T-2910

DEVELOPMENT OF TRANSIENT ELECTROMAGNETIC PHYSICAL MODELING FOR GEOPHYSICAL EXPLORATION

by

Akira Saito

A thesis submitted to the Faculty and the Board of Trustees of the Colorado School of Mines in partial fulfillments of the requirement of the degree of Doctor of Philosophy (Geophysics)

Golden, Colorado

Date Nov. 19, 1984

Signed: A. Saito
Akira Saito

Approved: [signature]
Dr. Alexander A. Kaufman
Thesis Advisor

Golden, Colorado

Date 21 Nov 84

Signed: [signature]
Dr. Phillip R. Romig
Professor, Head of
Geophysics Department

ABSTRACT

Electromagnetic physical modeling technique for geophysical exploration has been used for many years to study the actual electromagnetic systems through a small scale model, and has proved to be an extremely useful tool.

The basic principle of physical modeling, called electromagnetic similitude, is usually derived from Maxwell's equations, but the relations of parameters or measured voltages between nature and the model are not clear. Therefore, starting from well known equations of electromagnetic theory, clearer relations were developed in Chapter 1.

Due to the progress in microprocessor techniques, very flexible and capable physical modeling systems were developed. The ramp time of the inverter, which was developed by using a high voltage transistor, was about 35 microseconds. High gain and stable amplifier was designed by using OP-27 operational amplifier chips and ICL7650 chopper stabilized amplifiers. A data acquisition system was developed by using Motorolla M6809 microprocessor and an Analog devices AD578 A/D converter chip. Maximum sampling rate of 50 KHz was established with 12-bit resolution. Developed systems

were explained in detail, and the accuracy of these systems was checked using a conducting sphere.

For mining or geothermal exploration, the time constant of the field data is very important in estimating the size and conductivity of the ore body or geothermal reservoir. Since the time constant of rectangular plates was not known, physical modeling experiments were performed and an empirical equation was obtained.

If a receiver is far from a conducting plate, field due to the current in the plate create approximately the same field due to the current dipole at the center of the plate. And an empirical equation for the moment of plate was obtained, which allows us to calculate the field very easily.

Profiles of vertical thin plate and vertical thick plate were explained for the interpretation of field data. And in case of a vertical thick plate, reversal of the polarity of the measured voltage was observed at early time. From the shift of the zero crossing in the profile, the thickness of the plate can be estimated.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

TABLE OF NOTATIONS

$\bar{A}$	vector potential
a	frequency scale factor ($\frac{\omega_m}{\omega_n}$)
a	radius of sphere
$\bar{B}$	magnetic flux density
b	geometric scale factor ($\frac{r_m}{r_n}$)
b	length of smaller side of rectangular plate
c	time scale factor ($\frac{t_m}{t_n}$)
$\bar{D}$	electric flux density (electric displacement)
$\bar{E}$	electrical field
f	frequency
$\bar{H}$	magnetic field
h	dimensionless function
I	current
i	imaginary number ($i^2 = -1$)
$I_{\frac{1}{2}}$	associated Bessel function
$I_{-\frac{1}{2}}$	associated Bessel function
$\bar{j}$	current density
k	wave number
L	inductance
M_P	moment of rectangular plate
M_R	moment of receiver coil
M_T	moment of transmitter coil

m	suffix for the model
n	suffix for nature
q_1	first pole
R	distance between dipole source and observation point
r	geometrical parameter
t	time
V	complex amplitude of the electromotive force
V_o	primary electromotive force
Z	depth of ore body
α	normalized resistivity
α	$= \frac{1}{\sigma \mu a^2}$
β	normalized geometrical dimansion
γ	normalized magnetic permeability
δ	skin depth ($\sqrt{\frac{2}{\sigma \mu \omega}}$)
σ	conductivity
μ	magnetic permeability
τ_o	time constant
ω	angular frequency

ACKNOWLEDGEMENTS

The author would like to express deepest gratitude to Professor Alexander A. Kaufman for suggesting this thesis topic and for continuing guidance and encouragement.

The author also would like to thank Professor George V. Keller for the helpful suggestions, extensive corrections of the content and supporting this research.

The author is grateful to Professors Catherine Skokan, Charles Baer, James Pritchard and Robert Fisk for their suggestions and observations.

The author also wishes to express his gratitude to Professor Samuel B. Romberger, Dean of Graduate Studies, to be a thesis committee member and helping him to complete this research.

The author also would like to thank Emeritus Professor E. Yoshizumi, Professors K. Sassa and T. Sugano of Kyoto University, Japan for giving him the opportunity of studying at the Colorado School of Mines.

INTRODUCTION

Electromagnetic methods in geophysical prospecting have been used since the late 1910s to solve structual, mining, well logging, geothermal and engineering problems.

For the structural problem, magnetotelluric, frequency and transient sounding methods are among the techniques used to determine the depth of basement, resistivity and thickness of layers. The theory of these methods is mainly developed for a horizontally layered medium and interpretation is mainly developed for this type of problem. But for more complicated structures, which are often met in practice like a vertical contact, inclined layers, topography and heterogeneity within layers, the analytical approach is limited due to the complexity of the boundary value problems.

For mining geophysics, targets are usually conductive ore bodies of complicated shape, and analytical solutions for such a type of model can be obtained for some very limited cases like conductors of simple shape such as a sphere, ellipsoid, dyke or cylinder in free space or layered medium.

The induction method was introduced to well logging by Doll(1949) and has been developed to be one of the most commonly used resistivity tools to determine the electrical parameters of rocks penetrated by wells. In these days, very sophisticated multi-coil tools are used (Allaud 1977).

The electromagnetic physical modeling technique for geophysical exploration has been used for many years in studying actual electromagnetic systems through the use of small scale models. Many problems which are very difficult to solve analytically can be solved easily using physical models. Physical modeling has been used since the early stage of the development of the electromagnetic method (Slichter 1932), and has proved to be an extremely useful tool.

The basic principles of physical modeling experiments are explained in Chapter 1, including the principle of similitude which applies to scale modeling. Recent progress in microcomputer techniques allows us to develop very flexible systems for model experiments, and one of the objectives of this thesis is to develop a physical modeling system allowing the analysis of electromagnetic anomalies for various geoelectrical conditions. Instruments and related software developed are explained in detail in Chapter 2, and a check of accuracy using a conducting sphere is explained in Chapter 3. Also, some examples such as a conducting thin plate, and a conducting thick plate are explained in Chapter 4.

CHAPTER 1

ELECTROMAGNETIC PHYSICAL MODELING

1.1 History of physical modeling

1.1.1 Early stage of the development of the electromagnetic method

The earliest attempt to use the electromagnetic method was done by Harry R. Conklin, an American mining and electric engineer, in 1917 (Conklin 1917, Kelly 1950). A circular primary loop of 200 ft. in diameter was used, and ore bodies were determined from the distortion of equi-magnetic-intensity lines which were drawn by using two receiver coils connected in series. Conklin obtained a patent on the method and made the first experiments in Missouri (Heiland 1938).

At about the same time, development of the electromagnetic method was begun by H. Lundberg et al. in Sweden. Bergholm conducted the first model experiments in 1918. In following years, many ore deposits were discovered by Sundberg et al.(1925,1930) in Sweden using this method. They used a horizontal rectangular loop as a transmitter (about 200 by 100 m) and measured inphase and quadrature components of vertical and horizontal magnetic fields along transverse lines. And this method was introduced into America by the Swedish American Prospecting Corporation, and used for mining and oil explorations (Wilson 1929, Zuschlag 1932, Focken 1937).

In addition to measuring the intensity of the field, directions of the major and minor axis of the field were also measured by Sundberg et al., and theoretical investigation of the method was done by Watson (1931).

The Elbof method, which uses current electrodes as a transmitter and measure the direction of the magnetic field by nulling sound from a receiver coil (Mueser 1926), was used by Eve and Keys to investigate ore deposits at Caribou, Colorado (Eve and Keys 1928).

1.1.2 History of frequency domain physical modeling

The physical modeling technique has been used since the early stage of the development of the method. For example, Slichter(1932) explained the principles of similitude, which are derived from Maxwell's equations, for electromagnetic physical modeling. By using a physical model (3ft. diameter conducting sphere, and 13,000 to 50,000 Hz transmitter current), the response of a conducting ore body was investigated.

Bruckshaw(1936) measured the magnetic field due to conducting circular disks in a uniform magnetic field.

Hedstrom(1940) devised the Turam method, which makes use of phase measurements, with field examples obtained by the Swedish Geological Survey. Physical models for a rectangular dyke were used to interpret the field data.

Thereafter many physical modeling experiments were performed in the frequency domain. For example, the response of a conducting disk was studied by Douloff(1960), and by Poddar et al.(1966). The effect of overburden was studied by Lowrie and West(1965). Fraser and Ward(1967) published modeling results for a conducting cylinder, dike and sphere. The slingram method was investigated by using metal sheet models (Ketola and Puranen 1967). Frischknecht(1971) performed physical modeling with conducting bodies in a conducting medium using a water tank.

