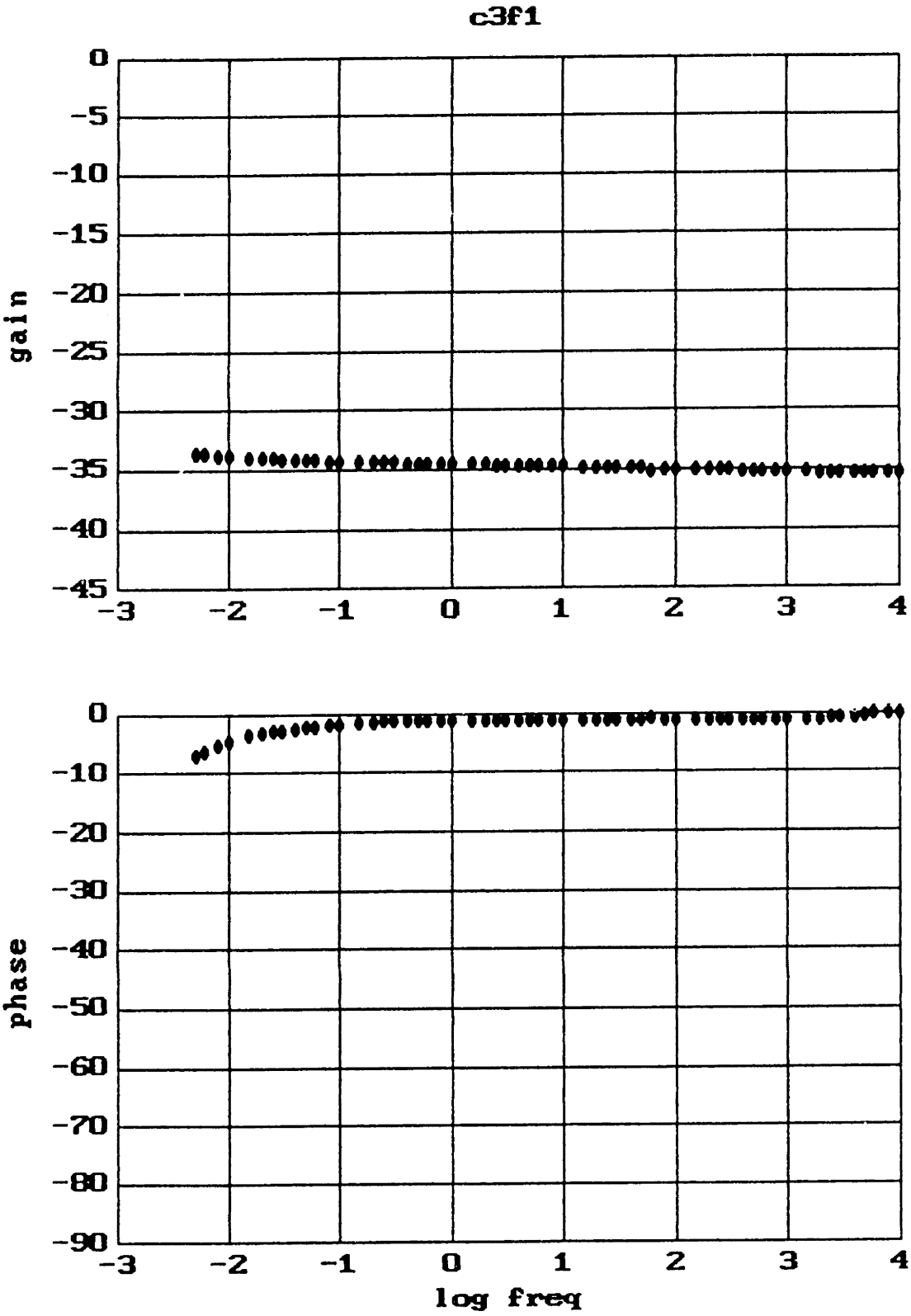


Figure 5.13: Measured Frequency Response of 6.7 μm Thick Parylene Coated Microchip in Air.

the two parameter estimation described in Section 5.3. Using this estimated complex bulk permittivity ($3.11x10^{-11}$ F/m and $5.12x10^{-13}$ S/m), an estimate was made of the complex bulk permittivity of the oxidized oil from the response of Fig. 5.14 at 0.08 Hz (−28.89 dB and −22.13°). The predicted frequency response (solid line in Fig. 5.14) was obtained using the estimated values for the oil ($\epsilon = 2.00x10^{-11}$ F/m, $\sigma = 2.13x10^{-11}$ S/m) and the estimated layer thickness but treating the parylene layer as being a pure insulator of $2.7x10^{-11}$ F/m. The dispersion in the parylene was neglected in Fig. 5.14 to reduce the amount of computation. The neglect of the dispersion in the parylene accounts for much of the difference between the predicted and measured responses in Fig 5.14. A standard bridge measurement of the oil at 20 Hz yielded values of $1.94x10^{-11}$ F/m and $1.56x10^{-11}$ S/m. Some of the difference between the two methods of measurement, microdielectrometer versus bridge, may be accounted for by the different frequencies at which the measurement was made, 0.08 Hz versus 20 Hz. Without verifying the lack of dispersion in the oil, a comparison based on an ohmic model is invalid.

An attempt at examining the dispersion of the oil was made by repeating the process described above at one frequency per decade. The results show the slight dispersion in the parylene (Fig 5.15) and the apparent lack of dispersion in the oil (Fig. 5.16). The solid line drawn in Fig. 5.16 represents the frequency dependence in ϵ'' of an ohmic conductor. The deviation from an ohmic behaviour at the high frequencies represents the sensitivity limit of the microdielectrometer, roughly a minimum loss tangent of 0.005. The deviation in the oil complex bulk permittivity estimates below 0.1 Hz is puzzling. There is still significant phase information the response at these frequencies, providing enough sensitivity in the estimates. Perhaps the assumption that the dispersion in the parylene remains the same in air and in oil is incorrect at these frequencies. Or, perhaps charge separation effects occurring at the coating–oil interface are the cause of the discrepancy (though one might expect the polarization, and hence, the permittivity, to increase in this case).

The inadequacy of the uniform medium model (incorporated into the microdielectrometer) for a coated microchip, is demonstrated in a plot of the estimated values

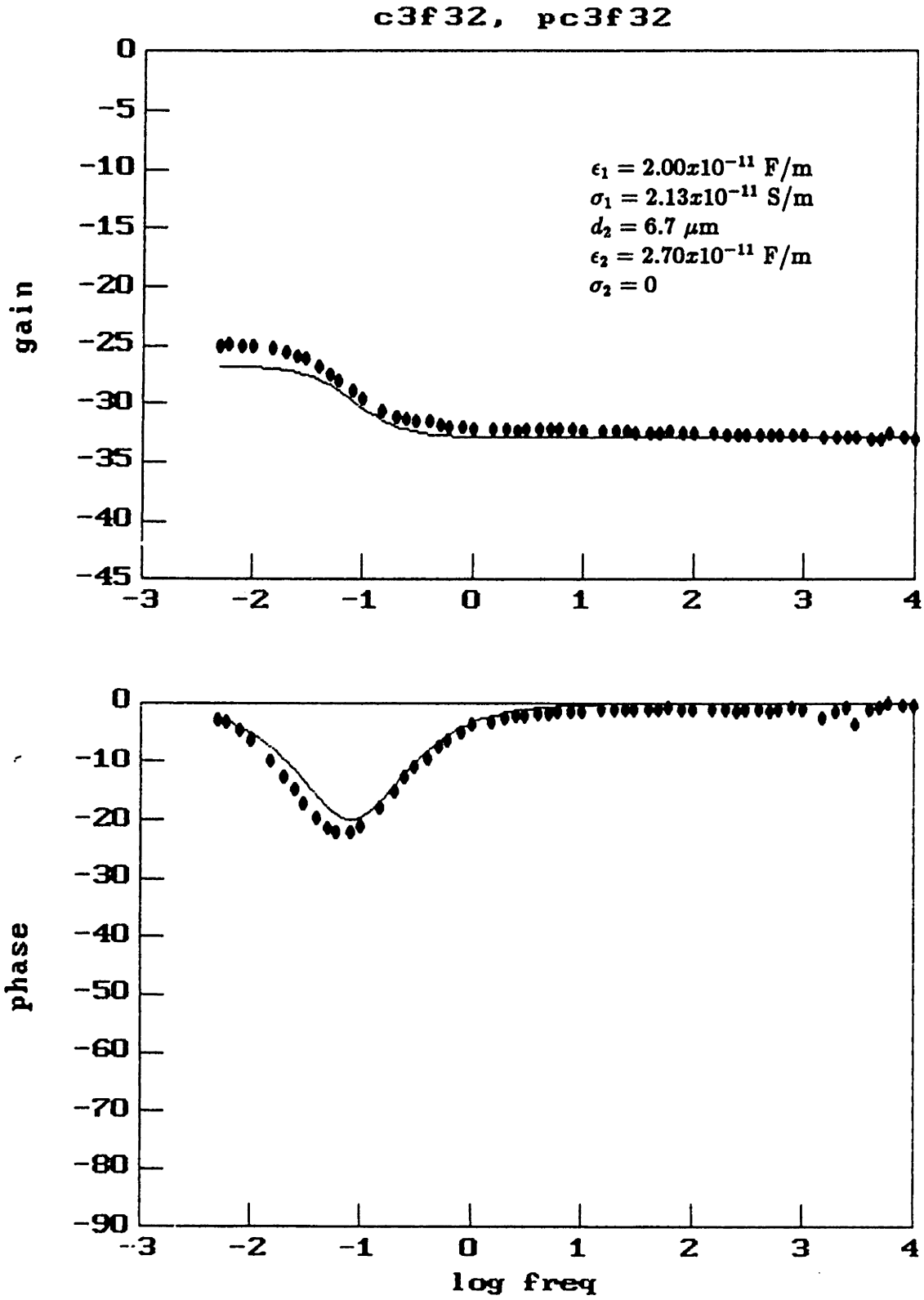


Figure 5.14: Measured (•) and Predicted (–) Frequency Response of a 6.7 μm Thick Coated Microchip in Unstirred, Oxidized Oil at 25°C.

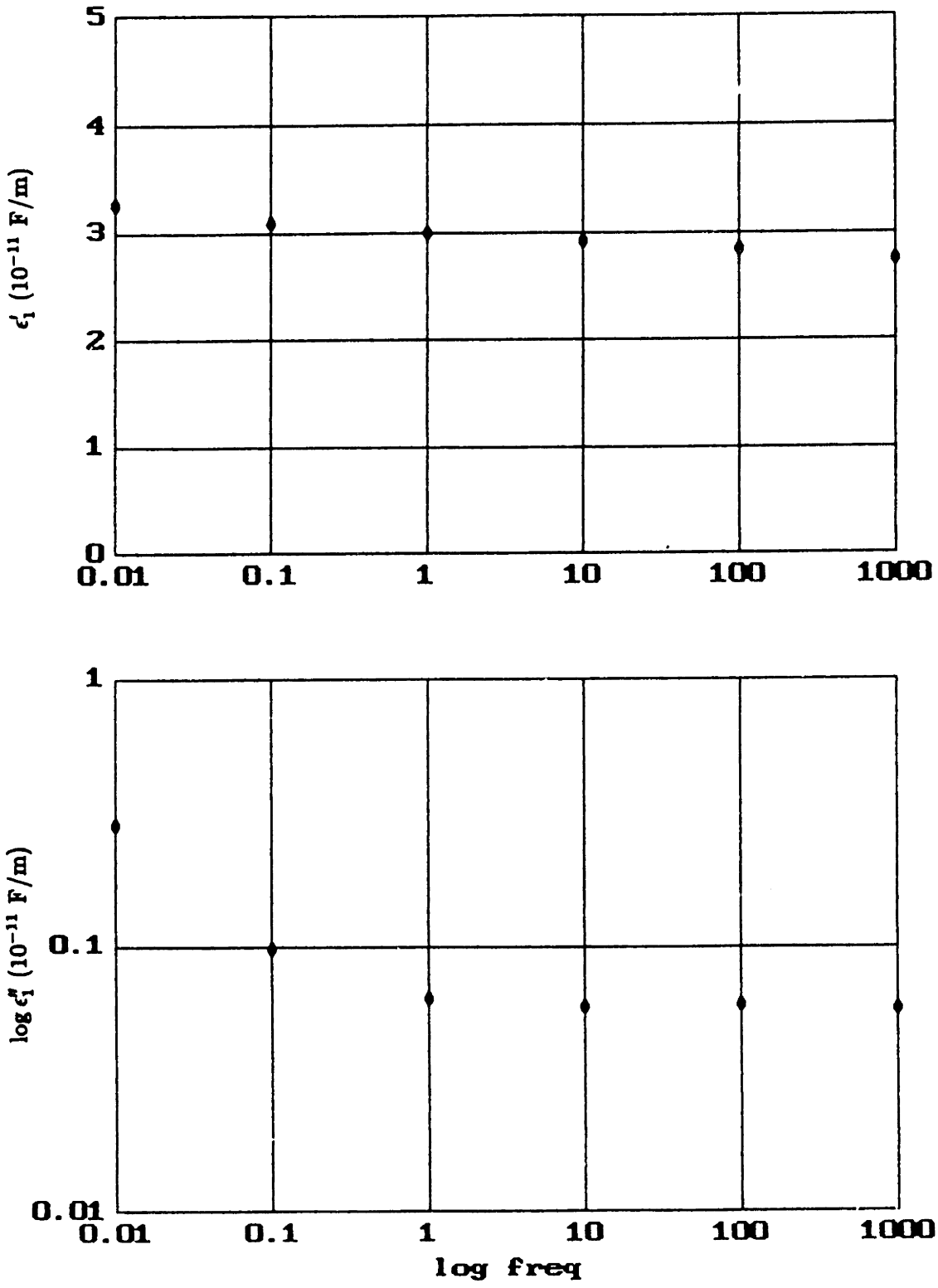


Figure 5.15: Estimated Dispersion in Complex Bulk Permittivity of Parylene From Measured Response of Fig. 5.13.

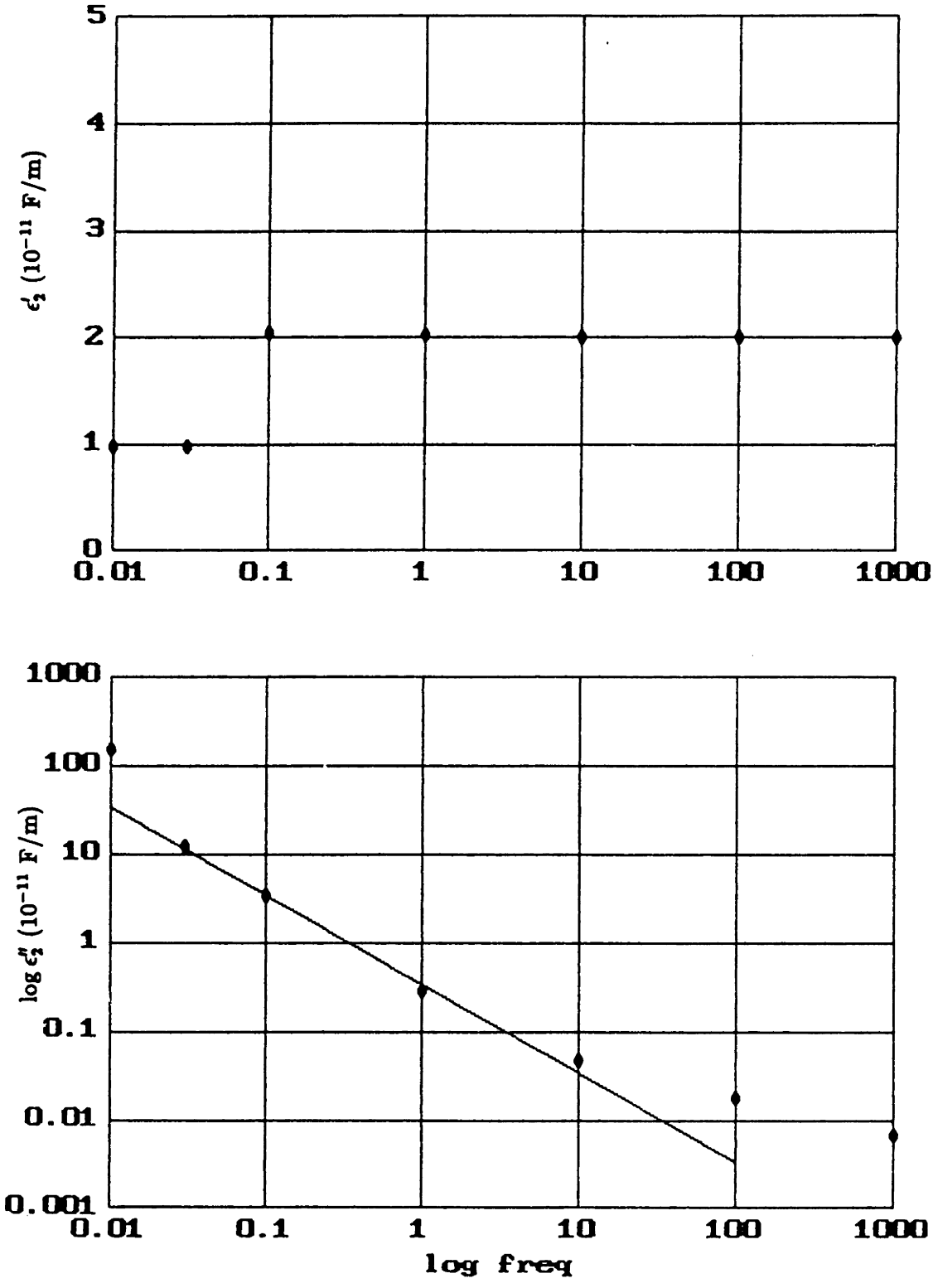


Figure 5.16: Estimated Dispersion in Complex Bulk Permittivity of Oil From Measured Response of Fig. 5.14 and using Estimated Dispersion of Parylene (Fig. 5.15).

from the microdielectrometer (Fig. 5.17). The estimated bulk permittivity using this model is $2.3x10^{-11}$ F/m, representing some average of the parylene layer and the oil. The straight line in Fig 5.17 represents the –1 slope that an ohmic material would display in this log–log plot. At 1 Hz, the estimated conductivity is $9.17x10^{-12}$ S/m, again representing some average value.

Problems with the domination of interfacial effects, at the coating–oil interface, can still arise at certain oil conditions. Typically, distorted frequency responses arise with high oil flows and decreased moisture content. An example of this is shown in Fig. 5.18 for a parylene coated microchip in stirred unused oil having a moisture content of 8 ppm. The occurrence of a large, negative DC offset is a good indicator of surface charging problems at the interface.

Although much of the problem can be avoided by performing measurements in unstirred oil, the best solution is to use a passivated macrochip. The advantage of the longer spatial wavelength is that more of the oil is sampled, reducing the significance of interfacial effects. These macrochips must also be encased in parylene as the alumina that forms the insulating oxide layer will absorb moisture and develop a measurable conductivity. The effect of the parylene coating on the response will also be small due to the longer spatial wavelength. These passivated macrochips then provide reasonable estimates of the oil conductivity, as shown by Li [23, Section 4.2]. Results of several oil measurements using passivated microchips and macrochips, and using a standard bridge, are given in Table 5.4. Generally, the conductivities were estimated using microchip and macrochip measurements in the range of 0.006–.2 Hz whereas the bridge measurements were made at 20–100 Hz. The estimated bulk conductivities using the passivated macrochip were obtained without including the dispersive nature of the parylene, perhaps accounting for some of the discrepancies. Also, the poor estimates of the bulk permittivity of the oil using the macrochip, due to its insensitivity, are not presented. For the passivated microchip, oil conductivity estimates obtained while accounting for the dispersion in the parylene, are presented. The estimated oil bulk permittivities, even using high frequency data, were 10–50% larger than values obtained by either an uncoated microchip or the bridge (the values given in Table 5.3).

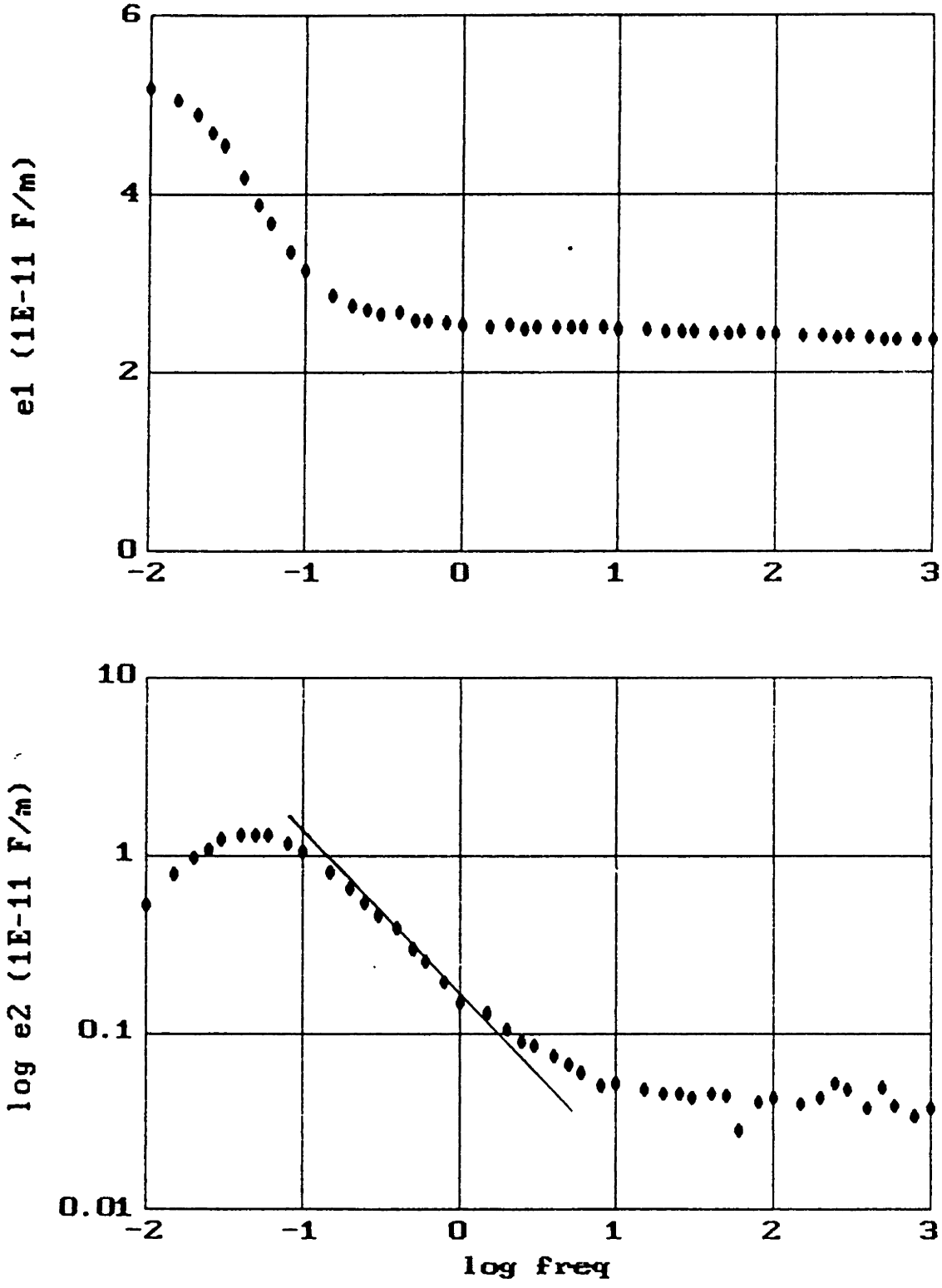


Figure 5.17: Estimated Dispersion in Complex Bulk Permittivity of Oil From Measured Response of Fig. 5.13 using Micromet Values.

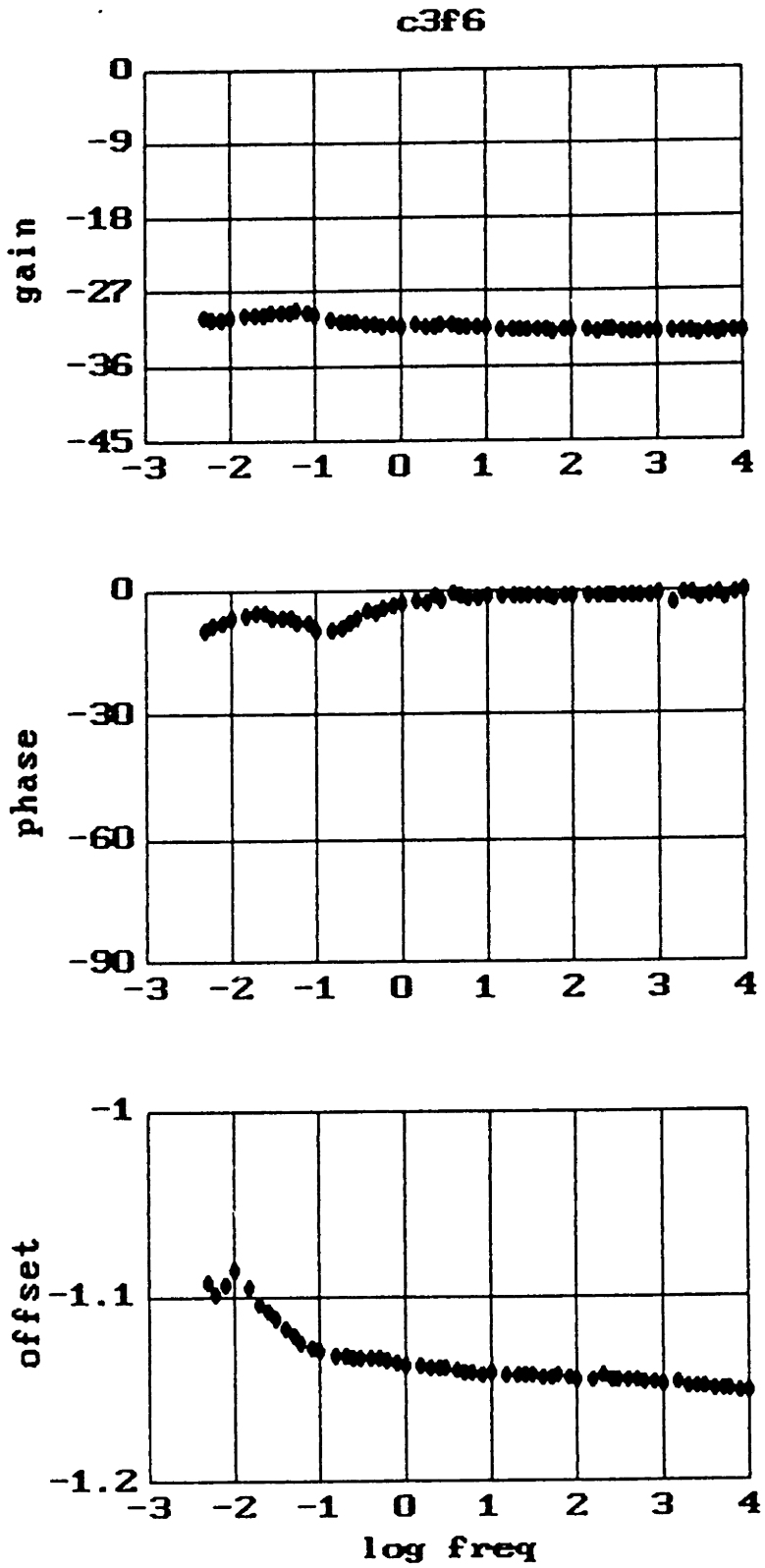


Figure 5.18: Measured Frequency Response for a Parylene Coated Microchip in Stirred, Unused Oil at 8 ppm Water.

Table 5.4: Bulk Conductivity Measurements With Passivated Electrodes.

Oil Condition	Conductivity (S/m)		
	Estimated		Measured
	Microchip	Macrochip	Bridge
unused	$4.52x10^{-13}$	$8x10^{-13}$	$2.3x10^{-13}$
"	$5.43x10^{-13}$	$4x10^{-13}$	$5.7x10^{-13}$
oxidized	$5.42x10^{-12}$	$5x10^{-12}$	$4x10^{-12}$
test facility	$1.94x10^{-11}$	$2x10^{-11}$	$2.55x10^{-11}$

The error here is due to DC offsets, on the order of -.7 V, that significantly affect the gain curve, and therefore, the estimated bulk permittivity, while not seriously changing the phase curve.

5.5 Complex Surface Conduction

Surface conduction, or conduction in regions so thin (relative to the spatial wavelength used for probing) that they can be modeled as surface conduction, has been observed in several instances. One example, presented in Fig. 5.7, represented surface conduction between electrodes along the silicon dioxide–oil interface arising from moisture adsorption and ensuing ionization of either the moisture or other adsorbed species. Using an ohmic conduction model, and representing the oil as a pure dielectric of permittivity $2x10^{-11}$ F/m, a surface conductivity of $2.52x10^{-14}$ S was estimated from the response. The predicted response using this ohmic model matches the measured response fairly well (solid line in Fig. 5.7).

Another example of surface conduction between the electrodes was observed when a thin film (<100 Å) of palladium, probably discontinuous, was vapor deposited onto the electrodes. In vacuum, the frequency response was that shown in Fig 5.19 (dots). Modeling the thin film as an ohmic material, a complex surface permittivity was estimated from the data at 1 Hz. The predicted response (solid line) using this complex surface permittivity ($\epsilon_s = 2.15x10^{-17}$ F and $\sigma_s = 1.12x10^{-15}$ S) matches the phase

peak fairly well but provides a poor match to the high frequency gain. The surface permittivity is contributing an unwanted capacitive component at high frequencies. Using only the surface conductivity and no surface permittivity, the predicted response (crosses) matches the measured high frequency gain but overshoots some on the phase, actually going positive at 1.5 Hz.

A second example of surface conduction due to a thin film coating, 350 Å of vapor deposited SnO_2, is shown in Fig. 5.20. The frequency response was measured while in room air at 35% RH. Again a complex surface permittivity was estimated, this time from data at 2.5 Hz. The predicted response (solid line) using this complex surface permittivity ($\epsilon_s = 4.06x10^{-17}$ F and $\sigma_s = 3.42x10^{-14}$ S) again matches the phase peak but not the high frequency gain. Using only the surface conductivity, the predicted response (crosses) not only misses on the phase but also does not match the high frequency gain. This indicates that either a surface permittivity is required at high frequencies or that the microchip is not correctly calibrated.

Assuming the microchip is correctly calibrated, the complex surface permittivity was estimated at several different frequencies (Fig. 5.21). It is useful to refer to a plot of the gain–phase space being searched in this case (Fig. 5.22). For the low surface permittivities of this example, there is a small window in the response within which accurate estimates can be made. Estimating any quantity at the overshoot, represented by the extreme kink in Fig 5.22 (in the neighborhood of -42 dB and $-15°$), is very unreliable. In fact, no estimates were able to be made at 80 and 100 Hz due to lack of convergence. Above 100 Hz, corresponding to phase between 0 and $-15°$, there is no sensitivity to surface permittivities less than 10^{-17} F. At the low frequencies, the dispersion below 0.5 Hz is artifactual as the estimated permittivity values reflect the fact that the gain does not actually go to unity at low frequencies. Due to possible mismatches in the on–chip transistors, the gain levels out at roughly –.2 dB. This mimics the effect of a low frequency capacitance or blocking layer, hence the estimated dispersion. This leaves a window from 0.5–100 Hz where reliable estimates of the complex surface permittivity may be made. In the estimated dispersion shown in Fig. 5.21, the log–log slope of ϵ_s'' is –.95, indicating that an ohmic model for the

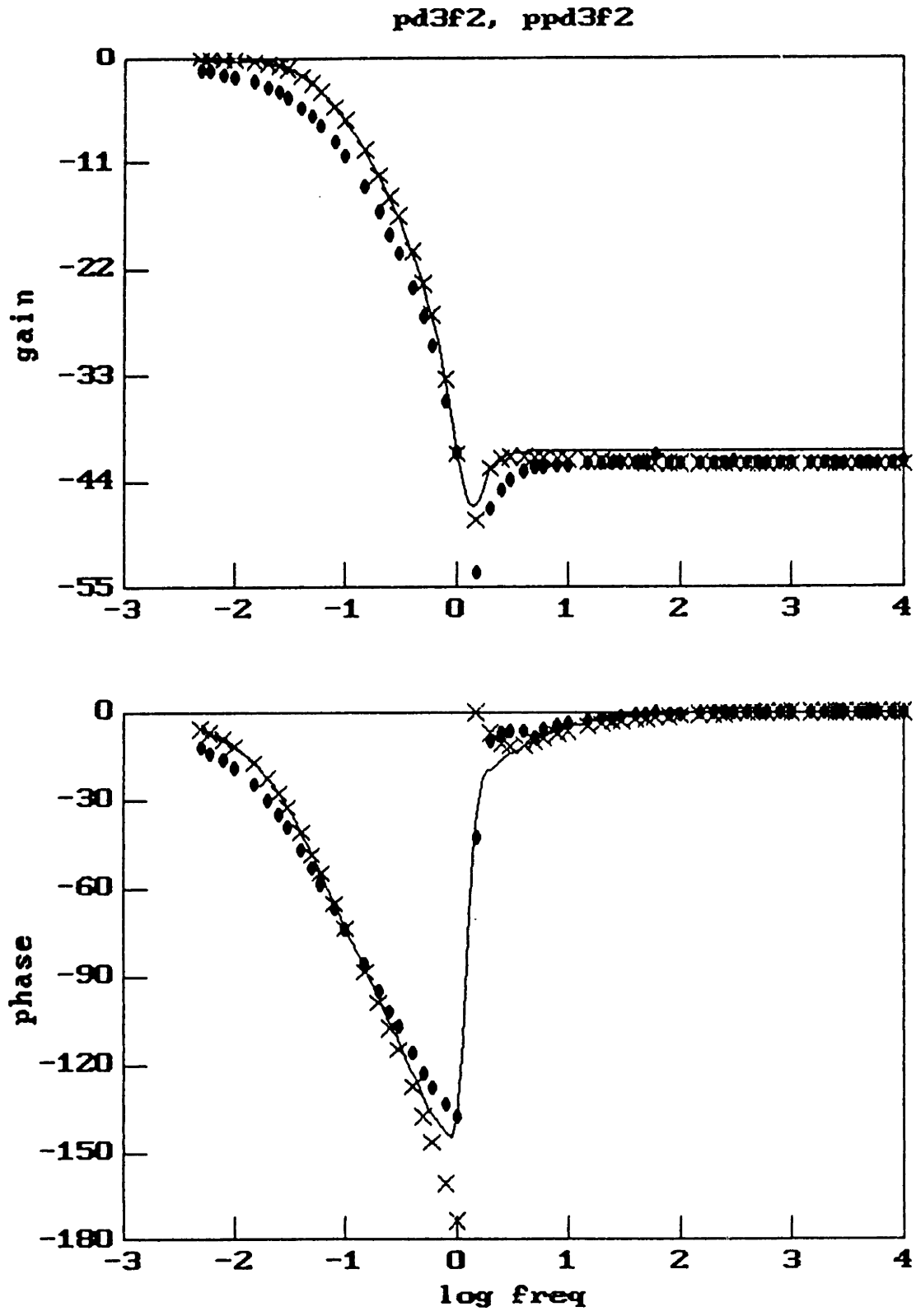


Figure 5.19: Measured (•) and Predicted Frequency Response for a 100 ÅThick Palladium Coated Microchip in Air ($\epsilon_{so} = 2.15x10^{-17}$ F and $\sigma_{so} = 1.12x10^{-15}$ S). For (–) use Complex Surface Permittivity, for (X) use only Surface Conductivity.

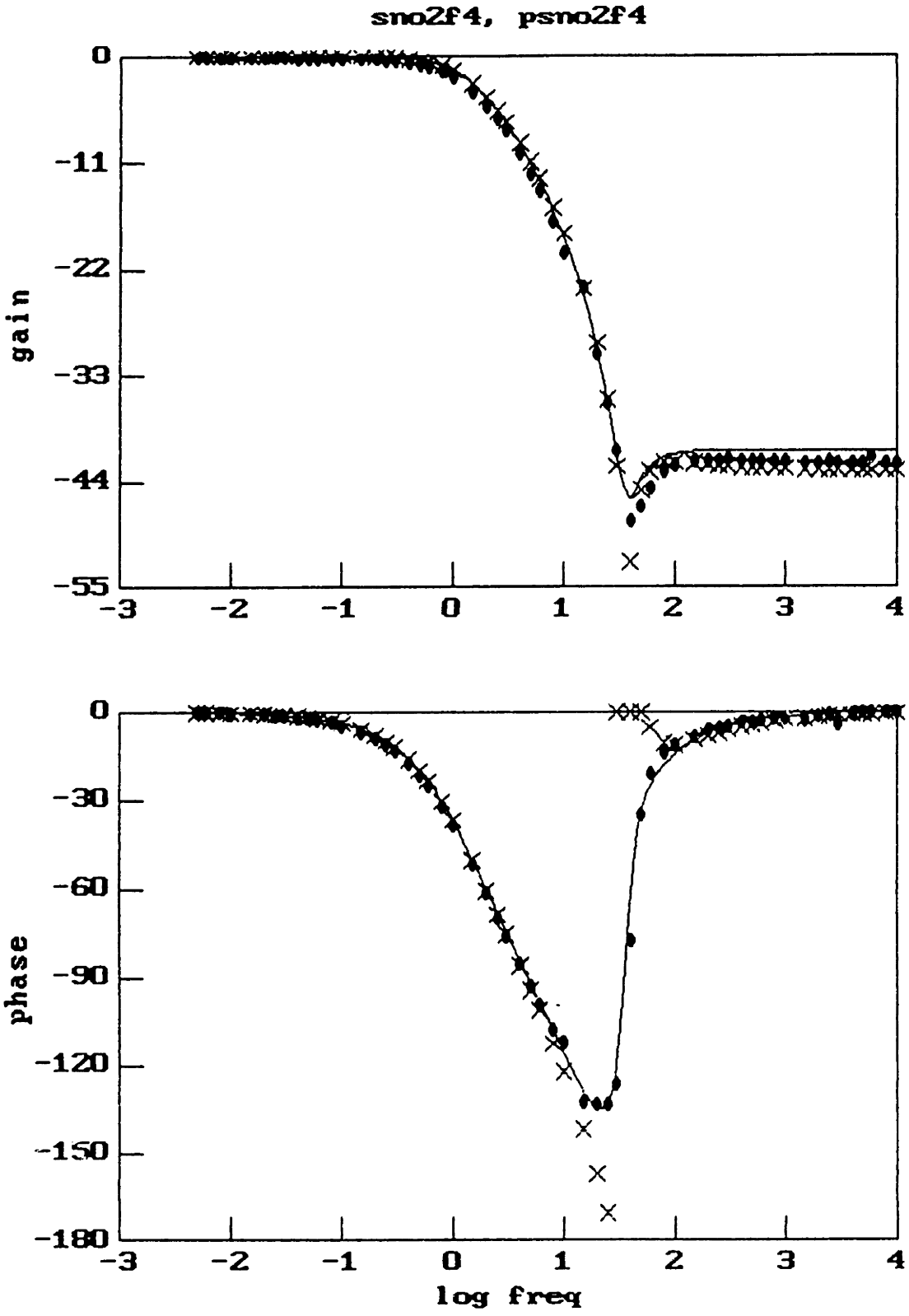


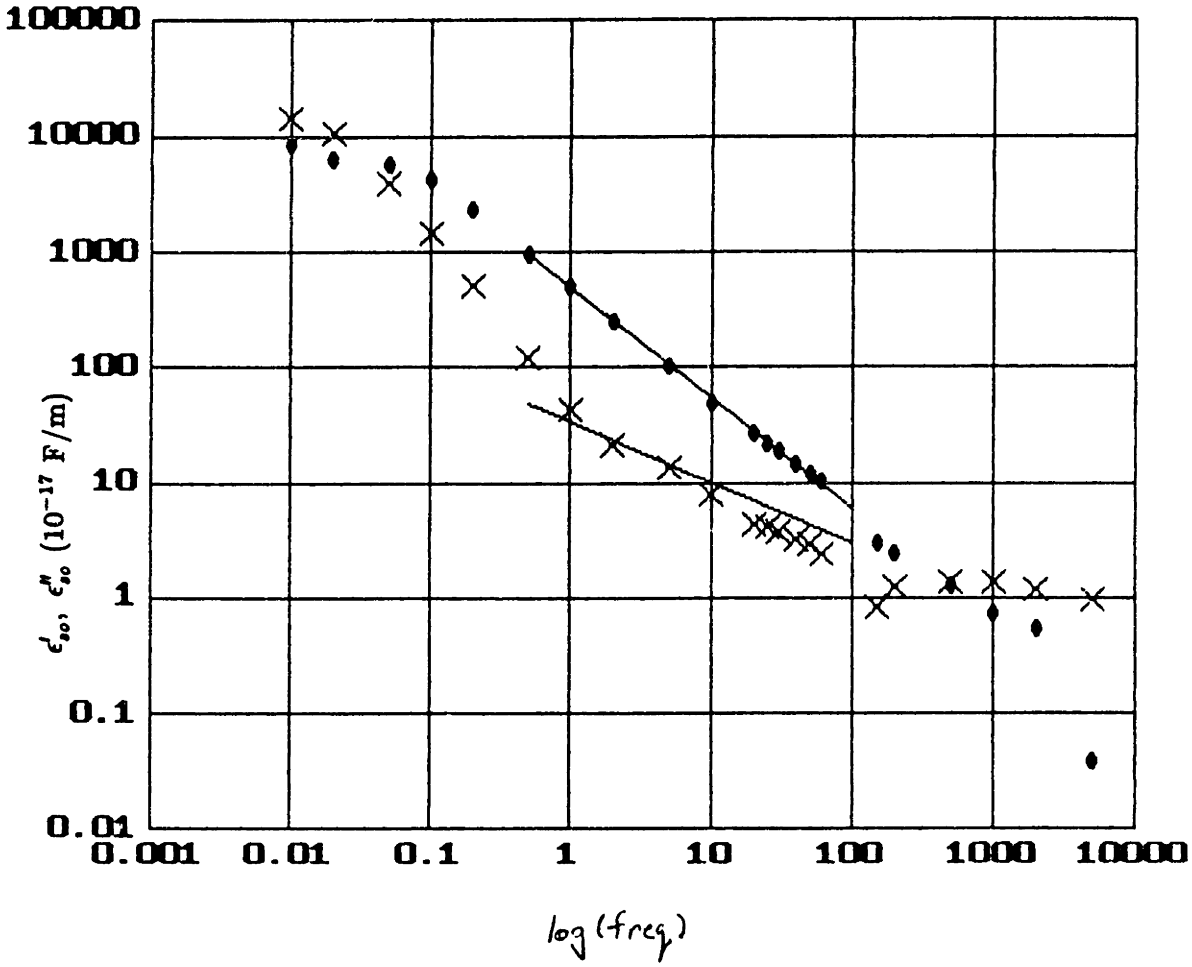
Figure 5.20: Measured (•) and Predicted Frequency Response for a 350 Å Thick Tin Oxide Coated Microchip in Air at 35%RH. ($\epsilon_{so} = 4.06x10^{-17}$ F and $\sigma_{so} = 3.42x10^{-14}$ S). For (–) use Complex Surface Permittivity, for (X) use only Surface Conductivity.

surface conduction might be valid. However, there is also a dispersion in the surface permittivity (log–log slope of –.52). This surface permittivity may indicate polarization occurring at blocking electrodes (similar to the effect seen with the thin film humidity sensors discussed in Section 3.3, case 4).

5.6 Estimation From the Frequency Response

An example is now given where a spatial distribution can be estimated from the frequency response, using the techniques demonstrated above. A plasma deposited bromobenzene coated microchip, Z24, was monitored in unstirred, oxidized oil that was heated to 99° C. The measured frequency response is shown in Fig. 5.23 (dots). The bromobenzene coating is thin enough, 2.4 μm, to allow the electric fields to sample the oil as well as the coating. There are clearly two relaxations occurring, one in the coating and one in the bulk of the oil. Using Theorem 3.2, each relaxation can be matched with the correct medium. Notice that between the relaxations there is a plateau in gain and the phase goes back towards zero, similar to the example given in Section 3.2.2 and shown in Fig. 3.10. If the higher frequency relaxation was occurring in the coating, then it would be expected that the gain would go to unity (0 dB) at frequencies below this relaxation as the electrodes would be intimately coupled. However, there is a plateau in the gain at roughly –18 dB, indicating that the coating is still relatively insulating at these frequencies. Thus, it is the oil that is relaxing at the higher frequencies and the coating that relaxes at the lower frequencies. The oil is highly conducting oil for two reasons: 1) the oil has been significantly degraded by oxidation (heating in room air at 90°C for a week) and 2) the oil conductivity varies inversely with viscosity, thus increasing at higher temperatures. The reason for the insulating bromobenzene coating is that a sample of heated oil, exposed to the atmosphere, will be dried out and have a very low moisture content.

The coating permittivity and thickness have been previously estimated in Section 5.3. The objective here is to obtain the complex bulk permittivities of the oil and the coating. For simplicity, it will be assumed that both the oil and the bromobenzene



Best Fit Lines

$$\log(\epsilon'_{so}) = -.52 \log f \ - 15.481$$

$$\log(\epsilon''_{so}) = -.95 \log f \ - 14.315$$

Figure 5.21: Estimated Dispersion in Complex Surface Permittivity using Measured Response of Fig. 5.20. (X) – ϵ'_{so}, (•) – ϵ''_{so}.

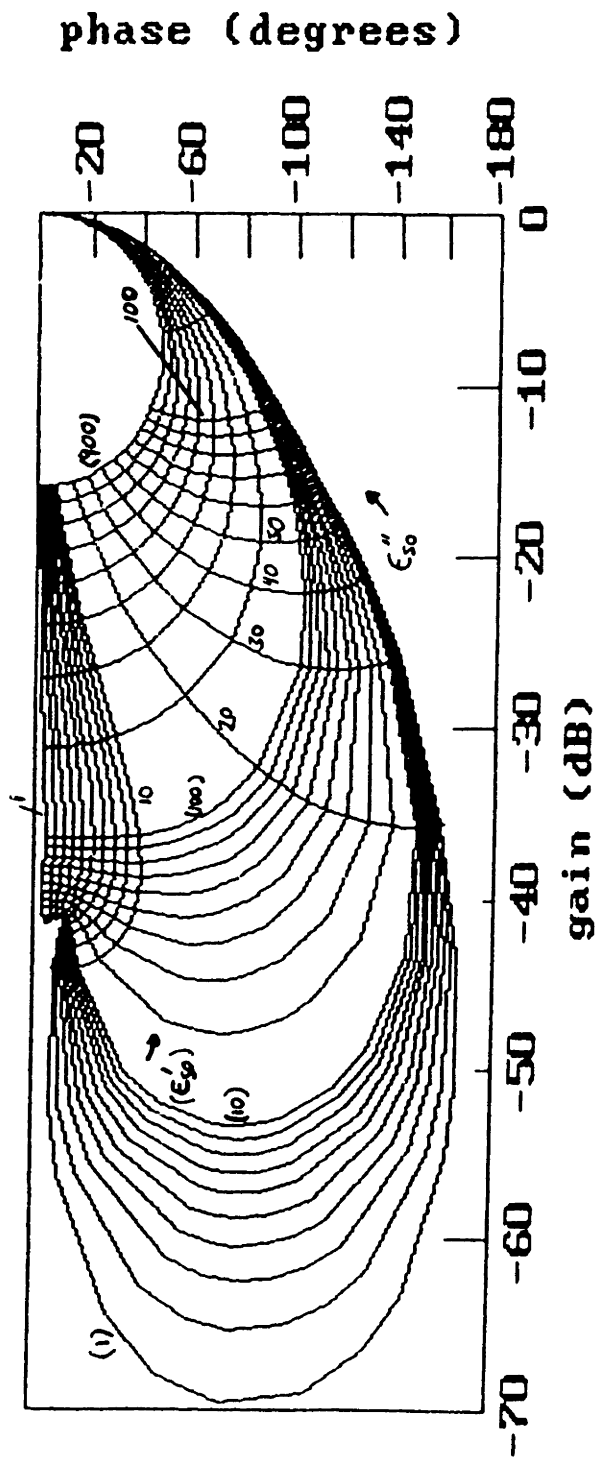


Figure 5.22: Parameter Space for Complex Surface Permittivity Estimation of a Microchip in Air with ϵ'_{so} in units of 10^{-18} F and ϵ''_{so} in units of 10^{-17} F.

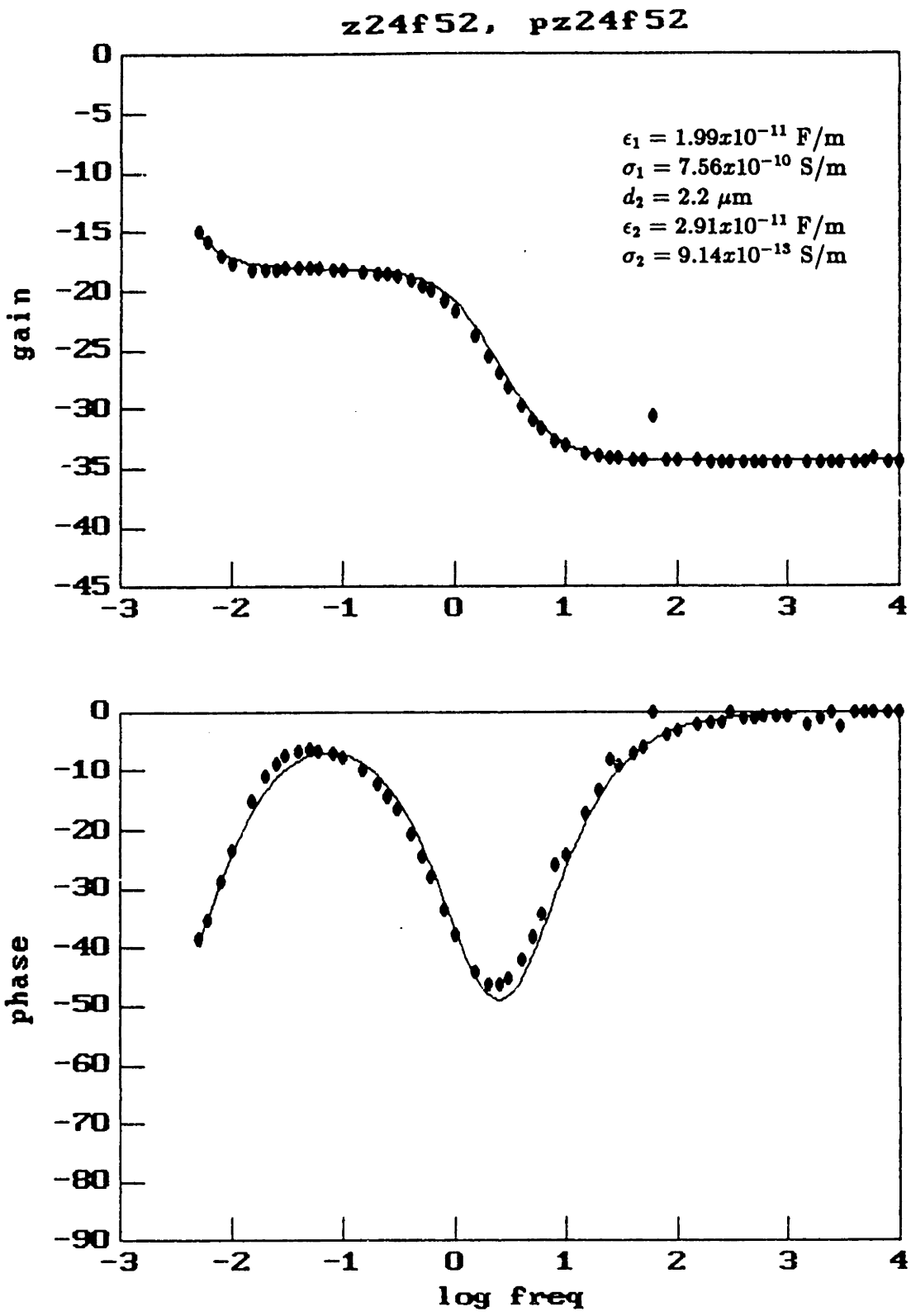


Figure 5.23: Measured (•) and Predicted (–) Frequency Response of 2.2 μm Thick PP–Bromobenzene Coated Microchip in Unstirred, Oxidized Oil at 99°C.

coating are ohmic conductors. It will be shown in Section 8.2.3 that the bromobenzene coating is actually dispersive. According to Theorem 3.2, both of these complex bulk permittivities can be estimated from the frequency response. At frequencies higher than 40 Hz, the contribution of the coating to the response is purely capacitive and depends only on the previously estimated coating thickness and permittivity. Using these estimated values, the complex bulk permittivity of the oil can be estimated from a high frequency datapoint, here the response at 50 Hz was used. Now, given the estimated complex bulk permittivity of the oil and the estimated coating thickness, an estimate can be made of the complex bulk permittivity of the coating from a low frequency datapoint, here the response at 0.006 Hz was used. The estimated parameters predict a frequency response that matches the measured response quite well (solid line in Fig. 5.23). Notice that the estimated bulk permittivity for the coating, obtained from the low frequency data in Fig. 5.23, matches the value obtained from the two parameter estimation problem treated in Section 5.3.

Another example of estimating spatial distributions from the frequency response is a measurement made of paper insulation using a passivated macrochip. This example illustrates the estimation of both a spatial distribution and a dispersion in complex bulk permittivity. The experiment, as performed and analyzed by Li [23, Section 5.2.2], involved spring loading a passivated macrochip against an 84 μm thick piece of paper insulation and immersing the entire setup in transformer oil. The moisture content of the oil was varied, significantly altering the paper moisture content due to the small volume ratio of paper to oil. A typical response is shown in Fig. 5.24. As the moisture content of the oil was decreased, the relaxation at higher frequencies would shift down in frequency while retaining the same shape. This relaxation was attributed to the paper insulation. The lower frequency relaxation remained constant during these changes and, in view of the insulating nature of the parylene, could not be attributed to the parylene. In addition, the plateau in gain in the neighborhood of 1 Hz was lower than would be expected if attributed entirely to the capacitance of the parylene coating. Thus, it was deduced that the paper was not in intimate contact with the macrochip and that a layer of oil existed between the parylene coating and

the paper. Given the permittivity of the oil, available from other measurements, and the properties of the parylene, the thickness of the oil layer was estimated. From low frequency measurements, where the paper was treated as a perfect conductor, the complex bulk permittivity of the oil was estimated. At high frequencies, where the oil was a perfect insulator, the complex bulk permittivity of the paper was estimated.

It was observed that the relaxation in the paper was not ohmic but, in fact, was dispersive. The estimated dispersion, using the known parameters for describing the oil and the parylene, is shown in Fig. 5.25. The real part of the complex permittivity (dots) has the static, nondispersive dielectric constant ($1.6x10^{-11}$ F/m) subtracted out and the imaginary part (crosses) has the nondispersive, dc conductivity ($2.3x10^{-11}$ S/m) subtracted out. This dispersion is best represented by a power law dependence of the general form

$$\epsilon^* = B(j\omega)^{n-1} \tag{5.1}$$

This type of of dispersion is discussed in greater detail in Section 8.2. Pertinent here is the fact that this dispersion was estimated from the frequency response of a multilayered structure.

The layer thickness and complex bulk and surface permittivity estimation routines of Sections 4.2 and 4.4 have been methodically applied to various experimental observations in this chapter. A rough mapping of estimation routine to experiment is given: 1) layer thickness – Sections 4.2.1 and 5.2.1, 2) complex bulk permittivity – Sections 4.2.2 and 5.4, 3) complex surface permittivity – Sections 4.2.2 and 5.5, and 4) layer thickness and permittivity – Sections 4.4 and 5.3. In addition, experiments requiring the use of a combination of two or more of the estimation routines were given in Section 5.6. A significant piece of information not given with the estimated parameters is the accuracy of the estimates. Although some feeling for the sensitivity of the estimation routines is given in the appropriate sections in Chapter 4, no variances were actually computed.

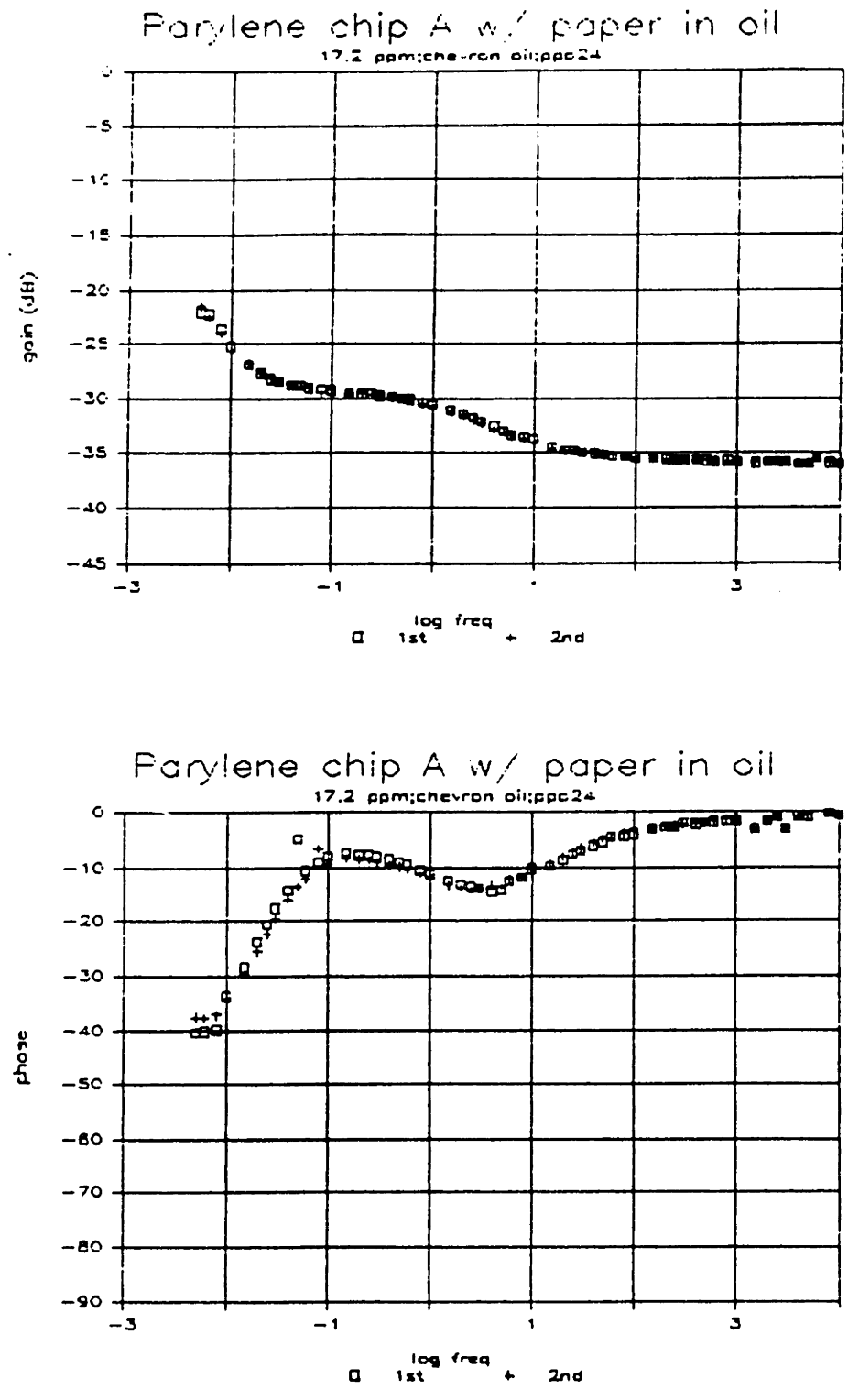


Figure 5.24: Measured Frequency Response of Parylene Coated Macrochip with Paper in Unstirred, Unused Oil at 28 ppm Water (from [23, p. 85]).

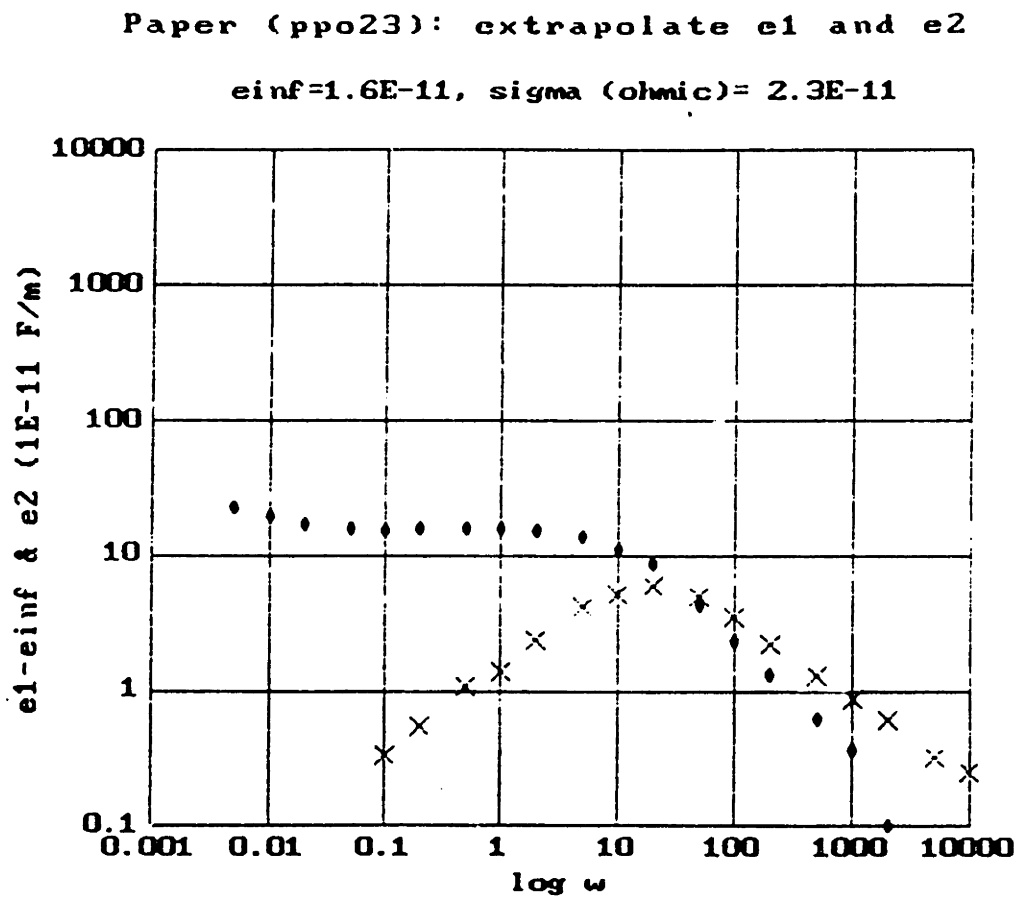


Figure 5.25: Estimated Dispersion in Paper using Measured Response of Fig. 5.24. (•) – ϵ', (X) – ϵ'' (from [23, p. 88]).

Chapter 6

Moisture Sensing – Concepts and Design Strategy

At the outset, it is important to get a good understanding of the various terms used in quantifying the amount of moisture in the gas, liquid and solid phases present in a transformer. Section 6.1 provides definitions for these terms and describes the many different equilibrium situations encountered. These concepts will be applied towards understanding how a moisture sensor would respond to moisture in the oil and providing a framework for relating on–line moisture measurements to parameters of direct interest in a transformer, such as water solubility in oil and paper moisture content. An example of using this framework to interpret actual data, obtained from both a test facility and an operating power transformer, is presented in Section 6.2. After demonstrating the usefulness of moisture monitoring, the design strategy for developing a moisture sensor is presented in Section 6.3. In this context, the rationale for coating the microchips using a plasma deposition process is also presented.

6.1 Equilibrium Concepts

There are two existing, commercially available methods for measuring moisture in oil. As described in Section 1.3.1, the Karl Fischer method measures *absolute moisture content* whereas the dew point hygrometer measures *relative saturation.* A pertinent

question arises: what information about the state of the transformer can be deduced from the relative saturation and the absolute moisture content?

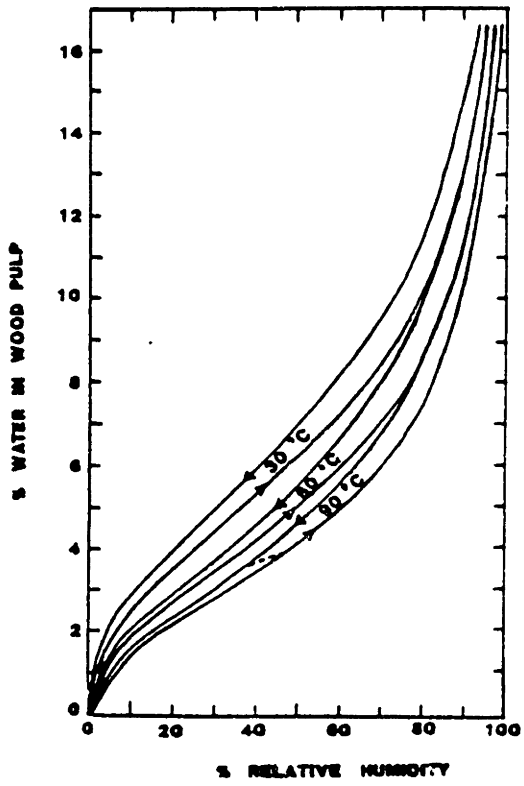
Answering this question will require a reexamination of the various equilibria in a system containing gas, oil and paper. There are gas–liquid (air–oil), gas–solid (air–paper) and liquid–solid (oil–paper) interactions that must be understood, at least in the case of equilibrium. Before that, some commonly used terms are defined. Relative humidity (RH), of a gas, is the ratio of the partial pressure of the water vapor to the vapor pressure of pure water at the same temperature [86, p. 279]. The solubility limit is the maximum amount of moisture held by the gas at 100% RH and a given temperature, corresponding to complete saturation. The relative saturation, of a liquid, is defined as the mass of moisture dissolved in a liquid divided by the maximum amount of moisture that could be dissolved at a given temperature. The solubility limit in a liquid is again the maximum amount of moisture held by the liquid, now at 100% relative saturation. Beyond this point, additional water will precipitate out, forming free water. This is true for both the liquid and gas phase. Moisture content, or absolute moisture content, when referring to a liquid, is the amount of water contained in terms of parts per million (ppm) on a mass basis. This equals the product of the relative saturation and the solubility limit.

The simplest situation is the gas–liquid equilibrium. Raoult's Law states that, at a given temperature, the vapor pressure of the solvent over a solution equals the vapor pressure of the pure solvent multiplied by the mole fraction of the solvent in the solution. At equilibrium, using Raoult's Law for an aqueous solution, the RH is equal to the mole fraction. [86, p. 279]. It has been shown that the solubility of moisture in oil is linearly proportional to the relative humidity [87]. *Thus, in equilibrium, the RH of the gas and the relative saturation of the oil must be equal.* If the volume of oil is small compared to that of the gas (at standard temperature and pressure) then the RH of the gas will drive the oil to the appropriate relative saturation. By small volume, it is meant that the amount of moisture exchanged between the gas and the oil does not significantly affect the RH of the gas. If the volume of oil is large compared to that of the gas (again at STP), then the relative saturation of the oil will drive the gas

to the appropriate RH. Now it is the relative saturation of the oil that should remain constant with the exchange of moisture. Thus, the requirement of a volume ratio of roughly 1000:1 for the dew point hygrometer.

For the gas–solid equilibrium the picture is not so simple. The paper insulation contains moisture in several different ways. There is the 'water of constitution', an integral part of the cellulose fiber structure. In addition, there is: 1) chemically absorbed water, 2) physically absorbed water and 3) capillary held water [88, p. 309]. Given this variety, all methods used to measure the moisture content of paper are calibrated against the definitive measurement, which is a gravimetric one. Paper is conditioned at a specific RH and temperature, weighed, dried in an oven at 105°C and reweighed. The resultant percent change in weight is defined as 'percent water, dry basis' or 'percent moisture content', depending on whether the change is taken from the dry or wet paper weight. Early data on sorption for wood pulp obtained by Jeffries [89] was plotted by Oomen [14] (see Fig. 6.1a). Notice the hysteresis in the curves due to the different ways in which moisture is held in the paper. In general, the adsorption curves have a sigmoidal shape. As the gas temperature is raised, at constant RH, the paper moisture content decreases (Fig. 6.1b). At higher temperatures, the gas has a greater affinity for moisture. Note that when conditioning the paper, the volume of gas is large enough so that the amount of moisture exchanged with the paper does not significantly alter the state of the gas. *Thus, in equilibrium, the absolute moisture content in paper is directly related to the gas RH and temperature.*

Given that the RH of the gas is in equilibrium with, in one case, the relative saturation of the oil, and in the other case, the moisture content of the paper, it is reasonable to draw the conclusion, barring unforeseen oil–paper interactions, that the relative saturation of the oil is related to the moisture content of the paper. Thus, the x–axis of Fig. 6.1b can be replaced with the relative saturation of the oil. Using experimentally measured solubility limits of a naphthenic oil, at various temperatures, and the fact that the moisture content in oil is linearly proportional to the relative saturation, Oomen further transforms the x–axis from relative saturation to moisture content (Fig 6.1c) [14]. An attempt to experimentally reproduce these curves would



(a)

Figure 2. Sorption Curves for Wood Pulp from Jeffries' Data, Table I

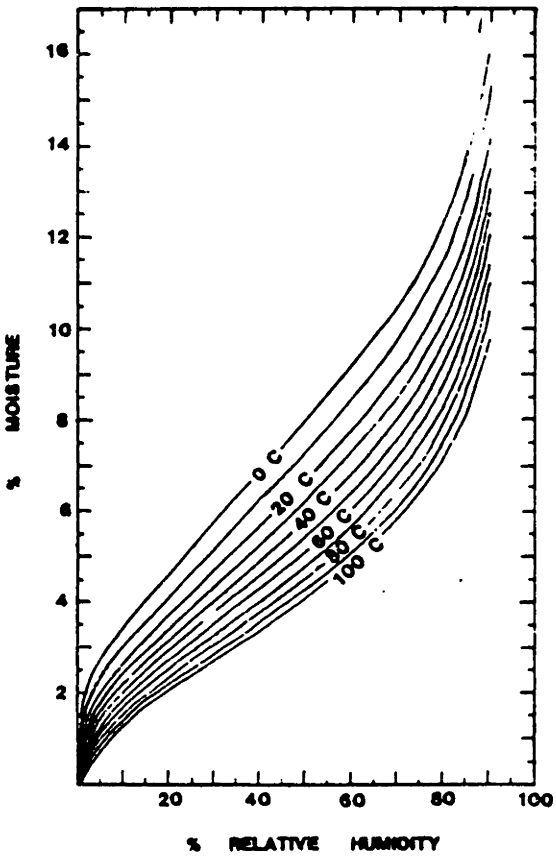


Figure 4. Adsorption Curves for Wood Pulp Based on Figure 3.

(b)

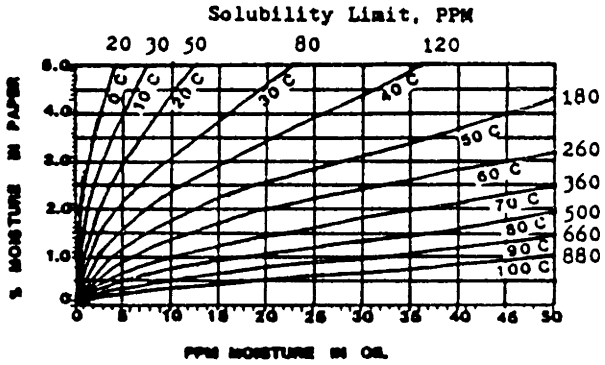


Figure 6. Moisture Equilibrium Curves for Paper-Oil System, Low Moisture Region

(c)

Figure 6.1: Moisture a) Sorption and b) Adsorption Curves for Wood Pulp and c) Moisture Equilibrium Curves for Paper-Oil System (from [14]).

be very difficult to do using the gravimetric method, though other techniques for observing changes in paper moisture content, such as electrical measurements, may be more feasible. *The conclusion drawn by Oomen is that there is a direct relationship between the moisture content of the paper and the combination of the temperature and the absolute moisture content of the oil.* Again, this assumes that the solubility limit is a known function of temperature.

There is a problem with this conclusion, however. As shown by Stannett [16], there can be large changes in solubility with degradation. What happens when the solubility limit of the oil changes due to degradation? Now solubility is a function of both temperature and aging. It is claimed here that Fig. 6.1c is still valid, except for the temperature markings. In other words, given the moisture content and solubility limit of the oil (at the same, but unknown, temperature), the paper moisture content can be determined. For example, oil sample 1, containing 5 ppm moisture and having a solubility limit of 50 ppm, will be in equilibrium with paper having a moisture content of 3% . Oil sample 2, containing 10 ppm moisture and having a solubility limit of 80 ppm, will be in equilibrium with paper also having a moisture content of 3% . The samples could represent oil at different temperatures not having suffered any degradation. According to Fig. 6.1c sample 1 would be at 20°C while sample 2 would be at 30°C. However, it could also be that both samples were at the same temperature but the oil has been significantly degraded so as to increase its solubility limit at a given temperature. Either way, the curves of Fig. 6.1c represent the paper-oil equilibrium. *Thus, the moisture content of the paper is determined by the solubility limit of the oil and either the relative saturation or the absolute moisture content of the oil.*

At any instant in time, if the relation between temperature and solubility is known, then knowing the temperature is as good as knowing the solubility limit. This leads one back to the conclusion of Oomen. Further, if this relation was known at several instants in time, then the evolution of the relation gives some indication of the aging of the oil. In a transformer, the paper drives the equilibrium, as it holds much more moisture than the oil could ever dissolve. From the above conclusion, for a given paper

moisture content, the oil with a higher solubility at temperature T_o, will be at a higher relative saturation (and higher absolute moisture content).

There is a major assumption in drawing the above conclusion. The assumption is that the influence of temperature on the curves of Fig. 6.1b is solely through the different solubility limit associated with each isotherm. Paper holds moisture through a variety of physical and chemical mechanisms. If the contribution to the moisture content due to physical adsorption is not a function of the solubility in the vapor, but is a function of temperature, and if this contribution is significant, then both the temperature and the solubility are relevant parameters. A way to resolve this question would be to measure the adsorption isotherms, varying temperature and RH, in air and in another atmosphere that has a significantly different solubility dependence on temperature. If the assumption is valid, then the second set of isotherms will be the same family as the first set, with a different temperature associated with each individual isotherm. If the assumption is invalid, then the temperature influences the equilibrium in ways other than just through changes in the solubility limit. This means that Fig. 6.1c is only valid as long as the solubility of the oil does not change. As soon as degradation occurs, the adsorption isotherms used to construct Fig. 6.1c are incorrect and no estimate of the paper moisture content can be made, even when using the temperature, moisture content and solubility of the oil. As a practical matter, it may be useful to treat the assumption as being reasonable, as the problem of hysteresis has not been addressed at all and the required accuracy of the paper moisture content is not very great.

Finally, the answer to the question posed must take into account the relation between moisture in the oil and problems associated with excess moisture – electrically weakened paper insulation, degraded oil and formation of free water. The physical parameters associated with these problems are the moisture content of the paper insulation for the former problem and the solubility limit of the oil for the latter two. Relative saturation can indicate the potential for free water formation. However, it does not provide a measure of the solubility limit of the oil. Thus, it cannot indicate oil degradation (oxidation), a phenomenon accompanied by increased acidity and interfa-

cial tension. The transformer decay products, such as sludge, are hygroscopic and will cause an increase in the solubility of the oil at a given temperature [90]. The previous discussion has shown that unless the oil solubility remains constant (no degradation), the relative saturation does not provide enough information for a determination of the moisture content of the paper. However, a combination of relative saturation and absolute moisture content will provide such information.

6.2 Moisture Distribution In A Transformer (Status of Oil and Paper Insulation – An Example

As concluded in Section 6.1, both the absolute moisture content and the relative saturation of the oil are required to determine the status of both the oil degradation (via changes in the moisture solubility) and the moisture content of the paper insulation. Using a moisture sensor of the type described in Chapter 7, an on–line means for continuously monitoring the absolute moisture content is possible. However, the method for measuring relative saturation is a slow process due to the need to establish an equilibrium between the gas and liquid phases.

An alternative approach is to use the information obtained during the regular load cycling of the transformer. Variations in the load cause changes in the operating temperature. Of particular importance are the changes in the winding temperature as it is this temperature that drives the exchange of moisture between the paper and the oil. If the thermal changes are slow compared to the kinetics of moisture absorption by the oil, a quasi–equilibrium analysis can be made of the moisture distribution between the paper and the oil. It will be shown that the parameters describing the condition of the paper and the oil can be extracted using this analysis. Experimental data from both a test transformer and a 784 MVA operating transformer will then be interpreted using this analysis.

6.2.1 Model

A Freundlich isotherm is used to represent the low and intermediate moisture ranges of the adsorption curves of Fig. 6.1b [14]. Normally the moisture content of the paper is constant as the paper is essentially an infinite reservoir of moisture for the oil. Therefore, the relationship is written as

$$\mathrm{RS}_{oil} = g(T)\ (\%\mathrm{mc}_{paper})^{n(T)} \tag{6.1}$$

where $g(T)$ and $n(T)$ incorporate the temperature dependence of the adsorption curves of Fig. 6.1b, $\%\mathrm{mc}_{paper}$ is the percentage moisture content of the paper, and the RH has been replaced by the relative saturation. By definition, the relative saturation is also related to the solubility limit (S)

$$\mathrm{RS}_{oil} = \frac{\mathrm{mc}_{oil}}{\mathrm{S}(T)} \tag{6.2}$$

Combining (6.1) and (6.2) and solving for the moisture content of the oil yields

$$\mathrm{mc}_{oil} = \mathrm{S}(T)\ g(T)\ (\%\mathrm{mc}_{paper})^{n(T)} \tag{6.3}$$

From experimental data, Oomen [14] has characterized the dependence of the solubility limit as

$$\log \mathrm{S} = K_1 - \frac{K_2}{T} \tag{6.4}$$

As the oil temperature increases, the oil can hold more moisture. Taking the logarithm of (6.3) and substituting in (6.4) yields

$$\log(\mathrm{mc}_{oil}) = K_1 - \frac{K_2}{T} + \log g(T) + n(T)\log(\%\mathrm{mc}_{paper}) \tag{6.5}$$

This equation relates the temperature dependence of the absolute moisture content to parameters of direct interest. K_1 and K_2 describe the solubility of the oil and will be a function of oil degradation. The effect of paper moisture content is also very evident. Provided the temporal variations are slow enough to insure equilibrium

between the moisture in the paper and in the oil, this functional dependence can be obtained experimentally through the load cycling of the transformer, eliminating the need for a measurement of the relative saturation. Changes in intercept and/or slope over time can be interpreted as changes in the fundamental parameters of the paper insulation and/or the solubility of the oil.

The temperature required has deliberately been left ambiguous. It is clear that the temperature of the windings is important as it reflects the heat driving moisture out of the paper insulation. The ambient oil temperature is also important because it determines the solubility of the oil. Ultimately, one or more temperatures may be required for this type of analysis. For the present purpose of illustration, the ambient oil temperature will be used.

6.2.2 Experimental Evidence

A fully instrumented 50 kVA pole–type distribution transformer was used for the experiments. Loading was provided by an identical transformer. An oil circulation system was installed to simulate the oil flow present in large power transformers. Seven thermocouples were mounted at various locations inside the transformer. A moisture sensor was mounted in the oil flow inlet, directly exposed to the flow ($\sim$ 9 gal/min) but outside of the transformer tank.

Over a period of 7 months, oil samples were extracted from the test facility and the moisture content was measured using a moisture analyzer. This data is plotted versus inverse absolute temperature in the upper part of Fig. 6.2 along with a best fit line. The other set of data, the boxed symbols, were obtained from Allegheny Power System. They represent the moisture content of oil samples removed from a 784 MVA power transformer located in the Harrison Power Station. The sampling was done over a period of 2.5 years. The boxed plus signs represent results with one oil in the transformer, the boxed asterisk, stars and crosses are results with a second oil. The oil change was due to a large measured static charge. The boxed asterisk represents the status immediately after the oil change, and the boxed stars represent later data with this second oil. The boxed crosses represent still later results obtained

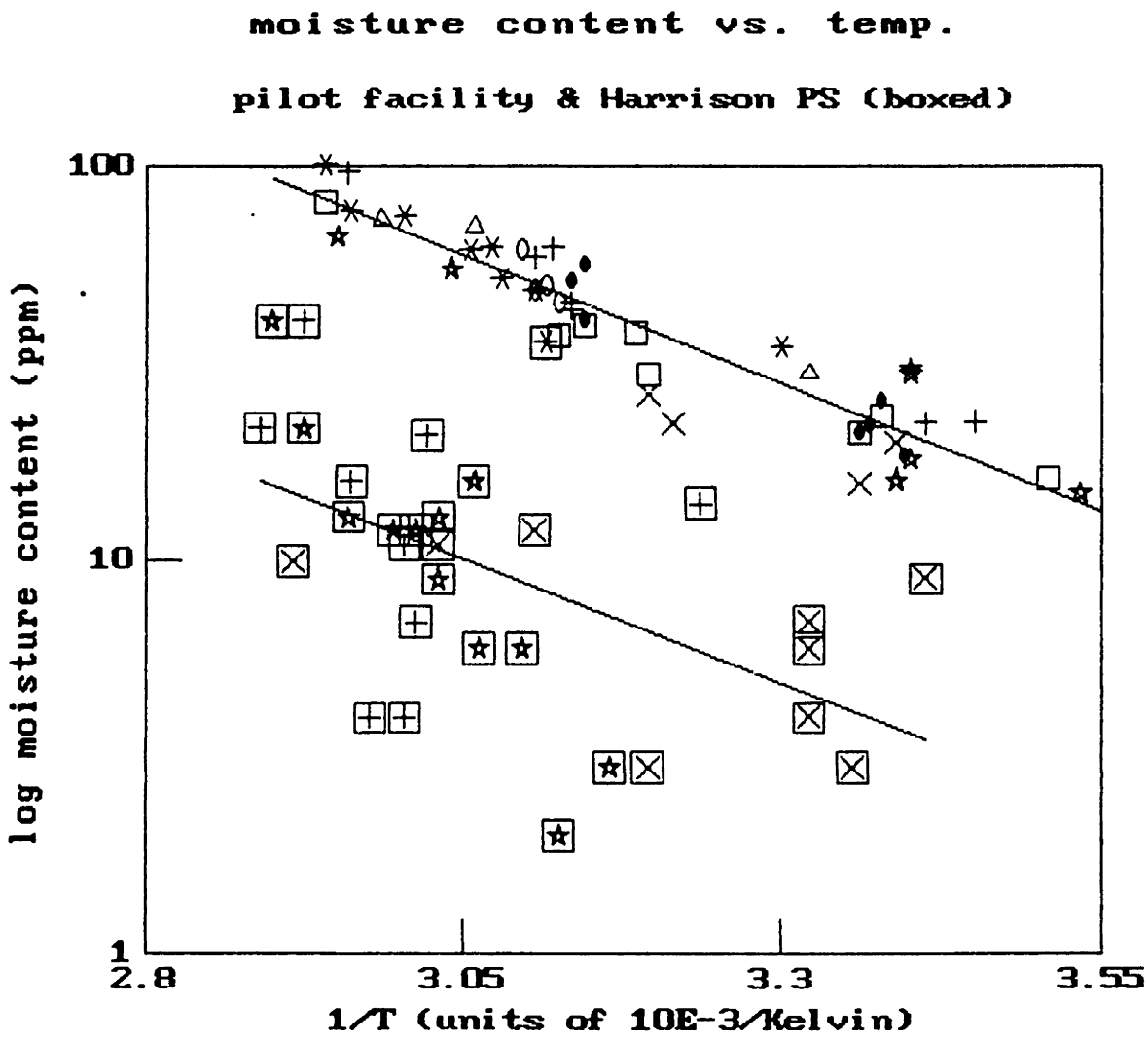
after the second oil was degassed. The oil temperature used probably represents the oil sample temperature when removed from the transformer. It is unknown whether the moisture distribution was in steady state within the transformer when the sampling was performed.

With these provisos in mind, a best fit line was also drawn for all the data for the Harrison facility. Although there was a lot more scatter in this data, a complete description of the moisture content versus temperature could be obtained only by using the complete data set. The slopes of the two lines are very close, −1303 versus −1261, perhaps reflecting the dominant influence of the temperature dependence of the moisture solubility in the oil. There is a significant difference in the intercepts, 5.752 versus 4.851. This clearly reflects the fact that the paper insulation in the test facility is much wetter than the operating power transformer. This is not surprising as no effort was made to dry out the paper insulation prior to energization of the test facility. Using (6.5) an estimate can be made of the difference in paper moisture content once the paper adsorption parameters have been sufficiently characterized.

6.3 Moisture Sensing in Oil – A Design Strategy

As shown in Chapter 5, moisture can alter the response of an uncoated microchip either by surface adsorption or double layer (charge separation) effects, both of which are not reproducible. Furthermore, it is known that the bulk conductivity of the oil changes only by small amounts (20-30%) with changes solely in moisture content. For example, used oil from a transformer exhibited a reduction in conductivity of 28% (from $1.04x10^{-11}$ to $7.41x10^{-12}$ S/m) as the moisture content was reduced from 48 to 10 ppm. However, with significant degradation of the oil, perhaps resulting in an increase in ionizable species, the effect of moisture can be significantly increased. Hence, changes in oil conductivity are not a good indication of moisture content.

The approach taken here is to deposit a thin film of material onto the microelectrode structure and monitor changes in its dielectric properties. There are several criteria that must be satisfied by the deposited material. First and foremost, it must interact



⊞ – Oil 1

⊡ – First Result with Oil 2

⊡ – Subsequent Results with Oil 2

⊠ – Degassed Oil 2

Best Fit Lines

Pilot: $\log(mc) = 5.752 - \frac{1303}{T}$

Harrison: $\log(mc) = 4.851 - \frac{1261}{T}$

Figure 6.2: Moisture Content Versus Inverse Absolute Temperature for Pilot Facility and Harrison Power Station.

with the moisture in the oil in such a way as to translate moisture content variations to changes in its dielectric properties. Second, the sensitivity to moisture content should be reproducible and selective, that is, independent of other chemical changes occurring in the oil. Third, the coating should be able to withstand exposure to hot oil over long periods of time and retain its physical integrity and moisture sensitivity. Fourth, the coating should move the region of interest, i.e. where the conduction process takes place, away from the oxide layer surface.

There are some additional constraints imposed by the availability of fabrication resources and the commercially available form of the microchips. Polyimide is used at some stage of the microfabrication process and the microchips are encapsulated in a Kapton package, limiting the temperature range of the device to roughly 200°C. This places a limitation on the types of deposition processes that can be considered. Coating of the microchips before packaging would necessitate additional fabrication steps, including masking to protect leads. Sputtering processes can be performed at low temperatures but are expensive and are generally used to deposit submicron thickness films.

Plasma deposition is a low (ambient) temperature coating process that can produce highly crosslinked polymer films of micron or submicron thickness. A description of the plasma deposition process, the apparatus used and the operating experience for coating the microchips can be found in Appendix C. A deposition apparatus, having a capacitively coupled RF plasma design, was available on a full–time basis. Operating parameters for producing good quality films had been previously established for several materials, including bromobenzene, by Benatar [91, p. 68]. Satisfying all the constraints imposed by the microchip packaging, this process was used to deposit bromobenzene for the purpose of developing a moisture sensor. The relevant, and measurable, process parameters for the coated chips discussed are given in Table 6.1. Bromobenzene coated chips are identified by Z# where # denotes the coating run number. Estimated coating thickness and permittivity have been reproduced from Section 5.3. Deposition rates were calculated using the estimated thickness.

An important issue that will be resolved in the next chapter is to determine whether

Table 6.1: Plasma Deposition Process Parameters for Bromobenzene Coated Microchips

Flow Rate (F) $\sim$ 4 sccm, Molecular Weight (M) = 157

W/FM $\sim 5x10^7$ J/kg @ 25 Watts

ID #	**Date**	**RF Power** (watts)	**Dep. Time** (min)	**Dep. Rate** (Å/min)	**Thick.** est. (μm)	**Perm.** est. ($x10^{-11}$ F/m)
Z3	3/5/86	150	38	421	1.6	2.44
Z4	3/7–11/86	50	1293	185	–	2.90
Z5	3/19/86	50	60	320	1.9	2.68
Z7	4/17/86	25	167	407	6.8	2.60
Z8	4/17/86	25	167	395	6.6	2.58
Z17	8/14/86	50	62	339	2.1	2.63
Z20	9/2/86	25	156	385	6.0	2.50
Z22	9/18/86	25	96	365	3.5	2.55
Z23	12/17/86	25	100	–	–	–
Z24	12/17/86	25	100	310	2.2	2.95

the coated chips respond to absolute moisture content, relative saturation or some combination of the two. For the answer to this question, it is necessary to continue the discussion of liquid–solid equilibria. If the coating behaves as the paper does then, based on the conclusions regarding the paper–oil system, it would be expected that the moisture content of the coating would likewise depend on the solubility limit and either the relative saturation or the absolute moisture content of the oil. Jumping ahead slightly, the electrical response of the coated chip is related to the absolute moisture content within the coating. Thus, the conclusion would be that the moisture sensor does not measure either purely moisture content or relative saturation of the oil. Instead, a set of curves similar to those of Fig. 6.1c would be needed to characterize the response of the moisture sensor. However, the plasma deposited bromobenzene coating may hold moisture in a very different manner than the paper. If it acts as a semipermeable membrane, allowing moisture to enter the coating while blocking any oil penetration, then the moisture content within the coating will be a function of the osmotic pressure. Thus, the moisture content within the coating will depend on the concentration, or absolute moisture content, in the oil. Measurements of the coated chip sensitivity with oil at different temperatures (Section 7.1.5), and degraded oil having a different solubility limit (Section 7.1.6), indicate that the moisture sensors do indeed respond to absolute moisture content in the oil.

Chapter 7

Moisture Sensor – Experimental Results

The bulk of the experimental results using plasma polymerized bromobenzene coated chips is presented in this chapter. Section 7.1 focuses on results obtained in controlled laboratory oil samples. First, it is demonstrated that the coated chip response to moisture in the oil can be normalized into a master curve, resulting in a simple calibration technique. Next, the moisture sensitivity of both fresh and aged coatings is examined. The coated chip frequency response is shown to be linear with changes in the electric field strength and ambient temperature. Experiments at variable oil temperature and oil degradation validate the concept of an absolute moisture content sensor. The rapid transient response to changes in moisture content is also presented. The problem with large excursions in the DC offset of the device, associated with high flow rates of dry oil across the film, is discussed. Finally, preliminary test results using the sensors on–line, in the oil circulation loop of an instrumented test transformer, are presented in Section 7.2.

7.1 Testing in Oil Samples

7.1.1 Experimental Procedure

Experiments were conducted in pyrex reaction kettles containing transformer oil and sealed from the atmosphere (Fig. 7.1). A magnetic stirrer was used to maintain uniformity within the kettle. The coated microchips were immersed in the fluid with electrical leads extending through rubber stoppers. Increases in moisture content were obtained by bubbling nitrogen through a beaker of tap water and then through the oil. At room temperature, the solubility limit is roughly 40–50 ppm for unused oil 'out of the barrel'. Drying of the oil was performed by either pulling a vacuum or bubbling dry nitrogen through the oil. Direct measurements of the moisture content were obtained by injecting oil samples into a Karl Fischer type moisture analyzer (Mitsubishi Moisturemeter, Model CA–03). The moisture contents reported represent the average of 4 to 6 samples from a particularly conditioned oil. Each sample volume of oil had at least 10 μg of water, where possible, to minimize errors due to the limits of precision of the device ($\pm$ 3 μg). The lowest measurable limit of moisture in oil is 2–3 ppm.

A problem encountered when performing experiments at elevated temperatures was one of maintaining significant moisture in the heated oil. As the oil–air system tried to reach equilibrium, the heating of the oil tended to drive the moisture into the air, resulting in very dry, hot oil. Attempts to increase the moisture content of the hot oil by bubbling in wet nitrogen were not very successful. A variation on this approach was to bubble nitrogen through boiling water and then through the oil. This worked too well in that free water formed in the oil due to condensation at the lower temperature. The solution was to place a piece of wet paper insulation in the hot oil. In this case, the mass transfer of moisture to the oil was greatly enhanced through the vehicle of the paper insulation.

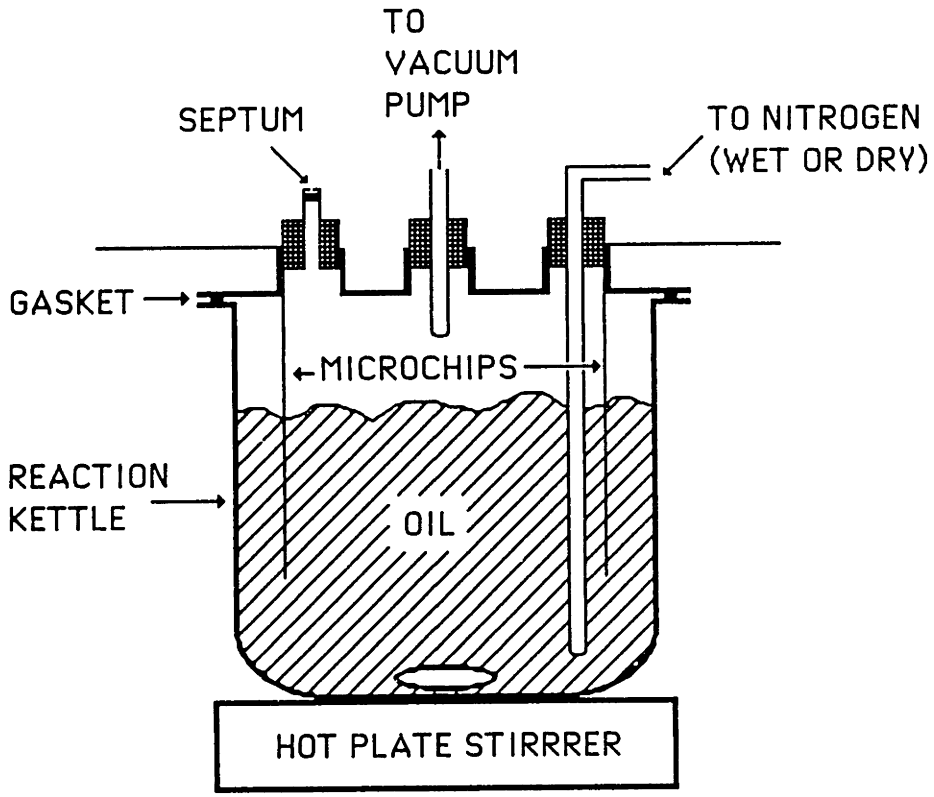


Figure 7.1: Apparatus for Conducting Moisture Sensitivity Measurements with Coated Microchip Moisture Sensors.

7.1.2 Master Curves For Coated Chips

The frequency response for a 6.8μm thick coated chip, Z7, as the moisture content of the oil was varied between 8 and 12 ppm, is shown in Fig. 7.2. A key feature of this behaviour is the shape invariance of the frequency response. The coated chip response, with moisture content as the external variable, fits Definition 3.1 and thus, a master curve can be constructed. Using f_p as the normalization parameter, the frequency at which the phase peak occurs, a master curve for Z7 is shown in Fig. 7.3. A cubic spline representation (supplied with the plotting routine) of the discrete frequency data is shown for clarity.

Two more examples of coated chip responses converted to master curves are for a 0.5 μm thick coating, Z3 (Figs. 7.4 and 7.5) and for a 6.0 μm thick coating, Z20 (Figs. 7.6 and 7.7). There are large differences in the range of moisture contents to which these chips are sensitive – Fig 7.4 represents a change from 24 to 53 ppm whereas Fig. 7.6 represents a change from 2 to 7.6 ppm. A discussion of these sensitivities is given in the following section. Note also that the phase curves are broader, with shallower peaks, than the curves of either Fig. 3.2 or Fig. 3.4 – particularly in the case of Z7 and Z20, the thicker coated chips. Predicting the shape of these responses is a difficult task and is left for Section 8.2.

The beauty of the master curves lies in their simplicity. Through application of Theorem 3.1, the dependence on moisture content can be reduced to a change in one parameter, f_p. Using a spline function to represent the master curve, any future gain–phase data can easily be interpolated to estimate f_p and thus obtain the moisture content of the oil. From now on, evaluation of the moisture sensor's performance will be in terms of f_p. This technique is used extensively for the on-line test results presented in Section 7.2.