For the magnetotelluric method, analogue models have also been used to investigate the effects of vertical faults and dikes (Dosso 1966), and the effects of the ocean coast-line (Hermance 1968).

1.1.3 History of transient physical modeling

Although the early stage of the development of the electromagnetic method was based on the frequency domain theory, the transient response was also investigated in early 1930s (Riordan 1931). Yost et al.(1952) used physical models to investigate the transient response of the earth.

Barringer (1962) developed INPUT transient airborne system, and physical models have been used intensively to investigate the responses of various types of ore bodies. For example, Becker et al.(1972) studied the transient response of a conducting dike using physical models.

For the ground system, Ramaprasada Rao(1973), Spies(1976) and Ogilvy(1983) showed the transient response of a conducting plate for a coincident loop system. Doll et al.(1984) used physical models of layers made by lead and copper plates to investigate the transient response of layered earth. Frischknecht has been studying the transient response of various ore bodies in the conducting medium using a water tank.

1.2 Similitude of electromagnetic physical modeling

In order to use physical modeling, the dimensions of modeling must either (1) coincide completely, or (2) differ only by constant, with the dimensions of the measurement in nature. For this purpose, the theory called electromagnetic similitude has been derived and described in many papers (for example, Slichter 1932, Stratton 1941, Sinclair 1948, Frischknecht 1971, Spies 1976, and so on).

This theory, which can be derived from Maxwell's equations, and which is explained in Appendix A, shows that the results of physical modeling can be used to relate electromagnetic fields to a different geoelectric section provided that certain parameters are kept constant in both cases.

In this chapter, these conditions will be formulated for both frequency and time domain by using the equations for an electromagnetic field in a conducting medium.

1.2.1 Frequency domain

As is well known, the complex amplitude of any component of a magnetic field, $\bar{H}$, can be expressed for frequency domain as follows :

$$\bar{H} = Hoh\left(\frac{r_1}{\delta_1}, \alpha, \beta, \gamma\right) \tag{1.1}$$

where Ho : any component of primary magnetic field

h : dimensionless function which is different for each vector component

r_1 : one of the geometric parameters, for example, separation between transmitter and receiver, or the thickness of the first layer and so on

$\alpha = \frac{\rho_2}{\rho_1}, \frac{\rho_3}{\rho_1}, \ldots$: resistivities normalized by the resistivity of one of the parts of the medium

$\beta = \frac{r_2}{r_1}, \frac{r_3}{r_1}, \ldots$: dimensions normalized by r

$\gamma = \frac{\mu_2}{\mu_1}, \frac{\mu_3}{\mu_1}, \ldots$: magnetic permeabilities normalized with respect to the magnetic permeability of one medium

δ_1: skin depth of the medium with resistivity and expressed as :

$$\delta_1 = \sqrt{\frac{2}{\sigma_1 \mu_1 \omega}} \tag{1.2}$$

where σ_1 : conductivity of medium 1

μ_1 : magnetic permeability of medium 1

ω : angular frequency

Therefore, if we use the subscripts n and m to represent the nature and model respectively, we have the following equations :

$$H_n = H_{on}\, h_n\left(\frac{r_{1n}}{\delta_{1n}}, \alpha_n, \beta_n, \gamma_n\right) \tag{1.3}$$

and

$$H_m = H_{om}\, h_m\left(\frac{r_{1m}}{\delta_{1m}}, \alpha_m, \beta_m, \gamma_m\right) \tag{1.4}$$

Performing modeling, we keep the ratio of resistivities (α), dimensions (β) and magnetic permeabilities (γ) to be the same for nature and its model:

$$\alpha_n = \alpha_m \tag{1.5}$$

$$\beta_n = \beta_m \tag{1.6}$$

$$\gamma_n = \gamma_m \tag{1.7}$$

Therefore, two functions h_n and h_m are the same if the following relation is valid :

$$\frac{r_{1n}}{\delta_{1n}} = \frac{r_{1m}}{\delta_{1m}}$$

and, by substituting equation (1.2) for the skin depth :

$$\sigma_{n1}\mu_{n1}\omega_n r_{n1}^2 = \sigma_{m1}\mu_{m1}\omega_m r_{m1}^2 \tag{1.8}$$

As follows from equations (1.5) through (1.8), we have the same relation for every medium, that is :

$$\sigma_n \mu_n \omega_n r_n^2 = \sigma_m \mu_m \omega_m r_m^2 \tag{1.9}$$

Thus, for the same type of arrays, when the same component of the field is measured, the following relation yields :

$$\frac{H_n}{H_{on}} = \frac{H_m}{H_{om}} \tag{1.10}$$

This equation shows that the fields measured in nature and in the model coincide with each other when they are normalized to their primary field strength.

The primary field H_{om} can be either measured or calculated, while H_m is measured by the modeling. Since the primary field in nature H_{on} can be calculated, the magnetic field in nature H_n can be obtained. This is the purpose of physical modeling.

Usually, in practice, the electromotive force in a receiver coil is measured. Therefore, the relation of the electromotive forces for nature and model should be investigated. As is well known, the magnetic field is the real part of following equation :

$$\hat{H} = H\, e^{i\omega t} \tag{1.11}$$

where $i^2 = -1$

Therefore, its derivative with time is :

$$\frac{\partial \hat{H}}{\partial t} = i\omega H\, e^{i\omega t} \tag{1.12}$$

then $V = i\omega\mu H_o M_R h$

$= V_o h$

where V : complex amplitude of the electromotive force in the receiver coil

$V_o = i\omega\mu H_o M_R$: primary electromotive force, i.e. the electromotive force caused by the primary magnetic field

M_R : moment of receiver coil

Therefore, for nature and the model, we have

$$V_n = V_{on}\, h_n \tag{1.13}$$

and

$$V_m = V_{om}\, h_m \tag{1.14}$$

Since functions h and h are the same as those in equations (1.3) and (1.4), we have the following relation for the electromotive forces for nature and the model :

$$\frac{V_n}{V_{on}} = \frac{V_m}{V_{om}} \tag{1.15}$$

if condition (1.9) holds. Because the primary electromotive force V_{on} can be calculated and V_{om} can be measured or calculated, and V_m is measured by the modeling, we can obtain the electromotive force V_n . Therefore the electromotive force in nature can be derived from physical modeling.

From this consideration, we have shown that the same condition (1.9) is applied to measure an electromotive force as to measure the magnetic field.

1.2.2 Time domain

For the time domain, the amplitude of any component of the magnetic field, $\bar{H}$, can be written as follows :

$$\bar{H} = H_0 h\left(\frac{\tau_1}{r_1}, \alpha, \beta, \gamma \right) \tag{1.16}$$

where τ_1 : the parameter which defines the behavior of transient response and expressed as :

$$\tau_1 = 2\pi\sqrt{\frac{2\rho_1 t}{\mu_1}} \tag{1.17}$$

Then, the functions h_n and h_m are the same if the following conditions hold :

$$\alpha_n = \alpha_m \tag{1.18}$$

$$\beta_n = \beta_m \tag{1.19}$$

$$\gamma_n = \gamma_m \tag{1.20}$$

$$\frac{\tau_{1n}}{r_{1n}} = \frac{\tau_{1m}}{r_{1m}} \tag{1.21}$$

By substituting equation (1.17), this equation can be rewritten as follows :

$$\frac{\sigma_{1n}\mu_{1n}r_{1n}^2}{t_n} = \frac{\sigma_{1m}\mu_{1m}r_{1m}^2}{t_m} \tag{1.22}$$

Taking into account equations (1.18) through (1.22), the following relations must be valid for every medium :

$$\frac{\sigma_n \mu_n r_n^2}{t_n} = \frac{\sigma_m \mu_m r_m^2}{t_m} \tag{1.23}$$

Therefore, by creating a model which satisfies equation (1.23) for every medium and by using the same type of array and measuring the same component, the field normalized by the primary field can be the same for nature and the model :

$$\frac{H_n}{H_{on}} = \frac{H_m}{H_{om}} \tag{1.24}$$

Since H_{on} is calculated, and H_{om} can be also calculated or measured, and H_m is measured by physical modeling, the magnetic field in nature H_n can be obtained.

The relation of electromotive forces for nature and the model can be considered by taking the time derivative of equation (1.16) :

$$\frac{\partial H}{\partial t} = H_o \frac{\partial h}{\partial x} \frac{\partial x}{\partial \tau} \frac{\partial \tau}{\partial t} \tag{1.25}$$

where $x = \frac{\tau}{r}$

Inasmuch as $\frac{\partial x}{\partial \tau} = \frac{1}{r}$ and $\frac{\partial \tau}{\partial t} = \frac{\tau}{2t}$, we have :

$$\frac{\partial H}{\partial t} = H_o \frac{\partial h}{\partial x} \frac{\tau}{2tr}$$

$$= H_o \frac{\partial h}{\partial x} \frac{x}{2t}$$

Then, the electromotive force for nature and the model can be written as follows :

$$V_n = H_{on} \mu_n M_{Rn} \frac{\partial h_n}{\partial x_n} \frac{x_n}{2t_n} \qquad (1.26)$$

$$V_m = H_{om} \mu_m M_{Rm} \frac{\partial h_m}{\partial x_m} \frac{x_m}{2t_m} \qquad (1.27)$$

If relation (1.23) is valid, two functions h_n' and h_m' are the same and $x_n = x_m$. Therefore the electromotive force for nature can be written as :

$$V_n = V_m \frac{H_{on} \mu_n M_{R_n} t_m}{H_{om} \mu_m M_{R_m} t_n} \qquad (1.28)$$

Since we know the right hand side of this equation, we can obtain the electromotive force in nature, V_n .

These results can be illustrated by considering the following example. Suppose a vertical magnetic dipole is located on the surface of a non-magnetic uniform half space. The vertical magnetic component H_Z is expressed as :

$$H_Z = - \frac{M_T}{4\pi r^3} h_Z$$
$$= H_o h_Z$$

where $H = - \dfrac{M_T}{4\pi r^3}$

r : distance from the transmitter

M_T : transmitter moment

and h_Z is a dimensionless function :

$$h_Z = 1 - (1 - \frac{9}{u^2}) \phi(u) - \sqrt{\frac{2}{\pi}} e^{-\frac{u^2}{2}} (\frac{9}{u} + 2u) \quad (1.29)$$

where $u = \dfrac{2\pi r}{\tau}$

and $\phi(u)$ is the error function expressed as :

$$\phi(u) = \sqrt{\frac{2}{\pi}} \int_0^{\infty} e^{-\frac{t^2}{2}} dt$$

The time derivative of H_Z can be written as :

$$\frac{\partial H}{\partial t} = - \frac{9 M_T \rho}{2\pi r^5} h_Z' \quad (1.30)$$

$$= \frac{18\rho}{r^2} H_o h_Z'$$

where $$h' = \phi(u) - \sqrt{\frac{2}{\pi}}\, e^{-\frac{u^2}{2}} u\left(1 + \frac{u^2}{3} + \frac{u^4}{9}\right) \qquad (1.31)$$

So the electromotive force for nature and model can be written as :

$$V_n = \frac{18\rho}{r_n^2} H_{on}\, h_{Z_n}' M_{Rn} \qquad (1.32)$$

$$V_m = \frac{18\rho}{r_m^2} H_{om}\, h_{Z_m}' M_{Rm} \qquad (1.33)$$

Then by substituting equation (1.21), the electromotive force in nature can be written as follows :

$$V_n = V_m \frac{H_{on}}{H_{om}} \frac{M_{Rn}}{M_{Rm}} \frac{t_m}{t_n} \qquad (1.34)$$

This equation is the same as equation (1.28).

1.2.3 Geometric and time scale factor of modeling

From the results of the previous section, a physical model must be constructed so that every corresponding medium or structures for nature and the model satisfy equations (1.9) or (1.23) for frequency and time domain respectively.

Equation (1.9) can be rewritten as :

$$\frac{\sigma_n \mu_n}{\sigma_m \mu_m} = a \cdot b^2 \qquad (1.35)$$

where $a = \frac{\omega_m}{\omega_n}$ is the frequency scale factor, and $b = \frac{r_m}{r_n}$

is the geometric scale factor.

From equation (1.23), we have :

$$\frac{\sigma_n \mu_n}{\sigma_m \mu_m} = \frac{b^2}{c} \tag{1.36}$$

where $c = \frac{t_m}{t_n}$ is the time scale factor of modeling. The conductivities of an earth structure σ_n vary from highly conductive ores with about 10^3 mho/m to resistive rocks with about 10^{-4} mho/m. Therefore, modeling materials with a wide range of conductivity σ_m are needed.

1.3 Approaches to physical modeling

1.3.1 Modeling materials

Physical models are constructed using various materials such as metals, graphite, electrolytic and artificial materials (composites of conducting and insulating materials). Table 1.1 shows the conductivities of materials used for modeling, and the conductivity spectrum is shown in figure 1.1.

First of all, physical modeling can be divided into several approaches which differ according to the materials used to construct the model, such as :

Material	Conductivity (mho/m)
Silver	6.17 x 10
Copper	5.81
Aluminum	3.84
Zinc	1.69
Brass	1.4-2.0
Steel	0.5-1.0
Bronze	0.5-0.8
Lead	0.48
Mercury	0.10
Graphite	0.7-1.2 x 10
Epoxy, copper loaded	0.25
Wax, bronze loaded	0.1-0.13
Resin, graphite loaded	0.01
Hydrochloric acid, 10% W	62.9
Sulfuric acid, 10% W	39.2
Sodium chloride, 10% W	12.0
Copper sulfate, 10% W	3.2
Sodium chloride, 0.3% W	0.054

Table 1.1 Conductivities of modeling materials

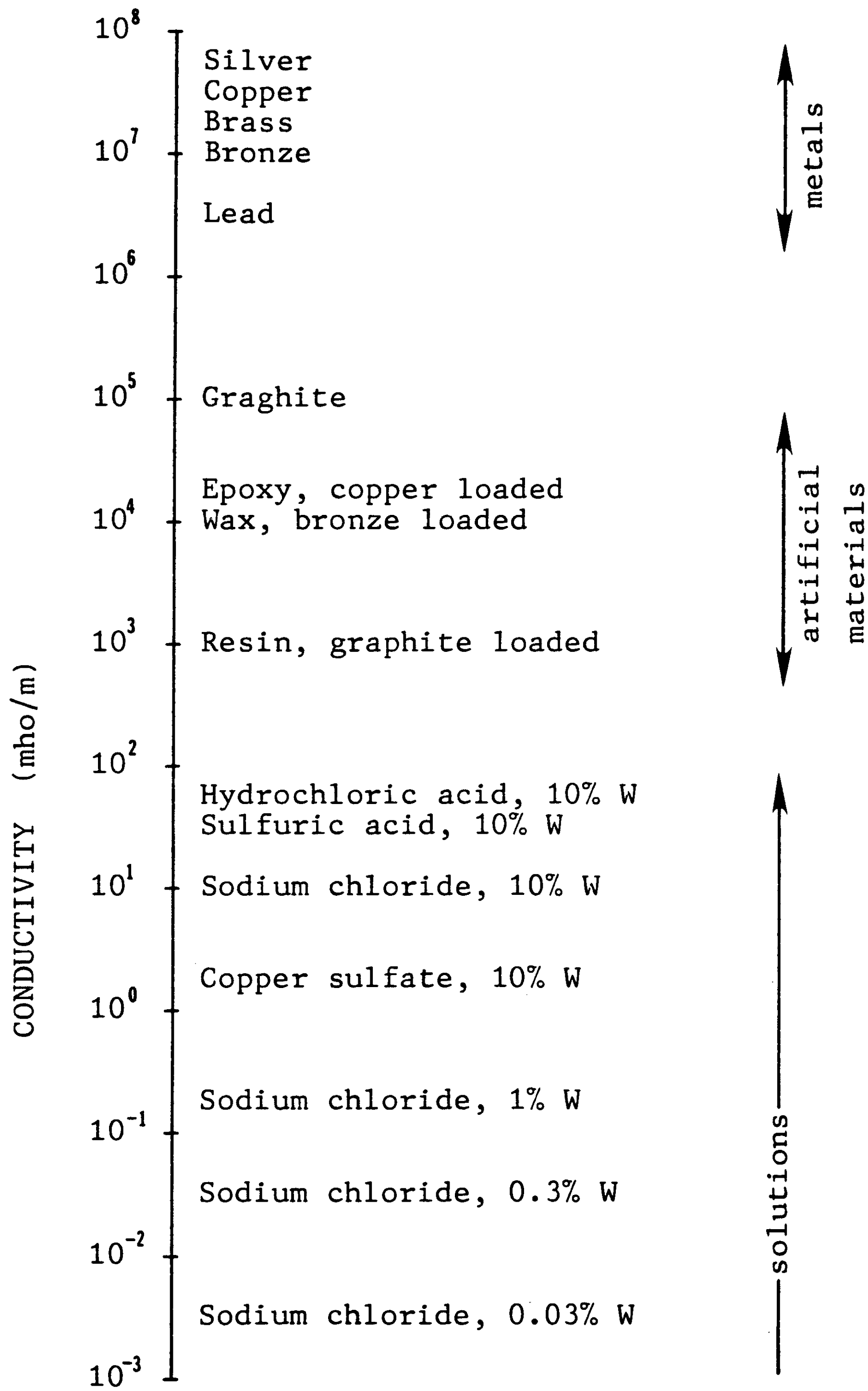


Figure 1.1 Spectrum of conductivities of modeling materials.