Furthermore, Theorem 3.1 lends credence to the belief that the same physical mechanism is governing the response of the coating for all the moisture content changes. Tracking moisture content using f_p is analogous to the usage of thin film humidity sensors in air, where changes in RH are monitored through changes in resistivity of

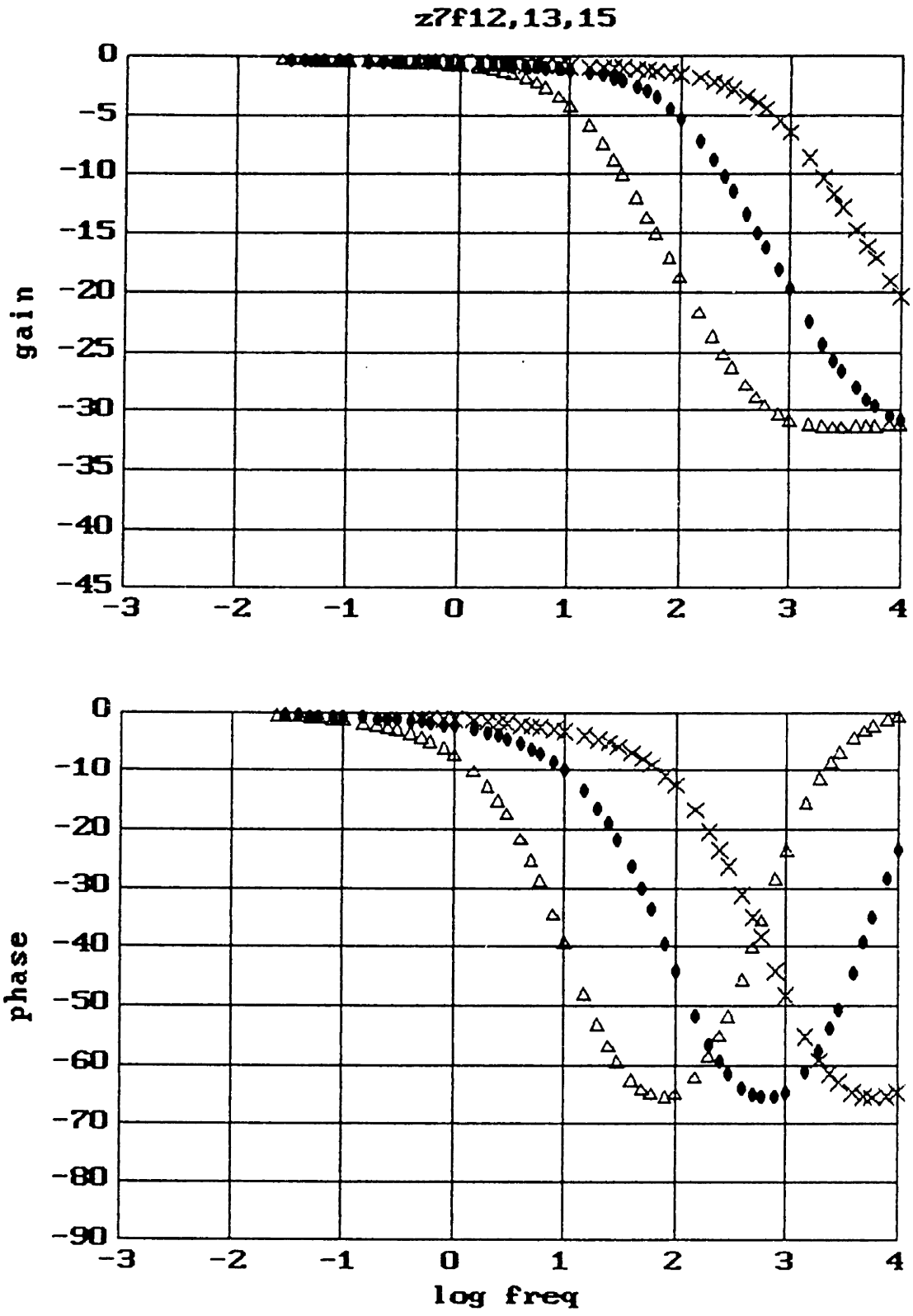


Figure 7.2: Measured Responses of Fresh Moisture Sensor Z7, 6.8 μm PP–Bromobenzene Coated Microchip, in Stirred, Unused Oil. Oil Moisture Content Varied From 12 to 8 ppm.

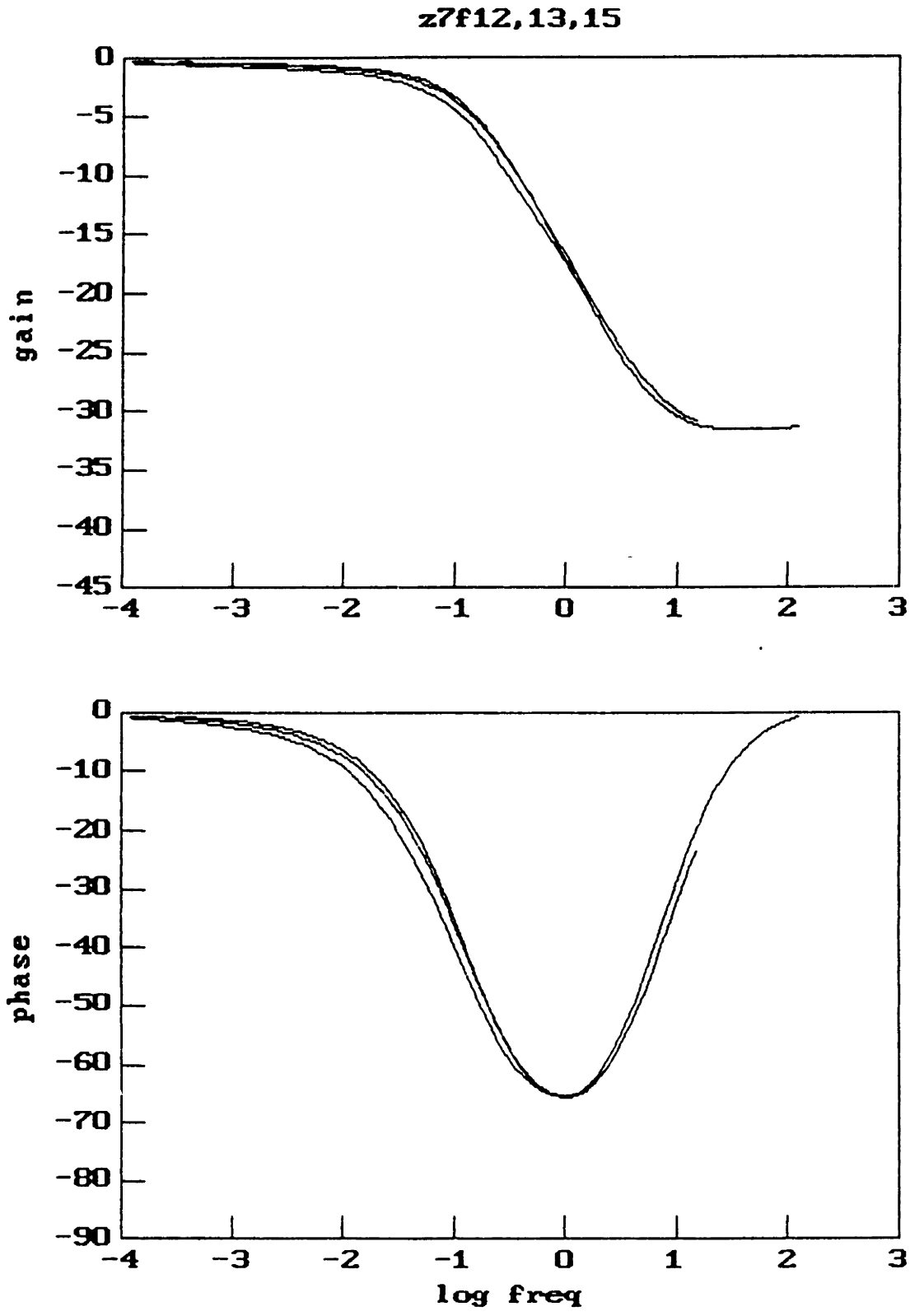


Figure 7.3: Master Curve for Fresh Moisture Sensor Z7 Produced by Normalizing Responses of Fig. 7.2 to f_p.

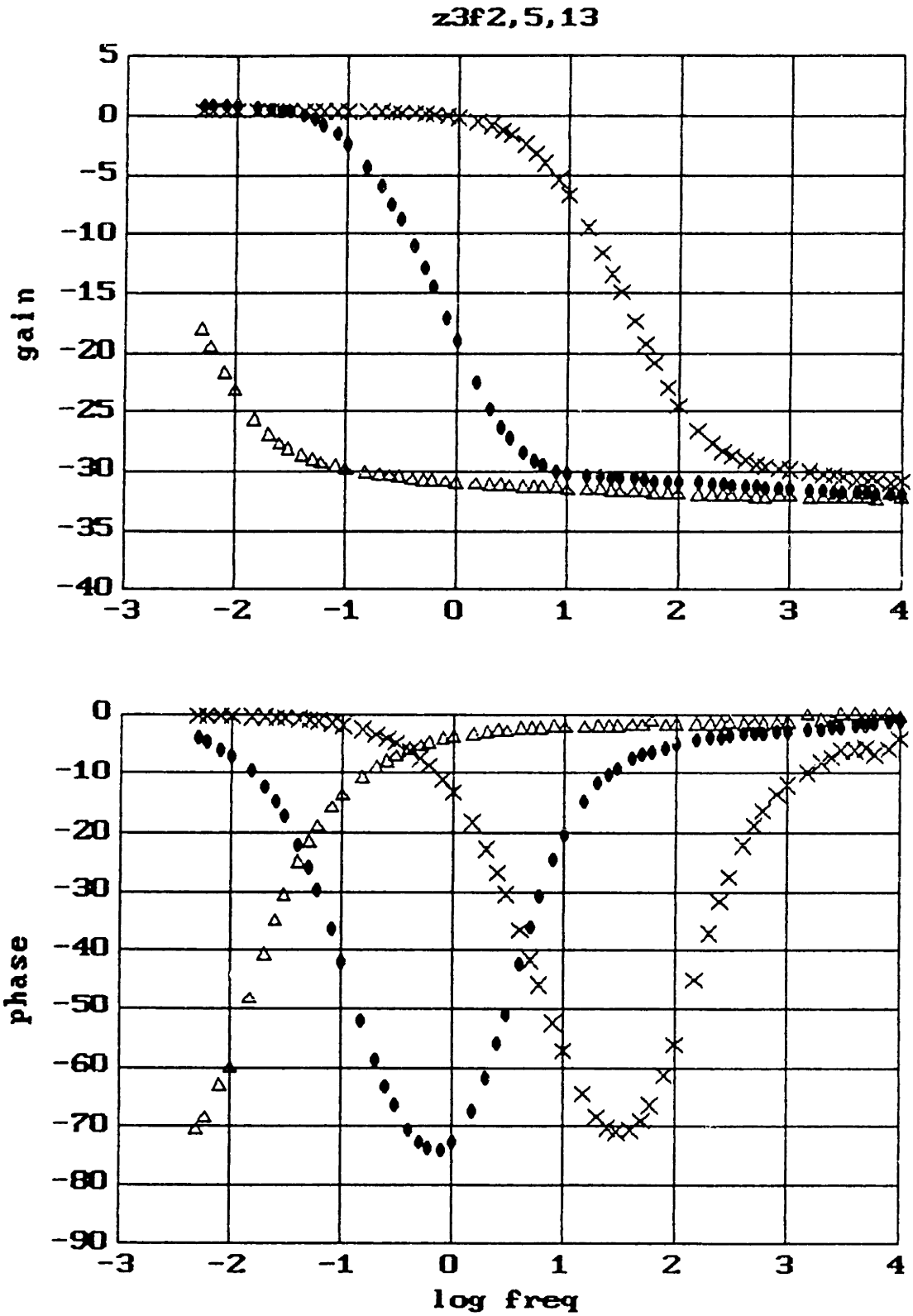


Figure 7.4: Measured Responses of Fresh Moisture Sensor Z3, 0.5 μm PP–Bromobenzene Coated Microchip, in Stirred, Unused Oil. Oil Moisture Content Varied From 24 to 53 ppm.

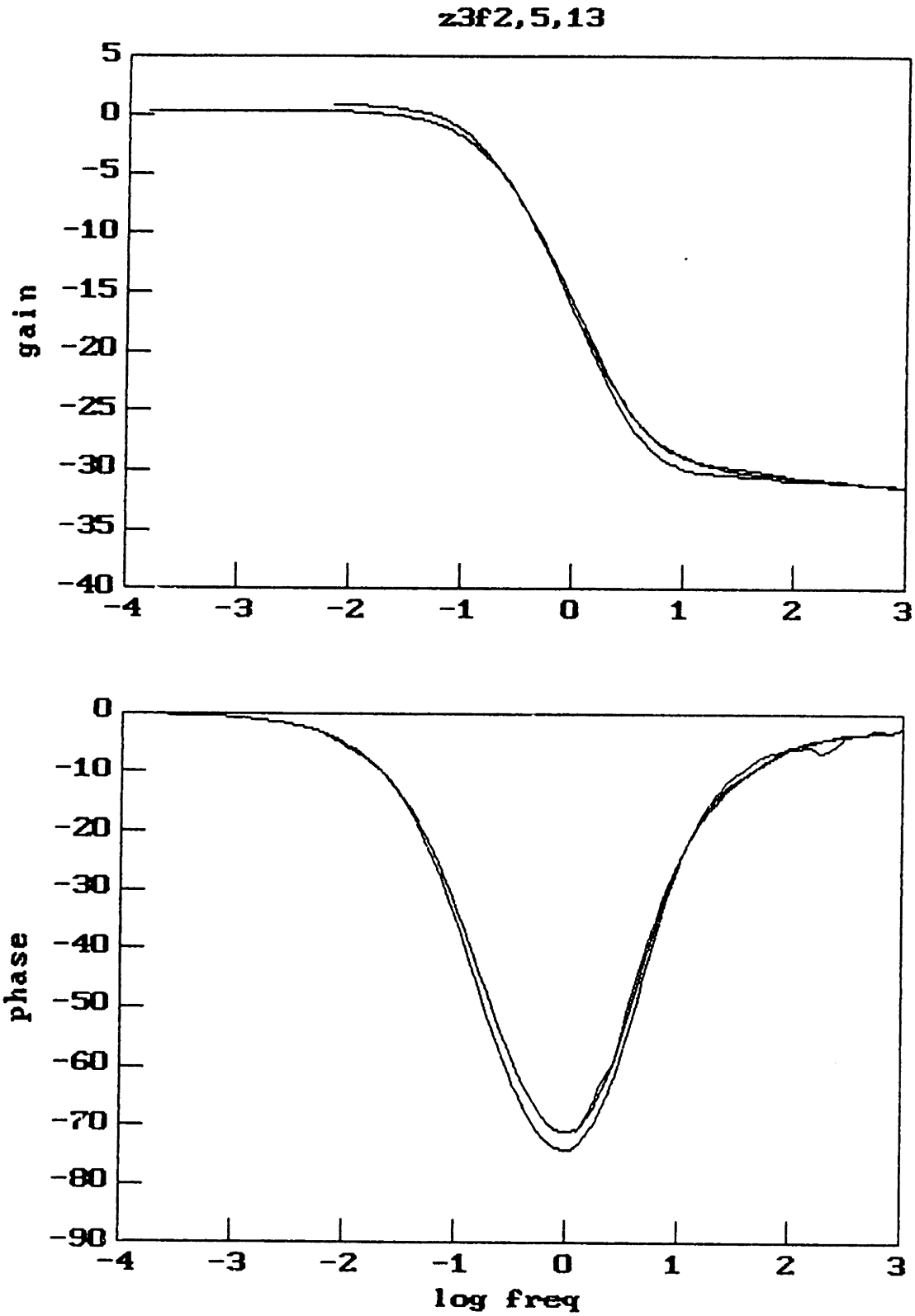


Figure 7.5: Master Curve for Fresh Moisture Sensor Z3 Produced by Normalizing Responses of Fig. 7.4 to f_p.

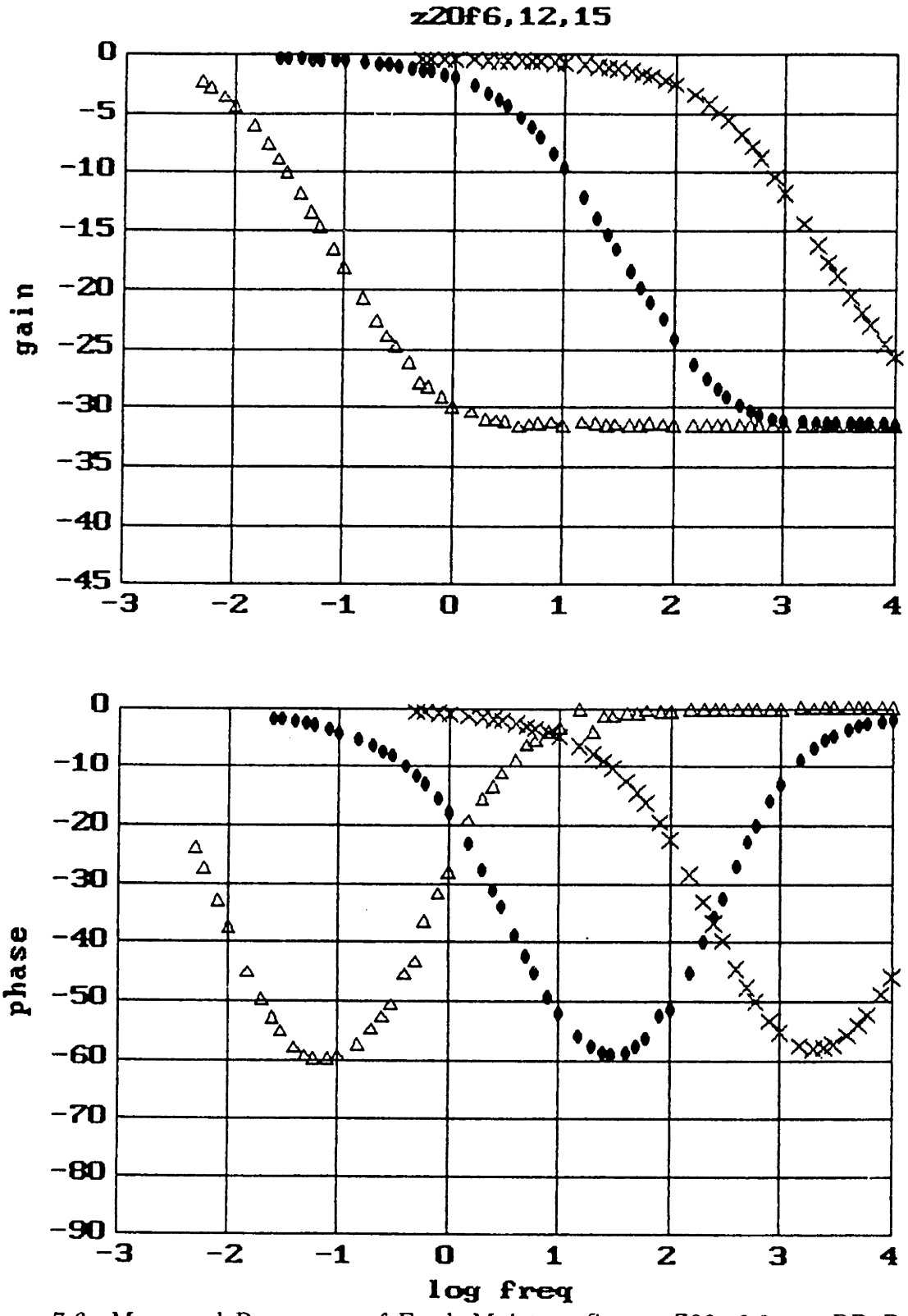


Figure 7.6: Measured Responses of Fresh Moisture Sensor Z20, 6.0 μm PP-Bromobenzene Coated Microchip, in Stirred, Unused Oil. Oil Moisture Content Varied From 2 to 7.6 ppm.

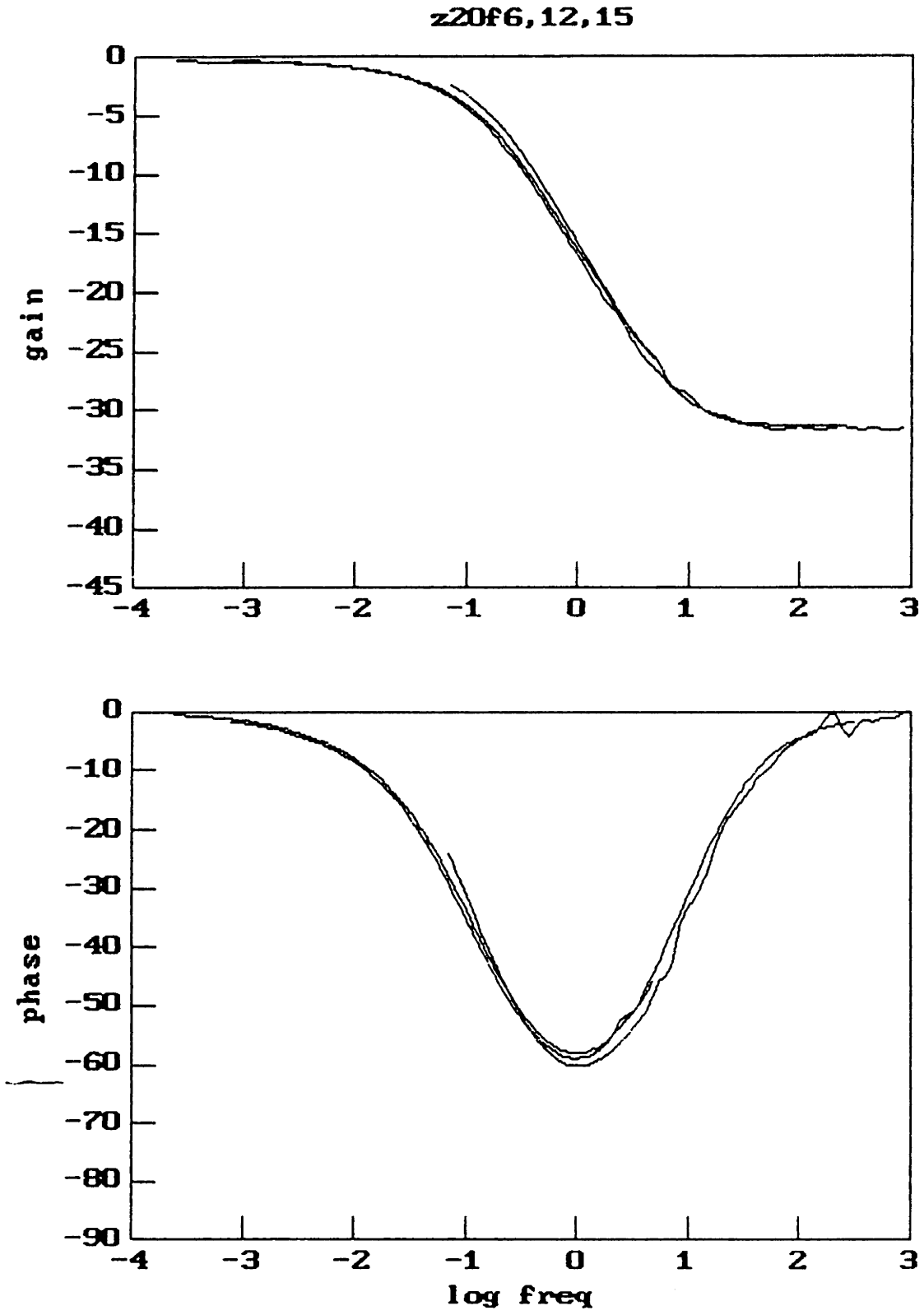


Figure 7.7: Master Curve for Fresh Moisture Sensor Z20 Produced by Normalizing Responses of Fig. 7.6 to f_p.

the thin film.

7.1.3 Coating Sensitivity to Moisture in Oil

Fresh Coatings

Measurements of the frequency response of moisture sensors in oil were taken soon after (within several days) coating the microchips. Fresh coatings refer to the length of time the coated microchips have been immersed in oil. Extracting f_p from the responses, the relationship between moisture content, as measured by the Mitsubishi Moisturemeter, and f_p, is plotted in Fig. 7.8 for several different sensors. A log–log scale was chosen to show that all the coated chips exhibited roughly the same sensitivity though the coating thickness varied from 1.6 μm to 6.8 μm. Using a power law relation between f_p and moisture content in the oil (mc_{oil}),

$$f_p = \alpha \, (\mathrm{mc}_{oil})^{\beta} \tag{7.1}$$

yields a linear relationship on a logarithmic scale,

$$\log(\mathrm{mc}_{oil}) = \mathrm{A} + \mathrm{B} \, \log(f_p) \tag{7.2}$$

Note that high sensitivity to moisture implies a relatively flat curve. This convention was chosen because the moisture sensor will ultimately provide an on–line moisture content based on the f_p. A least squares linear best fit was used to relate $\log(\mathrm{mc}_{oil})$ to $\log(f_p)$. Best fit parameters for individual chips, and an overall average best fit for all but Z3, are provided in Table 7.1. The lone exception, Z3, a 0.5 μm coated chip, was coated at a much higher RF power than all the others. This may have significantly affected the structure and properties of the deposited film (see Section 8.1). Though the sensitivity of Z3 is comparable, there is an increased intercept, resulting in a chip responsive to high moisture contents within the operational frequency range of the microdielectrometer. Differences in the process parameters can be seen in Table 6.1. In general, it is observed that the freshly coated chips were highly sensitive to low levels of moisture, with f_p varying over many orders of magnitude for a change of

Table 7.1: Best Fit Parameters for Freshly Coated Chips

$$\log(\mathrm{mc}_{oil}) = \mathrm{A} + \mathrm{B}\ \log(f_p)$$

Symbol	ID #	Thickness (μm)	A	B
o	Z17	2.1	0.497	0.184
*	Z23	~ 2	0.350	0.206
o	Z20	6.0	0.529	0.125
+	Z7	6.8	0.565	0.147
□	Z8	6.6	0.487	0.182
△	Z4	32	0.386	0.162
Average parameters of above coated chips ⟶			0.489	0.159
X	Z3	1.6	1.659	0.091

roughly one order of magnitude in moisture content. There does not seem to be any consistent pattern with regards to coating thickness.

Aged Coatings

After sitting in unstirred, unused oil at room temperature for 11 weeks, there was a noticeable change in the sensitivity of the coated chips. Fig. 7.9a shows a decrease in f_p of roughly 2 decades for Z3. For this combination, low frequency f_p's at high moisture contents, no sensitivity curve can be obtained. Reducing the moisture content would result in f_p's below 0.005 Hz, the lower frequency limit of the microdielectrometer. In addition, the relaxation of the oil starts influencing the response at these low frequencies. The upper limit on f_p is established by the solubility limit of the oil at room temperature.

A slightly more detailed change is shown in Fig. 7.9b for Z23. Here the decrease in sensitivity occurred over a 10 week span while the coated chip was immersed in unused oil at room temperature. It should be noted that, after the 10 weeks of aging, the coated chip was removed from the initial oil sample and placed into a fresh, unused oil sample for the second set of measurements.

More complete characterizations of the reduction in sensitivity are shown in Figs. 7.10

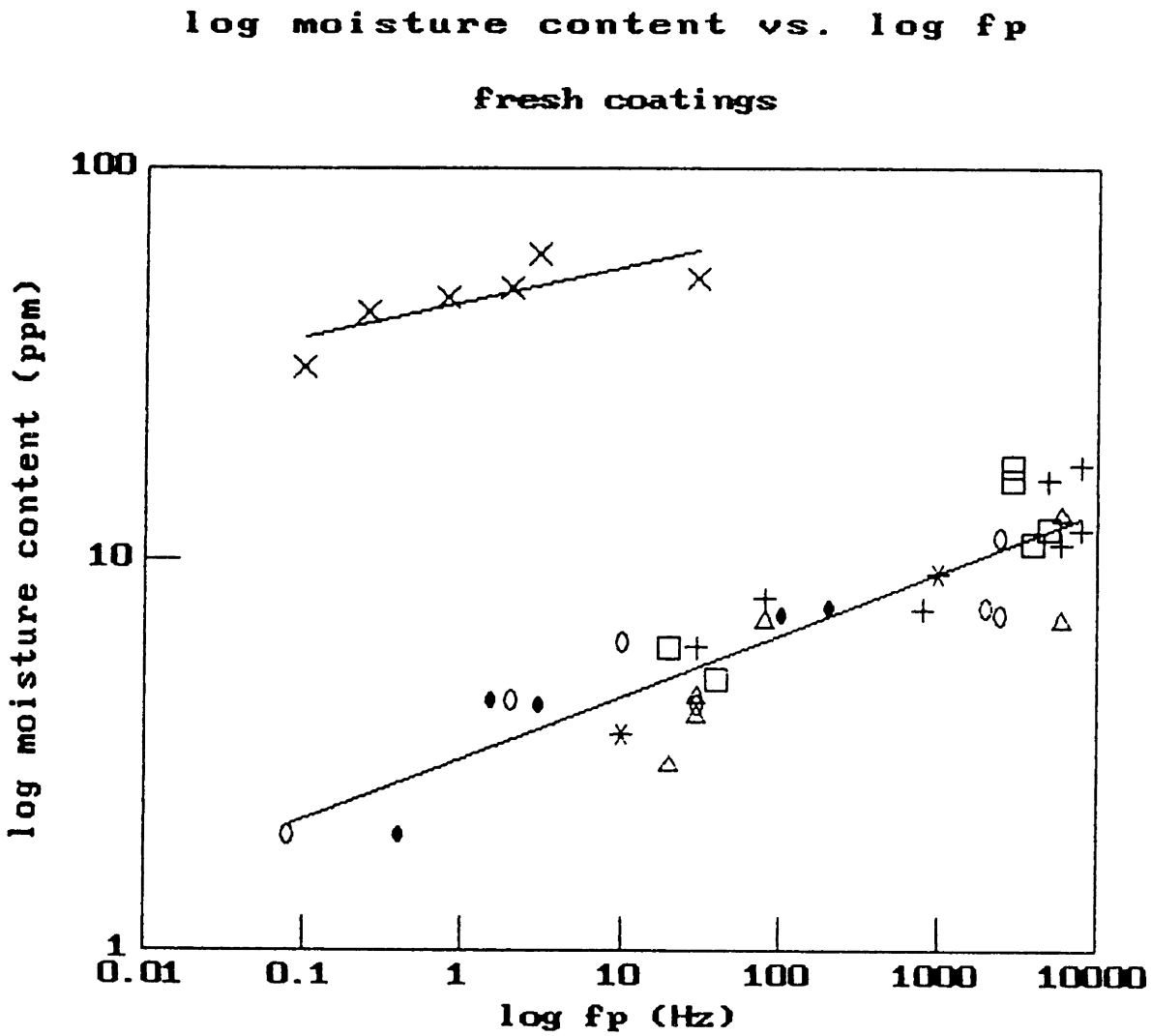
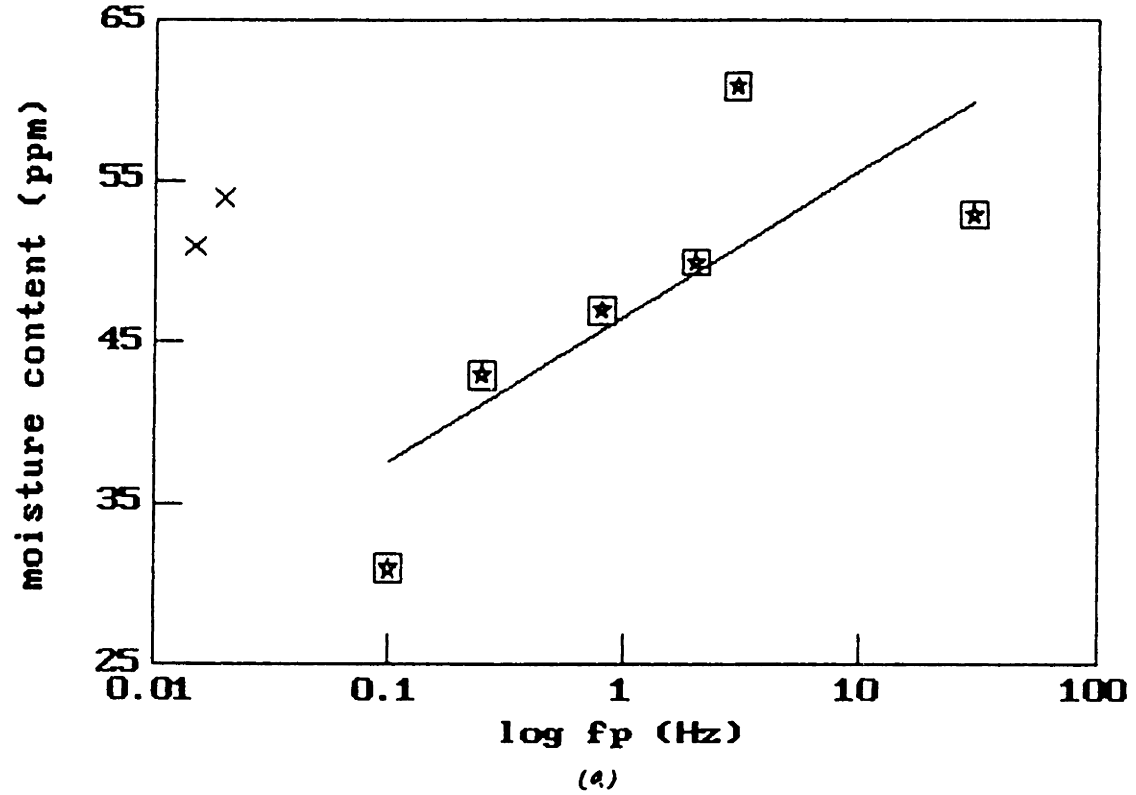


Figure 7.8: Compiled Moisture Sensitivity Data for Freshly Coated Moisture Sensors Z3,4,7,8,17,20,23 in Stirred, Unused Oil.



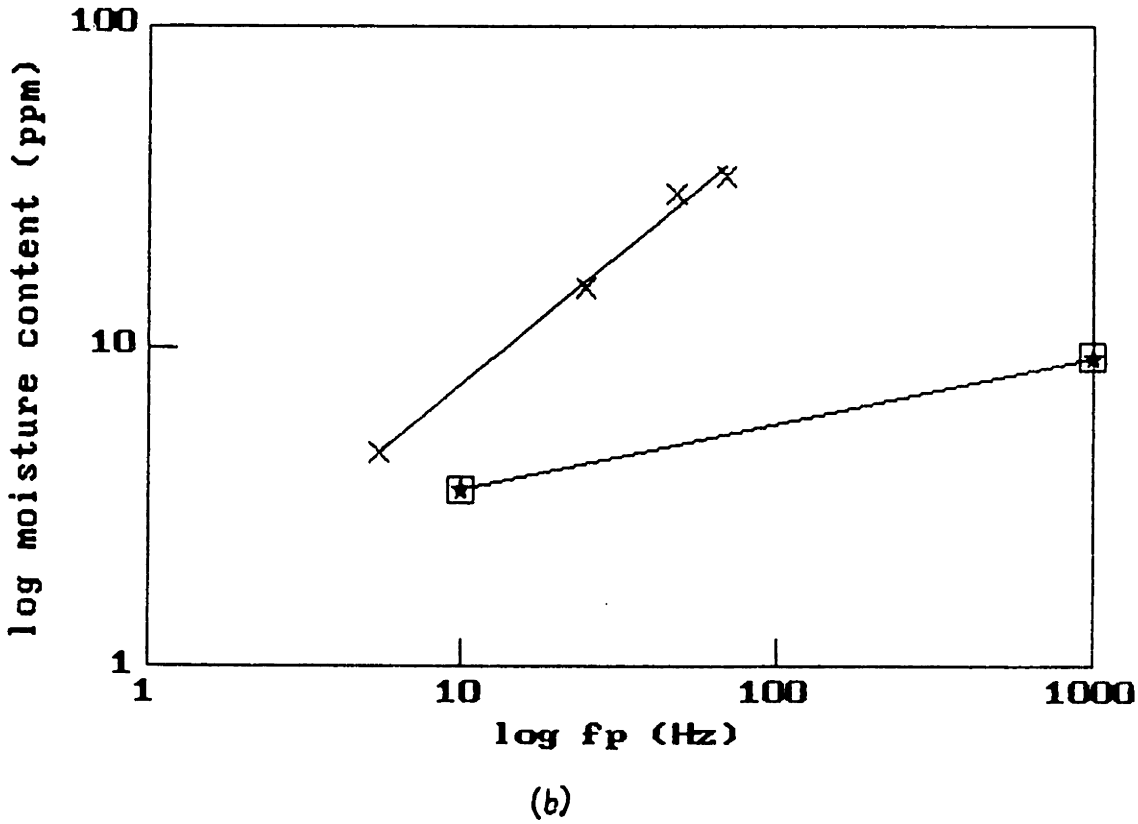
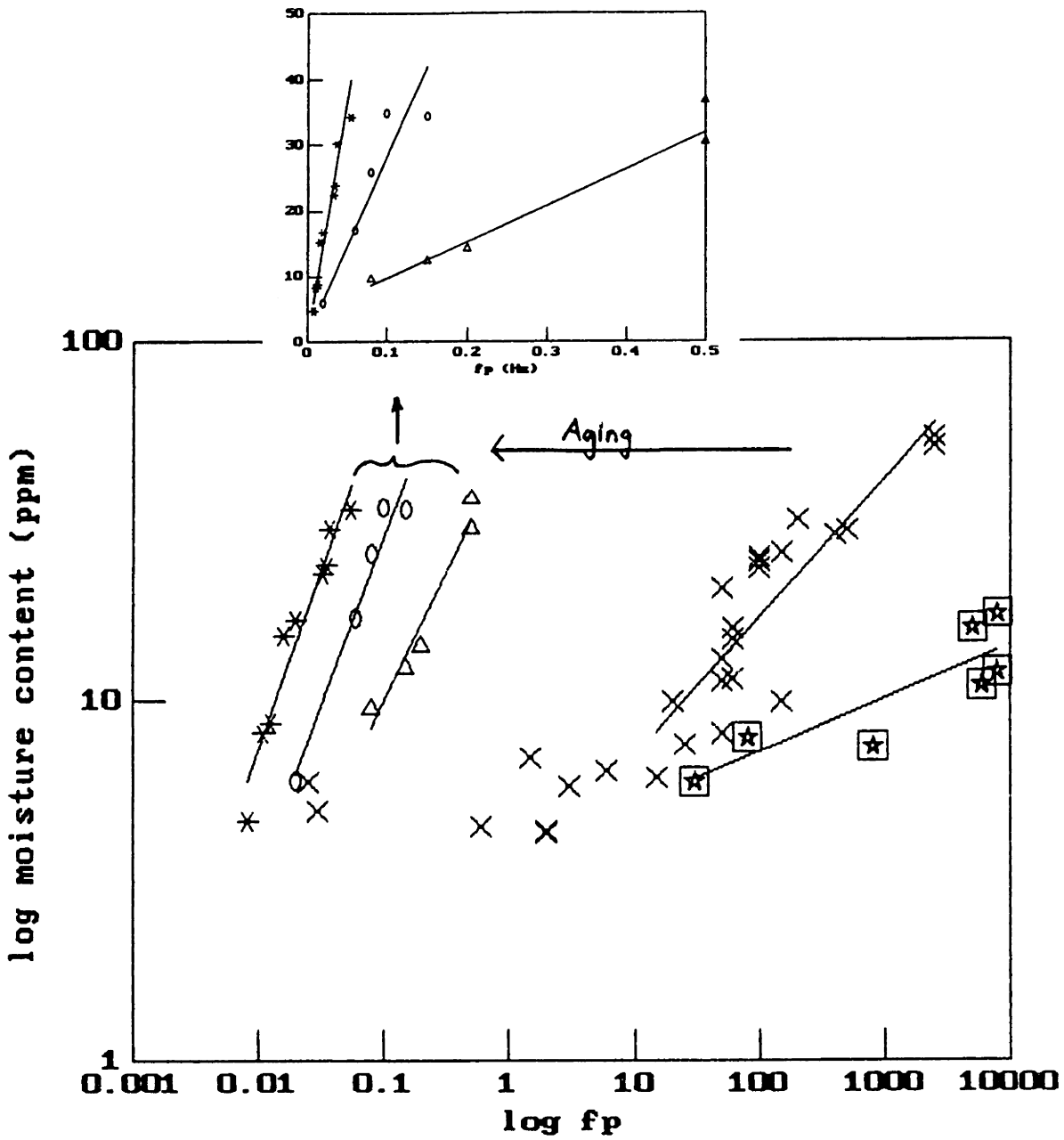


Figure 7.9: Moisture Sensitivity of Fresh and Aged Coated Microchip in Stirred, Unused Oil for a) Z3 and b) Z23.

and 7.11 for Z7 and Z20, respectively. In both cases, the fresh coating response is represented by the boxed star symbol. Note that a reduction in sensitivity corresponds to an increase in slope due to the choice of the abscissa and ordinate axes. For Z7 (Fig. 7.10), the first reduction in sensitivity (crosses) is again due to aging for 11 weeks in oil at room temperature, as with Z3. This first aged response can be divided into two regions, a very high sensitivity for low moisture contents (< 6 ppm) and a lower sensitivity, represented by the line, for higher moisture contents. Further aging occurred due to the combination of time and temperature. The total time elapsed between the data represented by the crosses and the triangles was 16 weeks. Measurements of the sensitivity were taken at elevated temperatures of 40°–60°C for 6 weeks of that time period. For the rest of the time, the chip was at room temperature in the oil. There was a significant decrease in sensitivity, and change in intercept, due to this aging. Between the data represented by the triangles and the circles, again 16 weeks had passed. This time only 1 week was spent at an elevated temperature of 42°C. A further decrease in sensitivity was observed. After another 6 weeks at room temperature, yet another decrease was observed (asterisks). The inset in Fig. 7.10 shows that the sensitivity of the aged coating decreases to a linear relationship, approaching a unity slope.

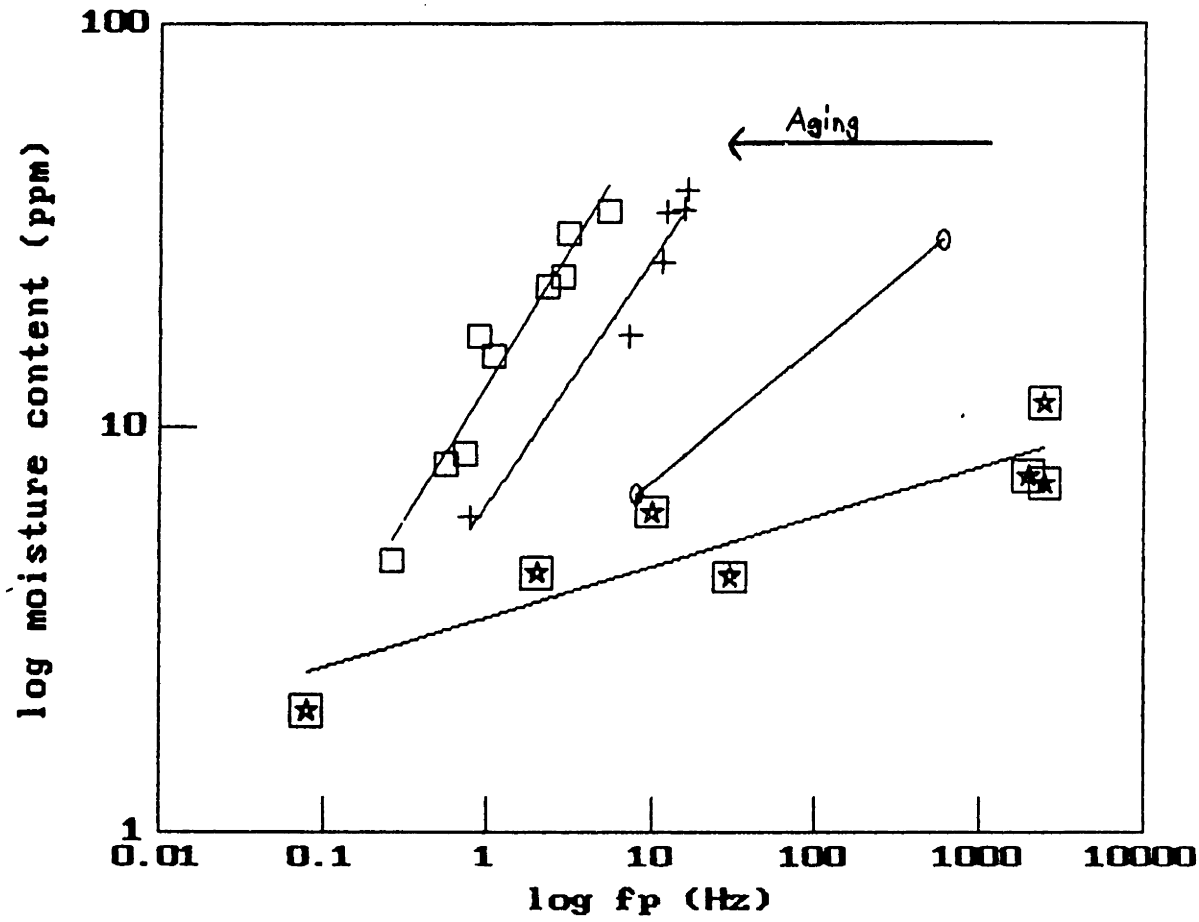
A similar decrease in sensitivity with aging, due to time and temperature, is shown in Fig. 7.11 for Z20. In this case, 18 weeks passed between the fresh coating measurements (boxed stars) and the first set of aged data (circles). For 5 weeks of this period the chip was operated at 40°C. Between the next two sets of data (circle and plus) the coated chip was operated in the oil circulation loop of an instrumented 50 kVA test transformer for 12 weeks. Details of this experiment are described in Section 7.2. The only information relevant to this section is that for those 12 weeks the coating was subjected to temperature variations of 10°–70°C and oil flows of up to 8 gal/min. After removal from the test facility, the coated chip was placed in unused oil for 6 weeks at room temperature. The sensitivity decreased again, as represented by the squares.

A discussion of the aging phenomenon is put off until Section 8.3. However, it is



Symbol	Sampling Period	Intercept (A)	Slope (B)
⋆	5/3/86–5/9/86	.565	.147
X	7/30/86–9/26/86	.467	.385
△	1/15/87–1/21/87	1.719	.721
o	5/12/87–5/18/87	2.398	.940
*	6/30/87–7/23/87	2.868	1.005

Figure 7.10: Change in Moisture Sensitivity with Aging of Moisture Sensor Z7 in Stirred, Unused Oil. Inset Shows Linear Sensitivity of Aged Coating.



Symbol	Sampling Period	Intercept (A)	Slope (B)
⋆	9/9/86–9/21/86	.529	.125
o	1/26/87–1/27/87	.538	.333
+	4/23/87–5/18/87	.815	.603
□	6/26/87–7/23/87	1.108	.667

Figure 7.11: Change in Moisture Sensitivity with Aging of Moisture Sensor Z20 in Stirred, Unused Oil.

observed here that this aging does not appear to be correlated with any degradation in the mechanical integrity of the coatings, such as cracking or peeling. If this type of degradation had occurred, a significant change in the capacitance of the coating would result. This would be reflected in a change in the high frequency gain of the coated microchips in oil. No such change was observed in the high frequency gain of fresh and aged coatings.

Due to a combination of time and temperature, the sensitivities of the coatings, as reflected in f_p, changed monotonically over the 6 to 12 months of testing. There were also small changes in the shape of the master curves, reflecting changes in the dispersive nature of the coatings. The latter changes for chips Z3, Z7 and Z20 are shown in Figs. 7.12, 7.13 and 7.14, respectively. In general, the phase seems to broaden, with reductions in the peak, with aging. The gain becomes more S–shaped, changing more gradually with frequency as a result of the aging. Interpretation of these changes in shape are put off until Section 8.2.3.

7.1.4 AC Electric Field Strength Dependence

A simple resistive divider was placed between the AC voltage source and the set of driven electrodes on the microchip. By varying a potentiometer, an attenuation of as much as 1000:1 in the driven voltage could be effected. The maximum practical reduction that could be used was only one order of magnitude due to the reduced signal to noise ratio. This 10:1 attenuation corresponded to a decrease of 20 dB in the measured gain. (Gains are always computed relative to a 1 volt peak amplitude signal in the microdielectrometer). A comparison between responses obtained using attenuated and full strength signals for Z7 are shown in Fig. 7.15. These results were typical in that the responses of all coated chips with an attenuated signal matched the response with full signal strength. Therefore, the master curve was independent of the applied AC electric field strength. Linearity in the frequency response of the coated microchips, with respect to the electric field, was obeyed here.

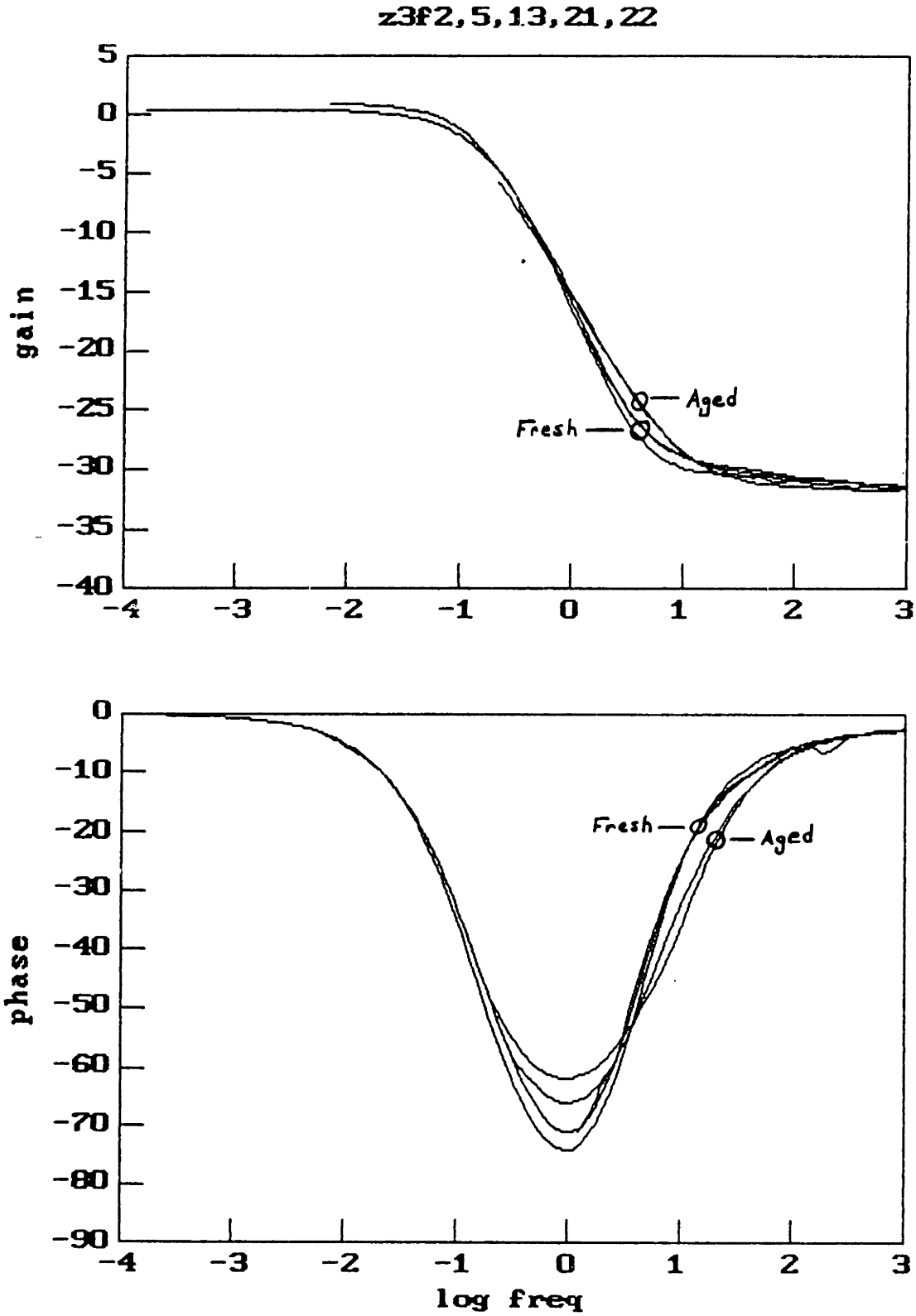


Figure 7.12: Change in Master Curve with Aging of Moisture Sensor Z3.

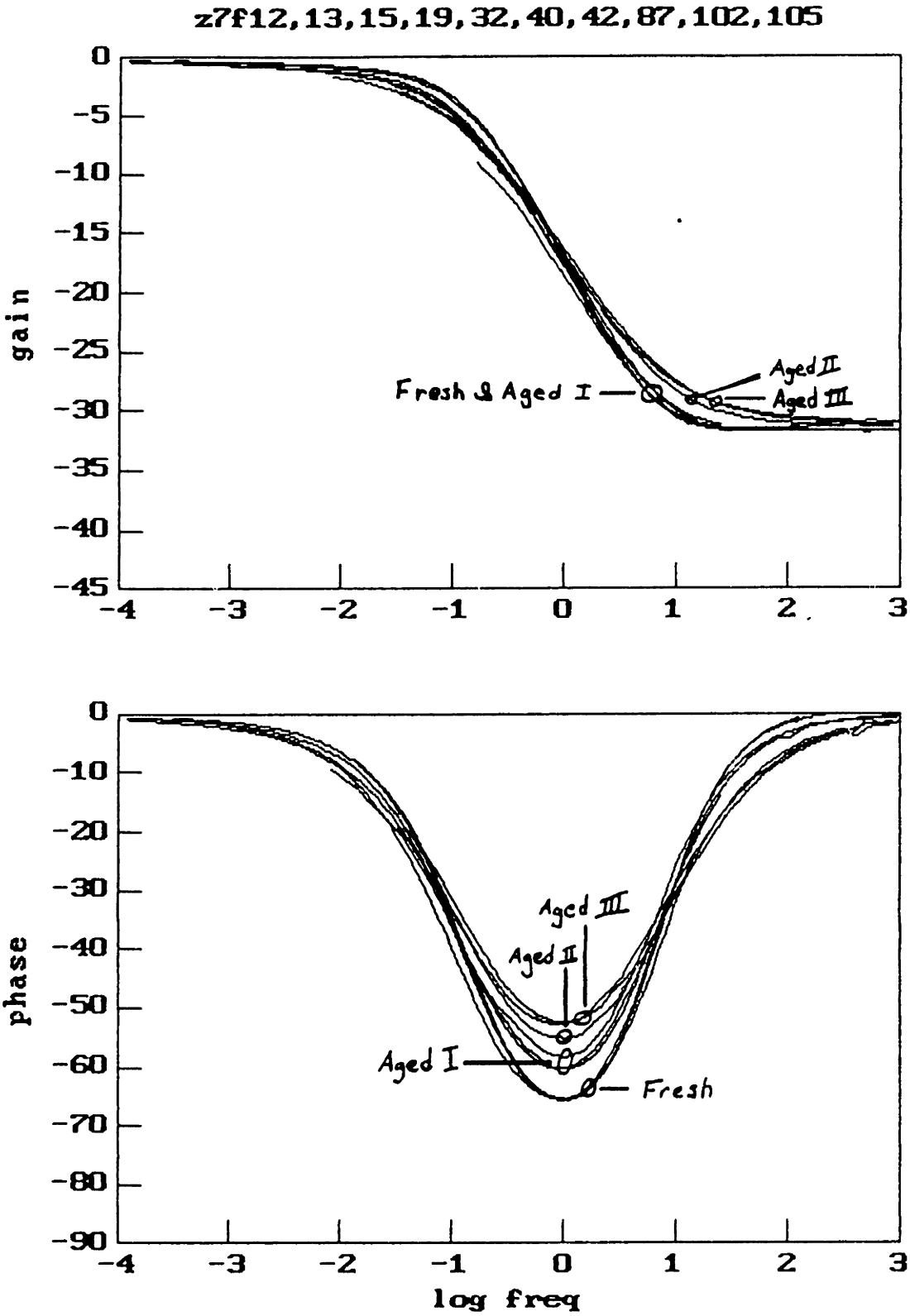


Figure 7.13: Change in Master Curve with Aging of Moisture Sensor Z7.

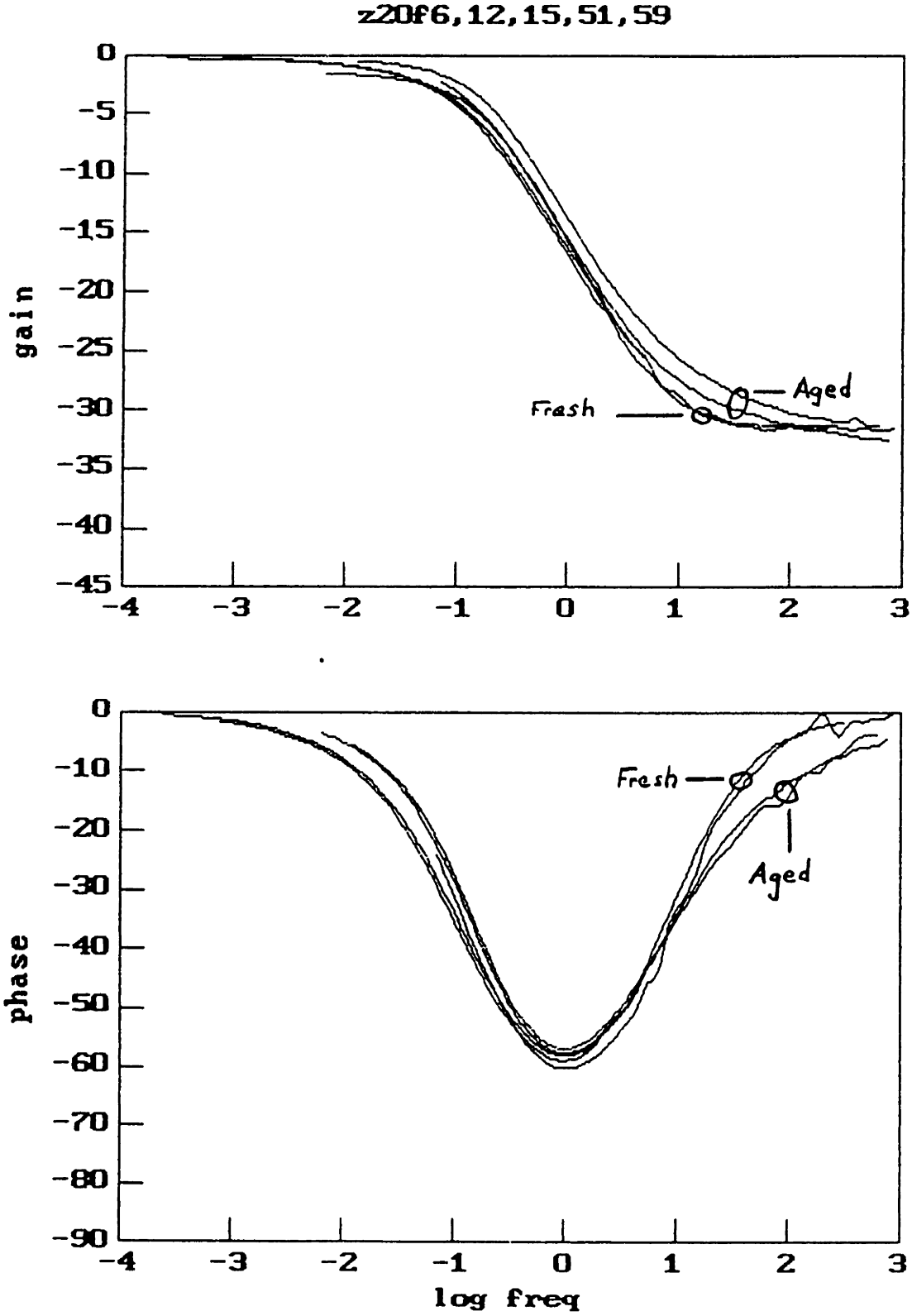


Figure 7.14: Change in Master Curve with Aging of Moisture Sensor Z20.

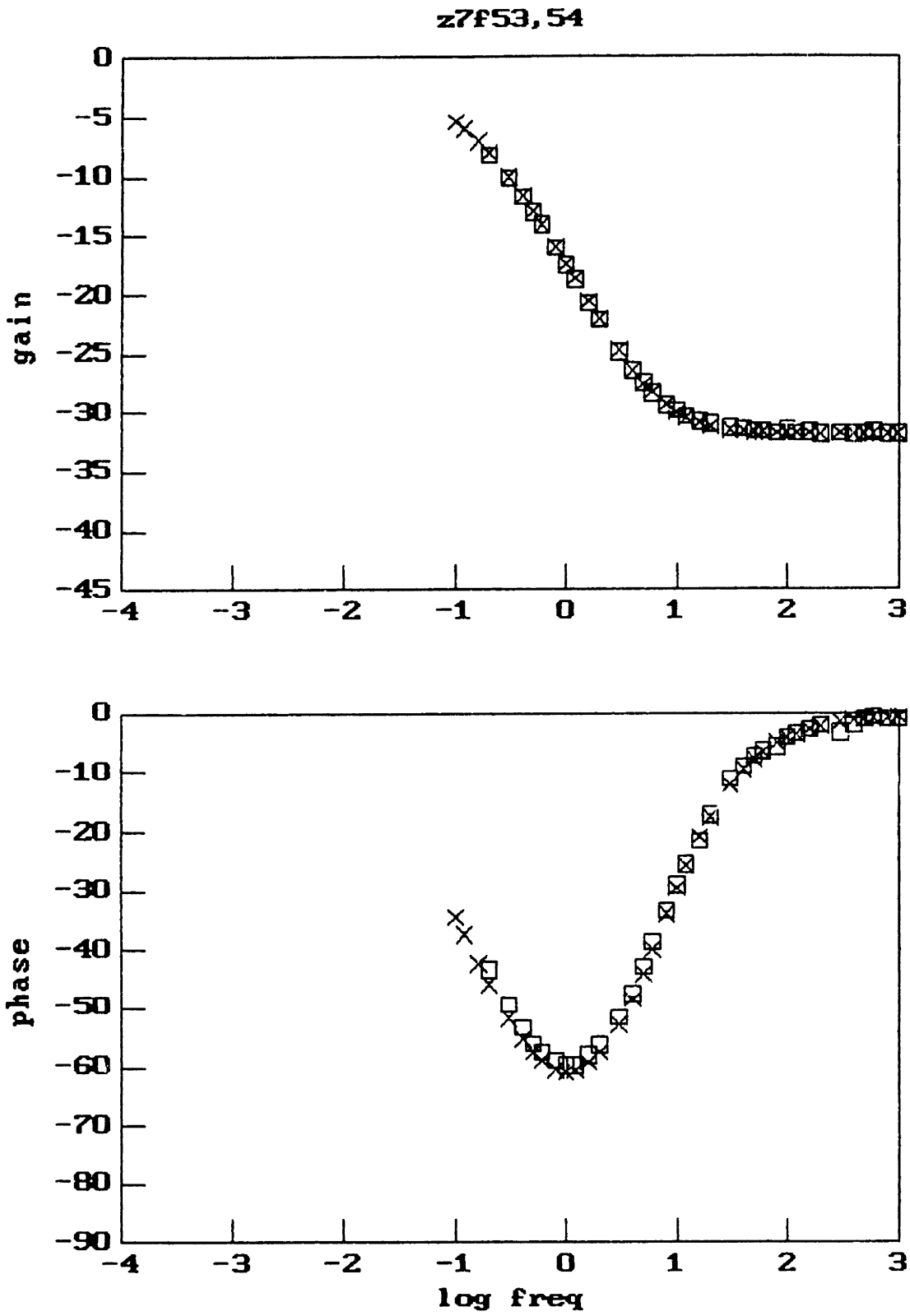


Figure 7.15: Independence of Master Curve on Applied Electric Field Strength for Moisture Sensor Z7 in Stirred, Unused Oil at $\sim$ 4 ppm Water. (X) - full voltage, 100%, (□) - attenuated to 9%

7.1.5 Temperature Dependence

The reproduction of the master curve was also observed with variations in the ambient oil temperature. This invariance is shown in Fig. 7.16 for Z7 as the temperature was varied from 27° to 60°C. There was actually a very slight deviation that can be attributed to a slight decrease in the oil permittivity of 2% with increased temperature. Although the oil conductivity was expected to increase with temperature, mainly due to the decrease in viscosity, this change resulted in a relaxation at frequencies still significantly below the relaxation in the coating. Thus, neither the change in oil permittivity nor conductivity significantly altered the curve shape.

The effect of temperature on the sensitivity of the moisture sensors could be the result of two effects. One effect would be the amount of moisture absorbed by the coating, a physicochemical effect. The second would be the change in dielectric properties due to temperature, an electrical effect. If the coating behaves as the paper does, then the amount of moisture within the coating would be a function of the solubility limit of the oil which, in turn, is a function of temperature. If the coating acts as a membrane, then the amount of moisture absorbed would be independent of the solubility and thus, temperature would only influence the response as related to the electrical effects. Using this second model, the membrane, the resulting relationship between f_p, moisture content and temperature, is hypothesized to be

$$f_p \;=\; \alpha\,(\mathrm{mc}_{oil})^{\beta}\; e^{-(E_A/kT)} \tag{7.3}$$

where E_A is a thermal activation energy, k is the Boltzmann constant and T is the absolute temperature. Paralleling (7.2), the relationship is expressed on a logarithmic scale as

$$\log(\mathrm{mc}_{oil}) \;=\; \mathrm{A} \;+\; \mathrm{B}\log(f_p) \;+\; \frac{E_A}{2.303kT} \tag{7.4}$$

This model predicts that sensitivity measurements, such as those shown in Figs. 7.8–7.11, at various temperatures, should produce lines with the same slope (B) but different intercepts, due to the temperature contribution. An example of this is shown

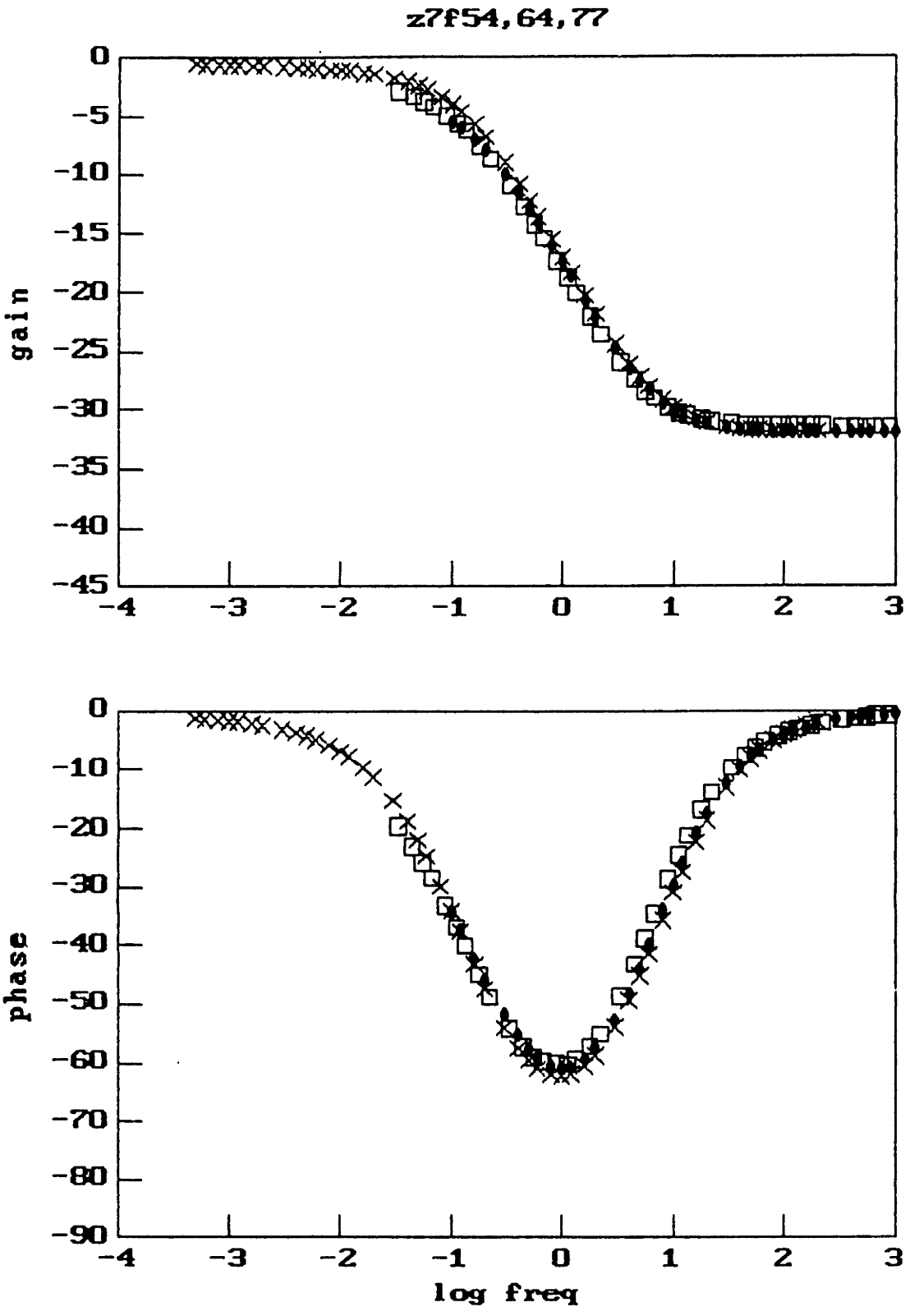


Figure 7.16: Independence of Master Curve on Temperature for Moisture Sensor Z7 in Stirred, Unused Oil. (•) - 27°C, 2 ppm, (X) - 45°C, 21 ppm, and (□) - 58°C, 11.2 ppm.

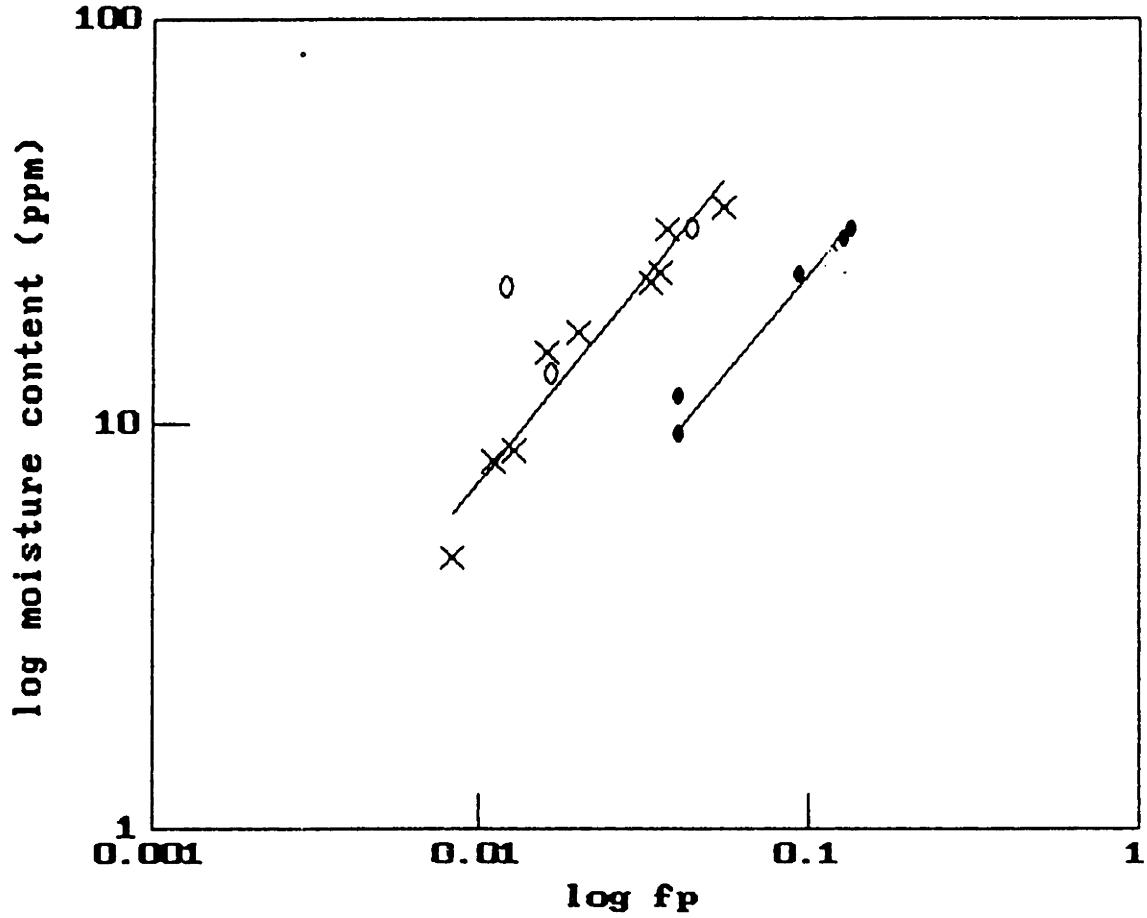
for the aged coated chip Z7 (Fig. 7.17). These lines may be considered parallel within the experimental error of the moisture analyzer. Using the difference in the intercepts of the two lines (Δint), an average slope (B) of 0.94, and

$$E_A = \frac{(8.314x10^{-3})(\Delta int)}{(96.48)\mathrm{B}(\log e)\Delta(\frac{1}{T})} \text{ eV} \tag{7.5}$$

yielded an activation energy of 0.66 eV. This value is close to the low temperature activation energy of 0.7 eV obtained by Tyczkowski and Kryszewski in the study of plasma polymerized organosilicon films [92]. It was not possible to obtain f_p for Z7 at higher temperatures for a significant range of moisture contents. The increased conductivity of the heated oil, combined with the low f_p's of Z7, resulted in the overlap, in frequency, of relaxation in the oil and in the coating. The response would no longer follow the master curve. It was difficult to obtain more experimental data as the other chips, not having aged as much as Z7, would exhibit a significant change in moisture sensitivity during the week it took to generate the several data points needed at an elevated temperature.

7.1.6 Tests with Aged Oil

Further proof of the independence of the moisture sensor from variations in solubility levels was obtained using an aged oil. This oil had been degraded by heating it at 90°C for 10 days while exposed to the atmosphere. There was a very noticeable darkening in color, even after 3 days. At room temperature, after bubbling wet nitrogen through the oil for several hours, the moisture content was 43 ppm as compared to 33 ppm for unused oil. The room temperature sensitivity of Z7 (Fig. 7.18a) and Z20 (Fig. 7.18b), in degraded and unused oil, is very similar. There appears to be some hysteresis in the behaviour of Z20. It was not possible to obtain f_p using a master curve for the case of the aged Z7 in degraded oil having a moisture content much less than 25 ppm. Again the increase in conductivity of the oil, now due to degradation, combined with the low f_p's of Z7 to result in the overlap, in frequency, of relaxation in the oil and in the coating. The response is no longer the master curve, as can be seen in Fig. 7.19



$$(\mathrm{X})\ -\ 30°\mathrm{C},\ \ \log(mc) = 2.84 + .99\log(f_p)$$

$$(\bullet)\ -\ 48°\mathrm{C},\ \ \log(mc) = 2.27 + .89\log(f_p)$$

$$E_A \sim 0.66\ \mathrm{eV}$$

Figure 7.17: Temperature Dependence of Moisture Sensitivity of Aged Moisture Sensor Z7 in Stirred, Unused Oil.

for Z7 in degraded oil at 8.5 ppm. It would be possible to estimate f_p by measuring the complex bulk permittivity of the oil and including its effect when estimating the properties of the bromobenzene coating from the frequency response of Fig. 7.19. This approach is similar to the one taken for measuring the complex bulk permittivity of the oil with a passivated microchip and including the dispersion in the parylene coating (Section 5.4.2).

7.1.7 Transient Response

The transient response of Z7 to the addition or removal of moisture to the oil was monitored. Shown in Fig. 7.20 is the transient response associated with the addition of moisture through the bubbling in of wet nitrogen. A change in moisture content of 6 to 26 ppm was effected while monitoring the chip response at 1 kHz. Initially the oil was dry, thus the low gain and phase. As moisture was added, the gain rapidly increased and the phase passed through a peak and then returned to a small value. This corresponds to tracing out the master curve, beginning with $(f/f_p) > 1$ and ending with $(f/f_p) < 1$.

Aside from the rate at which the moisture content was being changed, there was the question of how rapidly the change in moisture content reaching the coated chip. Assuming a turbulent mixing model for the moisture distribution within the reaction kettle, the rate at which moisture diffuses through the viscous laminar boundary layer must be examined. Using a boundary layer thickness on the order of 10 μm [93], and a diffusion coefficient of water in oil of $10^{-9}\text{m}^2/\text{s}$ at room temperature [94], the diffusion time

$$\tau_D = \frac{\ell^2}{D} \sim 0.1 \text{ sec} \tag{7.6}$$

is much smaller than the relevant time scale of the transient response. Even if the thickness ℓ was 10 times larger, τ_D would still be much shorter than the time scale of the response. Thus, diffusion through the laminar sublayer is estimated not to be the rate limiting step.

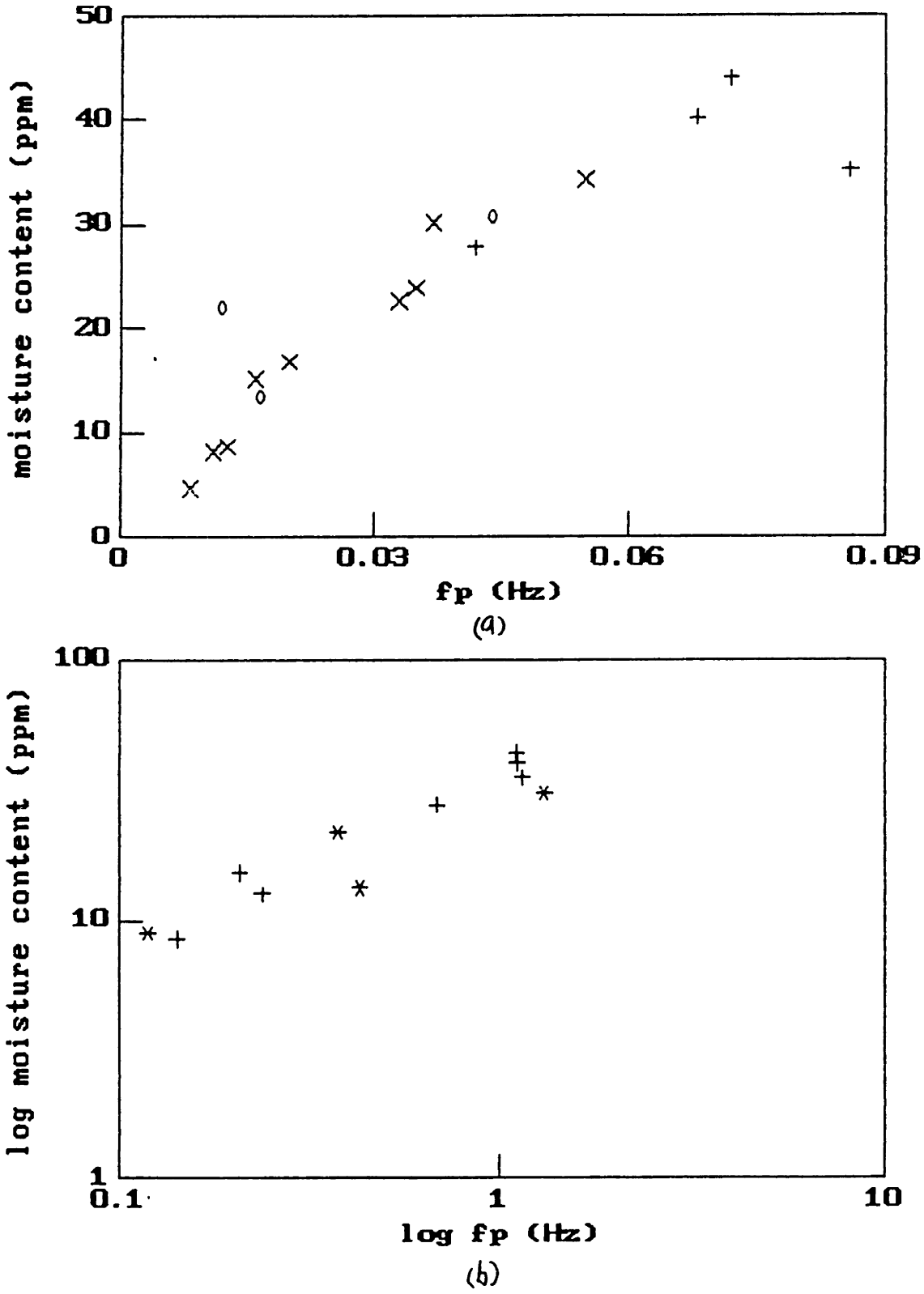


Figure 7.18: Moisture Sensitivity of Aged Moisture Sensors in Stirred, Unused and Oxidized Oil. a) Z7 in Unused (X,o) and Oxidized (+) Oil, and b) Z3 in Unused (*) and Oxidized (+) Oil.

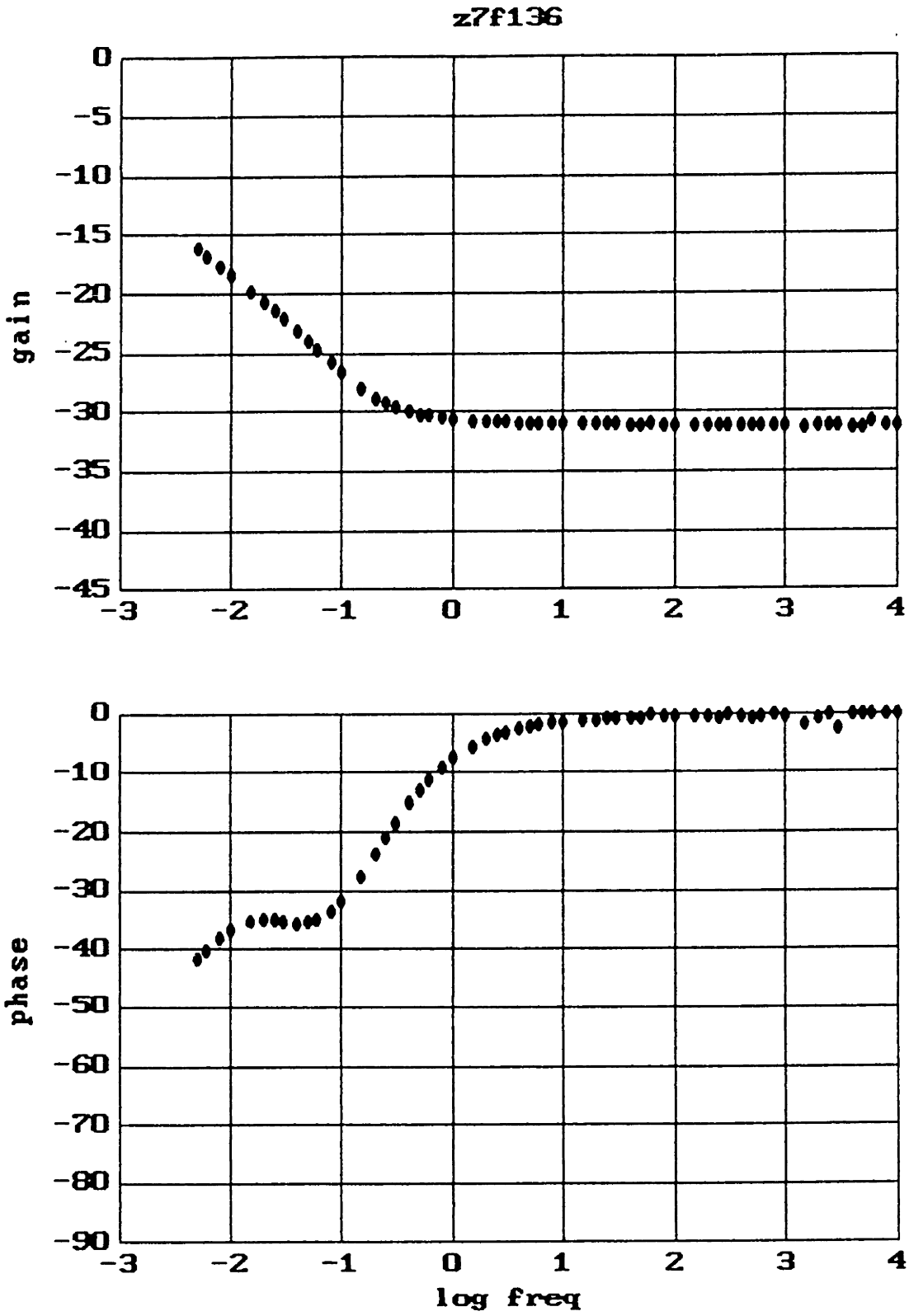


Figure 7.19: Measured Response of Aged Moisture Sensor Z7 in Stirred, Oxidized Oil at 8.5 ppm Water.

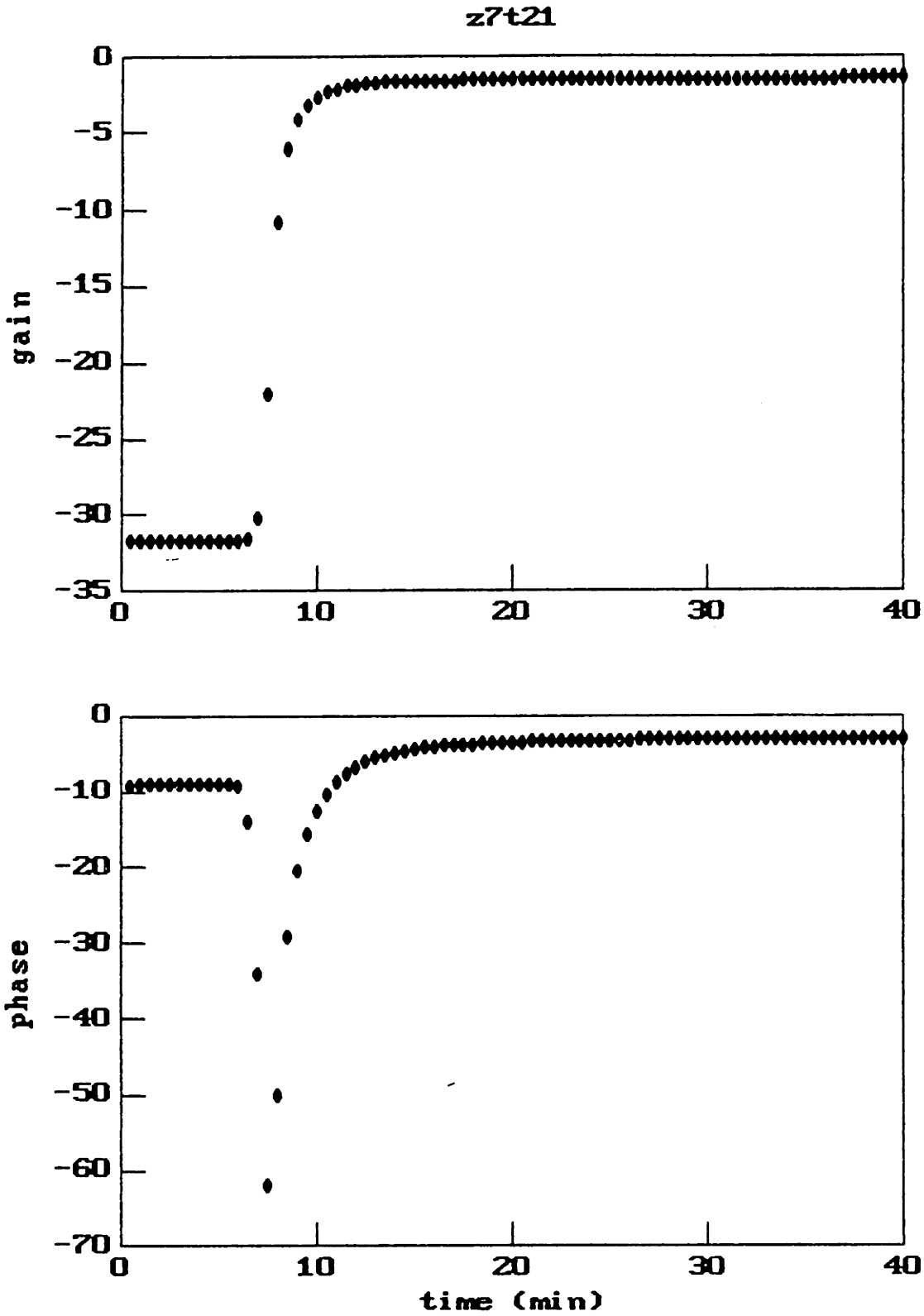


Figure 7.20: Transient Response at 1 kHz of Fresh Moisture Sensor Z7 in Unused Oil Going From 6 to 26 ppm Water.

Removal of moisture by bubbling in dry nitrogen resulted in a transient response on a similar time scale as addition of moisture (Fig. 7.21). As before, the path traced out followed the master curve, corresponding to the decrease in f_p with moisture content. The two step response corresponds to the introduction of dry nitrogen twice, the first time for 4 minutes, the second time for 2 minutes. Note that the speed of the response involves the rates at which moisture is removed from both the oil and the coating.

The time scale over which changes in the moisture content of the oil are reflected in the detector's response is on the order of minutes. With the thin coatings involved, 7 μm or less, this rapid response is to be expected.

7.1.8 DC Offset

Under normal operation, the DC offset displayed the following characteristics during a frequency scan: 1) a negative value of several hundred millivolts at high frequencies and 2) a transition to a less negative value as the frequency varied from high to low ($\Delta V \sim 60 - 120$ mV), accompanying the gain and phase transition. These characteristics are shown in Fig. 7.22a for the freshly coated Z20 in stirred, unused oil, having a moisture content of 4.4 ppm, and at room temperature. Two scans were taken, the first using an attenuated voltage (dots) and the second at full voltage (cross). The high frequency DC offset of the attenuated scan begins at a much less negative value and, consequently, there is a much smaller transition. The transition itself may be a characteristic of the operation of the FETs as it was observed for coated chips operated in air. This transition was also observed for uncoated chips embedded in 5 minute epoxy samples. Thus, this behaviour constituted normal operation of a coated microchip.

Dramatic changes in the average DC offset of Z20 were observed after heating the oil to 40°C and cycling the moisture content from dry to wet and back to dry again. The average DC offset changed upon the second drying out, increasing from roughly -.2 V to 2.9 V. Upon cooling back to room temperature, the average DC offset decreased to $\sim$ 1.9 V. At a moisture content of 6.9 ppm, the transition was still observed, again increasing roughly 50 mV as the frequency was decreased (Fig. 7.22b). This

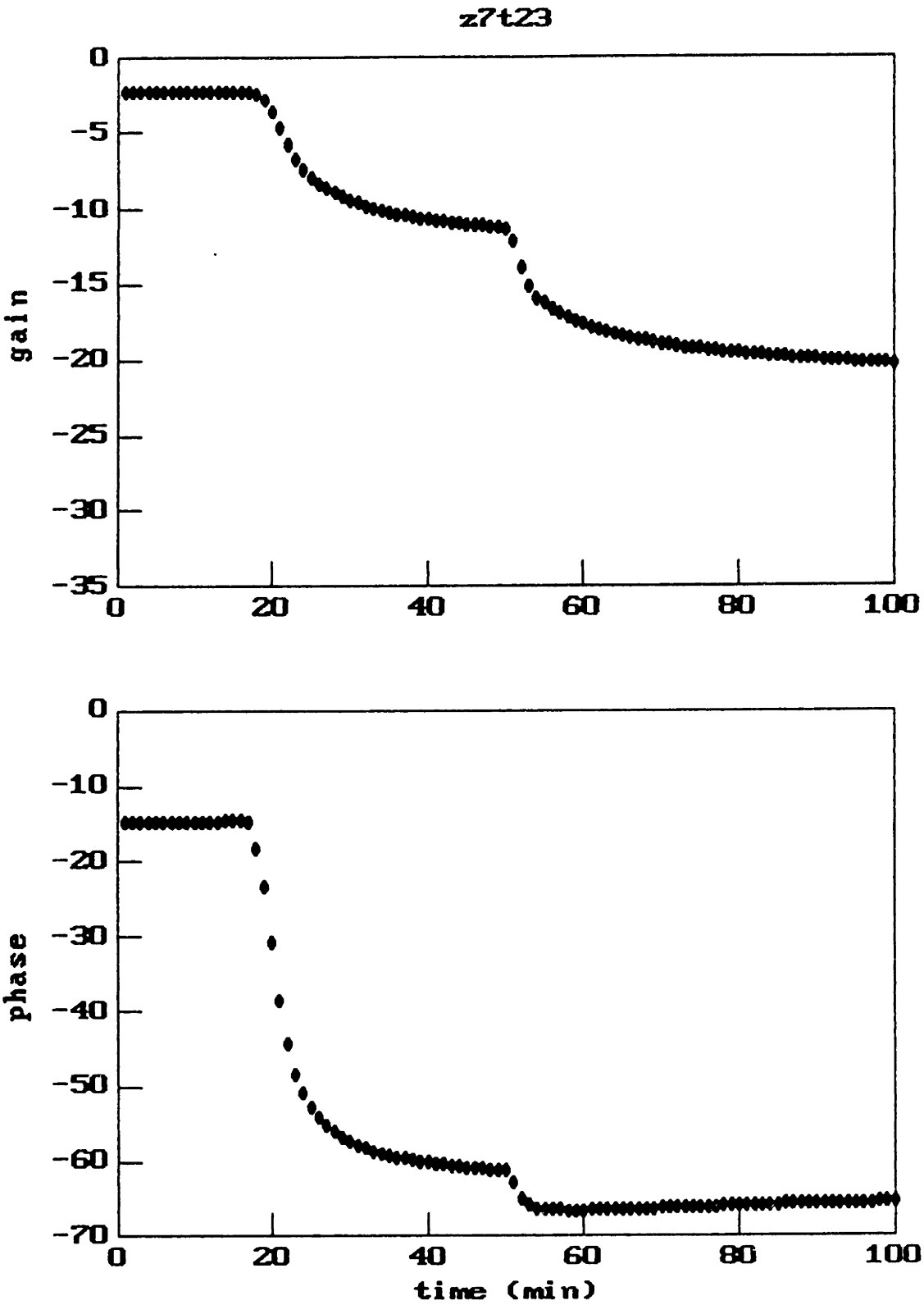


Figure 7.21: Transient Response at 10 kHz of Fresh Moisture Sensor Z7 in Unused Oil Going From 26 to 11 ppm Water.

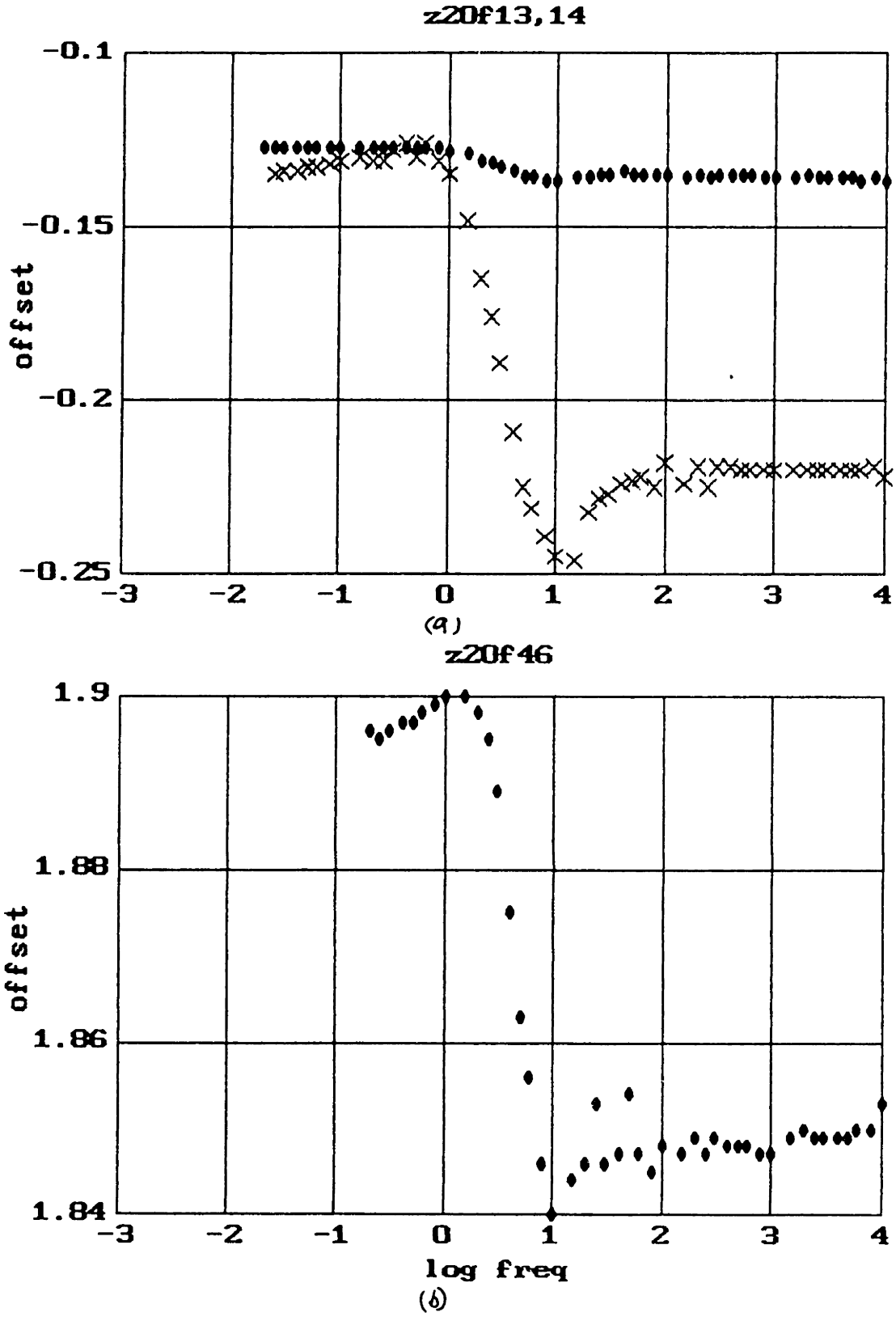


Figure 7.22: Measured Behaviour of DC Offset Voltage During a Frequency Scan for Moisture Sensor Z20 in Stirred, Unused Oil a) at 4.4 ppm Water: (•) – Attenuated Voltage to 8.8%, (X) – Full Voltage, 100% and b) at 6.9 ppm Water.

considerable average DC offset had no effect on the shape of the gain–phase response. At most, a slight vertical shift in the gain might be expected due to a possible error introduced into the correlation analysis by the DC offset. After the change in Z20 described above, its DC offset was very reproducible during further cycling of the oil temperature and moisture content while in the laboratory. Simultaneous conditioning of Z7 in the same oil did not result in this shift of DC offset voltage. The DC offset of Z7 was very reproducible throughout its entire lifetime except for the following problem.

Occasionally, the DC offset was associated with changes in the shape of the gain–phase response. This phenomenon occurred when the oil was so dry, and f_p so low, that the relaxation in the oil was beginning to be observed. An example of this is the response of Z7 in stirred, unused oil, having a moisture content of 4.7 ppm and at 47°C (Fig. 7.23). The DC offset changed rapidly at low frequencies, accompanied by a deviation from the master curve in the gain–phase response. This deviation prevents accurate calculation of f_p, making the moisture sensor unusable at this operating condition.

At the time of this example, Z7 was an extremely aged coating, having operated in oil of various temperatures for over a year. Consequently, the f_p's were generally much lower than when the coating was fresh, following the trend shown in Fig. 7.10. In addition, increasing the oil temperature shifted the relaxation in the oil to a higher frequency because of the increased oil conductivity. The combination of these two effects resulted in the overlap of relaxation in the oil and in the coating, altering the shape of the gain–phase response. When the oil temperature was lowered and/or the moisture content increased, the separation between the two relaxations increased, resulting in the master curve response. The moisture sensor was thus rendered unusable only until the condition of the oil changed. However, there is the risk of permanent damage if there is too much DC offset. The damage can occur to the FET transistor connected to the floating electrode due to excessive DC fields across the channel.