(1) modeling with metals

(2) modeling with electrolytic

(3) modeling with artificial materials

These are explained in the following sections.

1.3.2 Modeling with metals

Metals have been used for a long time to make models. They have the following features :

(1) high conductivity

(2) small range of conductivity

The conductivities of commonly available metals differ only one order, from 5.81 10 of copper to 4.76 10 of lead, as shown in figure 1.1. To make models, the ratio of conductivities must be kept constant for nature and the model. But due to the small range of conductivities, metal can not be used to make a model which has a wide range of conductivity contrast.

In some special cases, where the influence of the host rock is neglegible, a simplified model using metal can be used, that is :

(1) Host rock is so much more resistive than the ore body and the overburden that the current in the host rock is negligible compared to the current in the ore body and overburden.

(2) Thickness of overburden is much smaller than the skin depth δ in the frequency domain, or parameter τ in the time domain.

(3) Transmitter receiver separation is relatively large in the frequency domain.

If any of these conditions is valid, the overburden can be replaced by a conducting thin plane, and the effect of the host rock can be neglected. Therefore, very simplified modeling can be performed.

If the model has axial symmetry as in the case of induction logging, an integrator method has been developed, which uses a set of metal rings to represent the medium (Kaufman 1959).

The modeling by metals can be illustrated by considering the following conducting dike model shown in figure 1.2(a):

(1) Currents in host medium can be neglected.

(2) We use a geometrical scale factor b=1/1000 (figure 1.2(b)).

(3) We use aluminum to make the model ($\sigma_m = 3.84 \cdot 10^7$ mho/m).

(4) From equation (1.35), the frequency scale factor a is obtained as follows if we assume $\mu_m = \mu_n$:

$$a = \frac{\omega_m}{\omega_n} = \frac{\sigma_n}{\sigma_m} b^2 = 0.26$$

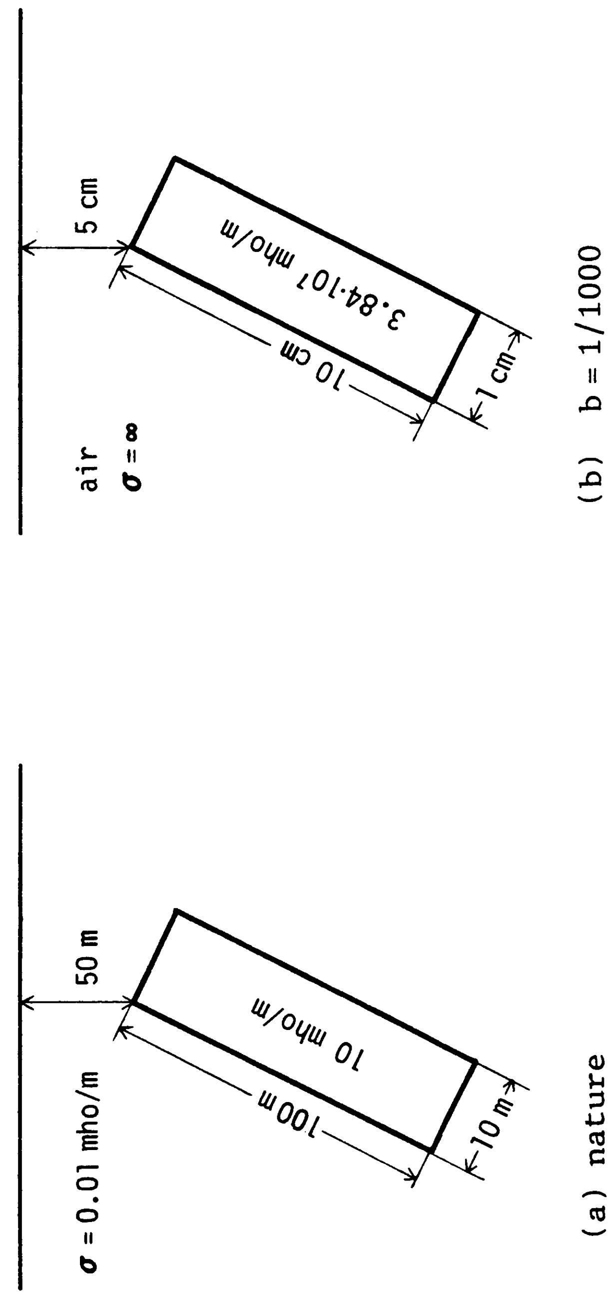


Figure 1.2 Modeling with metals

(5) From equation (1.36), the time scale factor c is obtained as follows if we assume $\mu_n = \mu_m$:

$$c = \frac{t_m}{t_n} = \frac{\sigma_n}{\sigma_m} b^2 = 3.84$$

Therefore, the frequency or time range for the modeling differs by a factor of nearly 4 from that used in nature. Usually, if metals are used for modeling, almost the same frequency or time ranges as those used in nature can be used.

Figure 1.3 shows an example of the frequency domain response for a confined ore body. The abscissa of the curves is one of the geometrical dimensions normalized by the skin depth, r/δ. From equation (1.1), if the parameters α, β and γ are fixed for the model, the nondimensional function h is defined only by r/δ. Therefore, the modeling results can be applied to various dike structures which have the same parameters α, β and γ.

Figure 1.4 shows an example of the time domain response of a confined ore body, plotted as a function of τ/a. Similar to the frequency domain, modeling results are applied to various dike structures which have same α, β and γ.

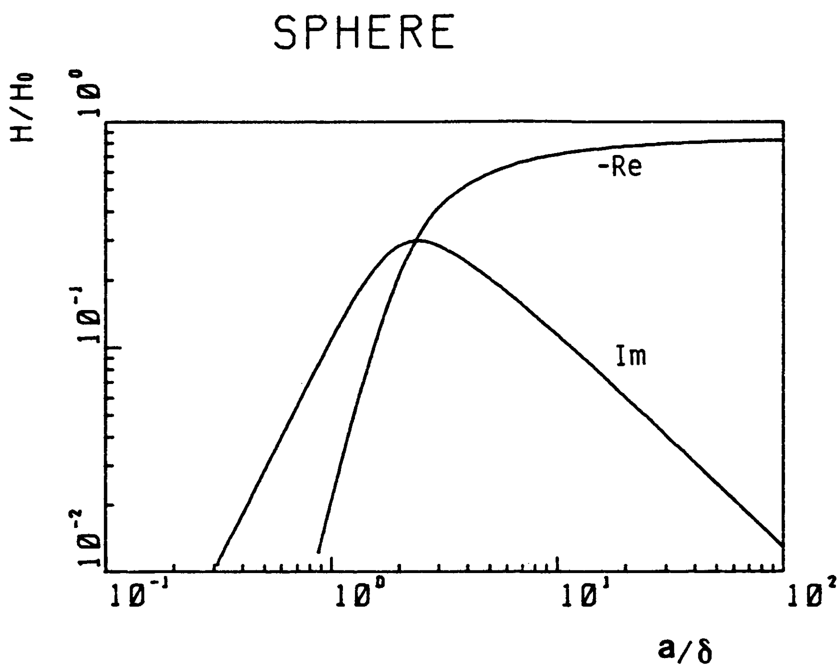


Figure 1.3 Frequency response of confined ore body

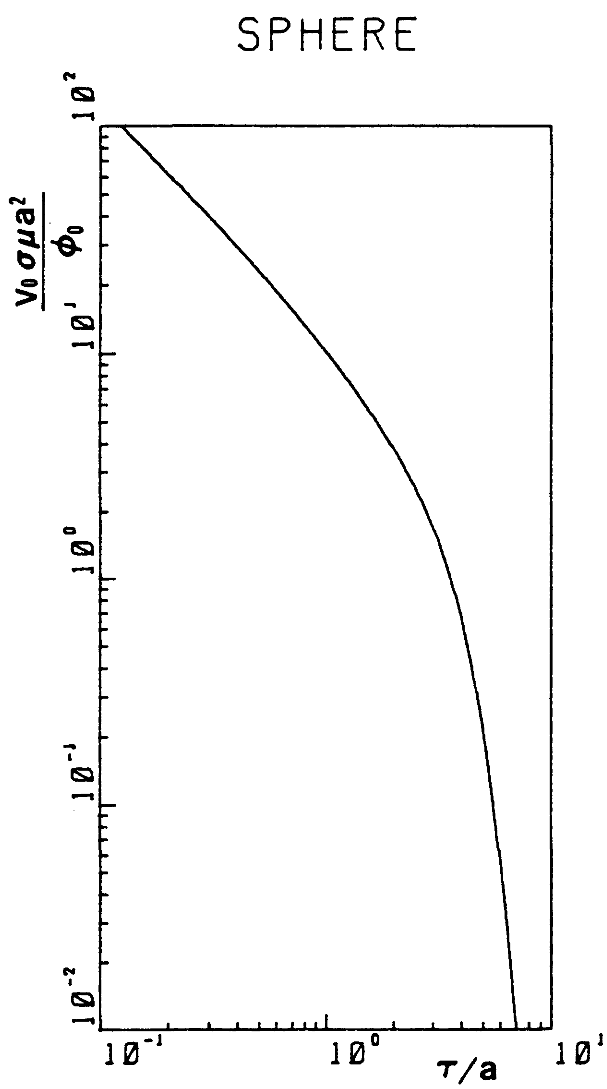


Figure 1.4 Time domain response of confined ore body

1.3.3 Modeling with electrolytic

Electrolytic solutions are also used for modeling. Unlike metals, they have a very wide range of conductivities but the maximum conductivity is only about 20 mho/m for a saturated sodium chloride solution (20°C). Due to this relatively small conductivity, large-size models and high frequencies (small times) are used.