The observed changes in DC offset were associated with static charge buildup at the coating–oil interface. In many ways, the conditions leading to this effect resemble

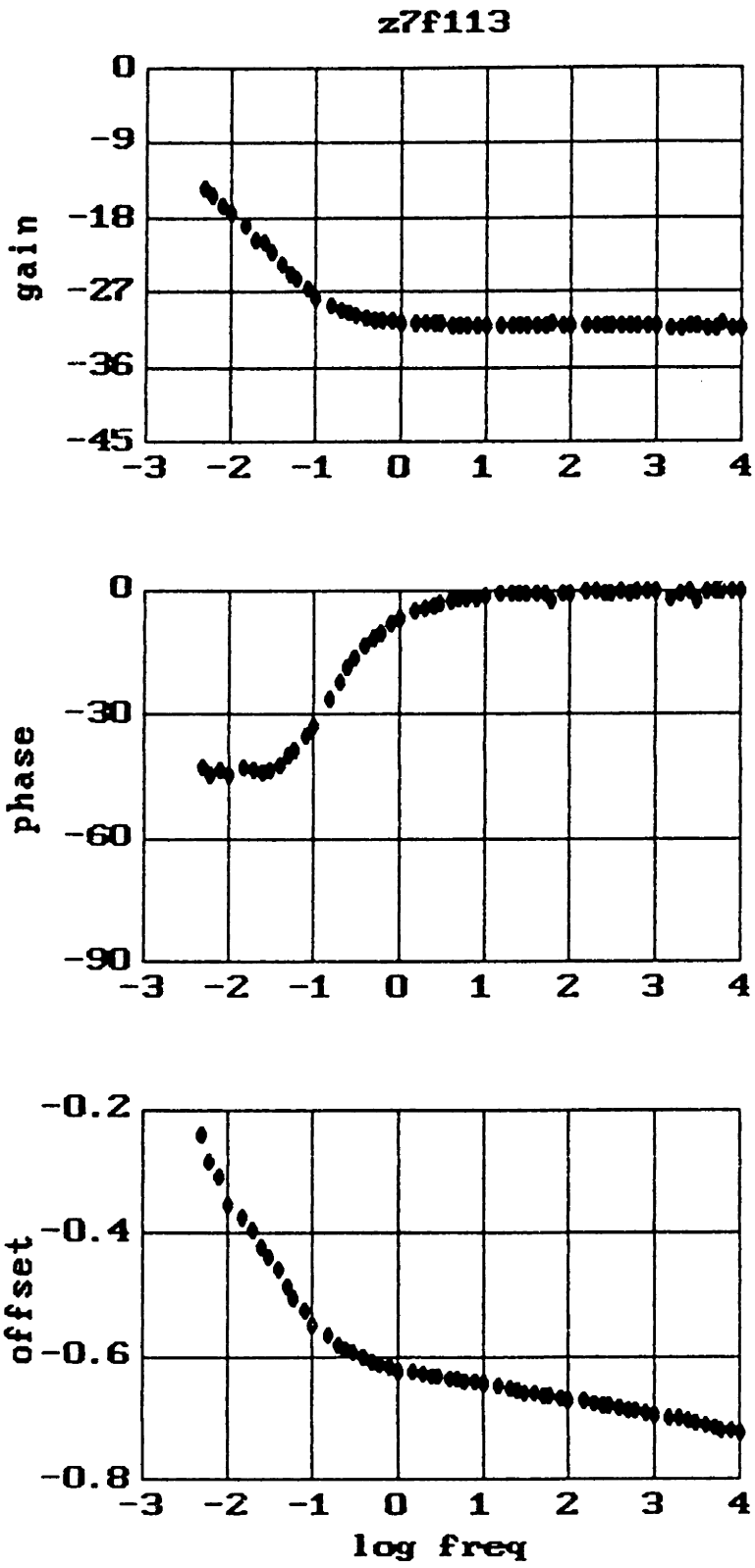


Figure 7.23: Effect of DC Offset on Measured Response of Aged Moisture Sensor Z7 in Stirred, Unused Oil at 4.7 ppm Water and 47°C.

the conditions attributed to causing static electrification problems in transformers – large volume flow rates of hot, dry oil. However, the problem can be avoided here by keeping the f_p of the coating from drifting to such low frequencies that the coating becomes as insulating as the oil.

7.2 Testing in Pilot Facility

7.2.1 Experimental Results

The purpose of this section is to describe preliminary tests of the moisture sensor, and primitive trend analysis schemes, in an on–line monitoring capacity. The sensor is mounted in the oil circulation loop of the test transformer (see Section 6.2.2). As the load on the transformer is altered, the temperature of the transformer changes. In particular, the change in temperature at the windings tends to drive the exchange of moisture between the paper insulation and the oil. Under normal operation, and at quasi–equilibrium (as defined in Section 6.2.2), this redistribution in moisture displays a reproducible dependence on temperature, as reflected by Fig. 6.2. This dependence can be used to predict the moisture content using

$$\log(\mathrm{mc}_{pred}) = \mathrm{C} - \frac{\mathrm{D}}{T} \tag{7.7}$$

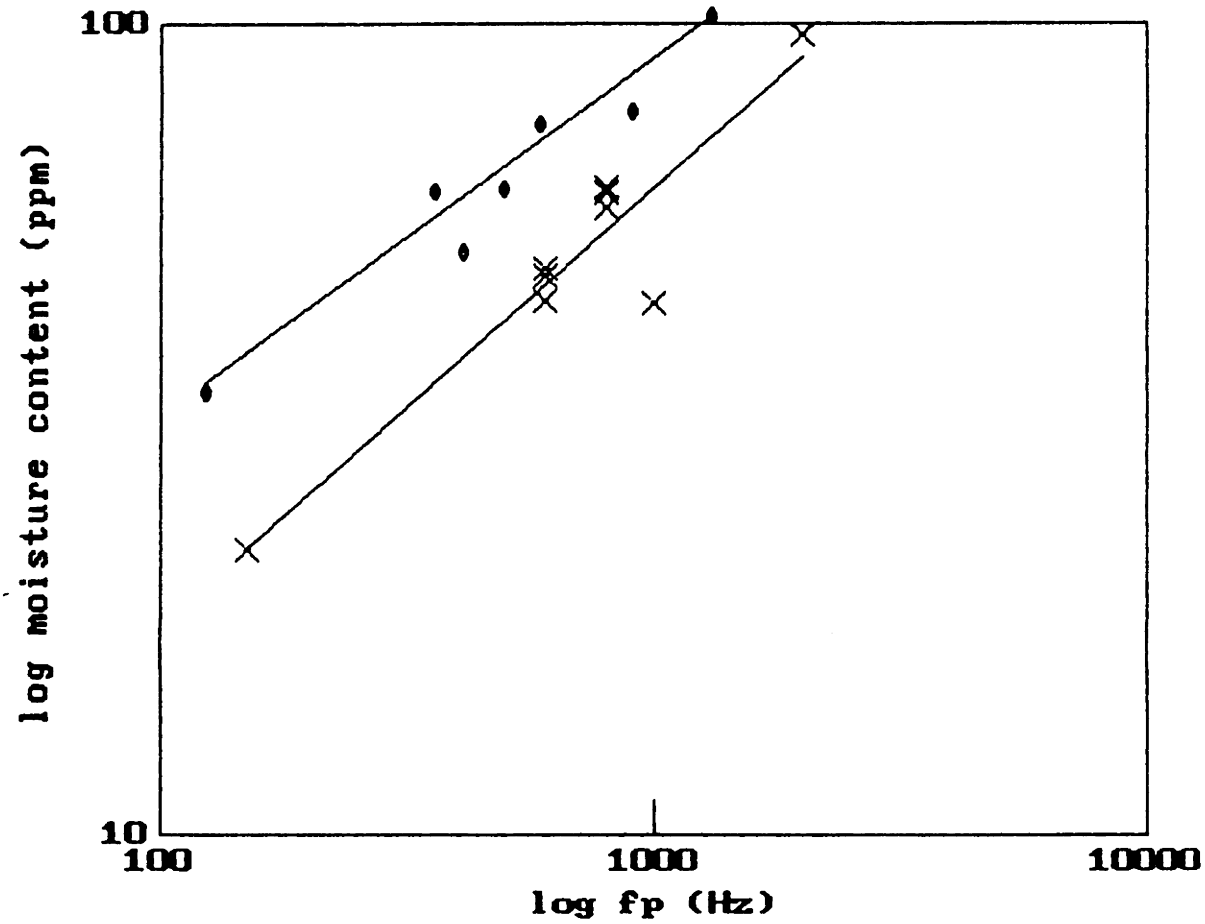
The moisture sensor also gives the moisture content of the oil, using a simple characterization such as (7.2). Taking the difference between moisture content as measured by the moisture sensor and as predicted under normal operation produces a residual. This residual may indicate that the transformer is deviating from normal operation. Further analysis, along the lines discussed in Section 6.2.1, may indicate a significant change in the moisture content of the paper insulation.

The software implementation of the data acquisition and analysis required for the on–line monitoring can be found in Appendix E in the program **wmod2.c**. This program, or water module, performed the following tasks daily. Beginning at midnight, a frequency scan of the moisture sensor was taken to re–establish the master curve.

A spline function was constructed to represent the master curve. After the frequency scan, the moisture sensor response was sampled, at a fixed frequency, on a periodic basis for the rest of the day. Typically, the sampling interval was once every 5 minutes. Using the spline function and the sampled data, an interpolated value of f_p was obtained and two moisture contents were estimated, one using (7.2) and the other using (7.7). The parameter values used in these equations were calculated from earlier sets of data and were stored in data files accessible to **wmod2.c**. Finally, a residual was computed from these two moisture contents.

The parameter values for (7.2) were calculated using data obtained from calibrating a moisture sensor (Z17, 2.1μm) with the moisture analyzer. This was performed using oil samples removed from the test facility (Fig. 7.24). Due to aging of the coating with time and temperature, this calibration changes, as can be seen by the shifts in Fig. 7.24. Initial moisture sensitivity data (crosses) represents calibration samples taken over the first 7 weeks of testing. The next set of data (dots) was taken over a four week period beginning 4 1/2 weeks after the end of the previous sampling period. The shift is not the same as those observed earlier, in Figs. 7.8–11. The difference lies in the fact that the earlier changes were recorded under isothermal conditions whereas, in the test facility, the change in temperature is driving the change in moisture content. Therefore, there are actually two components causing the change in f_p, as described by (7.4). Note that although there was a significant change in moisture sensitivity, the master curve remains the same before placement in the transformer and after operating for roughly 14 weeks in the oil circulation loop (Fig. 7.25).

Earlier testing of Z20 in the test facility also displayed the same shift in sensitivity over time (Fig. 7.26). Again, it was difficult to sort out the moisture sensitivity component from the temperature component. The significant decrease in moisture sensitivity, after being exercised in the transformer for 12 weeks, is shown in the difference between the circle and plus data in Fig. 7.11. The constancy of the master curve before and after operating in the test facility is evidenced by the small changes in Fig. 7.14. This sensor was removed from the facility due to intermittent malfunctions. As discussed in Section 7.1.8, Z20 had a large DC offset voltage prior to installation



$$(X) - \log(mc) = 0.19 + 0.54\log(f_p) \quad (5/1/87\text{–}6/19/87)$$

$$(\bullet) - \log(mc) = 0.64 + 0.44\log(f_p) \quad (7/22/87\text{–}8/17/87)$$

Figure 7.24: Moisture Sensitivity of Aging Moisture Sensor Z17 in Oil Circulation Loop of 50 kVA Test Transformer.

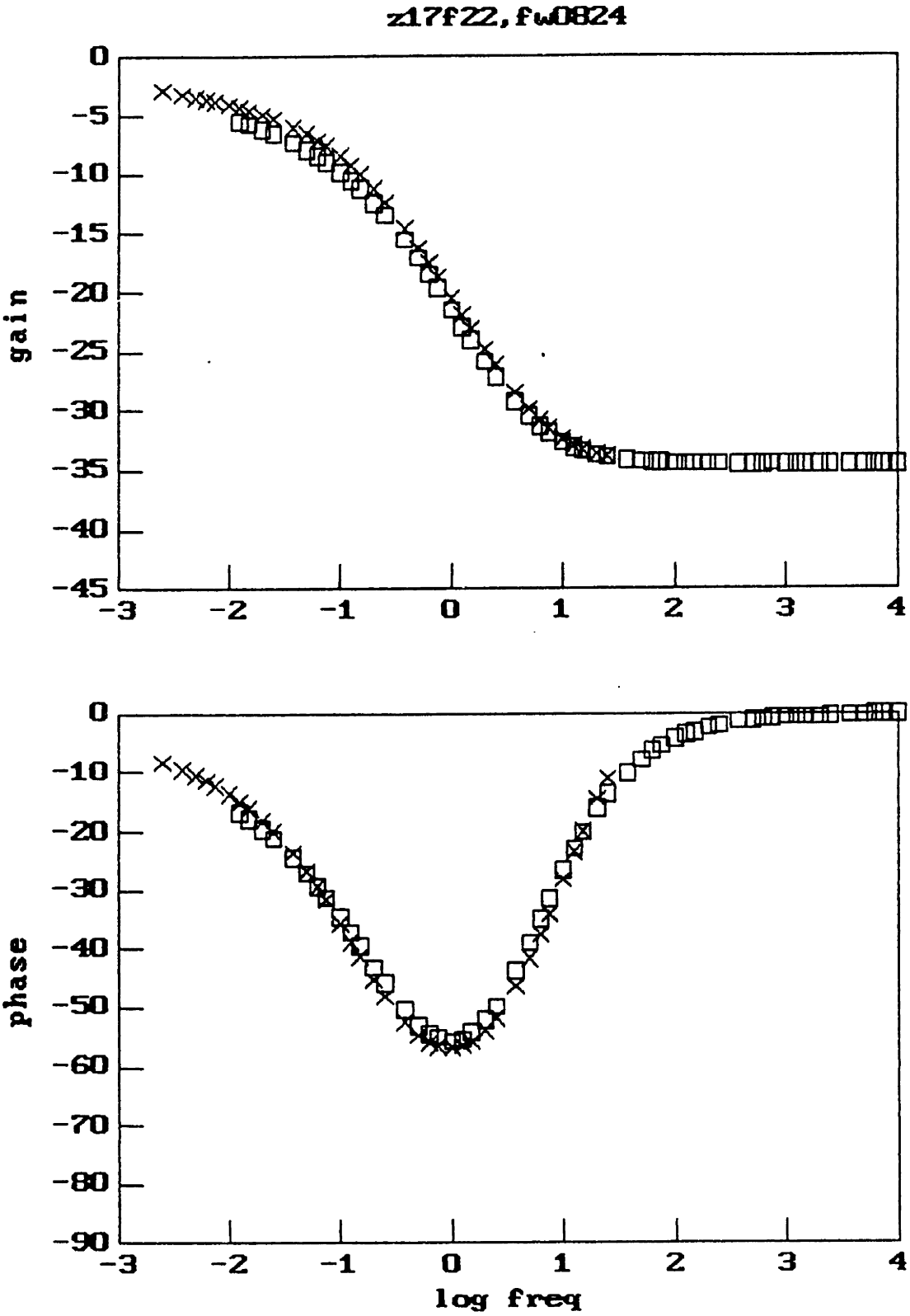


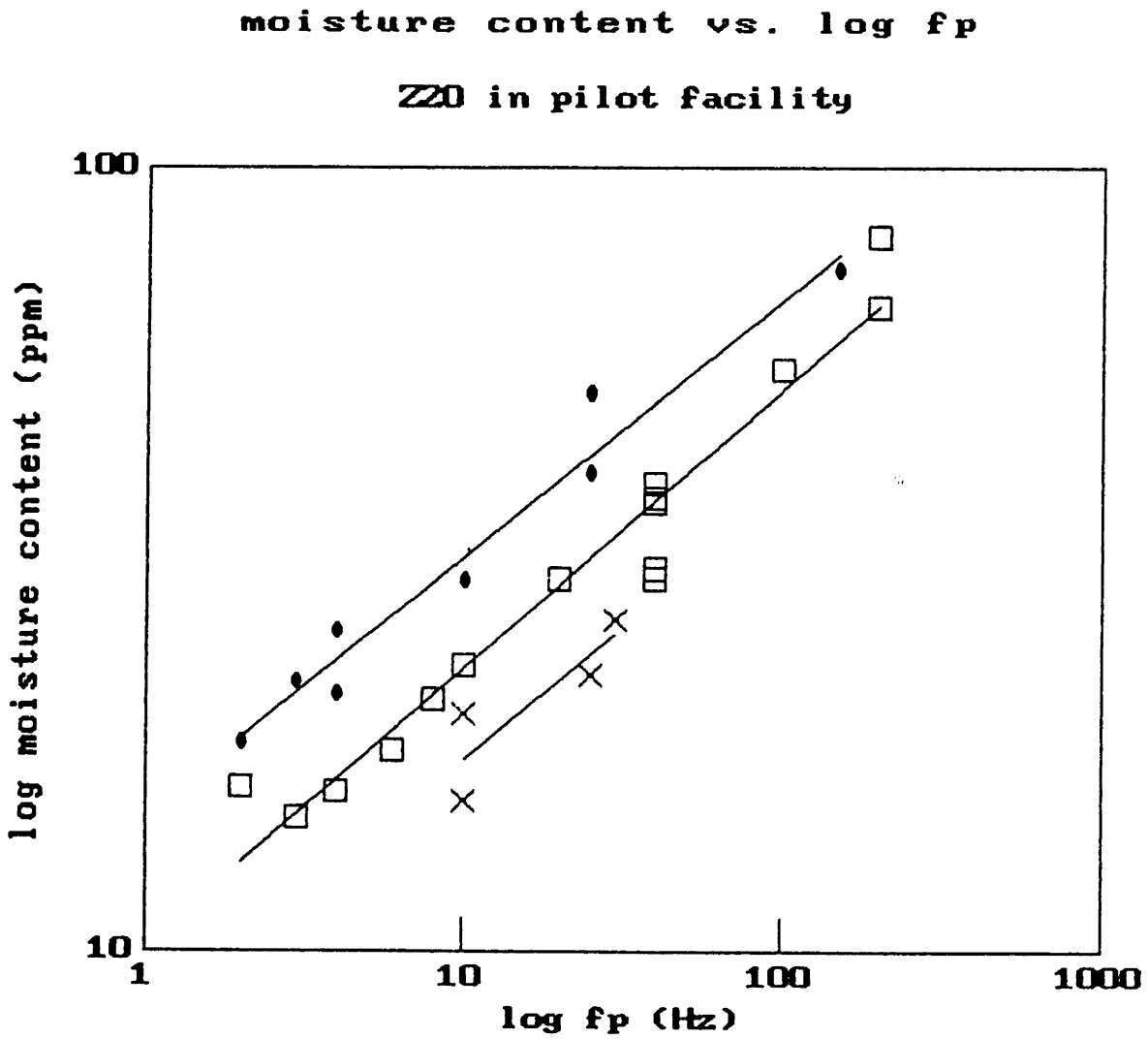
Figure 7.25: Measured Master Curves of Moisture Sensor Z17 Prior to Installation (□) and 14 Weeks After Installation (X) in Oil Circulation Loop of 50 kVA Test Transformer.

in the test facility. It is believed that the extreme flow rates the moisture sensor was exposed to (~8 gal/min), in combination with this large DC offset, resulted in damage to the FET transistor connected to the floating electrode. After removal, further testing in gently circulating oil was possible, producing the rest of the results shown in Fig. 7.11. The intermittent malfunction was associated with static charge buildup because the problem could be alleviated by disconnecting and reconnecting either the sensor to the interface box, or the cable connecting the interface box to the microdielectrometer.

An example of the on–line monitoring using Z17 is shown in Fig. 7.27. The gain, phase, offset and temperature (at the sensor) are sampled at 400 Hz every 5 minutes over the span of one day with midnight corresponding to $t = 0$. At this period in time, the transformer excitation current was being cycled on and off daily. The amplitude of the excitation current for this particular scan was 210 A. The sensor temperature (lower right), reflecting the ambient oil temperature flowing by the sensor, is decreasing prior to the application of the excitation current because the transformer is still cooling off from the previous day's cycling. At a little after $t = 500$ min. the temperature begins to increase, reflecting the application of the excitation current. In the evening the current is turned off, resulting in a delayed decrease in temperature. As is evident from the plot, the transformer is not in thermal equilibrium with this daily cycling.

The gain and phase in Fig. 7.27 (upper left and upper right respectively) appear to mimic the time variation in the temperature. The response is being monitored at 400 Hz, which is on the low frequency side of the phase peak ($f < f_p$). As the temperature decreases, the oil becomes drier, shifting the response to lower frequencies (f_p decreases). Therefore, both the gain and phase become more negative initially. With the temperature increase the oil absorbs more moisture from the paper resulting in an increasing f_p, gain and phase. The offset (lower left), for some reason, also appears to follow the change in temperature.

In the plot of f_p (upper left of Fig. 7.28), the discreteness observed is a result of the tolerance used in the interpolation routine. The predicted moisture content using f_p in (7.2), with A=.19 and B=.54, is shown in Fig. 7.28 (upper right). These parameter



(X) – $\log(mc) = 0.92 + 0.329\log(f_p)$ (2/11/87–2/14/87)

(□) – $\log(mc) = 1.01 + 0.353\log(f_p)$ (2/15/87–3/16/87)

(•) – $\log(mc) = 1.18 + 0.327\log(f_p)$ (3/18/87–4/10/87)

Figure 7.26: Moisture Sensitivity of Aging Moisture Sensor Z20 in Oil Circulation Loop of 50 kVA Test Transformer.

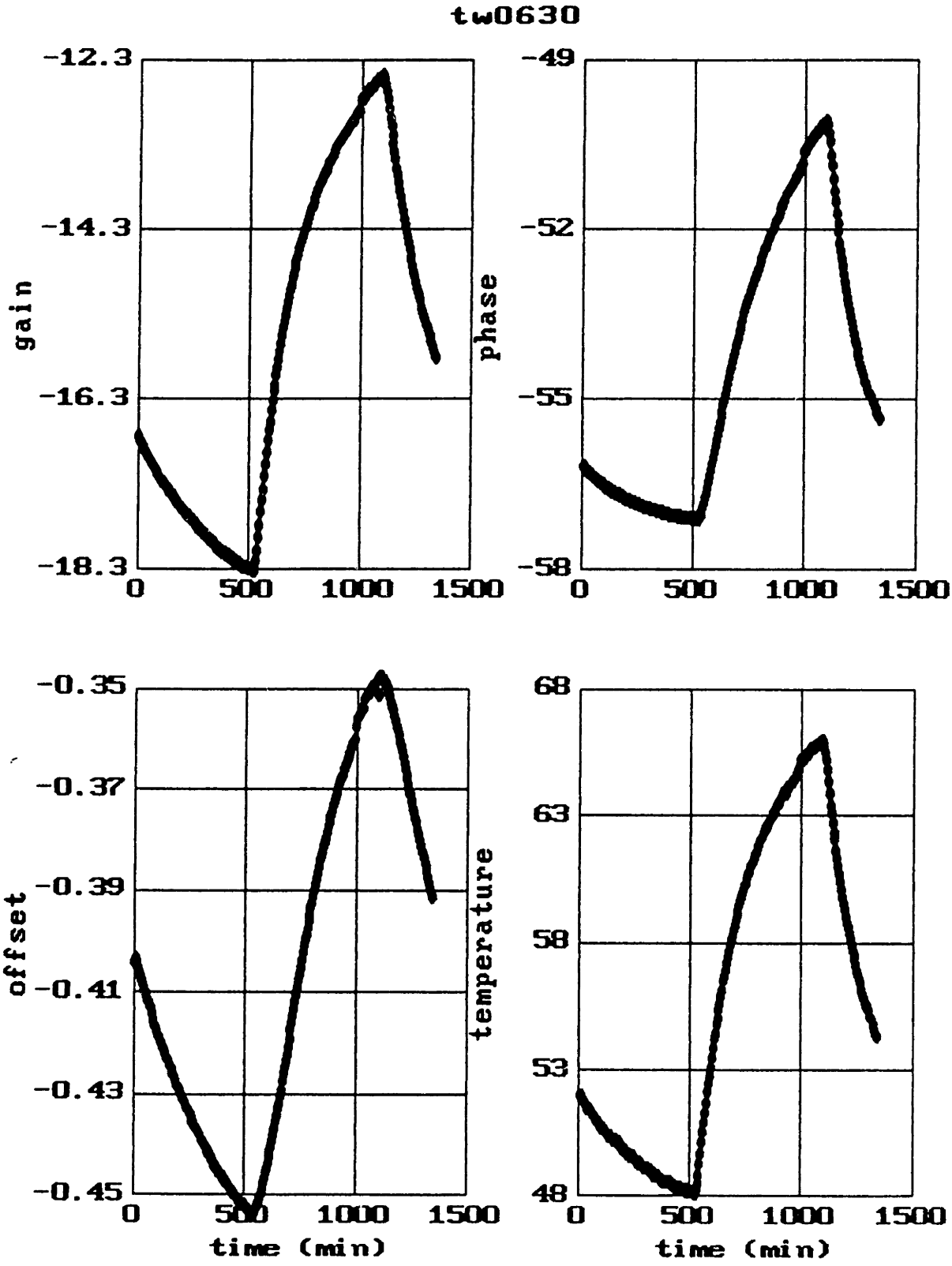


Figure 7.27: On-Line Monitoring of Gain, Phase, Offset and Temperature of Moisture Sensor Z17 at 400 Hz in Oil Circulation Loop of 50 kVA Test Transformer. Current Loading Cycled Between 0 and 210 A.

values are taken from the crosses in Fig. 7.24 – prior to further aging of the coating. The expected moisture content using (7.7), the sensor temperature, and C=5.443 and D=−1201, is also shown in Fig. 7.28 (lower left) along with the resulting residual (lower right). The values for C and D were obtained from the data, shown in Fig. 6.2, that was available at that time. It is observed that the residuals are very small, indicating normal changes in the moisture content with temperature.

Another set of data, taken 5 weeks later, is shown in Figs. 7.29 and 7.30. Here, the transformer had been operating at an excitation current of 150 A long enough to be in steady state. At a little after $t = 750$ min. the excitation current was increased to 200 A. The glitch at $t = 500$ min. was a result of turning the transformer off so as to remove oil samples for gas and moisture analysis. Again the measured response, monitored at 400 Hz, follows the changes in temperature very closely. However the residuals are very large this time. The parameter values used in (7.2) were different, A=.74 and B=.38, reflecting the aging of the moisture sensor (only 4 of the 7 dots in Fig. 7.24 were known at the time of the experiment resulting in a different best fit line), but the parameter values used in (7.7) were the same as before. The change in residual accompanying the change in excitation current is slightly larger than before, 6 ppm vs. 4 ppm, but there is an overall shift of 8–10 ppm in the average residual. This shift most likely reflects changes in the sensitivity parameters of the moisture sensor due to aging. Oil samples from the transformer, measured by the moisture analyzer, produced moisture content values that correlated with the normal operation of the transformer, as indicated by the upper set of datapoints in Fig. 6.2.

7.2.2 Discussion

There are many flaws in this implementation. The problem with the moisture sensor is the unstable nature of the coatings, casting some doubt on the correct parameter values for the moisture sensor beyond, roughly, a one or two week time period. On a monthly time scale, recalibration of the present moisture sensor would be a necessity. Due to the double valued nature of the phase there is also a problem interpolating f_p when the

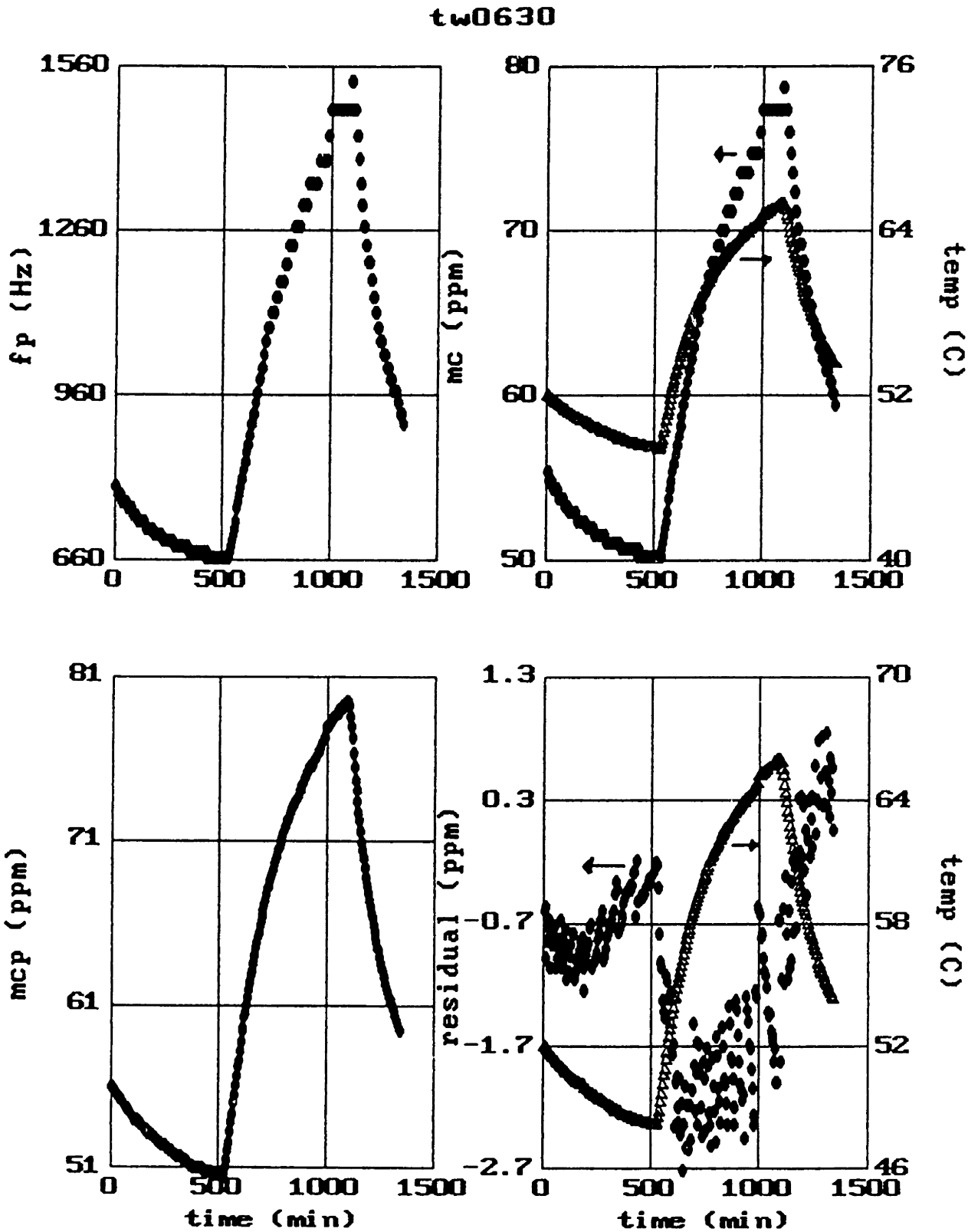


Figure 7.28: f_p, Measured and Predicted Moisture Content, and Residual Using Data of Fig. 7.27 and Equilibrium Moisture Model.

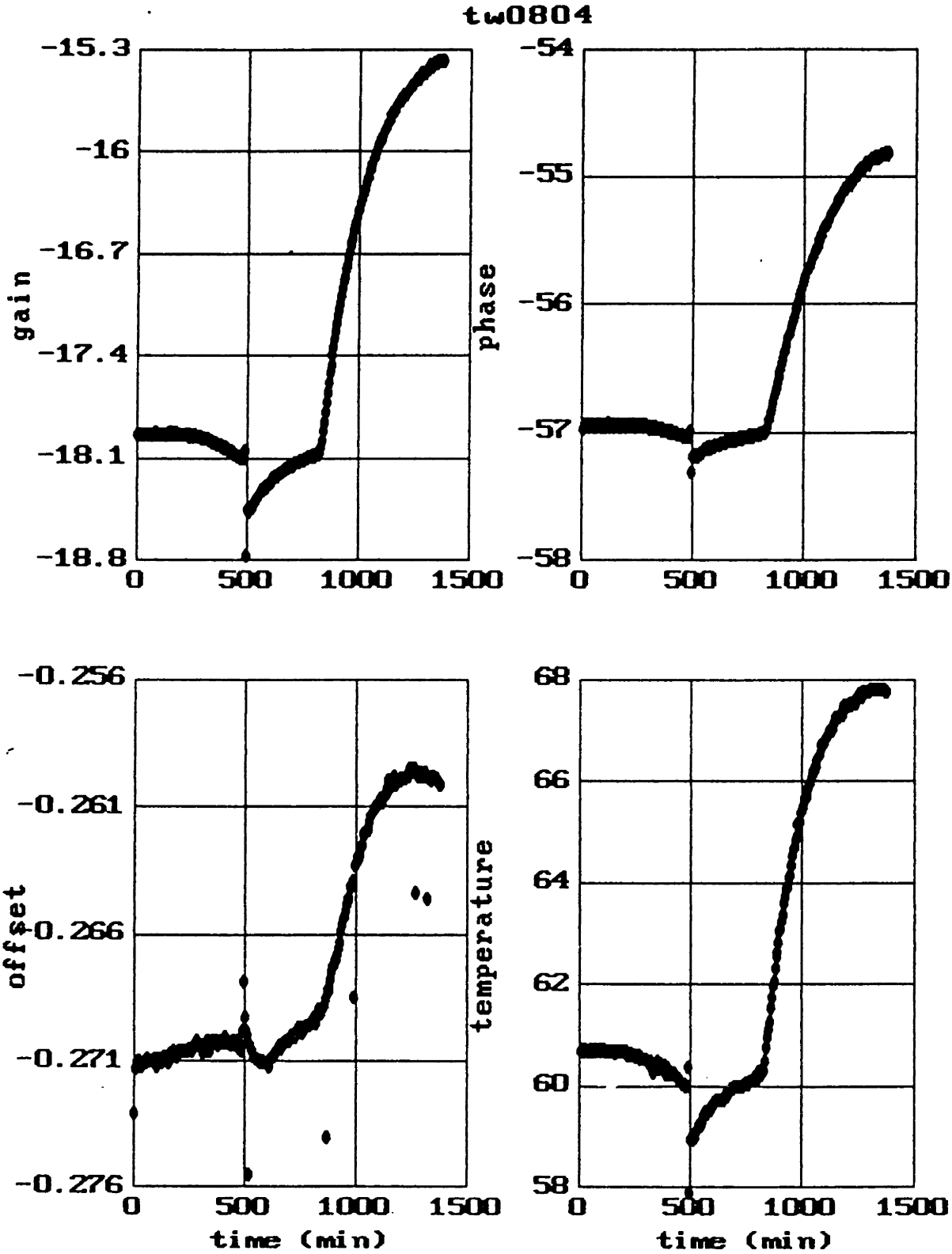


Figure 7.29: On-Line Monitoring of Gain, Phase, Offset and Temperature of Moisture Sensor Z17 at 400 Hz in Oil Circulation Loop of 50 kVA Test Transformer. Current Loading Increased From 150 to 200 A.

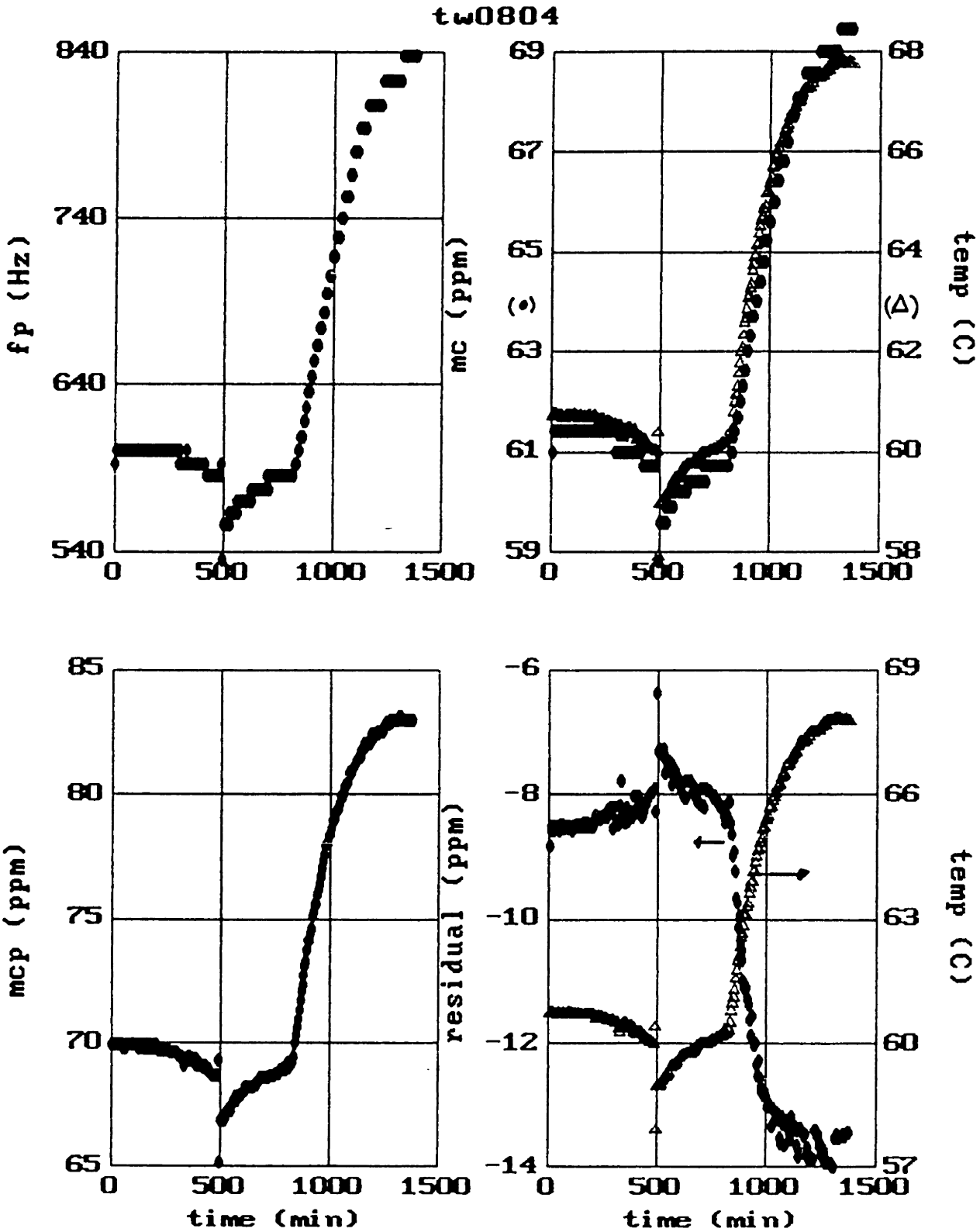


Figure 7.30: f_p, Measured and Predicted Moisture Content, and Residual Using Data of Fig. 7.27 and Equilibrium Moisture Model.

actual frequency is close to f_p. This problem can be avoided by automatically shifting the monitoring frequency so as to keep the ratio (f/f_p) constant. As the moisture content increases, f_p increases, thus the monitoring frequency f would also increase. The opposite behaviour would occur as the moisture content decreased.

As to the modeling issues, the simplified models used to compute the moisture content need to be updated. Characterization of the moisture sensor, presently using (7.2), should be expanded to include the temperature change recorded by the sensor, as in (7.4). It remains to be justified whether a quasi-equilibrium model for describing the moisture distribution in the transformer, using (7.7), is valid. A more complex model, including lagged temperatures, may be required to adequately simulate the dynamics driving the moisture exchange. It also is more reasonable to be using the winding temperature in (7.7) instead of the oil temperature as measured by the moisture sensor. However, increasing the complexity of these models should be governed by a determination of how accurate the estimates need to be.

Nevertheless, this work represents an important step in taking real-time moisture data from a transformer and attempting to correlate the observed behaviour with basic models of "normal" operation.

Chapter 8

Moisture Sensor – Interpretation of Results

The experimental results presented in Chapter 7 established the sensitivity of the coatings to absolute moisture content in the oil. The issues in this chapter are understanding the physical mechanism accounting for this sensitivity, interpreting the dielectric response of the coatings and examining the problem of coating stability. Much of the discussion in Sections 8.1 and 8.3 draws upon related work reported in the literature. Film morphology and its connection with the moisture permeation and electrical parameters of the film is presented in Section 8.1. Section 8.2 demonstrates the utility of the continuum model, and associated parameter estimation routines, in judging the validity of various electrical conduction models as applied to the plasma deposited bromobenzene material. Following the rejection of several different models relying on an ohmic conductivity, the dispersive nature of the coatings are revealed. Examples from the literature of other materials displaying similar dispersive properties, having responses that can be normalized with respect to changes in RH and temperature, are given. The change in moisture sensitivity with aging of the coatings is examined in Section 8.3. Examples are taken from the literature of techniques used during, and after, the plasma deposition process to improve stability.

8.1 Film Morphology – Effect On Moisture Permeation and Electrical Properties

The structure of plasma deposited thin films is a strong function of the operating conditions during deposition, as discussed in Appendix C. Coatings can vary from rigid to oily to powdery films, with combinations possible as well. SEM photographs of the witness strips for both thick and thin coatings indicate the existence of a biphasic structure, with particles embedded in a homogeneous film matrix. Fig. 8.1a shows the surface of the witness strip coated alongside chip Z22, 3.5 μm thick coating. The size scale is indicated by the bar at the lower right, with the marking of 20 μm underneath. A magnified view of a section of this coating is presented in Fig. 8.1b, the bar width at the lower right indicating 4 μm. Particles both spheroidal and box–like in shape appear to be both on the surface of, and embedded within a rigid film. The film is judged to be rigid, not oily, based on the retention of its physical and electrical integrity after physical handling and immersion in hot oil. Particle dimensions range from 2 μm down to submicron. A large particle of about 4 μm, located in the background and slightly off center to the lower left, appears to be a grouping of smaller, 2 μm sized particles.

Though no samples were made of the cross–section, it is believed that the particles exist throughout the film. This belief is based on 1) the observation of particles, besides the ones clearly on the surface, that appear to be embedded within the film and 2) similar observations in the literature. For example, by pyrolyzing plasma polymerized hexamethylcyclotrisilazane (HMCTSN) films for various durations, Grebowicz, et al. [95] revealed that the particles, in his case, were distributed throughout the entire film and were not just at the upper surface. The particle diameters in these HMCTSN films ranged from 1000–2500 Å.

The effect of the particle density on moisture sensitivity may be inferred by first comparing the SEM photograph of Fig. 8.2a to Fig. 8.1a. In Fig. 8.2a the witness strip examined had been coated alongside chip Z3, 0.6 μm thick coating. The scaling in Fig. 8.2a is given by the bar at the bottom of the picture, its length representing

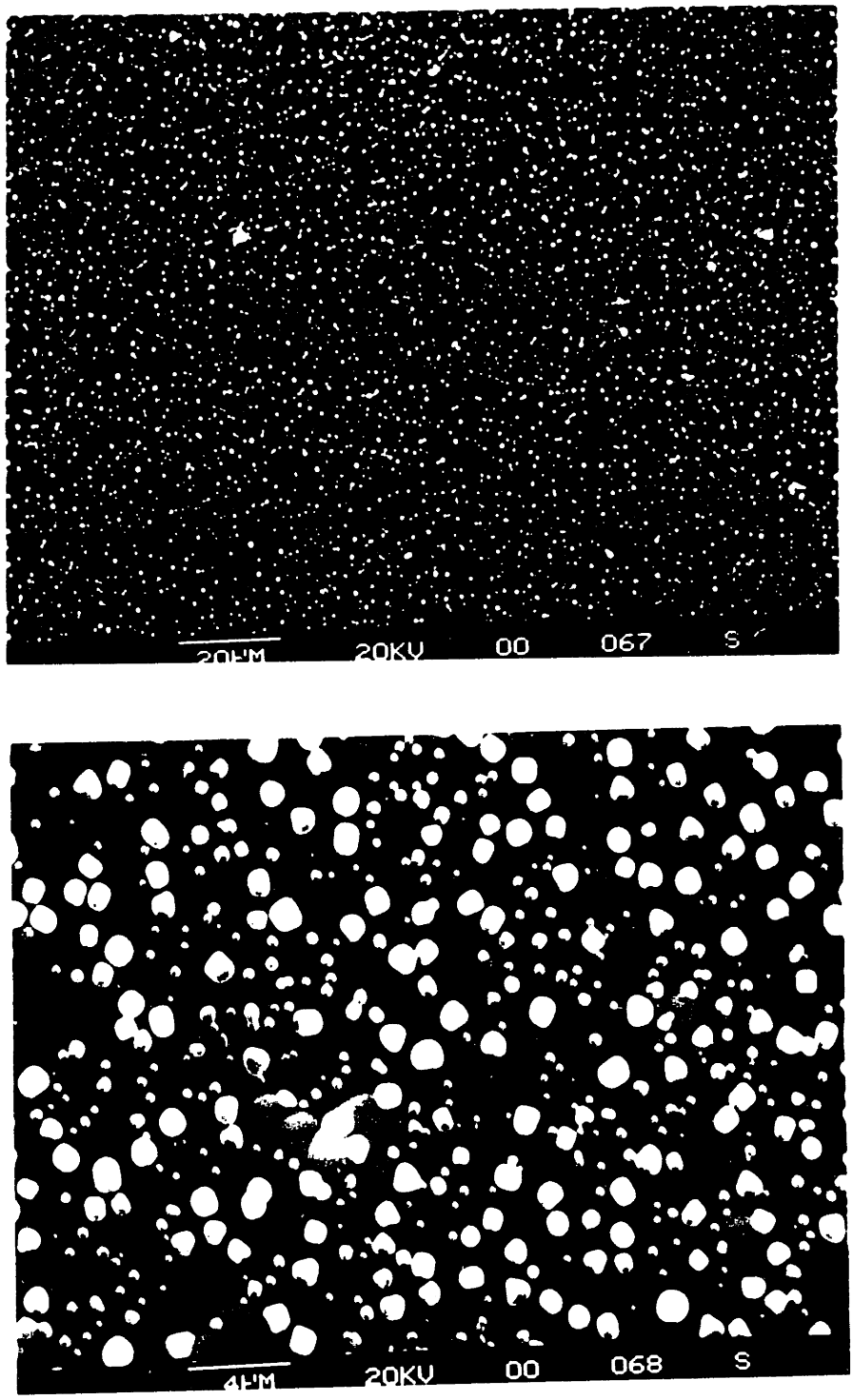


Figure 8.1: SEM Micrograph of Witness Strip Coated Alongside Moisture Sensor Z22. a) Scale 15 mm = 20 μm (650X), b) Scale 15 mm = 4 μm (3200X).

100 μm. A magnified view of this coating is shown in Fig. 8.2b, the primary object in this picture being one of the more spheroidal particles of Fig 8.2a. Here, the scaling represented by the bar at the bottom is 1 μm. The particle density in Fig. 8.2a is an order of magnitude less than the density in Fig. 8.1a. The main fabrication difference, besides the coating thickness, is the RF power level during deposition – 150 W for Z3 vs. 25 W for Z22. Experimentally, the moisture sensitivity of Z3 was much lower than all the other coated chips (Fig. 7.8).

For completeness, SEM photographs of a witness strip coated alongside chip Z7 is also shown in Fig. 8.3. The surface of this coating looks drastically different from the previous photographs. Large particles (5–20 μm in width) occupy a large part of the surface area. These particles appear to consist of two or more smaller particles agglomerated together (see Fig. 8.3a). This coating was produced by plasma depositing for a longer period of time than the previously shown coatings (see Table 6.1). Consequently, this coating was relatively thick, 6–7 μm. The moisture sensitivity of Z7, and Z20, was studied extensively in Chapter 7 and found to be just as sensitive as other chips having thinner coatings, such as Z22 (3.5 μm).

It is believed that the large particles on the coating surface, Figs. 8.1a and 8.3a, are not relevant to the moisture sensitivity or the electrical conduction process. Primarily, the justification for this conclusion is that the shape of the frequency responses obtained with thinner coated chips lacking these large surface particles, such as Z22, are very similar in appearance to the responses of Z7 and Z20. Also, given the isolated nature of these large particles, their contribution to the net conductivity of the material would be highly dependent on the electrical properties of the film between the particles. This scenario is not a satisfactory one, as will be shown in Section 8.2.

Instead, it is believed that it is the particles embedded within the film that are important. These particles can be seen in the background in Fig. 8.3a but are more clearly seen in the magnified view of Fig. 8.3b. They vary in size from roughly 2 μm to submicron. It is thought that these embedded particles increase the surface area and porosity within the film, resulting in greater moisture permeation, and thus, higher moisture sensitivity. This conclusion is borne out by the lower moisture sensitivity

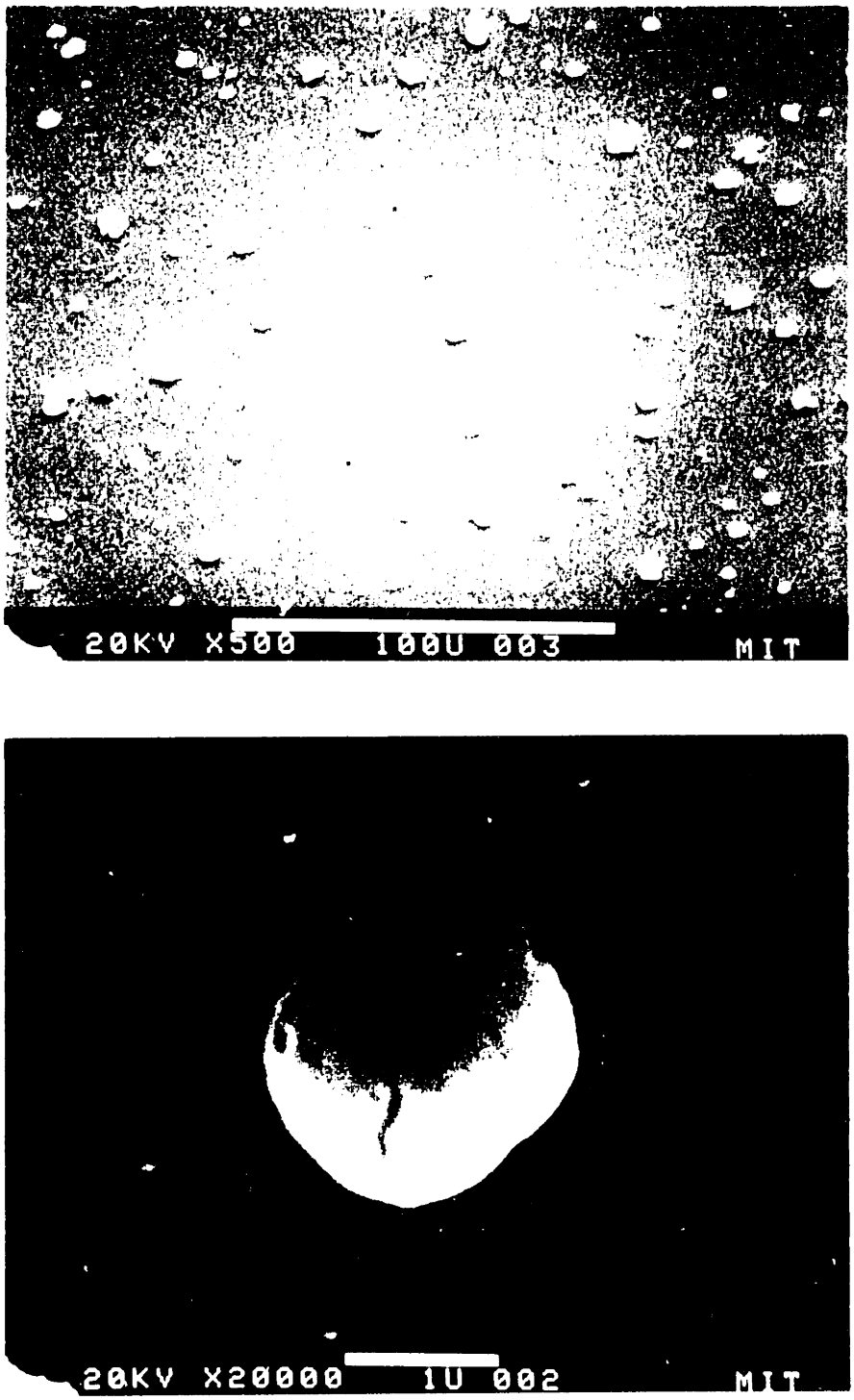


Figure 8.2: SEM Micrograph of Witness Strip Coated Alongside Moisture Sensor Z3 a) Scale 52 mm = 100 μm (500X), b) Scale 11 mm = 1 μm (20,000X).

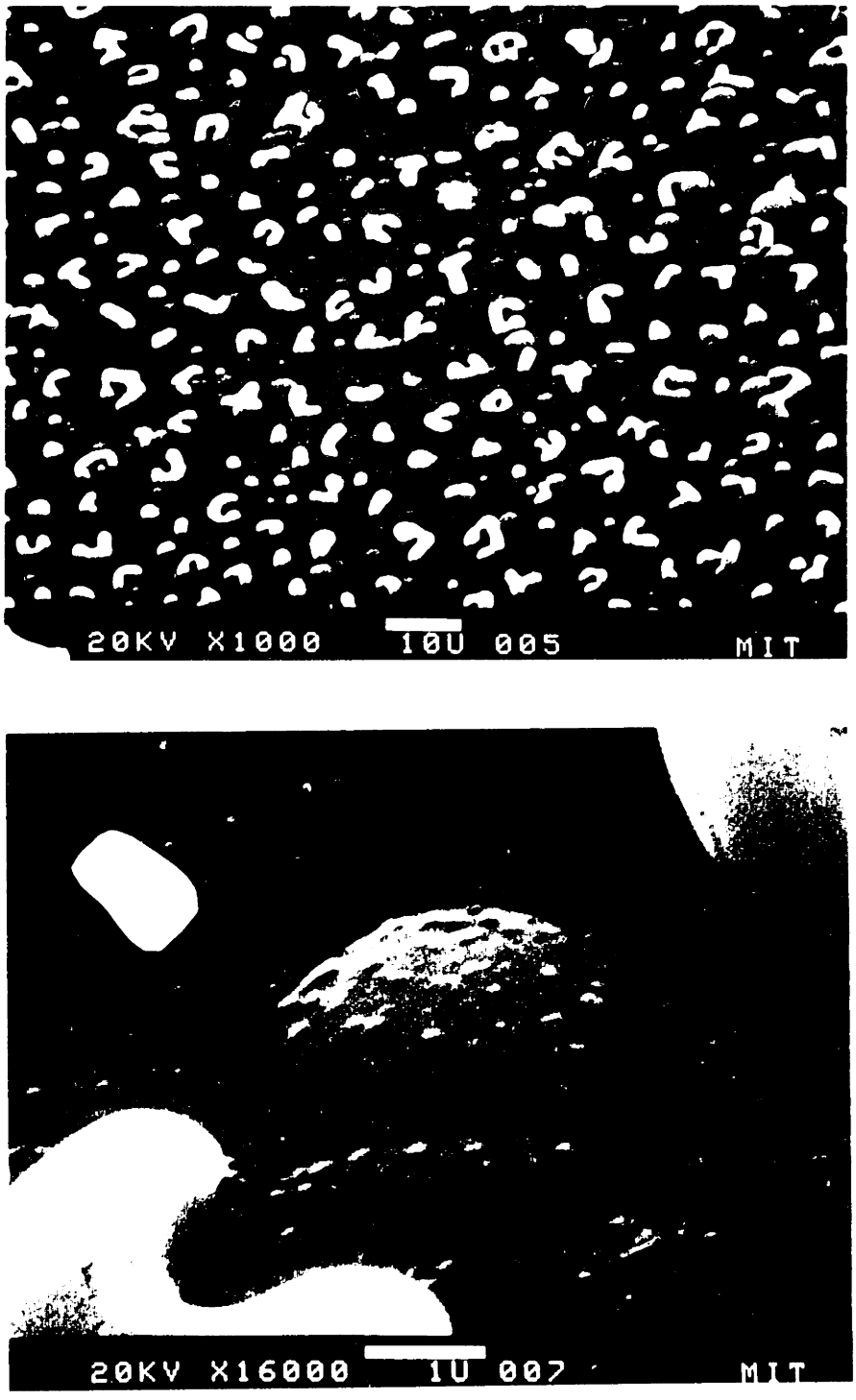


Figure 8.3: SEM Micrograph of Witness Strip Coated Alongside Moisture Sensor Z7 a) Scale 10 mm = 10 μm (1000X), b) Scale 16 mm = 1 μm (16,000X).

of Z3, having a coating with low particle density. This conclusion also implies that moisture sensitivity is not dependent on coating thickness, *per se*, but on the voidage and porosity of the coating. The moisture sensitivity plot of Fig 7.8, coupled with the observation of similar high particle density coatings for several other moisture sensors of varying coating thickness, demonstrates this lack of dependence on coating thickness.

The relationship between particle density and moisture permeation was also studied by Wertheimer, et al. [96]. Using hexamethyldisiloxane (HMDSO) and hexamethyldisilazane (HMDSN) monomers, the particle density was varied by changing the substrate temperature during deposition. Increasing the temperature from 25°C to 400°C resulted in the almost complete elimination of particles. This effect was reproduced by increasing the power level, as observed in this thesis also. In addition, the increase in either power level or substrate temperature resulted in an increasingly inorganic film (decreasing carbon content relative to the silicon, hydrogen and oxygen contents). Finally, these changes were accompanied by an order of magnitude decrease in the moisture permeation coefficient from 5 to $.5x10^{-8}$standard cm^3–cm/cm^2–s . The increase in film density was only roughly 10%. The decrease in moisture permeation was attributed to the reduced ability of the films to swell as they became more inorganic and more rigid.

The effect of film morphology on moisture permeation was also studied via measurements of the loss tangent and capacitance of thin, plasma polymerized HMDSO films by Ramu, et al. [97]. Again, increasing the substrate temperature (during deposition) to 400°C resulted in little, if any particle formation. The dispersion in the loss tangent was shown for samples prepared at various substrate temperatures as a function of RH. Measurements were made in the frequency range of 0.01-100 Hz. For a given sample, the peak loss shifted to higher frequencies with increasing RH. The magnitude of the loss appeared to decrease by a factor of three as the substrate temperature was increased from 25° to 250°C but the sensitivity, or change in f_p as a function of RH, increased with substrate temperature. The change in capacitance was a linear function of RH with a slope that decreased by an order of magnitude with

the change in substrate temperature. Again, the picture that emerged was one of regions of lower density that were surrounding the microspheres, allowing permeation of moisture. Furthermore, this moisture permeation was decreased by reducing particle formation at higher substrate temperatures.

An important observation to be made here is that the two studies characterizing film behaviour were performed in air. No study was ever found describing behaviour of these films in liquids, much less transformer oil.

8.2 Dielectric Response of Plasma Deposited Bromobenzene Films

8.2.1 Attempts to Model the Master Curves

The following sections discuss the many attempts that were made to predict the master curve response of the coated microchip moisture sensors using standard models of dielectric phenomena. After distinguishing between interfacial versus bulk loss mechanisms, and dipolar versus charge carrier loss mechanisms, various models utilizing ohmic conduction (using uniform or heterogeneous mediums) are considered and rejected. Finally, a mathematical fit to the measured response is obtained using a power law dispersion relation, possibly indicating the physical mechanism of a hopping conductivity.

Interfacial Phenomena

The first possibility considered is that the form of the frequency response was dominated by interfacial phenomena at the electrodes due to polarization, charge injection such as that occurring at a Schottky barrier, or electrochemical reactions catalyzed by the adsorbed moisture. This interpretation was somewhat appealing in view of the facts that there seemed to be no discernible dependence of moisture sensitivity on coating thickness and that many of the observed master curves displayed similar S–shaped gain functions and broad phase peaks.

A typical model of the coupling between the electrodes through the coating would have a parallel RC element representing the interfacial impedance between the metal electrode and the bulk in series with another parallel RC network representing the bulk coating. For the occurrence of a master curve with a significant interfacial impedance one of two scenarios must take place: 1) the interfacial impedance dominates or 2) both impedances change proportionately with external variables. The first scenario implies that the bulk impedance is insignificant relative to the interfacial impedance. This is inconsistent with the experimental observation that for $f \gg f_p$ the response is low gain and zero phase – indicating either an insulating bulk or unreasonably large electrode capacitances. As for the second scenario, the experimental evidence demonstrated independence in curve shape with respect to AC electric field strength and temperature variations, casting considerable doubt on the affect of the interfacial impedance. The behaviour of charge injection and electrochemical reactions tends to exhibit very non-linear dependencies on temperature and field strength. Significant variations in these parameters would induce large changes in the interfacial impedance without changing the volume impedance. This would not satisfy Theorem 3.1 and thus not reproduce the master curve response observed experimentally.

Regarding the possibility of electrode polarization, the consequence of gains, at low frequency, approaching unity (0 dB) implies little or no polarization. Actually, the gain at low frequency usually approaches –.2 and not 0 dB. However, this small deviation can arise from experimental errors due to: 1) the slight resistive drop in the coating, 2) the slight mismatch in the on–chip FETs (see Section 1.4.2), and 3) significant DC offset voltages (see Sections 5.4.1 and 7.1.8). For these cases, given the small deviations from unity gain, it is judged that the experimental errors are more significant than electrode polarization effects. Examples of significant electrode polarization when using the microchip sensor can be found in the work of Day, et al. [99] in the monitoring of epoxy resin cures.

Thus, it is believed that the observed response was not due to any significant contribution by interfacial impedances.

Bulk Phenomena

Dipolar Loss

The form of dielectric dispersion due to a simple Debye dipole loss model is

$$\begin{aligned} \epsilon' &= \epsilon_\infty + \frac{\epsilon_{st} - \epsilon_\infty}{1 + (\omega\tau)^2} \\ \epsilon'' &= \frac{\epsilon_{st} - \epsilon_\infty}{1 + (\omega\tau)^2}\omega\tau \end{aligned} \tag{8.1}$$

where ϵ_∞ is the high frequency optical dielectric constant, ϵ_{st} is the static dielectric constant and τ is the dielectric relaxation time [98, p. 18]. At low frequencies $\epsilon' = \epsilon_{st}$ and $\epsilon'' \rightarrow 0$, resulting in a purely capacitive coupling. For a coating exhibiting only dipolar losses, this would imply a low frequency gain significantly different from unity and a phase of zero. However, the low frequency gain did approach unity (within the aforementioned experimental errors), inconsistent with the dipolar loss model.

This low frequency unity gain is consistent with losses due to charged carrier conduction. With charge carriers, as $\omega \rightarrow 0$, the resistive coating essentially 'shorts' the electrodes, providing an almost lossless coupling.

Uniform Ohmic Conduction

The simplest approach was to model the coating as a uniform ohmic conductor. A comparison between the predicted response for a 6.8 μm thick ohmic conductor in oil and a measured master curve for Z7 (fresh) is shown in Fig. 8.4 and for a 2.2 μm thick coated chip (Z24) in Fig. 8.5. Note that the exact value of the conductivity used for the predicted response is irrelevant here as the shape of the curve is independent of conductivity. The predicted gain rises more steeply as frequency is decreased, and the phase peak is narrower and much larger than the experimental data. Hypothesizing that the coating thickness was nonuniform does not appear to help much. It is expected that the response would then reflect an average thickness, perhaps producing a more S–shaped gain and a broadened phase. However, judging from Fig. 3.4, it is not expected that a variation of even ± 1 μm would produce sufficient alteration in the response. Moreover, measurements of the witness strips, having the same surface area as the electrodes, indicated variations in thickness of only $\pm$ 0.2 μm at most.

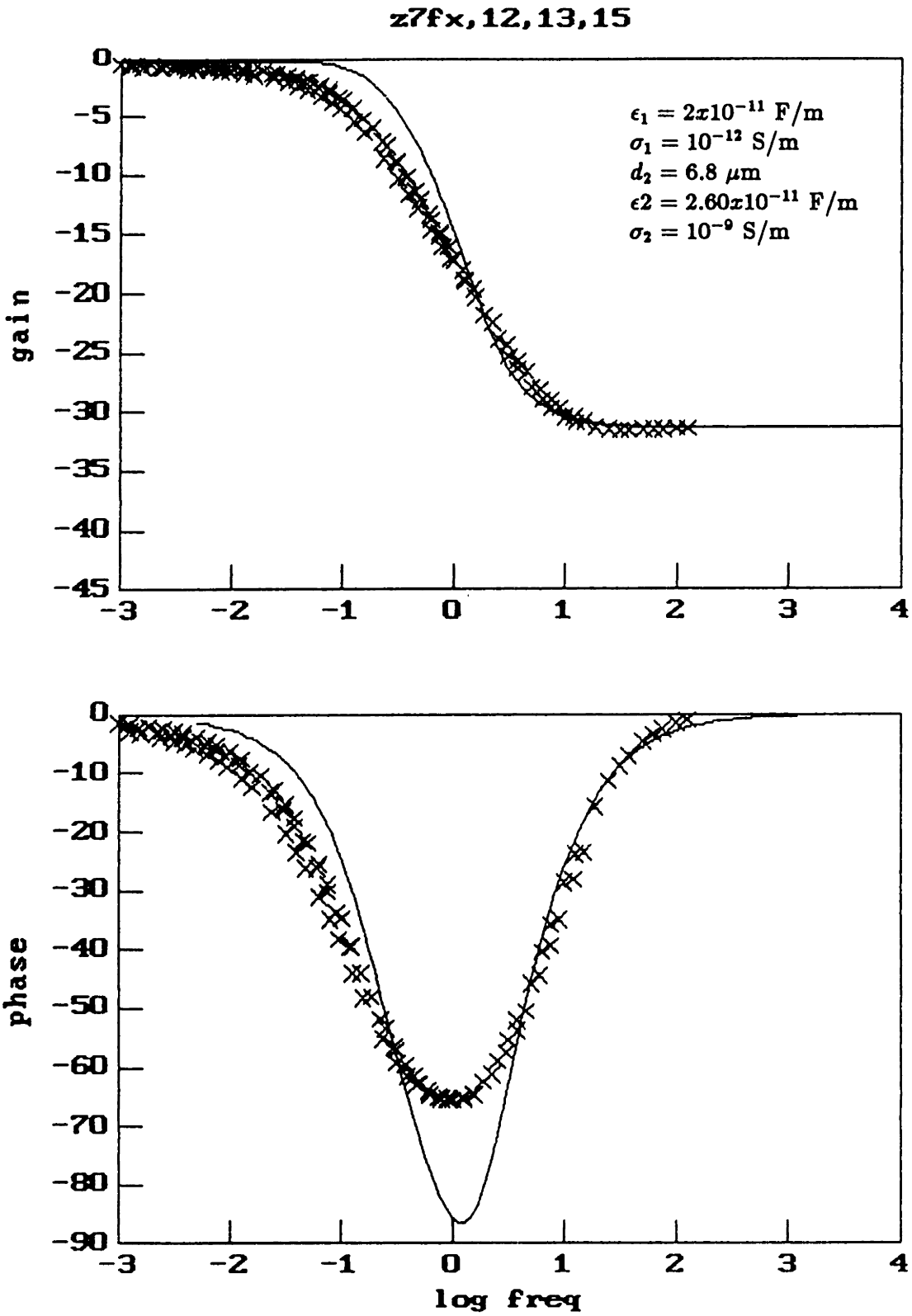


Figure 8.4: Measured Master Curve for Moisture Sensor Z7 Compared to Predicted Response using Bulk Ohmic Conduction Model.

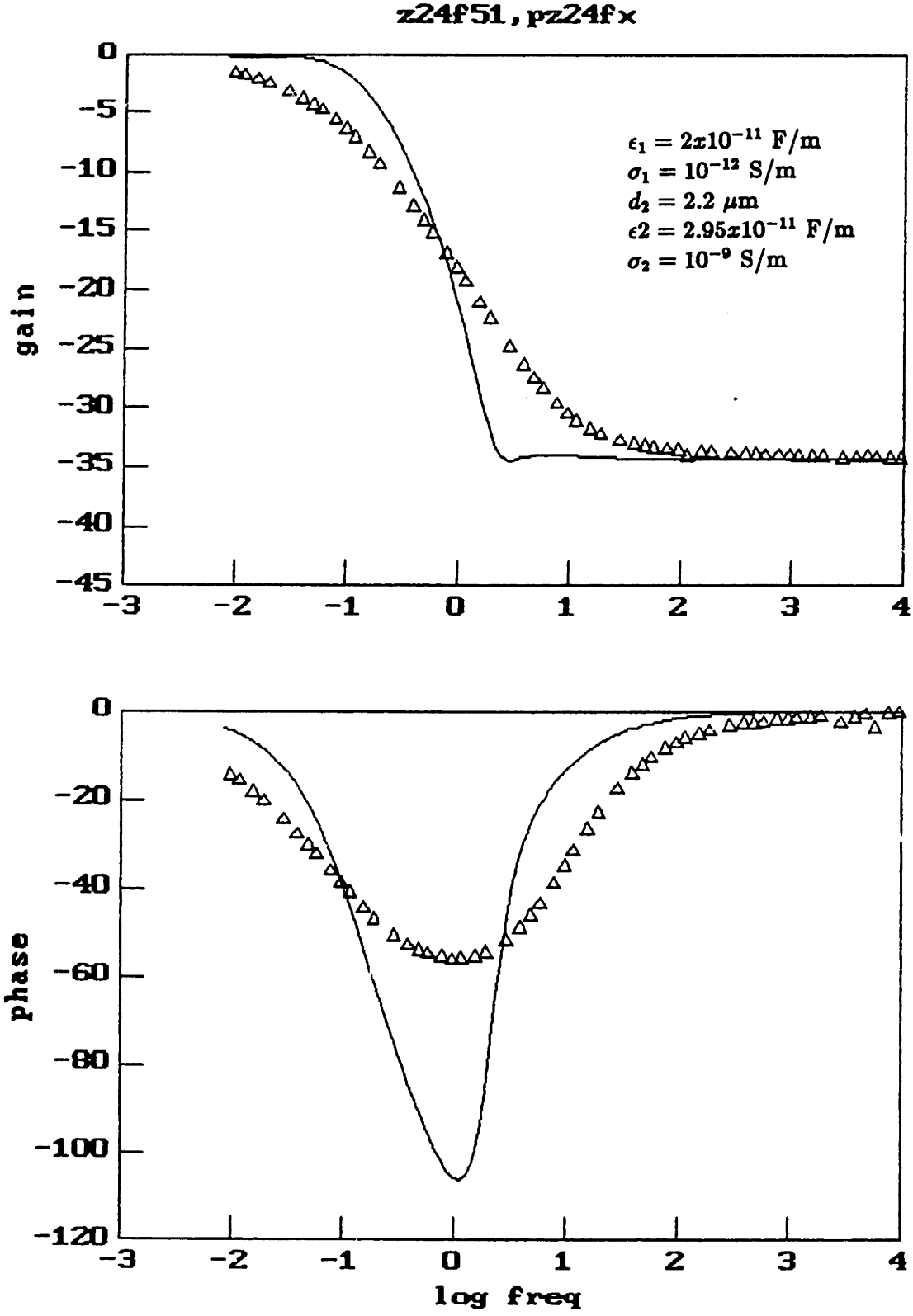


Figure 8.5: Measured Master Curve for Moisture Sensor Z24 Compared to Predicted Response using Bulk Ohmic Conduction Model.

Surface and Bulk Ohmic Conduction

The next thought was that perhaps the response reflected the influence of a surface conduction along the oil–coating interface, possibly as a result of a charge separation process at this interface. This view would still necessitate a bulk conduction occurring in the coating or else the low frequency gain would be less than unity. The response produced by such a model does not match the experimental data for Z7 (Fig. 8.6). This thought model fails even more so when trying to predict the response for a thin coating such as Z24 (Fig. 8.7). Additionally, in accordance with Theorem 3.1, both the surface and bulk conductivities would have to scale in the same manner with moisture content in order to reproduce a master curve. Given the different conduction mechanisms that could occur within the bulk versus along the surface, the likelihood of such a scenario appears to be very remote.

Spatial Distribution in Bulk Ohmic Conductivity

A more complex hypothesis introduced the possibility of heterogeneity in the coating, perhaps as a result of the deposition process, in the form of a spatial distribution of dielectric properties. The conduction process was still envisaged as being ohmic but there would be a spatial distribution in ohmic conductivities. Various distributions were used, both exponential and linear in nature. Success was obtained in modeling the response of Z7 using a linear distribution in ohmic conductivity, with an increase of two orders of magnitude proceeding from the electrode interface to the oil, with a constant permittivity (Fig. 8.8). This response was obtained by discretizing the coating (and spatial distribution) into 90 individual, homogeneous layers. In view of Theorem 3.1, in order to obtain a master curve with variations in moisture content, the spatial distribution of the conductivity would have to remain constant with moisture. While this scenario is plausible, there is nothing in the deposition process to indicate why the linear distribution in properties would arise. Furthermore, no distribution in dielectric properties, using ohmic conductivities, could account for the response in the thinner coatings. Any ohmic conductivity in the thin coatings gave rise to very large and narrow phase peaks as typified by Fig. 8.9. Thus, a spatial distribution in dielectric

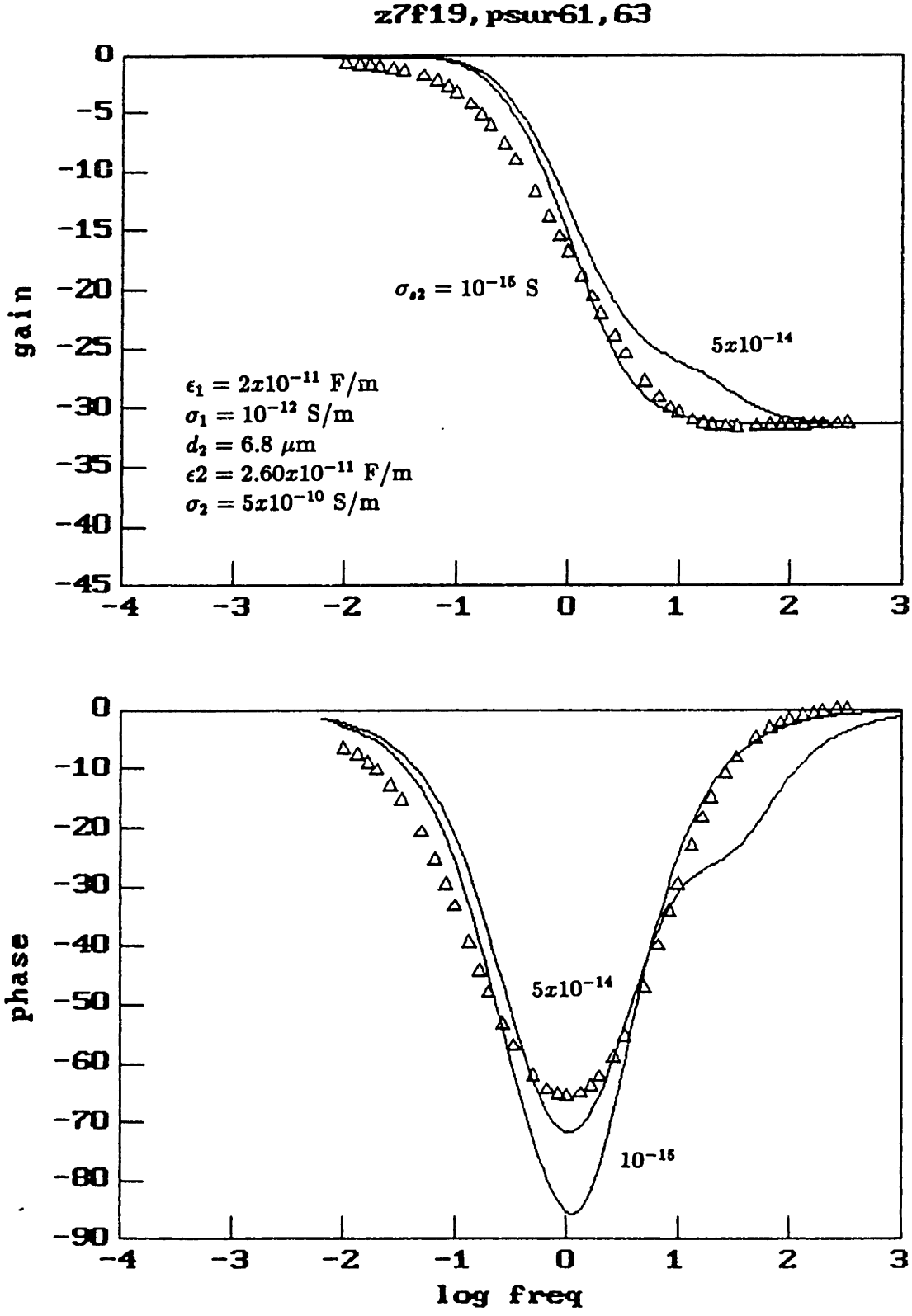


Figure 8.6: Measured Master Curve for Moisture Sensor Z7 Compared to Predicted Response using Bulk and Surface Ohmic Conduction Model.

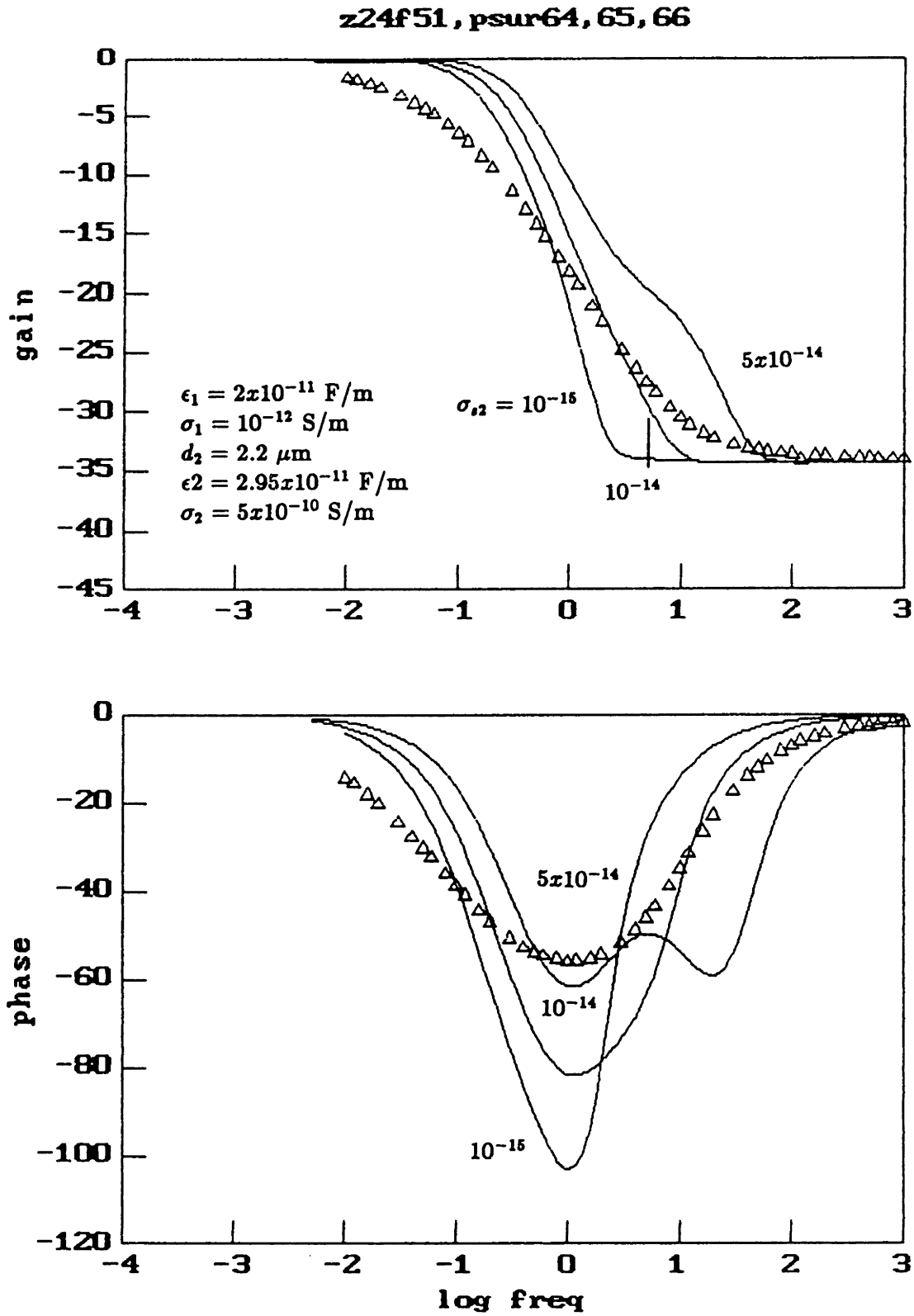


Figure 8.7: Measured Master Curve for Moisture Sensor Z24 Compared to Predicted Response using Bulk and Surface Ohmic Conduction Model.

properties displaying ohmic conduction was ruled out.

Maxwell–Wagner Polarization

Another possibility, motivated by SEM photographs discussed earlier (Figs. 8.1–8.3), was the existence of a dispersion in the dielectric properties of the coatings due to the heterogeneity of a biphasic material. This usually goes by the name of Maxwell–Wagner interfacial polarization, the charging and discharging of interfaces within a heterogeneous material. Using a Lorentz sphere model for the dispersion of small spheres embedded in the coating matrix, the effective coating permittivity would be

$$\epsilon^*_{eff} = \epsilon^*_1 \left[\frac{(2\epsilon^*_1 + \epsilon^*_2) - 2\phi(\epsilon^*_1 - \epsilon^*_2)}{(2\epsilon^*_1 + \epsilon^*_2) + \phi(\epsilon^*_1 - \epsilon^*_2)} \right] \tag{8.2}$$

ϵ^*_1 = bulk phase complex permittivity = $\epsilon_1 - j\sigma_1/\omega$

ϵ^*_2 = particle phase complex permittivity = $\epsilon_2 - j\sigma_2/\omega$

ϕ = particle volume fraction

An attempt was made to use this type of dispersion and fit the response for Z7 (Fig. 8.10). At a volume fraction of 0.4 (a fairly high amount) the magnitude of the main phase peak can be brought down by attributing a relaxation to the particle phase occurring at higher frequencies. However, this second relaxation was never observed and the gain and phase curves are still not well matched in shape. This type of dispersion cannot be used to explain the experimental responses.

In the work of Ramu, et al. [97] mentioned earlier, a Maxwell–Wagner dispersion relation was used to explain the observed frequency shifts in the peak loss tangent with RH. For this case, the assumptions were $\phi \ll 1$, $\epsilon_1 \ll \epsilon_2$ and $\sigma_1 \ll \sigma_2$. Although the exact formula given to predict the frequency of maximum loss tangent

$$f_m = \frac{\phi\sigma_2}{2\pi(\epsilon_1 + \phi\epsilon_2)}. \tag{8.3}$$

does not appear to come from the above dispersive relation, the form of the dispersion still requires two loss peaks, one associated with the particle phase and one at a lower frequency associated with the bulk medium. There should then be some intermediate

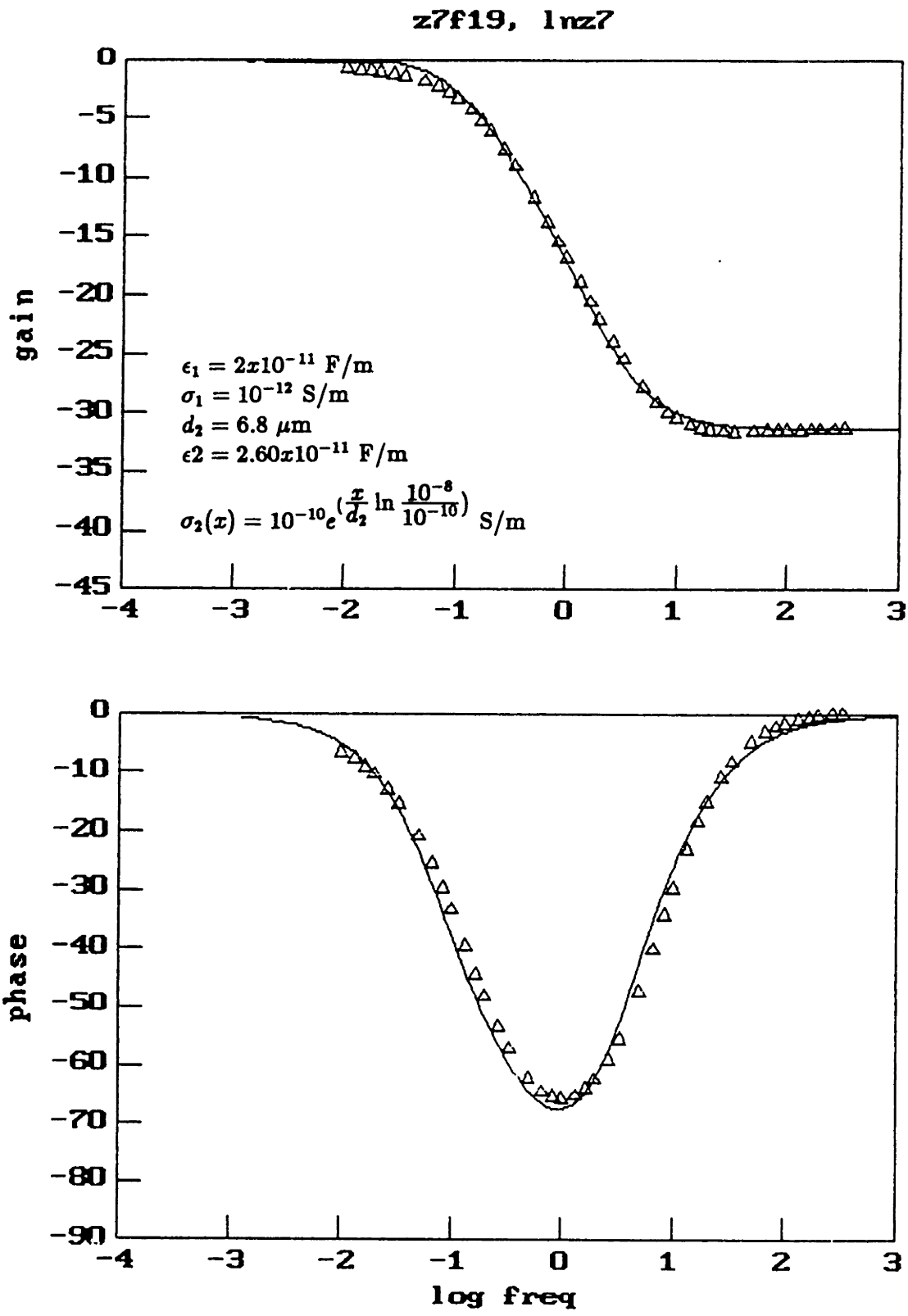


Figure 8.8: Measured Master Curve for Moisture Sensor Z7 Compared to Predicted Response using Linear Distribution in Bulk Ohmic Conduction.

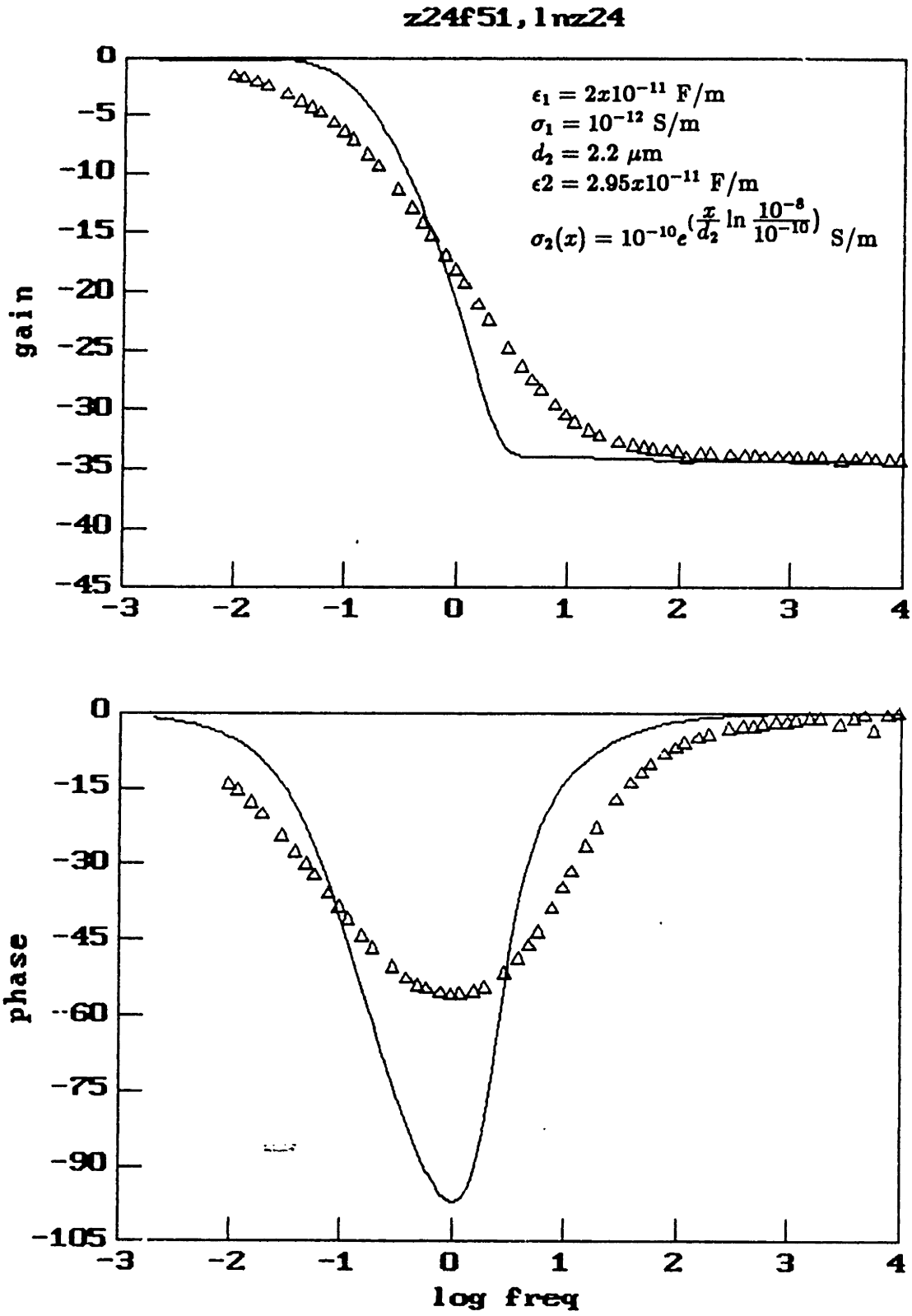


Figure 8.9: Measured Master Curve for Moisture Sensor Z24 Compared to Predicted Response using Linear Distribution in Bulk Ohmic Conduction.

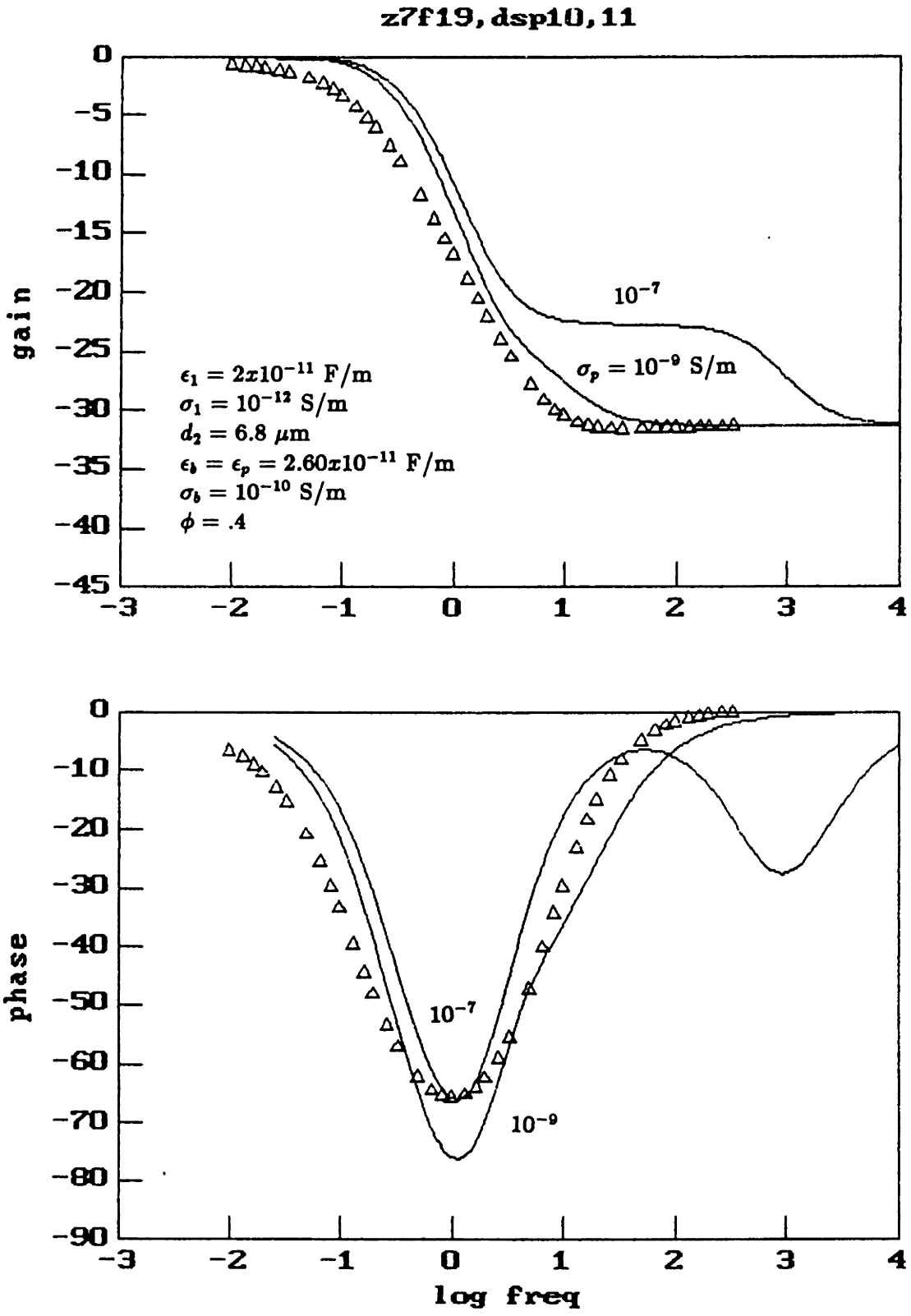


Figure 8.10: Measured Master Curve for Moisture Sensor Z7 Compared to Predicted Response using Dispersive Complex Bulk Permittivity Arising From Lorentz Sphere Model.

region during which the capacitance of the bulk material is evident. No such evidence is given. In fact, no mention is made of the existence, or nonexistence, of dispersion in the real part of the permittivity. Without presenting the dispersion of the complex permittivity it is not possible to ascribe any particular dispersive molecular model to the conduction process.

Dispersive Transport

All of the above charge carrier scenarios utilize nondispersive, ohmic conductors. In general, charge transport and relaxation in amorphous semiconductors and solid dielectrics are dispersive [100]. Very often, this dispersive transport is observed in the time domain. A step voltage is applied across a sample and the current decay is monitored.

One type of decay observed can be represented mathematically as

$$i(t) \sim e^{(-t/\tau)^{\beta}} \tag{8.4}$$

and is known as the Williams–Watts function. This function is very similar to (8.1) and thus, a physical interpretation of the existence of a distribution of dipole relaxation times is given. Another physical model, proposed by Scher and Montroll [101], uses the formalism of continuous time random walk to describe carrier motion with a distribution of event times. The event time could be due to either hopping between the energy levels of localized sites or release from a trap. The resulting behaviour of the dispersion is (8.4).

Other observed time decays are of the form

$$i(t) \sim t^{-s(t)} \qquad 0 < s(t) < 2 \tag{8.5}$$

and are usually accounted for mathematically using distributions in charge relaxation times or charge carrier hopping probabilities. There are several theories that attempt to provide a physical basis for these distributions.

One such theory is a defect–diffusion model proposed by Shlesinger [102] uses the concept of fractal time stochastic processes to describe the relaxation of a mobile defect

in a volume containing frozen-in dipoles (traps). When there are many more defects than traps, the form of the decay is (8.4). When the reverse situation applies, more traps than defects,

$$i(t) \sim t^{-\alpha} \qquad 0 < \alpha < 2 \tag{8.6}$$

In a related model, according to Jonscher [103], low frequency dispersion (LFD) is associated with very large polarization arising from the migration of charges through large but limited paths. In particular, LFD is associated with hopping, that is, low mobility charge carriers, whether ionic or electronic. The physical model involves a many–body interaction and results in a time domain response of the form of (8.6). The corresponding frequency domain response is

$$\epsilon^*(\omega) = B(j\omega/\omega_p)^{\alpha-1} \tag{8.7}$$

It will be shown that the observed behaviour of the moisture sensors follows this dispersive relationship and will produce the invariance in frequency response with respect to moisture content and temperature. This is not meant to imply any preference for one physical model over another. The emphasis here is on the correlation of the observed dispersive behaviour.

8.2.2 Low Frequency Dispersion

The power law dispersion, as described by Jonscher [104, p. 87], has been observed experimentally for a wide range of materials over many decades in frequency. Some of Jonscher's work can be found in [103] and [105] but, for a complete compilation of experimental results, the reader is referred to [104]. In it, examples are given of dispersion relations that scale with temperature and relative humidity in the manner of Theorem 3.1, producing what Jonscher calls universal responses but what are referred to here as master curves. Two examples are reproduced in Fig. 8.11. The first, Fig. 8.11a, shows the normalization of loss peaks for the α relaxation in polydian carbonate as a function of temperature. The second example, Fig. 8.11b, shows the

normalization of the complex permittivity of a sample of sand as a function of relative humidity.

Furthermore, experimental evidence of low frequency dispersion has been observed in plasma polymerized films by Hetzler and Kay [106]. Measurements of the dispersion in plasma deposited films of tetrafluoroethylene (TFE) mixed with argon as a transport gas displayed a power law dependence at 25°C with $n \sim 1$. Film thicknesses varied from submicron to 3 μm. Using an optical absorption technique, it was discovered that the observed extinction was due to Rayleigh scattering from particles up to several hundred Å in diameter. Thus, these films also exhibited a heterogeneity due to the inclusion of microspheres. However, no connection between the film morphology and the ambient temperature conduction process was made. Other than attributing the conduction mechanism to a hopping conductivity, there was no further discussion of the room temperature results as most of the effort was spent in understanding the dipole losses exhibited at elevated temperatures of 180°C. No mention was made of the RH of the atmosphere while the measurements were made.

As discussed earlier in Section 8.1, Ramu, et al. [97] observed a dispersion in plasma polymerized HMDSO films. Examination of the full behaviour of the complex permittivity might reveal a power law dependence of the dispersion.

8.2.3 Parameter Estimation of the Master Curve

Using the parameter estimation routines developed in Chapter 4, and the conclusion of bulk dispersive transport reached in the previous sections, an estimation was made of the dispersive dielectric properties of the coating from the experimentally measured responses.