Since there is more than a 5 order of magnitude difference of conductivity between metals and solutions, or more than a 3 orders difference between graphite and solutions, models made with metals or graphite and solutions can not be used in general.

It is not easy to separate the solutions of different conductivities while maintaining electrical contact. For this purpose, a non-disturbing boundary is sometimes used, which is made of an insulating plate containing many holes filled with conducting materials. Figure 1.5 shows an example of modeling with solutions using non-disturbing boundary.

Because of the wide range of the conductivity, modeling by solutions is often preferable to other methods. An example of modeling using electrolytic solutions is shown in section 1.3.5.

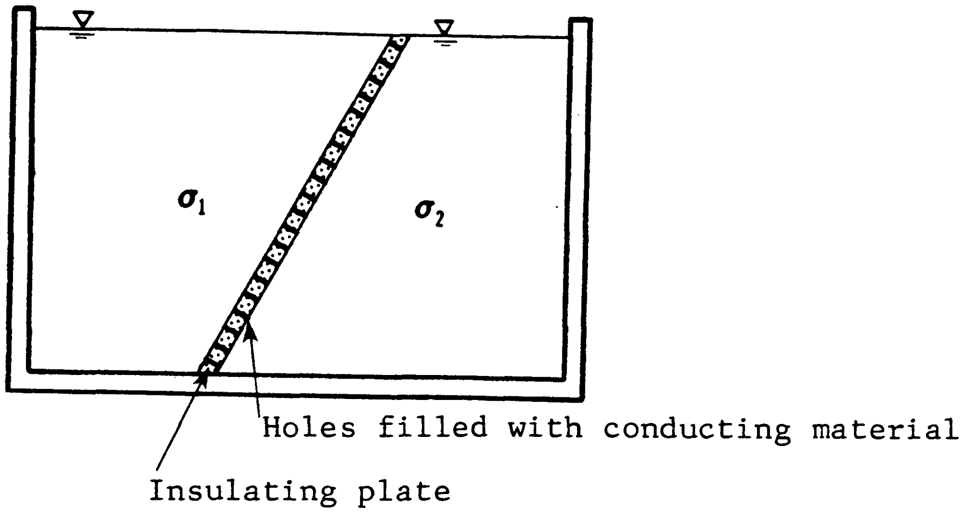


Figure 1.5 Modeling using non-disturbing boundary

1.3.4 Modeling with artificial materials

Since there is a large difference of conductivities between metal or graphite and solutions, there have been many attempts to introduce artificial materials to fill the gap.

By loading a suitable binder with a sufficient number of conducting particles, materials having this range of conductivity can be obtained. Pritchett(1955) used bronze particles in a mixture of wax and obtained a material having a conductivity of about $1.25 \cdot 10^4$ mho/m. Frischknecht(1971) used polyester resin mixed with flaked or powdered graphite to produce materials ranging in conductivity from about 10 to 10 mho/m.

But there are serious disadvantages such as :

(1) Conductivities are not stable with time.

(2) It is difficult to make such materials electrically uniform.

(3) Materials are not mechanically strong.

Therefore, modeling with artificial materials is not widely used.

1.3.5 Example of modeling using a solution as a host rock and artificial material as an ore body

To illustrate modeling using electrolytic solutions, let us consider the following example. Assume we have a conductive dike of conductivity 1 mho/m, surrounded by a medium of conductivity 0.01 mho/m, and geometrical parameters are shown in figure 1.6(a). If we make a model in which geometrical scale factor is 1/1000, and the time scale factor is unity (t_m=t_n), from equation (1.23), we have :

$$\sigma_m = \sigma_n \frac{r_n^2}{r_m^2} = \sigma_n \cdot 10^6 \qquad (1.39)$$

Therefore, we need a dike conductivity of 10 mho/m, width 1 cm, and a vertical length of 10 cm as shown in figure 1.6(b). The conductivity of the surrounding medium must be 10 mho/m. A solution of that conductivity can not be obtained.

If the scale factor is changed to 1/20, the conductivities of the dike and the surrounding medium became $4 \cdot 10^2$, and 4 mho/m respectively. These materials can be obtained but the size of the dike becomes too large (Figure 1.6(C)).

If we keep the geometrical scale factor b = 1/1000, and change the time scale factor to $c = t_m/t_n = 1/1000$, from equation (1.36), we have :

$$\sigma_m = \sigma_n \frac{c}{b^2} = 10^3 \cdot \sigma_n$$

Therefore, conductivities of the dyke and the surrounding medium become 10 and 10^3 mho/m respectively. Although the size of the model is the same as that in Figure 1.6(b), the time scale factor c is so small that we have to measure at very short times ($t_m = 10^{-3} t_n$). In nature, the measurement is done in time units of milliseconds, but for this modeling, we must deal with microseconds, and sophysticated equipments must be used.

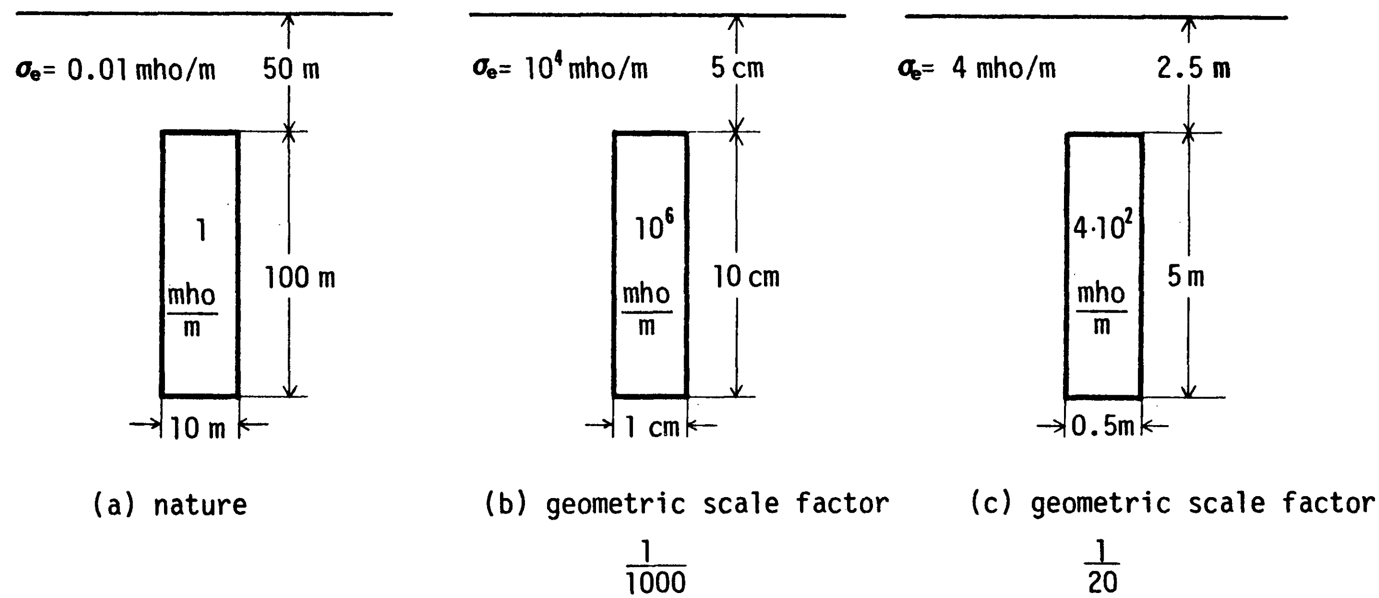


Figure 1.6 Modeling of vertical conducting dike

CHAPTER 2

PHYSICAL MODELING INSTRUMENTS

2.1 Introduction

Development of instruments play a major role in physical modeling experiments. Figure 2.1 shows a block diagram of the transient electromagnetic modeling system developed for this experiments.

The transmitter system consists of a power source, inverter and loop. The current in the transmitter loop supplied by the power source (battery) is turned on, turned off and reversed in direction by the inverter, which is controlled by the main processing unit to synchronize the turnoff time with that at the receiver system.

The receiver system is consist of a coil, amplifier, analog-to-digital converter and main processing unit. The main processing unit consists of CPU (Motorolla M6809 microprocessor), memory (RAM 58K, ROM 2K), programable clock, serial and parallel I/O ports and peripheral units (CRT terminal, graphic printer, digital X-Y plotter, modem and dual mini-floppydisk drive).

Data aquisition and interpretation software was written mainly using Pascal and assembly languages. Details of each

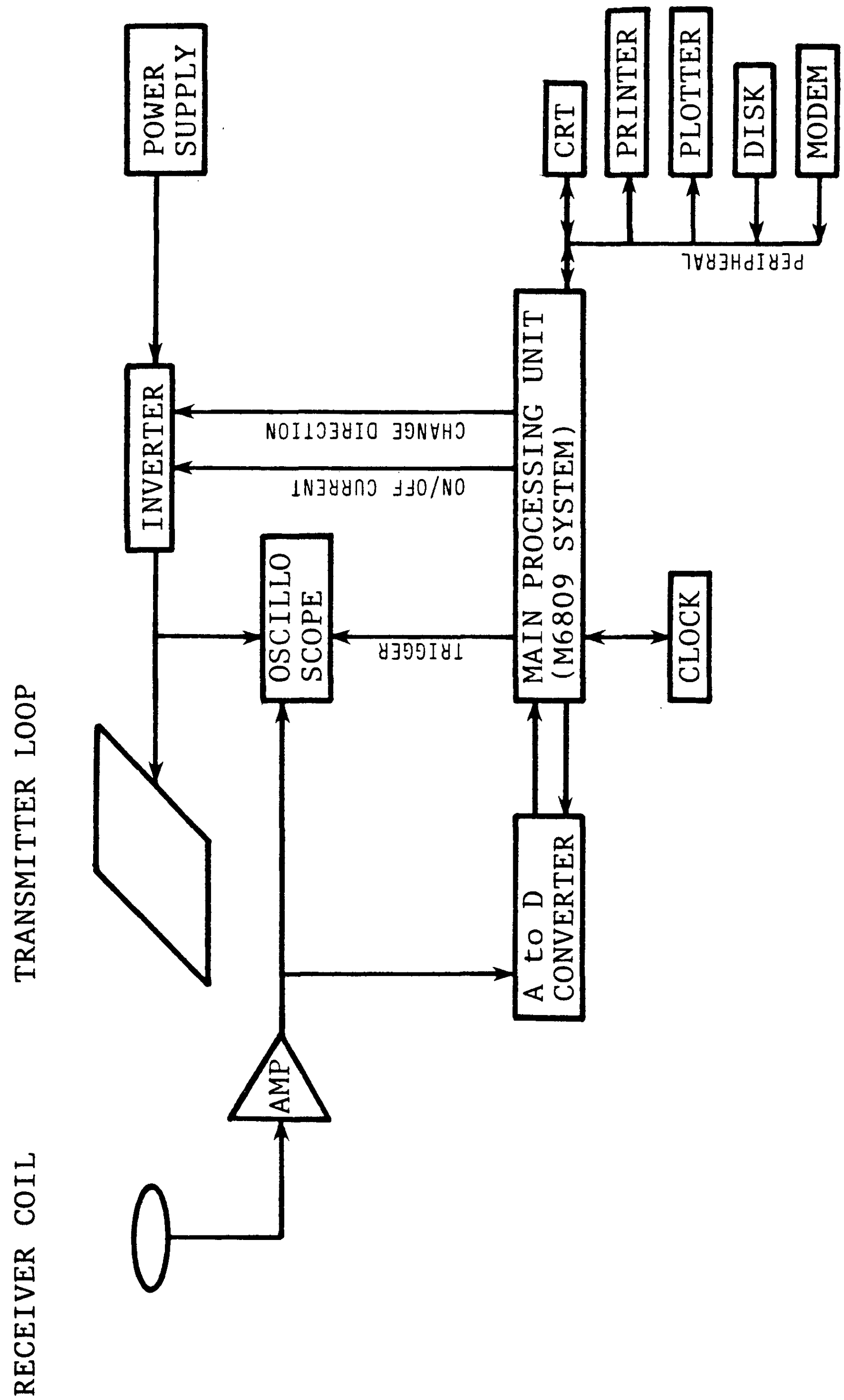


Figure 2.1 Block diagram of transient physical modeling system.

instrument and software are explained in the following sections.

2.2 Transmitter system

2.2.1 Power source

The power source supplies the current in the transmitter loop. To create more than two amperes, an 80AH battery was used, and for a current of less than two amperes, a constant current power supply was used.

2.2.2 Inverter

The current in the transmitter loop is turned on, off and reversed in direction by the inverter. To turn off the currents in the transmitter quickly is very difficult due to the overvoltage caused by the inductance of the loop. Since mechanical switches such as magnetic relays are not adequate due to the slow and unstable response to a control signal, an inverter circuit using a semiconductor switch was designed for the experiment, and is shown in Figure 2.2.

Signal lines from the main processing unit are isolated by an optical coupler and a mechanical relay to protect the processing unit from the overvoltage of the transmitter loop. If the control signal is high, the transistor TR1

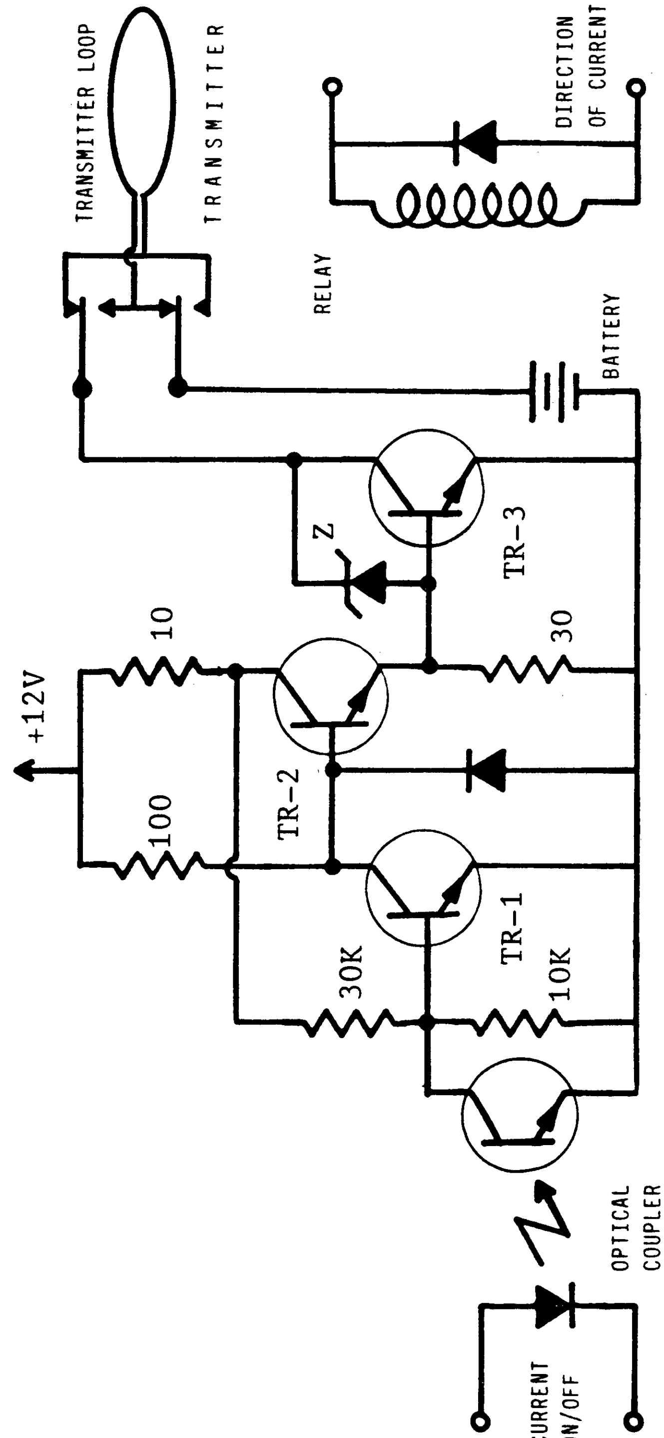


Figure 2.2 Inverter Circuit

should be off due to the collector currents of the photo-transistor. Therefore the transistor TR2 is on due to the base currents through the 100 ohm resistor, and then this emitter currents keep the TR3 on. If the control signal is removed, the reversed process turns the TR3 off. But due to the inductance of the load (transmitter loop), the over-voltage appears between the collector and emitter of the switching transistor TR3.