An example of this estimation process was to use a response of chip Z24, 2.2 μm thick coating, to unused oil having 48 ppm moisture content (Fig. 8.12) and estimate the permittivity and apparent conductivity at 3 points per decade as shown in Fig. 8.13. This estimation process was performed using a complex bulk permittivity of $\varepsilon_2 = 2x10^{-11}$ F/m and $\sigma_1 = 0$) for the oil above the coating. Essentially, the oil is so much more insulating than the coating that its conductivity may be neglected for this

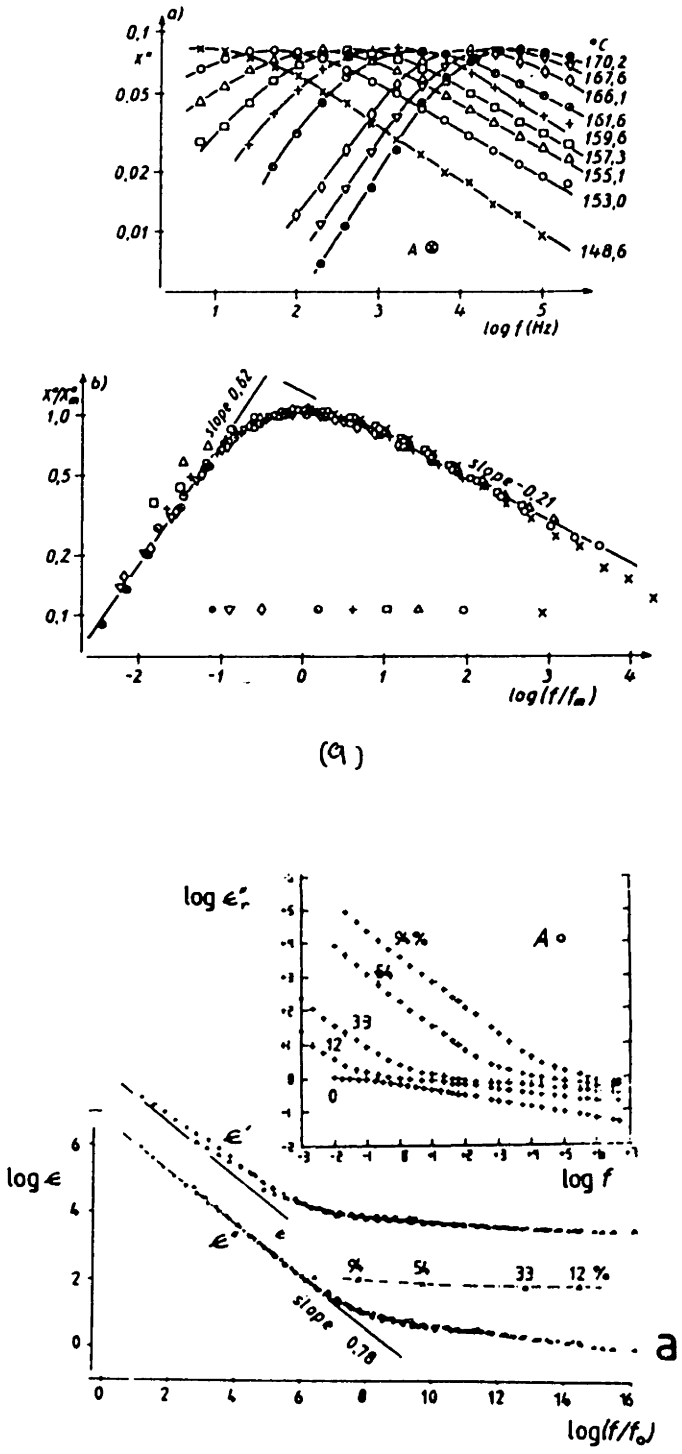


Figure 8.11: Experimental Data Showing Normalized, Power Law Dispersion of a) Polydian Carbonate with Temperature Dependence and b) Sand with RH dependence (from [104, p. 177 and 277]).

estimation problem. However, it is very important to include the oil permittivity as it will significantly affect the coating bulk permittivity estimate. Note that the coating permittivity estimate tends to the static dielectric constant ($\epsilon_{st} = 2.95x10^{-11}$ F/m) at the higher frequencies, and that this value corresponds to the estimated permittivity for Z24 in Table 5.1 (obtained using a different set of data). The apparent conductivity is dispersive, significantly deviating from an ohmic behaviour. In Fig. 8.14 the results are replotted to display the dispersive nature of the coating. The estimated static dielectric constant was subtracted from the permittivity ($\epsilon' - \epsilon_{st}$) (dots), and the conductivity was converted to ϵ'' by dividing by $\omega = 2\pi f$ (crosses). By drawing a best fit line through the data on a log–log plot, it appears that the dispersive relationship between the complex permittivity and frequency follows a power law dependence.

The form of the power law dispersion, adapted from Jonscher [104, p. 87] but expressed in terms amenable to Theorem 3.1, is

$$\begin{aligned} \epsilon^*(\omega) &= \epsilon' - j\epsilon'' \\ &= B(j\omega/\omega_p)^{n-1} \\ &= B[\sin(n\pi/2) - j\cos(n\pi/2)](\omega/\omega_p)^{n-1} \end{aligned} \tag{8.8}$$

with $0 < n < 1$ and $\omega_p = f(T, RH, c \ldots)$.

The constraint on n arises from the requirement, for a linear, causal system, that the Kramers–Kronig transformations be satisfied [104, Section 2.8]. Using fourier analysis and complex integration, these transformations relate the real and imaginary parts of the complex permittivity, after subtracting out any dc conductivity or static dielectric constant. For chip Z24, $n \sim 0.38$ and $B \sim 9.1x10^{-11}$ F/m. Using (8.8), the amount by which $\log(\epsilon' - \epsilon_{st})$ is offset from $\log(\epsilon'')$ is $\log(\tan(n\pi/2))$, and this correlates well with the difference in the estimated intercepts obtained using the best fit lines, as noted in Table 8.1.

The above exercise was repeated for the response of chip Z7, 6.8 μm thick coating, in unused oil having a moisture content of 14.2 ppm (Fig. 8.15). Again a power law dispersion was observed with $\epsilon_{st} = 2.53x10^{-11}$ F/m, roughly the same value as given in Table 5.1. Here, $n \sim 0.26$ and $B \sim 3.0x10^{-11}$ F/m. The lack of estimates of ϵ' at

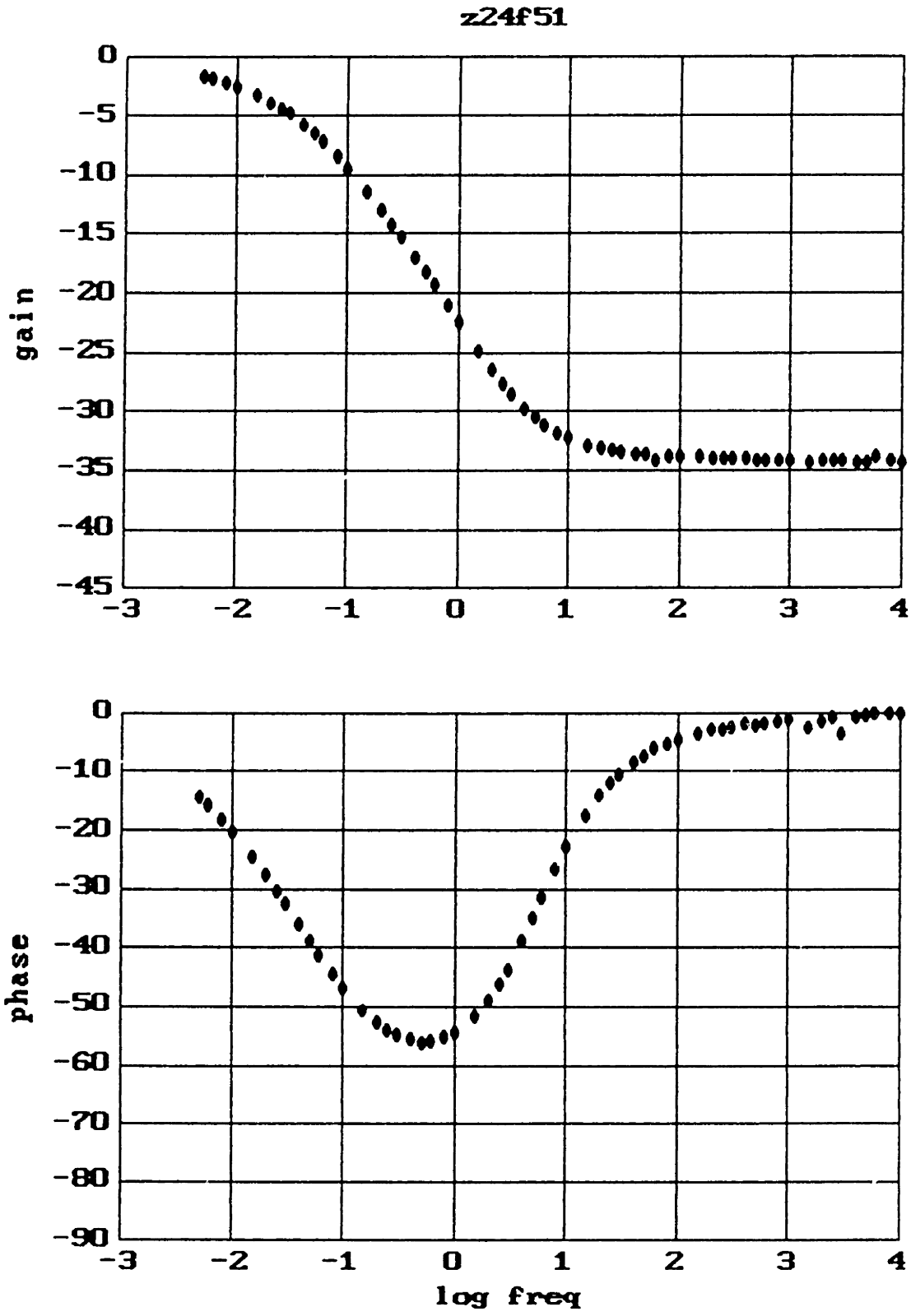


Figure 8.12: Measured Response of Aged Moisture Sensor Z24 in Stirred, Unused Oil at 48 ppm Water.

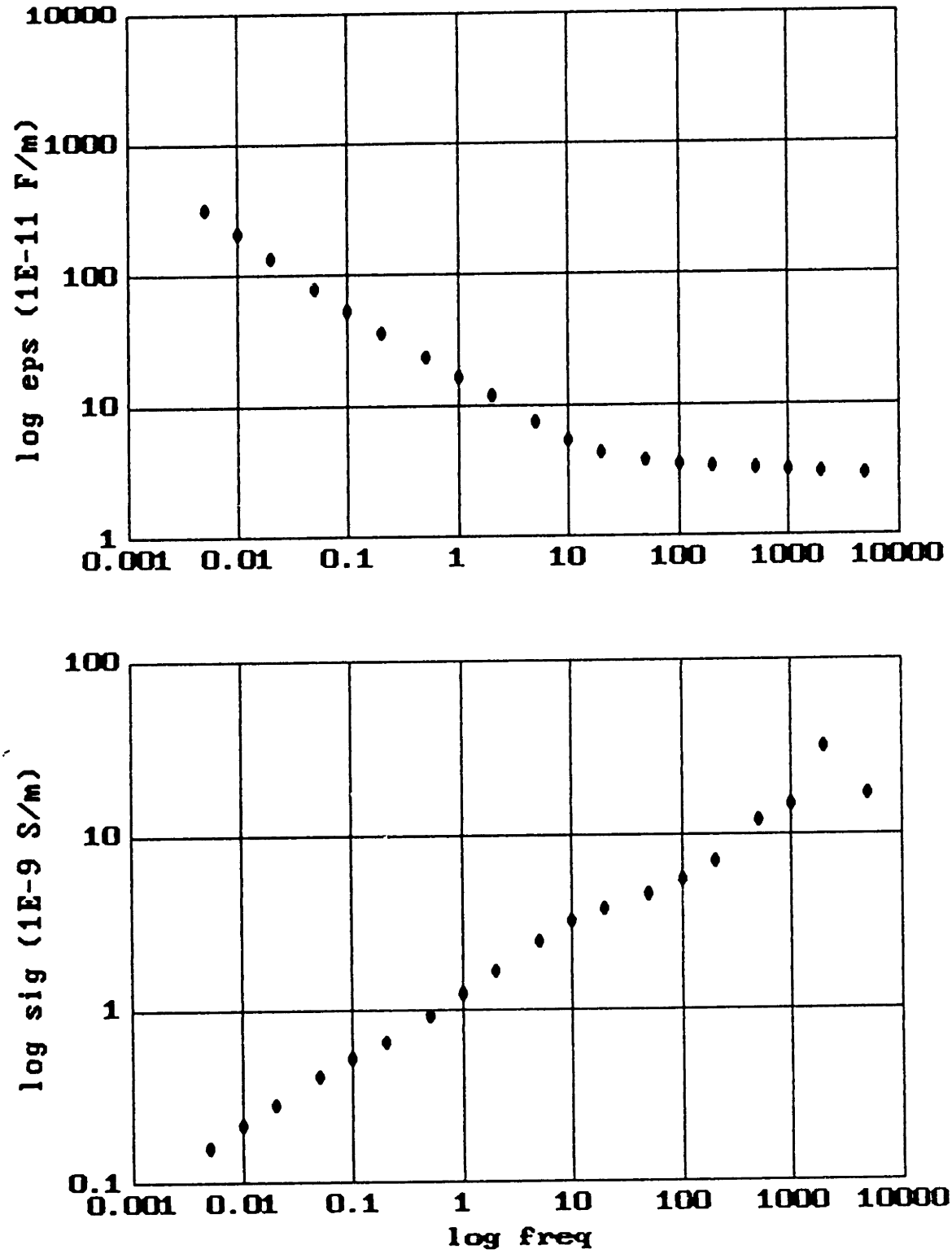


Figure 8.13: Estimated Dispersion in Bulk Permittivity and Conductivity of Z24 PP–Bromobenzene Coating on Z24 using Response of Fig. 8.12 ($d_2 = 2.23$ μm).

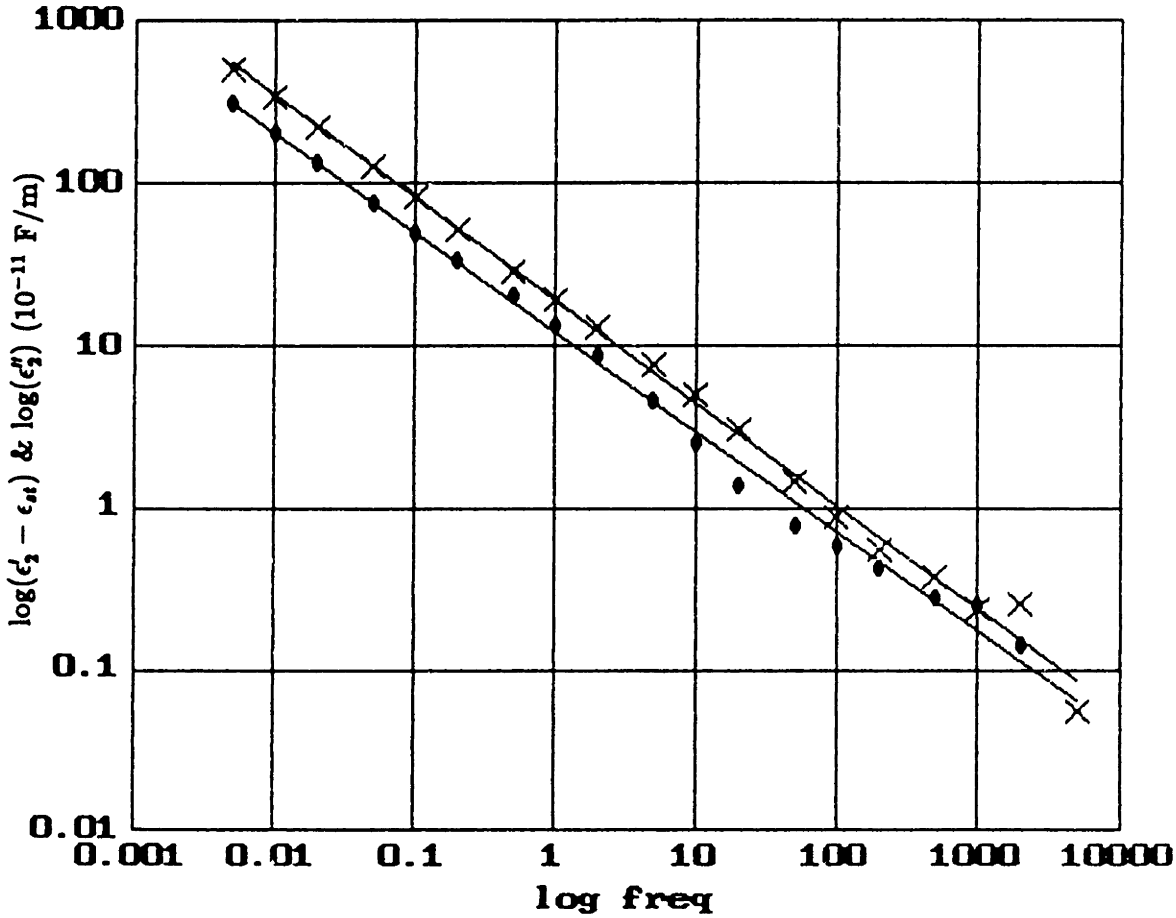


Figure 8.14: Estimated Dispersion in Complex Bulk Permittivity of Z24 PP–Bromobenzene Coating from Fig. 8.13. (•) – $\log(\epsilon_2' - \epsilon_{st})$, (X) – $\log(\epsilon_2'')$.

Table 8.1: Dispersion Parameters For Moisture Sensors

$$\epsilon^*(\omega) = B(j\omega/\omega_p)^{n-1} + \epsilon_{st}$$

Chip ID	Chip Cond.	n	B (10^{-11} F/m)	ϵ_{st} (10^{-11} F/m)	$\log\tan(\frac{n\pi}{2})$	$\Delta(int)$
Z24	Aged	0.38	9.1	2.95	−.167	−.201
Z7	Fresh	0.16	2.3	2.45	−.591	−.417
"	Aged I, Adjusted	0.26	2.7	2.65	−.364	−.265
"	Aged II	0.26	3.0	2.53	−.364	−.335
"	Aged III	0.28	3.4	2.60	−.327	−.309
Z20	Fresh	0.24	3.1	2.60	−.402	−.422
"	Aged	0.36	4.4	1.94	−.455	−.249
"	Adjusted, Aged	0.32	6.4	2.64	−.260	−.383

the high frequencies is due to the occurrence of a slight positive phase and dip in gain, a parasitic effect associated with the different capacitance seen by the floating gate FET versus the reference FET [70, Section 5.5].

The dispersive relations characterizing Z7 at several different stages of aging are given in Table 8.1. Aging of the films altered the master curves somewhat, as shown in Figs. 7.12–14. This effect can be seen in changes in the power law dependence as the coating of Z7 ages. The broadening effect with aging shows up in the increase of n and B, with some apparent change in ϵ_{st} also. Of course, the drastic change in moisture sensitivity with aging appears in the parameter f_p.

Presented next is the dispersive relation describing chip Z20, 6.0 μm thick coating, in unused oil having a moisture content of 17 ppm (Fig. 8.16). The resulting dispersion relation appears to exhibit a similar power law dependence with $\epsilon_{st} = 1.94x10^{-11}$ F/m except for the low frequencies. For many of the dispersions, deviations from the power law dependence occurred at the low frequencies, as seen in this case. The permittivity appears to level off and the loss appears to peak and begin descending. If this result were taken at face value, it would indicate a capacitance, perhaps due to blocking at an interface, dominating at these lower frequencies. However, as previously mentioned, it is known that the gain rarely ever goes to exactly 0 dB (unity gain) at low frequencies due to experimental errors. Here, the culprit is a large DC offset, on the order of –1 to –2 volts. This problem results in a roughly constant shift in the gain curve. A correction to the problem can be made by shifting the gain curve such that it does approach 0 dB, subtracting the difference between the low frequency gain (when it plateaus) and 0 dB, and re–estimating. Again, this correction assumes there is no significant electrode polarization and that the gain error is constant with frequency. This results in significant alterations in the dispersion only at the low frequencies. The results of such a computation (a gain correction of 1.9 dB) for the response of Z20 used above is shown in Fig. 8.17. The change in n and B give some idea of the sensitivity of these parameters. An additional benefit of this correction is that the re–estimated ϵ_{st} of $2.64x10^{-11}$ F/m is much closer to the value given in Table 5.1. It is seen that the behaviour of the estimated dispersion at the low frequencies is dominated by whether

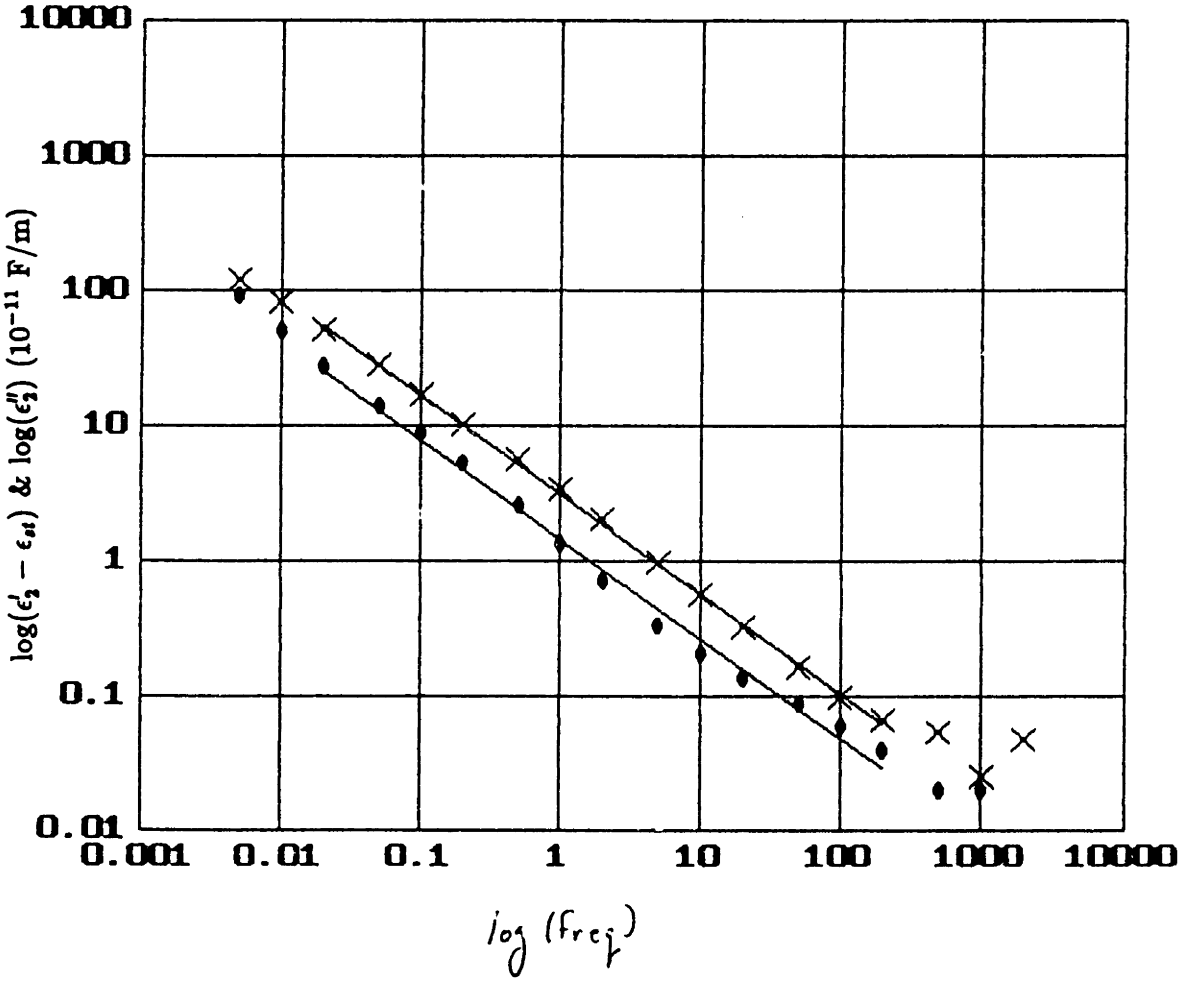


Figure 8.15: Estimated Dispersion in Complex Bulk Permittivity of Aged Z7 PP–Bromobenzene Coating. (•) – $\log(\epsilon_2' - \epsilon_{st})$, (X) – $\log(\epsilon_2'')$.

the nonzero dB gain is real or artifactual.

8.3 Film Aging and Electrical Conduction Mechanism

As the coatings aged, the parameters tabulated in Figs. 7.10 and 7.11, expressing the moisture sensitivity for Z7 and Z20, changed. In fact, both the slopes and the intercepts increased monotonically with time, as shown in Fig. 8.18. The different paths taken reflect the different temperature cyclings for each coating. Eliminating time as the independent variable and plotting slope–intercept pairs, as in Fig. 8.19, yields a very interesting result. The slope and intercept appear to be related to each other throughout the aging process. In fact, there appears to be a distinct trajectory along which the coatings age although the sample size is rather small (only two coatings).

The issue of aging is intimately linked with the conduction mechanism responsible for the electrical response of the film. What is carrying the conduction current and why is it changing with time? The following discussion relates work reported in the literature on this subject to the experimental observations of the previous chapter. Methods used to improve the stability of the coatings are also reported.

The physical picture given by Jonscher [104, Chapter 8] or Shlesinger [102], and associated with the dispersion relation of Section 8.2.3, is one of charge carriers (or mobile defects) with low mobilities traveling through paths limited by frozen dipoles or traps. These charge carriers could be ionic or electronic. This might be a good description of the interaction of impurities with free radicals trapped in the film. As discussed in Appendix C, free radicals play an important part in the formation of films via plasma deposition and can be present within the film in significant concentrations. The effect of moisture might then be to enhance the mobility of the impurities. Tyczkowski and Kryszewski have shown that moisture does not directly contribute to the conductivity by comparing results using regular and heavy water [92]. The ionized species from these two molecules have sufficiently different mobilities such that

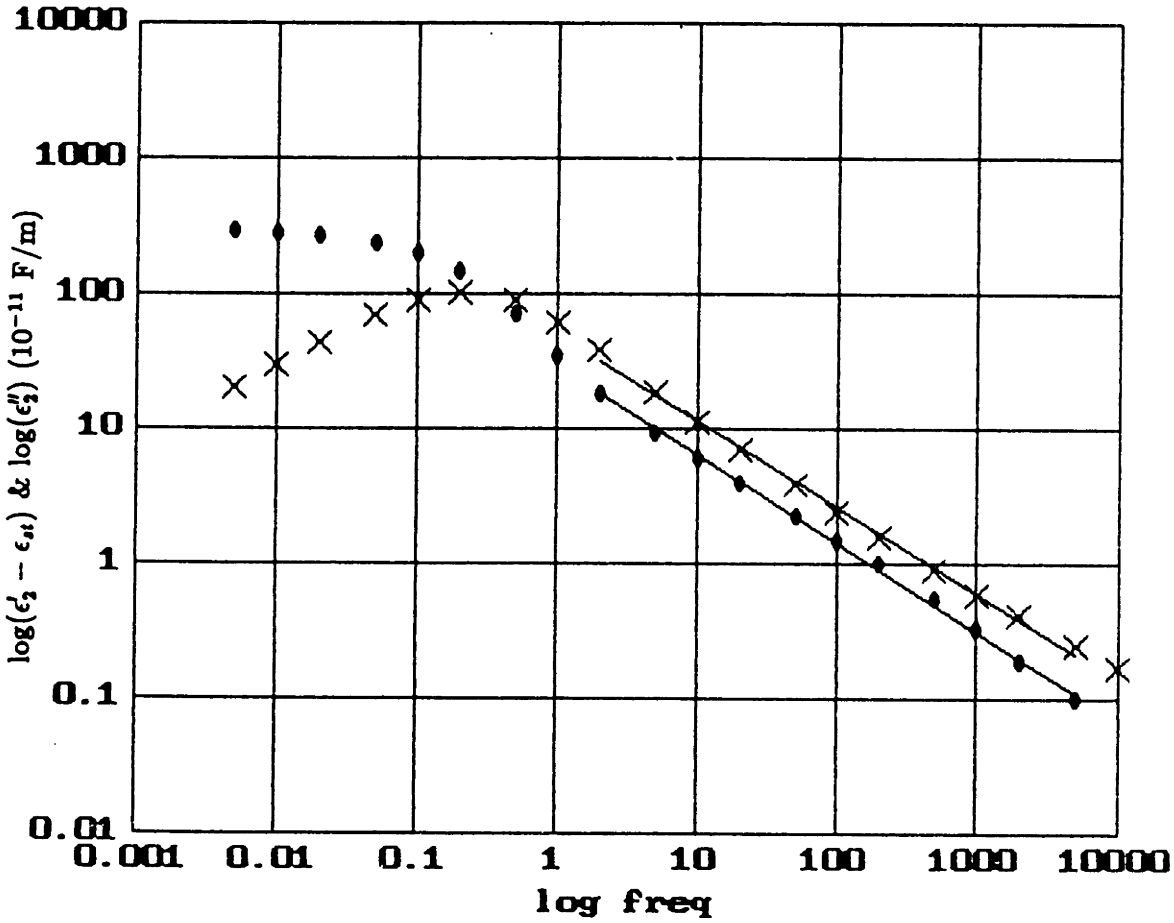


Figure 8.16: Estimated Dispersion in Complex Bulk Permittivity of Aged Z20 PP–Bromobenzene Coating. (•) – $\log(\epsilon'_2 - \epsilon_{st})$, (X) – $\log(\epsilon''_2)$.

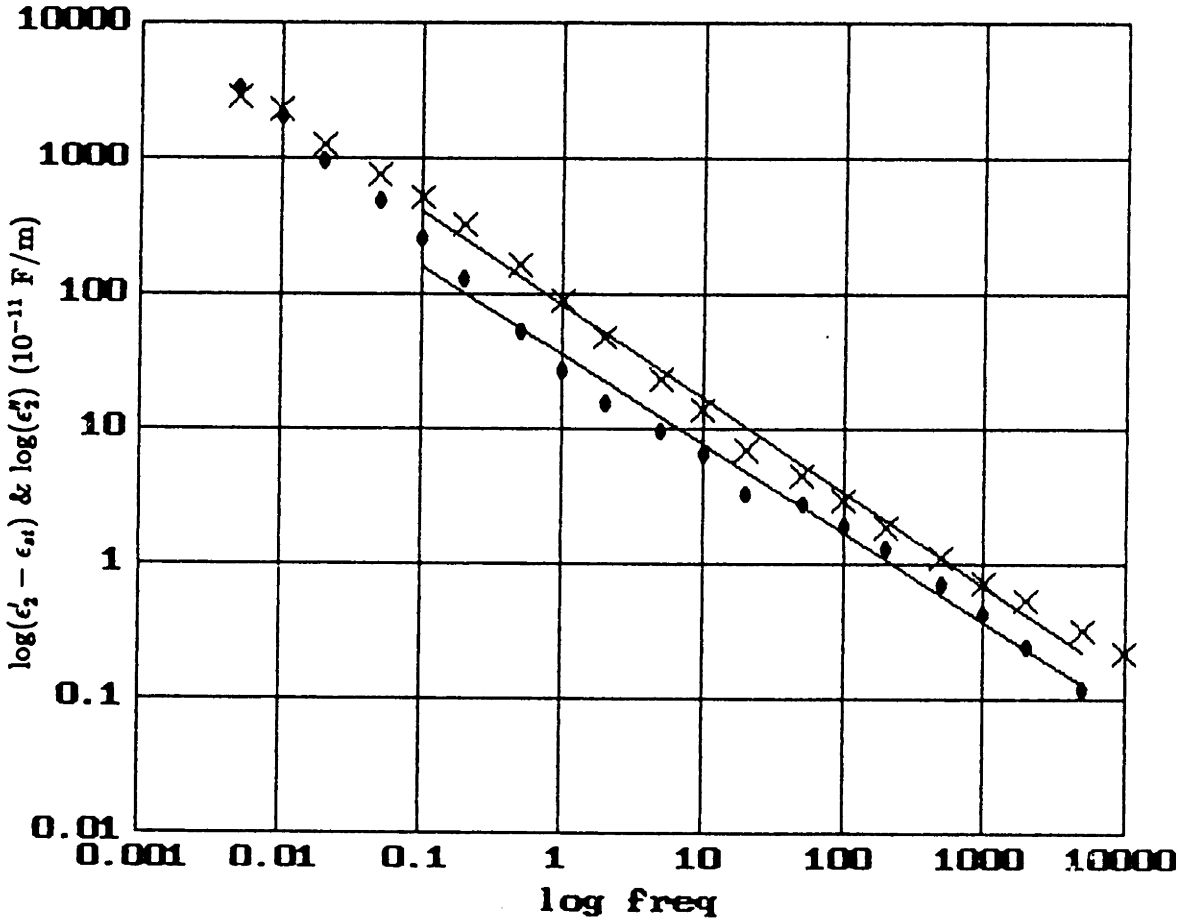


Figure 8.17: Estimated Dispersion in Complex Bulk Permittivity of Aged Z20 PP–Bromobenzene Coating using Corrected Response. (•) – $\log(\epsilon_2' - \epsilon_{st})$, (X) – $\log(\epsilon_2'')$.

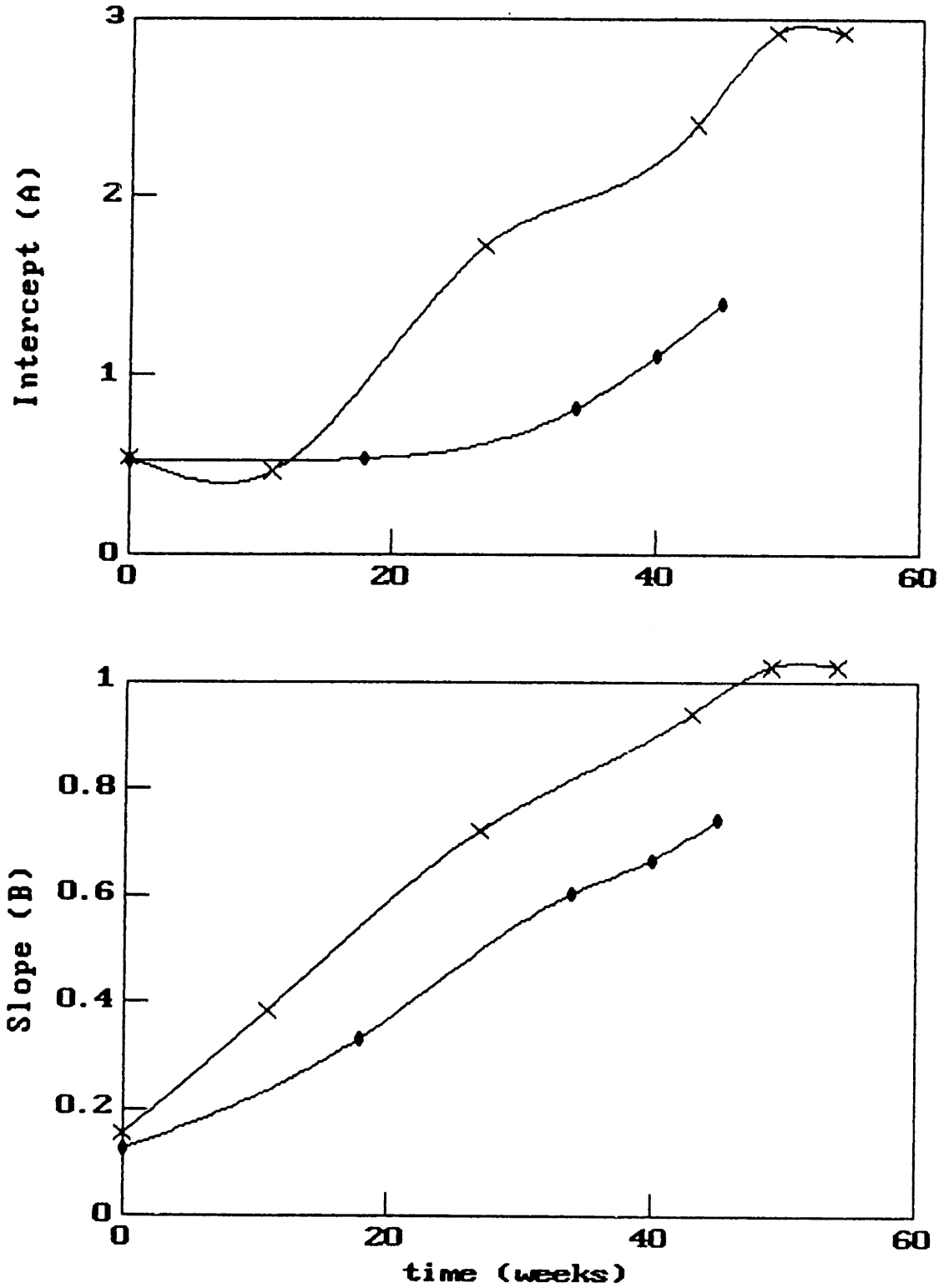


Figure 8.18: Time Dependence of Change in Parameters Describing Moisture Sensitivity for Z7 (X) and Z20 (•) with Aging.

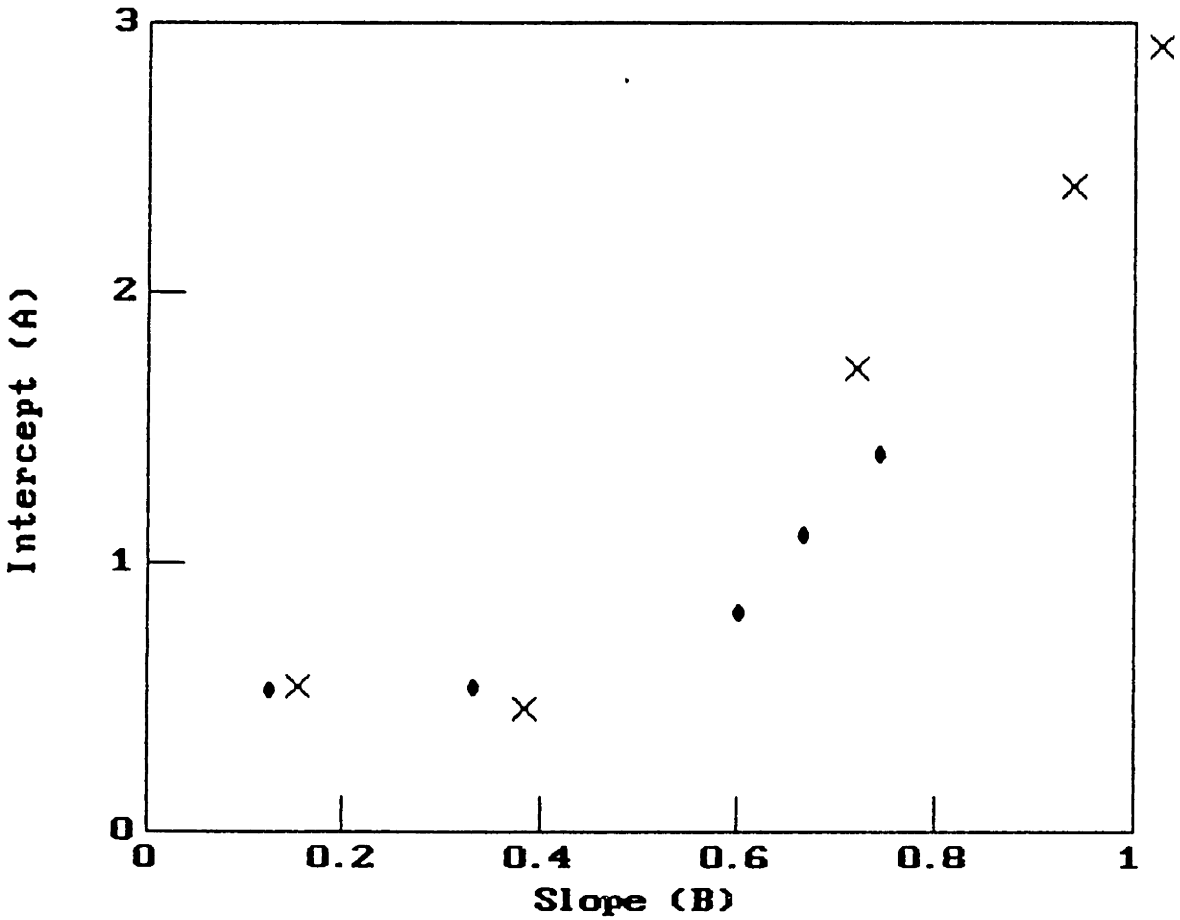


Figure 8.19: Correlation of Parameters Describing Moisture Sensitivity for Z7 (X) and Z20 (•) with Aging.

the effect on the conductivity due to dissociation of the absorbed moisture would be apparent. There was no measurable difference in conductivity so the conclusion was that the resulting moisture sensitivity was electronic in nature, due to an activation of initially inactive intrinsic acceptor centers. This conclusion does not take into account the possibility of absorbed moisture ionizing other impurities within the film or interaction with the trapped free radicals. Many examples of enhanced conductivity due to ionization arising from absorbed moisture in standard polymeric solids were compiled by Barker [107]. At low moisture concentrations (ppm), before any significant effect on the macroscopic dielectric constant occurred, partial dissociation of moisture and enhanced dissociation of impurities resulted in increased DC conductivities.

Aging might then be associated with a decreasing concentration of charge carriers over time. Lack of coating stability has been related to the interaction of oxygen with the trapped free radicals [83, Section 11.2.3]. One effect is the formation of polar groups arising from this interaction increasing the dipolar losses of the film. For example, Hetzler and Kay [106], in examining plasma polymerized TFE at 180°C, concluded that dipole losses dominated, a confusing result given the small amount of dipole concentration due to CF_x bonds. Using FTIR, it was discovered that C–O bonds were responsible for the dipole losses and that these bonds formed upon exposure of the film to oxygen in the air. In another type of interaction, DC conduction measurements by Morita, et al. [108], on fresh plasma deposited styrene films, displayed an irreversible decrease in conduction currents on exposure to oxygen but a reversible increase with exposure to humid air. Experiments by Tyczkowski and Kryszewski [92] on plasma polymerized organosilicon films showed an increase in DC conduction currents with exposure to oxygen at an elevated temperature of 380°C.

While the particular direction of the effect depends on the specific monomer used, be it hydrocarbon or organosilicon or fluorocarbon, it is clear from the above references that oxygen is having a significant effect on the electrical properties of the film. According to Yasuda [83, Section 11.2.3] the instability of the film over a long period of time is attributed to the low mobility of large molecules within the highly cross-linked structure. The surface of the film may be in equilibrium with the ambient oxygen

levels but it will take a significant amount of time for oxygen to penetrate the film and interact with free radicals trapped deep down in the bulk of the film. There are several posttreatment methods that reportedly improve the stability of the films. One is to quench the free radicals by deliberate exposure to oxygen [83, Section 11.2.3]. However, this may result in undesirably high dielectric losses. The second method is to thermally anneal the films in vacuum. The effectiveness of this method is mitigated by the low mobility of the polymer segments. Without an increase in this mobility at elevated temperatures there is the possibility of thermal creation of new free radicals due to the breaking of chemical bonds. Plasma polymerized organosilicon films, treated for 2 hours at 350°C in a nitrogen atmosphere, exhibited reduced DC conduction currents and were insensitive to changes in humidity [92]. Tuzov, et al. [109] found that the capacitance of plasma polymerized HMDSO films, heat treated in vacuum at 350°C for 1 hour, stabilized to ~120% of its initial value after 4 days. The largest change in capacitance and least stable film was produced by heating the film in air at 350°C for 1 hour. The exact conditions required for maximizing stability are a function of the monomer and polymerization conditions as well as the time and temperature of the posttreatment. The third method for improving stability is to increase the substrate temperature during deposition. Ramu, et al. [97] found that the loss tangent of plasma polymerized HMDSO films stabilized within 2 to 3 days when produced at a substrate temperature of 400°C.

However, there is a price to pay for any method of stabilizing the film properties – the resultant film may not have the required sensitivity or properties for the specific task. This qualification must be taken into account when investigating methods to reduce aging effects.

Chapter 9

Hydrogen Gas Sensors – Preliminary Research

9.1 Motivation

Among the various approaches to gas analysis outlined in Section 1.3.2, the one chosen for development was Fourier Transform Infrared (FTIR) spectroscopy. The development and testing of this approach is beyond the scope of this thesis. However, this option is not able to detect H_2 , thus, there is still a need to develop an H_2 gas sensor. Most of the H_2 gas detectors discussed in Section 1.3.2, with the exceptions of the flame ionization detector (FID), thermal conductivity detector (TCD) and electrochemical fuel cell, are still in the development stages. The unstructured doped metal oxides devices, such as the Taguchi Gas Sensor (TGS), are not selective enough to H_2 . However, the hydrogen contribution could possibly be determined by subtracting out the other gaseous components as measured by the FTIR. The structured metal oxide devices are not available for commercial use, presumably due to poor performance during long term operation. In any case, it was felt that the area of hydrogen gas sensing was still at such a developmental stage that it was worthwhile to explore promising techniques, perhaps exploiting the capabilities of the microdielectrometer.

As with any gas sensor, the five major areas of attention are: selectivity, sensitivity, reversibility, stability and operating conditions. The last area is of great design

importance – whether to operate the sensor immersed in the oil, or in a gas phase that is in equilibrium with the dissolved gases in the oil. In either case the gas sensor will be coated with a thin polymer film (membrane) such as parylene. This coating will improve the selectivity to hydrogen by impeding the diffusion of larger gas molecules and moisture, and/or protecting the device from the oil. The advantage to operating in the oil is the anticipated rapid equilibrium between the local gas concentration in the oil and the sensor via the thin ($\sim$ 5 μm) polymer film. However, a disadvantage is that the film must remain stable in its mechanical and physicochemical characteristics for a long period of time (years) in the hostile environment of hot transformer oil. Another important constraint is that the sensor must remain at the temperature of the oil ($<$ 90°C). Typically, the solid state gas sensors are operated at higher temperatures ($\geq$ 150°C). This lower temperature constraint, although aiding in the hydrogen selectivity through the favoring of the adsorption–desorption kinetics for hydrogen, generally results in a reduced sensitivity and slower transient response. Operating the sensor in a gas phase in equilibrium with the oil does not suffer from this constraint, nor the previous one. The major disadvantage to operating a solid state sensor in the gas phase is the requisite increase in complexity of having to establish yet another equilibrium, between the oil and gas phase. If the sensor is operated behind a membrane separating the gas and oil phase, the membrane must be designed as thin as possible so as to minimize the diffusion time through the membrane of the gases of interest.

For the preliminary work presented here, the devices will be evaluated in the gas phase. At present, parylene is being envisioned as serving as the membrane material, given its good performance as a moisture barrier and its compatibility with transformer oil. Two approaches that were explored are described in the following sections. The first approach was to sensitize a microchip to hydrogen gas by coating it with a thin film of tin oxide. This approach relies on the known gas sensitivity of tin oxide (see Section 1.3.2) but tries to adapt the concept to a high impedance measurement of the AC surface conductivity of thin films. No significant gas sensitivity was observed. However, these experiments were conducted at room temperature with no attempt

at controlling temperature variations. As a consequence, the RH and temperature variations had a large, and perhaps dominant, effect on the surface conductivity. The second approach involved the use of thin film palladium resistors. Here, the temperature of the experiments was carefully controlled and varied between 100° and 300°F. Bulk absorption of hydrogen by the resistors produced measurable changes in the resistance.

9.2 Tin Oxide Coated Microchip

9.2.1 Background

This approach combined the gas sensitivity of doped metal oxide thin films (see Section 1.3.2) with the detection capabilities of the microdielectrometer. Thin films ($\leq$ 2000 Å) of tin oxide (SnO_2) were vacuum deposited onto an uncoated microchip. Here, instead of measuring a change in DC conductivity of a macroscopic pellet of sintered tin oxide powder, an AC, high impedance (low current) measurement of the conductivity of a thin film was made. The potential advantages of this approach are: 1) AC operation avoids any DC polarization effects that may adversely alter long term performance, 2) high impedance implies low currents, yielding negligible temperature variations arising from the measurement and negligible formation of byproducts due to electrochemical reactions, and 3) given a microdielectrometer type of device in the monitoring system, adding another microchip sensor is cheap.

Uncoated microchips were placed inside a bell jar vacuum chamber and a gram of stannic oxide powder (Mallinckrodt) was placed in a tungsten boat. After the chamber was evacuated, high currents were passed through the boat – heating and vaporizing the powder. This vaporized powder diffused throughout the whole chamber, coating all surfaces including the microchip electrodes. The thickness was monitored using a quartz crystal mass monitor. After coating, the microchips were immediately exposed to the atmosphere, no post–treatment was performed.

An exploded schematic view of the deposited, granular thin film of tin oxide and

the microchip is shown in Fig 9.1. Although the surface conductivity between the electrodes was the quantity of interest here, there is no reason why the thin film could not be deposited on an insulating coating instead, removing the question of contact resistance (or polarization) between the electrodes and the thin film. In addition, it might be preferable to use the macrochip here because of the greater latitude in operating temperatures with the ceramic device. There will be an upper limit on coating thickness due to increasing film conductivity with thickness. At some point, the film would be too conducting to observe its behaviour with the frequency range of the microdielectrometer.

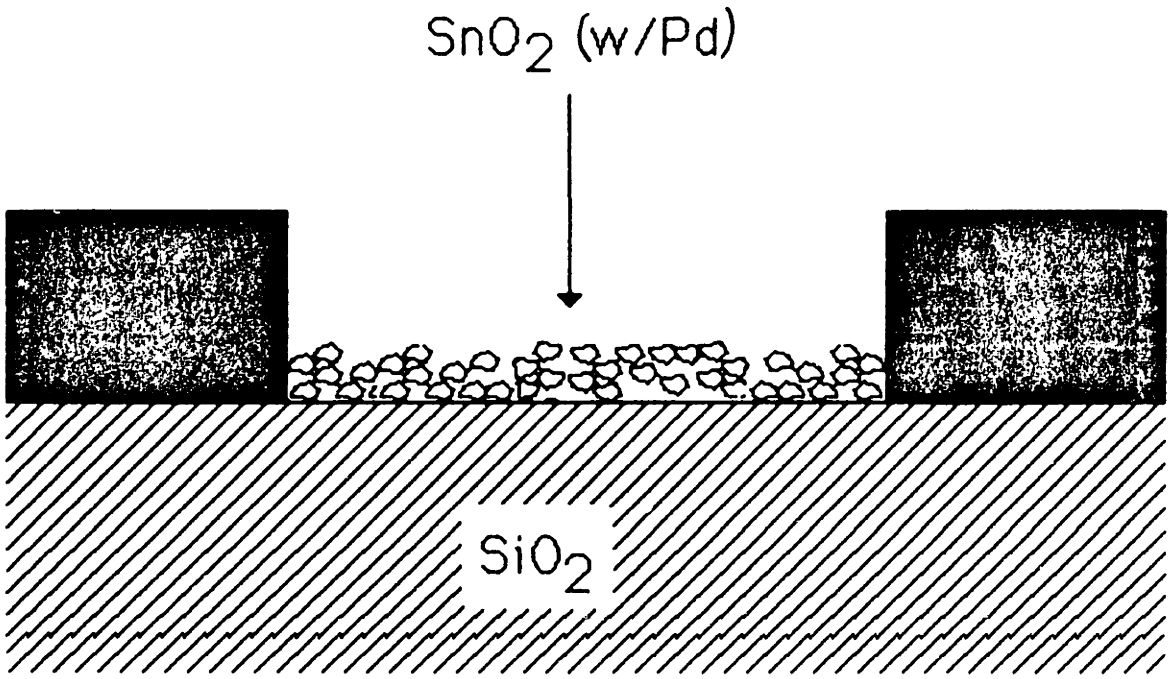


Figure 9.1: Schematic View of Tin Oxide Thin Film Deposited on a Microchip

The gases used were nitrogen or a factory mixture of 1000 ppm or 100 ppm H_2 in nitrogen. All the gases were of Matheson Purity grade (99.9995%) meaning $\leq$ 5 ppm total of oxygen, carbon monoxide, carbon dioxide, moisture, argon and total hydrocarbon content.

9.2.2 Experimental Results

Tin Oxide

A typical frequency response for the tin oxide coated microchips was shown in Fig. 5.20. That response was for microchip SnO2, having a 350 Åthick coating of tin oxide. As the external environment was varied, temperature or RH, the curves would remain the same shape and just slide up and down in frequency. Again normalizing the frequency to f_p a master curve can be produced (Fig. 9.2). This is reminiscent of the master curve behaviour of the moisture sensors, only now the response is due to a complex surface permittivity. As shown in Section 5.5, the response reflects what appears to be an ohmic surface conductivity with a dispersive surface permittivity, perhaps due to polarization at the electrodes. This response may be better modeled by separating the ohmic surface conductance from a barrier impedance due to polarization. The dispersion in the barrier impedance may then be estimated, perhaps yielding a power law dependence similar to those observed for the moisture sensors.

A plot of the estimated surface conductivity as a function of RH for microchip SnO2 is shown in Fig. 9.3. The dependence is linear on a log–log scale, producing a power law dependence of the surface conductivity on RH,

$$\sigma_{so} = \sigma_o(RH)^{\beta} \tag{9.1}$$

For these results, $\sigma_o = 3.33x10^{-12}$ S and $\beta = 1.7$. At this point it is appropriate to address the issue of possibly using this tin oxide coated microchip for moisture sensing in the oil. One problem with this approach is that the tin oxide, deposited by vacuum vaporization, is not bound very strongly to the material. Either a protective overcoat needs to be applied or the tin oxide must be deposited using a technique that produces better adhesion, such as sputtering. A more critical issue, though, is whether the tin oxide film would be sensitive to the small concentrations of moisture in the oil. It is felt that moisture sensitivity in air does not imply sensitivity in oil. One of the virtues of the bromobenzene moisture sensors was the amplification of changes in small moisture concentrations via bulk absorption. It is not clear that the thin tin oxide film will

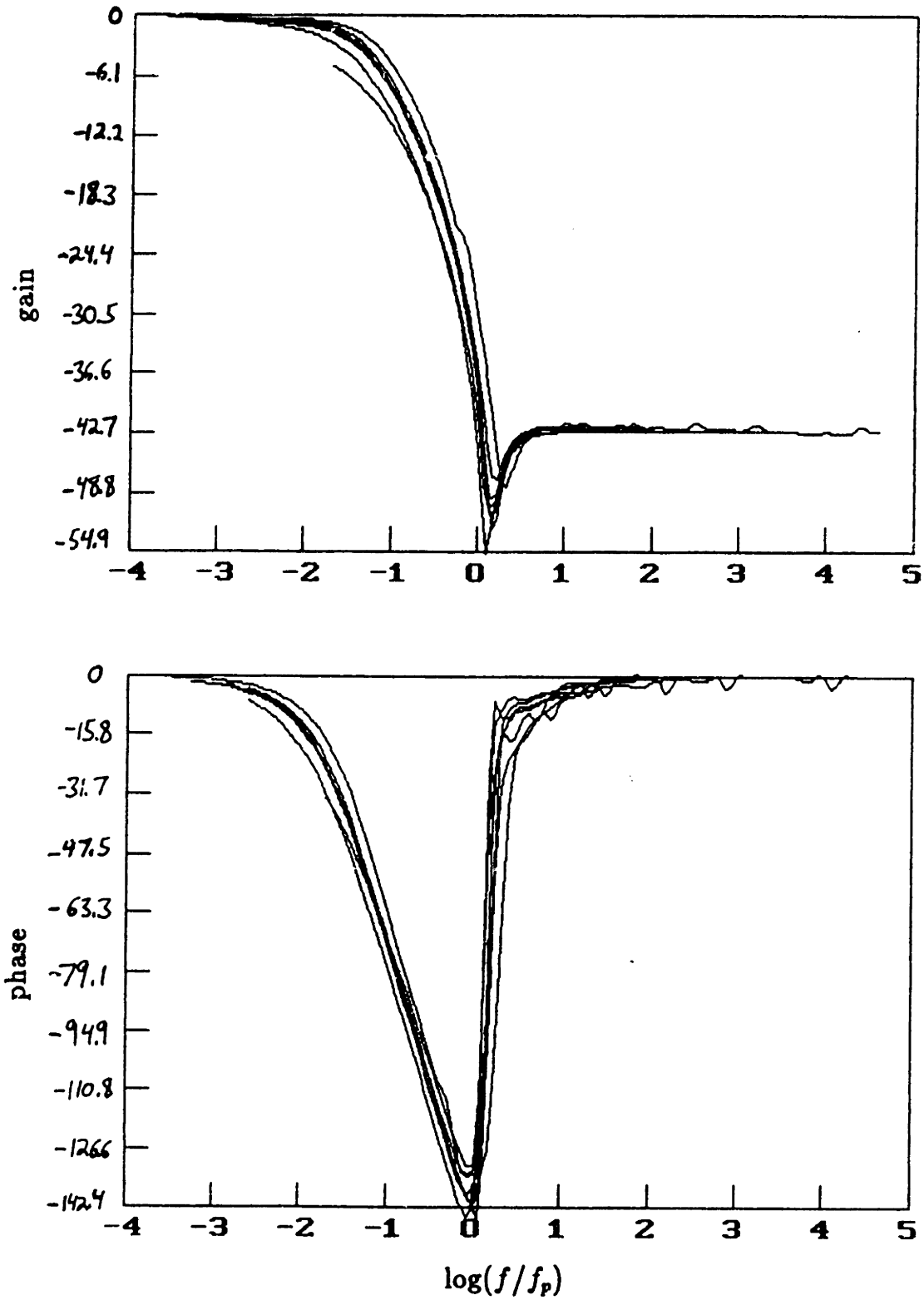


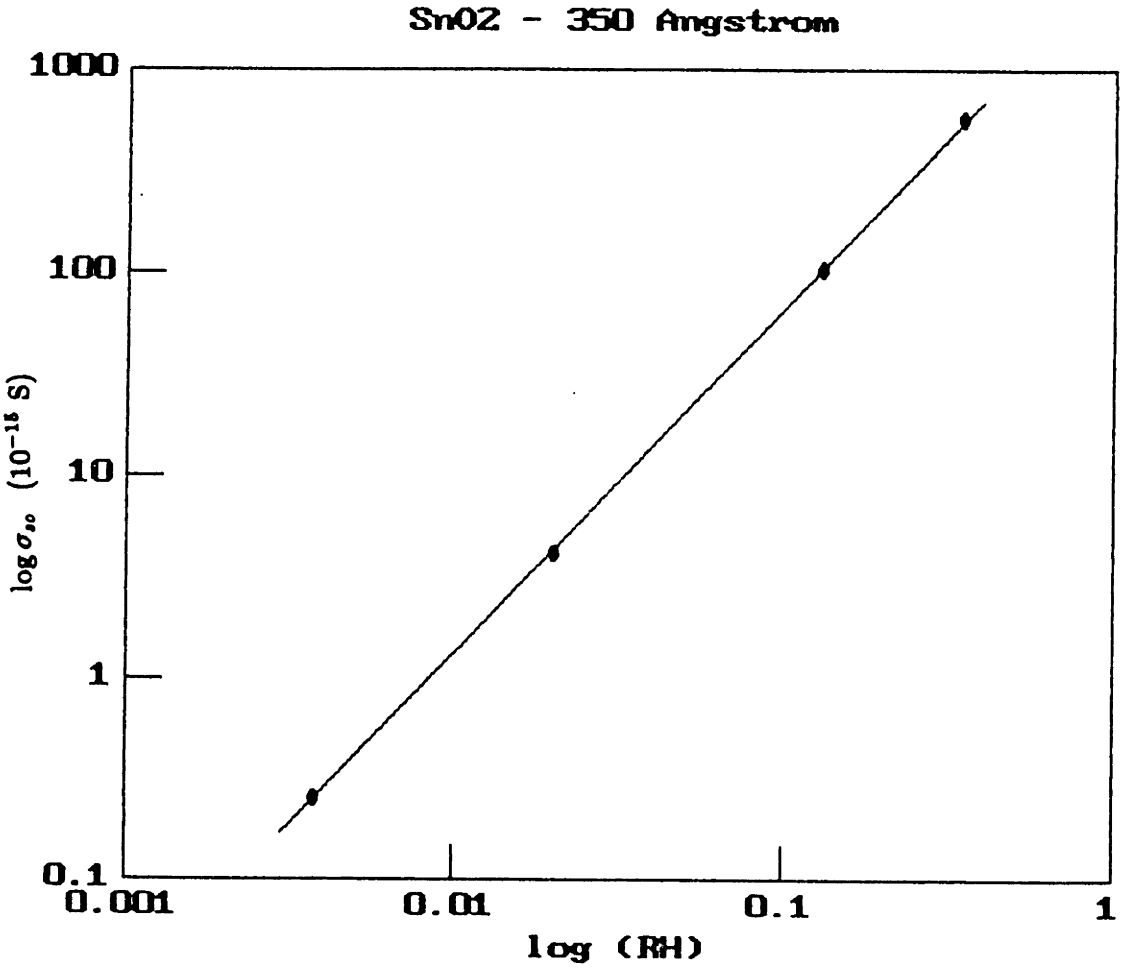
Figure 9.2: Master Curve for SnO2, a 350 Å Tin Oxide Coated Microchip, in Variable RH Environment.

produce the same effect.

Pd/Tin Oxide

The dominant effect of temperature on the response of the tin oxide thin films is shown in Fig. 9.4. This response was measured at 10 kHz using microchip SnO2, having in addition to the 350 Å thick coating of tin oxide, a 10 Å overcoat of palladium. This palladium overcoat was also vacuum deposited. It was thought that the palladium might serve as a catalyst and enhance the gas sensitivity, as observed in the literature. The main effect of the palladium appeared to be the dramatic increase in surface conductivity, necessitating the monitoring at high frequency. The measured phase at 10 kHz in Fig. 9.4 corresponds to the phase at 0.5 Hz in Fig. 5.20 (before the overcoating with palladium).

The microchip was placed inside a sealed beaker and exposed to either the dry nitrogen or the 1000 ppm hydrogen/nitrogen gas flow. Initially, the microchip was exposed to the dry nitrogen. There was a rapid but small decrease in gain, but no other significant change was observed. After roughly 4 minutes, the flow was switched to the 1000 ppm hydrogen/nitrogen gas mixture. Another rapid decrease in gain, followed by a slower decrease, was accompanied by a decrease in phase, increase in DC offset and decrease in temperature. Finally, after 25 minutes, the microchip was removed from the beaker and placed in a dessicator at 2–4% RH. The changes in the response produced by this action are on the same order as the gas composition changes. Although the removal of the microchip to the dessicator involved exposure to the ambient atmosphere, it is felt that the dominant effects were the change in RH and temperature. Notice that the change in gain, phase and temperature were the same regardless of gas composition but were opposite in direction upon removal to the dessicator. A plausible scenario is that the microchip was constantly being both dried and cooled by the gas flow, and absorbed moisture and heated up slightly upon removal from the gas flow. Once in the dessicator the absorbed moisture was slowly desorbing, as indicated by the slow decrease in gain during the second half of the observed time period. In any event, given the lack of temperature and humidity control, it does not appear that the gas sensitivity of these devices were truly tested.



$$\log(\sigma_{so}) = 1.7\log(RH) - 11.478$$

Figure 9.3: Relative Humidity Dependence of Estimated Surface Conductivity of 350 ÅTin Oxide Coated Microchip.

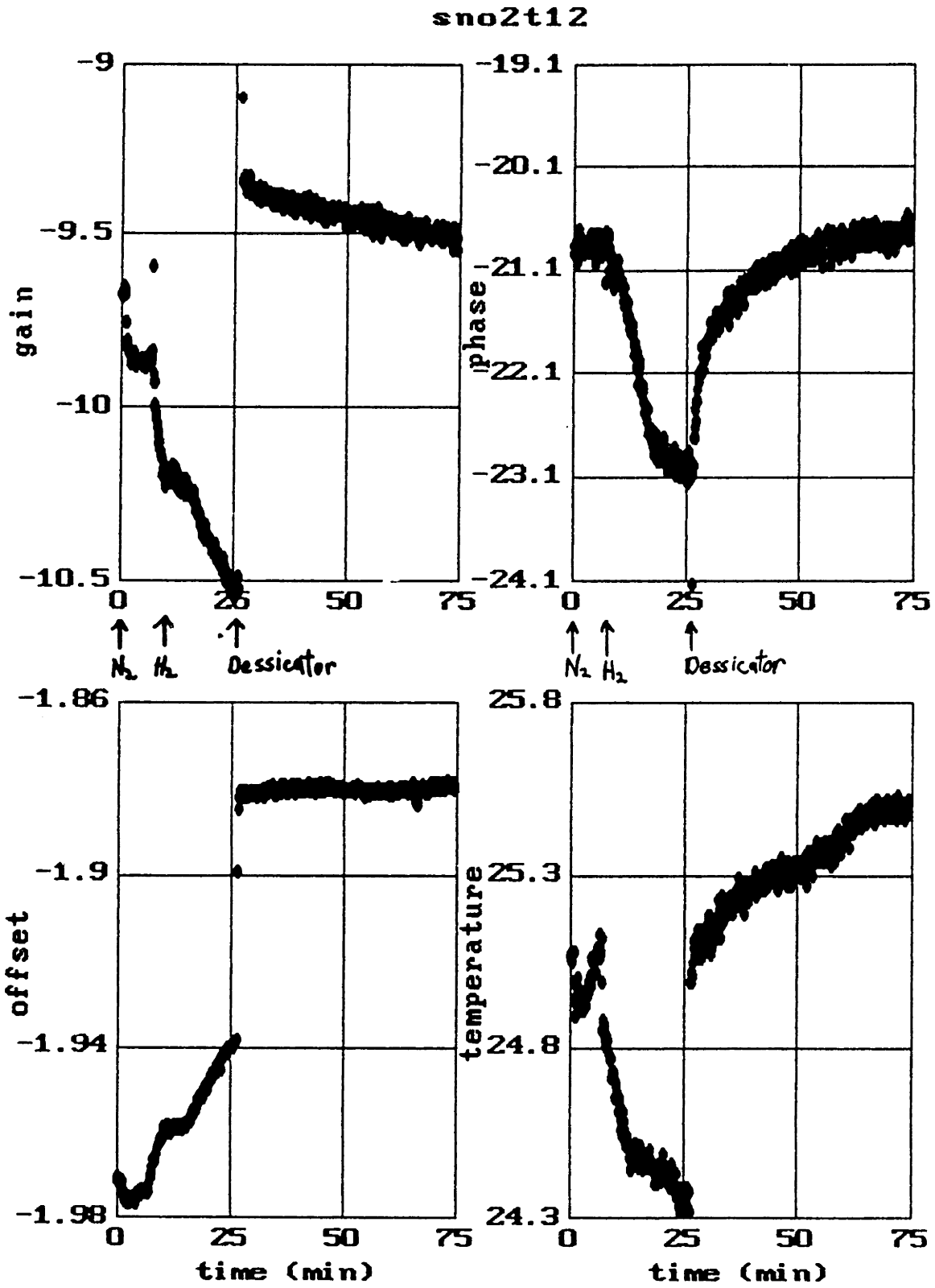


Figure 9.4: Measured Response at 10 kHz of Pd (10 Å)/Tin Oxide (350 Å) Coated Microchip in Gas Flow (Variable Composition).

9.3 Thin Film Palladium Resistor

9.3.1 Background

The approach explored here was to exploit the uniquely high solubility of hydrogen in palladium. Basically, once adsorbed onto the surface of palladium, hydrogen is rapidly absorbed into the bulk, giving rise to small, but measurable, decreases in bulk conductivity. Due to this unique property, it is expected that devices based on this principle will be highly selective to hydrogen in the presence of a gas mixture.

This behaviour is detailed by Lewis [110, Chapter 4], including descriptions of early results obtained using fine palladium wires. Percent changes in the resistance of the wires were related to the atomic ratio of hydrogen to palladium. The latter quantity was determined by measurement of gas volumes evolved during electrolysis. Changes in resistance of up to 70% were obtained at atomic H/Pd ratios of 0.7 at room temperature. Room and low temperature (77°K) experiments at low pressures, using evaporated Pd films (80–90 Å thick), exhibited much smaller percent changes in resistance [111]. Here, a pressure of 10^{-3} mm Hg of pure hydrogen, corresponding to roughly 0.1 H/Pd, resulted in a 1.5% increase in resistance. The kinetics of the reaction $H_2 \rightleftharpoons 2H$ (dissolved) on thin Pd and Pd–Ag foils (2.5–5 μm) was studied by Auer and Grabke [112]. It was shown that the rate determining step is the dissociation of the H_2–molecule on the metal surface. Therefore, the cleanliness of the surface, and the availability of surface adsorption sites, are very important in determining the magnitude change in the resistance and the speed of the transient response.

9.3.2 Experimental Results

Fabrication

The thin film palladium resistor was fabricated by the chemistry lab of Professor Mark Wrighton. Four palladium resistors, 500 Å thick by 100 μm wide and roughly 2.4 mm long, were sputtered onto an insulating substrate (Fig. 9.5). The substrate was composed of a silicon wafer base, a middle layer of thermally grown SiO_2, and an

upper layer of low pressure chemical vapor deposited (LPCVD) silicon nitride (Si_3N_4). Two of the resistors were sealed against exposure to the ambient atmosphere using a coating of Epoxi–Patch Kit (Hysol Aerospace & Industrial Products Div.), whereas the other two were exposed over 1/3 of their length (.8 mm). The 'chip' containing the resistors was then mounted and sealed in a glass tube with a septum at one end. This allowed testing of the resistor to be performed in an enclosed volume of roughly 0.4 mL.

Experimental Setup and Procedure

A two terminal measurement of the resistance was made using an HP3455A digital voltmeter, controlled by an IBM PC/XT. It was observed that, due to the small mass of the resistor, significant heating of the resistor occurred if the voltmeter was operated continuously in the resistance measurement mode and thus, always supplying current. The heat rise can be calculated assuming all the energy dissipated in the volume (V) of the resistor goes into heating

$$\Delta T = \frac{I^2 R}{\rho V c_p} dt \tag{9.2}$$

Using a current $I = .7$ mA, nominal resistance $R = 100\Omega$, density of palladium $\rho = 12x10^3$ kg/m^3, the dimensions of the resistor, and a heat capacity $c_p = .058$ cal/gm$-$°C (243J/kg in MKS units) yields

$$\Delta T = 6.73x10^3 dt \tag{9.3}$$

With the current applied continuously, a significant rise in temperature can occur causing a constant increase in resistance, as was observed experimentally. Therefore, the voltmeter was left in a voltage measurement mode (open circuit) except when taking the periodic resistance measurement.

The operating temperature of the device was controlled using an Omega 6100 temperature controller and a thermocouple. The resistor, and a thermocouple, were placed inside a larger diameter pyrex tube having a heating coil wrapped around it. Paper was used to fill up the remaining volume within the tube so as to prevent air

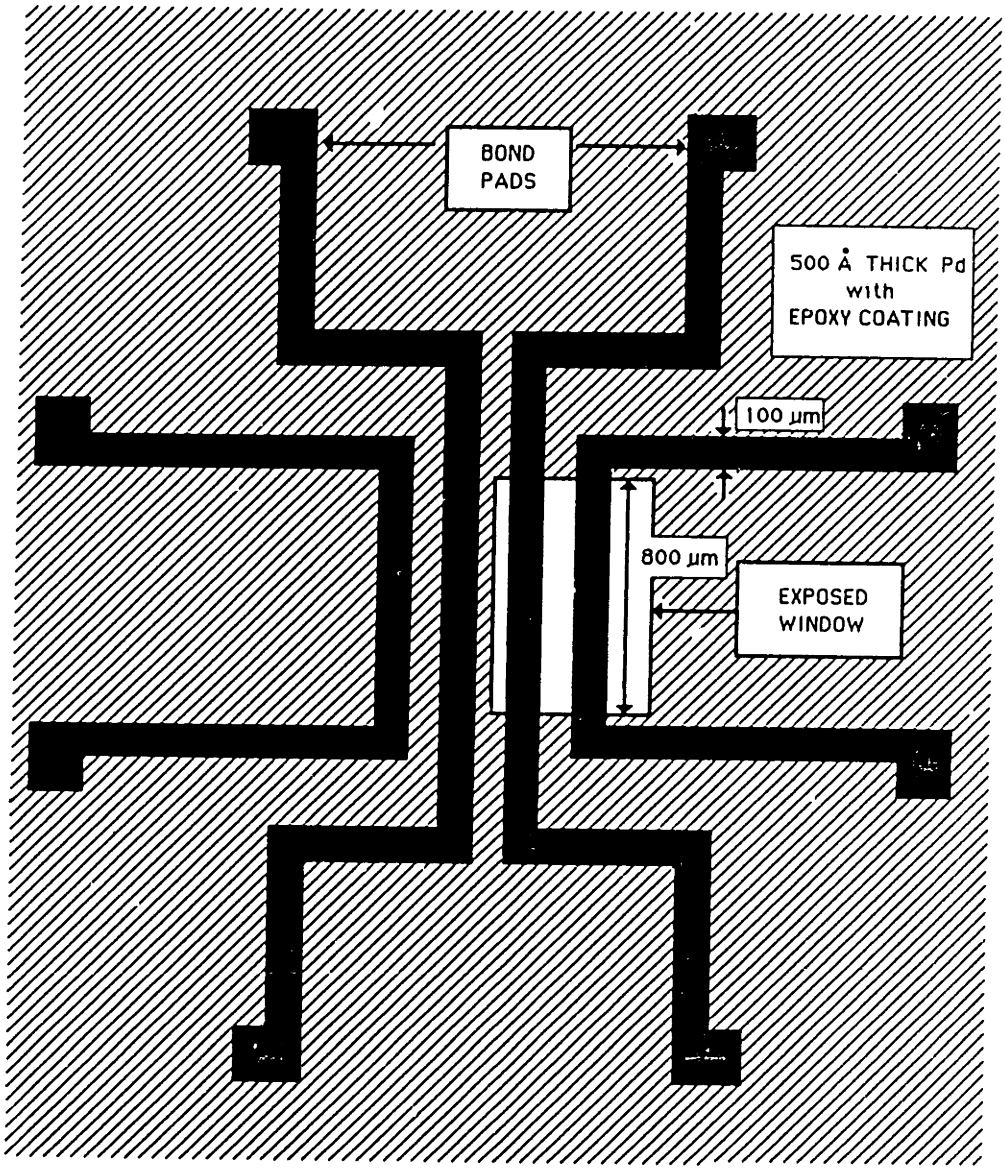


Figure 9.5: Layout of Thin Film Palladium Resistors.

circulation.

The same high purity gases mentioned above were flowed continuously through the enclosed volume via inlet and outlet syringe needles piercing through the septum. Using the experimental flow rates (roughly 280 mL/min.) and needle length, it was calculated that the gas temperature would be in equilibrium with the device temperature long before it actually reached the device.

Tests were conducted after the device had reached thermal equilibrium and had been exposed to the dry nitrogen flow for roughly one hour. Without this initial conditioning with the nitrogen, the early results would show a decrease in resistance when exposed to either the nitrogen or the H_2 /nitrogen mixture. The devices would be left exposed to room air overnight, thus the conditioning may be drying the surface of the thin film resistors and/or removing other adsorbed species, such as oxygen, occupying valuable surface adsorption sites.

Results

At a temperature of 225°F, the device is sensitive to both 1000 ppm (Fig. 9.6a) and 100 ppm (Fig. 9.6b) H_2 in nitrogen. The introduction of hydrogen into the resistor results in an increase in the resistance, as was observed in the references mentioned above. The transient response has both a rapid ($\sim$ 1–5 minutes) and a slow ($\sim$ 10–100 minutes) component. The slow component may be due to the slow diffusion of hydrogen through the epoxy barrier, the lateral diffusion of hydrogen throughout the length of the resistor to regions not exposed to the atmosphere, or slow diffusion of hydrogen into the substrate. The characteristic diffusion time is

$$\tau_D = \frac{\delta^2}{D} \tag{9.4}$$

where D is the molecular diffusivity and δ is the characteristic length for the diffusion process. For hydrogen in palladium, $D \sim 10^{-9}\text{m}^2/\text{s}$ at room temperature [113]. The vertical diffusion through the thin resistor ($\delta = 500$ Å) is very rapid ($\tau_D \sim 2.5\mu\text{s}$) but can be much longer in the lateral direction ($\delta \sim 1$ mm $\rightarrow \tau_D \sim 10^3$ s). The responses appear to be reversible to a large degree. The magnitude change in resistance at

1000 ppm is 0.45Ω versus 0.06Ω at 100 ppm. The expected dependence would be as the square root of the concentration [112].

Test were also conducted at a reduced temperature of 100°F. The response to 1000 ppm hydrogen (Fig. 9.7a) is comparable to the higher temperature result (Fig. 9.7b). However, at 100 ppm hydrogen the response (Fig. 9.7b) is beginning to get lost in the experimental error. Also, the transient response at the lower temperature and concentration appears to be much slower than the higher temperature response.

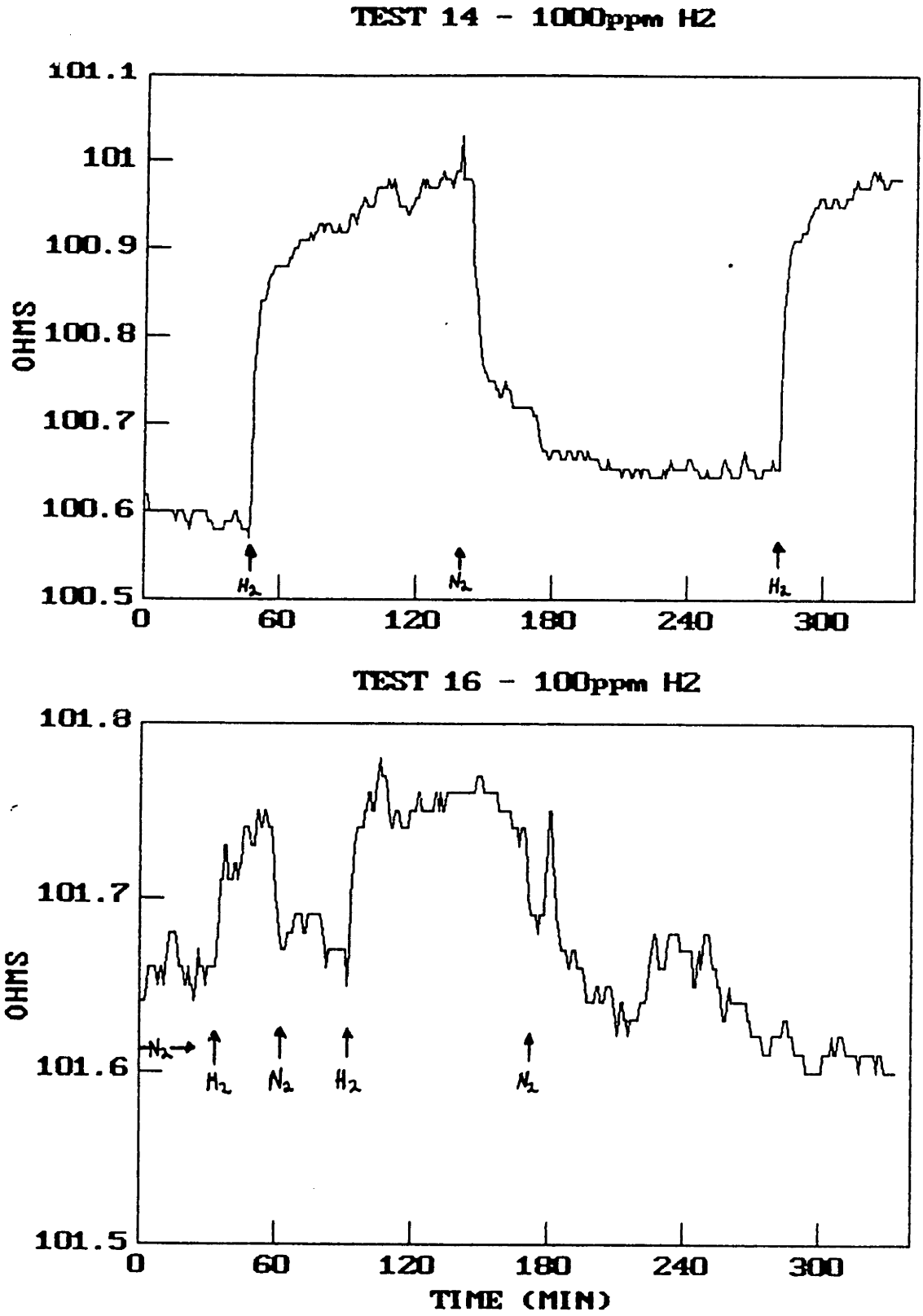


Figure 9.6: Response of Thin Film Palladium Resistor to a) 1000 ppm and b) 100 ppm H_2 in Nitrogen at 225°F.

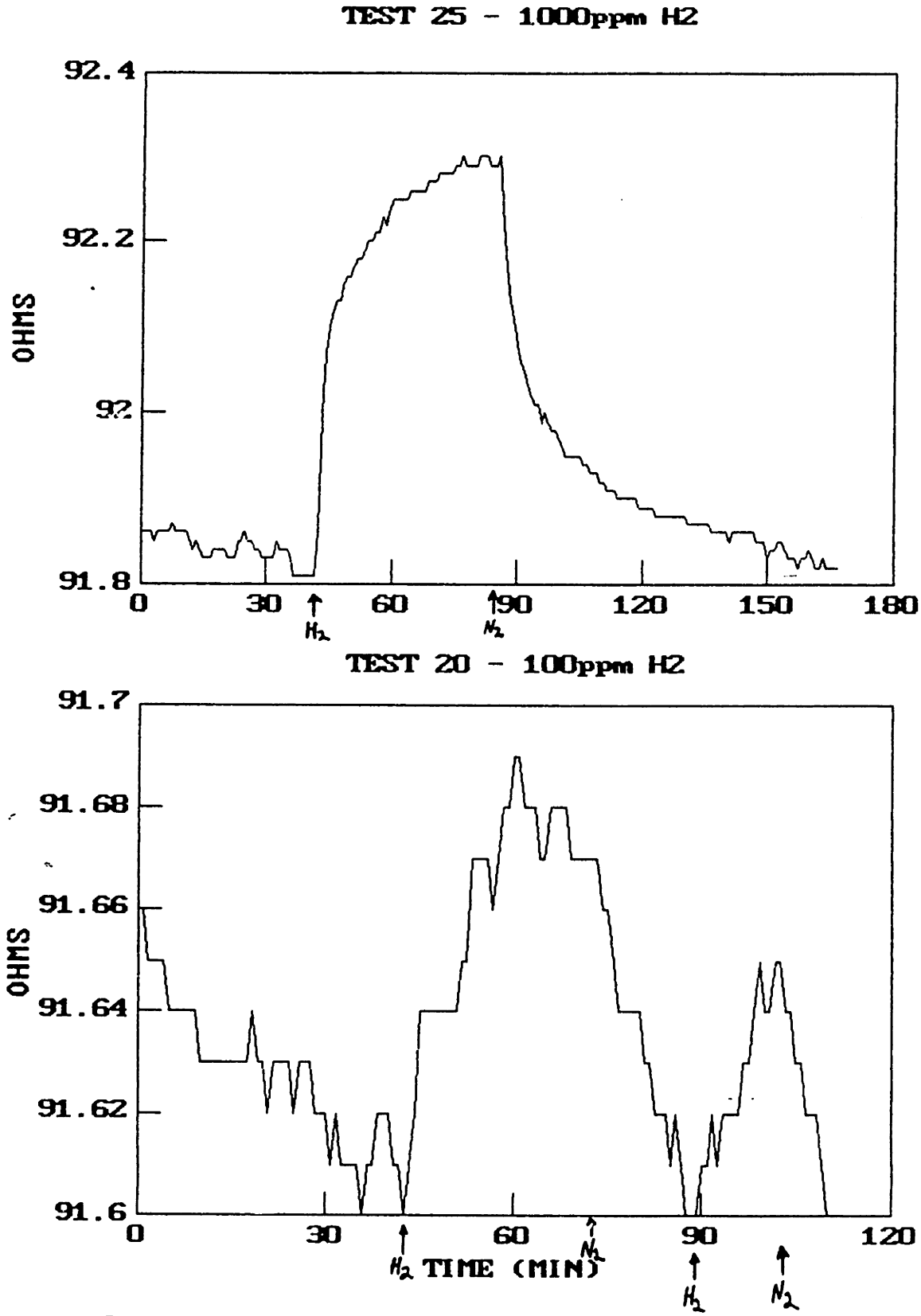


Figure 9.7: Response of Thin Film Palladium Resistor to a) 1000 ppm and b) 100 ppm H_2 in Nitrogen at 225°F.

Chapter 10

Conclusions and Future Work

10.1 Continuum Model

A modal representation relating the gain–phase response of the microdielectrometer to the continuum properties of a medium has been formulated so as to retain generality and flexibility in its application to various linear systems. This approach greatly extends the usefulness of the device by providing a framework for interpreting data obtained for heterogeneous media in terms of absolute continuum parameters rather than "lumped equivalents".

There are several areas that could either be improved or generalized to further increase the usefulness of the model. One area is computation speed. By taking advantage of fast fourier transform (FFT) techniques, the time savings for performing the summations in (2.22a) and (2.24a) could possibly be increased even more than already obtained using the approach outlined in Appendix A. A further generalization of the method would be to allow for the capability of representing the entire boundary with collocation points. This approach would enable implementation of a more detailed description of the boundary condition at the electrodes, possibly allowing the incorporation of electrode polarization effects. This capability might also allow a method with which the finite thickness of the electrodes could be simulated as well.

10.2 Parameter Estimation

Development of simple single and multiparameter estimation routines incorporating the continuum model provides the capability to infer continuum properties and geometry from experimental data. The key to parameter estimation, as applied to interdigital electrode structures, is in identifying the types of data that can be mapped into the continuum properties and geometry. Illustrated here have been one dimensional schemes for the identification of film thickness, the complex bulk permittivity of finite thickness films and the complex surface permittivity of thin films. These schemes have exploited the analyticity of the functional dependence of the gain–phase response on these parameters. Also developed was the estimation of both the thickness and permittivity of a film from two measurements of gain. Here, the known variable is the dielectric medium above the layer. This technique was useful for estimating the thickness and bulk permittivity of thin film coatings.

Several experiments, involving measurements of oil, paper insulation and thin polymer and metal films, demonstrated both the utility of the parameter estimation routines and the validity of the continuum model.

It was also demonstrated that a unique interpretation of spatial distributions in dielectric properties in a heterogeneous medium could not be made from the temporal frequency response unless several assumptions regarding the distribution of relaxation times, and the dispersive nature of the properties, were made. However, by varying the fundamental spatial wavelength of the electrodes, a new approach can be taken to sorting out distributions of properties. Exploiting the 'imposed ω–k' approach, and exponential decay of the electric fields normal to the plane of the electrodes, the multiparameter estimation problem was reduced to a series of sequential single parameter estimations that yielded smooth distributions of complex bulk permittivities. In conjunction with this scheme, cases were examined where known discontinuities could be estimated using a two parameter, instead of single parameter, search.

For several of these multiparameter estimations, iterating to refine the good estimates obtained on the first pass through the data appeared to actually produce

diverging subsequent estimations. It is believed that at least part of the problem lies in using the stair–step representation, introducing unrealistic discontinuities that contribute to the instability of the iterations. The next step would be to implement the exponential representation of Fig. 4.10b. With this representation, the spatial distribution is continuous, with the discontinuity first appearing in the first derivative of the distribution. Another possible remedy might be to go from an exactly constrained to an overspecified problem. That is, the number of experimental datapoints is greater than the number of parameters to be estimated. This approach may yield a convergent search but the cost is the increase in computation due to the necessity of performing the multiparameter estimation simultaneously rather than as a sequence of single parameter estimations.

All of the estimation routines described are simple Newton or secant method searches. The robustness of these routines, particularly the two parameter estimation and the multiparameter estimation using variable spatial wavelength data, could be improved using 'smarter' search routines such as Levenburg–Marquardt [79, Section 4.2.2]. Also, the confidence level of the estimates has not been stated on a quantitative basis.

As demonstrated in Section 5.6, there are occasions where parameters may be estimated from the frequency response. It would be valuable to perform not only single, but multiparameter estimations using the frequency response. An example of the utility of such a routine would be to estimate both the layer thickness and complex bulk permittivity of the evaporating oil layer mentioned in Section 5.2.1.

10.3 Complex Bulk Permittivity Of Oil

The microchip and macrochip have been successfully adapted to provide measurement of the complex bulk permittivity of transformer oil. This effort required the combination of passivation of the electrode structures to block moisture adsorption, and development of the continuum model and parameter estimation routines to allow estimation of the dielectric properties of the oil while correctly accounting for the dielectric

properties of the passivating medium. Parylene C was found to provide an excellent moisture barrier and appears to be compatible with transformer oil.

In the presence of dry flowing oil, interfacial problems would arise when using the passivated microchip. This can be avoided by placing the sensor in a minimal oil flow region. Another option is to use the longer spatial wavelength of the macrochip to reduce the sensitivity to localized effects at the interface. The problem with the present macrochip, that of insensitivity in gain, is a design problem, not a fundamental limitation. Therefore, it is felt that a redesigned passivated macrochip might prove to be the best solution.

With this sensor, the opportunity now exists to gain experience on monitoring the complex bulk permittivity of the oil. Therefore, the next step is to begin on–line monitoring in the test facility. A database must be built up to determine whether there is any useful information about the status of the oil to be gained from this measurement. There is very little, if any, experience in the literature on monitoring this quantity due to the lack of a simple measurement method (and perhaps lack of interest as well).

10.4 Moisture Sensors

A design strategy for developing electrochemical sensors in general, and moisture sensing in particular, using thin film microdielectrometry, was demonstrated. The concept of coating an electrode structure with a thin film whose dielectric properties were sensitive to the chemical constituent of interest in the oil was successfully applied to moisture sensing using a plasma deposited bromobenzene coating on a microchip. Moreover, this effort represents the first time the moisture content of the oil has been measured *in situ*. The coating behaved as a membrane in that the moisture content in the coating was directly related to the absolute moisture content of the oil. Further, the sensitivity to moisture in the oil, on the order of parts per million, produced measurable changes in the dielectric response of the coating when using the microdielectrometer. This dielectric response displayed a shape that was invariant

with temperature and moisture content (as long as the relaxation in the coating occurred at frequencies higher than the relaxation in the oil), producing a master curve that allowed for easy calibration. Coating thickness was not seen as a crucial parameter, a range of 2–7 μm in thickness appeared to be appropriate when using the microchip. The film morphology, a biphasic structure of particles embedded in a rigid film, was concluded to be responsible for the permeation of moisture into the film.

This moisture sensor is by no means a finished product, ready to be commercialized. The critical problem is the lack of stability of the coatings over time and operating temperature. More experimental results on coated microchips need to be performed to characterize the dependence of aging on temperature and time. Controlled experiments should be performed to verify that the coatings degrade only when immersed in oil and not when exposed to the atmosphere. It would be useful to examine, under an SEM, the coatings after aging in the oil over a long period of time. Perhaps a change in the biphasic structure is resulting in the decreased moisture sensitivity. As discussed in Section 8.3, there are several possible fabrication techniques that may improve the coating stability. If the plasma deposited bromobenzene proves inadequate, then other polymeric materials that have demonstrated better stability should be tried. For example, plasma deposited HMDSO appears to be a material that has been plasma polymerized to a high degree of stability and has been investigated by many researchers. A major constraint has been the maximum temperature to which the microchip can be subjected, due to the polyimide used on the chip itself and the Kapton used to encase the microchip. This limits the number of deposition processes that may be investigated. It may be helpful to either coat the microchip before the polyimide is applied or coat a macrochip. The increased maximum temperature may open the door to consideration of coating materials via alternative processes. If the macrochip is used, the coating should still be thin, on the order of micrometers, so as to maintain a rapid transient response. In any case, the procedure for developing a moisture sensor using other materials would be very similar to the path taken in this work.

An important unknown, perhaps critically related to the aging issue, is the physi-

cal mechanism responsible for electrical conduction in the bromobenzene. It has been demonstrated that the process is dispersive and references were made to several physical models for interpreting the dispersion, but more physical and chemical analysis of the coating needs to be done. Many of the analytical techniques used for examining plasma deposited films, described in Appendix C, remain to be applied to the study of this material. Given the importance, in the literature, ascribed to the presence of free radicals in the film, more attention should be given to the free radical concentration, its dependence on time, temperature and environment, and its relevance to the electrical conduction process.

One possible alternative for moisture sensing would be to use the paper insulation itself as the moisture sensitive material. Based on the discussion of Section 6.1, using a small mass of paper in a large volume (mass) of oil, the moisture content of the oil would determine the equilibrium paper moisture content. As shown by Li [23, Section 5.3], the dispersion in the paper (Fig. 5.24) can be normalized to the moisture content of the oil. That is, this dispersion can be normalized into a master curve, as was done for the plasma deposited bromobenzene. Therefore, a calibration of moisture content vs. f_p can again be made. The most likely location for the moisture sensor will be in the oil circulation loop, outside the main body of the transformer. In this region, the oil temperature never gets extremely hot, thus the paper should not suffer much degradation and remain relatively stable in its properties.

Once the moisture content in the oil is measured in an on–line fashion, the next step is to interpret the results. There are two modes of analysis here, one dynamic and the other static. A primitive attempt at relating the moisture content in the oil to the moisture content of the paper and the condition of the oil was outlined in Section 6.2.1. Under normal operating conditions a static model, such as the one represented by (6.5), is established, complete with parameter values. In a dynamic mode, on–line moisture measurements are made every 5 minutes, as demonstrated in Section 7.2. These measurements are then compared to predictions made using a dynamic model describing the moisture distribution between the paper and the oil as a function of one or more temperatures. This dynamic model may or may not be the

same as the static one, depending on how rapidly the moisture redistributes within the transformer. For the residuals computed in Section 7.2, the models were taken to be the same. However, other models, perhaps including information on the temperature and oil flow distribution within the transformer, may be more appropriate. The point is that the parameters used in the static model must be related to the parameters used in the dynamic one. Over a period of time on the order of 24 hours, significant residuals will indicate, on a basic level, a change in the static model parameters. In turn, this change represents the degree to which the moisture level deviates from normal operation and indicates a possible change in the state of the paper or the oil.

10.5 Gas Sensors

With the decision to use FTIR spectroscopy for on–line gas analysis, the only relevant gas of interest not detected using this apparatus is hydrogen. Two attempts were made at developing hydrogen gas sensors based on knowledge obtained from the literature. Doped tin oxide, known to be sensitive to reducing gases such as hydrogen, was deposited on a microchip. However, the results were not very conclusive. Several factors may be at work here: 1) the low temperature limitation of the microchip, 2) lack of proper doping levels using a noble metal such as palladium, 3) the tin oxide may need to be sintered to produce the correct granular structure, and 4) better control of the experimental temperature. Using a macrochip, rather than a microchip, perhaps the considerations of 1) and 3) may be corrected. Another approach might be to capitalize on the results using discontinuous ultrathin (< 100 Å) films of palladium, obtained by Jelley and Maclay [47]. There, the sensitivity of the impedance of these films to hydrogen at room temperature, and carbon monoxide at elevated temperatures, was demonstrated. Finally, an approach that parallels the sintered tin oxide, but uses plasma deposited organometallic films instead, may prove to be very adaptible to the microchip. Starting with tetramethylgermanium and tetramethyltin monomers, Inagaki and Hashimoto plasma deposited these monomers onto an interdigital structure [114]. Heating these structures anywhere from 200–400°C, the DC resistance was mon-

itored as the ambient gas concentration was varied. Sensitivity to combustible gases such as propane, hydrogen, methane, and carbon monoxide, was demonstrated at concentrations of 0.1 molar % (1000 ppm). In addition, sensitivity was increased by the vacuum evaporation of a thin film ($\sim$ 100 Å) of palladium chloride on top of the plasma deposited layer. Using the microdielectrometer, the impedance of these films at lower temperatures may be observable and provide a hydrogen gas sensor that operates in a manner very similar to the plasma deposited bromobenzene moisture sensor. This approach might prove to be better suited to adaptation to the microdielectrometer due to the upper limit on the measurable surface conductivity of the thin metal oxide films versus the required grain structure necessary for gas sensitivity.

Preliminary work using a thin film (500 Å) palladium resistor yielded encouraging results. Sensitivity to H_2 down to a 100 ppm level at temperatures $\geq 100°$ F was demonstrated. More data needs to be taken to quantify the sensitivity to H_2 concentration. Heating the device is important as moisture will corrupt the measurement. It is believed that the sensitivity could be improved further, perhaps by an order of magnitude, using a bridge arrangement. This differential measurement, using all 4 palladium resistors, would also automatically compensate for local temperature variations. An improved barrier to prevent hydrogen from affecting the two reference resistors will be needed. Although the device is expected to be selective to hydrogen due to the nature of the physical mechanism responsible for the change in resistance, this selectivity has not been verified experimentally. It would also be interesting, though not crucial, to test the device immersed in the oil.

The goal of developing microfabricated sensors, selectively tuned to the various gases of interest and immersed in the oil, is a much longer term project. Some idea of the large effort that is currently being given to developing methods of derivatizing and testing microelectrode structures using various redox active polymers can be obtained from [115]. At present, the emphasis has been on studying the behaviour of redox polymers when there is a chemical change in the environment. With the polymer suitably specialized by a catalyst or electroactive center, the chemical change can be translated into a physical change in the electron distribution within the polymer resulting in an

alteration of the conductivity or standard potential of the polymer. This resultant electrical change can be monitored using a two or three terminal configuration. In general, there is a literature of redox polymers that interact with molecules such as methane, carbon monoxide, carbon dioxide, and ethylene. This fact, coupled with the concepts to modify electrodes with polymers consisting of the chemically sensitive redox subunits, leads to the opportunity to fabricate a variety of chemically responsive devices.

10.6 Electrode Terminations

It would be desirable to have the capability of operating the microdielectrometer in either a high or low impedance mode. That is, being able to switch between a measurement of a complex gain using the floating gate potential and a measurement of the coupling admittance (impedance) using the current flow to a virtual ground. This would remove the limitation on extracting information at gains near unity and provide a broader view of the dispersive nature of materials such as the plasma deposited bromobenzene. It would also allow a check on the linearity of the dielectric behaviour by providing a larger range over which the Kramers–Kronig relationship may be verified.