The overvoltage V due to the inductance of the transmitter loop can be calculated as :

$$V = -L \frac{dI}{dt} \qquad (2.1)$$

where L is the inductance of the loop. The inductance of the loop consists of two part, internal inductance L_i due to the wire, and external inductance L_e due to the external part of the wire, and expressed as :

$$L = L_i + L_e \qquad (2.2)$$

In the case of a circuler loop, the following equations can be obtained to calculate L_i and L_e (Johnk, 1975) :

$$L_i = \frac{\mu l}{8\pi} \qquad (2.3)$$

and

$$L_e = b(\ln\frac{8b}{a} - 2)n^2 \quad a \ll b \qquad (2.4)$$

where μ : magnetic permeability

a : radius of the wire

b : radius of the loop

l : length of the wire

By assuming that the external inductance L of rectangular loop is the same as that of the circular loop of the same area, the inductance of the transmitter loop TLR1 (0.2 x 0.4 m, 40 turn, AWG #22) can be calculated to be about 2 milli-Henry. By assuming that the switching speed (dI/dt) of the transistor is 10 A/microsecond, the overvoltage of the transmitter loop TRL1 can reach approximately 20 kilovolts.

The switching transistor TR3 can be easily destroyed due to this high voltage unless some protection circuit is provided. The zener diode Z between the collector and the base of the transistor TR3 has the role of reducing the switching speed of TR3. When the voltage between the collector and the emitter is bigger than the zener voltage of Z, the base currents through this zener diode keep the TR3 on. The switching speed (ramp time) can be controlled by changing the zener voltage of Z. For example, for this physical modeling, a 300 V zener diode was used and the ramp time of the inverter was about 35 microseconds.

The direction of the currents in the transmitter loop can be reversed by a mechanical relay which is controlled by MPU only while transmitter currents are absent.

Figure 2.3 shows the control signals from the MPU (to turn on, off and change the direction of current) and the waveform of the transmitter currents.

2.2.3 Transmitter loop

The transmitter loop creates the primary magnetic field, and Table 2.1 shows the physical dimensions of the loops developed for the physical modeling. The name of the transmitter loop consists of three characters and one number, the first two characters TL mean transmitter loop and the third character shows the shape of the loop (R:rectangular, S:square, C:circular coincident, D:dipole). For the loops TLR1, TLR2 and TLS1, the gage AWG#22 wire was used, and AWG#22 wire was used for loops TLC1 and TLD1.

	a (m)	b (m)	r (m)	n (turn)	SHAPE
TLR1	0.2	0.4		40	RECTANGULAR
TLR2	0.1	0.2		48	RECTANGULAR
TLS1	0.2	0.2		30	SQUARE
TLC1			0.1	7	COINCIDENT
TLD1			0.025	10	DIPOLE

Table 2.1 Transmitter loop

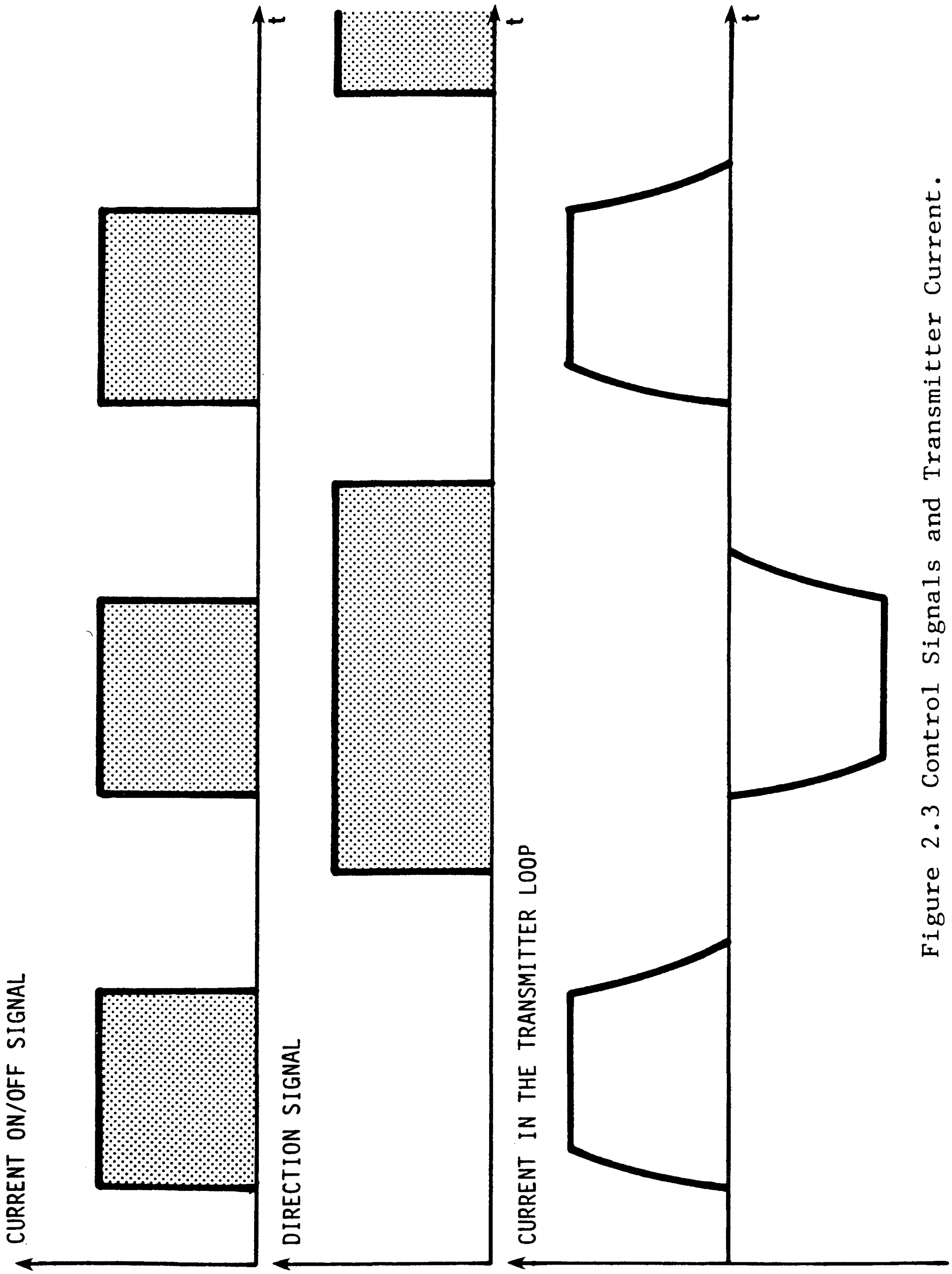


Figure 2.3 Control Signals and Transmitter Current.

2.3 Receiver system

2.3.1 Receiver coil

The receiver coil is used to measure the time-rate of change of the magnetic field, and following requirements should be met for physical modeling :

(1) Big moment

(2) Small dimension

(3) Well shielded (eliminate the effect of electric field)

(4) Small distributed capacitance

(5) Mechanically stable

The moment of the receiver coil is defined as follows :

$$M_R = n \times A$$

where Mr is the receiver moment, n is the number of turns and A is the area of the coil. But due to the small size of the coil, the area and therefore the moments can not be determined accurately. Therefore, the value of the receiver moment was measured by the experiment, for which a cshematic is shown in Figure 2.4.

The moment, then, can be calculated by the following equation :

$$M_R = \frac{V}{\omega \mu M_T H_Z} \tag{2.5}$$

where M_T : moment of the transmitter loop

M_R : moment of the receiver coil

ω : radial frequency of the transmitter currents

μ : magnetic permeability of the free space

V : measured voltage

H_Z : magnetic field at the center of the loop

If the transmitter loop is circular, the magnetic field at the center of the loop is given by:

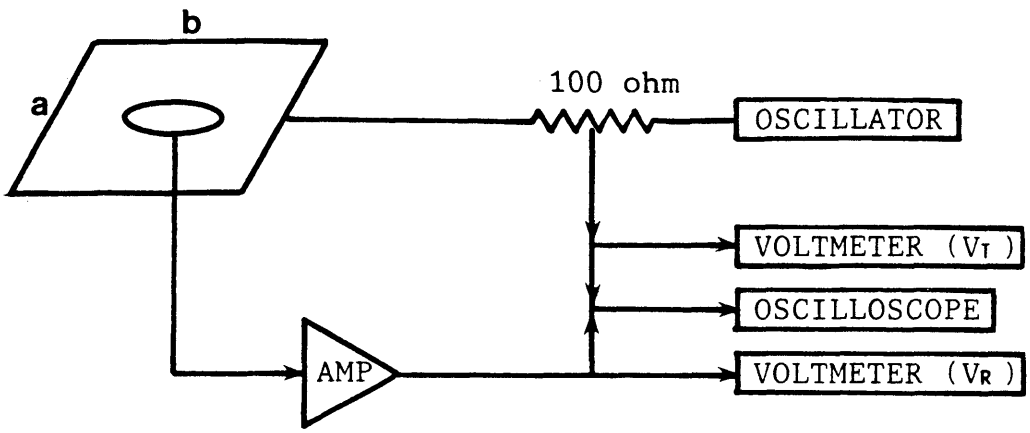
$$H_Z = \frac{I}{2r} \qquad (2.6)$$

where r is the radius of the loop, and I is the transmitter currents. If the loop is rectangular, the magnetic field at the center of the loop is calculated from Biot-Savart law, and we have :

$$H = 2I\frac{\sqrt{a^2 + b^2}}{\pi ab} \qquad (2.7)$$

where a and b are the sizes of the transmitter loop.