10.7 Double Layer Interactions

One of the benefits of the microfabricated nature of the microchip, coupled with the modeling approach taken here, is to be able to study phenomena having a characteristic length on the order of micrometers. An example of this is the diffuse double layer which, for the insulating oils used in transformers, can be anywhere from 1 100 μm in length. This charge separation layer, occurring at the paper insulation–oil interface, has been cited as the source of charge giving rise to static electrification problems [116,117]. Using a microchip at low field strengths, it may be possible to perturb the double layer and observe the response. Experiments demonstrating the feasibility of this approach are reported in Appendix D. Development of the appropriate model for

the perturbed double layer, using self consistent charge generation and recombination terms, will provide better insight into the structure of the double layer. Understanding the equilibrium at the paper–oil interface is a necessary step along the path to explaining what occurs in an energized transformer when charge is injected into the oil.

10.8 Applications to Other Physical Domains

Finally, it is not to difficult to imagine applying the concepts developed here to a wide variety of applications, ranging from measuring a complex susceptibility using current loops to measuring thermal properties using meandering resistors. In each case, the combination of microfabrication techniques, the 'imposed ω–k' approach, the spectral solution of the resulting mixed boundary value problem, parameter estimation techniques, and computing power, allow absolute measurements of physical parameters and observations of physical phenomena not previously available.

Appendix A

Rapidly Convergent Series

The individual terms in (2.22a) and (2.24a) that are to be summed over the indice n have the general form of

$$\sum_{n=1}^{\infty} F_n \frac{\sin(n\alpha)\cos(n\beta)}{n^2} \tag{A.1}$$

A more rapidly convergent series was used to reducing the computational time required for this summation. Expression (A.1) can be expanded to the form of

$$\sum_{n=1}^{\infty} H_n \frac{\sin(n\gamma)}{n^2} \tag{A.2}$$

Assuming $H_n \rightarrow H_{min}$ as $n \rightarrow \infty$ the sum

$$H_{min} \sum_{n=1}^{\infty} \frac{\sin(n\gamma)}{n^2} \tag{A.3}$$

will be computed and the difference between H_n and H_{min} will be accounted for later on. From Oberhettinger [118] comes the expression

$$\begin{aligned} \sum_{n=1}^{\infty} \frac{\sin(nx)}{n^2} &= -\int_0^x \ln\left[2\sin(\theta/2)\right]\, d\theta \\ &= -x\ln[2] - \int_0^x \ln\left[\sin(\theta/2)\right]\, d\theta \end{aligned} \tag{A.4}$$

Using a series expansion [119]

$$\begin{aligned}\ln[\sin(t)] &= \ln[t] - \frac{t^2}{6} - \frac{t^4}{180} - \frac{t^6}{2835} - \cdots - \frac{-1^{m+1}2^{2m}B_{2m}t^{2m}}{2m(2m!)} \\ &= \ln[t] - \sum_{m=1}^{\infty} \frac{-1^{m+1}2^{2m}B_{2m}t^{2m}}{2m(2m!)} \quad \text{for} \quad t^2 < \pi^2\end{aligned} \tag{A.5}$$

where B_i are the Bernoulli numbers, and substituting in (A.4) yields

$$\begin{aligned}\sum_{n=1}^{\infty} \frac{\sin(nx)}{n^2} &= -x\ln[2] - \int_0^x \left[\ln[\theta/2] - \sum_{m=1}^{\infty} \frac{-1^{m+1}B_{2m}\theta^{2m}}{2m(2m!)}\right] d\theta \\ &= x - x\ln[x] - \sum_{m=1}^{\infty} \frac{-1^{m+1}B_{2m}\theta^{2m+1}}{2m(2m+1)(2m!)}\end{aligned} \tag{A.6}$$

Due to the factorials present in the denominator in the summation term of (A6), only 5 to 10 terms need to be computed to obtain a high degree of accuracy (comparable to computing the first 1000 terms of the original series). At this point, a calculation of the terms for which H_n is significantly different from H_{min} is performed and the difference is added to the results of (A.6).

Appendix B

Surface Capacitance Density of Multiple Layers

The medium may actually be comprised of a finite number of layers, each layer described by a set of parameters, or layers may be used to approximate what is actually a smoothly inhomogeneous material. With the objective of describing either of these situations, suppose that the medium is composed of P layers, as shown in Fig. B.1. The jth layer has a thickness d_j and an upper surface designated by j. Surface properties of the $P+1$ surface, the electrode–medium interface, are handled differently from other continuum properties and are designated by ϵ^*_{so}. There is no ϵ^*_{s1} as this surface property is associated with the $j=1$ surface and will normally be at infinity. Solution of the field laws within the layer gives the transfer relations [75, p. 2.33],

$$\begin{bmatrix} \hat{D}_n^{(j)'} \\ \hat{D}_n^{(j+1)} \end{bmatrix} = \begin{bmatrix} \hat{A}_{11}^{(j)} & \hat{A}_{12}^{(j)} \\ \hat{A}_{21}^{(j)} & \hat{A}_{22}^{(j)} \end{bmatrix} \begin{bmatrix} \hat{\Phi}_n^{(j)'} \\ \hat{\Phi}_n^{(j+1)} \end{bmatrix} \tag{B.1}$$

where $\hat{D}_n^{(j+1)}$ and $\hat{D}_n^{(j+1)}$ are respectively the complex amplitudes of the nth Fourier components of the dielectric flux density and potential, evaluated just above the lower interface of the layer and $\hat{D}_n^{(j)'}$ and $\hat{\Phi}_n^{(j)'}$ are respectively these quantities evaluated just below the upper interface. In the case of a layer having a uniform complex permittivity [75, p. 2.33],

$$\hat{A}_{22}^{(j)} = -\hat{A}_{11}^{(j)} = \epsilon_j^* k_n \coth(k_n d_j)$$
$$\hat{A}_{12}^{(j)} = -\hat{A}_{21}^{(j)} = \frac{\epsilon_j^* k_n}{\sinh(k_n d_j)} \tag{B.2}$$

At the jth interface, the potential is continuous, but there is a complex surface permittivity ϵ_{sj}^* and hence a discontinuity in the dielectric flux.

$$\hat{\Phi}^{(j)} = \hat{\Phi}^{(j)'} \tag{B.3}$$

$$\hat{D}_n^{(j)} - \hat{D}_n^{(j)'} + k_n^2 \epsilon_{sj}^* \hat{\Phi}_n^{(j)} = 0 \tag{B.4}$$

The surface capacitance density of the jth interface is defined in terms of the quantities evaluated just above that interface.

$$\hat{C}_n^{(j)} = \frac{\hat{D}_n^{(j)}}{\hat{\Phi}_n^{(j)}} \tag{B.5}$$

From these equations, it follows that

$$\hat{C}_n^{(j+1)} = \hat{A}_{22}^{(j)} + \frac{\hat{A}_{21}^{(j)} \hat{A}_{12}^{(j)}}{\hat{C}_n^{(j)} - \hat{A}_{11}^{(j)} + k_n^2 \epsilon_{sj}^*} \tag{B.6}$$

This expression can be used repeatedly, starting from the top layer ($j = 1$) and working down to the Pth layer. By definition, the surface capacitance density called for in evaluating the complex gain is

$$\hat{C}_n \equiv \hat{C}_n^{(P+1)} \tag{B.7}$$

In the case where the first surface is at infinity, the mutual terms in (B.6) are zero so that $\hat{C}_n^{(2)}$ can be evaluated without $\hat{C}_n^{(1)}$. For the finite thickness layer bounded by an infinite half space, $P = 2$ and (3.3) follows from evaluation of (B.6) with $\hat{C}_n^{(2)} = \hat{A}_{22}^{(1)}$.

By flipping Fig. B.1 about the electrodes 180° the multiple layer $\hat{C}_n$ for the half space below can be derived. The form of the solution is the same as above with the

only difference being several minus signs. Thus, below the electrodes the quantity required is again (B.7) where P is now the number of layers below, and

$$\hat{C}_n^{(j+1)} = -\hat{A}_{22}^{(j)} + \frac{\hat{A}_{21}^{(j)}\hat{A}_{12}^{(j)}}{\hat{C}_n^{(j)} + \hat{A}_{11}^{(j)} - k_n^2\epsilon_{sj}^*} \tag{B.8}$$

where the $\hat{A}_{ij}$ are as previously defined and $\hat{C}_n^{(2)} = -\hat{A}_{22}^{(1)}$.

The ground plane of the microdielectrometer can be accounted for by using a very large surface conductivity at that boundary. This will result in the expression given in (3.1).

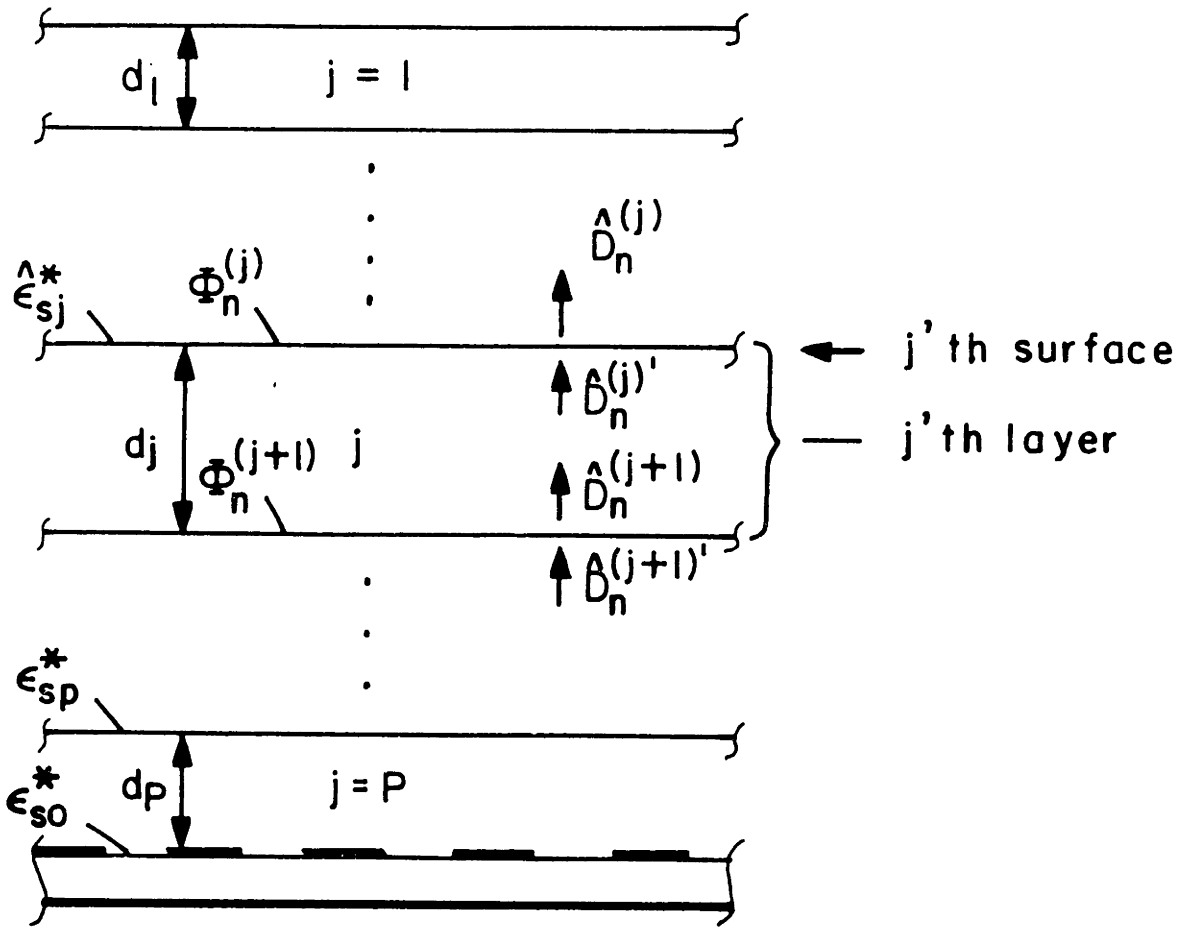


Figure B.1: Medium Above Electrodes Represented by a Multilayered Structure of P homogeneous layers.

Appendix C

Plasma Deposition

C.1 Description of Polymerization Process

Plasma deposition is a coating technique that is attractive because it can produce highly cross–linked thin films. Possible applications for these thin films have been as dielectrics, optical coatings [120], semipermeable membranes (for gas separation and reverse osmosis) [121,122], moisture barriers [123,124] and as an intermediary to improve adhesion [125]. Plasma polymerization falls under the general heading of a glow discharge deposition process, the specialization will be made clearer a little further on. It is referred to as a thermally nonequilibrium deposition process because the ions, electrons and neutrals comprising the plasma are at very different energy levels (temperatures) than the roughly ambient temperature of the substrate. Formation of thin films via this process was recognized as early as the mid–1870's, but only recently, since the 1960's, has the advantages of such films been realized. Consequently, though the studies of the process are quantitative now, there is a great deal of room for improvement and innovation regarding processing techniques, predicting coating quality and properties, and reproducing results. The following represents a brief introduction into the area of plasma deposition. There are several good references surveying the field over the years [126]–[129] and a good in–depth analysis provided by Yasuda [83].

In plasma deposition, a chamber containing the substrate to be coated, is evacuated to roughly 35 μm Hg to remove gaseous impurities. A monomer vapor, forming

the basic building block for the resulting polymer, is then drawn through the chamber, accompanied by an increase in pressure. Either inductively or capacitively, an electric field is produced within the evacuated chamber at a high enough field strength such that fragmentation and ionization of the monomer occurs. This process causes the formation of free radicals and ionized material. Diffusion of both to the substrate results in the formation of a thin film. This is accompanied by a drop in operating pressure as some of the vapor phase is being converted to solid phase. The thin film is actually formed via two different reaction paths as outline by Fig. C.1. One mechanism is radiation induced polymerization – the fragmented monomer diffuses to the substrate and polymerizes after deposition, while in the solid phase. Another mechanism is glow discharge or plasma polymerization, whereby the fragments nucleate in the vapor phase, forming a polymer intermediate, and then diffuse to the substrate – undergoing further crosslinking with the rest of the film matrix. This second mechanism is responsible for the formation of microspheroids which are then embedded in the film matrix, yielding a biphasic structure [95]. By varying operating parameters such as monomer flow rate, pressure, power density and substrate temperature, powder (microspheroid) formation can be controlled [83, Section 8.4.3]. Scaling specific operating parameters from one process chamber to another is a difficult task. However, powder formation appears to be enhanced by operating conditions such that a high density plasma meets a high concentration of monomer in a localized space. It also has been shown that increasing the substrate temperature can reduce powder formation [96]. This dependence was attributed to thermal effects on: 1) chemical reaction rates, 2) desorption of loosely bound constituents, 3) heat transfer from the substrate to the gas phase and 4) surface mobility of the impinging species.

Due to these different film formation mechanisms, it is noteworthy that the end product is very different from a conventionally polymerized material. Plasma polymerized styrene does not have the same characteristics as polystyrene, although both start from the same monomer. An example of the fragmenting and resultant three dimensional, highly cross-linked structure of a plasma polymerized ethylene film is shown in Fig. C.2.

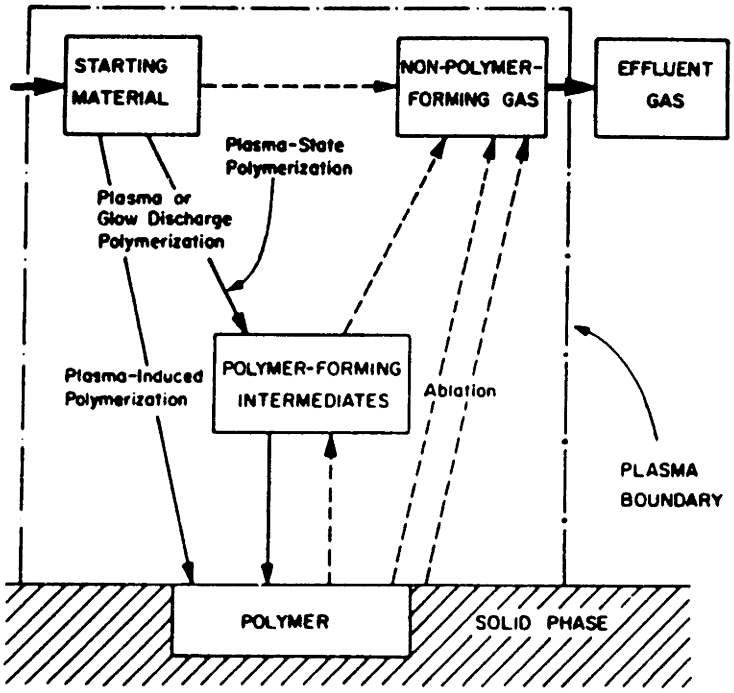


Figure C.1: Overall Mechanism of Glow Discharge Polymerization (from [83, p. 180]).

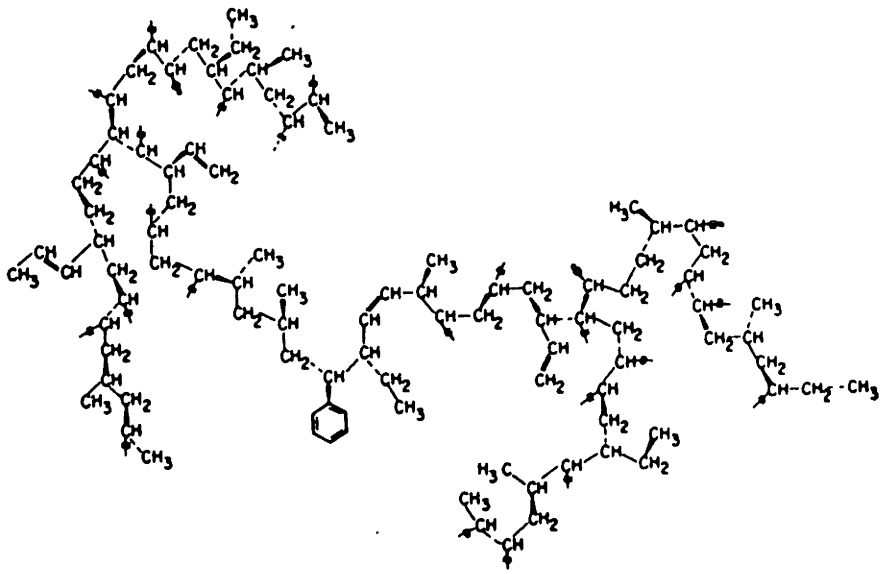


Figure C.2: Model Structure of a Plasma Polymer Film of Ethylene (from [83, p. 154]).

There are a several process parameters that are relevant. One major factor is the geometry of the vacuum chamber, either a bell-jar or, more likely nowadays, tubular. Included with this is the manner in which the discharge is produced, the location of the monomer and inert carrier gas (optional) inlet relative to the discharge and the location of the substrate within the chamber. Inductive coupling uses coils wrapped around the outside of the chamber. Capacitive coupling uses parallel plates placed inside the chamber. Other parameters are the flow rate of the monomer vapor and an optional inert carrier gas, background pressure before introduction of the monomer and operating pressure with the monomer flow. Varying the flow rate and pressure will tend to favor one of the two growth mechanisms described above. At a given operating power, films can vary from oily to rigid to powdery, as shown in Fig. C.3. This mapping will also change with power density and substrate temperature, as indicated above. The frequency, if AC, and power level of the discharge are also critical. The majority of the discharges are RF and performed at the FCC regulated frequency of 13.56 MHz. If the power level is too high, the competing mechanism of film ablation may result in etching of the substrate instead of film growth. Yasuda introduces the composite parameter W/FM which combines the power level (W) in watts, monomer flow rate (F) in sccm and monomer molecular weight (M) in grams/mole. It is usually expressed in J/kg by multiplying the former result by $1.34x10^9$. Although this is still a system dependent parameter (chamber geometry) it shows that there is some relation between changes in flow rate and power and allows some crosschecking with various monomers.

These parameters affect the deposition rate, quality, uniformity and electrical, mechanical and physicochemical properties of the film. A good summary of the empirical relationship between these parameters and the film produced can be found in Jensen, et al. [130] for polymerization of ethane, Wróbel, et al. for organosilicones [131,132] and Yasuda [83, Chapter 9] for a wide variety of monomers. An obvious parameter not mentioned previously is the monomer used. Generally, due to the fragmentation of the monomer and the atomic nature of the plasma polymerization, the resulting films are categorized more by the family in which the monomer is part of than the specific monomer. The various families are hydrocarbons, nitriles, organosilicones, fluorocar-

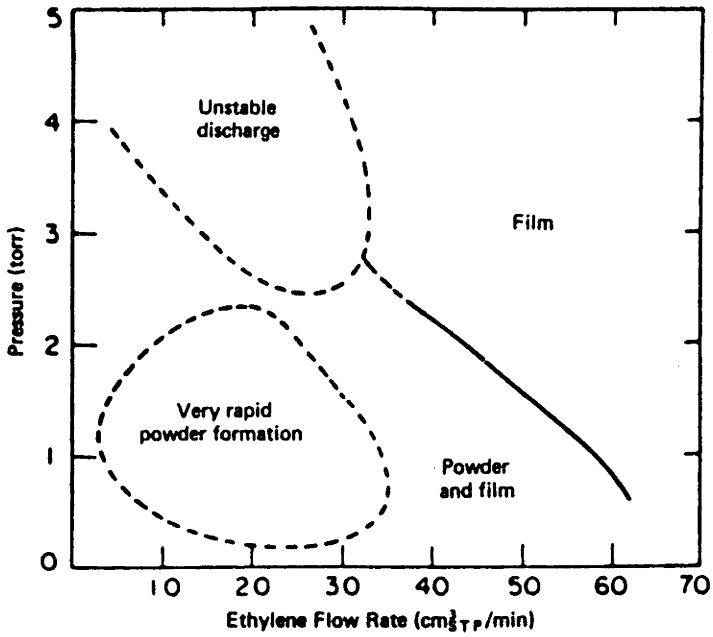


Figure C.3: Powder and Film Regions at 100 W using Ethylene Monomer (from [83, p. 244]).

bons, silicones, organometallics and oxygen containing compounds. Hydrocarbons can be further broken down into the various aromatics, chained, saturated, double and triple bonded compounds. Almost any vapor, whether it is the natural phase of the compound under STP or is pulled off of the liquid or solid phase, can be deposited. For example, plasma polymerized films of methane have been formed and have exhibited good adhesive qualities [125].

Various analytical tools are used to study the structure of these thin films. Elemental analysis is performed using NMR and mass spectrometry. Structural information on bonding within the film is obtained using electron spectroscopy for chemical analysis (ESCA), X–ray photoelectron spectroscopy (XPS), IR and FTIR spectroscopy. A technique called electron spin resonance (ESR) is used to determine the concentration of free radicals [83, Section 6.4.2]. Free radicals are negatively charged subgroups, having an unpaired electron, that form due to the fragmentation of the monomer [133]. They serve an important function in the growth of both conventional polymers and plasma polymers. Many of the free radicals participate in the growth reactions of

film formation but there may be a substantial concentration of free radicals trapped within the film after the deposition process. The quantity of trapped free radicals is a strong function of the chemical structure of the monomer. Once trapped in the film, these free radicals can affect the properties of the film such as electrical conduction and permeability. The radicals are usually difficult to remove or eliminate but their concentration can change slowly over time, contributing to the instability of the film properties.

C.2 Experimental Apparatus

The electrical configuration of the apparatus is sketched in Fig. C.4. The capacitively coupled RF system is made up of an RF generator, impedance matching tuner, voltage multiplying coil and parallel plate electrodes. The RF generator (ENI Power Systems, Inc., Model ACG-5) is capable of supplying up to 500W and is preset to operate at 13.56MHz. An impedance matching tuner (MFJ Enterprises, Inc., MFJ Versa Tuner V, 3kW series, model MFJ 989) was used to tune the resonant circuit of the electrode–coil system to 13.56MHz and thus cancel the imaginary contributions to the load impedance of this system and the plasma. The unit needed to be tuned only once and was performed under typical operating conditions. Once tuned (SWR as close to unity as possible), the load presented to the generator was a pure 50Ω, through proper design of the voltage multiplying coil and plate electrodes. The voltage multiplying coil was constructed from 1/4" copper tubing wound into a 13 turn (n) air core coil with a coil radius (a) of 1" and extending a length (b) of 5.5". The ends of the coil are connected to the electrodes, the coil is grounded at its midpoint and the RF power is tapped in approximately 1 1/4 turns from the ground contact. Thus the turns ratio yields a voltage amplification of 10.4. The electrodes are 8"x12" (A) aluminum plates spaced (d) 2.5" apart. Using

$$C = \frac{A\epsilon_o}{d}$$
$$L(\mu H) = \frac{a^2 n^2}{9a + 10b} \quad (C.1)$$

gives an inductance of 2.64 μH in parallel with a capacitance of 8.64 pF [134]. The resulting resonant frequency would be 33.3 MHz, emphasizing the need for the impedance tuner. This capacitance is not quite correct because there is a 8"x14"x1/16" aluminum plate located midway between the electrodes and grounded. It is on this plate that the substrates to be coated were mounted.

The mechanical configuration of the apparatus is shown in Fig. C.5. The main vacuum chamber was a 9"x6" Corning pyrex glass reducing tee. The chamber was sealed with 3/4" thick aluminum plates and teflon gaskets. All the aluminum plates were grounded. Vacuum pressure was monitored with an MKS vacuum gauge with digital readout (222AHS–MKS absolute vacuum gauge, PDR–C–1B MKS power supply) screwed into the plate that sealed the 6" diameter tee section. It is important to choose a pressure gauge whose operation would be unaffected by the RF fields. Vacuums down to 35 μm Hg were obtained using a Welch Duoseal (model 1397) mechanical roughing pump connected to the chamber through a cold trap. A butterfly valve at the bottom of the chamber allowed faster evacuation while the two Whitey valves allowed coarse and fine control of the pressure during deposition. The monomer liquid was placed in a flask immersed in an ice bath to prevent condensation of the vapors in the tubing. Monomer flow rate was controlled by a Whitey valve at the upper inlet to the chamber. Diffuser boxes at the inlet and outlet were used to provide an evenly distributed monomer flow between the electrodes and to help prevent sneakage around the electrodes. The entire device was enclosed by a wire mesh when the RF was on. This was both for safety reasons and to prevent noise on the radio bands.

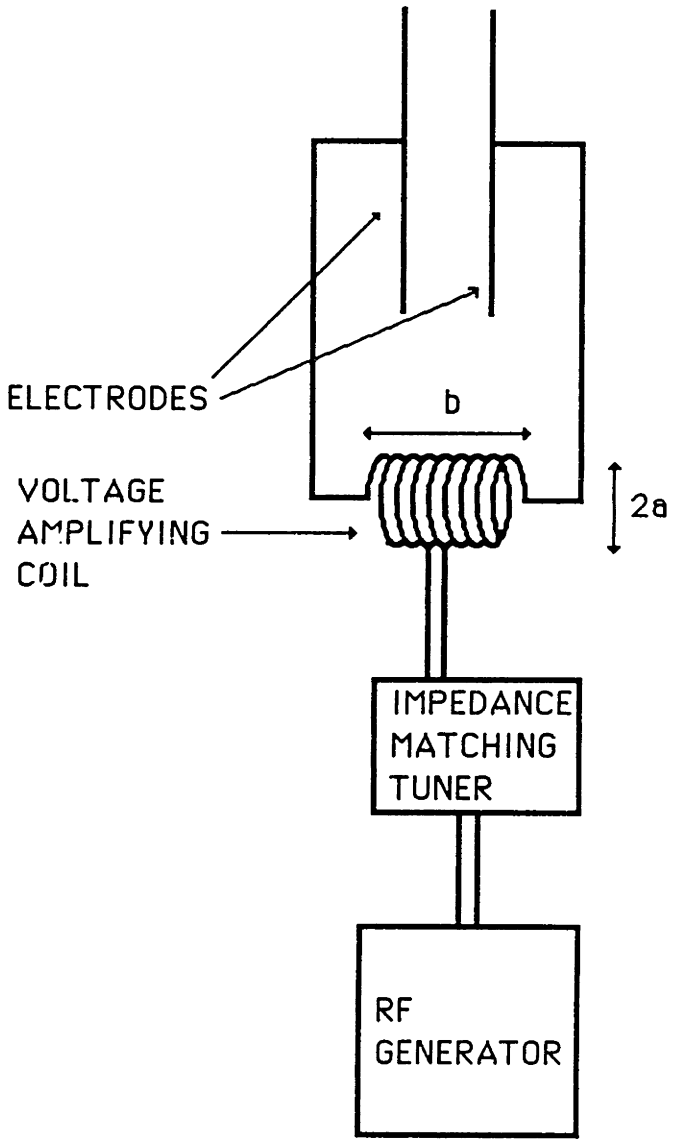


Figure C.4: Electrical Configuration of 13.56 MHz Capacitively Coupled RF Plasma Deposition Apparatus.

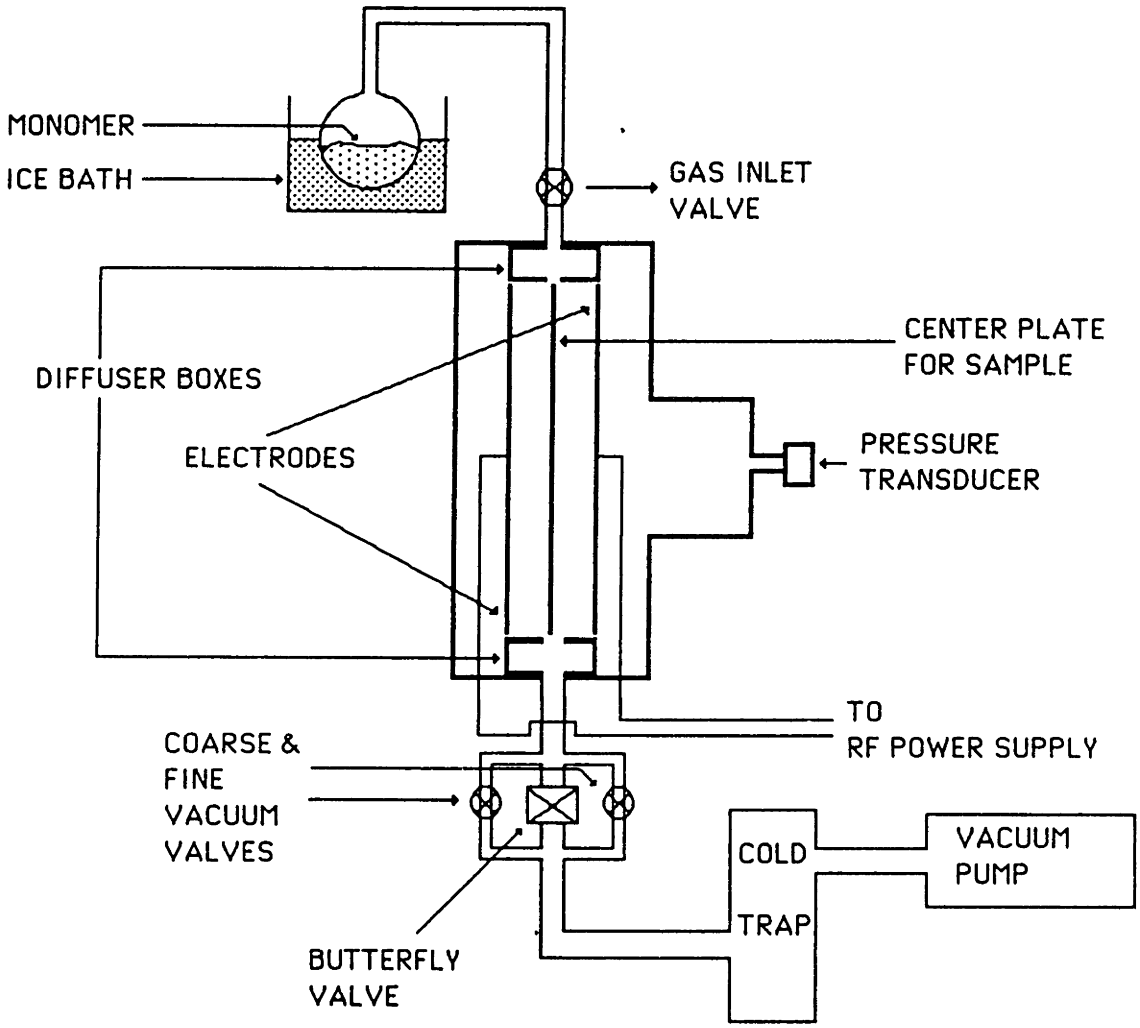


Figure C.5: Mechanical Configuration of Plasma Deposition Apparatus.

C.3 Operational Procedure

Generally, only the upper aluminum plate was unsealed for placement and removal of substrates, and cleaning of the chamber. The bottom and side plates were unsealed and cleaned very infrequently.

1. If the vacuum chamber, electrodes or aluminum plate were coated from a previous deposition then clean them with acetone. For an accumulation of coating residues a stronger solvent, methyl chloride was more effective (but required safety precautions for handling). A thin buildup did not seem to adversely affect the process.

2. Using Scotch tape, affix the substrates to be coated onto the aluminum plate. Generally, the chips and witness strips were located 4" downstream from the diffuser box at the inlet, 2.5" below the top of the electrodes, along a horizontal band 3" wide centered about the middle of the 8" wide plate.

3. Slide the aluminum plate into place between the electrodes in the chamber.

4. Check to ensure that the electrodes are isolated from ground.

5. Using a high vacuum grease with low vapor pressure, such as Cello-Grease^{TM}, liberally grease the upper lip of the chamber, both sides of the gasket and the aluminum plate and reassemble. Hand tighten the flange bolts.

6. Make sure all the valves are closed, including the butterfly valve. Fill the cold trap with liquid nitrogen and start the vacuum pump. After allowing the tubing leading to the chamber to be evacuated, open the butterfly valve to evacuate the chamber.

7. Evacuate down to 35–50 μm Hg. Check the flange bolts and retighten if necessary. It is important to get a good seal to prevent contamination by atmospheric gases during deposition. Check the leak rate by closing the butterfly valve and monitoring the rise in pressure. It should be less than 2–3 μm Hg per minute.

Due to the large volume of the chamber, this took several hours to evacuate and to form a good seal.

8. When the chamber is sufficiently well sealed, place the flask containing the monomer in an ice bath. Slowly open the inlet valve, allowing the monomer to degas, the tubing to evacuate and the pressure to drop back down to its prior low level. If microchips are being coated, their high frequency gain in air, used in computing the normalized capacitance, should be recorded now. Unfortunately, the monomer flow rate usually was not monitored. Instead, a rough estimate was made at this point by closing the butterfly valve and monitoring the initial pressure rise. At a fixed volume and temperature, the system pressure is proportional to the number of gas molecules present and thus the change in pressure is proportional to the change in the number of molecules. Therefore the volume flow rate in torr–liter/sec equals the chamber volume times the initial rate of pressure rise [135, p. 16]. Assuming the monomer flows mainly through the straight section (radius of 4.5", length of 18") yields a volume of 18.8 L. A typical pressure rise was 2.7 mT/sec giving a volume flow rate of 0.05 tL/s which is 4 standard ccm.

9. Open the coarse control valve at the bottom fully, partially open the fine control valve and close the butterfly valve. By adjusting these control valves and the inlet flow valve, let the chamber pressure equilibrate at roughly 200 μm Hg. Operating conditions are a function of the monomer used. The flow rate must be high and the pressure low to obtain rigid films (as can be seen from Fig. C.3).

10. Switch on the AC power for the RF generator and increase the power level. By flipping the power meter switch the forward and reverse power levels can be checked. If the tuning is off there will be significant reverse power reflected back from the load. The plasma glow will start slowly as the ionization process begins to buildup. This is accompanied by a decrease in a pressure as the monomer vapor starts nucleating, polymerizing and depositing. Usually a drop of 100 μm Hg from pre–discharge occurred in normal operation. When coating at 25

watts it was necessary to begin at 50 watts to 'get the plasma started' and then turn the power back down to 25 watts after a minute or two. The color of the discharge with bromobenzene was quite purplish. This is in contrast to discharges with air flowing through, which were more pink colored.

11. Leave the power on and coat for as long as desired. During first 1/2 hr. of coating the power may rise 5 watts as the discharge establishes itself – reset the power level if necessary. Check to make sure there is enough liquid nitrogen in the cold trap and ice in the ice bath. When there is too little of either coolant the pressure will start rising again. If this occurs turn the RF power off and replenish the appropriate coolants, leaving everything else operational. Then restart the discharge.

12. When finished coating turn off the RF generator. Close both pressure control valves and the monomer inlet valve. Seal off the monomer flask to prevent any leaking of vapors and disconnect from the vacuum chamber inlet.

13. If post–treatment is desired, i.e. letting the coated samples remain in the monomer vapor overnight, then leave the chamber sealed overnight. With a good seal, the chamber should not rise above 4 mm Hg overnight. Close the valve connecting the pressure gauge to the chamber as there is no need to expose the pressure sensor to possible degradation through film formation. The next day go to step 14. Due to the nature of a highly ionized and reactive surface upon removal of the discharge, introduction of oxygen immediately afterwards may lead to oxidation of the surface, resulting in unwanted surface properties regarding wettability.

14. If no post–treatment is desired then open the butterfly valve and re–evacuate the chamber for 1/2 hr. to remove the monomer vapor. Close the butterfly valve, turn off the vacuum pump and slowly open the monomer inlet valve and allow chamber to reach ambient pressure. Open the chamber up and vent the chamber through an exhaust hood via ducting to remove any remaining vapors.

C.4 Experimental Observations For Coating Microchips

An initial concern with the deposition process was the danger of damaging the FET connected to the floating gate when the plasma was on due to charge accumulating on the electrodes. In fact, it was thought that the first several attempts were unsuccessful because of this effect. It turned out that all but one of these coated chips worked fine eventually, just not immediately after turning off the RF power. The leads of the microchips were always left floating (electrically), without any ill effects. The only chip actually damaged irreversibly was one coated immediately after the impedance matching tuner was used for the first time and an unnecessarily high power level of 450 watts was applied. Most chips were not functional immediately after removal of the discharge due to the large DC offset values associated with trapped charge within the coating. These charges seemed to remain indefinitely while the chamber was evacuated. However, upon exposure to air the charge was able to leak off, leaving the coated chips usable. Two examples of this are shown in Figs C.6 and C.7 for chips Z13 ($\sim$ 1.2μm)and Z22 (3.5 μm). For Z13, room air was introduced into the evacuated chamber immediately after the deposition process. However, Z22 remained under vacuum for 3 days before the introduction of room air into the chamber. As the offset decreases to a reasonable level the response goes to a low gain and zero phase – just what is expected in the purely capacitively coupled regime. Note that both polarities of offset were obtained. A glaring exception to this behaviour is the tracking of the deposition process for Z4 (see Section 5.2.1). These measurements were made with the RF off but the chamber was still evacuated. It is unclear as to the reason why the offset was not a problem here.

An interesting phenomenom was observed before the realization that the coated chips were undamaged and usable. In an attempt to reduce the charge reaching the microchips, coatings were deposited on witness strips shielded with a metal mesh. The thought was to allow air flow in a zero field region above the substrate. The result, however, was the deposition of a powder instead of a rigid film. Presumably, based

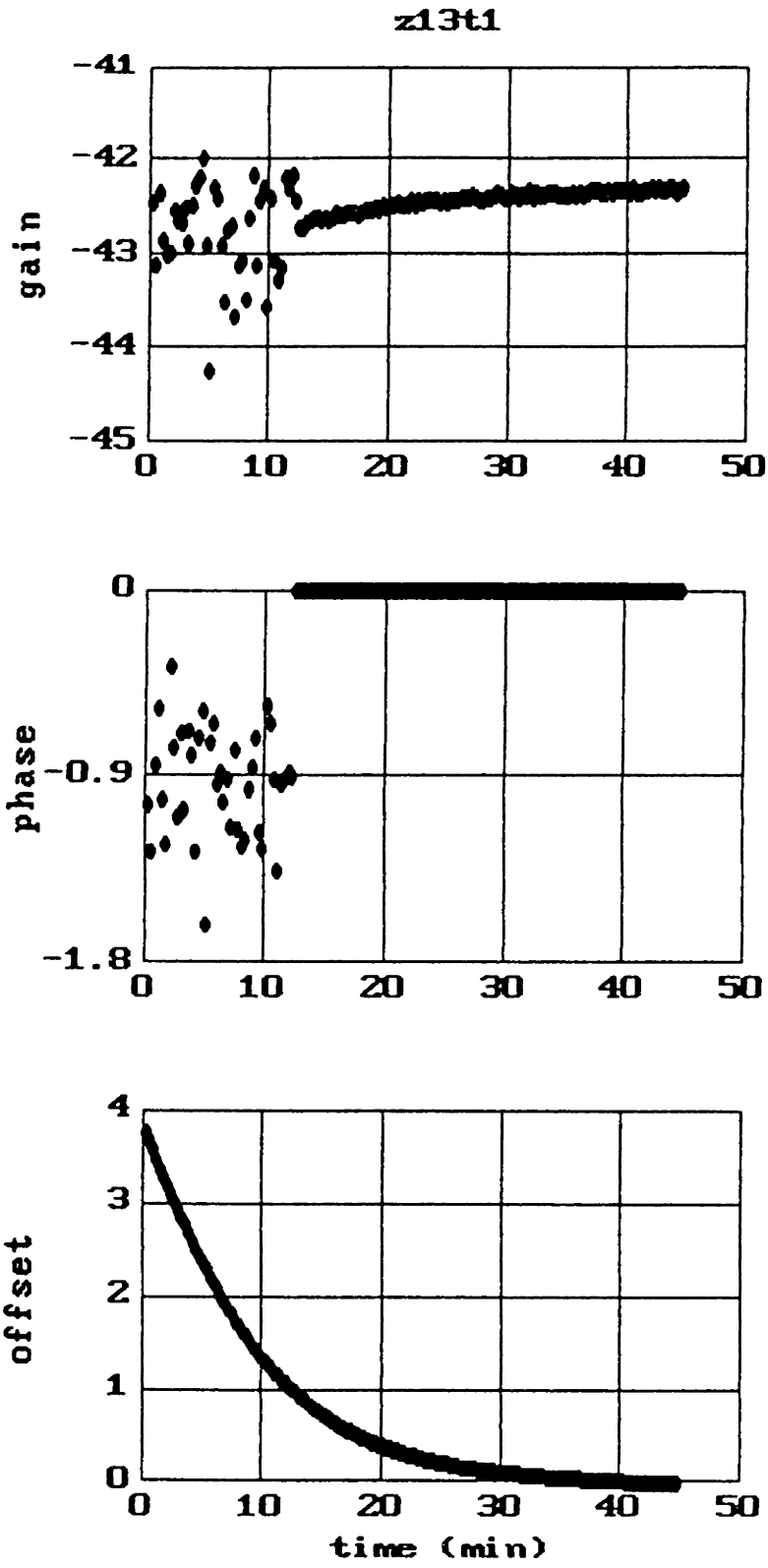


Figure C.6: Response at 1 kHz of Z13, a $\sim$ 1 μm Thick Bromobenzene Coated Microchip, as Room Air was Introduced into the Evacuation Chamber Immediately After Deposition.

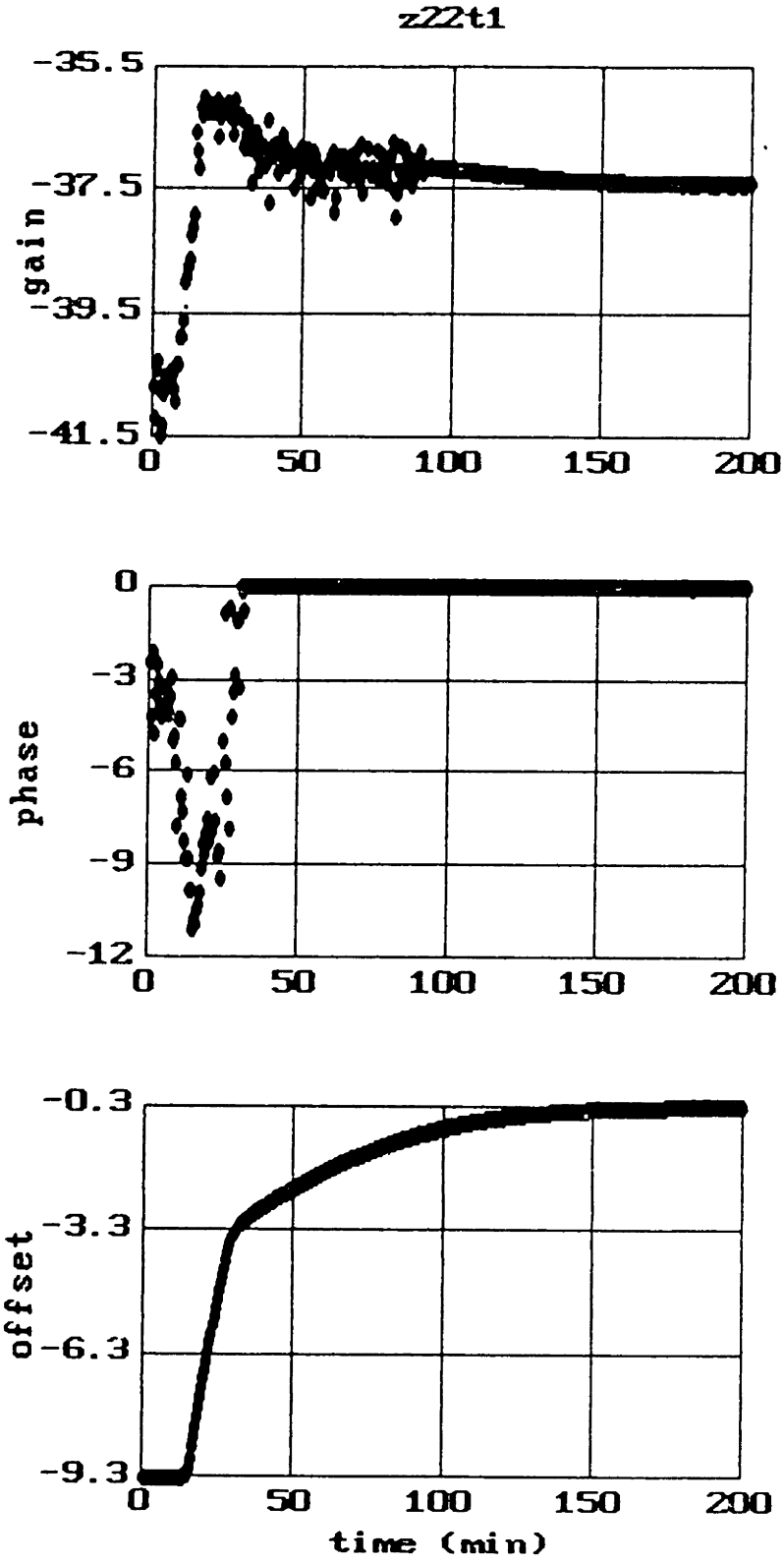


Figure C.7: Response at 10 kHz of Z22, a 3.5 μm Thick Bromobenzene Coated Microchip, as Room Air was Introduced into the Evacuated Chamber.

on Fig. C.3, this was due to a very low flow rate underneath the mesh. In any event, powder formation was always seen on the lower 1/3 of the center aluminum plate. This gave an indication of the uneven distribution of deposition throughout the chamber.

Another issue was the adhesion of the coatings. After many months of testing coated chips in transformer oil with temperature cycling and testing in vacuum ovens at temperatures up to 115°C there appeared to be no problem with adhesion to the electrodes. However, the coating did not adhere well to the smooth Kapton packaging that encapsulated the microchips.

Appendix D

Experiments With Doped Transformer Oil

In electrochemistry, the charge separation layer at an interface is generally described in terms of a lumped parameter impedance (perhaps frequency dependent) and is associated with polarization occurring at the electrodes [136, p. 345]. Using the microchip, an attempt can be made to describe the double layer using more fundamental, physical parameters. As an example of this, a sample of transformer oil was doped with ASA–3, a Shell antistatic agent [137]. The conductivity of the oil could be altered over several orders of magnitude by varying the doping concentration. This change, in turn, greatly affects the diffuse double layer thickness (λ_D) according to

$$\lambda_D = \sqrt{\frac{\epsilon D}{\sigma}} \tag{D.1}$$

where D is the molecular diffusivity of the ions in the fluid. At a high doping concentration, the concentration of ionizable species in the oil greatly increases, resulting in a large increase in conductivity. This results in a thinning down of the double layer thickness. These type of experiments were first performed by Mouayad [73, Section 5.3.2] and were repeated here. The frequency response of an uncoated microchip in a highly doped oil sample is shown in Fig. D.1. The relaxation observed is due to the highly conducting oil. However, notice that the low frequency gain does not go to unity, but levels out and then starts proceeding upwards again. This is accompanied

by another downturn in phase at the lowest frequencies. Another relaxation is occurring here, having a phase peak below the frequency range of the microdielectrometer. This relaxation is attributed to polarization occurring at the electrodes and, as a first attempt, is modeled using a thin, insulating layer adjacent to the electrodes, underneath the surrounding oil. First, the properties of the oil are estimated with data at 80 Hz, using a uniform medium model. The estimated oil permittivity and conductivity are $1.94x10^{-11}$ F/m and $1.80x10^{-9}$ S/m, respectively. Second, the double layer thickness and permittivity are estimated using the two parameter estimation technique described in Section 4.4. Here, the two datapoints are the high frequency gain using the oil permittivity, and the low frequency gain at 0.015 Hz, where the plateau occurs, using a very large permittivity to simulate a perfectly conducting surrounding medium. This gives a double layer thickness of 484 Å and permittivity of $1.04x10^{-11}$ F/m. Now, using the estimated bulk properties of the oil and the estimated layer thickness, the bulk properties of the double layer are estimated using the data at 0.005 Hz, where there is evidence of the second, lower frequency relaxation. The estimated double layer permittivity and conductivity are $1.04x10^{-11}$ F/m and $1.34x10^{-13}$ S/m, respectively. Note that the re-estimated bulk permittivity of the double layer agrees with the previous estimate obtained from different frequency datapoints. The resulting predicted frequency response matches the data very well (Fig. D.1).

The effect on the frequency response of stirring the oil was minimal in this case, indicating a thin double layer lying within the boundary layer created by the oil flow. The oil flow was on the order of 5 cm/sec. The double layer thickness can be computed using (D.1) and the Einstein relation relating diffusivity and mobility,

$$\frac{D}{b} = \frac{kT}{q} \tag{D.2}$$

where the thermal voltage (kT/q) is 25 mV, and b is the ion mobility. In hydrocarbon liquids, b is taken to be $(2x10^{-11}/\eta)$ $m^2/V\text{-}s$ where η is the absolute viscosity in MKS units [75, p. 5.4]. For transformer oil at room temperature, the viscosity was measured to be 0.02 kg/m-sec. This gives a double layer thickness of 5,190 Å. The estimated double layer thickness is an order of magnitude less than computed from the

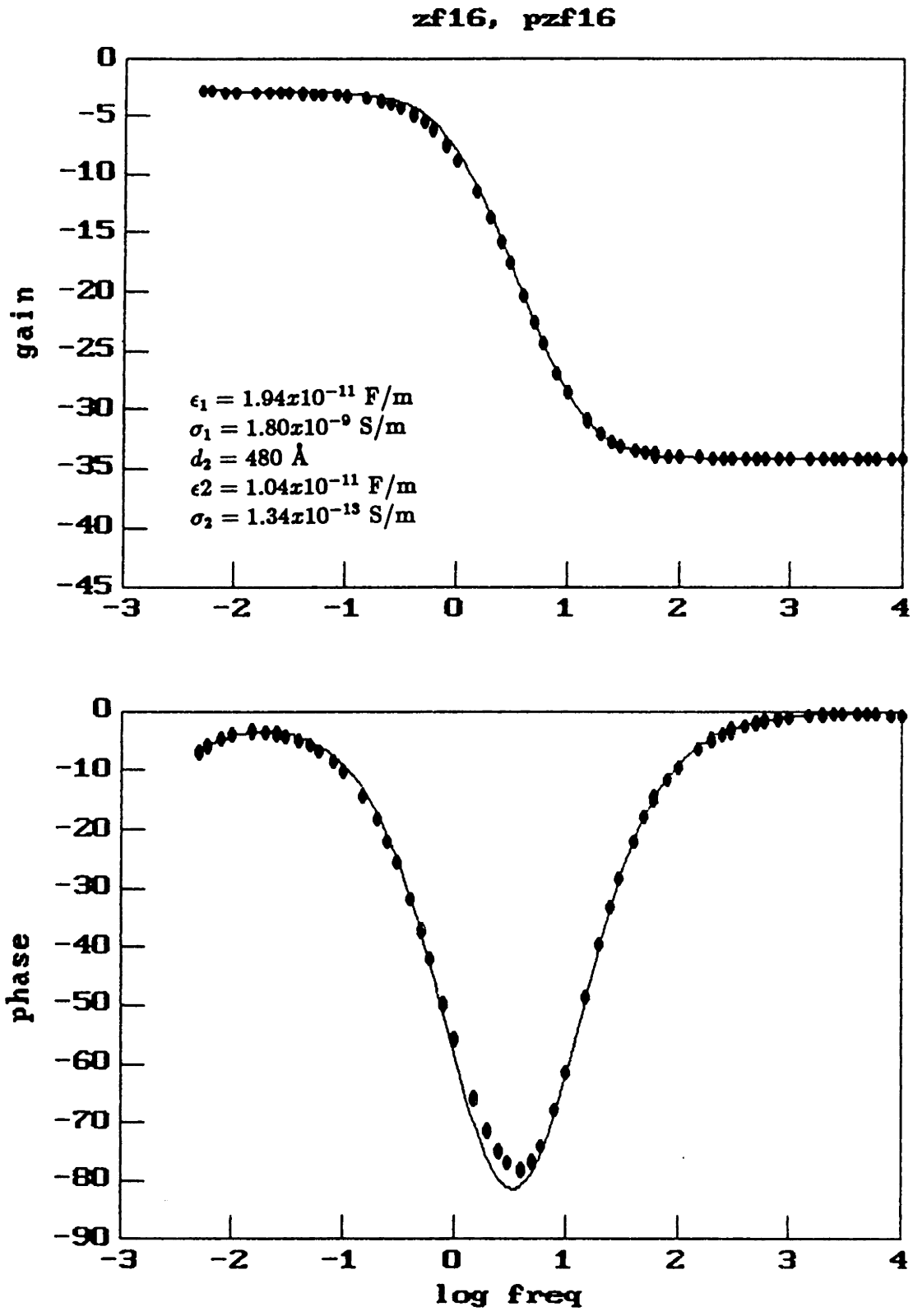


Figure D.1: Measured (•) and Predicted (–) Response of an Uncoated Microchip in Oil Highly Doped with ASA.

standard balancing of migration and diffusion currents. An important consideration is the electric field strength within the double layer relative to the applied AC electric field strength. The AC electric field strength is roughly 1 V divided by 12.5 μm, giving a field strength of $8x10^4$ V/m. However, at the low frequencies the gain was roughly –2.5 dB, implying a voltage difference of 0.25 V between the electrodes and a field reduction by a factor of 4. The double layer field can be approximated by dividing the thermal voltage by the layer thickness, giving a field strength of $5x10^5$ V/m using the estimated thickness and a field strength of $5x10^4$ V/m using the computed value. Perhaps the discrepancy in estimated and computed layer thicknesses can be explained by the following scenario. Given the relative field strengths, the applied AC field is strong enough to disrupt the charges further out in the diffuse region but only perturbs the equilibrium of the charges closer in. This might account for the much thinner 'average' layer thickness. Part of the problem also might be in trying to describe the space charge region in terms of a complex bulk permittivity instead of more fundamental parameters such as charge carrier concentrations.

The response of an uncoated chip in oil having a lower concentration of dopant is shown for both the stirred and unstirred case in Fig. D.2. Unlike the previous example, there is a large effect produced by the stirring of the fluid. The stirred response looks very much like it could be modeled just using the complex bulk permittivity of the oil (after correcting for the large, positive DC offset that is responsible for the upward shift of the gain curve). However, the unstirred response displays overlapping relaxations, making it is impossible to determine the complex bulk permittivities and double layer thickness here via a sequence of single and two parameter estimation schemes, if indeed such a model is relevant in the first place. Perhaps using a multi-parameter estimation routine based on frequency measurements could produce these parameters. The apparent decrease in the bulk conductivity of the oil is due to the decreased dopant concentration. Using the Micromet estimated value for ϵ'' at 0.15 Hz for the stirred experiment yields an oil conductivity of $8.01x10^{-13}$ S/m, significantly more insulating than in the highly doped case. According to (D.1), the double layer thickness here is 20 μm, much larger than in the previous case. Therefore, it is not

surprising that the response here is susceptible to the effects of stirring, as observed.

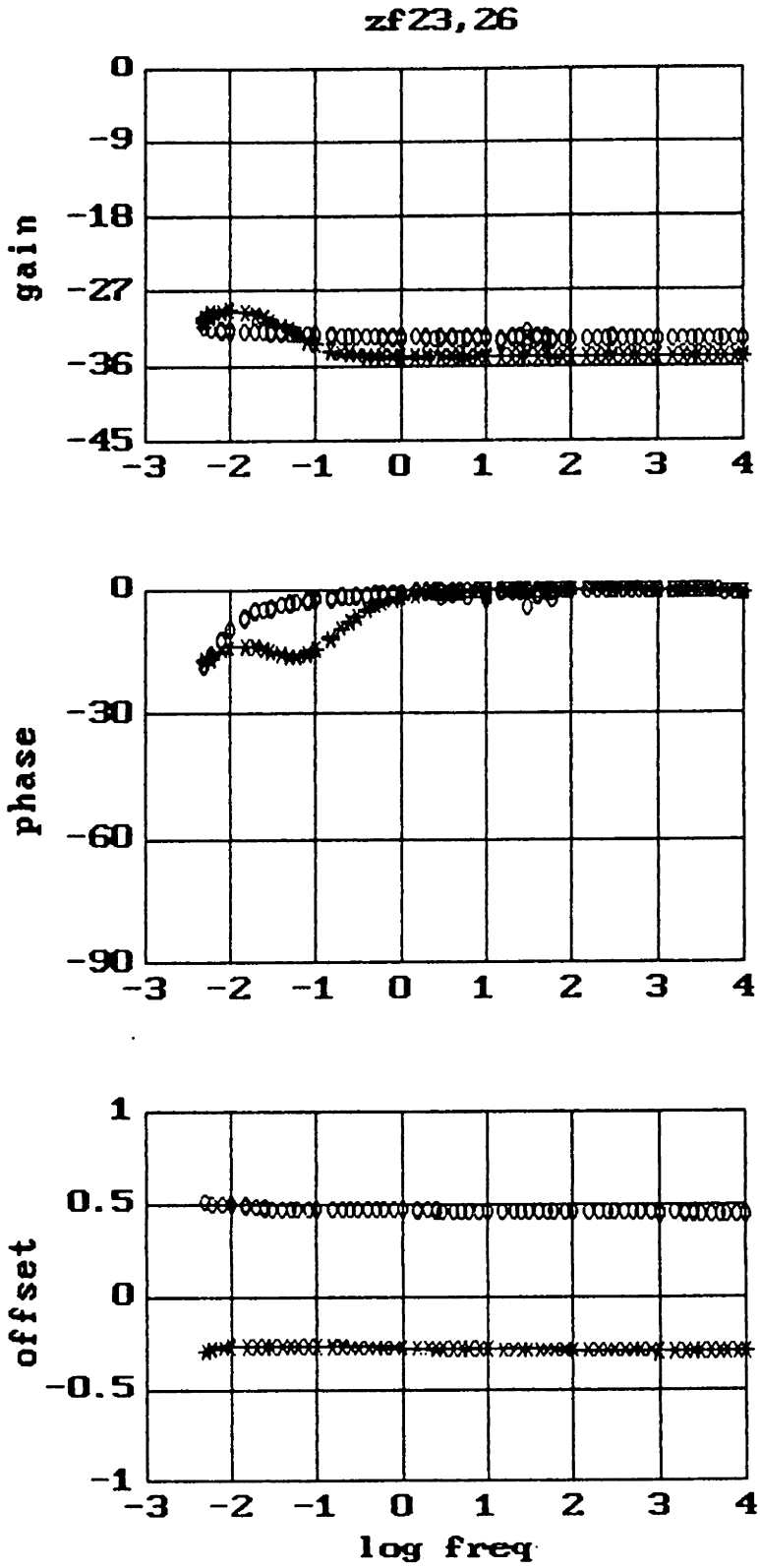


Figure D.2: Measured Response of an Uncoated Microchip in Oil Slightly Doped with ASA. (o) – Stirred, (*) – Unstirred).

Appendix E

Data Acquisition and Processing Routines

Data Acquisition

Main Routines

WMOD.C – complete water module going from time scan information to moisture contents and residuals.

TSCAN.C – gets raw data with a time scan at a selected frequency .

FCHIP5.C – performs frequency scan, used to generate raw data for master curve.

Auxiliary Subprogram

NEWFUNCT.C – contains microdielectrometer subroutines, issues commands to microdielectrometer and gets data back.

Auxiliary Files

MMET.H – specifies device port for microdielectrometer.

HEADER – contains the heading that goes at top of data files.

CHIPIDS – contains the chipid for each chip hooked up.

MCFP.PAR – contains relation between log(fp) and log(mc) for each miosture chip.

MCT.PAR – contains relation between log(mc) and 1/T for transformer.

Data Stripping

PLTFILE.C – strips raw data from frequency scans produced by FCHIP5.C into format suitable for XENIX plotting routine.

TPLTACM.C – strips raw data from time scans produced by WMOD.C in .acm files into format suitable for XENIX plotting routine.

TPLTHST.C – strips processed data from time scans produced by WMOD.C in .hst files into format suitable for XENIX plotting routine.

MSC.F – computes master curve spline function from raw data.

FPCALC.F – computes moisture content and residual from. latest reading.

SPLINE.F – Computes spline function from freq. response.

SPLINT.F – Performs spline interpolation to find fp.

BRENT.F – Finds function minimum (fp).

Table E.1: Directory of Sample Input Files and Program Listings

	Page Number	
Program	Sample Input File	Listing
WMOD.C	357	363
TSCAN.C	358	386
FCHIP5.C	359	392
NEWFUNCT.C	–	399
MMET.H	–	403
HEADER	–	403
CHIPIDS	–	403
MCFP.PAR	–	403
MCT.PAR	–	403
PLTFILE.C	–	404
TPLTACM.C	–	408
TPLTHST.C	–	412
MSC.F	361	416
FPCALC.F	362	419
SPLINE.F	–	422
SPLINT.F	–	423
BRENT.F	–	424

WMOD.C

This program is a complete water module, executing TSCAN.C and converting raw data to fp, moisture content and generating a residual in one routine. Requires NEWFUNCT.C, MMET.H, STDIO.H, SGTTY.H, MATH.H, TIME.H and STRING.H to compile.

Input File (INWMOD)

input number of channels to be monitored <CR>

loop : input channel # and freq. to monitor <CR>

input sampling interval (sec) <CR>

Sample Input File

2

2 200.0

4 1000.0

300

Run in background with command: wmod <inwmod> o & <CR>

WMOD.C looks for: 1) file HEADER to get heading for data files, 2) file CHIPIDS to get the proper chipid, 3) CHX.MC to get spline information for channel x, 4) file MCFP.PAR for parameters describing relationship between log(mc) and log(fp) for each channel having a moisture chip, and 5) file MCT.PAR for parameters describing relationship between log(mc) and 1/T for the transformer. WMOD.C opens 4 files per each channel monitored per day. These filenames are formed by concatenating twmmdd- (obtained from timestamp) with the channel # and either .acm, .rdg, .chr or .hst. The .acm file holds all the raw data for the scan, the .rdg file contains the raw data for the most recent reading. The .hst file holds some raw data and all the converted data, the .chr file contains the most recent entry to the .hst file. WMOD.C runs forever, opening a new file at the beginning of each day.

TSCAN.C

This program monitors several chips, each at a specified frequency, sampling at a regular interval for a given period of time. Requires NEWFUNCT.C, MMET.H, STDIO.H, SGTTY.H, TIME.H and STRING.H to compile.

Input File (INTSCAN)

input number of channels to be monitored <CR>

loop : input channel # and freq. to monitor <CR>

input generic filename (twmmdd–#) <CR>

input heading followed by ! (for operating conditions) <CR>

input sampling interval (sec) <CR>

Sample Input File

```
2
2 200.0
4 1000.0
tw0616–1
voltage is on, cycling xformer through loads
begin at 8:55AM – MZ
!
300
```

Run in background with command: tscan <intscan> o & <CR>

TSCAN.C also looks for a file CHIPIDS to get the proper chipid TSCAN.C opens 2 files per each channel monitored. These filenames are formed by concatenating twmmdd–# (as input) with the channel # and either .acm or .rdg. The .acm file holds all the raw data for the scan, the .rdg file contains only the raw data for the most recent reading. The .rdg file is in the proper form to be read by FPCALC.F routine which convert from raw data to fp and moisture content.

FCHIP5.C

This program runs a specified frequency scan for any amount of chips. Requires NEW-FUNCT.C, MMET.H, STDIO.H, SGTTY.H, TIME.H and STRING.H to compile.

Input File (INFCHIP5)

input number of channels to be scanned <CR>

loop : input each channel # <CR>

input heading followed by ! (for operating conditions) <CR>

input minimum frequency <CR>

input maximum frequency <CR>

input # of measurements at each freq. <CR>

ascending or descending (for now make always ascending) <CR>

if more than 1 reading at each freq. then do many sweeps (o) or one sweep (a) <CR>

Sample Input File

2

1

2

voltage is on, cycling xformer through loads

begin at 8:55AM - MZ

!

1

10000

1

a

Run in background with command: fchip5 <infchip5> o & <CR>

Note – to generate data for master curve spline function computation the scan should be done only once and done ascending

FCHIP5.C looks for a file called CHIPIDS to get the chipid #'s for each channel. FCHIP5.C opens 2 files per each channel scanned. These filenames are formed by concatenating fwmmdd-# (from timestamp) with the channel # and either .acm or .str. The .acm file holds all the raw data for the scan, the .str file contains the information for MSC.F, a routine that computes the master curve spline function.

MSC.F

This program computes a master curve spline function from a frequency scan. Requires BRENT.F, SPLINE.F, and SPLINT.F.

Inputs

input channel # <CR>

input generic filename (must be in form fwmmdd)! <CR>

Run with command: msc <CR>

MSC.F looks for the stripped data file as output from FCHIP.C. This file is a concatenation of fwmmdd–, the channel number and .str. This file contains only the frequency, gain and phase for one scan, with frequency ascending from low to high.

FPCALC.F

This converts raw data to fp. Requires SPLINT.F.

Inputs

input channel # <CR>

Run using command: fp <CR>

FPCALC.F reads most recent raw data from file TWCHX.RDG where X is the channel # and this file is the one produced from TSCAN.C. FPCALC.F also reads CHX.MC which is produced by MSC.F and contains the master curve spline function. FPCALC.F opens a .chr and .hst file, the .hst file containing the history of converted values, the .chr file containing the most recent entry to the .hst file.

```
/****************************************/
/*     wmod.c */
/*      6/23/87  */
/* */
/* scans several channels, each at */
/* an individual frequency */
/* records accumulated data in */
/* .acm file, present status in */
/* .rdg file, fp  & mc data */
/* in .chr and .hst files */
/* requires chx.mc for spline info.*/
/* mcfp.par for moisture info. */
/* mcT.par for xformer info. */
/* and header for file heading */
/* at midnight runs fchip5 and */
/* recalibrates master curve */
/****************************************/
#include <stdio.h>
#include <math.h>
#include <sgtty.h>
#include <time.h>
#include <string.h>
#include "mmet.h"
#include "newfunct.c"

float sign(a,b)
float a,b;
{
if (b>=0.0)
return(fabs(a));
else
return(-fabs(a));
}

int max(m,n)
int m,n;
{
if (m > n) return (m);
else  return (n);
}

float fmax(m,n)
float m,n;
{
if (m > n) return (m);
else  return (n);
}
```

```
float fmin(m,n)
float m,n;
{
if (m > n) return (n);
else  return (m);
}

/* computes fp from spline function */
/* of master curve and input gain-phase value */
/* reads chx.mc  */

float fpcalc(chan,x,g,y)
float g,y,x;
int chan;
{
float p[200],f[200],y2[200],gmid,p1,p2,tol,fest,fp;
extern float rtbis(),splint();
int N,i,kmid,klo,khi,k;
extern FILE *fopen();
FILE *fpd;
extern int fclose();
extern int fprintf();
char mcfl[20],dum[50],ext,done;

/* open .mc file and get spline information */

sprintf(ext,"ch%d.mc",chan);
strcpy(mcfl,ext,1);
fpd=fopen(mcfl,"r");
fscanf(fpd,"%f",&gmid);
fscanf(fpd,"%d",&N);
fscanf(fpd,"%d",&kmid);
for (i=1;i<=N;i++) {
fscanf(fpd,"%f %f %f",&f[i],&p[i],&y2[i]);
/* printf("%f %f %f\n",f[i],p[i],y2[i]);
*/ }
fclose(fpd);

/* determine on which side of phase peak we're on */
/* set bracketing interval from one end of scan */
/* to 10% over fp */

if (gmid<g) {
klo=1;
khi=kmid;
f[khi]=1.15;
}
else
```

```
{
klo=kmid;
khi=N;
f[klo]=.85;
}

p1=splint(f,p,y2,N,1.0)-y;
if (p1>=0.0) printf("problem - phase is lower than estimated peak\n");
/* printf("pmin=%f\n",p1);
p1=splint(f,p,y2,N,f[klo])-y;
p2=splint(f,p,y2,N,f[khi])-y;
printf("brackets for routine\n");
printf("klo=%d freq=%9.3f delta phase=%7.2f\n",klo,f[klo],p1);
printf("khi=%d freq=%9.3f delta phase=%7.2f\n",khi,f[khi],p2);
*/
/* use bisection method to find root */

tol=.01;
fest=rtbis(f[klo],f[khi],tol,y,f,p,y2,N);
if (fest>0.0)
fp=x/fest;
else
fp=x;
printf("fp=%9.3f Hz\n",fp);
return(fp);
}

/* bisection root finding algorithm */
/* from Numerical Recipes, p. 247 */
/* converted to C */

float rtbis(x1,x2,xacc,dc,f,p,y2,N)
float x1,x2,xacc,dc,p[200],f[200],y2[200];
int N;
{
int jmax,j;
float fmid,f1,dx,rtbis1,xmid,dx1;
jmax=40;
fmid=splint(f,p,y2,N,x2)-dc;
f1=splint(f,p,y2,N,x1)-dc;
if (f1*fmid>=0.0) return(0.0);
if (f1<0.0) {
rtbis1=x1;
dx=x2-x1;
}
else
{
rtbis1=x2;
```

```
dx=x1-x2;
}
for (j=1;j<=jmax;j++){
dx=dx*.5;
xmid=rtbis1+dx;
fmid=splint(f,p,y2,N,xmid)-dc;
if (fmid<0.0) rtbis1=xmid;
if ((fabs(dx)<xacc)||(fmid==0.0)) {
/* printf("iter=%d  fest=%f\n",j,rtbis1);
*/ return(rtbis1);
}
}
printf("too many bisections\n");
}

/*  spline interpolation routine */
/* from Numerical Recipes, p. 289 */
/* converted to C */

float splint(xa,ya,y2a,n,x)
float xa[200],ya[200],y2a[200],x;
int n;
{
float h,a,b,y;
int klo,khi,k;
klo=1;
khi=n;
while (khi-klo>1) {
k=(khi+klo)/2;
if (xa[k]>x)
khi=k;
else
klo=k;
}
h=xa[khi]-xa[klo];
if (h==0.0) {
printf("bad  xa input\n");
return(0.0);
}
a=(xa[khi]-x)/h;
b=(x-xa[klo])/h;
y=a*ya[klo]+b*ya[khi]+((a*a*a-a)*y2a[klo]+(b*b*b-b)*y2a[khi])*(h*h)/6;
return(y);
}

/* counts lines in a file between $'s */

countlines(fpa)
```

```
FILE *fpa;
{
int n;
char c;
n=0;
while((c=getc(fpa))!='$'){
;
}
c='i';
while((c=getc(fpa))!='$')
if (c=='\n') n++;
return(n-1);
}


/* spline computing function */
/* from Numerical Recipes, p. 88 */
/* converted to C */

spline(x,y,n,yp1,ypn,y2)
int n;
float x[200],y[200],y2[200],yp1,ypn;
{
int nmax,i,k;
float u[200],p,sig,qn,un;
nmax=200;
if (yp1>.99E30) {
y2[1]=0.0;
u[1]=0.0;
}
else
{
y2[1]=(-0.5);
u[1]=(3.0/(x[2]-x[1]))*((y[2]-y[1])/(x[2]-x[1])-yp1);
}
for (i=2;i<=(n-1);i++){
sig=(x[i]-x[i-1])/(x[i+1]-x[i-1]);
p=sig*y2[i-1]+2.0;
y2[i]=(sig-1.0)/p;
u[i]=(6.0*((y[i+1]-y[i])/(x[i+1]-x[i])-
                    (y[i]-y[i-1])/(x[i]-x[i-1]))/(x[i+1]-x[i-1])-
                     sig*u[i-1])/p;
}
if (ypn>.99E30) {
qn=0.0;
un=0.0;
}
else
```

```
{
qn=0.5;
un=(3.0/(x[n]-x[n-1]))*(ypn-(y[n]-y[n-1])/(x[n]-x[n-1]));
}
y2[n]=(un-qn*u[n-1])/(qn*y2[n-1]+1.0);
for (k=n-1;k>=1;k--)  y2[k]=y2[k]*y2[k+1]+u[k];
return;
}

/* minimization routine   */
/* from Numerical Recipes, p. 253 */
/* converted to C */

float brent(ax,bx,cx,tol,xmin,f,p,py2,N)
int N;
float ax,bx,cx,tol,*xmin,f[200],p[200],py2[200];
{
extern float splint(),fmin(),fmax(),sign();
int itmax,iter;
float cgold,zeps,a,b,u,v,w,x,e,d,fx,fv,fu,fw;
float xm,tol1,tol2,r,q,p1,etemp;
itmax=1000;
cgold=.3819660;
zeps=1.0E-10;
a=fmin(ax,cx);
b=fmax(ax,cx);
v=bx;
w=v;
x=v;
e=0.0;
fx=splint(f,p,py2,N,x);
printf("initial freq=%f\n",x);
printf("p=%f\n",fx);
fv=fx;
fw=fx;
for (iter=1;iter<=itmax;iter++){
xm=0.5*(a+b);
tol1=tol*fabs(x)+zeps;
tol2=2.0*tol1;
/* printf("fabs(x-xm)=%f\n",fabs(x-xm));
printf("tol2=%f halfdif=%f\n",tol2,.5*(b-a));
printf("toler=%f\n",tol2-0.5*(b-a));
*/ if (fabs(x-xm)<=(tol2-0.5*(b-a))) {
*xmin=x;
return(fx);
}
if (fabs(e)>tol1) {
r=(x-w)*(fx-fv);
```

```
q=(x-v)*(fx-fw);
p1=(x-v)*q-(x-w)*r;
q=2.0*(q-r);
if (q>0.0) p1=(-1.0)*p1;
q=fabs(q);
etemp=e;
e=d;
if ((fabs(p1)>=fabs(.5*q-etemp))||(p1<=q*(a-x))||(p1>=q*(b-x))) goto alpha;
d=p1/q;
u=x+d;
if (((u-a)<tol2)||((b-u)<tol2)) d=sign(tol1,xm-x);
goto beta;
}
  alpha:
if (x<=xm)
e=a-x;
else
e=b-x;
d=cgold*e;
  beta:
if (fabs(d)>=tol1)
u=x+d;
else
u=x+sign(tol1,d);
fu=splint(f,p,py2,N,u);
/* printf("iter=%d\n",iter);
printf("guess freq=%f\n",u);
printf("pmin=%f\n",fu);
*/ if (fu<=fx) {
if (u>=x)
a=x;
else
b=x;
v=w;
fv=fw;
w=x;
fw=fx;
x=u;
fx=fu;
}
else
{
if (u<x)
a=u;
else
b=u;
if ((fu<=fw)||(w==x)) {
v=w;
```

```
fv=fw;
w=u;
fw=fu;
}
if ((fu<=fv)||(v==x)||(v==w)) {
v=u;
fv=fu;
}
}
}
printf("brent exceed maximum iterations\n");
*xmin=x;
return(fx);
}

/* calculate the second derivatives for */
/* spline interpolation and fp for master curve */
/* important - data must be input from high to low freq. */
/* reads data from fwmmdd-x.str */
/* writes to chx.mc */

msc(chan)
int chan;
{
extern FILE *fopen();
extern int fclose();
extern int fflush();
extern int fprintf();
extern float brent(),splint();
extern struct tm *tmstr;
extern float id[10];
FILE *fpmc,*fpstr;
float g[200],p[200],f[200],py2[200],fmean,pmin,fp,tol;
float gy2[200],gmid;
int N,i,klo,khi,kmin,k;
char dfile[20],o1[20],ext,c;

/* make .str filename (fwmmdd) */

strcpy(dfile,"");
if ((*tmstr).tm_mon<10)
sprintf(ext,"fw0%d",(*tmstr).tm_mon+1);
else
sprintf(ext,"fw%d",(*tmstr).tm_mon+1);
strcat(dfile,ext,1);
if ((*tmstr).tm_mday<10)
sprintf(ext,"0%d-%d.str",(*tmstr).tm_mday,chan);
else
```

```
sprintf(ext,"%d-%d.str",(*tmstr).tm_mday,chan);
strcat(dfile,ext,1);
/* printf("fname for msc = %s\n",dfile); /

/* read in data, requires increasing order in x coordinate */

fpstr=fopen(dfile,"r");
N=countlines(fpstr);
fclose(fpstr);
printf("# of lines in %s= %d\n",dfile,N);
fpstr=fopen(dfile,"r");
while((c=getc(fpstr))!='$'){
;
}
for (i=1;i<=N;i++) {
fscanf(fpstr,"%f %f %f",&f[i],&g[i],&p[i]);
/* printf("%f %f %f\n",f[i],g[i],p[i]);
*/ }
fclose(fpstr);

/* calculate second derivatives for spline routine   */
/* use natural spline so set endpt. second derivatives to 0  */

spline(f,p,N,1E31,1E31,py2);
spline(f,g,N,1E31,1E31,gy2);

/* now find fp, where phase is at minimum */
/* using Brent's search routine */

/* first compute bracketing interval */

kmin=1;
pmin=p[1];
for (k=2;k<=N;k++){
if (p[k]<pmin){
pmin=p[k];
kmin=k;
}
}

klo=kmin-1;
khi=kmin+1;
printf("brackets for routine\n");
printf("klo=%d freq=%f\n",klo,f[klo]);
printf("khi=%d freq=%f\n",khi,f[khi]);

/* compute fp by searching for phase minimum */
/* and also determine gain at fp */
```

```
fmean=f[kmin];
tol=.001;
pmin=brent(f[klo],fmean,f[khi],tol,&fp,f,p,py2,N);
gmid=splint(f,g,gy2,N,fp);
printf("fp=%f gmid=%f pmin=%f\n",fp,gmid,pmin);

/* write to file phase, normalized freq and second derivs. */
/* open chx.mc  */

strcpy(o1,"ch");
sprintf(ext,"%d.mc",chan);
strcat(o1,ext,1);
fpmc=fopen(o1,"w");
fprintf(fpmc,"%6.2f\n",gmid);
fprintf(fpmc,"%d\n",N);
fprintf(fpmc,"%d\n",klo+1);
for (i=1;i<=N;i++) fprintf(fpmc,"%12.5f %7.2f %13.5f\n",f[i]/fp,p[i],py2[i]);
fclose(fpmc);
return;
}

/* performs frequency scan at midnight */

int fchip5()
{
extern FILE *fopen(),*fpo,*fpi;
extern int fclose(),port;
extern int fprintf();
extern struct tm *tmstr;
extern float id[10];
extern int nscan,q3[10],nchan,compute();
extern char buffer[80],ss[80];
FILE *fpd,*fpd1;
char c,q2,o1[20],ext,dfile[20];
float tem,fmin,fmax,qf[9],x,scale,f;
int j,k,l,m,n,rep,num,all,i,q1;

/* make new filename (fwmmdd) */

strcpy(dfile,"");
if ((*tmstr).tm_mon<10)
sprintf(ext,"fw0%d",(*tmstr).tm_mon+1);
else
sprintf(ext,"fw%d",(*tmstr).tm_mon+1);
strcat(dfile,ext,1);
if ((*tmstr).tm_mday<10)
sprintf(ext,"0%d",(*tmstr).tm_mday);
```

```
else
sprintf(ext,"%d",(*tmstr).tm_mday);
strcat(dfile,ext,1);
/* printf("fname = %s\n",dfile); */

for (i=1;i<=nscan;i++){
strcpy(o1,dfile);
sprintf(ext,"-%d.acm",q3[i]);
strcat(o1,ext,1);
fpd=fopen(o1,"w");
fprintf(fpd,"\t\t\t\t%s\n\n",o1);
/* printf("filename=%s\n",o1); */
fclose(fpd);

/* this just opens .str file the first time  */

strcpy(o1,dfile);
sprintf(ext,"-%d.str",q3[i]);
strcat(o1,ext,1);
fpd=fopen(o1,"w");
fprintf(fpd,"$\n");
fclose(fpd);
}

/* get heading */

for (i=1;i<=nscan;i++){
strcpy(o1,dfile);
sprintf(ext,"-%d.acm",q3[i]);
strcat(o1,ext,1);
fpd=fopen(o1,"a");
fpd1=fopen("header","r");
sprintf(buffer,"");
while (*buffer != '!'){
fscanf(fpd1,"%s",buffer);
fprintf(fpd,"%s ",buffer);
}
fclose(fpd1);
fclose(fpd);
}

/* set frequency range */

fmin=1.0;
fmax=10000.0;
fmin=fmin-.0001;

/*  no. of measurements at each freq. */
```

```
m=1;

/* for the present make all scans ascending */
/* this makes operation of fp routine simpler */

q1=1;
rep=1;
if (m>1)
{
printf("specify mode (o or a)\n\n");
printf("o - do m sweeps of freq. range\n");
printf("a - take m readings at each freq., sweep once\n");
scanf("%s",&q2);
if (q2=='a')
rep=1;
else
rep=m;
}

/* get chipids written in file */
/* and get initial temperatures */

for (i=1;i<=nscan;i++)
{
multi(q3[i]);
chipid(id[q3[i]]);
chiptemp(&tem);
printf("Channel %1d \tTemperature= %4.2f Celsius\n",q3[i],tem);
strcpy(o1,dfile);
sprintf(ext,"-%d.acm",q3[i]);
strcat(o1,ext,1);
fpd=fopen(o1,"a");
fprintf(fpd,"\n\n\t\tChipid # : %5.2f\n",id[q3[i]]);
fprintf(fpd,"\t\tTemperature= %4.2f Celsius\n",tem);
fprintf(fpd,"\nfreq\t\ttemp\tchan\tgain\tphase\te1\te2\toffset\n$\n");
fclose(fpd);
}
printf("this is a wasted reading\n");
freq(1000.0);              /*  this is   */
measure(&f,&f);          /* a wasted   */
freq(0.0);
convert(f,f,f,&f,&f);  /* reading    */
printf("*********************************\n");

/* subroutine compute() is used */
/* to take actual readings and */
/* write them into the correct */
```

```
/* file */

/* this is ascending scan */
if (q1==1){
qf[0]=5.0;
qf[1]=6.0;
qf[2]=8.0;
qf[3]=10.0;
qf[4]=15.0;
qf[5]=20.0;
qf[6]=25.0;
qf[7]=30.0;
qf[8]=40.0;
for(k=1;k<=rep;k++){
scale=.0001;
           for(i=(-3);i<=3;i++){
scale=scale*10.0;
for(j=0;j<=8;j++){
   f=qf[j]*scale;
   if ((f<=fmax)&&(f>=fmin-.001)){
if (rep==1){
for(num=1;num<=m;num++){
for (i=1;i<=nscan;i++){
multi(q3[i]);
compute(i,f,dfile);
}
}
}
else    {
for (i=1;i<=nscan;i++){
multi(q3[i]);
chipid(id[q3[i]]);
compute(i,f,dfile);
}
}
}
}
}
}
}

/* this is for descending scan */
else {
qf[0]=10.0;
qf[1]=8.0;
qf[2]=6.0;
qf[3]=5.0;
qf[4]=4.0;
```

```
qf[5]=3.0;
qf[6]=2.5;
qf[7]=2.0;
qf[8]=1.5;
for(k=1;k<=rep;k++){
scale=10000.0;
for(i=3;i>=(-3);i--){
scale=scale/10.0;
for(j=0;j<=8;j++){
f=qf[j]*scale;
if ((f>=fmin)&&(f<=fmax)){
if (rep==1){
for(num=1;num<=m;num++){
for (i=1;i<=nscan;i++){
multi(q3[i]);
chipid(id[q3[i]]);
compute(i,f,dfile);
}
}
}
else    {
for (i=1;i<=nscan;i++){
multi(q3[i]);
chipid(id[q3[i]]);
compute(i,f,dfile);
}
}
}
}
}

}
}
for (i=1;i<=nscan;i++){
strcpy(o1,dfile);
sprintf(ext,"-%d.acm",q3[i]);
strcat(o1,ext,1);
fpd=fopen(o1,"a");
fprintf(fpd,"$\n");
fclose(fpd);
strcpy(o1,dfile);
sprintf(ext,"-%d.str",q3[i]);
strcat(o1,ext,1);
fpd=fopen(o1,"a");
fprintf(fpd,"$\n");
fclose(fpd);
}
printf("output in files %s#.acm\n",dfile);
```

```
return;
}

/* subroutine to read chip, compute data and print out*/

int compute(ac,fc,defile)
char defile[20];
float fc;
int ac;
{
float g,p,e1,e2,off,amp,tem;
extern FILE *fopen(),*fpi,*fpo;
extern int fclose(),port;
extern int fprintf();
extern int q3[10];
extern float id[10];
extern char ss[80];
FILE *fpd;
char o1[20],ext;

chiptemp(&tem);
freq(fc); /*excite at this frequency*/
measure(&g,&p); /*make measurement*/
freq(0.0); /*stop driving chip*/
convert(g,p,fc,&e1,&e2); /*compute data*/
offset(&off,&amp); /*compute offset*/
printf("****************************************\n");
/*print border of screen*/

/* write to accumulation file */

strcpy(o1,defile);
sprintf(ext,"-%d.acm",q3[ac]);
strcat(o1,ext,1);
fpd=fopen(o1,"a");
fprintf(fpd,"%6.3f\t%3.2f\t%1d\t%4.3f\t%4.3f\t%3.3f\t%3.4f\t%3.3f\n",
fc,tem,q3[ac],g,p,e1,e2,off); /*print to file*/
fclose(fpd);

/* write to stripped file */
/* used for computing master curve */
/* with msc.f, a spline routine */

strcpy(o1,defile);
sprintf(ext,"-%d.str",q3[ac]);
strcat(o1,ext,1);
fpd=fopen(o1,"a");
fprintf(fpd,"%9.3f%6.2f%7.2f\n",fc,g,p);
```

```
fclose(fpd);
return(0);
}

/* main program utilizing subroutines above */

int nchan,nscan,nwat,q3[10],ab[10];
float id[10],a[10],b[10],c1,d,frq[10];
char buffer[80];
struct tm *tmstr;
main()
{
extern FILE *fopen(),*fpi,*fpo;
extern int fclose();
extern int fflush(),port;
extern int fprintf();
extern int chgfile(),fchip5(),compute();
extern struct tm *tmstr;
extern float rtbis(),splint(),fpcalc(),brent(),fmin(),fmax(),sign();
extern int nchan,nscan,nwat,q3[10],ab[10];
extern float id[10],a[10],b[10],c1,d,frq[10];
extern char buffer[80],ss[80];
FILE *fpd,*fpid,*fprdg,*fpchr,*fphst,*fpdfl;
long start,time();
int i,j,clock,k,oday;
float period,fp,mc,mcp,resid;
float tim,gain,phase,e1,e2,off,amp,tem;
char c,dfile[20],ext,o1[20],chrfl[20],hstfl[20],odfile[20];

/*   this stuff sets correct flags  */
/*   including baud rate, for port  */

struct sgttyb ttyjunk;
printf("mmet program\n");
port=open(mmetport,2);

gtty(port,&ttyjunk);
ttyjunk.sg_flags &= (~ECHO);
ttyjunk.sg_flags |= (RAW);
ttyjunk.sg_ispeed = ttyjunk.sg_ospeed =B1200;
stty(port,&ttyjunk);

fpi=fdopen(port,"r");
fpo=fdopen(port,"w");

/* get input info. to run program  */

printf("number of channels to be monitored?\n");
```

```
scanf("%d",&nscan);
for (i=1;i<=nscan;i++) {
printf("input channel number and freq. to monitor at\n");
scanf("%d %f",&q3[i],&frq[i]);
printf("scan channel %d at %f Hz\n",q3[i],frq[i]);
}


/* copy heading into header file   */

fpd=fopen("header","w");
printf("input heading,ended by !\n");
while((c=getchar())!='!')
putc(c,fpd);
putc('!',fpd);
fclose(fpd);

/* this command kluged to make communications work */
chiptemp(&tem);

printf("What is the sampling interval(sec)?\n");
scanf("%f",&period);
printf("starting\n");
oday=0;
j=1;
tim=0.0;
strcpy(odfile,"");
while (j=1)
{
for (i=1;i<=nscan;i++){
time(&clock);
tmstr=localtime(&clock);

/* if day changed or program restarted */
/* rewrite heading info. into .acm and */
/* .hst files, change mmdd of filename */
/* perform new fchip5 and recompute */
/* master curve spline functions */

if (oday!=(*tmstr).tm_mday) {
strcpy(odfile,dfile);
chgfile(dfile,odfile);
fchip5();
for (k=1;k<=nwat;k++)
if (ab[k]==q3[i]) msc(q3[i]);
oday=(*tmstr).tm_mday;
tim=0.0;
}
```

```
printf("Sampled at %3.3f minutes\n",tim);
multi(q3[i]);
chipid(id[q3[i]]);
chiptemp(&tem);
freq(frq[i]);
measure(&gain,&phase);
freq(0.0);
convert(gain,phase,frq[i],&e1,&e2);
offset(&off,&amp);

/* compute fp */

fp=fpcalc(q3[i],frq[i],gain,phase);

/* compute moisture content from fp */

for (k=1;k<=nwat;k++)
if (ab[k]==q3[i])
mc=pow(10.0,(a[k]+b[k]*log10(fp)));
printf("mc= %f ppm\n",mc);

/* compute predicted moisture content */
/* from temp. and xformer characteristics */
/* also compute residual */

mcp=pow(10.0,c1-d/(273.0+tem));
resid=mc-mcp;
printf("mcp = %f ppm\n",mcp);
printf("resid = %f\n",resid);
printf("*********************************************\n");

/* open accumulation file */

strcpy(o1,dfile);
sprintf(ext,"-%d.acm",q3[i]);
strcat(o1,ext,1);
fpd=fopen(o1,"a");
fprintf(fpd,"%s",asctime(tmstr));
        fprintf(fpd,"%4.3f\t%4.3f\t%4.3f\t%4.3f\t%3.3f\t%3.4f\t%3.4f\n"
,tim,tem,gain,phase,e1,e2,off);
fclose(fpd);

/* open reading file  */

strcpy(o1,"twch");
sprintf(ext,"%d.rdg",q3[i]);
strcat(o1,ext,1);
fprdg=fopen(o1,"w");
```

```
fprintf(fprdg,"%s",asctime(tmstr));
fprintf(fprdg,"freq\ttime\ttemp\tgain\tphase\toffset\n");
fprintf(fprdg,"%9.3f%8.3f%6.2f %6.2f%7.2f%7.3f\n"
,frq[i],tim,tem,gain,phase,off);
fclose(fprdg);

/* open characteristic file for processed data */

strcpy(chrfl,"fpch");
sprintf(ext,"%d.chr",q3[i]);
strcat(chrfl,ext,1);
fpchr=fopen(chrfl,"w");
fprintf(fpchr,"%s",asctime(tmstr));
fprintf(fpchr,"temp\toffset\tfp\tmc\tmcp\tresid\n");
fprintf(fpchr,"%6.2f %7.3f%9.3f %5.1f %5.1f %5.2f\n"
,tem,off,fp,mc,mcp,resid);
fclose(fpchr);

/* open history file for processed data */

strcpy(hstfl,dfile);
sprintf(ext,"-%d.hst",q3[i]);
strcat(hstfl,ext,1);
fphst=fopen(hstfl,"a");
fprintf(fphst,"%s",asctime(tmstr));
fprintf(fphst,"%6.2f %7.3f%9.3f %5.1f %5.1f %5.2f\n"
,tem,off,fp,mc,mcp,resid);
fclose(fphst);
}
start=time(NULL);
while ((time(NULL)-start)<period) i=0;
tim=tim+period/60;
}
close(port);
}

/* called at midnight to close old files */
/* and open new files */

int chgfile(fname,ofname)
char fname[20],ofname[20];
{
extern FILE *fopen();
extern int fclose();
extern int fprintf();
extern struct tm *tmstr;
extern int nchan,nscan,nwat,q3[10],ab[10];
extern float id[10],a[10],b[10],c1,d,frq[10];
```

```
extern char buffer[80];
FILE *fpid,*fpd,*fpd1;
int i,k,day,month;
char ext,o1[20],hstfl[20];


/* open file called chipids */
/* contains all chipids for */
/* the channels */

fpid=fopen("chipids","r");
fscanf(fpid,"%d",&nchan);
printf("nchan=%d\n",nchan);
for(i=1;i<=nchan;i++){
fscanf(fpid,"%f",&id[i]);
printf("chipid=%5.2f\n",id[i]);
}
fclose(fpid);
if (nscan>nchan) {
printf("can't scan more channels than have chipids for\n");
printf("stop program and try again\n");
}

/* open file called mcfp.par */
/* contains coefficients for */
/* relating fp to mc for each */
/* channel having a moisture chip */
/* log(mc)=a + b*log(fp) */

strcpy(o1,"");
sprintf(ext,"mcfp.par");
strcat(o1,ext,1);
fpd=fopen(o1,"r");
fscanf(fpd,"%d",&nwat);
for (i=1;i<=nwat;i++){
fscanf(fpd,"%d %f %f",&ab[i],&a[i],&b[i]);
printf("%d %5.2f %5.3f\n",ab[i],a[i],b[i]);
}
fclose(fpd);

/* open file called mcT.par */
/* contains coefficients for */
/* predicting mcp for a given temp */
/* log(mcp)=c1 - d/T */

strcpy(o1,"");
sprintf(ext,"mcT.par");
strcat(o1,ext,1);
```

```
fpd=fopen(o1,"r");
fscanf(fpd,"%f %f",&c1,&d);
printf("%f %f\n",c1,d);
fclose(fpd);

/*  close out old files */

for (i=1;i<=nscan;i++){
strcpy(o1,ofname);
sprintf(ext,"-%d.acm",q3[i]);
strcat(o1,ext,1);
fpd=fopen(o1,"a");
fprintf(fpd,"$\n");
fclose(fpd);
strcpy(o1,ofname);
sprintf(ext,"-%d.hst",q3[i]);
strcat(o1,ext,1);
fpd=fopen(o1,"a");
fprintf(fpd,"$\n");
fclose(fpd);
}

/* make new fname (twmmdd) */

strcpy(fname,"");
if ((*tmstr).tm_mon<10)
sprintf(ext,"tw0%d",(*tmstr).tm_mon+1);
else
sprintf(ext,"tw%d",(*tmstr).tm_mon+1);
strcat(fname,ext,1);
if ((*tmstr).tm_mday<10)
sprintf(ext,"0%d",(*tmstr).tm_mday);
else
sprintf(ext,"%d",(*tmstr).tm_mday);
strcat(fname,ext,1);
strcpy(ofname,fname);
/* printf("fname = %s\n",fname); */


/* print filename at top */

for (i=1;i<=nscan;i++){
strcpy(o1,fname);
sprintf(ext,"-%d.acm",q3[i]);
strcat(o1,ext,1);
fpd=fopen(o1,"a");
fprintf(fpd,"\t\t\t\t%s\n\n",o1);
printf("filename=%s\n",o1);
```

```
fclose(fpd);
strcpy(hstfl,fname);
sprintf(ext,"-%d.hst",q3[i]);
strcat(hstfl,ext,1);
fpd=fopen(hstfl,"a");
fprintf(fpd,"\t\t\t\t%s\n\n",hstfl);
fclose(fpd);
}


/* copy heading into all .acm  */
/* and .hst files */

for (i=1;i<=nscan;i++){
strcpy(o1,fname);
sprintf(ext,"-%d.acm",q3[i]);
strcat(o1,ext,1);
fpd=fopen(o1,"a");
fpd1=fopen("header","r");
sprintf(buffer,"");
while (*buffer != '!'){
fscanf(fpd1,"%s",buffer);
fprintf(fpd,"%s ",buffer);
}
fclose(fpd);
fclose(fpd1);
strcpy(hstfl,fname);
sprintf(ext,"-%d.hst",q3[i]);
strcat(hstfl,ext,1);
fpd=fopen(hstfl,"a");
fpd1=fopen("header","r");
sprintf(buffer,"");
while (*buffer != '!'){
fscanf(fpd1,"%s",buffer);
fprintf(fpd,"%s ",buffer);
}
fclose(fpd);
fclose(fpd1);
}

/* copy other pertinent info. */
/* into .acm and .hst files */

for (i=1;i<=nscan;i++){
strcpy(o1,fname);
sprintf(ext,"-%d.acm",q3[i]);
strcat(o1,ext,1);
fpd=fopen(o1,"a");
```

```
fprintf(fpd,"\n\n\t\t\tChannel %d\n",q3[i]);
fprintf(fpd,"\t\t\tChipid # :  %5.2f\n",id[q3[i]]);
fprintf(fpd,"\t sampled at freq. = %9.3f Hz\n",frq[i]);
fprintf(fpd,"\ntime\ttemp\tgain\tphase\te1\te2\toffset\n$\n");
fclose(fpd);
strcpy(hstfl,fname);
sprintf(ext,"-%d.hst",q3[i]);
strcat(hstfl,ext,1);
fpd=fopen(hstfl,"a");
for (k=1;k<=nwat;k++)
if (ab[k]==q3[i])
fprintf(fpd,"\n\nmoisture chip parameters\ta=%5.2f\tb=%5.2f\n",a[k],b[k]);
fprintf(fpd,"transformer parameters\tc1=%f\td=%f\n\n",c1,d);
fprintf(fpd,"\t\t\tChannel %d\n",q3[i]);
fprintf(fpd,"\t\t\tChipid # :  %5.2f\n",id[q3[i]]);
fprintf(fpd,"\t sampled at freq. = %9.3f Hz\n",frq[i]);
fprintf(fpd,"\ntemp\toffset\tfp\tmc\tmcp\tresid\n$\n");
fclose(fpd);
}
return(0);
}
```

```
/****************************************/
/*     tscan.c */
/*      6/19/87  */
/* */
/* scans several channels, each at */
/* an individual frequency */
/* records accumulated data in */
/* .acm file and present status  */
/****************************************/
#include <stdio.h>
#include <math.h>
#include <sgtty.h>
#include <time.h>
#include <string.h>
#include "mmet.h"
#include "newfunct.c"

int max(m,n)
int m,n;
{
if (m > n) return (m);
else  return (n);
}

int nchan,nscan,q3[10];
float id[10],frq[10];
char buffer[80];
main()
{
extern FILE *fopen();
extern int fclose();
extern int fflush();
extern int fprintf();
extern int chgfile();
extern int nchan,nscan,q3[10];
extern float id[10],frq[10];
extern char buffer[80];
FILE *fpd,*fpid,*fprdg,*fpdfl;
long start,time();
struct tm *tmstr;
int i,j,clock,k,oday;
float period;
float tim,gain,phase,e1,e2,off,amp,tem;
char c,dfile[20],ext,o1[20],odfile[20];

/*   this stuff sets correct flags  */
/*   including baud rate, for port  */
```

```
struct sgttyb ttyjunk;
printf("mmet program\n");
port=open(mmetport,2);

gtty(port,&ttyjunk);
ttyjunk.sg_flags &= (~ECHO);
ttyjunk.sg_flags |= (RAW);
ttyjunk.sg_ispeed = ttyjunk.sg_ospeed =B1200;
stty(port,&ttyjunk);

fpi=fdopen(port,"r");
fpo=fdopen(port,"w");

/* get input info. to run program  */

printf("number of channels to be monitored?\n");
scanf("%d",&nscan);
for (i=1;i<=nscan;i++) {
printf("input channel number and freq. to monitor at\n");
scanf("%d %f",&q3[i],&frq[i]);
printf("scan channel %d at %f Hz\n",q3[i],frq[i]);
      fprintf(fpd,"\tchannel %d monitored at freq.=%fHz\n\n",q3[i],frq[i]);
}


/* copy heading into header file   */

fpd=fopen("header","w");
printf("input heading,ended by !\n");
while((c=getchar())!='!')
putc(c,fpd);
putc('!',fpd);
fclose(fpd);

/* this command kluged to make communications work */
chiptemp(&tem);

printf("What is the sampling interval(sec)?\n");
scanf("%f",&period);
printf("starting\n");
oday=0;
j=1;
tim=0.0;
while (j=1)
{
for (i=1;i<=nscan;i++){
time(&clock);
tmstr=localtime(&clock);
```

```
/* if day changed or program restarted */
/* rewrite heading info. into .acm and */
/* change mmdd of filename */

strcpy(odfile,"");
if (oday!=(*tmstr).tm_mday) {
chgfile(dfile,tmstr,odfile);
oday=(*tmstr).tm_mday;
tim=0.0;
}
printf("Sampled at %3.3f minutes\n",tim);
multi(q3[i]);
chipid(id[q3[i]]);
chiptemp(&tem);
freq(frq[i]);
measure(&gain,&phase);
freq(0.0);
convert(gain,phase,frq[i],&e1,&e2);
offset(&off,&amp);
printf("************************************************\n");

/* open accumulation file */

strcpy(o1,dfile);
sprintf(ext,"-%d.acm",q3[i]);
strcat(o1,ext,1);
fpd=fopen(o1,"a");
fprintf(fpd,"%s",asctime(tmstr));
        fprintf(fpd,"%4.3f\t%4.3f\t%4.3f\t%4.3f\t%3.3f\t%3.4f\t%3.4f\n"
,tim,tem,gain,phase,e1,e2,off);
fclose(fpd);

/* open reading file  */

strcpy(o1,"twch");
sprintf(ext,"%d.rdg",q3[i]);
strcat(o1,ext,1);
fprdg=fopen(o1,"w");
fprintf(fprdg,"freq\ttime\ttemp\tgain\tphase\toffset\n");
fprintf(fprdg,"%9.3f%8.3f%6.2f %6.2f%7.2f%7.3f\n"
,frq[i],tim,tem,gain,phase,off);
fclose(fprdg);

}
start=time(NULL);
while ((time(NULL)-start)<period) i=0;
tim=tim+period/60;
```

```
}
close(port);
}

/* change filename at midnight */

int chgfile(fname,date,ofname)
char fname[20],ofname[20];
struct tm *date;
{
extern FILE *fopen();
extern int fclose();
extern int fprintf();
extern int nchan,nscan,q3[10];
extern float id[10],frq[10];
extern char buffer[80];
FILE *fpid,*fpd,*fpd1;
int i,k,day,month;
char ext,o1[20];


/* open file called chipids */
/* contains all chipids for */
/* the channels */

fpid=fopen("chipids","r");
fscanf(fpid,"%d",&nchan);
printf("nchan=%d\n",nchan);
for(i=1;i<=nchan;i++){
fscanf(fpid,"%f",&id[i]);
printf("chipid=%5.2f\n",id[i]);
}
fclose(fpid);
if (nscan>nchan) {
printf("can't scan more channels than have chipids for\n");
printf("stop program and try again\n");
}

/*  close out old files */

for (i=1;i<=nscan;i++){
strcpy(o1,ofname);
sprintf(ext,"-%d.acm",q3[i]);
strcat(o1,ext,1);
fpd=fopen(o1,"a");
fprintf(fpd,"$\n");
fclose(fpd);
}
```

```
/* make new fname (twmmdd) */

strcpy(fname,"");
if ((*date).tm_mon<10)
sprintf(ext,"tw0%d",(*date).tm_mon+1);
else
sprintf(ext,"tw%d",(*date).tm_mon+1);
strcat(fname,ext,1);
if ((*date).tm_mday<10)
sprintf(ext,"0%d",(*date).tm_mday);
else
sprintf(ext,"%d",(*date).tm_mday);
strcat(fname,ext,1);
strcpy(ofname,fname);
printf("fname = %s\n",fname);


/* print filename at top */

for (i=1;i<=nscan;i++){
strcpy(o1,fname);
sprintf(ext,"-%d.acm",q3[i]);
strcat(o1,ext,1);
fpd=fopen(o1,"a");
fprintf(fpd,"\t\t\t\t%s\n\n",o1);
printf("filename=%s\n",o1);
fclose(fpd);
}


/* copy heading into all .acm files */

for (i=1;i<=nscan;i++){
strcpy(o1,fname);
sprintf(ext,"-%d.acm",q3[i]);
strcat(o1,ext,1);
fpd=fopen(o1,"a");
fpd1=fopen("header","r");
sprintf(buffer,"");
while (*buffer != '!'){
fscanf(fpd1,"%s",buffer);
fprintf(fpd,"%s ",buffer);
}
fclose(fpd);
fclose(fpd1);
}
```

```
/* copy other pertinent info. */
/* into .acm files */

for (i=1;i<=nscan;i++){
strcpy(o1,fname);
sprintf(ext,"-%d.acm",q3[i]);
strcat(o1,ext,1);
fpd=fopen(o1,"a");
fprintf(fpd,"\n\n\t\t\tChannel %d\n",q3[i]);
fprintf(fpd,"\t\t\tChipid # :  %5.2f\n",id[q3[i]]);
fprintf(fpd,"\t sampled at freq. = %9.3f Hz\n",frq[i]);
fprintf(fpd,"\ntime\ttemp\tgain\tphase\te1\te2\toffset\n$\n");
fclose(fpd);
}
return(0);
}
```

```
/*    fchip5.c   */
/*    6/18/87  */
/*    frequency scan */

#include <stdio.h>
#include <sgtty.h>  /* contains flag values for port  */
#include <time.h>
#include <string.h>
#include "newfunct.c"
#include "mmet.h"

float f;
FILE *fpd,*fpid;
char dfile[50],o1[20],ext;
main()
{
extern FILE *fopen(),*fpd,*fpid;
extern int fclose();
extern int fprintf();
extern float f;
extern char dfile[50],o1[20],ext;
long start,time();
struct tm *date;
char c,q1,q2;
float tem,fmin,fmax,qf[9],x,scale,id[5];
int j,k,l,m,n,rep,num,all,nchan,nscan,q3[10],i,clock;

/*   this stuff sets correct flags  */
/*   including baud rate, for port  */

struct sgttyb ttyjunk;
printf ("mmet program\n");
port = open (mmetport, 2);  /*  2 means read 'n' write   */

/* set port for NO echo and raw mode and set baud to 1200 */
gtty (port, &ttyjunk);  /* get  */
ttyjunk.sg_flags &= (~ECHO);   /* it was so SIMPLE in the old days */
ttyjunk.sg_flags |= (RAW);
ttyjunk.sg_ispeed = ttyjunk.sg_ospeed = B1200;
stty(port,&ttyjunk);  /*  set  */

fpi=fdopen(port,"r");
fpo=fdopen(port,"w");

printf("number of channels to be scanned?\n");
scanf("%d",&nscan);
for (i=1;i<=nscan;i++) {
printf("input channel number\n");
```

```
scanf("%d",&q3[i]);
printf("scan channel %d\n",q3[i]);
}

/* open file called chipids */
/* contains all chipids for */
/* the channels */

fpid=fopen("chipids","r");
fscanf(fpid,"%d",&nchan);
printf("nchan=%d\n",nchan);
for(i=1;i<=nchan;i++){
fscanf(fpid,"%f",&id[i]);
printf("chipid=%5.2f\n",id[i]);
}
fclose(fpid);
if (nscan>nchan) {
printf("can't scan more channels than have chipids for\n");
printf("stop program and try again\n");
}

/* set proper filename using timestamp */

time(&clock);
date=localtime(&clock);
strcpy(dfile,"");
if ((*date).tm_mon<10)
sprintf(ext,"fw0%d",(*date).tm_mon+1);
else
sprintf(ext,"fw%d",(*date).tm_mon+1);
strcat(dfile,ext,1);
if ((*date).tm_mday<10)
sprintf(ext,"0%d",(*date).tm_mday);
else
sprintf(ext,"%d",(*date).tm_mday);
strcat(dfile,ext,1);

/* this is a brute force routine */
/* for opening all the correct */
/* data files, it is repeated */
/* throughout the program */

for (i=1;i<=nscan;i++){
strcpy(o1,dfile);
sprintf(ext,"-%d.acm",q3[i]);
strcat(o1,ext,1);
fpd=fopen(o1,"w");
fprintf(fpd,"\t\t\t\t%s\n\n",o1);
```

```
printf("filename=%s\n",o1);
fclose(fpd);

/* this just opens .str file the first time  */

strcpy(o1,dfile);
sprintf(ext,"-%d.str",q3[i]);
strcat(o1,ext,1);
fpd=fopen(o1,"w");
fclose(fpd);
}

printf("input heading,followed by ! at the end\n");
while((c=getchar())!='!')
for (i=1;i<=nscan;i++){
strcpy(o1,dfile);
sprintf(ext,"-%d.acm",q3[i]);
strcat(o1,ext,1);
fpd=fopen(o1,"a");
putc(c,fpd);
fclose(fpd);
}

printf("input minimum frequency\n");
scanf("%f",&fmin);
fmin=fmin-.0001;
printf("input maximum frequency\n");
scanf("%f",&fmax);
printf("input number of measurements at each freq.\n");
scanf("%d",&m);
/* printf("specify order (a or d)\n");
printf("\na - ascending, low to high frequency\n");
printf("d - descending, high to low frequency\n");  */
printf("for the present make all scans ascending\n");
printf("this makes operation of fp routine simpler\n");
scanf("%s",&q1);
rep=1;
if (m>1)
{
printf("specify mode (o or a)\n\n");
printf("o - do m sweeps of freq. range\n");
printf("a - take m readings at each freq., sweep once\n");
scanf("%s",&q2);
if (q2=='a')
rep=1;
else
rep=m;
}
```

```
/* get chipids written in file */
/* and get initial temperatures */

for (i=1;i<=nscan;i++)
{
multi(q3[i]);
chipid(id[q3[i]]);
chiptemp(&tem);
printf("Channel %1d \tTemperature= %4.2f Celsius\n",q3[i],tem);
strcpy(o1,dfile);
sprintf(ext,"-%d.acm",q3[i]);
strcat(o1,ext,1);
fpd=fopen(o1,"a");
printf("filename=%s\n",o1);
fprintf(fpd,"\t\tChipid # : %5.2f\n",id[q3[i]]);
fprintf(fpd,"\t\tTemperature= %4.2f Celsius\n",tem);
fprintf(fpd,"\nfreq\t\ttemp\tchan\tgain\tphase\te1
\te2\toffset\n$\n");
fclose(fpd);
}
printf("this is a wasted reading\n");
freq(1000.0);              /*  this is   */
measure(&f,&f);          /* a wasted   */
freq(0.0);
convert(f,f,f,&f,&f);  /* reading    */
printf("*******************************\n");

/* subroutine compute() is used */
/* to take actual readings and */
/* write them into the correct */
/* file */

/* this is ascending scan */
if (q1=='a'){
qf[0]=5.0;
qf[1]=6.0;
qf[2]=8.0;
qf[3]=10.0;
qf[4]=15.0;
qf[5]=20.0;
qf[6]=25.0;
qf[7]=30.0;
qf[8]=40.0;
for(k=1;k<=rep;k++){
scale=.0001;
          for(i=(-3);i<=3;i++){
scale=scale*10.0;
```

```
for(j=0;j<=8;j++){
   f=qf[j]*scale;
   if ((f<=fmax)&&(f>=fmin-.001)){
if (rep==1){
for(num=1;num<=m;num++){
for (i=1;i<=nscan;i++){
multi(q3[i]);
compute(i,q3);
}
}
}
else    {
for (i=1;i<=nscan;i++){
multi(q3[i]);
chipid(id[q3[i]]);
compute(i,q3);
}
}
}
}
}
}
}

/* this is for descending scan */
else {
qf[0]=10.0;
qf[1]=8.0;
qf[2]=6.0;
qf[3]=5.0;
qf[4]=4.0;
qf[5]=3.0;
qf[6]=2.5;
qf[7]=2.0;
qf[8]=1.5;
for(k=1;k<=rep;k++){
scale=10000.0;
for(i=3;i>=(-3);i--){
scale=scale/10.0;
for(j=0;j<=8;j++){
f=qf[j]*scale;
if ((f>=fmin)&&(f<=fmax)){
if (rep==1){
for(num=1;num<=m;num++){
for (i=1;i<=nscan;i++){
multi(q3[i]);
chipid(id[q3[i]]);
compute(i,q3);
```

```
}
}
}
else    {
for (i=1;i<=nscan;i++){
multi(q3[i]);
chipid(id[q3[i]]);
compute(i,q3);
}
}
}
}
}

}
}
for (i=1;i<=nscan;i++){
strcpy(o1,dfile);
sprintf(ext,"-%d.acm",q3[i]);
strcat(o1,ext,1);
fpd=fopen(o1,"a");
fprintf(fpd,"$\n");
fclose(fpd);
}
close (port);
printf("output in files %s#.acm\n",dfile);
}
compute(a,b)  /*subroutine to read chip,
compute data and print out*/
int a,b[10];
{
float g,p,e1,e2,off,amp,tem;
extern float f;
extern FILE *fpd,*fpid,*fopen();
extern int fclose();
extern int fprintf();
extern char dfile[50],o1[20],ext;
chiptemp(&tem);
freq(f); /*excite at this frequency*/
measure(&g,&p); /*make measurement*/
freq(0.0); /*stop driving chip*/
convert(g,p,f,&e1,&e2); /*compute data*/
offset(&off,&amp); /*compute offset*/
printf("*****************************************\n");
/*print border of screen*/

/* write to accumulation file */
```

```
strcpy(o1,dfile);
sprintf(ext,"-%d.acm",b[a]);
strcat(o1,ext,1);
fpd=fopen(o1,"a");
fprintf(fpd,"%6.3f\t%3.2f\t%1d\t%4.3f\t%4.3f\t%3.3f\t%3.4f\t%3.3f\n",
f,tem,b[a],g,p,e1,e2,off); /*print to file*/
fclose(fpd);

/* write to stripped file */
/* used for computing master curve */
/* with msc.f, a spline routine */

strcpy(o1,dfile);
sprintf(ext,"-%d.str",b[a]);
strcat(o1,ext,1);
fpd=fopen(o1,"a");
fprintf(fpd,"%9.3f%6.2f%7.2f\n",f,g,p);
fclose(fpd);
return;
}
```

```
/* contains subroutines used */
/* to send two letter commands */
/* to the microdielectrometer */

#include <stdio.h>
#include "mmet.h"

int port;
FILE *fpi, *fpo;
char ss[80];
multi(f)
int f;
{
extern int fflush();
extern int fprintf();
int chan;
fprintf(fpo,"ch%d\n",f);
fflush(fpo);
fgets(ss,80,fpi);
sscanf(ss,"%d\n",&chan);
printf("channel=%d\n",chan);
}
freq(f)
float f;
{
extern int fflush();
extern int fprintf();
float dummy;
fprintf (fpo,"fr%f\n",f);
fflush (fpo);
fgets(ss,80,fpi);
sscanf(ss,"%f",&dummy);
printf("freq=%f\n", dummy);
}
measure(pg,pp)
float *pg,*pp;
{
extern int fflush();
extern int fprintf();
printf("ml\n");
fprintf(fpo,"ml\n");
fflush(fpo);
fgets(ss,80,fpi);
sscanf(ss,"%f,%f",pg,pp);
printf("gain=%f phase=%f\n",*pg,*pp);
}
measurex(pg,pp)
float *pg,*pp;
```

```
{
extern int fflush();
extern int fprintf();
int i;
printf("ml\n");
for(i=0;i<=4;i++){
fprintf(fpo,"ml\n");
fflush(fpo);
fgets(ss,80,fpi);
/* printf("I read %s ... \n", ss);
*/ sscanf(ss,"%f,%f",pg,pp);
}
printf("gain=%f phase=%f\n",*pg,*pp);
}
convert(g,p,f,pe1,pe2)
float g,p,f;
float *pe1,*pe2;
{
extern int fflush();
extern int fprintf();
fprintf(fpo,"cv%f,%f,%f\n",g,p,f);
fflush(fpo);
fgets(ss,80,fpi);
/* printf("I read %s ... \n", ss);
*/ sscanf(ss,"%f,%f",pe1,pe2);
printf("e1=%f e2=%f\n",*pe1,*pe2);
}
convertc(g,p,f,pe1,pe2)
float g,p,f;
float *pe1,*pe2;
{
extern int fflush();
extern int fprintf();
fprintf(fpo,"cc%f,%f,%f\n",g,p,f);
fflush(fpo);
fgets(ss,80,fpi);
/* printf("I read %s ... \n", ss);
*/ sscanf(ss,"%f,%f",pe1,pe2);
printf("e1=%f e2=%f\n",*pe1,*pe2);
}

chiptemp(pt)
float *pt;
{
extern int fflush();
extern int fprintf();
fprintf(fpo,"ct\n");
fflush(fpo);
```

```
fgets(ss,80,fpi);
sscanf(ss,"%f",pt);
printf("%3.2f Celsius\n",*pt);
}
chipid(id)
float id;
{
extern int fflush();
extern int fprintf();
float dummy;
fprintf(fpo,"ci%f\n",id);
fflush(fpo);
fgets(ss,80,fpi);
sscanf(ss,"%f",&dummy);
printf("ci%5.2f \n",dummy);
}

offset(pg,pp)
float *pg,*pp;
        {
        extern int fflush();
        extern int fprintf();
        extern int fscanf();
        fprintf(fpo,"og\n");
/*      printf("og\n");
*/      fflush(fpo);
        fgets(ss,80,fpi);
/* printf("I read %s ... \n", ss);
*/ sscanf(ss,"%f,%f",pg,pp);
printf("offset=%f mul=%f\n",*pg,*pp);
        }
whatfreq(df,pf)
float df;
float *pf;
{

extern int fflush();
extern int fprintf();
extern int fscanf();
printf("wf%f\n",df);
fprintf(fpo,"wf%f\n",df);
fflush(fpo);
fgets(ss,80,fpi);
printf("I read %s ... \n", ss);
sscanf(ss,"%f",pf);
}
selftest(pa,pb,pc,pd)
float *pa,*pb,*pc,*pd;
```

```
{
extern int fflush();
extern int fprintf();
extern int fscanf();
printf("xt\n");
fprintf(fpo,"xt\n");
fflush(fpo);
fgets(ss,80,fpi);
printf("I read %s ... \n", ss);
sscanf(ss,"%f,%f,%f,%f",pa,pb,pc,pd);
}
```

MMET.H

```
#define mmetport "/dev/tty0"
```

HEADER

```
use wmod, does freq. scan & msc update - WHH
the actual .par values are best estimate from
previous runs
start from virtual console after sytem crashed at 03:11, 07/16/87
start time 11:25, 07/16/87
renamed fw0716-2.acm to fw0716-2.1ac to preserve original midnight
frequency scan when process is restarted  WHH
!
```

CHIPIDS

```
2
88979.31
88006.31
```

MCFP.PAR

```
1
2  .743 .376
```

MCT.PAR

```
5.683  1265
```

```
/************************************************/
/* pltfile.c    */
/* 8/28/87      */
/*      */
/* converts frequency scan data files to   */
/* plot files for   */
/* XENIX  plot routine */
/************************************************/

#include <stdio.h>
#include <math.h>

float g[800],p[800],f[800],tem[800],e1[800],e2[800],off[800],arr[800];

countlines(fpd)
FILE *fpd;
{
int n;
char c;
n=0;
while((c=getc(fpd))!='$'){
;
}
c='i';
while((c=getc(fpd))!='$')
if (c=='\n') n++;
return(n-1);
}
sort(arr,amax,amin,nl)
float *amax,*amin,arr[800];
int nl;
{
int n;
*amax=arr[1];
*amin=arr[1];
for (n=2;n<=nl;n++){
if (arr[n]<=*amin){
*amin=arr[n];
}
else if (arr[n]>=*amax){
*amax=arr[n];
}
}
return(0);
}
main()
{
char dfile[20],o1[20],o2[20],o3[20],ext,c,q2;
```

```
extern float g[800],p[800],f[800],tem[800],e1[800],e2[800],off[800];
float omax,omin,tmax,tmin,emax,emin;
int nl,n,q1;
FILE *fpd,*fpo1,*fpo2,*fpo3,*fpo4,*fopen();
printf("input file to be processed\n");
scanf("%s",dfile);
printf("# of columns in data file?\n");
scanf("%d",&q1);
if (q1>=6) {
printf("do you want plots of e1 and e2 (y or n)?\n");
scanf("%s",&q2);
}
fpd=fopen(dfile,"r");
strcpy(o1,dfile);
sprintf(ext,"gp");
strcat(o1,ext,1);
fpo1=fopen(o1,"w");
if (q1>3) {
strcpy(o2,dfile);
sprintf(ext,"ot");
strcat(o2,ext,1);
fpo2=fopen(o2,"w");
if (q2=='y') {
strcpy(o3,dfile);
sprintf(ext,"e12");
strcat(o3,ext,1);
fpo3=fopen(o3,"w");
}
}
nl=countlines(fpd);
fclose(fpd);
fpd=fopen(dfile,"r");
printf("# of lines %d\n",nl);
while((c=getc(fpd))!='$'){
;
}
for (n=1;n<=nl;n++){
if (q1==3){
fscanf(fpd,"%f\t%f\t%f",&f[n],&g[n],&p[n]);
}
else if (q1==4){
fscanf(fpd,"%f\t%f\t%f\t%f",&f[n],&g[n],
&p[n],&off[n]);
}
else if (q1==6){
fscanf(fpd,"%f\t%f\t%f\t%f\t%f\t%f",&f[n],&g[n],
&p[n],&e1[n],&e2[n],&off[n]);
}
```

```
else if (q1==7){
fscanf(fpd,"%f\t%f\t%f\t%f\t%f\t%f\t%f",&f[n],
&tem[n],&g[n],&p[n],&e1[n],&e2[n],&off[n]);
}
else {
fscanf(fpd,"%f\t%f\t%*d\t%f\t%f\t%f\t%f\t%f",&f[n],
&tem[n],&g[n],&p[n],&e1[n],&e2[n],&off[n]);
}
if (p[n]>0.0) p[n]=0.0;
}
fprintf(fpo1,"window horizontal 135/140 vertical 1/2\n");
fprintf(fpo1,"title top \" %s \"\n",dfile);
fprintf(fpo1,"label left \" gain \" \n");
fprintf(fpo1,"x axis scale -3 4 ticks 7 grid\n");
fprintf(fpo1,"y axis scale -45 0 ticks 9 grid\n");
if (q1==3){
fprintf(fpo1,"plot spline\n");
}
else
{
fprintf(fpo1,"plot dots\n");
}
for (n=1;n<=nl;n++)  fprintf(fpo1,"%f\t%f\n",log10(f[n]),g[n]);
fprintf(fpo1,"\n window horizontal 135/140 vertical 1/2\n");
fprintf(fpo1,"label left \" phase \" \n");
fprintf(fpo1,"label bottom \" log freq \" \n");
fprintf(fpo1,"x axis scale -3 4 ticks 7 grid\n");
fprintf(fpo1,"y axis scale -90 0 ticks 9 grid\n");
if (q1==3){
fprintf(fpo1,"plot spline\n");
}
else
{
fprintf(fpo1,"plot dots\n");
}
for (n=1;n<=nl;n++) fprintf(fpo1,"%f\t%f\n",log10(f[n]),p[n]);
if (q1>3){
sort(off,&omax,&omin,nl);
fprintf(fpo2,"window horizontal 135/140 vertical 1/2\n");
fprintf(fpo2,"title top \" %s \"\n",dfile);
fprintf(fpo2,"label left \" offset \" \n");
fprintf(fpo2,"x axis scale -3 4 ticks 7 grid\n");
fprintf(fpo2,"y axis scale %f %f grid\n",omin,omax);
fprintf(fpo2,"plot dots\n");
for (n=1;n<=nl;n++)
    fprintf(fpo2,"%f\t%f\n",log10(f[n]),off[n]);
if (q1>6){
sort(tem,&tmax,&tmin,nl);
```

```
fprintf(fpo2,"\n window horizontal 135/140
      vertical 1/2  \n");
fprintf(fpo2,"label left \" temperature \" \n");
fprintf(fpo2,"label bottom \" log freq \" \n");
fprintf(fpo2,"x axis scale -3 4 ticks 7 grid\n");
fprintf(fpo2,"y axis scale %f %f grid\n",tmin,tmax);
fprintf(fpo2,"plot dots\n");
for (n=1;n<=nl;n++)
   fprintf(fpo2,"%f\t%f\n",log10(f[n]),tem[n]);
}
if (q2=='y'){
sort(e1,&emax,&emin,nl);
fprintf(fpo3,"window horizontal 135/140
               vertical 1/2\n");
fprintf(fpo3,"title top \" %s \"\n",dfile);
fprintf(fpo3,"label left \" e1 (1E-11
F/m) \" \n");
fprintf(fpo3,"x axis scale -3 4 ticks 7
                grid\n");
fprintf(fpo3,"y axis scale %f %f grid\n"
                ,emin,emax);
fprintf(fpo3,"plot dots\n");
for (n=1;n<=nl;n++)  fprintf(fpo3,"%f\t%f\n"
     ,log10(f[n]),e1[n]*8.854E-12/1E-11);
fprintf(fpo3,"\n window horizontal 135/140
                 vertical 1/2\n");
fprintf(fpo3,"label left \" log e2 (1E-11 F/m)
                \" \n");
fprintf(fpo3,"label bottom \" log freq \" \n");
fprintf(fpo3,"x axis scale -3 4 ticks 7
               grid\n");
fprintf(fpo3,"y axis log grid\n");
fprintf(fpo3,"plot dots\n");
for (n=1;n<=nl;n++) fprintf(fpo3,"%f\t%f\n"
         ,log10(f[n]),e2[n]*8.854E-12/1E-11);
}
}
fclose(fpd);
fclose(fpo1);
if (q1>3) {
fclose(fpo2);
if (q2=='y') fclose(fpo3);
}
}
```

```
/************************************************/
/* tpltacm.c    */
/* 6/19/87      */
/*      */
/* converts .acm files to plot files for   */
/* XENIX  plot routine */
/************************************************/

#include <stdio.h>
#include <math.h>

float g[800],p[800],t[800],tem[800],e1[800],e2[800],off[800],arr[800];

/* counts number of lines between $'s */
/* in a data file */

countlines(fpd)
FILE *fpd;
{
int n;
char c;
n=0;
while((c=getc(fpd))!='$'){
;
}
c='i';
while((c=getc(fpd))!='$')
if (c=='\n') n++;
return(n-1);
}

/* finds minimum and maximum values of  */
/* a column of numbers */

sort(arr,amax,amin,nl)
float *amax,*amin,arr[800];
int nl;
   {
int n;
*amax=arr[1];
*amin=arr[1];
for (n=2;n<=nl;n++){
if (arr[n]<=*amin){
*amin=arr[n];
}
else if (arr[n]>=*amax){
*amax=arr[n];
}
```

```
}
return(0);
}
main()
{
char dfile[20],o1[20],o2[20],ext,c,q;
extern float g[800],p[800],t[800],tem[800],e1[800],e2[800],off[800];
float omax,omin,tmax,tmin,tmax1,tmin1,scale,gmax,gmin,pmax,pmin;
int nl,n,q1;
FILE *fpd,*fpo1,*fpo2,*fpo3,*fpo4,*fopen();
printf("input file to be processed\n");
scanf("%s',dfile);

/* number of columns stored in q1 */

q1=7;
printf("input time scale factor (in minutes)\n");
scanf("%f",&scale);

/* files have a timestamp
printf("does file have a timestamp (y or n)?");
scanf("%s",&q);
*/ q='y';

/* open 3 files to plot fp, mc, offset, temp, mcp and residuals */

fpd=fopen(dfile,"r");
strcpy(o1,dfile);
sprintf(ext,"gp");
strcat(o1,ext,1);
fpo1=fopen(o1,"w");
strcpy(o2,dfile);
sprintf(ext,"ot");
strcat(o2,ext,1);
fpo2=fopen(o2,"w");
nl=countlines(fpd);
fclose(fpd);

/* open data file and countlines */

fpd=fopen(dfile,"r");
while((c=getc(fpd))!='$'){
;
}
if (q=='y') nl=nl/2;
printf("# of lines %d\n",nl);

/* store data in arrays */
```

```
for (n=1;n<=nl;n++){
if (q1==3){
fscanf(fpd,"%f\t%f\t%f",&t[n],&g[n],&p[n]);
}
else if (q1==4){
fscanf(fpd,"%f\t%f\t%f\t%f",&t[n],&g[n],
&p[n],&off[n]);
}
else if (q1==6){
fscanf(fpd,"%f\t%f\t%f\t%f\t%f\t%f",&t[n],&g[n],
&p[n],&e1[n],&e2[n],&off[n]);
}
else {
if (q=='n'){
   fscanf(fpd,"%f\t%f\t%f\t%f\t%f\t%f\t%f",&t[n],
&tem[n],&g[n],&p[n],&e1[n],&e2[n],&off[n]);
}
else
{
   fscanf(fpd,"%*s\t%*s\t%*s\t%*s\t%*s\n
         %f\t%f\t%f\t%f\t%f\t%f\t%f"
,&t[n],&tem[n],&g[n],&p[n],&e1[n]
,&e2[n],&off[n]);
}
}
/* set positive phase to zero */
if (p[n]>0.0) p[n]=0.0;
t[n]=scale*n;
}

/* setup files for gain and phase */

sort(t,&tmax1,&tmin1,nl);
sort(g,&gmax,&gmin,nl);
fprintf(fpo1,"window horizontal 135/140 vertical 1/2\n");
fprintf(fpo1,"title top \" %s \"\n".dfile);
fprintf(fpo1,"label left \" gain \" \n");
fprintf(fpo1,"x axis scale %f %f grid\n",tmin1,tmax1);
fprintf(fpo1,"y axis scale %f %f grid\n",gmin,gmax);
if (q1==3){
fprintf(fpo1,"plot spline\n");
}
else
{
fprintf(fpo1,"plot dots\n");
}
for (n=1;n<=nl;n++)  fprintf(fpo1,"%f\t%f\n",t[n],g[n]);
```

```
sort(p,&pmax,&pmin,nl);
fprintf(fpo1,"\n window horizontal 135/140 vertical 1/2\n");
fprintf(fpo1,"label left \" phase \" \n");
fprintf(fpo1,"label bottom \" time (min) \" \n");
fprintf(fpo1,"x axis scale %f %f grid\n",tmin1,tmax1);
fprintf(fpo1,"y axis scale %f %f grid\n",pmin,pmax);
if (q1==3){
fprintf(fpo1,"plot spline\n");
}
else
{
fprintf(fpo1,"plot dots\n");
}
for (n=1;n<=nl;n++) fprintf(fpo1,"%f\t%f\n",t[n],p[n]);

/* setup files for offset and temp. */

if (q1>3){
sort(off,&omax,&omin,nl);
fprintf(fpo2,"window horizontal 135/140 vertical 1/2\n");
fprintf(fpo2,"title top \" %s \"\n",dfile);
fprintf(fpo2,"label left \" offset \" \n");
fprintf(fpo2,"x axis scale %f %f grid\n",tmin1,tmax1);
fprintf(fpo2,"y axis scale %f %f grid\n",omin,omax);
fprintf(fpo2,"plot dots\n");
for (n=1;n<=nl;n++)
    fprintf(fpo2,"%f\t%f\n",t[n],off[n]);
if (q1==7){
sort(tem,&tmax,&tmin,nl);
fprintf(fpo2,"\n window horizontal 135/140
          vertical 1/2  \n");
fprintf(fpo2,"label left \" temperature \" \n");
fprintf(fpo2,"label bottom \" time (min) \" \n");
fprintf(fpo2,"x axis scale %f %f grid\n",tmin1,tmax1);
fprintf(fpo2,"y axis scale %f %f grid\n",tmin,tmax);
fprintf(fpo2,"plot dots\n");
for (n=1;n<=nl;n++)
   fprintf(fpo2,"%f\t%f\n",t[n],tem[n]);
}
}
fclose(fpd);
fclose(fpo1);
fclose(fpo2);
}
```

```
/************************************************/
/* tplthst.c    */
/* 6/19/87      */
/*      */
/* converts .hst files to plot files for   */
/* XENIX  plot routine */
/************************************************/

#include <stdio.h>
#include <math.h>

float fp[800],mc[800],t[800],tem[800],mcp[800],resid[800],off[800];

/* counts number of lines between $'s */
/* in a data file */

countlines(fpd)
FILE *fpd;
{
int n;
char c;
n=0;
while((c=getc(fpd))!='$'){
;
}
c='i';
while((c=getc(fpd))!='$')
if (c=='\n') n++;
return(n-1);
}

/* finds minimum and maximum values of  */
/* a column of numbers */

sort(arr,amax,amin,nl)
float *amax,*amin,arr[800];
int nl;
   {
int n;
*amax=arr[1];
*amin=arr[1];
for (n=2;n<=nl;n++){
if (arr[n]<=*amin){
*amin=arr[n];
}
else if (arr[n]>=*amax){
*amax=arr[n];
}
```

```
}
return(0);
}
main()
{
char dfile[20],o1[20],o2[20],o3[20],ext,c,q;
extern float fp[800],mc[800],t[800],tem[800],mcp[800],resid[800];
float fpmax,fpmin,tmax,tmin,tmax1,tmin1,scale,mcmax,mcmin;
float mcpmax,mcpmin,residmax,residmin,omin,omax;
int nl,n,q1;
FILE *fpd,*fpo1,*fpo2,*fpo3,*fpo4,*fopen();
printf("input file to be processed\n");
scanf("%s",dfile);

/* number of columns stored in q1 */

q1=6;
printf("input time scale factor (in minutes)\n");
scanf("%f",&scale);

/* files have a timestamp
printf("does file have a timestamp (y or n)?");
scanf("%s",&q);
*/ q='y';

/* open 3 files to plot fp, mc, offset, temp, mcp and residuals */

fpd=fopen(dfile,"r");
strcpy(o1,dfile);
sprintf(ext,"fm");
strcat(o1,ext,1);
fpo1=fopen(o1,"w");
strcpy(o2,dfile);
sprintf(ext,"ot");
strcat(o2,ext,1);
fpo2=fopen(o2,"w");
strcpy(o3,dfile);
sprintf(ext,"mr");
strcat(o3,ext,1);
fpo3=fopen(o3,"w");
nl=countlines(fpd);
fclose(fpd);

/* open data file and countlines */

fpd=fopen(dfile,"r");
while((c=getc(fpd))!='$'){
;
```

```
}
if (q=='y') nl=nl/2;
printf("# of lines %d\n",nl);

/* store data in arrays */

for (n=1;n<=nl;n++){
if (q1==6){
fscanf(fpd,"%*s\t%*s\t%*s\t%*s\t%*s\n
%f\t%f\t%f\t%f\t%f\t%f"
,&tem[n],&off[n],&fp[n],&mc[n],&mcp[n],&resid[n]);
}
else {
fscanf(fpd,"%*s\t%*s\t%*s\t%*s\t%*s\n
%f\t%f\t%f\t%f\t%f\t%f"
,&t[n],&tem[n],&off[n],&fp[n],&mc[n],&mcp[n]
,&resid[n]);
}
t[n]=scale*n;
}

/* setup plot file for fp and mc */

sort(t,&tmax1,&tmin1,nl);
sort(fp,&fpmax,&fpmin,nl);
fprintf(fpo1,"window horizontal 135/140 vertical 1/2\n");
fprintf(fpo1,"title top \" %s \"\n",dfile);
fprintf(fpo1,"label left \" fp (Hz)\" \n");
fprintf(fpo1,"x axis scale %f %f grid\n",tmin1,tmax1);
fprintf(fpo1,"y axis scale %f %f grid\n",fpmin,fpmax);
fprintf(fpo1,"plot dots\n");
for (n=1;n<=nl;n++)  fprintf(fpo1,"%f\t%f\n",t[n],fp[n]);
sort(mc,&mcmax,&mcmin,nl);
fprintf(fpo1,"\n window horizontal 135/140 vertical 1/2\n");
fprintf(fpo1,"label left \" mc (ppm) \" \n");
fprintf(fpo1,"label bottom \" time (min) \" \n");
fprintf(fpo1,"x axis scale %f %f grid\n",tmin1,tmax1);
fprintf(fpo1,"y axis scale %f %f grid\n",mcmin,mcmax);
fprintf(fpo1,"plot dots\n");
for (n=1;n<=nl;n++) fprintf(fpo1,"%f\t%f\n",t[n],mc[n]);

/* setup plot file for offset and temp */

sort(off,&omax,&omin,nl);
fprintf(fpo2,"window horizontal 135/140 vertical 1/2\n");
fprintf(fpo2,"title top \" %s \"\n",dfile);
fprintf(fpo2,"label left \" offset \" \n");
fprintf(fpo2,"x axis scale %f %f grid\n",tmin1,tmax1);
```

```
fprintf(fpo2,"y axis scale %f %f grid\n",omin,omax);
fprintf(fpo2,"plot dots\n");
for (n=1;n<=nl;n++)     fprintf(fpo2,"%f\t%f\n",t[n],off[n]);
sort(tem,&tmax,&tmin,nl);
fprintf(fpo2,"\n window horizontal 135/140 vertical 1/2  \n");
fprintf(fpo2,"label left \" temperature \" \n");
fprintf(fpo2,"label bottom \" time (min) \" \n");
fprintf(fpo2,"x axis scale %f %f grid\n",tmin1,tmax1);
fprintf(fpo2,"y axis scale %f %f grid\n",tmin,tmax);
fprintf(fpo2,"plot dots\n");
for (n=1;n<=nl;n++)   fprintf(fpo2,"%f\t%f\n",t[n],tem[n]);

/* setup plot file for mcp and residual */

sort(mcp,&mcpmax,&mcpmin,nl);
fprintf(fpo3,"window horizontal 135/140 vertical 1/2\n");
fprintf(fpo3,"title top \" %s \"\n",dfile);
fprintf(fpo3,"label left \" mcp (ppm) \" \n");
fprintf(fpo3,"x axis scale %f %f grid\n",tmin1,tmax1);
fprintf(fpo3,"y axis scale %f %f grid\n",mcpmin,mcpmax);
fprintf(fpo3,"plot dots\n");
for (n=1;n<=nl;n++)     fprintf(fpo3,"%f\t%f\n",t[n],mcp[n]);
sort(resid,&residmax,&residmin,nl);
fprintf(fpo3,"window horizontal 135/140 vertical 1/2\n");
fprintf(fpo3,"label left \" residual (ppm) \" \n");
fprintf(fpo3,"label bottom \" time (min) \" \n");
fprintf(fpo3,"x axis scale %f %f grid\n",tmin1,tmax1);
fprintf(fpo3,"y axis scale %f %f grid\n",residmin,residmax);
fprintf(fpo3,"plot dots\n");
for (n=1;n<=nl;n++)     fprintf(fpo3,"%f\t%f\n",t[n],resid[n]);
fclose(fpd);
fclose(fpo1);
fclose(fpo2);
fclose(fpo3);
}
```

```
c       msc.f
c       6/19/87
c
c       calculate the second derivatives for
c       spline interpolation and fp for master curve
c       important - data must be input from high to low freq.
c       reads data from fwmmdd-x.str
c       writes to chx.mc
c       requires brent.f, splint.f and spline.f
c
        program msc

        real g(200),p(200),f(200),py2(200),fmean,pmin,fp,tol
        real gy2(200),gmid
        integer N,i,klo,khi,kmin
        character mcfile*20,chan*1,strfl*20,dfile*6
        common /set/g,p,f,py2,gy2,N
        print*,'input channel #'
        read 5,chan
  5        format(A1)
        print*,'input generic filename (must be in this form fwmmdd)!'
        read 6,dfile
  6        format(A6)
        mcfile='ch'//chan//'.mc'
        strfl=dfile//'-'//chan//'.str'
        print*,'reading',strfl
c
c       reverse data order to low to high freq
c       easier to do here and spline routine
c       requires this order in x coordinate
c
        open(unit=3,file=strfl,status='old')
        rewind(unit=3)
        do 10 i=1,200
        read (3,12,end=19) f(i),g(i),p(i)
  12        format(F9.3,F6.2,F7.2)
c       print*,'i=',i,'f=',f(i),'p=',p(i)
  10        continue
        close(unit=3)
  19        print*,'reached end of file'
        print*,'there are ',i-1,'total datapts.'
        N=i-1
c
c       calculate second derivatives for spline routine
c       use natural spline so set endpt. second derivatives to 0
c
        call spline(f,p,N,1E31,1E31,py2)
        call spline(f,g,N,1E31,1E31,gy2)
```

```
c
c       now find fp, where phase is at minimum
c       using Brent's search routine
c
c       first compute bracketing interval
c
        kmin=1
        pmin=p(1)
        do 1 k=2,N
        if (p(k).lt.pmin) then
                pmin=p(k)
                kmin=k
        endif
  1        continue
c       if (p(kmin-1).lt.p(kmin+1)) then
c               klo=kmin-1
c               khi=kmin
c       else
c               klo=kmin
c               khi=kmin+1
c       end if
        klo=kmin-1
        khi=kmin+1
        print*,'brackets for routine'
        print*,'klo= ',klo,'freq=',f(klo)
        print*,'khi= ',khi,'freq=',f(khi)
c
c       compute fp by searching for phase minimum
c       and also determine gain at fp
c
        fmean=f(kmin)
        tol=.001
        pmin=brent(f(klo),fmean,f(khi),tol,fp)
        gmid=splint(f,g,gy2,N,fp)
        print*,'fp=',fp,' gmid=',gmid,' pmin=',pmin
c
c       write to file phase, normalized freq and second derivs.
c
        open(unit=3,file=mcfile)
        rewind(unit=3)
        write (3,26) gmid
  26        format (F6.2)
        write (3,25) N
  25        format(I4)
        write (3,27) klo+1
  27        format(I2)
        do 15 i=1,N
        write(3,17) f(i)/fp,p(i),py2(i)
```

```
17        format(F12.5,F7.2,E13.5)
15        continue
      endfile(unit=3)
      close(unit=3)
      end
```

```
c       fpcalc.f
c       6/18/87
c
c       computes fp from spline function
c       of master curve and input gain-phase value
c       reads chx.mc and twchx.rdg
c       writes to fpchx.chr
c       requires splint.f
c
        program fpcalc

        real g,y,x,p(200),f(200),y2(200)
        integer N,i
        character rdgfl*20,mcfl*20,chan*1,chrfl*20,hstfl*20
        character dum*50
        common /set/p,f,y2,N
        print*,'input channel #'
        read 3,chan
  3        format(A1)
        rdgfl='twch'//chan//'.rdg'
        mcfl='ch'//chan//'.mc'
        chrfl='fpch'//chan//'.chr'
        hstfl='fpmmdd-'//chan//'.hst'
        open(unit=3,file=rdgfl,status='old')
        rewind(unit=3)
        read (3,25) dum
  25        format(A50)
        read (3,27) x,t,temp,g,y,off
  27        format(F9.3,F8.3,F6.2,1X,F6.2,F7.2,F7.3)
        close(unit=3)
        open(unit=3,file=mcfl)
        rewind(unit=3)
        read (3,2) gmid
  2        format(F6.2)
        read (3,7) N
  7        format(I4)
        read (3,8) kmid
  8        format(I2)
        do 10 i=1,N
        read (3,12) f(i),p(i),y2(i)
  12        format(F12.5,F7.2,E13.5)
c       print*,'f=',f(i),' p=',p(i),' y2=',y2(i)
  10        continue
        close(unit=3)
c
c
c       determine on which side of
c       phase peak we're on
```

```
c       set bracketing interval from
c       one end of scan to 10% over
c       fp
c
        if (gmid.lt.g) then
                klo=1
                khi=kmid
                f(khi)=1.1
        else
                klo=kmid
                khi=N
                f(klo)=.9
        endif
        p1=splint(f,p,y2,N,1.0)-y
        if (p1.ge.0.0) print*,'problem - phase lower than estimated peak'
c
c       use bisection method to find root
c       from Numerical Recipes
c
        tol=.01
        fest=rtbis(f(klo),f(khi),tol,y)
        if (fest.ne.0.0) then
                fp=x/fest
        else
                fp=x
        end if
        print*,'fp=',fp
        open(unit=3,file=chrfl)
        rewind(unit=3)
        write(3,100)
  100       format(3X,'time',TR6,'chan',TR7,'temp',TR9,'off',TR11,'fp')
        write(3,110) t,chan,temp,off,fp
  110       format(F8.3,TR7,A1,TR7,F6.2,TR7,F6.3,TR7,F9.3)
        endfile(unit=3)
        close(unit=3)
        end
c
c       bisection root finding algorithm
c       from Numerical Recipes
c
        function rtbis(x1,x2,xacc,dc)
        parameter (jmax=40)
        real p(200),f(200),y2(200)
        integer N
        common /set/p,f,y2,N
        fmid=splint(f,p,y2,N,x2)-dc
        f1=splint(f,p,y2,N,x1)-dc
        if (f1*fmid.ge.0.) then
```

```
            rtbis=0.0
            return
        end if
        if (f1.lt.0.) then
            rtbis=x1
            dx=x2-x1
        else
            rtbis=x2
            dx=x1-x2
        end if
        do 11 j=1,jmax
            dx=dx*.5
            xmid=rtbis+dx
            fmid=splint(f,p,y2,N,xmid)-dc
            if (fmid.lt.0.0) rtbis=xmid
            if ((abs(dx).lt.xacc).or.(fmid.eq.0.0)) then
                print*,'iter=',j
                return
            end if
11      continue
        pause 'too many bisections'
        end
```

```
c       spline computing function
c       from Numerical Recipes, p. 88
c
        subroutine spline(x,y,n,yp1,ypn,y2)
        parameter (nmax=100)
        dimension x(n),y(n),y2(n),u(nmax)
        if (yp1.gt..99E30) then
                y2(1)=0
                u(1)=0
        else
                y2(1)=-.5
                u(1)=(3./(x(2)-x(1)))*((y(2)-y(1))/(x(2)-x(1))-yp1)
        endif
        do 11 i=2,n-1
                sig=(x(i)-x(i-1))/(x(i+1)-x(i-1))
                p=sig*y2(i-1)+2.
                y2(i)=(sig-1.)/p
                u(i)=(6.*((y(i+1)-y(i))/(x(i+1)-x(i))-(y(i)-y(i-1))
     x               /(x(i)-x(i-1)))/(x(i+1)-x(i-1))-sig*u(i-1))/p
  11        continue
        if (ypn.gt..99E30) then
                qn=0
                un=0
        else
                qn=.5
                un=(3./(x(n)-x(n-1)))*(ypn-(y(n)-y(n-1))/(x(n)-x(n-1)))
        endif
        y2(n)=(un-qn*u(n-1))/(qn*y2(n-1)+1.)
        do 12 k=n-1,1,-1
                y2(k)=y2(k)*y2(k+1)+u(k)
  12        continue
        return
        end
```

```
c       spline interpolation routine
c       from Numerical Recipes, p. 289
c
        function splint(xa,ya,y2a,n,x)
        dimension xa(n),ya(n),y2a(n)
c       print*,'searching with freq=',x
        klo=1
        khi=n
  1        if (khi-klo.gt.1) then
                k=(khi+klo)/2
c               print*,'k=',k
c               print*,'xa(k)= ',xa(k)
                if (xa(k).gt.x) then
                        khi=k
                else
                        klo=k
                end if
        goto 1
        endif
c       print*,'brackets for routine'
c       print*,'klo= ',klo,'khi= ',khi
        h=xa(khi)-xa(klo)
        if (h.eq.0.) pause 'bad  xa input'
        a=(xa(khi)-x)/h
        b=(x-xa(klo))/h
        y=a*ya(klo)+b*ya(khi)+
     x          ((a**3.-a)*y2a(klo)+(b**3.-b)*y2a(khi))*(h**2.)/6
        splint=y
        return
        end
```

```
c
c       minimization routine
c       from Numerical Recipes, p.253
c
        function brent(ax,bx,cx,tol,xmin)
        parameter (itmax=1000,cgold=.3819660,zeps=1.0E-10)
        real p(200),f(200),py2(200),gy2(200),g(200)
        integer N
        common /set/g,p,f,py2,gy2,N
        a=amin1(ax,cx)
        b=amax1(ax,cx)
        v=bx
        w=v
        x=v
        e=0.
        fx=splint(f,p,py2,N,x)
        print*,'initial freq=',x
        print*,'p=',fx
        fv=fx
        fw=fx
        do 11 iter=1,itmax
                xm=.5*(a+b)
                tol1=tol*abs(x)+zeps
                tol2=2.*tol1
                print*,'x-xm=',abs(x-xm)
                print*,'tol2=',tol2,'halfdif=',.5*(b-a)
                print*,'toler=',tol2-0.5*(b-a)
                z1=abs(x-xm)
                z2=tol2-0.5*(b-a)
c               if (abs(x-xm) .le. tol2-0.5*(b-a)) then
                if (z1.le.z2) then
                        print*,'if true'
                         goto 3
                end if
                z1=abs(e)
c               if (abs(e).gt.tol1) then
                if (z1.gt.tol1) then
                        r=(x-w)*(fx-fv)
                        q=(x-v)*(fx-fw)
                        p1=(x-v)*q-(x-w)*r
                        q=2.*(q-r)
                        if (q.gt.0.) p1=-p1
                        q=abs(q)
                        etemp=e
                        e=d
                        z1=abs(p1)
                        z2=abs(.5*q-etemp)
                        z3=q*(a-x)
```

```
                        z4=q*(b-x)
c                    if ((abs(p1).ge.abs(.5*q-etemp)).or.(p1.le.q*(a-x))
c     x                        .or.(p1.ge.q*(b-x))) goto 1
                        if ((z1.ge.z2).or.(p1.le.z3).or.(p1.ge.z4)) goto 1
                        d=p1/q
                        u=x+d
                        z1=u-a
                        z2=b-u
c                       if (((u-a).lt.tol2).or.((b-u).lt.tol2))
c     x                             d=sign(tol1,xm-x)
                    if ((z1.lt.tol2).or.(z2.lt.tol2)) d=sign(tol1,xm-x)
                        goto 2
                end if
  1                 if (x.ge.xm) then
                        e=a-x
                else
                        e=b-x
                end if
                d=cgold*e
                z1=abs(d)
c  2                 if (abs(d).ge.tol1) then
  2                 if (z1.ge.tol1) then
                        u=x+d
                else
                        u=x+sign(tol1,d)
                end if
                fu=splint(f,p,py2,N,u)
                print*,'iter=',iter
                print*,'guess freq=',u
                print*,'pmin=',fu
c               print*,'fu=',fu,'fx=',fx
                if (fu.le.fx) then
c               print*,'fu is less than fx'
                        if (u.ge.x) then
                                a=x
                        else
                                b=x
                        end if
                        v=w
                        fv=fw
                        w=x
                        fw=fx
                        x=u
                        fx=fu
                else
                        if (u.lt.x) then
                                a=u
                        else
```

```
                        b=u
                    end if
                    if ((fu.le.fw).or.(w.eq.x)) then
                        v=w
                        fv=fw
                        w=u
                        fw=fu
                    else if ((fu.le.fv).or.(v.eq.x).or.(v.eq.w)) then
                        v=u
                        fv=fu
                    end if
                end if
11              continue
      pause 'brent exceed maximum iterations'
3        xmin=x
      brent=fx
      return
      end
```

Appendix F

Continuum Model and Parameter Estimation Routines

Notes

- A brief description of the routines and accompanying subroutines is given, along with references to the appropriate text in the thesis.
- Next, sample input files for the various main programs are given.
- Finally, the actual program listings are given.
- These fortran routines were compiled using the Ryan McFarland Fortran software for Xenix operating systems.
- Though most of the inputs to the routine are dimensioned, the routines perform the computations using normalized parameters. Lengths are normalized to the spatial wavelength λ, admittances are normalized to $j\omega\epsilon_{ox}M_{el}$, bulk permittivities to ϵ_{ox} and bulk conductivities to $\omega\epsilon_{ox}$ – creating a normalized complex bulk permittivity $\underline{\epsilon}' - j\underline{\epsilon}''$, and surface permittivities to $\lambda\epsilon_{ox}$ and surface conductivities to $\omega\lambda\epsilon_{ox}$ – creating a normalized complex surface permittivity $\underline{\epsilon}'_s - j\underline{\epsilon}''_s$.

Main Programs

These programs utilize the continuum model (Chapter 2) as a subroutine for computing a gain–phase response. The MZ*.FOR routines calculate the response at one frequency, a user input frequency range, or at the 58 discrete frequencies used by the microdielectrometer. The other routines are all parameter estimation routines.

MZ1.FOR – calculates gain–phase frequency response for ohmic, layered media above electrodes (Section 3.1 and Appendix B).
MZ2.FOR – calculates gain–phase frequency response allowing for ohmic, layered media above and below the electrodes (Section 3.1 and Appendix B).
MZ3.FOR – calculates gain–phase frequency response for dispersive medium using Lorentz sphere model (Section 8.2.1).
MZ4.FOR – calculates gain–phase frequency response for medium displaying power law dispersion (Section 8.2.1).
THEST.FOR – estimates layer thickness given real high frequency gain and permittivities of layer and surrounding medium (Section 4.2.1).
PAREST.FOR – estimates complex bulk or surface permittivity (Section 4.2.2).
HETEST.FOR – performs variable spatial wavelength multiparameter estimation (Section 4.3.3).
DISCON1.FOR – two parameter estimation of layer thickness and surrounding medium's complex bulk permittivity using spatial wavelength data (Section 4.3.6).
DISCON2.FOR – two parameter estimation of layer complex surface permittivity and surrounding medium's complex bulk permittivity using spatial wavelength data (Section 4.3.7).
TH4EST.FOR – two parameter estimation of layer thickness and bulk permittivity using variable dielectric surrounding medium (Section 4.4).
HEST.FOR – estimation of oxide layer thickness using real high frequency gain of uncoated microchip in air (Section 5.1.1).

Continuum Model

After inputting the electrode array structure and the surface capacitance density, the calculation of a gain and phase is performed by the following subroutines (as shown in Fig. 2.2). This includes solution of the mixed boundary value problem and calculation of the Y–parameters.

For electrode structures such as the microchip and macrochip having only an insulating layer and ground plane below the electrodes use:

GAIN1.FOR – Calls the subroutines listed below for the actual calculations.
COEF1.FOR – calculates COEFficients for A and X matrices using (2.21–2.24) via rapidly converging series method outlined in Appendix A.
SOLV.FOR – SOLVes matrix equation $A \bullet \mathcal{V} = X$ using gaussian elimination.

ADMIT1.FOR – computes complex ADMITtances Y_{11} and Y_{12} using (2.27).
BODEX1.FOR – converts complex voltage ratio from cartesian to polar form (gain–phase).

COEF1a.FOR – calculates COEFficients for A and X matrices using (2.21–2.24) directly.

For electrode structures having a multilayered medium above and below use:

GAIN2.FOR – Calls the subroutines listed below for the actual calculations.
COEF2.FOR – calculates COEFficients for A and X matrices using (2.21–2.24) via rapidly converging series method outlined in Appendix A.

SOLV.FOR – SOLVes matrix equation $A \bullet \mathcal{V} = X$ using gaussian elimination.

ADMIT2.FOR – computes complex ADMITtances Y_{11} and Y_{12} using (2.27).

BODEX2.FOR – converts complex voltage ratio from cartesian to polar form (gain–phase).

Auxiliary Subprograms

These subprograms are used for: 1) to get the parameters describing the electrode array structure (chip), 2) to get the parameters for computing various types of surface capacitance densities, 3) to calculate the various types of surface capacitance densities, 4) to perform the necessary matrix manipulations, and 5) to generate an input file for a discretized exponential or linear distribution in complex bulk permittivity.

CFO.FOR – gets Chip inFOrmation on electrode structure used: spatial wavelength, oxide layer thickness and permittivity, and load capacitance.
HCFO.FOR – gets Heterogeneous Chip inFOrmation on electrode structure used for variable spatial wavelength estimation: spatial wavelength, and oxide layer thickness and permittivity.

For homogeneous layers only above electrodes use:

SCFO1.FOR – gets Surface Capacitance inFOrmation such as number of layers and their complex bulk and surface permittivities for layered media above electrodes.
SCAP1.FOR – Surface CAPacitance computation using transfer relations for homogeneous ohmic layers (Appendix B).

For exponential layers only above electrodes use:

SCFO1.FOR – gets Surface Capacitance inFOrmation such as number of layers and their complex bulk and surface permittivities for layered media above electrodes.
SCAP1a.FOR – Surface CAPacitance computation using transfer relations for exponential spatial variation in complex bulk permittivity (see [75, p. 2.53]).

For homogeneous layers above and below electrodes use:

SCFO2.FOR – gets Surface Capacitance inFOrmation such as number of layers and their complex bulk and surface permittivities for ohmic, layered media above and below electrodes.
SCAP2.FOR – Surface CAPacitance computation using transfer relations for homogeneous ohmic layers above and below electrodes.

For Lorentz sphere model use:

SCFO3.FOR – gets Surface Capacitance inFOrmation for dispersive medium using Lorentz sphere model.
SCAP3.FOR – Surface CAPacitance computation using transfer relations for dispersive medium using Lorentz sphere model.

For power law dispersion use:

SCFO4.FOR – gets Surface Capacitance inFOrmation for medium displaying power law dispersion.
SCAP4.FOR – Surface CAPacitance computation using transfer relations for medium displaying power law dispersion.

For estimation routines using homogeneous layers above electrodes:

SCAPEST1.FOR – Surface CAPacitance computation for ESTimation routines using homogeneous, ohmic layer transfer relations, no normalization performed in this routine.

For estimation routines using exponential layers above electrodes:

SCAPEST1a.FOR – Surface CAPacitance computation for ESTimation routines using exponential layer transfer relations, no normalization performed in this routine.

For matrix manipulations use:

MATMPY1.FOR – MATrix MultiPlication using complex arrays. DISCON1.FOR and DISCON2.FOR.
MINV1.FOR – MATrix Inversion of complex arrays using gaussian elimination. For DISCON1.FOR and DISCON2.FOR.

LAYGEN.FOR – generates input file for a discretized exponential or linear distribution in complex bulk permittivity when computing a gain–phase response.

Table F.1: Directory of Sample Input Files and Program Listings

Program	Page Number: Sample Input File	Page Number: Listing
MZ1.FOR	434	445
MZ2.FOR	435	449
MZ3.FOR	436	453
MZ4.FOR	437	457
THEST.FOR	438	461
PAREST.FOR	439	465
HETEST.FOR	440	476
DISCON1.FOR	441	489
DISCON2.FOR	442	499
TH4EST.FOR	443	510
HEST.FOR	444	516
GAIN1.FOR	–	520
COEF1.FOR	–	522
SOLV.FOR	–	528
ADMIT1.FOR	–	530
BODEX1.FOR	–	533
COEF1A.FOR	–	534
GAIN2.FOR	–	537
COEF2.FOR	–	539
ADMIT2.FOR	–	546
BODEX2.FOR	–	549
CFO.FOR	–	550
HCFO.FOR	–	551
SCFO1.FOR	–	552
SCAP1.FOR	–	553
SCAP1A.FOR	–	555
SCFO2.FOR	–	558
SCAP2.FOR	–	560
SCFO3.FOR	–	563
SCAP3.FOR	–	565
SCFO4.FOR	–	568
SCAP4.FOR	–	570
SCAPEST1.FOR	–	572
SCAPEST1A.FOR	–	574
MATMPY1.FOR	–	577
MINV1.FOR	–	578
LAYGEN.FOR	–	582

Table F.2: Definition of Symbols used in Input Files

Symbol	Definition
a	interelectrode spacing (meters)
α	lambda extension (see eq. (4.26))
B	proportionality constant for power law dispersion (see (8.8)
$\underline{C}_\ell$	normalized load capacitance, $C_\ell/\epsilon_{ox}M_{el}$
d_i	thickness of layer i above electrodes (meters)
d_{ib}	thickness of layer i below electrodes (meters)
ϵ_a, σ_a	bulk permittivity (F/m) and conductivity (S/m) immediately above electrodes
ϵ_b, σ_b	bulk permittivity (F/m) and conductivity (S/m) immediately below electrodes
$\epsilon_{bu}, \sigma_{bu}$	bulk permittivity (F/m) and conductivity (S/m) of dense phase for Lorentz sphere model
ϵ_i, σ_i	bulk permittivity (F/m) and conductivity (S/m) of layer i above electrodes
$\epsilon_{ib}, \sigma_{ib}$	bulk permittivity (F/m) and conductivity (S/m) of layer i below electrodes
ϵ_{ox}	oxide layer permittivity (F/m)
ϵ_s, σ_s	surface permittivity (F) and conductivity (S) on particle surface for Lorentz sphere model
$\epsilon_{si}, \sigma_{si}$	surface permittivity (F) and conductivity (S) of layer i above electrodes
$\epsilon_{sib}, \sigma_{sib}$	surface permittivity (F) and conductivity (S) of layer i below electrodes
$\epsilon_{so}, \sigma_{so}$	surface permittivity (F) and conductivity (S) between electrodes
ϵ_{st}	stationary dielectric constant for power law dispersion
f_m	measured frequency (Hz)
ϕ	particle volume fraction for Lorentz sphere model
g_m	measured gain (dB)
$gtol$	gain error tolerance (dB)
h	oxide layer thickness (meters)
k	number of collocation points
λ	spatial wavelength (meters)
M_{el}	electrode meander length (meters)
N2	number of fourier modes
n_p	exponent for power law dispersion
num	number of layers above electrodes
numb	number of layers below electrodes
p_m	measured phase (deg.)
$ptol$	phase error tolerance (deg.)
R	particle radius in Lorentz sphere model (meters)
$scale$	increment for next guess when computing local derivative
ω_p	temporal (angular) frequency of phase peak for power law dispersion

MZ1.FOR

Link using

cc -Ml -o mz1 mz1.o cfo.o scfo1.o scap1.o gain1.o coef1.o solv.o admit1.o bodex1.o -lf

Sample Input File (IN16A) – Sample file for computing response of Fig. 8.7, ($\sigma_{s2} = 10^{-15}$ S). Let $h = 10$ μm and normalized load capacitance be 10 (not actual values). Place output in test1.

Input File	Description	Field (Fortran)
test1	output filename	10A
100	N2	I4
25	k	I4
5.00E-5	λ	E9.3
3.45E-11	ϵ_{ox}	E9.3
1.25	$\lambda/4h$	E9.0
.25	a/λ	E9.0
10.0	$\underline{C}_\ell$	E9.0
2.95E-11 5.00E-10	ϵ_a, σ_a	2E8.0
0 0	ϵ_{so}, σ_{so}	2E8.0
2	num	I3
1000	d_1	E9.0
2.00E-11 1.00E-12	ϵ_1, σ_1	2E8.0
0 0	ϵ_{s1}, σ_{s1}	2E8.0
2.2E-6	d_2	E9.0
2.95E-11 5.00E-10	ϵ_2, σ_2	2E8.0
0 1.00E-15	ϵ_{s2}, σ_{s2}	2E8.0
y	frequency range option	10A

MZ2.FOR

Link using

cc -Ml -o mz2 mz2.o cfo.o scfo2.o scap2.o gain2.o coef1.o solv.o admit2.o bodex2.o -lf

Sample Input File (IN16B) – Sample file for computing response having same medium above electrodes as before but also having two layers below the electrodes plus a ground plane. A ground plane is simulated using a large surface conductivity on layer 2 below. Let $h = d_{1b} = 10$ μm and normalized load capacitance be 10 (not actual values). Place output in test1.

Input File	Description	Field (Fortran)
test1	output filename	10A
100	N2	I4
25	k	I4
5.00E-5	λ	E9.3
3.45E-11	ϵ_{ox}	E9.3
1.25	$\lambda/4h$	E9.0
.25	a/λ	E9.0
10.0	$\underline{C}_\ell$	E9.0
2.95E-11 5.00E-10	ϵ_a, σ_a	2E8.0
0 0	ϵ_{so}, σ_{so}	2E8.0
3.95E-11 5.00E-12	ϵ_b, σ_b	2E8.0
2	num	I3
1000	d_1	E9.0
2.00E-11 1.00E-12	ϵ_1, σ_1	2E8.0
0 0	ϵ_{s1}, σ_{s1}	2E8.0
2.2E-6	d_2	E9.0
2.95E-11 5.00E-10	ϵ_2, σ_2	2E8.0
0 1.00E-15	ϵ_{s2}, σ_{s2}	2E8.0
3	numb	I3
1000	d_{1b}	E9.0
0 0	ϵ_{1b}, σ_{1b}	2E8.0
0 0	ϵ_{s1b}, σ_{s1b}	2E8.0
2.2E-6	d_{2b}	E9.0
2.00E-11 1.00E-10	ϵ_{2b}, σ_{2b}	2E8.0
0 1	ϵ_{s2b}, σ_{s2b}	2E8.0
3.0E-6	d_{3b}	E9.0
3.45E-11 0	ϵ_{3b}, σ_{3b}	2E8.0
0 0	ϵ_{s3b}, σ_{s3b}	2E8.0
y	frequency range option	10A

mz3.for

link using

cc -ml -o mz3 mz3.o cfo.o scfo3.o scap3.o gain1.o coef1.o solv.o admit1.o bodex1.o -lf

sample input file (in16c) - sample file for computing response of fig. 8.10, ($\sigma_p = 10^{-7}$ s/m). let $h = 10$ μm and normalized load capacitance be 10 (not actual values). place output in test1. note - the routine is set to calculate the response of a two layer problem with the layer next to the electrodes having the dispersion.

input file	description	field (fortran)
test1	output filename	10a
100	n2	i4
25	k	i4
5.00e-5	λ	e9.3
3.45e-11	ϵ_{ox}	e9.3
1.25	$\lambda/4h$	e9.0
.25	a/λ	e9.0
10.0	c_ℓ	e9.0
2.60e-111.00e-10	ϵ_a, σ_a	2e8.0
0 0	ϵ_{so}, σ_{so}	2e8.0
2	num	i3
1000	d_1	e9.0
2.00e-111.00e-12	ϵ_1, σ_1	2e8.0
0 0	ϵ_{s1}, σ_{s1}	2e8.0
6.8e-6	d_2	e9.0
2.60e-111.00e-10	ϵ_2, σ_2	2e8.0
0 0	ϵ_{s2}, σ_{s2}	2e8.0
2.60e-111.00e-10	ϵ_{bu}, σ_{bu}	2e8.0
2.60e-111.00e-7	ϵ_p, σ_p	2e8.0
0 0	ϵ_s/r, σ_s/r	2e8.0
.4	ϕ	e9.0
y	frequency range option	10a

mz4.for

link using

cc -ml -o mz4 mz4.o cfo.o scfo4.o scap4.o gain1.o coef1.o solv.o admit1.o bodex1.o -lf

sample input file (in16d) – sample file for computing response of a layer displaying a power law dispersion as denoted in (eq:pow). let $h = 10$ μm and normalized load capacitance be 10 (not actual values). place output in test1. note – the routine is set to calculate the response of a two layer problem with the layer next to the electrodes having the dispersion.

input file	description	field (fortran)
test1	output filename	10a
100	n2	i4
25	k	i4
5.00e-5	λ	e9.3
3.45e-11	ϵ_{ox}	e9.3
1.25	$\lambda/4h$	e9.0
.25	a/λ	e9.0
10.0	$\underline{c}_\ell$	e9.0
2.95e-110	ϵ_a, σ_a	2e8.0
0 0	ϵ_{so}, σ_{so}	2e8.0
2	num	i3
1000	d_1	e9.0
2.00e-111.00e-12	ϵ_1, σ_1	2e8.0
0 0	ϵ_{s1}, σ_{s1}	2e8.0
2.2e-6	d_2	e9.0
2.95e-110	ϵ_2, σ_2	2e8.0
0 0	ϵ_{s2}, σ_{s2}	2e8.0
2.95e-11	ϵ_{st}	e9.0
.25	np	e9.0
1.00e-9	b	e9.0
628.3	ω_p	e9.0
y	frequency range option	10a

thest.for

link using

cc -ml -o thest thest.o cfo.o scapest1.o gain1.o coef1.o solv.o admit1.o bodex1.o -lf

sample input file (inth1) – sample file for estimating thickness of a layer having a bulk permittivity $2x10^{-11}$ f/m using a microchip in air giving a high frequency gain of –41.46 db. let $h = 10$ μm and normalized load capacitance be 10 (not actual values). place output in th1. note – program assumes only two layers above electrodes.

input file	description	field (fortran)
th1	output filename	10a
100	n2	i4
5.00e-5	λ	e9.3
3.45e-11	ϵ_{ox}	e9.3
1.25	$\lambda/4h$	e9.0
.25	a/λ	e9.0
10.0	c_ℓ	e9.0
8.85e-12	ϵ_1	e8.0
2.00e-11	ϵ_2	e8.0
-41.46	g_m, in air	

parest.for

link using
cc -ml -o parest parest.o cfo.o scapest1.o gain1.o coef1.o solv.o admit1.o bodex1.o -lf

sample input file (inpar) – sample file for estimating complex bulk permittivity of a 5 μ thick layer in a surrounding medium of complex bulk permittivity $\epsilon_1 = 2x10^{-11}$ f/m and $\sigma_1 = 10^{-13}$ s/m using a microchip. estimate using a measured gain–phase of –18.00 db and –56.12° at 2.5 hz. let $h = 10$ μm and normalized load capacitance be 10 (not actual values). place output in pz1. use variable collocation point technique.

input file	description	field (fortran)
pz1	output filename	10a
100	n2	i4
y	use variable collocation pts.?	10a
5.00e-5	λ	e9.3
3.45e-11	ϵ_{ox}	e9.3
1.25	$\lambda/4h$	e9.0
.25	a/λ	e9.0
10.0	c_ℓ	e9.0
b	bulk or surface property?	10a
2	layer number for estimation	i3
2	num	i3
1000	d_1	e9.0
2.00e-111.00e-13	ϵ_1, σ_1	2e8.0
0 0	ϵ_{s1}, σ_{s1}	2e8.0
5.0e-6	d_2	e9.0
0 0	ϵ_{s2}, σ_{s2}	2e8.0
0 0	ϵ_{so}, σ_{so}	2e8.0
-18.00-56.12 2.5	g_m, p_m and f_m	e6.2, e7.2, e9.3
.05	*gtol*	e9.0
.10	*ptol*	e9.0
n	input guess?	10a

hetest.for

link using

cc -ml -o hetest hetest.o hcfo.o scapest1.o gain1.o coef1.o solv.o admit1.o bodex1.o -lf

sample input file (hin) – sample file for case 2, table 4.3 ($d_2 = 50$ μm). use 10 refinement iterations.

input file	description	field (fortran)
try20	output filename	10a
100	n2	i4
y	use variable collocation pts.?	10a
3.45e-11	ϵ_{ox}	e9.3
1.0e-5	h	e9.0
6	number of spatial wavelengths	i3
.10	f_m	e9.0
-45.30 -71.25768.78 2.000e-5	g_{m1}, p_{m1}, $\varsigma_{\ell 1}$, λ_1	e6.2, e7.2, e6.2, e9.3
-37.53 -80.66165.84 1.000e-4	g_{m2}, p_{m2}, $\varsigma_{\ell 2}$, λ_2	e6.2, e7.2, e6.2, e9.3
-39.77 -87.56 90.50 2.000e-4	g_{m3}, p_{m3}, $\varsigma_{\ell 3}$, λ_3	e6.2, e7.2, e6.2, e9.3
-49.65 -57.92 63.11 4.000e-4	g_{m4}, p_{m4}, $\varsigma_{\ell 4}$, λ_4	e6.2, e7.2, e6.2, e9.3
-51.02 -19.12 43.45 8.000e-4	g_{m5}, p_{m5}, $\varsigma_{\ell 5}$, λ_5	e6.2, e7.2, e6.2, e9.3
-41.81 -94.04 36.62 1.000e-3	g_{m6}, p_{m6}, $\varsigma_{\ell 6}$, λ_6	e6.2, e7.2, e6.2, e9.3
.25	α	e9.0
.05	*gtol*	e9.0
.10	*ptol*	e9.0
y	refinements?	10a
verb+10.0+	number of refinements	i3
n	input guess?	10a

discon1.for

link using

cc -ml -o discon1 discon1.o hcfo.o scapest1.o gain1.o coef1.o solv.o admit1.o bodex1.o minv1.o matmpy1.o -lf

sample input file (indis1) – sample file for case 4, table 4.6 ($d_2 = 50\ \mu$m).

input file	description	field (fortran)
try	output filename	10a
100	n2	i4
25	k	10a
3.45e-11	ϵ_{ox}	e9.3
1.0e-5	h	e9.0
.25	a/λ	e9.0
2	number of variables	i3
2	number of datapoints	i3
1.0	f_m	e9.0
1.02	*scale*	e9.0
2.00e-111.00e-12	ϵ_2, σ_2	2e8.0
-50.94 -15.79 63.11 4.000e-4	g_{m1}, p_{m1}, $\subseteq_{\ell 1}$, λ_1	e6.2, e7.2, e6.2, e9.3
-50.26 -32.11 36.62 1.000e-3	g_{m2}, p_{m2}, $\subseteq_{\ell 2}$, λ_2	e6.2, e7.2, e6.2, e9.3
.05	*gtol*	e9.0
.10	*ptol*	e9.0
n	input guess?	10a

discon2.for

link using

cc -ml -o discon2 discon2.o hcfo.o scapest1.o gain1.o coef1.o solv.o admit1.o bodex1.o minv1.o matmpy1.o -lf

sample input file (indis2) – sample file for case 5, table 4.7 ($\sigma_{s2} = 10^{-16}$ s).

input file	description	field (fortran)
try	output filename	10a
100	n2	i4
25	k	10a
3.45e-11	ϵ_{ox}	e9.3
1.0e-5	h	e9.0
.25	a/λ	e9.0
2	number of variables	i3
2	number of datapoints	i3
.01	f_m	e9.0
1.02	*scale*	e9.0
5.0e-6	d_2	e9.0
2.00e-111.00e-10	ϵ_2, σ_2	2e8.0
-8.96 -74.18 63.11 4.000e-4	g_{m1}, p_{m1}, $\varsigma_{\ell 1}$, λ_1	e6.2, e7.2, e6.2, e9.3
-15.02-104.40 36.62 1.000e-3	g_{m2}, p_{m2}, $\varsigma_{\ell 2}$, λ_2	e6.2, e7.2, e6.2, e9.3
.05	*gtol*	e9.0
.10	*ptol*	e9.0
n	input guess?	10a

th4est.for

link using

cc -ml -o th4est th4est.o cfo.o scapest1.o gain1.o coef1.o solv.o admit1.o bodex1.o matmpy.o minv.o -lf

sample input file (inth41) – sample file for estimating thickness and permittivity of a layer using a microchip. in air the high frequency gain is -39.00 db and in oil ($\epsilon_1 = 2.0x10^{-11}$ f/m) the high frequency gain is -31.00 db. let $h = 10$ μm and normalized load capacitance be 10 (not actual values). place output in th41.

input file	description	field (fortran)
th41	output filename	10a
100	n2	i4
y	use variable collocation pts?	10a
5.00e-5	λ	e9.3
3.45e-11	ϵ_{ox}	e9.3
1.25	$\lambda/4h$	e9.0
.25	a/λ	e9.0
10.0	$\underline{c}_\ell$	e9.0
2	number of data points?	i3
8.85e-12	ϵ_1	e8.0
-39.00	g_m, in air	
2.00e-11	ϵ_2	e8.0
-31.00	g_m, in oil	
n	input guess?	10a

hest.for

link using
cc -ml -o hest hest.o cfo.o scapest1.o gain1.o coef1.o solv.o admit1.o bodex1.o -lf

sample input file (inh1) – sample file for estimating oxide layer thickness of a microchip having a high frequency gain of –41.46 db in air. let the normalized load capacitance be 10 (not actual value). place output in h1. automatically uses 25 collocation points.

input file	description	field (fortran)
h1	output filename	10a
100	n2	i4
5.00e-5	λ	e9.3
3.45e-11	ϵ_{ox}	e9.3
1.25	$\lambda/4h$	e9.0
.25	a/λ	e9.0
10.0	$\underline{c}_\ell$	e9.0
-41.46	g_m, in air	

```
c                mz1.for
c
c     this program computes gain-phase response over a specified
c     frequency range given a set of complex permittivities, layer
c     thicknesses, chip dimensions etc. using continuum model
c
c
        program mz1

        parameter (pi=3.141592654)
        complex a(50,50),x(50),v(50),l(1000),lmin,y11,y12,yl
        complex e(100),se(100),re(100),rse(100),c(1000)
        complex ea,sea,gp(60),rea,rsea
        real i1,index(60),inc,freq(60)
        real lam,eox,yload,h,g,en0,d(100),w1
        character q1,q2,ofile*10
        integer f,n1,n2
c       always need these guys
        common /comp/rea,rsea
        common /comp1/lmin,y11,y12,yl
        common /flot/h,g,eox,lam,yload
        common /flot1/en0
        common /tin1/n1,n2
c       need for homogeneous layers
        common /comp2/e,se,re,rse
        common /flot2/w1
        common /tin2/num
c       need for numerical aspect
        common /tin/k,n,nmax
c       max. number of collocation pts.
        n1=50
c       no. of summation terms (fourier modes)
        n2=1000
        print*,'output filename? '
        read 10,ofile
   10       format(10a)
        open(unit=3,file=ofile)
c
c       get information about numerical aspect
c
        print*,'input # of summation (fourier) terms '
        read 5,n
   5       format(i4)
        print*,'input # of collocation pnts. '
        read 5,k
c
c         get information about chip
c
```

```
        call chipinfo()
c
c       get information about electrical properties just above
c       interface
c
c       print*,'input normalized ambient dc field (e0*lambda/v0) '
c       read 15,e0
   15         format(e9.0)
c       e0 represents dc field
        e0=0.0
c       en0 represents ac field with no spatial periodicity
        en0=0.0
        print*,'input complex bulk permittivity above electrodes (ea) '
        read 21,ea
  21        format(2e8.0)
        print*,ea
        print*,'input complex surface permittivity at electrodes (sea) '
        read 21,sea
        print*,sea
c
c       get information necessary for computing the surface capacitance
c       density, cn
c
        call scapinfo1(d)
c
c     write the relevant data into the output file
c
        write (3,160) k,n
  160   format ('# of pts. = ',i3,2x,' # of summation terms = ',i4)
c       write (3,163) e0
c 163        format ('ambient dc field e0*lambda/v0 = ',f3.2)
        write (3,103) ea
  103        format ('bulk permittivity above electrodes = ',2e8.3)
        write (3,107) sea
  107        format ('surface permittivity above electrodes = ',2e8.3)
c
c       set up appropriate scan sequence
c
        print*,'freq. scan from .005 - 10000 hz? (y or n) '
        read 10,q1
        if(q1.eq.'y') then
                imin=-3
                imax=4
                jmax=8
                index(1)=5.0
                index(2)=6.0
                index(3)=8.0
                index(4)=10.0
```

```
                index(5)=15.0
                index(6)=20.0
                index(7)=25.0
                index(8)=30.0
                index(9)=40.0
        else
                print*,'freq. scan or just one freq? (s or o) '
                read 10,q2
                if(q2.eq.'o') then
                        imin=-1
                        imax=-1
                        jmax=0
                        index(1)=1.0
                else
                        print*,'input minimum frequency '
                        read 15,fmin
                        print*,'input maximum frequency '
                        read 15,fmax
                        print*,'input increment '
                        read 15,inc
                        imin=0
                        imax=0
                        jmax=(fmax-fmin)/inc
                        do 110 j=0,jmax
                         index(j+1)=fmin+j*inc
  110                        continue
                end if
        end if
c
c       compute gain at requested frequencies
c
c
        yl=cmplx(yload,0.0)
        do 120 i=imin,imax
        i1=1.0*i
           do 130 j=1,jmax+1
           f=9*(i+3)+j
           if((f.le.58).and.(jmax.lt.61)) then
                freq(f)=index(j)*10**i1
                print*,'freq=',freq(f)
                write (3,172) freq(f)
  172                format('freq=',f10.4)
                w1=2*pi*freq(f)
c
c     normalize the permittivity and conductivity
c
                rea=ea/eox
                  rsea=sea/(eox*lam)
```

```
                rea=cmplx(real(rea),-aimag(rea)/w1)
                  rsea=cmplx(real(rsea),-aimag(rsea)/w1)
c
c       get surface capacitance density above electrodes, cn
c
                call scap1(d,c)
c
c       compute gain
c
                call gain1(c,a,x,v,l,gp(f))
           end if
 130            continue
 120         continue
       write (3,170)
 170       format (' frequency',tr12,' gain',tr12,' phase')
       write (3,173)
 173       format(' $')
       do 175 i=1,58
       write (3,171) freq(i),real(gp(i)),aimag(gp(i))
 171       format(f10.4,tr12,f8.3,tr16,f8.3)
 175       continue
       write (3,173)
       close(3)
       end
```

```
c                mz2.for
c
c     this program computes gain-phase response over a specified
c     frequency range given a set of complex permittivities, layer
c     thicknesses, chip dimensions etc.
c     allows for layers above and below electrodes
c
        program mz2

        parameter (pi=3.141592654)
        complex a(50,50),x(50),v(50),l(1000),lmin,y11,y12,yl
        complex e(100,2),se(100,2),re(100,2),rse(100,2)
        complex c(1000,2),c0(2)
        complex el(2),sel,gp(60),rel(2),rsel
        real i1,index(60),inc,freq(60)
        real lam,eox,yload,h,g,vn0(2),d(100,2),w1
        character q1,q2,ofile*10
        integer f,num(2),n1,n2
c       always need these guys
        common /comp/rel,rsel
        common /comp1/lmin,y11,y12,yl,c0
        common /flot/h,g,eox,lam,yload
        common /flot1/vn0
        common /tin1/n1,n2
c       need for homogeneous layers
        common /comp2/e,se,re,rse
        common /flot2/w1
        common /tin2/num
c       need for numerical aspect
        common /tin/k,n,nmax
c       max. number of collocation pts.
        n1=50
c       no. of summation terms (fourier modes)
        n2=1000
        print*,'output filename? '
        read 10,ofile
   10       format(10a)
        open(unit=3,file=ofile)
c
c       get information about numerical aspect
c
        print*,'input # of summation (fourier) terms '
        read 5,n
    5      format(i4)
        print*,'input # of collocation pnts. '
        read 5,k
c
c         get information about chip
```

```
c
        call chipinfo()
c
c       get information about electrical properties just above
c       interface
c
c       print*,'input normalized ambient dc field above (e0*lambda/v0) '
c       read 15,e0
   15         format(e9.0)
c       e0 represents dc field
        e0=0.0
c
c       no ambient ac field above either
c
        vn0(1)=0.0
        print*,'input complex bulk permittivity above electrodes
     x              (el(1)) '
        read 21,el(1)
  21        format(2e8.0)
        print*,el(1)
        print*,'input complex surface permittivity at electrodes (sel) '
        read 21,sel
        print*,sel
c
c       get information about electrical properties just below
c       interface
c
c       ambient ac field below electrodes uses potential at
c       driven electrode so
c
        vn0(2)=1.0
c
        print*,'input complex bulk permittivity below electrodes
     x       (el(2)) '
        read 21,el(2)
        print*,el(2)
c
c       get information necessary for computing the surface
c       capacitance density, cn, above and below electrodes
c
        call scapinfo2(d)
c
c     write the relevant data into the output file
c
        write (3,160) k,n
  160   format ('# of pts. = ',i3,2x,' # of summation terms = ',i4)
c       write (3,163) e0
c 163         format ('ambient dc field e0*lambda/v0 = ',f3.2)
```

```
      write (3,103) el(1)
  103      format ('bulk permittivity above electrodes = ',2e8.3)
      write (3,703) el(2)
  703      format ('bulk permittivity below electrodes = ',2e8.3)
      write (3,107) sel
  107      format ('surface permittivity at electrodes = ',2e8.3)
c
c     set up appropriate scan sequence
c
      print*,'freq. scan from .005 - 10000 hz? (y or n) '
      read 10,q1
      if(q1.eq.'y') then
              imin=-3
              imax=4
              jmax=8
              index(1)=5.0
              index(2)=6.0
              index(3)=8.0
              index(4)=10.0
              index(5)=15.0
              index(6)=20.0
              index(7)=25.0
              index(8)=30.0
              index(9)=40.0
      else
              print*,'freq. scan or just one freq? (s or o) '
              read 10,q2
              if(q2.eq.'o') then
                      imin=-1
                      imax=-1
                      jmax=0
                      index(1)=1.0
              else
                      print*,'input minimum frequency '
                      read 15,fmin
                      print*,'input maximum frequency '
                      read 15,fmax
                      print*,'input increment '
                      read 15,inc
                      imin=0
                      imax=0
                      jmax=(fmax-fmin)/inc
                      do 110 j=0,jmax
                       index(j+1)=fmin+j*inc
  110                      continue
              end if
      end if
c
```

```
c       compute gain at requested frequencies
c
c
        yl=cmplx(yload,0.0)
        do 120 i=imin,imax
        i1=1.0*i
           do 130 j=1,jmax+1
           f=9*(i+3)+j
           if((f.le.58).and.(jmax.lt.61)) then
                freq(f)=index(j)*10**i1
                print*,'freq=',freq(f)
                write (3,172) freq(f)
 172                format('freq=',f10.4)
                w1=2*pi*freq(f)
c
c     normalize the permittivity and conductivity
c
                do 717 m=1,2
                rel(m)=el(m)/eox
                rel(m)=cmplx(real(rel(m)),-aimag(rel(m))/w1)
 717                continue
                  rsel=sel/(eox*lam)
                  rsel=cmplx(real(rsel),-aimag(rsel)/w1)
c
c       get surface capacitance density above and below electrodes, cn
c
                call scap2(d,c)
c
c       compute gain
c
                call gain2(c,a,x,v,1,gp(f))
           end if
 130            continue
 120        continue
        write (3,170)
 170        format (' frequency',tr12,' gain',tr12,' phase')
        write (3,173)
 173        format(' $')
        do 175 i=1,58
        write (3,171) freq(i),real(gp(i)),aimag(gp(i))
 171        format(f10.4,tr12,f8.3,tr16,f8.3)
 175        continue
        write (3,173)
        end
```

```
c               mz3.for
c
c     this program computes gain-phase response over a specified
c     frequency range given a layer with a dispersive complex
c     permittivity of some thickness, chip dimensions etc.
c     uses Lorentz sphere dispersion
c
      program mz3

      parameter (PI=3.141592654)
      complex A(50,50),X(50),V(50),L(1000),Lmin,y11,y12,yl
      complex e(100),se(100),re(100),rse(100),C(1000)
      complex ea,sea,gp(60),rea,rsea,esR
      real i1,index(60),inc,freq(60)
      real LAM,eox,YLOAD,h,g,En0,d(100),w1,phi
      character q1,q2,ofile*10
      integer f,N1,N2
c     always need these guys
      common /comp/rea,rsea
      common /comp1/Lmin,y11,y12,yl
      common /flot/h,g,eox,LAM,YLOAD
      common /flot1/En0
      common /tin1/N1,N2
c     need for homogeneous layers
      common /comp2/e,se,re,rse
      common /flot2/w1
      common /tin2/num
c     need for numerical aspect
      common /tin/k,N,nmax
c     need for dispersive media
      common /flot3/phi,esR
c     max. number of collocation pts.
      N1=50
c     no. of summation terms (fourier modes)
      N2=1000
      print*,'output filename? '
      read 10,ofile
 10       format(10a)
      open(unit=3,file=ofile)
c
c     get information about numerical aspect
c
      print*,'input # of summation (fourier) terms '
      read 5,N
 5        format(I4)
      print*,'input # of collocation pnts. '
      read 5,k
c
```

```
c         get information about chip
c
        call chipinfo()
c
c       get information about electrical properties just above
c       interface
c
c       print*,'input normalized ambient DC field (E0*lambda/V0) '
c       read 15,E0
   15         format(E9.0)
c       E0 represents DC field
        E0=0.0
c       En0 represents AC field with no spatial pe iodicity
        En0=0.0
        print*,'input complex bulk permittivity above electrodes (ea) '
        read 21,ea
  21        format(2E8.0)
        print*,ea
        print*,'input complex surface permittivity at electrodes (sea) '
        read 21,sea
        print*,sea
c
c       get information necessary for computing the surface capacitance
c       density, Cn
c
        call scapinfo3(d)
c
c     write the relevant data into the output file
c
        write (3,160) k,N
  160   format ('# of pts. = ',I3,2x,' # of summation terms = ',I4)
c       write (3,163) E0
c 163        format ('ambient DC field E0*lambda/V0 = ',F3.2)
        write (3,103) ea
  103        format ('bulk permittivity above electrodes = ',2E8.3)
        write (3,107) sea
  107        format ('surface permittivity above electrodes = ',2E8.3)
c
c       set up appropriate scan sequence
c
        print*,'freq. scan from .005 - 10000 Hz? (y or n) '
        read 10,q1
        if(q1.eq.'y') then
                imin=-3
                imax=4
                jmax=8
                index(1)=5.0
                index(2)=6.0
```

```
                index(3)=8.0
                index(4)=10.0
                index(5)=15.0
                index(6)=20.0
                index(7)=25.0
                index(8)=30.0
                index(9)=40.0
        else
                print*,'freq. scan or just one freq? (s or o) '
                read 10,q2
                if(q2.eq.'o') then
                        imin=-1
                        imax=-1
                        jmax=0
                        index(1)=1.0
                else
                        print*,'input minimum frequency '
                        read 15,fmin
                        print*,'input maximum frequency '
                        read 15,fmax
                        print*,'input increment '
                        read 15,inc
                        imin=0
                        imax=0
                        jmax=(fmax-fmin)/inc
                        do 110 j=0,jmax
                         index(j+1)=fmin+j*inc
  110                        continue
                end if
        end if
c
c       compute gain at requested frequencies
c
c
        yl=cmplx(YLOAD,0.0)
        do i20 i=imin,imax
        i1=1.0*i
           do 130 j=1,jmax+1
           f=9*(i+3)+j
           if((f.le.58).and.(jmax.lt.61)) then
                freq(f)=index(j)*10**i1
                print*,'freq=',freq(f)
                write (3,172) freq(f)
  172                format('freq=',F10.4)
                w1=2*PI*freq(f)
c
c     normalize the surface permittivity and conductivity
c
```

```
                rsea=sea/(eox*LAM)
                rsea=cmplx(real(rsea),-aimag(rsea)/w1)
c
c       get surface capacitance density above electrodes, Cn
c
                call scap3(d,C)
                rea=re(2)
                print*,'e(3) = ',eox*real(re(3)),-w1*eox*aimag(re(3))
                print*,'e(4) = ',eox*real(re(4)),-w1*eox*aimag(re(4))
                print*,'e(2) = ',eox*real(re(2)),-w1*eox*aimag(re(2))
c
c       compute gain
c
                call gain1(C,A,X,V,L,gp(f))
           end if
 130            continue
 120         continue
        write (3,170)
 170        format (' frequency',TR12,' gain',TR12,' phase')
        write (3,173)
 173        format(' $')
        do 175 i=1,58
        write (3,171) freq(i),real(gp(i)),aimag(gp(i))
 171        format(F10.4,TR12,F8.3,TR16,F8.3)
 175        continue
        write (3,173)
        end
```

```
c                mz4.for
c
c     this program computes gain-phase response over a specified
c     frequency range given a layer with a dispersive complex
c     permittivity of some thickness, chip dimensions etc.
c     uses power law dispersion
c
        program mz4

        parameter (PI=3.141592654)
        complex A(50,50),X(50),V(50),L(1000),Lmin,y11,y12,yl
        complex e(100),se(100),re(100),rse(100),C(1000)
        complex ea,sea,gp(60),rea,rsea
        real i1,index(60),inc,freq(60)
        real LAM,eox,YLOAD,h,g,En0,d(100),w1,est
        real B,np,wp
        character q1,q2,ofile*10
        integer f,N1,N2
c       always need these guys
        common /comp/rea,rsea
        common /comp1/Lmin,y11,y12,yl
        common /flot/h,g,eox,LAM,YLOAD
        common /flot1/En0
        common /tin1/N1,N2
c       need for homogeneous layers
        common /comp2/e,se,re,rse
        common /flot2/w1
        common /tin2/num
c       need for numerical aspect
        common /tin/k,N,nmax
c       need for dispersive media
        common /flot3/np,B,est,wp
c       max. number of collocation pts.
        N1=50
c       no. of summation terms (fourier modes)
        N2=1000
        print*,'output filename? '
        read 10,ofile
   10       format(10a)
        open(unit=3,file=ofile)
c
c       get information about numerical aspect
c
        print*,'input # of summation (fourier) terms '
        read 5,N
   5        format(I4)
        print*,'input # of collocation pnts. '
        read 5,k
```

```
c
c         get information about chip
c
        call chipinfo()
c
c       get information about electrical properties just above
c       interface
c
c       print*,'input normalized ambient DC field (E0*lambda/V0) '
c       read 15,E0
   15         format(E9.0)
c       E0 represents DC field
        E0=0.0
c       En0 represents AC field with no spatial periodicity
        En0=0.0
        print*,'input complex bulk permittivity above electrodes (ea) '
        read 21,ea
  21        format(2E8.0)
        print*,ea
        print*,'input complex surface permittivity at electrodes (sea) '
        read 21,sea
        print*,sea
c
c       get information necessary for computing the surface capacitance
c       density, Cn
c
        call scapinfo4(d)
c
c     write the relevant data into the output file
c
        write (3,160) k,N
  160   format ('# of pts. = ',I3,2x,' # of summation terms = ',I4)
c       write (3,163) E0
c 163        format ('ambient DC field E0*lambda/V0 = ',F3.2)
        write (3,103) ea
  103        format ('bulk permittivity above electrodes = ',2E8.3)
        write (3,107) sea
  107        format ('surface permittivity above electrodes = ',2E8.3)
c
c       set up appropriate scan sequence
c
        print*,'freq. scan from .005 - 10000 Hz? (y or n) '
        read 10,q1
        if(q1.eq.'y') then
                imin=-3
                imax=4
                jmax=8
                index(1)=5.0
```

```
                index(2)=6.0
                index(3)=8.0
                index(4)=10.0
                index(5)=15.0
                index(6)=20.0
                index(7)=25.0
                index(8)=30.0
                index(9)=40.0
        else
                print*,'freq. scan or just one freq? (s or o) '
                read 10,q2
                if(q2.eq.'o') then
                        imin=-1
                        imax=-1
                        jmax=0
                        index(1)=1.0
                else
                        print*,'input minimum frequency '
                        read 15,fmin
                        print*,'input maximum frequency '
                        read 15,fmax
                        print*,'input increment '
                        read 15,inc
                        imin=0
                        imax=0
                        jmax=(fmax-fmin)/inc
                        do 110 j=0,jmax
                         index(j+1)=fmin+j*inc
  110                        continue
                end if
        end if
c
c       compute gain at requested frequencies
c
c         '
        yl=cmplx(YLOAD,0.0)
        do 120 i=imin,imax
        i1=1.0*i
           do 130 j=1,jmax+1
           f=9*(i+3)+j
           if((f.le.58).and.(jmax.lt.61)) then
                freq(f)=index(j)*10**i1
                print*,'freq=',freq(f)
                write (3,172) freq(f)
  172                format('freq=',F10.4)
                w1=2*PI*freq(f)
c
c     normalize the surface permittivity and conductivity
```

```
c
                rsea=sea/(eox*LAM)
                rsea=cmplx(real(rsea),-aimag(rsea)/w1)
c
c       get surface capacitance density above electrodes, Cn
c
              call scap4(d,C)
              rea=re(2)
              print*,'e(2) = ',eox*real(re(2)),-w1*eox*aimag(re(2))
c
c       compute gain
c
              call gain1(C,A,X,V,L,gp(f))
           end if
 130           continue
 120        continue
       write (3,170)
 170       format (' frequency',TR12,' gain',TR12,' phase')
       write (3,173)
 173       format(' $')
       do 175 i=1,58
       write (3,171) freq(i),real(gp(i)),aimag(gp(i))
 171       format(F10.4,TR12,F8.3,TR16,F8.3)
 175       continue
       write (3,173)
       end
```

```
c                thest.for
c                 7/31/87
c
c    estimates thickness given high freq. gain and
c    permittivities of layers 1 and 2 (Section 5.2.1)
c    uses secant method
c
        program thest

        complex*A(50,50),X(50),V(50),L(1000),Lmin,y11,y12,yl
        complex e(100),se(100),re(100),rse(100),C(1000)
        complex ea,sea,rea,rsea
        real i1,freq
        real LAM,eox,YLOAD,h,g,En0,d(100),w1
        integer f,N1,N2
c       always need these guys
        common /comp/rea,rsea
        common /comp1/Lmin,y11,y12,yl
        common /flot/h,g,eox,LAM,YLOAD
        common /flot1/En0
        common /tin1/N1,N2
c       need for homogeneous layers
        common /comp2/e,se,re,rse
        common /flot2/w1
        common /tin2/num
c       need for numerical aspect
        common /tin/k,N,nmax
        complex gp,eup,edown
        real HFG,dold,dnew,Gnew,Gold,slope
        character ofile*10
        integer m
        PI=3.141592654
c       max. number of collocation pts.
        N1=50
c       no. of summation terms (fourier modes)
        N2=1000
c
c            I/O
c
        print*,'output filename? '
        read 10,ofile
   10       format(10a)
        open(unit=3,file=ofile)
c
c       get info about numerical aspect
c
        print*,'input # of summation (fourier) terms '
        read 5,N
```

```
   5        format(I3)
c       set number of collocation points
        k=25
c
c       get info about chip
c
        call chipinfo()
c
c       print*,'input normalized ambient DC field (E0*lambda/V0) '
c       read 15,E0
   15         format(E9.0)
c       E0 represents DC field
        E0=0.0
c
c       get info on known properties of semi-infinite half space
c       and layer
c
        num=2
        d(1)=10000
        print*,d(1)
        print*,'input layer 1 (semi-infinite half space) epsilon (F/m)'
        read 20,e(1)
  20    format(E8.0,E8.0)
        print*,'e1=',e(1)
        print*,'input layer 2 epsilon (F/m)'
        read 20,e(2)
        print*,'e2=',e(2)
        print*,'input measured high freq. gain (signed) '
        read 15,HFG
        print*,'HFG=',HFG
c
c       setup output file
c
        write (3,160) k,N
  160   format ('# of pts. = ',I3,2x,' # of summation terms = ',I4)
        write (3,163) E0
  163        format ('ambient DC field E0*lambda/V0 = ',F3.2)
        write (3,165) real(e(1)),real(e(2))
  165        format ('e1= ',1PE8.2,5x,'e2= ',1PE8.2)
        write (3,166) HFG
  166        format ('HFG= ',F6.2)
c
        yl=cmplx(YLOAD,0.0)
c
c       normalize
c
        do 140 m=1,num
        re(m)=e(m)/eox
```

```
        rse(m)=cmplx(0.0,0.0)
  140        continue
        eup=re(1)
        edown=re(num)
        dold=0.0
c
c       first try with no layer
c       represented by very thick layer
c       having smaller of the two permittivities
c
        d(2)=1000
        if (cabs(edown)>cabs(eup)) then
           re(2)=eup
        else
           re(2)=edown
        end if
        m=1
        goto 130
  131        Gold=real(gp)
c
c   check if experimental data bad
c
        if ((HFG.lt.Gold).and.(m.eq.1)) then
           print*,'not possible, check data'
           goto 194
        end if
        m=2
        re(2)=edown
          dnew=.02
  120        d(2)=dnew
  130        call scapest1(d,C)
        call gain1(C,A,X,V,L,gp)
        print *,' thickness=',LAM*(1E6)*d(2),' gain=',real(gp)
        write (3,*) (LAM*(1E6)*d(2)),real(gp)
c
c    first calculation is to generate zero thickness value
c      and to check data
c
        if (m.eq.1) then
           print*,'computed zero thickness gain'
           goto 131
        end if
        Gnew=real(gp)
        slope=(Gnew-Gold)/(dnew-dold)
c
c   on flat part of curve (for microchip)
c
        if (abs(slope).le.1) then
```

```
            print*,'at limit of sensitivity '
            goto 193
          end if
c
c         breakout if ok else
c             go back and try again
c
          if ((abs(1-Gnew/HFG).lt.(.005)).and.
     x          (abs(HFG-Gnew).lt.(.03))) then
            goto 193
          end if
c         update registers
          dold=d(2)
            Gold=Gnew
c
c         increment guess using eq. (4.3)
c
          dnew=dnew+(HFG-Gnew)/slope
          goto 120
c
c         done
c
  193     print*,'final thickness=',LAM*(1E6)*d(2),' microns'
          write (3,195) LAM*(1E6)*d(2)
  195         format('final thickness=',F6.3)
  194         end
```

```
c                parest.for
c                 8/22/87
c
        program parest
c
c    guesses bulk or surface parameters (Section 4.2.2)
c    input an experimental data pt. (including freq)
c    and known bulk and surface properties of both layers
c    computes using secant method
c    can do variable collocation technique
c    stops after 100 iterations for a given # of grid pts.
c
        complex A(50,50),X(50),V(50),L(1000),Lmin,y11,y12,yl
        complex e(100),se(100),re(100),rse(100),C(1000)
        complex ea,sea,rea,rsea
        real i1,freq
        real LAM,eox,YLOAD,h,g,En0,d(100),w1
        integer f,N1,N2
c       always need these guys
        common /comp/rea,rsea
        common /comp1/Lmin,y11,y12,yl
        common /flot/h,g,eox,LAM,YLOAD
        common /flot1/En0
        common /tin1/N1,N2
c       need for homogeneous layers
        common /comp2/e,se,re,rse
        common /flot2/w1
        common /tin2/num
c       need for numerical aspect
        common /tin/k,N,nmax
        complex gp(2),r(2)
        complex egp,dgpdes(2),delgp(2),deltaes
        complex Vprim,Vdprim,guess
        real ef,err,gtol,ptol
        integer iter,m,m1,prop2,loop,ITMAX
        character q,q2,q3,ofile*10,prop1
        PI=3.141592654
        TLOGe=.43429448
c
c       r(1) and r(2) are registers that hold old and new
c       guesses for estimated permittivity and conductivity
c
c       delgp(1) and delgp(2) are registers that hold old and
c       new gain-phase errors (difference between measured and
c       experimental)
c
c       deltaes holds increment in permittivity and conductivity
c
```

```
c       Vprim and Vdprim are actually discretized of eq. (4.7)
c       adapted to the secant method of eq. (4.3)
c
c       max. number of collocation pts.
        N1=50
c       no. of summation terms (fourier modes)
        N2=1000
        ITMAX=50
c
c
c            I/O
c
        print*,'output filename? '
        read 10,ofile
   10        format(10a)
        open(unit=3,file=ofile)
c
c       get information about numerical aspect
c
        print*,'input # of summation (fourier) terms '
        read 5,N
   5        format(I4)
        print*,' do you wish computer to automatically
     x          perform search over variable grid pnts?
     x          (y or n)'
        read 10,q
        if (q.eq.'n') then
           print*,'input # of pnts. '
           read 5,k
        else
           k=2
        end if
c
c         get information about chip
c
        call chipinfo()
c
c       print*,'input normalized ambient DC field (E0*lambda/V0) '
c       read 15,E0
   15        format(E9.0)
c       E0 represents DC field
        E0=0.0
c       En0 represents AC field with no spatial periodicity
        En0=0.0
c
c       determine property to be estimated
c
        print*,'input property to be estimated'
```

```
      print*,' '
      print*,'  b for bulk'
      print*,'  s for surface'
      read 10,prop1
      if (prop1.eq.'s') then
         print*,'estimate only surface conductivity by suppressing'
         print*,'surface permittivity (y or n)?'
         read 10,q2
      end if
      print*,'indicate appropriate layer for unknown quantity'
      print*,' '
      print*,' 1 for top layer'
      print*,' 2 for next lower layer, etc.'
      print*,' 0 for electrode surface sigma'
      read 5,prop2
c
c     input all known properties
c
      print*,'input # of layers above electrode '
      print*,'(including infinite half-space) '
      read 5,num
      print*,'starting with infinite half space '
      do 100 i=1,num
      print*,'input layer',i,'thickness '
      read 15,d(i)
      d(i)=d(i)/LAM
      print*,d(i)
 20   format(E8.0,E8.0)
      if ((i.ne.prop2).or.((i.eq.prop2).and.(prop1.ne.'b'))) then
         print*,'input layer',i,'epsilon and sigma '
         read 20,e(i)
         print*,'e=',e(i)
         if (i.eq.num) then
            ea=e(i)
         end if
      end if
      if ((i.ne.prop2).or.((i.eq.prop2).and.(prop1.ne.'s'))) then
         print*,'input layer',i,'surface epsilon and sigma '
         read 20,se(i)
         print*,'se=',se(i)
      end if
 100  continue
      if (prop2.ne.0) then
         print*,'input complex surface permittivity at electrodes
     x               (sea) '
         read 20,sea
         print*,sea
      end if
```

```
      print*,'input experimental gain, phase (both signed)
     x    and frequency'
      read 21,egp,ef
 21       format(E6.2,E7.2,E9.3)
      print*,'gain and phase=',egp,' freq=',ef
      print*,'input gtol (in dB)'
      read 15,gtol
      print*,'input ptol (in degrees)'
      read 15,ptol
c
c   setup output file
c
c
      write (3,77) gtol,ptol
 77       format ('gtol = ',F5.3,' ptol = ',F5.3)
      write (3,160) k,N
 160  format ('# of pts. = ',I4,2x,' # of summation terms = ',I4)
      write (3,163) E0
 163       format ('ambient DC field E0*lambda/V0 = ',F3.2)
      do 105 i=1,num
      write (3,165) i
 165       format ('LAYER  ',I2)
      write (3,166) d(i)*LAM
 166       format ('thickness = ',1PE8.2)
      if ((i.ne.prop2).or.((i.eq.prop2).and.(prop1.ne.'b'))) then
         write (3,167) real(e(i)),aimag(e(i))
 167          format ('e = ',1PE8.2,' c = ',1PE8.2)
      end if
      if ((i.ne.prop2).or.((i.eq.prop2).and.(prop1.ne.'s'))) then
         write (3,137) real(se(i)),aimag(se(i))
 137          format ('se = ',1PE8.2,' sc = ',1PE8.2)
      end if
      write (3,169)
 169  format ('**************************************')
 105  continue
      if ((prop1.ne.'b').or.((prop1.eq.'b').and.(prop2.ne.num))) then
         write (3,116) real(ea),aimag(ea)
 116          format ('ea= ',1PE8.2,' sa= ',1PE8.2)
      end if
      if (prop2.ne.0) then
         write (3,117) real(sea),aimag(sea)
 117          format ('eso = ',1PE8.2,' cso= ',1PE8.2)
      end if
      write (3,168) egp,ef
 168       format('gain = ',F6.2,' phase = ',F7.2,' freq = ',1PE10.3)
      write (3,169)
c
      yl=cmplx(YLOAD,0.0)
```

```
        freq=ef
        w1=2*PI*freq
c
c       allow for guesses to be input
c
        print*,'do you wish to guess at parameter? (y or n)'
        read 10,q3
        if (q3.eq.'y') then
           print*,'input guess'
           read 20,guess
           print*,'guess=',guess
           if (prop1.eq.'b') then
                e(prop2)=guess
                if (prop2.eq.num) then
                   ea=guess
                end if
           else
                if (prop2.ne.0) then
                   se(prop2)=guess
                else
                   sea=guess
                end if
           end if
        else
           if (prop1.eq.'b') then
                e(prop2)=cmplx(8.854E-12,0.0)
                guess=e(prop2)
                if (prop2.eq.num) then
                   ea=guess
                end if
           else
                if (prop2.ne.0) then
                   se(prop2)=cmplx(0.0,1E-17)
                   guess=se(prop2)
                else
                   sea=cmplx(0.0,1E-17)
                   guess=sea
                end if
           end if
        end if
c
c    normalize complex permittivites
c
        do 140 m=1,num
        re(m)=e(m)/eox
        re(m)=cmplx(real(re(m)),-aimag(re(m))/w1)
        rse(m)=se(m)/(eox*LAM)
        rse(m)=cmplx(real(rse(m)),-aimag(rse(m))/w1)
```

```
  140        continue
        rea=ea/eox
        rea=cmplx(real(rea),-aimag(rea)/w1)
        rsea=sea/(eox*LAM)
        rsea=cmplx(real(rsea),-aimag(rsea)/w1)
        if (prop1.eq.'b') then
           guess=guess/eox
        else
           guess=guess/(eox*LAM)
        end if
        guess=cmplx(real(guess),-aimag(guess)/w1)
c
c    compute first gain-phase
c
        egp=cmplx(real(egp)/20.0,TLOGe*PI*aimag(egp)/180.0)
c
c         for variable collocation point technique
c       each successive loop represents estimating using
c       a greater number of collocation points
c
        loop=1
  200        iter =1
        m=1
        call scapest1(d,C)
        call gain1(C,A,X,V,L,gp(m))
        iter=iter+1
        gp(m)=cmplx(real(gp(m)/20.0),TLOGe*PI*aimag(gp(m))/180.0)
        delgp(m)=egp-gp(m)
c
c       error
c
        print*,'delG=',20*real(delgp(m)),
     x                 ' delP=',180*aimag(delgp(m))/(PI*TLOGe)
c
c       breakout if good enough
c
         if ((abs(real(delgp(m))).le.(gtol/20.0)).and.
     x      (abs(aimag(delgp(m))).le.(ptol*TLOGe*PI/180.0))) then
           goto 190
        end if
c
c    setup registers
c
        if (loop.eq.1) then
            r(1)=guess
        else
           r(1)=r(2)
        end if
```

```
c
c    next try is 1.02 times initial try unless
c    estimating a bulk parameter with no a priori info
c
        if ((loop.eq.1).and.(q3.eq.'n').and.(prop1.eq.'b')) then
           r(2)=cmplx(1.1*real(r(1)),-1E-13/(w1*eox))
        else
           r(2)=1.02*r(1)
        end if
        if (prop1.eq.'b') then
           re(prop2)=r(2)
           if (prop2.eq.num) then
                rea=r(2)
           end if
        else
           if (prop2.ne.0) then
                rse(prop2)=r(2)
           else
                rsea=r(2)
           end if
        end if
        if (prop1.eq.'b') then
           print*,'next guess: e=',eox*real(r(2)),' +j '
     x        ,-w1*eox*aimag(r(2))
           write (3,*) eox*real(r(2)),-w1*eox*aimag(r(2))
        else
           print*,'next guess: se=',eox*LAM*real(r(2)),
     x             ' +j ',-w1*eox*LAM*aimag(r(2))
           write (3,*) eox*LAM*real(r(2)),-w1*eox*LAM*aimag(r(2))
        end if
c
c
c    second calculation
c
  150       m=2
        call scapest1(d,C)
        call gain1(C,A,X,V,L,gp(m))
        iter=iter+1
        print*,'iter=',iter
        gp(m)=cmplx(real(gp(m)/20.0),TLOGe*PI*aimag(gp(m))/180.0)
        delgp(m)=egp-gp(m)
        print*,'delG=',20*real(delgp(m)),
     x                ' delP=',180*aimag(delgp(m))/(PI*TLOGe)
c
c       breakout if good enough
c
         if ((abs(real(delgp(m))).le.(gtol/20.0)).and.
     x     (abs(aimag(delgp(m))).le.(ptol*TLOGe*PI/180.0))) then
```

```
            goto 190
         end if
         if (iter.eq.3) then
            dgpdes(1)=0
         end if
c
c        prevent division by zero due to lack of convergence
c
         if (r(2).eq.r(1)) then
            print*,'problem with convergence'
            write (3,129)
  129       format ('problem with convergence')
            goto 190
         endif
c
c    compute sensitivities (slopes)
c
         dgpdes(2)=(gp(2)-gp(1))/(r(2)-r(1))
         dgpdes(1)=dgpdes(2)
         write (3,169)
         write (3,121) iter
  121        format('iteration = ',I2)
         write (3,120) 20*real(delgp(2)),
     x                      180*aimag(delgp(m))/(PI*TLOGe)
 120        format('delgp=',E8.2,' +j ',E8.2)
         write (3,126) dgpdes(2)
  126        format(' dgpdes=',E8.2,' +j ',E8.2)
c
c       compute increments
c
c
c     allow breakout for difficulty in converging
c     when data is not informative or bad
c
         if (iter.eq.ITMAX) then
            print*,'problems estimating with this data point,
     x                           reached itmax'
            write (3,128)
 128            format('problems estimating with this data point,
     x                           reached itmax')
            goto 190
         end if
c
c     actual increment here
c
         Vprim=-delgp(2)*dgpdes(2)
         Vdprim=(dgpdes(2))**2
         deltaez=-Vprim/Vdprim
```

```
         if (prop1.eq.'b') then
c           print*,'deltaes=',eox*real(deltaes),-eox*w1*aimag(deltaes)
            print*,'deltaes=',real(deltaes),aimag(deltaes)
         else
            print*,'deltaes=',eox*LAM*real(deltaes),
     x             -eox*w1*LAM*aimag(deltaes)
         end if
c
c        update registers
c
         gp(1)=gp(2)
         r(1)=r(2)
         r(2)=r(2)+deltaes
         if ((prop1.eq.'s').and.(q2.eq.'y')) then
            r(1)=cmplx(0.0,aimag(r(1)))
            r(2)=cmplx(0.0,aimag(r(2)))
         end if
         delgp(1)=delgp(2)
c
c     put bounds on minimum and maximum epsilon or sigma
c     epsilon min = eo,  surface epsilon min = 0
c
          if ((prop1.eq.'b').and.(real(r(2)).lt.(8.854E-12/eox))) then
           r(2)=cmplx(9.7E-12/eox,aimag(r(2)))
         end if
         if ((prop1.eq.'s').and.(real(r(2)).lt.0.0)) then
            r(2)=cmplx(0.0,aimag(r(2)))
         end if
c
c     max epsilon = 10000eo, surface epsilon max = 10000eo/LAM
c
          if ((prop1.eq.'b').and.(real(r(2)).gt.(8.854E-8/eox))) then
           r(2)=cmplx(8.854E-8/eox,aimag(r(2)))
         end if
       if ((prop1.eq.'s').and.(real(r(2)).gt.(8.854E-8/(eox*LAM)))) then
            r(2)=cmplx(8.854E-8/(eox*LAM),aimag(r(2)))
         end if
c
c     min sigma or surface sigma = 0
c
         if (aimag(r(2)).gt.(0.0)) then
            r(2)=cmplx(real(r(2)),-1E-14)
         end if
c
c     max sigma = 10E6 eo, surface sigma max = 10E6 eo/LAM
c
          if ((prop1.eq.'b').and.(-aimag(r(2)).gt.
     x                          (8.854E-6/(w1*eox)))) then
```

```
            r(2)=cmplx(real(r(2)),-8.854E-6/(w1*eox))
         end if
         if ((prop1.eq.'s').and.(-aimag(r(2)).gt.
     x                        (8.854E-6/(w1*eox*LAM)))) then
            r(2)=cmplx(real(r(2)),-8.854E-6/(w1*eox*LAM))
         end if
         write (3,122)
  122         format('next guess')
         if (prop1.eq.'b') then
            print*,'next guess: e=',eox*real(r(2)),' +j '
     x         ,-w1*eox*aimag(r(2))
            write (3,*) eox*real(r(2)),-w1*eox*aimag(r(2))
         else
            print*,'next guess: se=',eox*LAM*real(r(2)),
     x         ' +j ',-w1*eox*LAM*aimag(r(2))
            write (3,*) eox*LAM*real(r(2)),-w1*eox*LAM*aimag(r(2))
         end if
         if (prop1.eq.'b') then
            re(prop2)=r(2)
            if (prop2.eq.num) then
                 rea=r(2)
            end if
         else
            if (prop2.ne.0) then
                 rse(prop2)=r(2)
            else
                 rsea=r(2)
            end if
         end if
          goto 150
c
c       loop if coarse grid (k<25) and using variable
c       collocation technique
c
  190         if ((q.eq.'y').and.(k.ne.25)) then
            loop=loop+1
            if (loop.eq.2) then
                 k=10
            else
                 k=25
            end if
            write (3,201) k
  201            format ('k=',I3)
            print*,'k=',k
            write (3,122)
            if (prop1.eq.'b') then
                 print*,'next guess: e=',eox*real(r(2)),' +j '
     x               ,-w1*eox*aimag(r(2))
```

```
                write (3,*) eox*real(r(2)),-w1*eox*aimag(r(2))
            else
                print*,'next guess: se=',eox*LAM*real(r(2)),
     x                ' +j ',-w1*eox*LAM*aimag(r(2))
                write (3,*) eox*LAM*real(r(2)),-w1*eox*LAM*aimag(r(2))
            end if
            if (prop1.eq.'b') then
                re(prop2)=r(2)
                if (prop2.eq.num) then
                   rea=r(2)
                end if
            else
                if (prop2.ne.0) then
                   rse(prop2)=r(2)
                else
                   rsea=r(2)
                end if
            end if
            q3='y'
            goto 200
         end if
c
c     done
c
         write (3,123)
  123         format('final values')
         if ((m.eq.1).and.(loop.eq.1)) then
            r(1)=guess
         end if
         if (prop1.eq.'b') then
            write (3,*) eox*real(r(m)),-w1*eox*aimag(r(m))
            print*,'final values'
            print*,'e=',eox*real(r(m)),' +j ',-w1*eox*aimag(r(m))
         else
            write (3,*) eox*LAM*real(r(m)),-w1*eox*LAM*aimag(r(m))
            print*,'final values'
            print*,'se=',eox*LAM*real(r(m)),' +j ',
     x           -w1*eox*LAM*aimag(r(m))
         end if
  191          end
```

```
c                hetest.for
c                 10/7/87
c
      program hetest
c
c    guesses distribution of bulk parameters (Section 4.3.4)
c    using sequence of single parameter estimations
c    incorporates basic algorithm of parest.for using secant method
c    input experimental data pts. and load cap. at given
c    spatial wavelength for one freq.
c    can do variable collocation technique
c    includes refinement loop for multiple passes through the data
c    use only lambda/2 on refinement
c    allows input of guesses
c    allows for input of error tolerances
c
      complex A(50,50),X(50),V(50),L(1000),Lmin,y11,y12,yl
      complex e(100),se(100),re(100),rse(100),C(1000)
      complex ea,sea,rea,rsea
      real i1,freq
      real LAM,eox,YLOAD,h,g,En0,d(100),w1
      integer f,N1,N2
c     always need these guys
      common /comp/rea,rsea
      common /comp1/Lmin,y11,y12,yl
        common /flot/h,g,eox,LAM,YLOAD
      common /flot1/En0
      common /tin1/N1,N2
c     need for homogeneous layers
      common /comp2/e,se,re,rse
      common /flot2/w1
      common /tin2/num
c     need for numerical aspect
      common /tin/k,N,nmax
      complex gp(2),r(2)
      complex egp,dgpdes(2),delgp(2),deltaes
      complex Vprim,Vdprim,guess,cgain(100),tmp
      real ef,err,wlen(100),h1,egmin,epmin
      real alpha,gtol,ptol,ylode(100),dsum
      integer iter,m,m2,m3,m4,loop,nlayer,ref,ITMAX
      integer boundel,consel,boundsl,conssl
      integer boundeh,conseh,boundsh,conssh
      character q,q1,ofile*10,flag
      common h1
      PI=3.141592654
      TLOGe=.43429448
c
c     r(1) and r(2) are registers that hold old and new
```

```
c       guesses for estimated permittivity and conductivity
c
c       delgp(1) and delgp(2) are registers that hold old and
c       new gain-phase errors (difference between measured and
c       experimental)
c
c       deltaes holds increment in permittivity and conductivity
c
c       Vprim and Vdprim are actually discretized of eq. (4.7)
c       adapted to the secant method of eq. (4.3)
c
c       max. number of collocation pts.
        N1=50
c       no. of summation terms (fourier modes)
        N2=1000
        ITMAX=50
c
c       iter = number of iterations for a given layer property
c       k = number of collocation pts.
c       m,m3 = dummy counters
c       m4 = counter for refinement loop
c       m2 = counter for layer whose property is being estimated
c       ref = number of refinement loops
c
c            I/O
c
        print*,'output filename? '
        read 10,ofile
   10       format(10a)
        open(unit=3,file=ofile)
c
c       get information about numerical aspect
c
        print*,'input # of summation (fourier) terms '
        read 5,N
   5        format(I4)
        print*,' do you wish computer to automatically'
        print*,' perform search over variable collocation pnts?'
        print*,' (y or n)'
        read 10,q
        if (q.eq.'n') then
           print*,'input # of pnts. '
           read 5,k
        else
           k=2
        end if
c
c         get information about chip
```

```
c
        call chipinfo()
c
c       print*,'input normalized ambient DC field (E0*lambda/V0) '
c       read 15,E0
   15         format(E9.0)
c       E0 represents DC field
        E0=0.0
c       En0 represents AC field with no spatial periodicity
        En0=0.0
c
c       input data points
c
        print*,'input # of measurements '
        read 5,nlayer
        print*,'input frequency '
        read 15,ef
        print*,'input experimental gain, phase (both signed) '
        print*,'yload and frequency'
        print*,'from shortest to longest wavelength'
        do 100 i=1,nlayer
        read 21,cgain(i),ylode(i),wlen(i)
  21        format(E6.2,E7.2,E6.2,E9.3)
        print*,'wavelength=',wlen(i)
        print*,'gain and phase=',cgain(i),' yload=',ylode(i)
  100        continue
c
c     determines extent to which a particular spatial wavelength
c     'sees' into the medium. usually one-quarter wavelength
c
        print*,'input lambda extension'
        read 15,alpha
c
c     set error tolerances determining convergence
c
        print*,'input gtol (in dB)'
        read 15,gtol
        print*,'input ptol (in degrees)'
        read 15,ptol
        print*,'do you want refinement (y or n)?'
        read 10,flag
        if (flag.eq.'y') then
           print*,'input number of refinements'
           read 5,ref
        else
           ref=1
        end if
        print*,'do you want to input guesses (y or no)?'
```

```
      read 10,q1
      if (q1.eq.'y') then
         print*,'input complex permittivity'
         print*,'start from uppermost layer and'
         print*,'work downwards'
         do 28 m=1,nlayer
              read 11,e(m)
 11               format (E8.0,E8.0)
 28           continue
      end if
c
c   setup output file
c
c
      write (3,177) alpha
 177       format('alpha=',F5.3)
      write (3,77) gtol,ptol
 77        format ('gtol = ',F5.3,' ptol = ',F5.3)
      write (3,160) k,N
 160  format ('# of pts. = ',I3,2x,' # of summation terms = ',I4)
      write (3,163) E0
 163       format ('ambient DC field E0*lambda/V0 = ',F3.2)
      write (3,165) ef
 165       format ('freq. = ',1PE10.3)
      do 105 i=1,nlayer
      write (3,164) wlen(i)
 164       format(' wlen = ',1PE10.3)
      write (3,168) cgain(i),ylode(i)
 168       format('gain = ',F6.2,' phase = ',F7.2,' yload = ',F6.2)
      write (3,169)
 169  format ('**************************************')
 105  continue
      if (q1.eq.'y') then
         do 14 m=1,nlayer
              write (3,12) real(e(m)),aimag(e(m))
 12               format('guess e = ',1PE8.2,'guess s = ',1PE8.2)
 14           continue
      end if

      freq=ef
      w1=2*PI*freq
c
c     estimate from shortest to longest wavelength
c
      num=0
c
c     outer loop is for refinement, inner loop is for marching from
c     shortest to longest wavelength
```

```
c
      do 205 m4=1,ref
c
c          skip first loop if guesses are input for all the layers
c
      if ((q1.eq.'y').and.(m4.eq.1)) then
         goto 30
      end if
c
         do 135 m2=1,nlayer
         print*,'m4=',m4,'m2=',m2
         write (3,801) m4,m2
 801          format('m4=',I2,5x,'m2=',I2)
c
c     to use variable collocation technique only for first time
c     through uncomment the following line and comment the line
c     after it
c
c        if ((q.eq.'y').and.(m4.eq.1)) then
         if (q.eq.'y') then
              k=2
         end if
c
c     setup the appropriate load admittance, complex gain and
c     oxide layer thickness for the given spatial wavelength
c
         yl=cmplx(ylode(m2),0.0)
         LAM=wlen(m2)
         egp=cgain(m2)
         egp=cmplx(real(egp)/20.0,TLOGe*PI*aimag(egp)/180.0)
         h=h1/LAM
c
c     first time through, the unknown medium is represented by
c     an infinite half space, reducing the total number of layers
c
         if (m4.eq.1) then
              num=num+1
         else
              num=nlayer
         end if
c
c     setup layer thicknesses
c
         d(1)=1000
         if ((m2.gt.1).and.(m4.eq.1)) then
              d(m2)=alpha*wlen(1)
              do 137 m3=m2-1,2,-1
              d(m3)=alpha*(wlen(m2-m3+1)-wlen(m2-m3))
```

```
  137               continue
          end if
c
c       for refinement, use only calculated values
c       out to lambda/2, then make semi-infinite
c
          if (m4.gt.1) then
               d(nlayer)=alpha*wlen(1)
               dsum=d(nlayer)
               do 206 m3=nlayer-1,2,-1
                  if (dsum.lt.wlen(m2)/2) then
                       d(m3)=alpha*(wlen(nlayer-m3+1)-wlen(nlayer-m3))
                       dsum=dsum+d(m3)
                  else
                       d(m3)=1000
                  end if
  206               continue
          end if
c
c       setup permittivity for first march
c       first layer guess is air, the guess for following layers
c       is the preceding layer's estimate
c
          if (m4.eq.1) then
               if (m2.eq.1) then
                  e(1)=cmplx(8.854E-12,0.0)
               else
                  e(1)=e(2)
               end if
               ea=e(m2)
          else
               ea=e(nlayer)
          end if
c
c       normalize to oxide layer permittivity
c       convert imaginary part to loss factor from conductivity
c
          do 140 m=1,num
          d(m)=d(m)/LAM
          re(m)=e(m)/eox
          re(m)=cmplx(real(re(m)),-aimag(re(m))/w1)
          rse(m)=cmplx(0.0,0.0)
  140          continue
          rea=cmplx(real(ea),-aimag(ea)/w1)/eox
          rsea=cmplx(0.0,0.0)
          if (m4.eq.1) then
               guess=re(1)
          else
```

```
                guess=re(nlayer-m2+1)
            end if
c
c    compute first gain-phase
c
            loop=1
  200           iter =1
            m=1
            call scapest1(d,C)
            call gain1(C,A,X,V,L,gp(m))
            iter=iter+1
            gp(m)=cmplx(real(gp(m)/20.0),TLOGe*PI*aimag(gp(m))/180.0)
            delgp(m)=egp-gp(m)
            print*,'delG=',20*real(delgp(m)),' delP=',
     x                 180*aimag(delgp(m))/(PI*TLOGe)
c
c       set limits for breakout
c       reduced tolerance for k<25
c
            if (k.eq.25) then
                egmin=gtol/20.0
                epmin=ptol*TLOGe*PI/180.0
            else
                egmin=.2/20.0
                epmin=.4*TLOGe*PI/180.0
            end if
c
c       reset r(1) if estimating with different k
c
            if (loop.eq.1) then
                 r(1)=guess
            else
                r(1)=r(2)
            end if
c
c
c       breakout if good enough
c
            if ((abs(real(delgp(m))).le.egmin).and.
     x      (abs(aimag(delgp(m))).le.epmin)) then
                goto 190
            end if
c
c    next try is 1.02 times initial try unless
c    estimating a bulk parameter with no a priori info
c    first change the registers
c
 1000           if ((loop.eq.1).and.(m4.eq.1)) then
```

```
                r(2)=cmplx(1.1*real(r(1)),-1E-13/(w1*eox))
            else
                r(2)=1.02*r(1)
            end if
c
c       then update actual values used in model
c
            if (m4.eq.1) then
                re(1)=r(2)
                rea=re(m2)
            else
                re(nlayer-m2+1)=r(2)
                rea=re(nlayer)
            end if
            print*,'next guess: e=',eox*real(r(2)),' +j '
     x         ,-w1*eox*aimag(r(2))
            write (3,*) eox*real(r(2)),-w1*eox*aimag(r(2))
c
c    second calculation
c    set counters that note when have problems with bounds
c
            boundel=0
            boundsl=0
            boundeh=0
            boundsh=0
            consel=0
            conssl=0
            conseh=0
            conssh=0
c
c       second gain--phase calculation
c
  150           m=2
            call scapest1(d,C)
            call gain1(C,A,X,V,L,gp(m))
            iter=iter+1
        gp(m)=cmplx(real(gp(m)/20.0),TLOGe*PI*aimag(gp(m))/180.0)
            delgp(m)=egp-gp(m)
            print*,'delG2=',20*real(delgp(m)),' delP2=',
     x                      180*aimag(delgp(m))/(PI*TLOGe)
c
c       breakout if good enough
c
            if ((abs(real(delgp(m))).le.egmin).and.
     x      (abs(aimag(delgp(m))).le.epmin)) then
                goto 190
            end if
c
```

```
c     allow breakout for difficulty in converging
c     when data is not informative or bad
c
         if (iter.eq.ITMAX) then
            print*,'problems estimating with this data point,
     x                           reached itmax'
            write (3,128)
 128           format('problems estimating with this data point,
     x                           reached itmax')
            goto 190
         end if
c
c    compute sensitivities (slopes)
c    do not worry about second derivatives
            dgpdes(2)=(gp(2)-gp(1))/(r(2)-r(1))
c
c        breakout if hung up in low sensitivity area
c
            if (cabs(dgpdes(2)).eq.0.0) goto 190
            dgpdes(1)=dgpdes(2)
            write (3,169)
            write (3,121) iter
  121           format('iteration = ',I3)
            write (3,120) 20*real(delgp(2)),
     x                     180*aimag(delgp(2))/(PI*TLOGe)
 120          format('delgp=',E8.2,' +j ',E8.2)
c           write (3,126) dgpdes(2)
c  126           format(' dgpdes=',E8.2,' +j ',E8.2)
c
c     actual increment here
c
            Vprim=-delgp(2)*dgpdes(2)
            Vdprim=(dgpdes(2))**2
            deltaes=-Vprim/Vdprim
            print*,'deltaes=',eox*real(deltaes),-eox*w1*aimag(deltaes)
c
c        only increment r(2) and check if
c        it is within bounds
c
            tmp=r(2)
            r(2)=r(2)+deltaes
c
c     put bounds on minimum epsilon or sigma
c     epsilon min = eo,  surface epsilon min = 0
c     min sigma or surface sigma = 0
c
            if (real(r(2)).lt.(8.854E-12/eox)) then
                r(2)=cmplx(8.853E-12/eox,aimag(r(2)))
```

```
                boundel=boundel+1
                if (boundel.eq.1) then
                   consel=boundel+iter
                end if
c               print*,'hit e bound'
c               if (iter.ne.3)        goto 1000
             end if
             if (aimag(r(2)).gt.(0.0)) then
                r(2)=cmplx(real(r(2)),-0.0)
                boundsl=boundsl+1
                if (boundsl.eq.1) then
                   conssl=boundsl+iter
                end if
c               print*,'hit s bound'
c               if (iter.ne.3)        goto 1000
             end if
c
c       bounds if maximum limits exceeded
c
             if (real(r(2)).gt.(8.854E-8/eox)) then
                r(2)=cmplx(8.853E-8/eox,aimag(r(2)))
                boundeh=boundeh+1
                if (boundeh.eq.1) then
                   conseh=boundeh+iter
                end if
c               print*,'hit e max bound'
c               if (iter.ne.3)        goto 1000
             end if
             if (-aimag(r(2)).gt.(8.854E-6/(w1*eox))) then
                r(2)=cmplx(real(r(2)),-8.854E-6/(w1*eox))
                boundsh=boundsh+1
                if (boundsh.eq.1) then
                   conssh=boundsh+iter
                end if
c               print*,'hit s max bound'
c               if (iter.ne.3)        goto 1000
             end if
c
c       reset registers now that it is known
c       that next guess is within bounds
c
             gp(1)=gp(2)
             r(1)=tmp
             delgp(1)=delgp(2)
c
c       if first time through then this estimate is to be used for
c       the next iteration which by definition is the infinite half
c       space. otherwise this estimate is the value for the particular
```

```
c       layer being refined
c
           if (m4.eq.1) then
                re(1)=r(2)
                rea=re(m2)
           else
                re(nlayer-m2+1)=r(2)
                rea=re(nlayer)
           end if
c
c       if have hit against boundary on sigma or epsilon consecutively
c       then breakout and continue on to next layer
c
           if ((boundel.eq.2).or.(boundsl.eq.2)) then
                if ((consel.eq.(boundel+iter-2)).or.
     x                    (conssl.eq.(boundsl+iter-2))) then
                   goto 190
                else
                   boundel=0
                   boundsl=0
                end if
           end if
           if ((boundeh.eq.2).or.(boundsh.eq.2)) then
                if ((conseh.eq.(boundeh+iter-2)).or.
     x                    (conssh.eq.(boundsh+iter-2))) then
                   goto 190
                else
                   boundeh=0
                   boundsh=0
                end if
           end if
           write (3,122)
  122           format('next guess')
           print*,'next guess: e=',eox*real(r(2)),' +j '
     x        ,-w1*eox*aimag(r(2))
           write (3,*) eox*real(r(2)),-w1*eox*aimag(r(2))
c
c       continue iteration for this layer
c
           goto 150
c
c      loop if coarse grid (k<25) and using variable collocation
c      technique
c
  190           if ((q.eq.'y').and.(k.ne.25)) then
                loop=loop+1
                if (loop.eq.2) then
                   k=10
```

```
                else
                   k=25
                end if
                write (3,201) k
  201                format ('k=',I3)
                print*,'k=',k
                write (3,122)
                print*,'next guess: e=',eox*real(r(2)),' +j '
     x                ,-w1*eox*aimag(r(2))
                write (3,*) eox*real(r(2)),-w1*eox*aimag(r(2))
c
c       reset registers
c
                if (m4.eq.1) then
                   re(1)=r(2)
                   rea=re(m2)
                else
                   re(nlayer-m2+1)=r(2)
                   rea=re(nlayer)
                end if
                goto 200
             end if
c
c       now done with this loop
c
             write (3,123)
  123             format('final values')
             if (iter.eq.1) then
                  r(2)=r(1)
             end if
             write (3,*) eox*real(r(2)),-w1*eox*aimag(r(2))
             print*,'final values'
             print*,'e=',eox*real(r(2)),' +j ',-w1*eox*aimag(r(2))
c
c       save estimated value
c
             if (m4.eq.1) then
                  e(1)=eox*cmplx(real(r(2)),-w1*aimag(r(2)))
             else
                  e(nlayer-m2+1)=eox*cmplx(real(r(2)),-w1*aimag(r(2)))
             end if
             ea=eox*cmplx(real(rea),-w1*aimag(rea))
             write (3,169)
             if (m4.eq.1) then
                  do 149 m3=1,m2
                  write (3,718) m3
  718                   format ('LAYER  ',I2)
                  write (3,766) d(m3)*LAM
```

```
  766               format ('thickness = ',1PE8.2)
                  write (3,767) real(e(m3)),aimag(e(m3))
  767               format ('e = ',1PE8.2,' c = ',1PE8.2)
                  write (3,169)
  149             continue
            else
                  do 849 m3=1,nlayer
                  write (3,718) m3
                  write (3,766) d(m3)*LAM
                  write (3,767) real(e(m3)),aimag(e(m3))
                  write (3,169)
  849             continue
            end if
            write (3,716) real(ea),aimag(ea)
  716             format ('ea= ',1PE8.2,' sa= ',1PE8.2)
            write (3,169)
c
c       setup layer permittivities for
c       next estimation
c
            if ((m4.eq.1).and.(m2.ne.nlayer)) then
                  do 138 m3=m2,1,-1
                  e(m3+1)=e(m3)
  138                   continue
            end if
  135             continue
  30    continue
  205         continue
  191          end
```

```
c                discon1.for
c                 10/4/87
c
c    estimates layer thickness and upper permittivity
c    from variable wavelength data, uses Newton's method
c    Section 4.1.2
c    NOTE - this technique is using a complex multidimensional
c           search
c
       program discon1

       parameter (PI=3.141592654)
       complex A(50,50),X(50),V(50),L(1000),Lmin,y11,y12,yl
       complex e(100),se(100),re(100),rse(100),C(1000)
       complex ea,sea,rea,rsea
       real LAM,eox,YLOAD,h,g,En0,d(100),w1
       character ofile*10
c      always need these guys
       common /comp/rea,rsea
       common /comp1/Lmin,y11,y12,yl
       common /flot/h,g,eox,LAM,YLOAD
       common /flot1/En0
       common /tin1/N1,N2
c      need for homogeneous layers
       common /comp2/e,se,re,rse
       common /flot2/w1
       common /tin2/num
c      need for numerical aspect
       common /tin/k,N,nmax
c
       complex Gpnew(10,3),Gold(10),reslope(10,2),renew(2),reold(2)
       complex err(10),eguess(2),dslope(10),gp,egp(10)
       complex jac(50,50),ijac(50,50),pjac(50,50),tjac(50,50)
       complex ipjac(50,50),temp2(50,50),pjac1(2500),det
       common h1
       real dold,dnew,ef,wt(10,10),h1,gtol,ptol,ylode(10)
       real dguess,wtd,wte(2),wlen(10),scale,totalerrold,totalerrnew
       character q3
       integer i1,m,m1,chk,dnum,flag1,iter,nvar,count,count1,ITMAX
       TLOGe=.43429448
c      max. number of collocation pts.
       N1=50
c      no. of summation terms (fourier modes)
       N2=1000
c
c      totalerrold and totalerrnew are registers holding the magnitude
c      squared of the total error for old and new guesses - they are
c      tested to insure the error is always decreasing
```

```
c
c       dold and dnew are registers holding old and new thickness
c       guesses
c
c       reold and renew are registers holding old and new complex
c       permittivity guesses
c
c            I/O
c
        print*,'output filename? '
        read 10,ofile
   10       format(10a)
        open(unit=3,file=ofile)
c
c       get information about numerical aspect
c
        print*,'input # of summation (fourier) terms '
        read 5,N
    5      format(I4)
        print*,'input # of collocation pnts. '
        read 5,k
c
c         get information about chip
c
        call chipinfo()
c       print*,'input normalized ambient DC field (E0*lambda/V0) '
c       read 15,E0
c       E0 represents DC field
        E0=0.0
        num=2
        d(1)=10000
        print*,'input # of variables to be estimated'
        read 5,nvar
        print*,'input # of data points'
        read 5,dnum
        print*,'input frequency'
        read 13,ef
   13      format(E9.0)
        print*,'input scale factor for derivative computation'
        read 13,scale
        print*,'scale=',scale
        if (nvar.eq.2) then
           print*,'input lower permittivity and conductivity, e2'
           read 16,e(2)
   16          format(2E8.0)
        end if
        print*,'input data points from short to long wavelength'
        do 30 m=1,dnum
```

```
      print*,'input gain, phase, load capacitance and wavelength'
      read 20,egp(m),ylode(m),wlen(m)
 20   format(F6.2,F7.2,F6.2,F9.3)
      print*,' wavelength=',wlen(m)
      print*,'gain and phase=',egp(m),' yload=',ylode(m)
c     print*,'input weighting for thickness and e1'
c     read 27,wt(1,m),wt(2,m)
c 27      format(2F4.0)
c     print*,wt(1,m),wt(2,m)
 30       continue
      print*,'input gtol (in dB)'
      read 15,gtol
      print*,'input ptol (in degrees)'
      read 15,ptol
      print*,'do you wish to guess the thickness and permittivities?
     x           (y or n)'
      read 10,q3
      if (q3.eq.'y') then
         print*,'input thickness'
         read 15,dguess
 15       format(E8.2)
         print*,'thickness guess ',dguess
              do 29 m=1,nvar-1
              print*,'input permittivity from upper to lower layer'
              read 16,eguess(m)
              print*,'permittivity guess ',eguess(m)
 29               continue
         flag1=1
      else
         flag1=0
      end if
c
c     write the relevant data into the output file
c
      write (3,160) k,N
 160  format ('# of pts. = ',I3,2x,' # of summation terms = ',I4)
c     write (3,163) E0
c 163       format ('ambient DC field E0*lambda/V0 = ',F3.2)
      write (3,165) ef
      write (3,777) gtol,ptol
 777      format ('gtol = ',F5.3,' ptol = ',F5.3)
 165      format ('freq. = ',1PE10.3)
      if (nvar.eq.2) then
         write (3,167) e(2)
 167           format ('e2 = ',1PE8.2,' s2 = ',1PE8.2)
      end if
      write (3,272) scale
 272      format ('for derivative comp. scale = ',F5.3)
```

```
      do 148 m=1,dnum
      write (3,168) wlen(m)
  168      format('wlen = ',1PE10.3)
      write (3,768) egp(m),ylode(m)
  768      format('gain = ',F6.2,' phase = ',F7.2,' yload = ',F6.2)
      write (3,169)
  169 format ('****************************************')
  148      continue
      iter=1
      rsea=cmplx(0.0,0.0)
      do 141 m1=1,num
      rse(m1)=cmplx(0.0,0.0)
  141      continue
c
c     allow for guesses to be input
c
      if (flag1.eq.1) then
         write (3,283)
  283         format ('guess input')
         dnew=dguess
         do 70 m1=1,nvar-1
         renew(m1)=eguess(m1)/eox
  70         continue
      else
         write (3,282)
  282         format ('no guess input')
          dnew=wlen(1)/8
         do 71 m1=1,nvar-1
         renew(m1)=cmplx(m1*9.07E-12,3E-12/m1)/eox
  71         continue
      end if
c
c     scale measured gain--phase responses
c
      do 87 m=1,dnum
        egp(m)=cmplx(real(egp(m))/20.0,TL0Ge*PI*aimag(egp(m))/180.0)
  87     continue
      count=0
      count1=0
c
c     compute first gain-phase response
c
 120     d(2)=dnew
      if (nvar.eq.2) then
         renew(2)=e(2)/eox
      end if
      print *,' thickness=',d(2)
      print*,' e1=',eox*renew(1)
```

```
        print*,' e2=',eox*renew(2)
        write (3,*) d(2)
        write (3,*) eox*renew(1)
        write (3,*) eox*renew(2)
        totalerrnew=0.0
        do 73 m=1,dnum
        yl=cmplx(ylode(m),0.0)
        LAM=wlen(m)
        h=h1/LAM
        d(2)=dnew/LAM
           do 142 m1=1,num
           re(m1)=cmplx(real(renew(m1)),-aimag(renew(m1))/(2*PI*ef))
  142           continue
        rea=re(2)
        call scapest1(d,C)
         call gain1(C,A,X,V,L,gp)
        gp=cmplx(real(gp)/20.0,TLOGe*PI*aimag(gp)/180.0)
        Gold(m)=gp
        err(m)=egp(m)-Gold(m)
        print*,'err(',m,')= ',20*real(err(m)),
     x               180*aimag(err(m))/(PI*TLOGe)
        totalerrnew=totalerrnew+cabs(err(m))
c       write (3,*) gp
  73        continue
        print*,'iter=',iter
        print*,'totalerrnew= ',totalerrnew
        iter=iter+1
c
c   go back and try again
c
c       breakout if good enough
c
        chk=0
        do 146 m=1,dnum
        if ((abs(real(err(m))).lt.(gtol/20.0)).and.
     x    (abs(aimag(err(m))).lt.(ptol*TLOGe*PI/180.0))) then
           chk=chk
        else
           chk=1+chk
        end if
  146        continue
        print*,'chk=',chk
        if (chk.eq.0) then
           goto 193
        end if
c
c       if error is not decreasing go back and try
c       using half of the original step size
```

```
c       (should reold and dold be updated?)
c
        if ((totalerrold.lt.totalerrnew).and.(iter.ne.2)) then
           count=count+1
           if (count.eq.1) then
c               goto 1000
c          else if (count.eq.2) then
                print*,'go other way'
                write (3,203)
  203                format ('go other way')
                dnew=2*dold-dnew
                do 35 m=1,nvar-1
                renew(m)=2*reold(m)-renew(m)
  35               continue
                 totalerrold=totalerrnew
                call bound(renew,reold,dnew,dold,eox,nvar,wlen)
                dold=dnew
                do 179 m1=1,nvar-1
                reold(m1)=renew(m1)
  179                continue
           else
                count=0
                count1=0
                goto 1000
           end if
           goto 1001
        end if
        count=0
c
c       update old registers
c
 1000        totalerrold=totalerrnew
        dold=dnew
        do 34 m=1,nvar-1
        reold(m)=renew(m)
  34        continue
c
c       next guess is scale times previous one
c       to obtain sensitivity
c       change only variable being searched
c       first change thickness
c
        do 111 m=1,dnum
        yl=cmplx(ylode(m),0.0)
        LAM=wlen(m)
        h=h1/LAM
        d(2)=scale*dold/LAM
           do 31 m1=1,num
```

```
            re(m1)=cmplx(real(renew(m1)),
     x                   -aimag(renew(m1))/(2*PI*ef))
  31           continue
       rea=re(2)
       call scapest1(d,C)
       call gain1(C,A,X,V,L,gp)
       gp=cmplx(real(gp)/20.0,TLOGe*PI*aimag(gp)/180.0)
       Gpnew(m,1)=gp
       dslope(m)=(.25*wlen(1))*(Gpnew(m,1)-Gold(m))/((scale-1)*dold)
       jac(m,1)=dslope(m)
c      print*,'deltaG=',Gpnew(m,1)-Gold(m)
       if ((m.ne.1).and.(m.le.nvar)) then
          do 76 m1=1,dnum
c
c      must tailor this section for each estimation
c      now changing appropriated complex bulk permittivity
c
               do 145 i=1,nvar-1
               if (i.eq.(m-1)) then
                  re(i)=scale*cmplx(real(renew(i)),
     x                        -aimag(renew(i))/(2*PI*ef))
               else
                re(i)=cmplx(real(renew(i)),-aimag(renew(i))/(2*PI*ef))
               end if
  145               continue
c
          yl=cmplx(ylode(m1),0.0)
          LAM=wlen(m1)
          d(2)=dold/LAM
          h=h1/LAM
          rea=re(2)
          call scapest1(d,C)
          call gain1(C,A,X,V,L,gp)
          gp=cmplx(real(gp)/20.0,TLOGe*PI*aimag(gp)/180.0)
          Gpnew(m1,m)=gp
          reslope(m1,m-1)=(Gpnew(m1,m)-Gold(m1))/((scale-1)*re(m-1))
          jac(m1,m)=reslope(m1,m-1)
  76          continue
       end if
  111       continue
c
c      take transpose of jacobian, tjac
c
          do 180 m=1,nvar
               do 181 m1=1,dnum
c      print*,'jac(',m,',',m1,')= ',jac(m,m1)
               tjac(m,m1)=jac(m1,m)
c         print*,'tjac(',m,m1,')= ',tjac(m,m1)
```

```
  181               continue
  180            continue
c
c       multiply tjac and jac as in eq. (4.10)
c
        call matmpy1(tjac,jac,pjac,nvar,dnum,nvar)
c
c       read the jacobian into columnar form
c       for matrix inversion routine
c
        do 143 m=1,nvar
           do 144 m1=1,nvar
           pjac1(nvar*(m1-1)+m)=pjac(m,m1)
c       print*,'pjac',m,m1,')=',pjac(m,m1)
  144           continue
  143        continue
c
c       invert product tjac*jac
c
        call minv1(pjac1,nvar,det)
c
c       read inverse back into multidimensional form
c
           do 156 m=1,nvar
                do 157 m1=1,nvar
                ijac(m,m1)=pjac1(nvar*(m1-1)+m)
c       print*,'ijac(',m,m1,')=',ijac(m,m1)
  157               continue
  156            continue
c
c       check inverse
c
c       print*,'ident'
c       call matmpy1(ijac,pjac,temp2,nvar,nvar,nvar)
c       do 136 m=1,nvar
c          do 137 m1=1,nvar
c          print*,' ident(',m,m1,')= ',temp2(m,m1)
c          print*,' jac(',m,m1,')= ',jac(m,m1)
c          print*,' ijac(',m1,m,')= ',ijac(m1,m)
c          read 138,jac(m,m1)
c  138           format (2F6.0)
c  137           continue
c  136        continue
c
c       multiply inverse ijac by tjac as in eq. (4.9)
c
        call matmpy1(ijac,tjac,pjac,nvar,nvar,dnum)
c
```

```
c       compute increments as in eq. (4.9)
c
        do 77 m=1,nvar
           do 78 m1=1,dnum
c          print*,'pjac(',m,',',m1,')=',pjac(m,m1)
           if (m.eq.1) then
              dnew=dnew+(.25*wlen(1))*real(pjac(1,m1)*err(m1))
           end if
           if (m.ne.1) then
            re(m-1)=cmplx(real(renew(m-1)),-aimag(renew(m-1))/(2*PI*ef))
              re(m-1)=re(m-1)+pjac(m,m1)*err(m1)
                renew(m-1)=cmplx(real(re(m-1)),-2*PI*ef*aimag(re(m-1)))
           end if
  78          continue
  77       continue
 1001        print*,'delta d = ',(dnew-dold)
        do 79 m=1,nvar-1
        print*,'delta re(',m,')= ',eox*(renew(m)-recld(m))
  79        continue
c
c       check that updated guesses are within physical bounds
c
        call bound(renew,reold,dnew,dold,eox,nvar,wlen)
        goto 120
c
c       done
c
  193   print*,'final thickness =',dnew
        write (3,195) dnew
  195        format('final thickness = ',1PE8.2)
        do 84 m=1,nvar-1
        print*,'final permittivity= ',eox*renew(m)
        write (3,196) eox*renew(m)
  196        format('final permittivity = ',2E10.2)
  84        continue
  194        end

c
c       set bounds
c
        subroutine bound(renew,reold,dnew,dold,eox,nvar,wlen)
        complex renew(100),reold(100)
        real dnew,dold,eox,wlen(10)
        integer nvar,m
        do 83 m=1,nvar-1
        if (real(renew(m)).lt.(8.854E-12/eox)) then
           print*,'e min'
c          renew(m)=cmplx(real(reold(m)),aimag(renew(m)))
```

```
            renew(m)=cmplx(8.854E-12/eox,aimag(renew(m)))
         end if
         if (aimag(renew(m)).lt.0.0) then
            print*,'s min'
c           renew(m)=cmplx(real(renew(m)),aimag(reold(m)))
            renew(m)=cmplx(real(renew(m)),0.0)
         end if
         if (real(renew(m)).gt.(1E-9/eox)) then
            print*,'e max'
            renew(m)=cmplx(real(reold(m)),aimag(renew(m)))
         end if
         if (aimag(renew(m)).gt.(1E-7/eox)) then
            print*,'s max'
            renew(m)=cmplx(real(renew(m)),aimag(reold(m)))
         end if
  83      continue
         if (dnew.lt.0.0) then
            print*,'d min bound'
            dnew=wlen(1)/50
         end if
         if (dnew.gt.(wlen(1)/3)) then
            print*,'d max bound'
            dnew=dold
         end if
         return
         end
```

```
c                discon2.for
c                 10/4/87
c
c    estimates surface permittivity and upper permittivity
c    from variable wavelength data, uses Newton's method
c    Section 4.1.2
c    NOTE - this technique is using a complex multidimensional
c           search
c
       program discon2

       parameter (PI=3.141592654)
       complex A(50,50),X(50),V(50),L(1000),Lmin,y11,y12,yl
       complex e(100),se(100),re(100),rse(100),C(1000)
       complex ea,sea,rea,rsea
       real LAM,eox,YLOAD,h,g,En0,d(100),w1
       character ofile*10
c      always need these guys
       common /comp/rea,rsea
       common /comp1/Lmin,y11,y12,yl
       common /flot/h,g,eox,LAM,YLOAD
       common /flot1/En0
       common /tin1/N1,N2
c      need for homogeneous layers
       common /comp2/e,se,re,rse
       common /flot2/w1
       common /tin2/num
c      need for numerical aspect
       common /tin/k,N,nmax
c
       complex Gpnew(10,3),Gold(10),reslope(10,2),renew(2),reold(2)
       complex err(10),eguess(2),gp,egp(10)
       complex jac(50,50),ijac(50,50),pjac(50,50),tjac(50,50)
       complex ipjac(50,50),temp2(50,50),pjac1(2500),det
       complex rsenew,rseold,seguess,rsslope(2)
       common h1
       real ef,wt(10,10),h1,dguess,gtol,ptol,ylode(10)
       real wtd,wte(2),wlen(10),scale,totalerrold,totalerrnew
       character q3,q,q2
       integer i1,m,m1,chk,dnum,flag1,iter,nvar,count,ITMAX
       TLOGe=.43429448
c      max. number of collocation pts.
       N1=50
c      no. of summation terms (fourier modes)
       N2=1000
c
c      totalerrold and totalerrnew are registers holding the magnitude
c      squared of the total error for old and new guesses - they are
```

```
c       tested to insure the error is always decreasing
c
c       rseold and rsenew are registers holding old and new complex
c       surface permittivity guesses
c
c       reold and renew are registers holding old and new complex
c       permittivity guesses
c
c            I/O
c
        print*,'output filename? '
        read 10,ofile
   10       format(10a)
        open(unit=3,file=ofile)
c
c       get information about numerical aspect
c
        print*,'input # of summation (fourier) terms '
        read 5,N
    5      format(I4)
        print*,'input # of collocation pnts. '
        read 5,k
c
c         get information about chip
c
        call chipinfo()
c       print*,'input normalized ambient DC field (E0*lambda/V0) '
c       read 15,E0
c       E0 represents DC field
        E0=0.0
        num=2
        d(1)=10000
        print*,'input # of variables to be estimated'
        read 5,nvar
        print*,'input # of data points'
        read 5,dnum
        print*,'input frequency'
        read 13,ef
   13      format(E9.0)
        print*,'input scale factor for derivative computation'
        read 13,scale
        print*,'scale=',scale
        if (nvar.eq.2) then
           print*,'input layer thickness (meters)'
           read 15,dguess
   15         format (E9.2)
         print*,'input lower complex permittivity and conductivity, e2'
           read 16,e(2)
```

```
  16          format(2E8.0)
       end if
       print*,'estimate only surface conductivity while'
       print*,'suppressing surface permittivity (y or n)?'
       read 10,q2
       print*,'input data points from short to long wavelength'
       do 30 m=1,dnum
       print*,'input gain, phase, load capacitance and wavelength'
       read 20,egp(m),ylode(m),wlen(m)
  20   format(F6.2,F7.2,F6.2,F9.3)
       print*,' wavelength=',wlen(m)
       print*,'gain and phase=',egp(m),' yload=',ylode(m)
c      print*,'input weighting for thickness and e1'
c      read 27,wt(1,m),wt(2,m)
c 27        format(2F4.0)
c      print*,wt(1,m),wt(2,m)
  30       continue
       print*,'input gtol (in dB)'
       read 15,gtol
       print*,'input ptol (in degrees)'
       read 15,ptol
     print*,'do you wish to guess the surface and bulk permittivities?
    x           (y or n)'
       read 10,q3
       if (q3.eq.'y') then
          print*,'input surface perm.'
          read 16,seguess
          print*,'surface perm. guess ',seguess
               do 29 m=1,nvar-1
               print*,'input permittivity from upper to lower layer'
               read 16,eguess(m)
               print*,'permittivity guess ',eguess(m)
  29               continue
          flag1=1
       else
          flag1=0
       end if
c
c    write the relevant data into the output file
c
       write (3,160) k,N
  160  format ('# of pts. = ',I3,2x,' # of summation terms = ',I4)
c      write (3,163) E0
c 163       format ('ambient DC field E0*lambda/V0 = ',F3.2)
       write (3,165) ef
       write (3,777) gtol,ptol
  777       format ('gtol = ',F5.3,' ptol = ',F5.3)
  165       format ('freq. = ',1PE10.3)
```

```
      if (nvar.eq.2) then
         write (3,281) dguess
  281          format ('d(2)= ',1PE8.2)
         write (3,167) e(2)
  167           format ('e2 = ',1PE8.2,' s2 = ',1PE8.2)
      end if
      write (3,272) scale
  272      format ('for derivative comp. scale = ',F5.3)
      do 148 m=1,dnum
      write (3,168) wlen(m)
  168      format('wlen = ',1PE10.3)
      write (3,768) egp(m),ylode(m)
  768      format('gain = ',F6.2,' phase = ',F7.2,' yload = ',F6.2)
      write (3,169)
  169 format ('**************************************')
  148      continue
      iter=1
      yl=cmplx(YLOAD,0.0)
      rsea=cmplx(0.0,0.0)
      do 141 m1=1,num
      rse(m1)=cmplx(0.0,0.0)
  141      continue
c
c     allow for guesses to be input
c
      if (flag1.eq.1) then
         write (3,283)
  283          format ('guess input')
         rsenew=seguess/eox
         do 70 m1=1,nvar-1
         renew(m1)=eguess(m1)/eox
  70         continue
      else
         write (3,282)
  282          format ('no guess input')
         rsenew=cmplx(0.0,8.854E-12*2*PI*ef*wlen(1)/4)/eox
         do 71 m1=1,nvar-1
         renew(m1)=cmplx(9.07E-12,8.854E-12*2*PI*ef)/eox
  71         continue
      end if
c
c     scale measured gain--phase responses
c
      do 87 m=1,dnum
        egp(m)=cmplx(real(egp(m))/20.0,TLOGe*PI*aimag(egp(m))/180.0)
  87     continue
      count=0
c
```

```
c       compute first gain-phase response
c
  120       d(2)=dguess
        if (nvar.eq.2) then
           renew(2)=e(2)/eox
        end if
        print*,' thickness=',d(2)
        print*,' se2=',eox*rsenew
        print*,' e1=',eox*renew(1)
        print*,' e2=',eox*renew(2)
        write (3,*) d(2)
        write (3,*) eox*rsenew
        write (3,*) eox*renew(1)
        write (3,*) eox*renew(2)
        totalerrnew=0.0
        do 73 m=1,dnum
        yl=cmplx(ylode(m),0.0)
        LAM=wlen(m)
        h=h1/LAM
        d(2)=dguess/LAM
        rse(2)=rsenew/LAM
        rse(2)=cmplx(real(rse(2)),-aimag(rse(2))/(2*PI*ef))
           do 142 m1=1,num
           re(m1)=cmplx(real(renew(m1)),-aimag(renew(m1))/(2*PI*ef))
  142           continue
        rea=re(2)
        call scapest1(d,C)
         call gain1(C,A,X,V,L,gp)
        gp=cmplx(real(gp)/20.0,TLOGe*PI*aimag(gp)/180.0)
        Gold(m)=gp
        err(m)=egp(m)-Gold(m)
        print*,'err(',m,')= ',20*real(err(m)),
     x              180*aimag(err(m))/(PI*TLOGe)
        totalerrnew=totalerrnew+cabs(err(m))
c       write (3,*) gp
  73        continue
        print*,'iter=',iter
        print*,'totalerrnew= ',totalerrnew
        iter=iter+1
c
c   go back and try again
c
c       breakout if good enough
c
        chk=0
        do 146 m=1,dnum
        if ((abs(real(err(m))).lt.(gtol/20.0)).and.
     x    (abs(aimag(err(m))).lt.(ptol*TLOGe*PI/180.0))) then
```

```
            chk=chk
         else
            chk=1+chk
         end if
  146       continue
         print*,'chk=',chk
         if (chk.eq.0) then
            goto 193
         end if
c                .
c        if error is not decreasing go back and try
c        using half of the original step size
c        (should reold and dold be updated?)
c
         if ((totalerrold.lt.totalerrnew).and.(iter.ne.2)) then
            count=count+1
            if (count.eq.1) then
c                goto 1000
c           else if (count.eq.2) then
                 print*,'go other way'
                 write (3,203)
  203                 format ('go other way')
                 rsenew=2*rseold-rsenew
                 do 35 m=1,nvar-1
                 renew(m)=2*reold(m)-renew(m)
  35              continue
                  totalerrold=totalerrnew
                 call bound(renew,reold,rsenew,rseold,eox,nvar,wlen)
                 rseold=rsenew
                 do 179 m1=1,nvar-1
                 reold(m1)=renew(m1)
  179                 continue
            else
                 count=0
                 goto 1000
            end if
            goto 1001
         end if
         count=0
c
c        update old registers
c
 1000       totalerrold=totalerrnew
         rseold=rsenew
         do 34 m=1,nvar-1
         reold(m)=renew(m)
  34        continue
c
```

```
c       next guess is scale times previous one
c       to obtain sensitivity
c       change only variable being searched
c       first change complex surface permittivity
c
        do 111 m=1,dnum
        yl=cmplx(ylode(m),0.0)
        LAM=wlen(m)
        d(2)=dguess/LAM
        h=h1/LAM
        rse(2)=rsenew/LAM
        rse(2)=scale*cmplx(real(rse(2)),-aimag(rse(2))/(2*PI*ef))
           do 31 m1=1,num
           re(m1)=cmplx(real(renew(m1)),
     x                -aimag(renew(m1))/(2*PI*ef))
  31       continue
        rea=re(2)
        call scapest1(d,C)
        call gain1(C,A,X,V,L,gp)
        gp=cmplx(real(gp)/20.0,TLOGe*PI*aimag(gp)/180.0)
        Gpnew(m,1)=gp
        rsslope(m)=(.25*wlen(1))*
     x               (Gpnew(m,1)-Gold(m))/((scale-1)*rseold)
        jac(m,1)=rsslope(m)
c       print*,'deltaG=',Gpnew(m,1)-Gold(m)
c       print*,'jac(',m,'1)= ',jac(m,1)
        if ((m.ne.1).and.(m.le.nvar)) then
           do 76 m1=1,dnum
c
c       must tailor this section for each estimation
c       now changing appropriated complex bulk permittivity
c
              do 145 i=1,nvar-1
              if (i.eq.(m-1)) then
                 re(i)=scale*cmplx(real(renew(i)),
     x                    -aimag(renew(i))/(2*PI*ef))
              else
               re(i)=cmplx(real(renew(i)),-aimag(renew(i))/(2*PI*ef))
              end if
  145           continue
c
          yl=cmplx(ylode(m1),0.0)
          LAM=wlen(m1)
          d(2)=dguess/LAM
          h=h1/LAM
          rse(2)=rsenew/LAM
          rse(2)=cmplx(real(rse(2)),-aimag(rse(2))/(2*PI*ef))
          rea=re(2)
```

```
            call scapest1(d,C)
            call gain1(C,A,X,V,L,gp)
            gp=cmplx(real(gp)/20.0,TLOGe*PI*aimag(gp)/180.0)
            Gpnew(m1,m)=gp
            reslope(m1,m-1)=(Gpnew(m1,m)-Gold(m1))/((scale-1)*re(m-1))
            jac(m1,m)=reslope(m1,m-1)
c           print*,'jac(',m1,m,')= ',jac(m1,m)
  76          continue
         end if
  111       continue
c
c        take transpose of jacobian, tjac
c
            do 180 m=1,nvar
                 do 181 m1=1,dnum
                 tjac(m,m1)=jac(m1,m)
c           print*,'tjac(',m,m1,')= ',tjac(m,m1)
  181                continue
  180            continue
c
c        multiply tjac and jac as in eq. (4.10)
c
         call matmpy1(tjac,jac,pjac,nvar,dnum,nvar)
c
c        read the jacobian into columnar form
c        for matrix inversion routine
c
         do 143 m=1,nvar
            do 144 m1=1,nvar
            pjac1(nvar*(m1-1)+m)=pjac(m,m1)
c        print*,'pjac',m,m1,')=',pjac(m,m1)
  144           continue
  143         continue
c
c        invert product tjac*jac
c
         call minv1(pjac1,nvar,det)
c
c        read inverse back into multidimensional form
c
            do 156 m=1,nvar
                 do 157 m1=1,nvar
                 ijac(m,m1)=pjac1(nvar*(m1-1)+m)
c        print*,'ijac(',m,m1,')=',ijac(m,m1)
  157                continue
  156             continue
c
c        check inverse
```

```
c
c       print*,'ident'
c       call matmpy1(ijac,pjac,temp2,nvar,nvar,nvar)
c       do 136 m=1,nvar
c          do 137 m1=1,nvar
c          print*,' ident(',m,m1,')= ',temp2(m,m1)
c          print*,' jac(',m,m1,')= ',jac(m,m1)
c          print*,' ijac(',m1,m,')= ',ijac(m1,m)
c          read 138,jac(m,m1)
c  138           format (2F6.0)
c  137           continue
c  136        continue
c
c       multiply inverse ijac by tjac as in eq. (4.9)
c
        call matmpy1(ijac,tjac,pjac,nvar,nvar,dnum)
c
c       compute increments as in eq. (4.9)
c
        do 77 m=1,nvar
           do 78 m1=1,dnum
           if (m.eq.1) then
              rsenew=rsenew+(.25*wlen(1))*
     x                          cmplx(0.0,aimag(pjac(1,m1)*err(m1)))
              if (q2.eq.'y') rsenew=cmplx(0.0,aimag(rsenew))
           end if
           if (m.ne.1) then
             re(m-1)=cmplx(real(renew(m-1)),-aimag(renew(m-1))/(2*PI*ef))
              re(m-1)=re(m-1)+pjac(m,m1)*err(m1)
                renew(m-1)=cmplx(real(re(m-1)),-2*PI*ef*aimag(re(m-1)))
           end if
  78          continue
  77        continue
 1001        print*,'delta se = ',eox*(rsenew-rseold)
        do 79 m=1,nvar-1
        print*,'delta re(',m,')= ',eox*(renew(m)-reold(m))
  79        continue
c
c       check that updated guesses are within physical bounds
c
        call bound(renew,reold,rsenew,rseold,eox,nvar,wlen)
        goto 120
c
c       done
c
  193   print*,'final surface perm. =',eox*rsenew
        write (3,195) eox*rsenew
  195        format('final surface perm. = ',2E10.2)
```

```
      do 84 m=1,nvar-1
      print*,'final permittivity= ',eox*renew(m)
      write (3,196) eox*renew(m)
196       format('final permittivity = ',2E10.2)
84       continue
194       end

c
c     set bounds
c
      subroutine bound(renew,reold,rsenew,rseold,eox,nvar,wlen)
      complex renew(100),reold(100),rsenew,rseold
      real eox,wlen(10)
      integer nvar,m
      do 83 m=1,nvar-1
      if (real(renew(m)).lt.(8.854E-12/eox)) then
         print*,'e min'
c        renew(m)=cmplx(real(reold(m)),aimag(renew(m)))
         renew(m)=cmplx(8.854E-12/eox,aimag(renew(m)))
      end if
      if (aimag(renew(m)).lt.0.0) then
         print*,'s min'
c        renew(m)=cmplx(real(renew(m)),aimag(reold(m)))
         renew(m)=cmplx(real(renew(m)),0.0)
      end if
      if (real(renew(m)).gt.(1E-9/eox)) then
         print*,'e max'
         renew(m)=cmplx(real(reold(m)),aimag(renew(m)))
      end if
      if (aimag(renew(m)).gt.(1E-7/eox)) then
         print*,'s max'
         renew(m)=cmplx(real(renew(m)),aimag(reold(m)))
      end if
83       continue
      if (real(rsenew).lt.0.0) then
         print*,'se real min'
c        rsenew=cmplx(real(rseold),aimag(rsenew))
         rsenew=cmplx(0.0,aimag(rsenew))
      end if
      if (aimag(rsenew).lt.0.0) then
         print*,'se imag min'
         rsenew=cmplx(real(rsenew),aimag(rseold))
      end if
      if (real(rsenew).gt.(1E-14/eox)) then
         print*,'se real max'
         rsenew=cmplx(real(rseold),aimag(rsenew))
      end if
      if (aimag(rsenew).gt.(1E-9/eox)) then
```

```
      print*,'se imag max'
      rsenew=cmplx(real(rsenew),aimag(rseold))
   end if
   return
   end
```

```
c                th4est.for
c                  8/22/87
c
c    estimates layer thickness and permittivity from
c    two high freq. gains with different bulk permittivities
c    above, i.e. air and oil
c    uses Newton method (Section 4.4)
c    can perform variable collocation technique
c    stops after 50 iterations for a given # of grid pts.
c
        program th4est

        complex A(50,50),X(50),V(50),L(1000),Lmin,y11,y12,yl
        complex e(100),se(100),re(100),rse(100),C(1000)
        complex ea,sea,rea,rsea
        real i1,index(60),inc,freq(60)
        real LAM,eox,YLOAD,h,g,En0,d(100),w1
c       always need these guys
        common /comp/rea,rsea
        common /comp1/Lmin,y11,y12,yl
        common /flot/h,g,eox,LAM,YLOAD
        common /flot1/En0
        common /tin1/N1,N2
c       need for homogeneous layers
        common /comp2/e,se,re,rse
        common /flot2/w1
        common /tin2/num
c       need for numerical aspect
        common /tin/k,N,nmax
        complex eps(2,100),gp
        real HFG(2),dnew,Gnew(2),renew,jac(2,2)
        real dguess,e2guess,det,err(2)
        character q,q1,ofile*10
        integer f,m,chk,dnum,flag1,iter,ITMAX,loop
        PI=3.141592654
c       max. number of collocation pts.
        N1=30
c       no. of summation terms (fourier modes)
        N2=1000
        ITMAX=50
c
c            I/O
c
        print*,'output filename? '
        read 10,ofile
   10       format(10a)
        open(unit=3,file=ofile)
c
```

```
c       get info about numerical aspect
c
        print*,'input # of summation (fourier) terms '
        read 5,N
   5        format(I4)
        print*,' do you wish computer to automatically
     x          perform search over variable grid pnts?
     x          (y or n)'
        read 10,q1
        if (q1.eq.'n') then
           print*,'input # of collocation pnts. '
           read 5,k
        else
           k=2
        end if
c
c       get info about chip
c
        call chipinfo()
c
c       print*,'input normalized ambient DC field (E0*lambda/V0) '
c       read 15,E0
c       E0 represents DC field
        E0=0.0
c
c       get info on known properties of layers
c
        num=2
        d(1)=10000
        print*,d(1)
        print*,'input number of gain-epsilon pairs '
        read 5,dnum
        do 30 m=1,dnum
        print*,'for pair #',m
        print*,'input layer 1 epsilon '
        read 20,eps(m,1)
  20    format(E8.0,E8.0)
        print*,'eps(',m,',1)=',eps(m,1)
        print*,'input measured high freq. gain (signed) '
        read 15,HFG(m)
   15        format(E8.0)
        print*,'HFG(',m,')=',HFG(m)
  30        continue
        print*,'do you wish to guess a thickness and permittivity?
     x            (y or n)'
        read 10,q
        if (q.eq.'y') then
           print*,'input thickness'
```

```
            read 15,dguess
            print*,'thickness guess ',dguess
            print*,'input permittivity'
            read 15,e2guess
            print*,'permittivity guess ',e2guess
            flag1=1
         else
            flag1=0
         end if
c
c        setup output file
c
         write (3,160) k,N
  160    format ('# of pts. = ',I3,2x,' # of summation terms = ',I4)
         write (3,163) E0
  163         format ('ambient DC field E0*lambda/V0 = ',F3.2)
         do 148 m=1,dnum
         write (3,170) real(eps(m,1)),HFG(m)
  170          format ('permittivity = ',1PE8.2,5x,' HFG = ',0PF6.2)
  148         continue
         write (3,171)
  171    format(6x,'thickness',9x,'permittivity')
c
         yl=cmplx(YLOAD,0.0)
c
c        normalize
c
         do 141 m=1,num
         rse(m)=cmplx(0.0,0.0)
  141         continue
c        setup first guess
         if (flag1.eq.1) then
            dnew=dguess/LAM
            renew=e2guess/eox
         else
              dnew=.02
            renew=9.07E-12/eox
         end if
c
c          for variable collocation point technique
c        each successive loop represents estimating using
c        a greater number of collocation points
c
         loop=1
  120         iter=1
         d(2)=dnew
         re(2)=cmplx(renew,0.0)
         write (3,*) (LAM*d(2)),eox*real(re(2))
```

```
        print *,' thickness=',LAM*d(2),' epsilon=',eox*real(re(2))
c
c   compute first set of gains
c
        do 142 m=1,dnum
          re(1)=eps(m,1)/eox
         call scapest1(d,C)
        call gain1(C,A,X,V,L,gp)
        Gnew(m)=real(gp)
        err(m)=HFG(m)-Gnew(m)
        print*,'err(',m,')= ',err(m)
  142        continue
        print*,'iter=',iter
        iter=iter+1
c
c   check if can breakout
c   error tolerance of within .05 dB and .5%
c
        chk=0
        do 146 m=1,dnum
        if ((abs(err(m)/HFG(m)).lt.(.005)).and.
     x    (abs(err(m)).lt.(.05))) then
           chk=chk
        else
           chk=1+chk
        end if
  146        continue
        print*,'chk=',chk
        if (chk.eq.0) then
           goto 193
        end if
c
c    no breakout
c    must compute jacobian slopes
c    first get sensitivities to thickness
c
       d(2)=1.1*dnew
       do 143 m=1,dnum
         re(1)=eps(m,1)/eox
        call scapest1(d,C)
       call gain1(C,A,X,V,L,gp)
       jac(m,1)=(Gnew(m)-real(gp))/(.1*dnew)
  143        continue
c
c    then get sensitivities to permittivity
c
       d(2)=dnew
       re(2)=cmplx(1.1*renew,0.0)
```

```
        do 144 m=1,dnum
          re(1)=eps(m,1)/eox
         call scapest1(d,C)
        call gain1(C,A,X,V,L,gp)
        jac(m,2)=(Gnew(m)-real(gp))/(.1*renew)
  144         continue
c
c       compute increment using jacobian as shown in Section 4.4
c       eq. (4.27), then increment thickness and permittivity estimates
c
        det=jac(1,1)*jac(2,2)-jac(1,2)*jac(2,1)
        dnew=dnew-(jac(2,2)*err(1)-jac(1,2)*err(2))/det
        renew=renew-(jac(1,1)*err(2)-jac(2,1)*err(1))/det
c
c    check that the updated guesses are within physical bounds
c
        if (renew.lt.(8.854E-12/eox)) then
           print*,'e2 min bound'
           renew=8.854E-12/eox
        end if
        if (renew.gt.(8.854E-10/eox)) then
           print*,'e2 max bound'
           renew=8.854E-10/eox
        end if
        if (dnew.lt.0.0) then
           print*,'d min bound'
           dnew=.02
        end if
        if (dnew.gt..333) then
           print*,'d max bound'
           dnew=.125
        end if
c
c    check that routine is not off in
c    outer space
c
        if (iter.eq.ITMAX) then
           print*,'exceeded max iterations, cannot converge'
           write (3,197)
  197           format('exceeded max iterations, cannot converge')
           goto 194
        end if
        goto 120
c
c      loop if coarse grid (k<25) and using variable technique
c
  193       if ((q1.eq.'y').and.(k.ne.25)) then
          loop=loop+1
```

```
            if (loop.eq.2) then
                 k=10
            else
                 k=25
            end if
            write (3,201) k
  201            format ('k=',I3)
            print*,'k=',k
            goto 120
          end if
c
c         done
c
          print*,'final thickness =',LAM*1E6*d(2),' microns'
          write (3,195) LAM*1E6*d(2)
  195         format('final thickness = ',F8.3)
          print*,'final permittivity= ',eox*real(re(2))
          write (3,196) eox*real(re(2))
  196         format('final permittivity = ',1PE8.2)
  194         end
```

```
c                hest.for
c                 8/27/87
c
c    calibration for microchip (Section 5.1)
c    estimates oxide layer thickness given high freq. gain in air
c    starts with computing initial slopes
c    uses secant method
c
        program hest

        complex A(50,50),X(50),V(50),L(1000),Lmin,y11,y12,yl
        complex e(100),se(100),re(100),rse(100),C(1000)
        complex ea,sea,rea,rsea
        real i1,freq
        real LAM,eox,YLOAD,h,g,En0,d(100),w1
        integer f,N1,N2
c       always need these guys
        common /comp/rea,rsea
        common /comp1/Lmin,y11,y12,yl
        common /flot/h,g,eox,LAM,YLOAD
        common /flot1/En0
        common /tin1/N1,N2
c       need for homogeneous layers
        common /comp2/e,se,re,rse
        common /flot2/w1
        common /tin2/num
c       need for numerical aspect
        common /tin/k,N,nmax
        complex gp
        real HFG,Gnew,Gold,slope,hold
        character ofile*10
        integer m
        PI=3.141592654
c       max. number of collocation pts.
        N1=50
c       no. of summation terms (fourier modes)
        N2=1000
c
c            I/O
c
        print*,'output filename? '
        read 10,ofile
   10       format(10a)
        open(unit=3,file=ofile)
c
c       get info about numerical aspect
c
        print*,'input # of summation (fourier) terms '
```

```
      read 5,N
    5     format(I3)
c     set number of collocation points
      k=25
c
c     get info about chip
c
      call chipinfo()
c
c     print*,'input normalized ambient DC field (E0*lambda/V0) '
c     read 15,E0
   15       format(E9.0)
c     E0 represents DC field
      E0=0.0
c
c     infinite half space is air
c
      num=1
      d(1)=10000
      e(1)=cmplx(8.854E-12,0.0)
      print*,'input measured high freq. gain (signed) '
      read 15,HFG
      print*,'HFG=',HFG
c
c     setup output file
c
      write (3,160) k,N
  160 format ('# of pts. = ',I3,2x,' # of summation terms = ',I4)
      write (3,163) E0
  163      format ('ambient DC field E0*lambda/V0 = ',F3.2)
      write (3,165) real(e(1))
  165      format ('e1= ',1PE8.2)
      write (3,166) HFG
  166      format ('HFG= ',F6.2)
c
      yl=cmplx(YLOAD,0.0)
c
c     normalize
c
      do 140 m=1,num
      re(m)=e(m)/eox
      rse(m)=cmplx(0.0,0.0)
  140      continue
      m=1
        call scapest1(d,C)
      call gain1(C,A,X,V,L,gp)
      print *,' thickness=',LAM*(1E6)*h,' gain=',real(gp)
      write (3,*) (LAM*(1E6)*h),real(gp)
```

```
         Gnew=real(gp)
c
c        breakout if gain error is both less than .5%
c        and within .03 dB, else
c            go back and try again
c
         if ((abs(1-Gnew/HFG).lt.(.005)).and.
     x           (abs(HFG-Gnew).lt.(.03))) then
            goto 193
         end if
         Gold=Gnew
         hold=h
c
c        second calculation
         m=2
           h=1.01*hold
  130         call scapest1(d,C)
         call gain1(C,A,X,V,L,gp)
         print *,' thickness=',LAM*(1E6)*h,' gain=',real(gp)
         write (3,*) (LAM*(1E6)*h),real(gp)
         Gnew=real(gp)
         slope=(Gnew-Gold)/(h-hold)
c
c    on flat part of curve (for microchip)
c
         if (abs(slope).le.1) then
            print*,'at limit of sensitivity '
            goto 193
         end if
c
c        breakout if ok else
c            go back and try again
c
         if ((abs(1-Gnew/HFG).lt.(.005)).and.
     x           (abs(HFG-Gnew).lt.(.03))) then
            goto 193
         end if
c        update registers
         hold=h
           Gold=Gnew
c
c        increment guess using eq. (4.3)
c
         h=h+(HFG-Gnew)/slope
         goto 130
c
c        done
c
```

```
193   print*,'final thickness=',LAM*(1E6)*h,' microns'
      write (3,195)
195        format('final thickness (um)')
      write (3,*) LAM*(1E6)*h
194        end
```

```
c       gain1.for
c
c       control program for scap1,coef1 or coef1a,solv,admit1 and bodex1
c       for layers above electrodes
c
        subroutine gain1(C,A,X,V,L,gp)
        parameter (PI=3.141592654)
        complex gp,C(N2)
        complex A(N1,N1),X(N1),V(N1),L(N2),Lmin,y11,y12,yl
        real LAM,eox,YLOAD,h,g
        integer m
        common /comp1/Lmin,y11,y12,yl
        common /flot/h,g,eox,LAM,YLOAD
        common /tin/k,N,nmax
        common /tin1/N1,N2
c
c       compute L(n) - the parallel addition of surface capacitance
c       densities above and below the electrodes where
c       Cbelow=-1/tanh(2mh*PI) from eq. (3.1)
c
                do 100 m=1,N2
                L(m)=(1/tanh(2*PI*m*h))+C(m)/(2*PI*m)
c               print*,'L(',m,')=',L(m)
c       stop calculating if approaching limit for large m
c
                if (m.gt.1) then
                   if (cabs(L(m-1)-L(m)).lt.1E-5) then
                        go to 58
                   end if
                end if
  100                continue
   58                nmax=m-1
c               print*,'nmaxL=',nmax
                Lmin=L(nmax)
c       fill up rest of matrix with Lmin
                do 59 m=nmax+1,N2
                L(m)=Lmin
   59                continue
c
c     calculate unknown grid voltages and thus, field distribution
c
        call coef1(A,X,L)
        call solve(A,X,V)
        call admit1(V,L)
        call bodex1(gp)
c         uncomment to print voltage distribution, elements of
c       A and X matrices, and y11 and y12
c       write (3,157)
```

```
c 157        format ('V(m)')
c       do 155 m=1,k
c          do 156 m1=1,k
c          print*,'A(',m,',',m1,')=',A(m,m1)
c 156           continue
c       print*,'X(',m,')=',X(m)
c       print*,'V(',m,')=',V(m)
c         write (3,*) V(m)
c 155       continue
c       print*,' y11=',y11
c       print*,' y12=',y12
c       print*,' yl=',yl
c       write (3,*) y11
c       write (3,*) y12
c       write (3,*) yl
        print*,' gain, phase =',gp
        return
        end
```

```
c
c               coef1.for
c
c     this program computes the matrix elements
c     used to solve AV=X using eqs. (2.21-2.24)
c     and approach of Appendix A.
c
        subroutine coef1(A,X,L)
        parameter (PI=3.141592654)
        complex A(N1,N1),X(N1),L(N2),Lmin,y11,y12,yl
        complex temp,temp1,fhsum,rea,rsea
        integer flag,r,c
        real h,g,En0,f1,grid,y1,y0,eox,LAM
        common /comp1/Lmin,y11,y12,yl
        common /comp/rea,rsea
        common /flot1/En0
        common /flot/h,g,eox,LAM,YLOAD
        common /tin/k,N,nmax
        common /tin1/N1,N2
        flag=0
        dpi=2.0*PI
        y0=grid(0)
        y1=grid(1)
c
c     this first section computes A1(i,j) according to eq. (2.22a)
c     this computation utilizes the "symmetry"
c     of A1(i,j) - namely, that A1(1,1)=A1(k,k), A1(1,2)=A1(k,k-1)
c     and so on, back to the halfway point.
c     also, the infinite sum of C(n)sin(nx)/n^2 is transformed to a
c     more rapidly converging series using fhsum.for and fsum.for
c     also compute X1(i,j) here according to eq. (2.24a)
c
        do 500 r=1,k
        if (r.eq.k) then
                s1=grid(k+1)
        else
                s1=(grid(r+1)+grid(r))/2.0
        end if
        if (r.eq.1) then
                s2=y0
        else
                s2=(grid(r)+grid(r-1))/2.0
        end if
        if ((mod(k,2).eq.0).and.(r.eq.(k/2+1))) flag=1
           do 510 c=1,k
           if ((mod(k,2).eq.1).and.(r.eq.((k+1)/2)).and
     x             .(c.eq.((k+3)/2))) flag=1
           if (flag.eq.1) then
```

```
                A(r,c)=A(k+1-r,k+1-c)
            else
                da= 1.0/(grid(c+1)-grid(c))
                db= 1.0/(grid(c)-grid(c-1))
                temp1=cmplx(0.0,0.0)
                temp=fhsum(dpi*(s1+grid(c)),L)
c               print*,'temp=',temp
                temp1=temp+temp1
                temp=fhsum(dpi*(s1-grid(c)),L)
                temp1=temp+temp1
                temp=fhsum(dpi*(s2+grid(c)),L)
                temp1=temp1-temp
                temp=fhsum(dpi*(s2-grid(c)),L)
                temp1=temp1-temp
                A(r,c)=temp1*(da+db)
c               print*,' 1A(',r,',',c,')=',A(r,c)
                temp1=cmplx(0.0,0.0)
                temp=fhsum(dpi*(s2+grid(c+1)),L)
                temp1=temp+temp1
                temp=fhsum(dpi*(s2-grid(c+1)),L)
                temp1=temp+temp1
                temp=fhsum(dpi*(s1+grid(c+1)),L)
                temp1=temp1-temp
                temp=fhsum(dpi*(s1-grid(c+1)),L)
                temp1=temp1-temp
                A(r,c)=da*temp1+A(r,c)
c               print*,' 2A(',r,',',c,')=',A(r,c)
                temp1=cmplx(0.0,0.0)
                temp=fhsum(dpi*(s2+grid(c-1)),L)
                temp1=temp+temp1
                temp=fhsum(dpi*(s2-grid(c-1)),L)
                temp1=temp+temp1
                temp=fhsum(dpi*(s1+grid(c-1)),L)
                temp1=temp1-temp
                temp=fhsum(dpi*(s1-grid(c-1)),L)
                temp1=temp1-temp
                A(r,c)=db*temp1+A(r,c)
c               print*,' 3A(',r,',',c,')=',A(r,c)
                sq= PI**2
                A(r,c)=(A(r,c))/(2*sq)
c               print*,' 4A(',r,',',c,')=',A(r,c)
             end if
  510 continue
        temp1=cmplx(0.0,0.0)
        temp=fhsum(dpi*(s1+y0),L)
        temp1=temp+temp1
        temp=fhsum(dpi*(s1-y0),L)
        temp1=temp+temp1
```

```
        temp=fhsum(dpi*(s2+y1),L)
        temp1=temp+temp1
        temp=fhsum(dpi*(s2-y1),L)
        temp1=temp+temp1
        temp=fhsum(dpi*(s2+y0),L)
        temp1=temp1-temp
        temp=fhsum(dpi*(s2-y0),L)
        temp1=temp1-temp
        temp=fhsum(dpi*(s1+y1),L)
        temp1=temp1-temp
        temp=fhsum(dpi*(s1-y1),L)
        temp1=temp1-temp
        sq=PI*PI
        X(r)=(-1.0/(2.0*sq*(y1-y0)))*temp1
c       print*,' X(',r,')=',X(r)
  500        continue
c
c    this section computes A2(r,c), A3(r,c), X2(r,c) and X3(r,c)
c
        do 520 r=1,k
        if(r.eq.1) then
           dc=(.5)*(grid(2)+y1)-y0
        else if(r.eq.k) then
           dc=grid(k+1)-(.5)*(grid(k)+grid(k-1))
        else
           dc=(.5)*(grid(r+1)-grid(r-1))
        end if
c
c    X2(r) as in eq. (2.24b) in report
c
        X(r)=X(r)-dc*(y0+y1)/h-dc*En0*rea
           do 530 c=1,k
           da=grid(c+1)-grid(c-1)
c
c     A2(r,c) and A3(r,c) as in eqs. (2.22b) and (2.22c)
c
           A(r,c)=A(r,c)+(dc*da/h)+((f1(r,c))*rsea)
c           print*,' A(',r,',',c,')=',A(r,c)
  530      continue
c
c     X3(r) as in eq. (2.24c)
c
         if(r.eq.1) then
          X(r)=X(r)+(1/(y1-y0))*rsea
c         print*,'X=',X(1)
        end if
c       print*,' X(',r,')=',X(r)
  520        continue
```

```
      return
      end
c
c    compute location of k collocation points
c  presently uses a cosinusoidal distribution
      real function grid(i)
      parameter (PI=3.141592654)
      common /flot/h,g,eox,LAM,YLOAD
      common /tin/k,N,nmax
      real h,g,eox,LAM,YLOAD
      integer k,N,nmax
      grid=.25-(.5)*g*cos(PI*i/(k+1))
      return
      end
c
c   computes f1(r,c) (the delta functions) for surface contribution
c   A3(r,c)
c
      function f1(r,c)
      integer r,c
      da=1.0/(grid(r+1)-grid(r))
      db=1.0/(grid(r)-grid(r-1))
      dc=grid(r+1)-grid(r-1)
      if (r.eq.c) then
         f1=dc*db*da
      else if((r-c).eq.1) then
         f1=(-1.0)*db
      else if ((c-r).eq.1) then
         f1=(-1.0)*da
      else
         f1=0.0
      end if
      return
      end
c              FHSUM.for
c     This program computes the sum C(n)sin(nx)/(n*n)
c  using fsum.c and the fact that for a large enough n
c  C(n) approaches a fixed value.  This program computes the
c  original series sum until n=nmax and then computes the
c  rest by using fsum.c and subtracting the first 20 terms
c  of sin(nx)/(n*n). See Appendix A.
c
      complex function fhsum(fx,L)
      complex L(N2),Lmin,y11,y12,yl
      complex sum,L1
      common /comp1/Lmin,y11,y12,yl
      common /tin/k,N,nmax
      common /tin1/N1,N2
```

```
      real fx,fy
      sum=cmplx(0.0,0.0)
      do 400 i=1,nmax
      L1=L(i)-Lmin
      fy=(sin(i*fx))/i**2
        sum=sum+fy*L1
400       continue
      fhsum=sum+Lmin*fsum(fx)
c     print*,'sum=',sum
c     print*,'Lmin=',Lmin
c     print*,'fsum(fx)=',fsum(fx)
      return
      end
c                  FSUM.for
c          This program computes sin(nx)/(n*n) using an
c   series which converges more rapidly and with higher
c   accuracy.  The series is obtained by first noting that
c   sin(nx)/(n*n) equals the integral of ln[2sin(t/2)] from
c   0 to x.  Then use a series expansion of ln[sin(x)] and
c   integrate term by term.   This series is computed until
c   the last term is .01% of the sum. See Appendix A.
c
      function fsum(fsx)
      dimension Bern(20)
       real i1,fsx
      integer sign
c     Bernoulli numbers
      Bern(1)=.166667
      Bern(2)=.033333
      Bern(3)=.023809
      Bern(4)=.033333
      Bern(5)=.075757
      Bern(6)=.253113
      Bern(7)=1.166666
      Bern(8)=7.092156
      Bern(9)=54.971177
      Bern(10)=529.124242
      Bern(11)=6192.123188
      Bern(12)=8.658025311e4
      Bern(13)=1.425517167e6
      Bern(14)=2.729823107e7
      Bern(15)=6.015808739e8
      Bern(16)=1.51163157e10
      Bern(17)=4.29614643le11
      Bern(18)=1.371165521e13
      Bern(19)=4.883323190e14
      Bern(20)=1.929657934e16
      if (fsx.lt.0) then
```

```
            z=abs(fsx)
            sign=(-1)
         end if
         if (fsx.gt.0) then
            z=fsx
            sign=1
         end if
          if (fsx.eq.0) then
            a=0.0
            sign=1
         else
            a=z*(alog(z)-1.0)
            b=a
            i=1
  440       if(abs(b/a).le.(.0001)) go to 430
            i1=i*1.0
            c=(Bern(i))*z**(2*i1+1)
            d=2.0*i1*(2.0*i1+1.0)*fact(2*i)
            b=c/d
            a=a-b
            i=i+1
            go to 440
  430    continue
           end if
         fsum=(-a)*sign
         return
         end
c
c      fact(y)
c
c    computes factorials
c
         function fact(y)
         integer y
         y1=1.0
         do 420 i=0,y-1
         y1=y1*(y-i)
  420         continue
         fact=y1
         return
         end
```

```
c
c
c               solv.for
c               10/85
c
c  solves k equations for k unknowns using
c  gaussian elimination. Format  AV=X, dimension k x k
c
        subroutine solve(A,X,V)
        complex A(N1,N1),X(N1),V(N1)
        complex b(0:99,0:99),z(0:99),temp(0:99)
        complex tempz,tempr
        common /tin/k,N,nmax
        common /tin1/N1,N2
        integer r,c,k
c             initialize both b & z
        do 500 r=0,k-1
        z(r)=X(r+1)
           do 510 c=0,k-1
             b(r,c)=A(r+1,c+1)
  510           continue
  500   continue
        do 520 i=0,k-2
c            handles the case with a zero leading coeff
           if(cabs(b(i,i)).eq.0) then
               tempz=z(i)
               do 530 c=i,k-1
                 temp(c)=b(i,c)
  530               continue
               do 540 r=i,k-2
               z(r)=z(r+1)
                  do 550 c=i,k-1
                    b(r,c)=b(r+1,c)
  550                  continue
  540               continue
               z(k-1)=tempz
               do 560 c=i,k-1
                 b(k-1,c)=temp(c)
  560               continue
           end if
c            generates the new set of equations
           do 570 r=i+1,k-1
           z(r)=z(r)-z(i)*b(r,i)/b(i,i)
              do 580 c=i+1,k-1
                b(r,c)=b(r,c)-b(i,c)*b(r,i)/b(i,i)
  580              continue
  570      continue
  520   continue
```

```
c           back substitution
        do 590 r=k-1,0,-1
        V(r+1)=z(r)/b(r,r)
           do 600 c=r+1,k-1
             V(r+1)=V(r+1)-b(r,c)*V(c+1)/b(r,r)
  600           continue
  590        continue
        return
        end
```

```
c
c                admit1.for
c
c     this subroutine calculates y11 and
c     y12 given a voltage distribution V. Uses
c     eq. (2.27).
c
        subroutine admit1(V,L)
        parameter (PI=3.141592654)
        complex V(N1),L(N2),Lmin,y11,y12,yl
        real h,En0,g,eox,LAM
        complex temp,temp1,temp2,temp3,rea,rsea
        common /comp1/Lmin,y11,y12,yl
        common /comp/rea,rsea
        common /flot1/En0
        common /flot/h,g,eox,LAM,YLOAD
        common /tin/k,N,nmax
        common /tin1/N1,N2
        y11=cmplx(0.0,0.0)
        y12=cmplx(0.0,0.0)
        temp3=cmplx(1.0,0.0)
        temp2=cmplx(1.0,0.0)
        temp=cmplx(0.0,0.0)
c
c    compute summation in eq. (2.13)
c
        do 600 m=1,N
        temp1=cmplx(0.0,0.0)
        sq=1.0/(PI*PI*m)
c
c    compute phi a (m) using eq. (2.13)
c
           do 610 j=1,k
           da=1.0/(grid(j+1)-grid(j))
           db=1.0/(grid(j)-grid(j-1))
           dc=sq*((da+db)*cos(2*PI*m*grid(j))-
     x  da*cos(2*PI*m*grid(j+1))-db*cos(2*PI*m*grid(j-1)))/m
              temp1=temp1+dc*V(j)
  610           continue
        dc=sq*(cos(2*PI*m*grid(0))-cos(2*PI*m*grid(1)))
     x      /((grid(1)-grid(0))*m)
        temp1=temp1+dc
c
c    temp1=phi a of m
c
c       write (3,630)
c 630        format('phi a of n')
c       write (3,*) temp1
```

```
c
        temp1=temp1*L(m)
        dc= sin(2*PI*m*grid(k+1))
        temp2=dc*temp1
        dc=sin(2*PI*m*grid(0))-sin(2*PI*m*grid(k+1))
        y11=y11+dc*temp1
        y12=y12+temp2
c
c       write (3,631)
c 631         format ('fourier sum for y11')
c       write (3,*) dc*temp1
c       write (3,632)
c 632        format ('fourier sum for y12')
c       write (3,*) temp2
c
        temp=(.000001)*y11
        temp1=(.000001)*y12
  600   continue
        temp1=cmplx(0.0,0.0)
c
c    compute phi a (0) using eq. (2.13)
c
        do 620 j=1,k
        da=grid(j+1)-grid(j-1)
          temp1=temp1+da*V(j)
  620       continue
        temp1=temp1+grid(0)+grid(1)
c
c        temp1= phi a of 0
c       write (3,633)
c 633        format ('phi a of 0')
c       write (3,*) temp1
c
        temp1=temp1/h
        if ((En0.gt.0.0).or.(En0.lt.0.0)) then
                temp1=temp1+En0*rea
        end if
          da=grid(0)-grid(k+1)+(.5)
        y11=y11+da*temp1
c
c       write (3,634)
c 634        format ('y11 0 term')
c       write (3,*) da*temp1
c
        temp=((.5)-grid(k+1))*temp1
        y12=y12-temp
c
c       write (3,635)
```

```
c 635        format ('y12 0 term')
c       write (3,*) temp
c
        temp=(1.0/(grid(k+1)-grid(k)))*V(k)
        temp1=(1.0/(grid(1)-grid(0)))*(1-V(1))
        y11=2*(y11+(temp1-temp)*rsea)
c
c       write (3,636)
c 636        format ('y11 surface term')
c       write (3,*) (temp1-temp)*rsea
c
        temp=temp*rsea
        y12=2*(temp+y12)
c
c       write (3,637)
c 637        format ('y12 surface term')
c       write (3,*) temp
c
        return
        end
```

```
c                       bodex1.for
c                       12/85
c
c   This program converts complex voltage ratio
c   from cartesian to polar form given y11,y12 and yload.
c
        subroutine bodex1(pgp)
        parameter (PI=3.141592654)
        complex Lmin,y11,y12,yl
        complex pgp,temp
        real gain,phase
        common /comp1/Lmin,y11,y12,yl
        temp=y12/(y11+y12+yl)
        gain=20*alog10(cabs(temp))
        phase=180*(atan2(aimag(temp),real(temp)))/PI
c
c    branch cut at negative real axis
c
        if (aimag(temp).gt.0.0) then
           phase=phase-360
        end if
        pgp=cmplx(gain,phase)
        return
        end
```

```
c
c               coef1a.for
c
c     this program computes the matrix elements
c     used to solve AV=X  using eqs. (2.21-2.24).
c     without faster converging series
c
        subroutine coef1a(A,X,L)
        parameter (PI=3.141592654)
        complex A(N1,N1),X(N1),L(N2),Lmin,y11,y12,yl
        complex temp,temp1,fhsum,rea,rsea
        integer flag,r,c
        real h,g,En0,f1,grid,y1,y0,eox,LAM
        common /comp1/Lmin,y11,y12,yl
        common /comp/rea,rsea
        common /flot1/En0
        common /flot/h,g,eox,LAM,YLOAD
        common /tin/k,N,nmax
        common /tin1/N1,N2
        flag=0
        dpi=2.0*PI
        y0=grid(0)
        y1=grid(1)
c
c     this first section computes A1(i,j) according to eq. (2.22a)
c     this computation utilizes the "symmetry"
c     of A1(i,j) - namely, that A1(1,1)=A1(k,k), A1(1,2)=A1(k,k-1)
c     and so on back to the halfway point.
c     also compute X1(i,j) here according to eq. (2.24a)
c
        do 500 r=1,k
        if (r.eq.k) then
                s1=grid(k+1)
        else
                s1=(grid(r+1)+grid(r))/2.0
        end if
        if (r.eq.1) then
                s2=y0
        else
                s2=(grid(r)+grid(r-1))/2.0
        end if
        if ((mod(k,2).eq.0).and.(r.eq.(k/2+1))) flag=1
           do 510 c=1,k
           if ((mod(k,2).eq.1).and.(r.eq.((k+1)/2)).and
     x             .(c.eq.((k+3)/2))) flag=1
           if (flag.eq.1) then
                A(r,c)=A(k+1-r,k+1-c)
           else
```

```
                da= 1.0/(grid(c+1)-grid(c))
                db= 1.0/(grid(c)-grid(c-1))
                temp1=cmplx(0.0,0.0)
                do 77 m1=1,N
              temp1=temp1+(L(m1)/m1**2)*(sin(2*PI*m1*s1)-sin(2*PI*m1*s2))
     x          *((da+db)*cos(2*PI*m1*grid(c))-da*cos(2*PI*m1*grid(c+1))
     x                -db*cos(2*PI*m1*grid(c-1)))
   77              continue
                sq= PI**2
                A(r,c)=temp1/sq
c               print*,' 4A(',r,',',c,')=',A(r,c)
              end if
  510 continue
        temp1=cmplx(0.0,0.0)
        do 78 m1=1,N
        temp1=temp1+(L(m1)/m1**2)*(sin(2*PI*m1*s1)-sin(2*PI*m1*s2))*
     x        (cos(2*PI*m1*y0)-cos(2*PI*m1*y1))
   78        continue
        sq=PI*PI
        X(r)=(-1.0/(sq*(y1-y0)))*temp1
c       print*,' X(',r,')=',X(r)
  500        continue
c
c    this section computes A2(r,c), A3(r,c), X2(r,c) and X3(r,c)
c
        do 520 r=1,k
        if(r.eq.1) then
           dc=(.5)*(grid(2)+y1)-y0
        else if(r.eq.k) then
           dc=grid(k+1)-(.5)*(grid(k)+grid(k-1))
        else
           dc=(.5)*(grid(r+1)-grid(r-1))
        end if
c
c    X2(r) as in eq. (2.24b) in report
c
        X(r)=X(r)-dc*(y0+y1)/h-dc*En0*rea
           do 530 c=1,k
           da=grid(c+1)-grid(c-1)
c
c     A2(r,c) and A3(r,c) as in eqs. (2.22b) and (2.22c)
c
           A(r,c)=A(r,c)+(dc*da/h)+((f1(r,c))*rsea)
c           print*,' A(',r,',',c,')=',A(r,c)
  530      continue
c
c     X3(r) as in eq. (2.24c)
c
```

```
          if(r.eq.1) then
           X(r)=X(r)+(1/(y1-y0))*rsea
         end if
c        print*,' X(',r,')=',X(r)
  520          continue
         return
         end
c
c    compute location of k collocation points
c  presently uses a cosinusoidal distribution
         real function grid(i)
         parameter (PI=3.141592654)
         common /flot/h,g,eox,LAM,YLOAD
         common /tin/k,N,nmax
         real h,g,eox,LAM,YLOAD
         integer k,N,nmax
         grid=(.25)-(.5)*g*cos(PI*i/(k+1))
         return
         end
c
c   computes f1(r,c) (the delta functions) for surface contribution
c   A3(r,c)
c
         function f1(r,c)
         integer r,c
         da=1.0/(grid(r+1)-grid(r))
         db=1.0/(grid(r)-grid(r-1))
         dc=grid(r+1)-grid(r-1)
         if (r.eq.c) then
            f1=dc*db*da
         else if((r-c).eq.1) then
            f1=(-1.0)*db
         else if ((c-r).eq.1) then
            f1=(-1.0)*da
         else
            f1=0.0
         end if
         return
         end
```

```
c       gain2.for
c
c       control program for coef2,solv,admit2 and bodex2
c       for layers above and below electrodes
c
        subroutine gain2(C,A,X,V,L,gp)
        parameter (PI=3.141592654)
        complex gp,C(N2,2),C0(2)
        complex A(N1,N1),X(N1),V(N1),L(N2),Lmin,y11,y12,yl
        real LAM,eox,YLOAD,h,g
        integer m
        common /comp1/Lmin,y11,y12,yl,C0
        common /flot/h,g,eox,LAM,YLOAD
        common /tin/k,N,nmax
        common /tin1/N1,N2
c
c       compute L(n) - the parallel addition of surface capacitance
c       densities above and below the electrodes
c
        do 100 m=1,N2
        L(m)=(C(m,1)-C(m,2))/(2*PI*m)
c       print*,'C(',m,',1)=',C(m,1)
c       print*,'C(',m,',2)=',C(m,2)
c       print*,'L(',m,')=',L(m)
c       stop calculating if approaching limit for large m
c
        if (m.gt.1) then
           if (cabs(L(m-1)-L(m)).lt.1E-5) then
                go to 58
           end if
        end if
  100        continue
   58        nmax=m-1
c       print*,'nmaxL=',nmax
        Lmin=L(nmax)
c       fill up rest of matrix with Lmin
        do 59 m=nmax+1,N2
        L(m)=Lmin
   59        continue
c
c     calculate unknown grid voltages and thus, field distribution
c
        call coef2(A,X,L)
        call solve(A,X,V)
        call admit2(V,L)
        call bodex2(gp)
c         uncomment to print voltage distribution, elements of
c       A and X matrices, and y11 and y12
```

```
c       write (3,157)
c 157        format ('V(m)')
c       do 155 m=1,k
c          do 156 m1=1,k
c          print*,'A(',m,',',m1,')=',A(m,m1)
c 156           continue
c       print*,'X(',m,')=',X(m)
c       print*,'V(',m,')=',V(m)
c         write (3,*) V(m)
c 155        continue
c       print*,' y11=',y11
c       print*,' y12=',y12
c       write (3,*) y11
c       write (3,*) y12
        print*,' gain, phase =',gp
        return
        end
```

```
c
c               coef2.for
c
c     this program computes the matrix elements
c     used to solve AV=X using eq. (2.21-2.24).
c     and approach of Appendix A.
c     for layers above and below electrodes
c
        subroutine coef2(A,X,L)
        parameter (PI=3.141592654)
        complex A(N1,N1),X(N1),L(N2),Lmin,y11,y12,yl
        complex temp,temp1,fhsum,rel(2),rsel,C0(2)
        integer flag,r,c
        real h,g,Vn0(2),f1,grid,y1,y0,eox,LAM
        common /comp1/Lmin,y11,y12,yl,C0
        common /comp/rel,rsel
        common /flot1/Vn0
        common /flot/h,g,eox,LAM,YLOAD
        common /tin/k,N,nmax
        common /tin1/N1,N2
        flag=0
        dpi=2.0*PI
        y0=grid(0)
        y1=grid(1)
c
c     this first section computes A1(i,j) according to eq. (2.22a)
c     this computation utilizes the "symmetry"
c     of A1(i,j) - namely, that A1(1,1)=A1(k,k), A1(1,2)=A1(k,k-1)
c     and so on back to the halfway point.
c     also, the infinite sum of C(n)sin(nx)/n^2 is transformed to a more
c     rapidly converging series using fhsum.for and fsum.for
c     also compute X1(i,j) here according to eq. (2.24a)
c
        do 500 r=1,k
        if (r.eq.k) then
                s1=grid(k+1)
        else
                s1=(grid(r+1)+grid(r))/2.0
        end if
        if (r.eq.1) then
                s2=y0
        else
                s2=(grid(r)+grid(r-1))/2.0
        end if
        if ((mod(k,2).eq.0).and.(r.eq.(k/2+1))) flag=1
           do 510 c=1,k
           if ((mod(k,2).eq.1).and.(r.eq.((k+1)/2)).and
     x            .(c.eq.((k+3)/2))) flag=1
```

```
            if (flag.eq.1) then
                 A(r,c)=A(k+1-r,k+1-c)
            else
                 da= 1.0/(grid(c+1)-grid(c))
                 db= 1.0/(grid(c)-grid(c-1))
                 temp1=cmplx(0.0,0.0)
                 temp=fhsum(dpi*(s1+grid(c)),L)
c                print*,'temp=',temp
                 temp1=temp+temp1
                 temp=fhsum(dpi*(s1-grid(c)),L)
                 temp1=temp+temp1
                 temp=fhsum(dpi*(s2+grid(c)),L)
                 temp1=temp1-temp
                 temp=fhsum(dpi*(s2-grid(c)),L)
                 temp1=temp1-temp
                 A(r,c)=temp1*(da+db)
c                print*,' 1A(',r,',',c,')=',A(r,c)
                 temp1=cmplx(0.0,0.0)
                 temp=fhsum(dpi*(s2+grid(c+1)),L)
                 temp1=temp+temp1
                 temp=fhsum(dpi*(s2-grid(c+1)),L)
                 temp1=temp+temp1
                 temp=fhsum(dpi*(s1+grid(c+1)),L)
                 temp1=temp1-temp
                 temp=fhsum(dpi*(s1-grid(c+1)),L)
                 temp1=temp1-temp
                 A(r,c)=da*temp1+A(r,c)
c                print*,' 2A(',r,',',c,')=',A(r,c)
                 temp1=cmplx(0.0,0.0)
                 temp=fhsum(dpi*(s2+grid(c-1)),L)
                 temp1=temp+temp1
                 temp=fhsum(dpi*(s2-grid(c-1)),L)
                 temp1=temp+temp1
                 temp=fhsum(dpi*(s1+grid(c-1)),L)
                 temp1=temp1-temp
                 temp=fhsum(dpi*(s1-grid(c-1)),L)
                 temp1=temp1-temp
                 A(r,c)=db*temp1+A(r,c)
c                print*,' 3A(',r,',',c,')=',A(r,c)
                 sq= PI**2
                 A(r,c)=(A(r,c))/(2*sq)
c                print*,' 4A(',r,',',c,')=',A(r,c)
              end if
  510 continue
        temp1=cmplx(0.0,0.0)
        temp=fhsum(dpi*(s1+y0),L)
        temp1=temp+temp1
        temp=fhsum(dpi*(s1-y0),L)
```

```
        temp1=temp+temp1
        temp=fhsum(dpi*(s2+y1),L)
        temp1=temp+temp1
        temp=fhsum(dpi*(s2-y1),L)
        temp1=temp+temp1
        temp=fhsum(dpi*(s2+y0),L)
        temp1=temp1-temp
        temp=fhsum(dpi*(s2-y0),L)
        temp1=temp1-temp
        temp=fhsum(dpi*(s1+y1),L)
        temp1=temp1-temp
        temp=fhsum(dpi*(s1-y1),L)
        temp1=temp1-temp
        sq=PI*PI
        X(r)=(-1.0/(2.0*sq*(y1-y0)))*temp1
c       print*,' X(',r,')=',X(r)
  500       continue
c
c    this section computes A2(r,c), A3(r,c), X2(r,c) and X3(r,c)
c
        do 520 r=1,k
        if(r.eq.1) then
           dc=(.5)*(grid(2)+y1)-y0
        else if(r.eq.k) then
           dc=grid(k+1)-(.5)*(grid(k)+grid(k-1))
        else
           dc=(.5)*(grid(r+1)-grid(r-1))
        end if
c
c    X2(r) as in eq. (2.24b) in report
c
c       print*,' X(',r,')=',X(r)
c       print*,'C01=',C0(1)
c       print*,'C02=',C0(2)
c       print*,'dc=',dc
c       print*,'(y0+y1)=',y0+y1
        X(r)=X(r)-dc*(C0(1)-C0(2)*(y0+y1))
c       print*,' X(',r,')=',X(r)
           do 530 c=1,k
           da=grid(c+1)-grid(c-1)
c
c     A2(r,c) and A3(r,c) as in eqs. (2.22b) and (2.22c)
c
c          print*,' A(',r,',',c,')=',A(r,c)
c          print*,'da=',da
           A(r,c)=A(r,c)-dc*da*C0(2)+((f1(r,c))*rsel)
c          print*,' A(',r,',',c,')=',A(r,c)
  530      continue
```

```
c
c     X3(r) as in eq. (2.24c)
c
          if(r.eq.1) then
           X(r)=X(r)+(1/(y1-y0))*rsel
c          print*,'X=',X(1)
        end if
c       print*,' X(',r,')=',X(r)
  520         continue
        return
        end
c
c    compute location of k collocation points
c  presently uses a cosinusoidal distribution
        real function grid(i)
        parameter (PI=3.141592654)
        common /flot/h,g,eox,LAM,YLOAD
        common /tin/k,N,nmax
        real h,g,eox,LAM,YLOAD
        integer k,N,nmax
        grid=.25-(.5)*g*cos(PI*i/(k+1))
        return
        end
c
c   computes f1(r,c) (the delta functions) for surface contribution
c   A3(r,c)
c
        function f1(r,c)
        integer r,c
        da=1.0/(grid(r+1)-grid(r))
        db=1.0/(grid(r)-grid(r-1))
        dc=grid(r+1)-grid(r-1)
        if (r.eq.c) then
           f1=dc*db*da
        else if((r-c).eq.1) then
           f1=(-1.0)*db
        else if ((c-r).eq.1) then
           f1=(-1.0)*da
        else
           f1=0.0
        end if
        return
        end
c               FHSUM.for
c     This program computes the sum C(n)sin(nx)/(n*n)
c  using fsum.c and the fact that for a large enough n
c  C(n) approaches a fixed value.  This program computes the
c  original series sum until n=nmax and then computes the
```

```
c  rest by using fsum.c and subtracting the first 20 terms
c  of sin(nx)/(n*n). See Appendix A.
c
        complex function fhsum(fx,L)
        complex L(N2),Lmin,y11,y12,yl,C0(2)
        complex sum,L1
        common /comp1/Lmin,y11,y12,yl,C0
        common /tin/k,N,nmax
        common /tin1/N1,N2
        real fx,fy
        sum=cmplx(0.0,0.0)
        do 400 i=1,nmax
        L1=L(i)-Lmin
        fy=(sin(i*fx))/i**2
          sum=sum+fy*L1
  400      continue
        fhsum=sum+Lmin*fsum(fx)
c       print*,'sum=',sum
c       print*,'Lmin=',Lmin
c       print*,'fsum(fx)=',fsum(fx)
        return
        end
c                   FSUM.for
c           This program computes sin(nx)/(n*n) using an
c   series which converges more rapidly and with higher
c   accuracy.  The series is obtained by first noting that
c   sin(nx)/(n*n) equals the integral of ln[2sin(t/2)] from
c   0 to x.  Then use a series expansion of ln[sin(x)] and
c   integrate term by term.   This series is computed until
c   the last term is .01% of the sum. See Appendix A.
c
        function fsum(fsx)
        dimension Bern(20)
         real i1,fsx
        integer sign
c       Bernoulli numbers
        Bern(1)=.166667
        Bern(2)=.033333
        Bern(3)=.023809
        Bern(4)=.033333
        Bern(5)=.075757
        Bern(6)=.253113
        Bern(7)=1.166666
        Bern(8)=7.092156
        Bern(9)=54.971177
        Bern(10)=529.124242
        Bern(11)=6192.123188
        Bern(12)=8.658025311e4
```

```
      Bern(13)=1.425517167e6
      Bern(14)=2.729823107e7
      Bern(15)=6.015808739e8
      Bern(16)=1.511631577e10
      Bern(17)=4.296146431e11
      Bern(18)=1.37116552le13
      Bern(19)=4.883323190e14
      Bern(20)=1.929657934e16
      if (fsx.lt.0) then
          z=abs(fsx)
          sign=(-1)
      end if
      if (fsx.gt.0) then
          z=fsx
          sign=1
      end if
       if (fsx.eq.0) then
          a=0.0
          sign=1
      else
         a=z*(alog(z)-1.0)
         b=a
         i=1
440      if(abs(b/a).le.(.0001)) go to 430
         i1=i*1.0
         c=(Bern(i))*z**(2*i1+1)
         d=2.0*i1*(2.0*i1+1.0)*fact(2*i)
         b=c/d
         a=a-b
         i=i+1
         go to 440
430   continue
        end if
      fsum=(-a)*sign
      return
      end
c
c     fact(y)
c
c   computes factorials
c
      function fact(y)
      integer y
      y1=1.0
      do 420 i=0,y-1
      y1=y1*(y-i)
420        continue
      fact=y1
```

```
return
end
```

```
c
c                admit2.for
c
c     this subroutine calculates y11 and
c     y12 given a voltage distribution V. Uses
c     eq. (2.27).
c     for layers above and below electrodes
c
        subroutine admit2(V,L)
        parameter (PI=3.141592654)
        complex V(N1),L(N2),Lmin,y11,y12,y1,C0(2)
        real h,Vn0(2),g,eox,LAM
        complex temp,temp1,temp2,temp3,rel(2),rsel
        common /comp1/Lmin,y11,y12,y1,C0
        common /comp/rel,rsel
        common /flot1/Vn0
        common /flot/h,g,eox,LAM,YLOAD
        common /tin/k,N,nmax
        common /tin1/N1,N2
        y11=cmplx(0.0,0.0)
        y12=cmplx(0.0,0.0)
        temp3=cmplx(1.0,0.0)
        temp2=cmplx(1.0,0.0)
        temp=cmplx(0.0,0.0)
c
c    compute summation in eq. (2.13)
c
        do 600 m=1,N
        temp1=cmplx(0.0,0.0)
        sq=1.0/(PI*PI*m)
c
c    compute phi a (m)
c
           do 610 j=1,k
           da=1.0/(grid(j+1)-grid(j))
           db=1.0/(grid(j)-grid(j-1))
           dc=sq*((da+db)*cos(2*PI*m*grid(j))-
     x  da*cos(2*PI*m*grid(j+1))-db*cos(2*PI*m*grid(j-1)))/m
              temp1=temp1+dc*V(j)
  610           continue
        dc=sq*(cos(2*PI*m*grid(0))-cos(2*PI*m*grid(1)))
     x      /((grid(1)-grid(0))*m)
        temp1=temp1+dc
c
c    temp1=phi a of m
c
c       write (3,630)
c  630        format('phi a of n')
```

```
c       write (3,*) temp1
c
        temp1=temp1*L(m)
        dc= sin(2*PI*m*grid(k+1))
        temp2=dc*temp1
        dc=sin(2*PI*m*grid(0))-sin(2*PI*m*grid(k+1))
        y11=y11+dc*temp1
        y12=y12+temp2
c
c       write (3,631)
c  631         format ('fourier sum for y11')
c       write (3,*) dc*temp1
c       write (3,632)
c  632        format ('fourier sum for y12')
c       write (3,*) temp2
c
        temp=(.000001)*y11
        temp1=(.000001)*y12
  600   continue
        temp1=cmplx(0.0,0.0)
c
c    compute phi a (0) using eq. (2.13)
c
        do 620 j=1,k
        da=grid(j+1)-grid(j-1)
          temp1=temp1+da*V(j)
  620        continue
        temp1=temp1+grid(0)+grid(1)
c
c        temp1= phi a of 0
c       write (3,633)
c  633        format ('phi a of 0')
c       write (3,*) temp1
c
        temp1=C0(1)-temp1*C0(2)
          da=grid(0)-grid(k+1)+(.5)
        y11=y11+da*temp1
c
c       write (3,634)
c  634        format ('y11 0 term')
c       write (3,*) da*temp1
c
        temp=((.5)-grid(k+1))*temp1
        y12=y12-temp
c
c       write (3,635)
c  635        format ('y12 0 term')
c       write (3,*) temp
```

```
c
        temp=(1.0/(grid(k+1)-grid(k)))*V(k)
        temp1=(1.0/(grid(1)-grid(0)))*(1-V(1))
        y11=2*(y11+(temp1-temp)*rsel)
c
c       write (3,636)
c 636        format ('y11 surface term')
c       write (3,*) (temp1-temp)*rsel
c
        temp=temp*rsel
        y12=2*(temp+y12)
c
c       write (3,637)
c 637        format ('y12 surface term')
c       write (3,*) temp
c
        return
        end
```

```
c                       bodex2.for
c                       12/85
c
c   This program converts complex voltage ratio
c   from cartesian to polar form given y11,y12 and yload.
c
        subroutine bodex2(pgp)
        parameter (PI=3.141592654)
        complex Lmin,y11,y12,yl,C0(2)
        complex pgp,temp
        real gain,phase
        common /comp1/Lmin,y11,y12,yl,C0
        temp=y12/(y11+y12+yl)
        gain=20*alog10(cabs(temp))
        phase=180*(atan(aimag(temp)/real(temp)))/PI
c
c    branch cut at negative real axis
c
        if (real(temp).lt.0.0) then
           if (aimag(temp).lt.0.0) then
              phase=phase-180
           else
              phase=180+phase
           end if
        end if
        pgp=cmplx(gain,phase)
        return
        end
```

```
c       cfo.for
c
c       obtain chip information
c       LAM,eox,a,h,YLOAD
c
        subroutine chipinfo()
        real LAM,eox,h,g,YLOAD
        common /flot/h,g,eox,LAM,YLOAD
        print*,'input spatial wavelength (lambda)'
        read 4,LAM
   4        format(E9.3)
        print*,'input insulating layer permittivity (eox)'
        read 4,eox
        print*,'input normalized oxide layer thickness (lambda/4h) '
        read 15,rw
   15        format(E9.0)
        h=1.0/(4.0*rw)
        print*,'h=',h
        print*,'input normalized interelectrode spacing (a/lambda) '
        read 15,g
        print 25,rw,g
   25       format(' rw=',F5.2,' g=',F3.2)
        print*,'input normalized load capacitance '
        read 15,YLOAD
        write (3,150) LAM
  150       format ('spatial wavelength = ',1PE9.3)
        write (3,151) eox
  151       format ('insulating layer permittivity = ',1PE9.3)
        write (3,161) rw
  161       format ('width  (lambda/4h) = ',F5.2)
        write (3,162) g
  162       format ('interelectrode spacing (a/lambda) = ',F5.4)
        write (3,164) YLOAD
  164       format ('normalized load capacitance = ',F5.2)
        return
        end
```

```
c       hcfo.for
c
c       obtain chip information
c       for multiple wavelength estimation
c       eox,a,h
c
        subroutine chipinfo()
        real LAM,eox,h,g,YLOAD,h1
        common /flot/h,g,eox,LAM,YLOAD
        common h1
        print*,'input insulating layer permittivity (eox)'
        read 4,eox
  4        format(E9.3)
        print*,'input insulating layer thickness (h) '
        read 15,h1
   15        format(1PE9.2)
        print*,'h1=',h1
c       print*,'input interelectrode spacing (a) '
c       read 15,g
        g=.25
        write (3,151) eox
  151       format ('insulating layer permittivity = ',1PE9.3)
        write (3,161) h1
  161       format ('insulating layer thickness (h) = ',1PE8.2)
        write (3,162) g
  162       format ('interelectrode spacing (a/lambda) = ',F5.4)
        return
        end
```

```
c
c           scfo1.for
c
c       gets necessary information for computing Cn
c       this one is for layers of homogeneous media
c       only above electrodes
c
        subroutine scapinfo1(d)
        complex e(100),se(100),re(100),rse(100)
        real d(100),eox,LAM,g,h,YLOAD
        integer i
        common /comp2/e,se,re,rse
        common /tin2/num
        common /flot/h,g,eox,LAM,YLOAD
        print*,'input # of layers above electrode '
        print*,'(including infinite half-space) '
        read 5,num
    5       format(I3)
        print*,'starting with infinite half space '
        do 100 i=1,num
        print*,'input layer',i,'thickness (meters)'
        read 15,d(i)
   15         format(E9.0)
        print*,d(i)
        d(i)=d(i)/LAM
        print*,'input layer',i,'epsilon (F/m) and sigma (S/m)'
        read 20,e(i)
   20   format(E8.0,E8.0)
        print*,'e=',e(i)
        print*,'input layer',i,'surface epsilon (F) and sigma (S)'
        read 20,se(i)
        print*,'se=',se(i)
  100   continue
        write (3,169)
   169  format ('****************************************')
        do 105 i=1,num
        write (3,165) i
   165      format ('LAYER  ',I2)
        write (3,166) d(i)*LAM
   166      format ('thickness = ',1PE8.2)
        write (3,167) real(e(i)),aimag(e(i))
   167      format ('e = ',1PE8.2,' c = ',1PE8.2)
        write (3,168) real(se(i)),aimag(se(i))
   168      format('se = ',1PE8.2,' sc = ',1PE8.2)
        write (3,169)
   105  continue
        return
        end
```

```
c
c          scap1.for
c
c       computes Cn for layers of homogeneous media
c       matches algorithm outlined in Appendix B
c       of report
c       only layers above electrodes
c
        subroutine scap1(d,C)
        parameter (PI=3.141592654)
        complex e(100),se(100),re(100),rse(100),C(N2)
        complex Ckmin,A22,A12,tmp,tmp1
        real d(100),eox,LAM,w1,g,h,YLOAD
        integer m,j
c
c       needs e,se,re,rse,d,num,nmax,eox,LAM
c
        common /comp2/e,se,re,rse
        common /flot2/w1
        common /tin2/num
        common /flot/h,g,eox,LAM,YLOAD
        common /tin/k,N,nmax
        common /tin1/N1,N2
c
c       functions defined in Appendix B
c
        A12(j,m)=re(j)*2*PI*m/sinh(2*PI*m*d(j))
        A22(j,m)=re(j)*2*PI*m/tanh(2*PI*m*d(j))
c
c       normalize permittivity and conductivity
c
        do 140 j=1,num
        re(j)=e(j)/eox
          rse(j)=se(j)/(eox*LAM)
        re(j)=cmplx(real(re(j)),-aimag(re(j))/w1)
          rse(j)=cmplx(real(rse(j)),-aimag(rse(j))/w1)
  140      continue
c
        do 31 m=1,N2
        tmp=cmplx(1.0,0.0)
        C(m)=A22(1,m)
        if (num.gt.1) then
           do 111 j=2,num
c       check to prevent overflow when evaluating sinh function
           if ((2*PI*m*d(j)).lt.44) then
c               print*,'C(m)=',C(m)
c               print*,'rse(j)=',rse(j)
                C(m)=A22(j,m)-((A12(j,m))**2)/(C(m)+A22(j,m)
```

```
     x                +rse(j)*((2*PI*m)**2))
c               print*,'A22=',A22(j,m)
c               print*,'A12=',A12(j,m)
           else
                C(m)=A22(j,m)
           end if
  111           continue
        end if
c
c       find out when Cn/kn approaches a limit
c
        if (m.gt.1) then
           tmp=C(m-1)/(2*PI*(m-1))
           tmp1=C(m)/(2*PI*m)
c          print*,'tmp=',tmp
c          print*,'tmp1=',tmp1
           tmp=tmp-tmp1
c          print*,'tmp=',tmp
           if (abs(aimag(tmp)).lt.1E-20) then
                tmp=cmplx(real(tmp),0.0)
           end if
           tmp=cmplx(cabs(tmp),0.0)
        end if
c       stop calculating if have approached limit
        if (real(tmp).lt.1E-5) then
           go to 32
        end if
c       print*,'C(m)=',C(m)
   31       continue
   32       nmax=m-1
c       print*,'nmax=',nmax
        Ckmin=C(nmax)/(2*PI*nmax)
c       print*,'Ckmin=',Ckmin
c       fill up rest of Cm with Ckmin
        do 33 m=nmax+1,N2
        C(m)=Ckmin*2*PI*m
   33       continue
        return
        end
```

```
c
c          scap1a.for
c          10/22/87
c
c       computes Cn for layers of exponential media
c       matches algorithm outlined in Appendix B of report
c       uses exponential transfer relations
c       only layers above electrodes
c
        subroutine scap1a(d,C)
        parameter (PI=3.141592654)
        complex e(100),se(100),re(100),rse(100),C(N2)
        complex Ckmin,A11,A21,A22,A12,tmp,tmp1,eta,gamma,arg
        real d(100),eox,LAM,w1,g,h,YLOAD
        integer m,j
c
c       needs e,se,re,rse,d,num,nmax,eox,LAM
c
        common /comp2/e,se,re,rse
        common /flot2/w1
        common /tin2/num
        common /flot/h,g,eox,LAM,YLOAD
        common /tin/k,N,nmax
        common /tin1/N1,N2
c
c       define complex sinh and complex tanh
c
        csinh(arg)=cosh(real(arg))*
     x                  cmplx(tanh(real(arg))*cos(aimag(arg)),
     x                  sin(aimag(arg)))
        ccoth(arg)=cmplx(cos(aimag(arg)),
     x                  tanh(real(arg))*sin(aimag(arg)))/
     x             cmplx(tanh(real(arg))*cos(aimag(arg)),
     x                  sin(aimag(arg)))
c
c       functions defined in Appendix B but for
c       exponential tranfer relations
c
        A11(j,m)=re(j)*gamma*cexp(2*eta*d(j))*
     x                        (eta/gamma - ccoth(gamma*d(j)))
        A12(j,m)=cexp(eta*d(j))*re(j)*gamma/csinh(gamma*d(j))
        A21(j,m)=-cexp(eta*d(j))*re(j)*gamma/csinh(gamma*d(j))
        A22(j,m)=re(j)*gamma*(eta/gamma + ccoth(gamma*d(j)))
c
c       normalize permittivity and conductivity
c
        do 140 j=1,num
        re(j)=e(j)/eox
```

```
          rse(j)=se(j)/(eox*LAM)
        re(j)=cmplx(real(re(j)),-aimag(re(j))/w1)
          rse(j)=cmplx(real(rse(j)),-aimag(rse(j))/w1)
  140        continue
c
        do 31 m=1,N2
        tmp=cmplx(1.0,0.0)
c
c       first (upper) layer will be uniform to infinity
c
        C(m)=re(1)*2*PI*m/tanh(2*PI*m*d(1))
c       print*,'C(',m,')=',C(m)
        if (num.gt.1) then
           do 111 j=2,num
           eta=clog(re(j-1)/re(j))/(2*d(j))
           gamma=csqrt(eta**2+(2*PI*m)**2)
           print*,'gamma=',gamma
           print*,'eta=',eta
c       check to prevent overflow when evaluating cosh function
           if (real(gamma*d(j)).lt.44) then
c               print*,'rse(j)=',rse(j)
c               print*,'A22=',A22(j,m)
c               print*,'A12=',A12(j,m)
c               print*,'A11=',A11(j,m)
c               print*,'A21=',A21(j,m)
                C(m)=A22(j,m)+(A21(j,m)*A12(j,m))/(C(m)-A11(j,m)
     x                 +rse(j)*((2*PI*m)**2))
                print*,'C(',m,')=',C(m)
           else
                   C(m)=re(j)*2*PI*m/tanh(2*PI*m*d(j))
           end if
  111           continue
        end if
c
c       find out when Cn/kn approaches a limit
c
        if (m.gt.1) then
           tmp=C(m-1)/(2*PI*(m-1))
           tmp1=C(m)/(2*PI*m)
c          print*,'tmp=',tmp
c          print*,'tmp1=',tmp1
           tmp=tmp-tmp1
c          print*,'tmp=',tmp
           if (abs(aimag(tmp)).lt.1E-20) then
                tmp=cmplx(real(tmp),0.0)
           end if
           tmp=cmplx(cabs(tmp),0.0)
        end if
```

```
c       stop calculating if have approached limit
        if (real(tmp).lt.1E-5) then
           go to 32
        end if
c       print*,'C(m)=',C(m)
   31        continue
   32        nmax=m-1
c       print*,'nmax=',nmax
        Ckmin=C(nmax)/(2*PI*nmax)
c       print*,'Ckmin=',Ckmin
c       fill up rest of Cm with Ckmin
        do 33 m=nmax+1,N2
        C(m)=Ckmin*2*PI*m
   33        continue
        return
        end
```

```
c
c           scfo2.for
c
c       gets necessary information for computing Cn
c       this one is for layers of homogeneous media
c       with layers above and below electrodes
c
        subroutine scapinfo2(d)
        complex e(100,2),se(100,2),re(100,2),rse(100,2)
        real d(100,2),eox,LAM,g,h,YLOAD
        integer i,num(2)
        common /comp2/e,se,re,rse
        common /tin2/num
        common /flot/h,g,eox,LAM,YLOAD
c
c       above
c
        print*,'input # of layers above electrode '
        print*,'(including infinite half-space) '
        read 5,num(1)
   5        format(I3)
        print*,'starting with infinite half space '
        do 100 i=1,num(1)
        print*,'input layer',i,'thickness (meters)'
        read 15,d(i,1)
   15         format(E9.0)
        print*,d(i,1)
        d(i,1)=d(i,1)/LAM
        print*,'input layer',i,'epsilon (F/m) and sigma (S/m)'
        read 20,e(i,1)
  20    format(E8.0,E8.0)
        print*,'e=',e(i,1)
        print*,'input layer',i,'surface epsilon (F) and sigma (S)'
        read 20,se(i,1)
        print*,'se=',se(i,1)
 100    continue
        write (3,169)
  169   format ('****************************************')
        do 105 i=1,num(1)
        write (3,165) i
  165        format ('UPPER LAYER  ',I2)
        write (3,166) d(i,1)*LAM
  166        format ('thickness = ',1PE8.2)
        write (3,167) real(e(i,1)),aimag(e(i,1))
  167        format ('e = ',1PE8.2,' c = ',1PE8.2)
        write (3,168) real(se(i,1)),aimag(se(i,1))
  168        format('se = ',1PE8.2,' sc = ',1PE8.2)
        write (3,169)
```

```
  105   continue
c
c       below
c
        print*,'input # of layers below electrode '
        print*,'(including infinite half-space) '
        read 5,num(2)
        print*,'starting with infinite half space '
        do 700 i=1,num(2)
        print*,'input layer',i,'thickness (meters)'
        read 15,d(i,2)
        print*,d(i,2)
        d(i,2)=d(i,2)/LAM
        print*,'input layer',i,'epsilon (F/m) and sigma (S/m)'
        read 20,e(i,2)
        print*,'e=',e(i,2)
        print*,'input layer',i,'surface epsilon (F) and sigma (S)'
        read 20,se(i,2)
        print*,'se=',se(i,2)
  700   continue
        write (3,169)
        do 705 i=1,num(2)
        write (3,765) i
  765        format ('LOWER LAYER  ',I2)
        write (3,166) d(i,2)*LAM
        write (3,167) real(e(i,2)),aimag(e(i,2))
        write (3,168) real(se(i,2)),aimag(se(i,2))
        write (3,169)
  705   continue
        return
        end
```

```
c
c          scap2.for
c
c       computes Cn for layers of homogeneous media
c       matches algorithm outlined in Appendix B
c       of report
c       for layers above and below electrodes
c
        subroutine scap2(d,C)
        parameter (PI=3.141592654)
        complex e(100,2),se(100,2),re(100,2),rse(100,2),C(N2,2),C0(2)
        complex Ckmin(2),A22,A220,A12,A120,tmp,tmp1,Lmin,y11,y12,yl
        real d(100,2),eox,LAM,w1,g,h,YLOAD,dsum,Vn0(2)
        integer m,j,num(2),m1
c
c       needs e,se,re,rse,d,num,nmax,eox,LAM
c
        common /comp2/e,se,re,rse
        common /comp1/Lmin,y11,y12,yl,C0
        common /flot2/w1
        common /tin2/num
        common /flot/h,g,eox,LAM,YLOAD
        common /tin/k,N,nmax
        common /tin1/N1,N2
        common /flot1/Vn0
c
c       functions defined in Appendix B
c
        A12(j,m,m1)=re(j,m1)*2*PI*m/sinh(2*PI*m*d(j,m1))
        A22(j,m,m1)=re(j,m1)*2*PI*m/tanh(2*PI*m*d(j,m1))
        A120(j,m1)=re(j,m1)/d(j,m1)
        A220(j,m1)=re(j,m1)/d(j,m1)
c
c       normalize permittivity and conductivity
c
        do 180 m1=1,2
        do 140 j=1,num(m1)
        re(j,m1)=e(j,m1)/eox
          rse(j,m1)=se(j,m1)/(eox*LAM)
        re(j,m1)=cmplx(real(re(j,m1)),-aimag(re(j,m1))/w1)
          rse(j,m1)=cmplx(real(rse(j,m1)),-aimag(rse(j,m1))/w1)
  140      continue
c
        sgn=((-1)**(m1+1))
c
c       for n=0 term
c
        C0(m1)=A220(1,m1)*sgn
```

```
c       print*,'C0(',m1,')=',C0(m1)
        if (num(m1).gt.1) then
           do 811 j=2,num(m1)
c               print*,'d(',j,',',m1,')=',d(j,m1)
c               print*,'re(',j,',',m1,')=',re(j,m1)
c               print*,'rse(',j,',',m1,')=',rse(j,m1)
c               print*,'A220=',A220(j,m1)
c               print*,'A120=',A120(j,m1)
           C0(m1)=sgn*A220(j,m1)-((A120(j,m1))**2)/
     x      (C0(m1)+sgn*A220(j,m1))
c
c       recognizes ground plane below by setting
c       imag{se(j,2)} to 1
c
           if (aimag(se(j,m1)).eq.1) then
                C0(m1)=A220(j,m1)*sgn
           end if
c       print*,'C0(',m1,')=',C0(m1)
  811           continue
        end if
        C0(m1)=C0(m1)*Vn0(m1)
c       print*,'C0(',m1,')=',C0(m1)
        do 31 m=1,N2
        tmp=cmplx(1.0,0.0)
        C(m,m1)=A22(1,m,m1)*sgn
        if (num(m1).gt.1) then
           do 111 j=2,num(m1)
c       check to prevent overflow when evaluating sinh function
           if ((2*PI*m*d(j,m1)).lt.44) then
c               if (m1.eq.2) then
c               print*,'C(',m,',',m1,')=',C(m,m1)
c               print*,'d(',j,',',m1,')=',d(j,m1)
c               print*,'re(',j,',',m1,')=',re(j,m1)
c               print*,'rse(',j,',',m1,')=',rse(j,m1)
c               print*,'A22=',A22(j,m,m1)
c               print*,'A12=',A12(j,m,m1)
c               end if
                C(m,m1)=sgn*A22(j,m,m1)-((A12(j,m,m1))**2)/
     x      (C(m,m1)+sgn*(A22(j,m,m1)+rse(j,m1)*((2*PI*m)**2)))
c               if (m1.eq.2) then
c               print*,'C(',m,',',m1,')=',C(m,m1)
c               end if
           else
c               print*,'overflow'
c               print*,'j=',j
                C(m,m1)=sgn*A22(j,m,m1)
c               print*,'C(',m,',',m1,')=',C(m,m1)
           end if
```

```
  111           continue
        end if
c
c       find out when Cn/kn approaches a limit
c
        if (m.gt.1) then
           tmp=C(m-1,m1)/(2*PI*(m-1))
           tmp1=C(m,m1)/(2*PI*m)
c          print*,'tmp=',tmp
c          print*,'tmp1=',tmp1
           tmp=tmp-tmp1
c          print*,'tmp=',tmp
           if (abs(aimag(tmp)).lt.1E-20) then
                tmp=cmplx(real(tmp),0.0)
           end if
           tmp=cmplx(cabs(tmp),0.0)
        end if
c       stop calculating if have approached limit
        if (real(tmp).lt.1E-5) then
           go to 32
        end if
c       print*,'C(',m,',',m1,')=',C(m,m1)
   31        continue
   32        nmax=m-1
c       print*,'nmax=',nmax
        Ckmin(m1)=C(nmax,m1)/(2*PI*nmax)
c       print*,'Ckmin(',m1,')=',Ckmin(m1)
c       fill up rest of Cm with Ckmin
        do 33 m=nmax+1,N2
        C(m,m1)=Ckmin(m1)*2*PI*m
   33        continue
  180        continue
c       print*,'C0(1)=',C0(1)
c       print*,'C0(2)=',C0(2)
        return
        end
```

```
c
c           scfo3.for
c
c       gets necessary information for computing Cn
c       this one is for one layer of dispersive media
c       plus semi-infinite half space
c
        subroutine scapinfo3(d)
        complex e(100),se(100),re(100),rse(100),esR
        real d(100),eox,LAM,g,h,YLOAD,phi
        integer i
        common /comp2/e,se,re,rse
        common /tin2/num
        common /flot/h,g,eox,LAM,YLOAD
        common /flot3/phi,esR
        print*,'input # of layers above electrode '
        print*,'(including infinite half-space) '
        read 5,num
    5       format(I3)
        print*,'starting with infinite half space '
        do 100 i=1,num
        print*,'input layer',i,'thickness (meters)'
        read 15,d(i)
   15         format(E9.0)
        print*,d(i)
        d(i)=d(i)/LAM
        print*,'input layer',i,'epsilon (F/m) and sigma (S/m)'
        read 20,e(i)
   20   format(E8.0,E8.0)
        print*,'e=',e(i)
        print*,'input layer',i,'surface epsilon (F) and sigma (S)'
        read 20,se(i)
        print*,'se=',se(i)
  100   continue
        print*,'input dense phase epsilon (F/m) and sigma (S/m)'
        read 20,e(3)
        print*,'input dilute phase epsilon (F/m) and sigma (S/m)'
        read 20,e(4)
        print*,'input ratio of complex surface permittivity to R'
        print*,'(particle radius in meters)'
        read 20,esR
        print*,'input volume fraction of dilute phase (0 to 1) '
        read 15,phi
        write (3,169)
   169  format ('****************************************')
        do 105 i=1,num
        write (3,165) i
   165      format ('LAYER  ',I2)
```

```
      write (3,166) d(i)*LAM
166        format ('thickness = ',1PE8.2)
      write (3,167) real(e(i)),aimag(e(i))
167        format ('e = ',1PE8.2,' c = ',1PE8.2)
      write (3,168) real(se(i)),aimag(se(i))
168        format('se = ',1PE8.2,' sc = ',1PE8.2)
      write (3,169)
105   continue
      write (3,171) real(e(3)),aimag(e(3))
171          format ('dense phase e = ',1PE8.2,' c = ',1PE8.2)
      write (3,172) real(e(4)),aimag(e(4))
172        format ('dilute phase e = ',1PE8.2,' c = ',1PE8.2)
      write (3,174) esR
174        format ('es/R = ',1PE8.2,1PE8.2)
      write (3,173) phi
173   format ('volume fraction phi = ',1PE8.2)
      write (3,169)
      return
      end
```

```
c
c          scap3.for
c
c       computes Cn for layers of homogeneous media
c       matches algorithm outlined in Appendix B
c       of report using Lorentz sphere dispersion
c
        subroutine scap3(d,C)
        parameter (PI=3.141592654)
        complex e(100),se(100),re(100),rse(100),C(N2)
        complex Ckmin,A22,A12,tmp,tmp1
        complex top,bottom,esR,resR
        real d(100),eox,LAM,w1,g,h,YLOAD,phi
        integer m,j
c
c       needs rea,resea,e,se,re,rse,d,num,nmax,eox,LAM
c
        common /comp/rea,rsea
        common /comp2/e,se,re,rse
        common /flot2/w1
        common /tin2/num
        common /flot/h,g,eox,LAM,YLOAD
        common /tin/k,N,nmax
        common /tin1/N1,N2
        common /flot3/phi,esR
c
c       functions defined in Appendix B
c
        A12(j,m)=re(j)*2*PI*m/sinh(2*PI*m*d(j))
        A22(j,m)=re(j)*2*PI*m/tanh(2*PI*m*d(j))
c
c       normalize
c
        print*,'es/R=',esR
        resR=esR/eox
        resR=cmplx(real(resR),-(aimag(resR))/w1)
        do 140 j=1,num+2
        re(j)=e(j)/eox
          rse(j)=se(j)/(eox*LAM)
        re(j)=cmplx(real(re(j)),-aimag(re(j))/w1)
          rse(j)=cmplx(real(rse(j)),-aimag(rse(j))/w1)
        print*,'re(',j,')=',re(j)
  140       continue
c       print*,'e(3) = ',eox*real(re(3)),-w1*eox*aimag(re(3))
        top=((2*re(3)+re(4)+2*resR)-2*phi*(re(3)-re(4)-2*resR))
c       print*,'top = ',eox*real(top),-w1*eox*aimag(top)
        bottom=(2*re(3)+re(4)+2*resR+phi*(re(3)-re(4)-2*resR))
c       print*,'bottom = ',eox*real(bottom),-w1*eox*aimag(bottom)
```

```
        re(2)=re(3)*top/bottom
c       print*,'e(2) = ',eox*real(re(2)),-w1*eox*aimag(re(2))
        rea=re(2)
c
        do 31 m=1,1000
        tmp=cmplx(1.0,0.0)
        C(m)=A22(1,m)
        if (num.gt.1) then
           do 111 j=2,num
c       check to prevent overflow when evaluating sinh function
           if ((2*PI*m*d(j)).lt.44) then
c               print*,'C(m)=',C(m)
c               print*,'rse(j)=',rse(j)
                C(m)=A22(j,m)-((A12(j,m))**2)/(C(m)+A22(j,m)
     x                 +rse(j)*((2*PI*m)**2))
c               print*,'A22=',A22(j,m)
c               print*,'A12=',A12(j,m)
           else
                C(m)=A22(j,m)
           end if
  111           continue
        end if
c
c       find out when Cn/kn approaches a limit
c
        if (m.gt.1) then
           tmp=C(m-1)/(2*PI*(m-1))
           tmp1=C(m)/(2*PI*m)
c          print*,'tmp=',tmp
c          print*,'tmp1=',tmp1
           tmp=tmp-tmp1
c          print*,'tmp=',tmp
           if (abs(aimag(tmp)).lt.1E-20) then
                tmp=cmplx(real(tmp),0.0)
           end if
           tmp=cmplx(cabs(tmp),0.0)
        end if
c       stop calculating if have approached limit
        if (real(tmp).lt.1E-5) then
           go to 32
        end if
c       print*,'C(m)=',C(m)
   31       continue
   32       nmax=m-1
c       print*,'nmax=',nmax
        Ckmin=C(nmax)/(2*PI*nmax)
c       print*,'Ckmin=',Ckmin
c       fill up rest of Cm with Ckmin
```

```
      do 33 m=nmax+1,1000
      C(m)=Ckmin*2*PI*m
33        continue
      return
      end
```

```
c
c           scfo4.for
c
c       gets necessary information for computing Cn
c       this one is for one layer of dispersive media
c       plus semi-infinite half space
c       power law dispersion
c
        subroutine scapinfo4(d)
        complex e(100),se(100),re(100),rse(100)
        real d(100),eox,LAM,g,h,YLOAD,est,B,np,wp
        integer i
        common /comp2/e,se,re,rse
        common /tin2/num
        common /flot/h,g,eox,LAM,YLOAD
        common /flot3/np,B,est,wp
        print*,'input # of layers above electrode '
        print*,'(including infinite half-space) '
        read 5,num
   5        format(I3)
        print*,'starting with infinite half space '
        do 100 i=1,num
        print*,'input layer',i,'thickness (meters)'
        read 15,d(i)
   15         format(E9.0)
        print*,d(i)
        d(i)=d(i)/LAM
        print*,'input layer',i,'epsilon (F/m) and sigma (S/m)'
        read 20,e(i)
   20   format(E8.0,E8.0)
        print*,'e=',e(i)
        print*,'input layer',i,'surface epsilon (F) and sigma (S)'
        read 20,se(i)
        print*,'se=',se(i)
  100   continue
        print*,'input est (stationary permittivity)'
        read 15,est
        print*,'input np,B,wp'
        read 15,np
        read 15,B
        read 15,wp
        write (3,169)
   169  format ('****************************************')
        do 105 i=1,num
        write (3,165) i
   165       format ('LAYER  ',I2)
        write (3,166) d(i)*LAM
   166       format ('thickness = ',1PE8.2)
```

```
      write (3,167) real(e(i)),aimag(e(i))
167        format ('e = ',1PE8.2,' c = ',1PE8.2)
      write (3,168) real(se(i)),aimag(se(i))
168        format('se = ',1PE8.2,' sc = ',1PE8.2)
      write (3,169)
105   continue
      write (3,173) est
173   format ('est = ',1PE8.2)
      write (3,176) np
176        format ('np = ',F4.2)
      write (3,174) B
174        format ('B = ',1PE8.2)
      write (3,175) wp
175        format ('wp = ',1PE8.2)
      write (3,169)
      return
      end
```

```
c
c          scap4.for
c
c       computes Cn for layers of homogeneous media
c       matches algorithm outlined in Appendix B
c       of report using power law dependence
c
        subroutine scap4(d,C)
        parameter (PI=3.141592654)
        complex e(100),se(100),re(100),rse(100),C(N2)
        complex Ckmin,A22,A12,tmp,tmp1
        real d(100),eox,LAM,w1,g,h,YLOAD,est
        real np,B,wp
        integer m,j
c
c       needs rea,rsea,se,re,rse,d,num,nmax,eox,LAM
c
        common /comp/rea,rsea
        common /comp2/e,se,re,rse
        common /flot2/w1
        common /tin2/num
        common /flot/h,g,eox,LAM,YLOAD
        common /tin/k,N,nmax
        common /tin1/N1,N2
        common /flot3/np,B,est,wp
c
c       functions defined in Appendix B
c
        A12(j,m)=re(j)*2*PI*m/sinh(2*PI*m*d(j))
        A22(j,m)=re(j)*2*PI*m/tanh(2*PI*m*d(j))
c
c       normalize permittivity and conductivity
c
        e(2)=(B*w1/wp)*cmplx(sin(np*PI/2),-cos(np*PI/2))
        e(2)=cmplx(est+real(e(2)),-w1*aimag(e(2)))
        do 140 j=1,num
        re(j)=e(j)/eox
          rse(j)=se(j)/(eox*LAM)
        re(j)=cmplx(real(re(j)),-aimag(re(j))/w1)
          rse(j)=cmplx(real(rse(j)),-aimag(rse(j))/w1)
        print*,'re(',j,')=',re(j)
  140       continue
        print*,'e(2) = ',eox*real(re(2)),-w1*eox*aimag(re(2))
        rob=re(2)
c
        do 31 m=1,1000
        tmp=cmplx(1.0,0.0)
        C(m)=A22(1,m)
```

```
      if (num.gt.1) then
         do 111 j=2,num
c        check to prevent overflow when evaluating sinh function
         if ((2*PI*m*d(j)).lt.44) then
c             print*,'C(m)=',C(m)
c             print*,'rse(j)=',rse(j)
              C(m)=A22(j,m)-((A12(j,m))**2)/(C(m)+A22(j,m)
     x               +rse(j)*((2*PI*m)**2))
c             print*,'A22=',A22(j,m)
c             print*,'A12=',A12(j,m)
         else
              C(m)=A22(j,m)
         end if
  111         continue
      end if
c
c     find out when Cn/kn approaches a limit
c
      if (m.gt.1) then
         tmp=C(m-1)/(2*PI*(m-1))
         tmp1=C(m)/(2*PI*m)
c        print*,'tmp=',tmp
c        print*,'tmp1=',tmp1
         tmp=tmp-tmp1
c        print*,'tmp=',tmp
         if (abs(aimag(tmp)).lt.1E-20) then
              tmp=cmplx(real(tmp),0.0)
         end if
         tmp=cmplx(cabs(tmp),0.0)
      end if
c     stop calculating if have approached limit
      if (real(tmp).lt.1E-5) then
         go to 32
      end if
c     print*,'C(m)=',C(m)
   31    continue
   32    nmax=m-1
c     print*,'nmax=',nmax
c     fill up rest of Cm with Ckmin
      Ckmin=C(nmax)/(2*PI*nmax)
c     print*,'Ckmin=',Ckmin
      do 33 m=nmax+1,1000
      C(m)=Ckmin*2*PI*m
   33    continue
      return
      end
```

```
c
c          scapest1.for
c
c       computes Cn for layers of homogeneous media
c       matches algorithm outlined in Appendix B
c       of report - difference is no normalization
c       done in subroutine here, for estimation routines
c       layers above electrodes
c
        subroutine scapest1(d,C)
          parameter (PI=3.141592654)
        complex e(100),se(100),re(100),rse(100),C(N2)
        complex Ckmin,A22,A12,tmp,tmp1
        real d(100),eox,LAM,w1,g,h,YLOAD
        integer m,j
c
c       needs e,se,re,rse,d,num,nmax,eox,LAM
c
        common /comp2/e,se,re,rse
        common /flot2/w1
        common /tin2/num
        common /flot/h,g,eox,LAM,YLOAD
        common /tin/k,N,nmax
        common /tin1/N1,N2
c
c       functions defined in Appendix B
c
        A12(j,m)=re(j)*2*PI*m/sinh(2*PI*m*d(j))
        A22(j,m)=re(j)*2*PI*m/tanh(2*PI*m*d(j))
c
        do 31 m=1,1000
        tmp=cmplx(1.0,0.0)
        C(m)=A22(1,m)
        if (num.gt.1) then
           do 111 j=2,num
c       check to prevent overflow when evaluating sinh function
           if ((2*PI*m*d(j)).lt.44) then
c               print*,'C(m)=',C(m)
c               print*,'rse(j)=',rse(j)
                C(m)=A22(j,m)-((A12(j,m))**2)/(C(m)+A22(j,m)
     x                 +rse(j)*((2*PI*m)**2))
c               print*,'A22=',A22(j,m)
c               print*,'A12=',A12(j,m)
           else
                C(m)=A22(j,m)
           end if
  111           continue
        end if
```

```
c
c       find out when Cn/kn approaches a limit
c
        if (m.gt.1) then
           tmp=C(m-1)/(2*PI*(m-1))
           tmp1=C(m)/(2*PI*m)
c          print*,'tmp=',tmp
c          print*,'tmp1=',tmp1
           tmp=tmp-tmp1
c          print*,'tmp=',tmp
           if (abs(aimag(tmp)).lt.1E-20) then
                tmp=cmplx(real(tmp),0.0)
           end if
           tmp=cmplx(cabs(tmp),0.0)
        end if
c       stop calculating if have approached limit
        if (real(tmp).lt.1E-5) then
           go to 32
        end if
c       print*,'C(m)=',C(m)
   31        continue
   32        nmax=m-1
c       print*,'nmax=',nmax
        Ckmin=C(nmax)/(2*PI*nmax)
c       print*,'Ckmin=',Ckmin
c       fill up rest of Cm with Ckmin
        do 33 m=nmax+1,1000
        C(m)=Ckmin*2*PI*m
   33        continue
        return
        end
```

```
c
c          scapest1a.for
c          10/22/87
c
c       computes Cn for layers of exponential media
c       matches algorithm outlined in Appendix B
c       of report - difference is no normalization
c       done in subroutine here
c       uses exponential transfer relations
c       layers above electrodes
c
        subroutine scapest1a(d,C)
        parameter (PI=3.141592654)
        complex e(100),se(100),re(100),rse(100),C(N2)
        complex Ckmin,A11,A21,A22,A12,tmp,tmp1,eta,gamma,arg
        real d(100),eox,LAM,w1,g,h,YLOAD
        integer m,j
c
c       needs e,se,re,rse,d,num,nmax,eox,LAM
c
        common /comp2/e,se,re,rse
        common /flot2/w1
        common /tin2/num
        common /flot/h,g,eox,LAM,YLOAD
        common /tin/k,N,nmax
        common /tin1/N1,N2
c
c       define complex sinh and complex tanh
c
        csinh(arg)=cosh(real(arg))*
     x                  cmplx(tanh(real(arg))*cos(aimag(arg)),
     x                  sin(aimag(arg)))
        ccoth(arg)=cmplx(cos(aimag(arg)),
     x                  tanh(real(arg))*sin(aimag(arg)))/
     x            cmplx(tanh(real(arg))*cos(aimag(arg)),
     x                  sin(aimag(arg)))
c
c       functions defined in Appendix B but
c       for exponential tranfer relations
c       see Cont. Elect. by Melcher, p.
c
        A11(j,m)=re(j)*gamma*cexp(2*eta*d(j))*
     x                       (eta/gamma - ccoth(gamma*d(j)))
        A12(j,m)=cexp(eta*d(j))*re(j)*gamma/csinh(gamma*d(j))
        A21(j,m)=-cexp(eta*d(j))*re(j)*gamma/csinh(gamma*d(j))
        A22(j,m)=re(j)*gamma*(eta/gamma + ccoth(gamma*d(j)))
c
        do 31 m=1,N2
```

```
      tmp=cmplx(1.0,0.0)
c
c     first (upper) layer will be uniform to infinity
c
      C(m)=re(1)*2*PI*m/tanh(2*PI*m*d(1))
c     print*,'C(',m,')=',C(m)
      if (num.gt.1) then
         do 111 j=2,num
         eta=clog(re(j-1)/re(j))/(2*d(j))
         gamma=csqrt(eta**2+(2*PI*m)**2)
c        print*,'gamma=',gamma
c        print*,'eta=',eta
c     check to prevent overflow when evaluating cosh function
         if (real(gamma*d(j)).lt.44) then
c             print*,'rse(j)=',rse(j)
c             print*,'A22=',A22(j,m)
c             print*,'A12=',A12(j,m)
c             print*,'A11=',A11(j,m)
c             print*,'A21=',A21(j,m)
              C(m)=A22(j,m)+(A21(j,m)*A12(j,m))/(C(m)-A11(j,m)
     x                +rse(j)*((2*PI*m)**2))
c             print*,'C(',m,')=',C(m)
         else
                 C(m)=re(j)*2*PI*m/tanh(2*PI*m*d(j))
         end if
  111         continue
      end if
c
c     find out when Cn/kn approaches a limit
c
      if (m.gt.1) then
         tmp=C(m-1)/(2*PI*(m-1))
         tmp1=C(m)/(2*PI*m)
c        print*,'tmp=',tmp
c        print*,'tmp1=',tmp1
         tmp=tmp-tmp1
c        print*,'tmp=',tmp
         if (abs(aimag(tmp)).lt.1E-20) then
              tmp=cmplx(real(tmp),0.0)
         end if
         tmp=cmplx(cabs(tmp),0.0)
      end if
c     stop calculating if have approached limit
      if (real(tmp).lt.1E-5) then
         go to 32
      end if
c     print*,'C(m)=',C(m)
   31    continue
```

```
   32        nmax=m-1
c       print*,'nmax=',nmax
        Ckmin=C(nmax)/(2*PI*nmax)
c       print*,'Ckmin=',Ckmin
c       fill up rest of Cm with Ckmin
        do 33 m=nmax+1,N2
        C(m)=Ckmin*2*PI*m
   33        continue
        return
        end
```

```
c       matmpy.for
c
c       performs complex matrix multiplication
c       a*b=ab
c       ra is rows of a
c       ca is columns of a
c       cb is columns of b
c
        subroutine matmpy1(a,b,ab,ra,ca,cb)
        complex a(50,50),b(50,50),ab(50,50),tmp(50,50)
        integer i,j,ra,cb,ca
        do 850 i=1,ra
           do 860 j=1,cb
c       print*,'a(',i,j,')=',a(i,j)
c       print*,'b(',i,j,')=',b(i,j)
           tmp(i,j)=cmplx(0.0,0.0)
                do 870 m=1,ca
                tmp(i,j)=a(i,m)*b(m,j)+tmp(i,j)
  870                continue
  860           continue
  850         continue
        do 880 i=1,ra
           do 890 j=1,cb
           ab(i,j)=tmp(i,j)
c          print*,'ab(',i,j,')= ',ab(i,j)
  890           continue
  880         continue
        return
        end
```

```
      SUBROUTINE MINV1(A,N,D)
C
C     .....................................................................
C
C        SUBROUTINE MINV (from IMSL)
C
C        PURPOSE
C           INVERT A MATRIX (adapted to be complex)
C
C        USAGE
C           CALL MINV(A,N,D,L,M)
C
C        DESCRIPTION OF PARAMETERS
C           A - INPUT MATRIX, DESTROYED IN COMPUTATION AND REPLACED BY
C               RESULTANT INVERSE.
C           N - ORDER OF MATRIX A
C           D - RESULTANT DETERMINANT
C           L - WORK VECTOR OF LENGTH N
C           M - WORK VECTOR OF LENGTH N
C
C        REMARKS
C           MATRIX A MUST BE A GENERAL MATRIX
C
C        SUBROUTINES AND FUNCTION SUBPROGRAMS REQUIRED
C           NONE
C
C        METHOD
C           THE STANDARD GAUSS-JORDAN METHOD IS USED. THE DETERMINANT
C           IS ALSO CALCULATED. A DETERMINANT OF ZERO INDICATES THAT
C           THE MATRIX IS SINGULAR.
C
C     .....................................................................
C
      DIMENSION L(100),M(100)
        complex A(*),BIGA,HOLD,D
C
C        .....................................................................
C
C        IF A DOUBLE PRECISION VERSION OF THIS ROUTINE IS DESIRED, THE
C        C IN COLUMN 1 SHOULD BE REMOVED FROM THE DOUBLE PRECISION
C        STATEMENT WHICH FOLLOWS.
C
C     DOUBLE PRECISION A,D,BIGA,HOLD
C
C        THE C MUST ALSO BE REMOVED FROM DOUBLE PRECISION STATEMENTS
C        APPEARING IN OTHER ROUTINES USED IN CONJUNCTION WITH THIS
C        ROUTINE.
C
```

```
C        THE DOUBLE PRECISION VERSION OF THIS SUBROUTINE MUST ALSO
C        CONTAIN DOUBLE PRECISION FORTRAN FUNCTIONS.  ABS IN STATEMENT
C        10 MUST BE CHANGED TO DABS.
C
C
C        SEARCH FOR LARGEST ELEMENT
C
      D=cmplx(0.0,0.0)
      NK=-N
      DO 80 K=1,N
      NK=NK+N
      L(K)=K
      M(K)=K
      KK=NK+K
      BIGA=A(KK)
      DO 20 J=K,N
      IZ=N*(J-1)
      DO 20 I=K,N
      IJ=IZ+I
c      print*,'A(',IJ,')=',A(IJ)
   10 IF( cABS(BIGA)- cABS(A(IJ))) 15,20,20
   15 BIGA=A(IJ)
      L(K)=I
      M(K)=J
   20 CONTINUE
C
C        INTERCHANGE ROWS
C
      J=L(K)
      IF(J-K) 35,35,25
   25 KI=K-N
      DO 30 I=1,N
      KI=KI+N
      HOLD=-A(KI)
      JI=KI-K+J
      A(KI)=A(JI)
   30 A(JI)=HOLD
C
C        INTERCHANGE COLUMNS
C
   35 I=M(K)
      IF(I-K) 45,45,38
   38 JP=N*(I-1)
      DO 40 J=1,N
      JK=NK+J
      JI=JP+J
      HOLD=-A(JK)
      A(JK)=A(JI)
```

```
   40 A(JI)=HOLD
C
C        DIVIDE COLUMN BY MINUS PIVOT (VALUE OF PIVOT ELEMENT IS
C        CONTAINED IN BIGA)
C
   45 IF(cabs(BIGA).eq.0.0) then
         D=cmplx(0.0,0.0)
         RETURN
      end if
   48 DO 55 I=1,N
      IF(I-K) 50,55,50
   50 IK=NK+I
      A(IK)=A(IK)/(-BIGA)
   55 CONTINUE
C
C        REDUCE MATRIX
C
      DO 65 I=1,N
      IK=NK+I
      HOLD=A(IK)
      IJ=I-N
      DO 65 J=1,N
      IJ=IJ+N
      IF(I-K) 60,65,60
   60 IF(J-K) 62,65,62
   62 KJ=IJ-I+K
      A(IJ)=HOLD*A(KJ)+A(IJ)
   65 CONTINUE
C
C        DIVIDE ROW BY PIVOT
C
      KJ=K-N
      DO 75 J=1,N
      KJ=KJ+N
      IF(J-K) 70,75,70
   70 A(KJ)=A(KJ)/BIGA
   75 CONTINUE
C
C        PRODUCT OF PIVOTS
C
      D=D*BIGA
C
C        REPLACE PIVOT BY RECIPROCAL
C
      A(KK)=cmplx(1.0,0.0)/BIGA
   80 CONTINUE
C
C        FINAL ROW AND COLUMN INTERCHANGE
```

```
C
      K=N
  100 K=(K-1)
      IF(K) 150,150,105
  105 I=L(K)
      IF(I-K) 120,120,108
  108 JQ=N*(K-1)
      JR=N*(I-1)
      DO 110 J=1,N
      JK=JQ+J
      HOLD=A(JK)
      JI=JR+J
      A(JK)=-A(JI)
  110 A(JI)=HOLD
  120 J=M(K)
      IF(J-K) 100,100,125
  125 KI=K-N
      DO 130 I=1,N
      KI=KI+N
      HOLD=A(KI)
      JI=KI-K+J
      A(KI)=-A(JI)
  130 A(JI)=HOLD
      GO TO 100
  150 RETURN
      END
```

```
c               laygen
c
c       this program generates an input file for the homogeneous model
c       which approximates some continuous complex permittivity profile.

        program laygen


        integer   i,j,k,N,num
        real          lam,eox,rw,g,t,e0,s0,e1,s1,e2,s2,t1,stemp,yl
        real          etemp
        character infile*10,ofile*10


        print*,'Name to be given input file? '
        read 10,infile
        print*,'Name to be given output file? '
        read 10,ofile
   10       format (10a)
        print*,'input # of summation terms (fourier modes): '
        read 5,N
        print*,'input # of collocation points: '
        read 5,k
        print*,'Input spatial wavelength: '
        read 4,lam
    4   format(1PE9.3)
        print*,'input insulating layer permittivity: '
        read 4,eox
        print*,'input normalized width (lambda/4h): '
        read 15,rw
   15       format(E6.0)
        print*,'input normalized interelectrode spacing (a/lambda) '
        read 15,g
        print*,'input load impedance: '
        read 7,yl
    7       format(F4.2)
        print*,'Input thickness of layer to be discretized: '
        read 15,t
        print*,'input # of desired discretized points: '
        read 5,num
        print*,'input epsilon and sigma of infinite half space: '
        read 20,e0,s0
   20       format(1PE8.0,1PE8.0)
        print*,'Exponential (1) or linear (2) conductivity profile? '
        read 5,j
    5       format(I4)
        print*,'j=',j
        print*,'input epsilon and sigma of layer at infinite half'
```

```
      print*,'space boundary: '
      read 20,e1,s1
      print*,'input epsilon and sigma of layer at electrode'
      print*,'interface: '
      read 20,e2,s2
      open(unit=3,file=infile)
      write(3,10) ofile
      write(3,5) N
      write(3,5) k
      write(3,4) lam
      write(3,4) eox
      write(3,16) rw
16        format(F6.2)
      print*,'rw= ',rw
      write(3,16) g
      print*,'g= ',g
      write(3,45) yl
45    format(F4.2)
      write(3,32) e2,s2
32    format(1PE8.2,E8.2)
      write(3,35)
35        format('0         0')
      write(3,5) num+2
      write(3,30)
30        format('1000')
      write(3,32) e0,s0
      write(3,35)
      t1=t/num
      do 100 i=num,0,-1
      if (j .eq. 1) then
      print*,'exp: j=',j
              stemp=s2*exp(log(s1/s2)*i*t1/t)
              etemp=e2*exp(log(e1/e2)*i*t1/t)
      else
      print*,'lin: j=',j
              stemp=s2 + (s1-s2)*i*t1/t
              etemp=e2 + (e1-e2)*i*t1/t
      endif
      write(3,40) t1
40        format(1PE6.1E1)
      write(3,32) etemp,stemp
      write(3,35)
100       continue
      write(3,47)
47    format('y')
      end
```

Bibliography

[1] J.C. Zuercher and J.R. Melcher, "Double-Layer Transduction At A Mercury-Electrolyte Interface With Imposed Temporal And Spatial Periodicity", J. of Electrostatics, Vol. 5, pps. 21–31, 1978.

[2] J. Melcher, "Electrohydrodynamic Surface Waves", in *Waves On Fluid Interfaces*, R.E. Meyer, ed., Academic Press:NY, pps. 167–200, 1983.

[3] J.R. Melcher, "Charge Relaxation on a Moving Liquid Interface," Phys. Fluids, No. 10, pps. 325–331, 1967.

[4] S.M. Gasworth, J.R. Melcher and M. Zahn, "Induction Sensing of Electrokinetic Streaming Current", in Conf. on Interfacial Phenomena in Practical Insulating Systems, Nat'l. Bureau of Standards, Gaithersburg, MD, Sept. 19–20, 1983.

[5] A.C.M. Wilson, *Insulating liquids: their uses, manufacture and properties*, Peter Pereginus Ltd., UK, chapter 3, 1980.

[6] ASTM D117–78, "Electrical Insulating Oils of Petroleum Origin", 1978.

[7] R. Baehr, W. Breuer, F. Flottmeyer, J. Kotschnigg, R. Müller, and H. Nieschwietz, "Diagnostic Techniques And Preventive Maintenance Procedures For Large Transformers", CIGRE Conf., Sept., paper 12–13, 1982.

[8] R.R. Rogers, "IEEE And IEC Codes To Interpret Incipient Faults In Transformers, Using Gas In Oil Analysis", IEEE Trans. Electr. Insul., Vol. FI–13, No. 5, Oct. 1978.

[9] C.L.C. Sobral Vieira, "Correlation Between Results of Dissolved Gas Analysis and Actual Transformer Inspection", Doble conf. 53AIC86, Sec. 6D–01, 1986.

[10] R.F. Wolff, "Cut Failures By Analyzing Transformer Oil", Electrical World, Oct. 1, pps. 52–54, 1979.

[11] R.M. Glass, "Dissolved Gas Analysis Of Transformer Oil", Transmission & Distribution, p. 22, Jan. 1977.

[12] F.B. Waddington, "Instruments to Measure the Salient Properties of Fluid Dielectrics", Proc. of 14th Electr./Electronics Insul. Conf., IEEE and NEMA, pps. 22–25, 1979.

[13] T.V. Oomen, "Moisture Equilibrium Charts for Transformer Insulation Drying Practice", IEEE Trans. on Power App. Sys., Vol. PAW-103, No. 10, pps. 3063-3067, Oct., 1984.

[14] T.V. Oomen, "Moisture Equilibrium In Paper-Oil Insulation Systems", IEEE Int. Symp. on Elect. Insul., p. 162, 1983.

[15] F.C. Pratt, "Determination Of Transformer Condition And Preventive Maintenance Procedures In Service", CIGRE Conf., Paper 12-14, Sept., 1982.

[16] A.W. Stannett, "The Measurement of Water in Power Transformers", Proc. IEE, Pt. A, Supp. 3, Vol. 109, pps. 80-85, 1962.

[17] T.V. Oomen, and E.M. Petrie, "Electrostatic Charging Tendency Of Transformer Oils", IEEE Trans. on PAS, Vol. PAS-103, No. 7, July, 1984.

[18] J.P. Crine, M. Duval and C. Lamarre, "Evaluation Of Transformer Oil Aging Under Service Load From Various Chemical And Dielectric Measurements", IEEE Int. Symp. on Elect Insul., p. 35, 1985.

[19] T. Ishii and M. Ueda, "Problems in Checking Condition of Transformer Oil in Service by Dielectric Loss", Conf. on Elec. Insul. and Diel. Phenom., Nat. Acad. Sci., Washington, p. 175, 1975.

[20] A.A. El-Sulaiman, A.S. Ahmed, and M.I. Qureshi, "High Field DC Conduction Current and Spectroscopy Of Aged Transformer Oil", IEEE Trans. on Power App. Sys., Vol. PAS-101, No. 11, p. 4358, Nov., 1982.

[21] T.O. Rouse, "Evaluation of Alternative Insulating Oils for Use in Transformers and Other Electrical Apparatus: Characteristics of Insulating Oils for Electrical Application", Summary Report EL-809-SY, February 1980 prepared by GE for EPRI Research Project 562-1.

[22] R.G. Olivier, C. Vincent and J.P. Crine, "Analysis Of Particles In Transformer Oil By Ferrography", Doble conf. 53AIC86, Sec. 10A-01, 1986.

[23] P. Li, *Low Frequency, Millimeter Wavelength, Interdigital Dielectrometry of Insulating Media in a Transformer Environment*, SM Thesis, EE, MIT, Cambridge, MA, May, 1987.

[24] T.W. Hayes, "Investigation To Determine The Location Of A Low-Energy, Audible Electrical Arcing In A Power Transformer", Doble conf., 53AIC86, Sec. 6A-01, 1986.

[25] R.L. Spells, "Failure Of A General Electric 150/200/250 MVA, 512GR.Y/295.6-230GR.Y/132.79-13.8KV, Single-Phase Autotransformer With Tertiary Winding", Doble conf., Sec. 6B-01, 1986.

[26] P.W. Brunson, Jr., "Investigation Of The Failure Of A 500-230 KV, 480 MVA Westinghouse Shell-Type Power Transformer", Doble conf., 53AIC86, Sec. 6F-01, 1986.

[27] M. Thibault-Carballeira, D. Allaire, J. Delhaye, P. Moro and J. Samat, "Fault Detection And Location In Transformers", CIGRE Conf., paper 12-01, Sept., 1982.

[28] M.A. Salvant, "Transformer Incipient Fault Confirmed By Dissolved Gas-In-Oil Analysis", Doble conf., 44AIC77, Sec. 6A-01, 1977.

[29] A.M. Corvo, "Diagnostic Technique And Proceedings Of Preventive Maintenance Of Large Transformers", CIGRE Conf., paper 12-11, Sept., 1982.

[30] P.J. Burton, J. Graham, A.C. Hall, J.A. Laver and A.J. Oliver, "Recent Developments By CEGB To Improve The Prediction And Monitoring Of Transformer Performance", CIGRE Conf., paper 12-09, Sept., 1984.

[31] G. Bélanger, G. Missout and J.P. Gibeault, "Laboratory Testing Of A Sensor For Hydrogen Dissolved In Transformer Oil", IEEE Trans. on Elect. Insul., Vol. EI-15, No. 2, p. 144, April, 1980.

[32] R.D. Sparks and J.H. Turney, "Tennessee Valley Authority's Application Of The Hydran 201R On-line Hydrogen Monitor To 500 KV Current Transformers", Doble conf. 53AIC86, Sec. 6H-01, 1986.

[33] W. McDermid and J.F. Millward, "On-line Hydrogen-In-Line Monitoring (A Progress Report)", Doble conf. 52AIC85, Sec. 6G-01, 1985.

[34] T.D. Poyser, D.A. Yannucci, J.B. Templeton and B.N. Lenderking, "On-line Monitoring of Power Transformers", IEEE Trans. on PAS, Vol. PAS-104, No. 1, p. 207, Jan., 1985.

[35] A.J. Polak, S. Petty-Weeks and A.J. Beuhler, "Applications of Novel Proton-Conducting Polymers to Hydrogen Sensing", Sensors and Actuators, Vol. 9, pps. 1-7, 1986.

[36] K. Ihokura, "Tin Oxide Gas Sensor for Deoxidizing Gas", New Materials and New Processes in Electrochem Tech., Vol. 1, pps. 43-50, 1981.

[37] T. Seiyama, K. Fueki, J. Shiokawa and S. Suzuki, *Chemical Sensors*, Proc. Int'l, Meeting on Chemical Sensors, Fukuoka, Japan, Sept. 19-22, 1983, Analyt. Chem. Symp. Series, Vol. 17, Elsevier Press:NY, 1983.

[38] P.K. Clifford and D.T. Tuma, "Characteristics of Semiconductor Gas Sensors, I. Steady State Gas Response", Sensors and Actuators, Vol. 3, pps. 233-254, 1982/3.

[39] P.K. Clifford and D.T. Tuma, "Characteristics of Semiconductor Gas Sensors, II. Transient Response to Temperature Change", Sensors and Actuators, Vol. 3, pps. 255-281, 1982/3

[40] G.N. Advani and A.G. Jordan, "Thin Films of SnO_2 as Solid State Gas Sensors", J. of Electron. Mater., Vol. 9, No. 1, pps. 29-49, 1980.

[41] T. Oyabu, "Sensing Characteristics of SnO_2 Thin Film Gas Sensor", J. Appl. Phys., Vol. 53, No. 4, pps. 2785–7, April, 1982.

[42] S. Chang, "Sensing Mechanisms of Thin Film Tin Oxide", in [37], pps. 78–83.

[43] H. Windischmann and P. Mark, "A Model for the Operation of a Thin-Film SnO_x Conductance-Modulation Carbon Monoxide Sensor", J. Electrochem. Soc., Vol. 126, No. 4, pps. 627–633, April, 1979.

[44] B. Keramati and J.N. Zemel, "Pd-Thin-SiO_2-Si diode. I. Isothermal Variation of H_2-Induced Interfacial Trapping States", J. Appl. Phys., Vol. 53, No. 2, pps. 1091–1099, Feb., 1982.

[45] P.F. Ruths, S. Ashok, S.J. Fonash and J.M. Ruths, "A Study of Pd/Si MIS Schottky Barrier Diode Hydrogen Detector", IEEE Trans. on Electron Devices, Vol. ED-28, No. 9, pps. 1003–1009, Sept., 1981.

[46] K. Dobos and G. Zimmer, "Performance of Carbon Monoxide-Sensitive MOSFET's with Metal-Oxide Semiconductor Gates", IEEE Trans. on Electron Devices, Vol. ED-32, No. 7, pps. 1165–1169, July, 1985.

[47] G.J. Maclay, "A Dual-Mechanism Solid-State Carbon-Monoxide and Hydrogen Sensor Utilizing an Ultrathin Layer of Palladium", IEEE Trans. on Electron Dev., Vol. ED-34, No. 10, pps. 2086–2097, Oct., 1987.

[48] T.L. Poteat and B. Lalevic, "Pd-MOS Hydrogen and Hydrocarbon Sensor Device", IEEE Electron Device Lett., Vol. EDL-2, No. 4, pps. 82–84, April, 1981.

[49] N. Yamamoto, S. Tonomura, T. Matsuoka andH. Tsubomura, "A Study on a Palladium-Titanium Oxide Schottky Diode as a Detector For Gaseous Components", Surface Science, Vol. 92, pps. 400-406, 1980.

[50] I. Lundström, "Hydrogen Sensitive MOS-Structures Part I: Principles and Applications", Sensors and Actuators, Vol. 1, pps. 403–426, 1981.

[51] S.J. Fonash, H. Huston and S. Ashok, "Conducting MIS Diode Gas Detectors: The Pd/SiO_x/Si Hydrogen Sensor", Sensors and Actuators, Vol. 2, pps. 363–369, 1982.

[52] I. Lundström and C. Svensson, "Gas-Sensitive Metal Gate Semiconductor Devices", in *Solid State Chemical Sensors*, ed. by J. Janata adn R.J. Huber, Academic Press, pps. 1–63, 1985.

[53] B. Keramati and J.N. Zemel, "Pd-Thin-SiO_2-Si diode. II. Theoretical Modeling and the H_2 Response", J. Appl. Phys., Vol. 53, No. 2, pps. 1100–1109, Feb., 1982.

[54] N.J. Evans, M.C. Petty and G.G. Roberts, "Interface State Effects In Pd-Gate MOS Hydrogen Sensors", Sensors and Actuators, Vol. 9, pps. 165-175, 1986.

[55] C. Nylander, M. Armgarth and C. Svensson, "Hydrogen Induced Drift in Palladium Gate Metal–Oxide–Semiconductor Structures", J. Appl. Phys., Vol. 56, No. 4, pps. 1177–1188, Aug., 1984.

[56] M. Yamada, Y. Nomura, Y. Katayama, T. Ishii, O. Imamura and K. Tsuchiyama, "Automatic Field Monitoring Of Dissolved Gases In Transformer Oil", IEEE Trans. on PAS, Vol. PAS–1000, No. 4, p. 1538, April, 1981.

[57] H. Tsukioka, K. Sugawara, E. Mori, S. Hukomori and S. Sakai, "New Apparatus For Detecting H_2, CO and CH_4 Dissolved In Transformer Oil", IEEE Trans. on Elect. Insul., Vol. EI–18, No. 4, p. 409, Aug., 1983.

[58] R.A. Kurz, "On–line Incipient Fault Detector", Proc. of the 13th Electrical/Electronic Insulation Conf., pps. 174–178, 1977.

[59] R.A. Kurz, "On–line Incipient Fault Detector (Field Follow Experience)", Proc. of the 15th Electrical/Electronic Insulation Conf., pps. 1–5, 1981.

[60] H. Tsukioka, K. Sugawara and E. Mori, "Study On Continuous Monitoring Of Hydrogen Gas Dissolved In Transformer Oil", Elect. Eng. in Japan, Vol. 100, No. 1, p. 1, 1980.

[61] H. Tsukioka, K. Sugawara and E. Mori, "Apparatus For Continuousiy Monitoring Hydrogen Gas Dissolved In Transformer Oil", IEEE Trans. on Elect. Insul., Vol. EI–16, No. 6, p. 502, Dec., 1981.

[62] H. Tsukioka, K. Sugawara, E. Mori and H. Yamaguchi, " New Apparatus For Detecting Transformer Faults", IEEE Trans. on Electr. Insul., Vol. EI–21, No. 2, p. 221, April, 1986.

[63] B. Herget, Nicolet, Madison, WI, private communication.

[64] N.F. Sheppard, Jr., D.R. Day, H.L. Lee and S.D. Senturia, "Microdielectrometry", Sensors and Actuators, Vol. 2, pps. 263–274, 1982.

[65] S.D. Senturia and S.L. Garverick, "Method and Apparatus for Microdielectrometry", U.S. Patent No. 4,423,371, Dec. 27, 1983, assigned to MIT, licensed to Micromet Instruments, Inc.

[66] Low Conductivity Sensor, Model S-20, available from Micromet Instruments Inc., 21 Erie St., Suite # 22, Cambridge, MA 02139.

[67] Eumetric SYSTEM II Microdielectrometer manufactured by Micromet Instruments Inc., 21 Erie St., Suite # 22, Cambridge, MA 02139.

[68] M.C.W. Coln, *A High Performance Dielectric Measurement System*, Phd Thesis, EE, MIT, Cambridge, MA, June, 1985.

[69] S.D., Senturia, N.F. Sheppard, Jr., H.L. Lee and D.R. Day, "In–situ Measurement of the Properties of Curing Systems with Microdielectrometry", J. Adhesion, Vol. 15, pps. 69–90, 1982.

[70] H.L. Lee, *Optimization of a Resin Cure Sensor*, EE Thesis, MIT, Cambridge, MA, Aug., 1982.

[71] S.L. Garverick and S.D. Senturia, "An MOS Device for AC Measurement of Surface Impedance with Application to Moisture Monitoring", IEEE Trans. on Electron Devices, Vol. ED-29, No. 1, pps. 90-94, Jan., 1982.

[72] T.M. Davidson and S.D. Senturia, "The Moisture Dependence of the Electrical Sheet Resistance of Aluminum Oxide Thin Films with Application to Integrated Moisture Sensors", in Proc. IEEE, Int'l. Reliability Physics Symp., San Diego, CA, pps. 249-252, March, 1982.

[73] L. Mouayad, *Monitoring of Transformer Oil using Microdielectrometric Sensors*, SM Thesis, EE, MIT, Cambridge, MA, Feb., 1985.

[74] High Temperature Ceramic Sensor, Model S-60, available from Micromet Instruments Inc., 21 Erie St., Suite # 22, Cambridge, MA 02139.

[75] J.R. Melcher, *Continuum Electromechanics*, MIT Press:MA, 1981.

[76] H. Engan, "Excitation of Elastic Surface Waves by Spatial Harmonics of Interdigital Transducers", IEEE Trans. on Electron Devices, Vol. ED-16, No. 12, pps. 1014-1017, Dec., 1969.

[77] S. Senturia, private communication.

[78] L. Ljung, *System Identification: Theory for the User*, Prentice-Hall, Inc.:NJ, 1987.

[79] T.R. Cuthbert, Jr., *Optimization Using Personal Computers*, Wiley:NY, 1987.

[80] W.H. Press, B.P. Flannery, S.A. Teukolsky and W.T. Vetterling, *Numerical Recipes, The Art of Scientific Computing*, Cambridge University Press:NY, 1986.

[81] Stuart Inkpen, personal communication

[82] J.R. Melcher and M.C. Zaretsky, "Apparatus and Methods For Measuring Permittivity In Materials", U.S. Patent Application Serial No. 104,179, Oct. 2, 1987.

[83] H. Yasuda, *Plasma Polymerization*, Academic Press:Florida, 1985.

[84] G.E. Loeb, M.J. Bak, M. Salcman, and E.M. Schmidt, "Parylene as a Chronically Stable, Reproducible Microelectrode Insulator", IEEE Trans. Biomed. Eng., Vol. BME-24, No. 2, pps. 121-128, March, 1977.

[85] Paratronix, Inc., 129 Bank St., Attleboro, MA, 02703.

[86] G.W. Castellan, *Physical Chemistry*, 3rd edition, Addison-Wesley, MA, 1983.

[87] J. Bingeli, J. Froidevaux and R. Kratzer, "The Treatment of Transformers, Quality and Completion Criteria and the Process", CIGRE Paper No. 110, 1966.

[88] R.J. McGill, *Measurement and Control in Papermaking*, Adam Hilger Ltd., England, 1980.

[89] R. Jeffries, "The Sorption of Water by Cellulose and Eight Other Textile Polymers", J. of the Textile Inst. Trans., Vol. 51, No. 9, pps. 339–374, 1960.

[90] L. Szepes, K. Torkos, R. Dobó and Á. Székely, "A New Analytical Method for the Determination of the Water Conent of Transformer Oils", IEEE Trans. on Electr. Insul., Vol. EI–17, No. 4, pps. 345–349, August, 1982.

[91] A. Benatar, *Effects of Surface Modification of Graphite Fibers by Plasma Treatment on the Hygral Behaviour of Composites*, SM Thesis, ME, MIT, Cambridge, MA, May, 1983.

[92] J. Tyczkowski and M. Kryszewski, "Electrical Conduction in Plasma–Polymerized Organosilicon Films: Influence of Water and Oxygen", J. Appl. Polym. Sci.: Appl. Polym. Symp. 38, pps. 149 161, 1984.

[93] S.M. Gasworth, J.R. Melcher and M. Zahn, "Flow–Induced Charge Accumulation and Field Generation in Thin Insulating Tubes", Conf. on Interfacial Phenomena in Practical Insulating Systems, NBS, Gaithersburg, MD, Sept. 19–20, 1983.

[94] A.F. Howe, "Diffusion of Moisture Through Power–Transformer Insulation", Proc. IEE, Vol. 125, No. 10, pps. 978–986, Oct., 1978.

[95] J. Grebowicz, T. Pakula, A.M. Wróbel and M. Kryszewski, "Electron Microscopy Studies of Plasma–Polymerized Organosilicon Thin Films", Thin Solid Films, Vol. 65, pps. 351–359, 1980.

[96] M.R. Wertheimer, J.E. Klemberg–Sapieha and H.P. Schreiber, "Advances in Basic and Applied Aspects of Microwave Plasma Polymerization", Thin Solid Films, Vol. 115, pps. 109–124, 1984.

[97] T.S. Ramu, M.R. Wertheimer and J.E. Klemberg–Sapieha, "Dielectric Properties of Plasma–Polymerized Hexamethyldisiloxane Films: 1 Complex Permittivity", IEEE Trans. on Electr. Insul., Vol. EI–21, No. 4, pps. 549–556, Aug., 1986.

[98] V.V. Daniel, *Dielectric Relaxation*, Academic Press, NY, 1967.

[99] D.R. Day, T.J. Lewis, H.L. Lee and S.D. Senturia, "The Role of Boundary Layer Capacitance at Blocking Electrodes in the Interpretation of Dielectric Cure Data in Adhesives", J. Adhesion, Vol. 18, p. 18, 1973.

[100] H. Scher, "Dispersive Transport and Relaxation", Proc. 2nd Int. Conf. on Cond. and Breakdown in Sol. Diel., IEEE/Elect. Insul. Soc., pps. 13–21, 1986.

[101] H. Scher and E. Montroll, "Anomalous transit–time dispersion in amorphous solids", Phys. Rev. B, Vol. 12, No. 6, pps. 2455–2477, Sept., 1985.

[102] M.F. Shlesinger, "Williams–Watts Dielectric Relaxation: A Fractal Time Stochastic Process", J. Stat. Phys., Vol. 36, Nos. 5/6, pps. 639–648, 1984.

[103] A.K. Jonscher, "Maxwell–Wagner Effect and Strong Low–Frequency Dispersion", IEEE Conf. on Electr. Insul. and Diel. Phenom., Annual Report, pps. 479–485, 1983.

[104] A.K. Jonscher, *Dielectric Relaxation in Solids*, Chelsea Dielectric Press, London, 1983.

[105] A.K. Jonscher, "Alternating Current Diagnostics of Poorly Conducting Thin Films", Thin Solid Films, Vol. 36, pps. 1–20, 1976.

[106] U. Hetzler and E. Kay, "Conduction Mechanism in Plasma–Polymerized Tetrafluoroethylene Films". J. App. Phys., Vol. 49, No. 11, pps. 5617–5623, Nov., 1978.

[107] R.E. Barker, "Mobility and Conductivity of Ions in and into Polymeric Solids", J. Pure and Appl. Chem., Vol. 46, pps. 157–170, 1976.

[108] S. Morita, G. Sawa and M. Ieda, "Influence of oxygen on electrical properties of styrene thin film polymerized in a glow discharge", J. Appl. Phys., Vol. 44, No. 5, pps. 2435–2436, May, 1973.

[109] L.S. Tuzov, V.M. Kolotyrkin and N.N. Tunitskii, "Stability of the dielectric properties of polymer films formed in a glow discharge", Int. Chem. Eng., Vol. 11, No. 1, pps. 60–64, Jan., 1971.

[110] F.A. Lewis, *The Palladium Hydrogen System*, Academic Press:NY, 1967.

[111] R. Suhrmann, G. Wedler and G. Schumicki, "Influence of Hydrogen on the Electrical Resistance of Evaporated Palladium Films", in *Proc. of Int'l. Conf. on Structure and Properties of Thin Films*, ed. by A.C. Neugebauer, Bolton Landing, NY, Wiley:NY, pps. 268–281, 1959.

[112] W. Auer and H.J. Grabke, "The Kinetics of Hydrogen Absorption in Palladium (α– and β–phase) and Palladium–Silver–Alloys", Deutsche Berichte Der Bunsen Gesellschaft Für Physikalische Chemie, Vol. 78, pps. 58–67, 1974.

[113] H. Züchner and N. Boes, "", Deutsche Berichte Der Bunsen Gesellschaft Für Physikalische Chemie, Vol. 76, pps. 783–786, 1972.

[114] N. Inagaki and Y. Hashimoto, "Gas Sensor Devices Plasma–Polymerized From Organotin Compound", in Proc. of ACS Div. of Polymeric Mat.: Sci. and Eng., Vol. 56, Denver, CO, pps. 515–519, Spring Mtg., 1987.

[115] M.S. Wrighton, J.W. Thackeray, M.J. Natan, D.K. Smith, G.A. Lane and D. Bélanger, "Modification of Microelectrode Arrays: New Microelectrochemical Devices for Sensor Applications", submitted to Philosophical Trans. of the Royal Society, Series B, 1986.

[116] D.W. Crofts, "The Static Electrification Phenomena in Power Transformers", Ann. Rep. of the CEIDP, pps. 223–236, Nov., 1986.

[117] J. Melcher, D. Lyon and M. Zahn, "Flow Electrification in Transformer Oil/Cellulosic Systems", Ann. Rep. of the CEIDP, pps. 257–265, Nov., 1986.

[118] F. Oberhettinger, *Fourier Expansions, A Collection of Formulas*, Academic Press:NY, p. 9, formula 1.25, 1973.

[119] W.H. Beyer, *CRC Standard Math Tables*, CRC Press:FL, p. 350, 1981.

[120] A.H. Bailey, D.A. Darbyshire, A.P. Overbury, C.W. Pitt and J. Newton, "Plasma Deposition of metal oxide films for integrated optics", Vacuum, Vol. 36, Nos. 1–3, pps. 139–142, 1986.

[121] A.F. Stancell and A.T. Spencer, " Composite Permselective Membrane by Deposition of an Ultrathin Coating from a Plasma", J. Appl. Polym. Sci., Vol. 16, pps. 1505–1515, 1972.

[122] T. Wydeven and J.R. Hollahan, "Semipermeable Membranes Produced by Plasma Polymerization", in *Techniques and Applications of Plasma Chemistry* ed. by J.R. Hollahan and A.T. Bell, Wiley:NY, chap. 5, 1974.

[123] T. Yasuda, M. Gazicki and H. Yasuda, "Effects of Glow Discharges on Fibers and Fabrics", J. Appl. Polym. Sci.: Appl. Polym. Symp. 38, pps. 201–214, 1984.

[124] M.F. Bottin, H.P. Schreiber, J. Klemberg–Sapieha and M.R. Wertheimer, "Modification of Paper Surface Properties by Microwave Plasma", J. Appl. Polym. Sci.: Appl. Polym. Symp. 38, pps. 193–200, 1984.

[125] H. Yasuda, "Plasma Polymerization for Protective Coatings and Composite Membranes", J. of Membrane Sci., Vol. 18, pps. 273–284, 1984.

[126] M. Millard, "Synthesis of Organic Polymer Films in Plasma", in *Techniques and Applications of Plasma Chemistry* ed. by J.R. Hollahan and A.T. Bell, Wiley:NY, chap. 5, 1974.

[127] V.M. Kolotyrkin, A.B. Gil'man and A.K. Tsapuk, "Production of Organic Surface Films by the Action of Electrons, Ultraviolet Radiation and the Glow Discharge", Russian Chem. Reviews, Vol. 36, No. 8, pps. 579–591, Aug., 1967.

[128] M.R. Havens, M.E. Biolsi and K.G. Mayhan, "Survey of low temperature rf plasma polymerization and processing", J. Vac. Sci. Technol., Vol. 13, No. 2, pps. 575–584, Mar./April, 1976.

[129] H. Yasuda, "Glow Discharge Polymerization", in *Thin Film Processes*, ed. by J.L. Vossen, Academic Press:Florida, pps. 362–398, 1978.

[130] R.J. Jensen, A.T. Bell and D.S. Soong, "Plasma Polymerization of Ethane. I. Experimental Studies of Effluent Gas Composition and Polymer Deposition Rates", Plasma Chemistry and Plasma Processing, Vol. 3, No. 2, pps. 139–161, 1983.

[131] A,M, Wróbel, M.R. Wertheimer, J. Dib and H.P. Schreiber, "Polymerization of Organosilicones in Microwave Discharges", J. Macromol. Sci. – Chem., Vol. A14, No. 3, pps. 321–337, 1980.

[132] M. Kryszewski, A.M. Wróbel and J. Tyczkowski, "Plasma–Polymerized Organosilicon Thin Films – Structure and Properties", in *Plasma Polymerization*, ed. by M. Shen and A.T. Bell, ACS Symp. Ser. 108, Am. Chem. Soc., Washington, DC, chap. 13, 1979.

[133] Fred W. Billmeyer, Jr., *Textbook of Polymer Science*, Interscience Publishers, Wiley:NY, p. 6, 1966.

[134] *The Radio Amateur's Handbook*, American Radio Relay League, 47th Edition, p. 26, 1970.

[135] B. Chapman, *Glow Discharge Processes*, J. Wiley:NY, p. 16, 1980.

[136] K.J. Vetter, *Electrochemical Kinetics, Theoretical Aspects*, Academic Press:NY, 1967.

[137] Shell Chemical Company, Technical Bulletin, sc:164–177, 1977.