Figure 2.5 shows the moments of the receiver coils measured as a function of frequency. The moments of the receiver coils are almost constant up to 10 KHz except the coil RL1, which has the largest moment. Due to distributed



$$I = \frac{V_T}{100} \quad (A)$$

$$M_T = abIn$$

$$M_R = \frac{V_R}{\omega \mu M_T H_o}$$

$$H_o = 2I\frac{\sqrt{a^2 + b^2}}{\pi a b} \quad \text{(Rectangular loop)}$$

$$H_o = \frac{I}{2r} \quad \text{(Circular loop)}$$

where r : radius of circular transmitter loop

n : number of turns of transmitter loop

Figure 2.4 Measurement of receiver moment

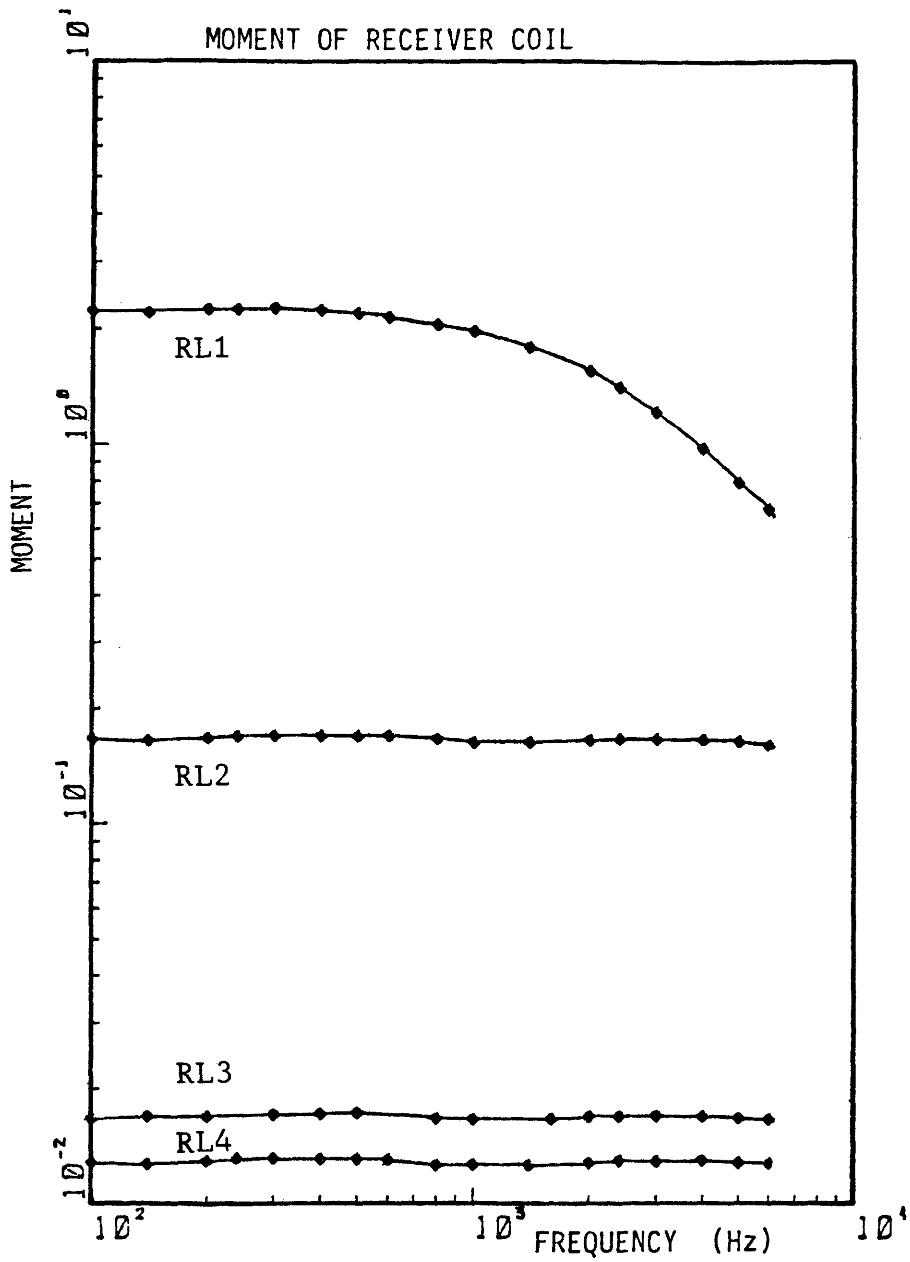


Figure 2.5 Moment of Receiver Coil

capacitance in the RL1 coil, the moment decreases when the frequency is more than 500 Hz. Table 2.2 shows the parameters of the receiver coil.

	MOMENT (m)	r (cm)	h (cm)	n (turn)
RL1	$2.2\cdot10^{0}$	2.2	1.0	5000
RL2	$1.7\cdot10^{-1}$	2.0	0.3	500
RL3	$1.7\cdot10^{-2}$	0.5	1.0	800
RL4	$1.3\cdot10^{-2}$	0.4	0.8	800

Table 2.2 Parameters of the receiver coil

2.3.2 Amplifier

The electromotive foece in the receiver coil is amplified by the amplifier to the appropriate voltage to the analog-to-digital converter for the data acquisition. Due to the small transmitter and receiver moments, the signals are usually small and contain a relatively wide frequency spectrum, so very high gain and wideband amplifiers are required. The required specifications of the amplifier for the transient electromagnetic modeling are as follows :

(1) Bandwidth : DC to 10 KHz

(2) Maximum gain : 2^{20} (1048576 times)

(3) Input noise : less than 0.1 microvolts (p-p 0.1-10Hz)

(4) Offset drift : less than 0.2 microvolts/°C

Because of the conflicts of these specifications, it is almost impossible to satisfy all the conditions at the same time, but due to the remarkable progress in the integrated circuit technique, excellent low noise and low drift operational amplifier chips are available. By using such chips, the design of the amplifiers was very simple. Table 2.3 shows some examples of specifications of typical operational amplifier chips.

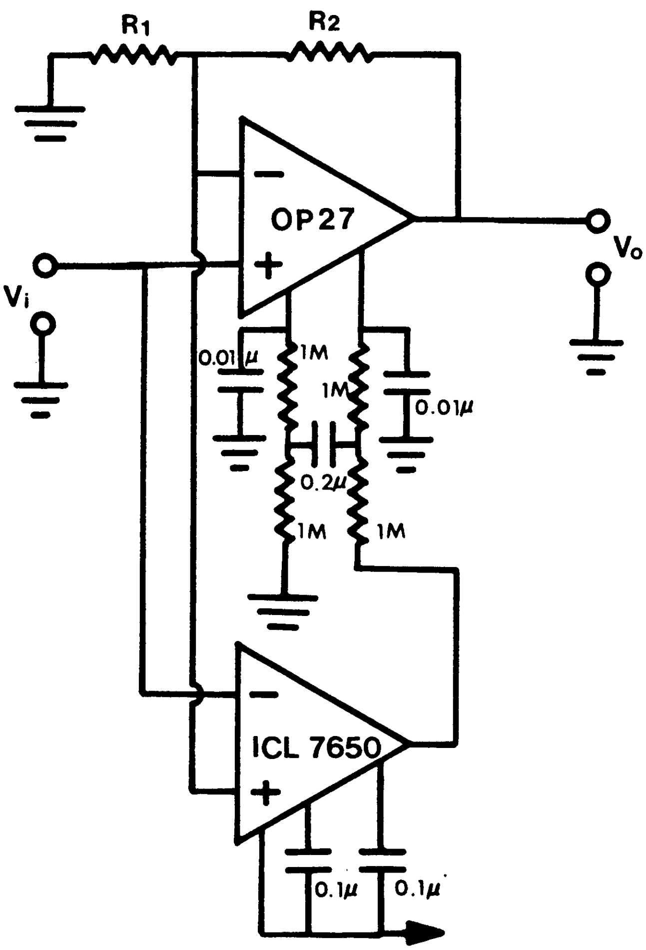
A three stage amplifier was designed using OP-27 operational amplifier chips. The first stage amplifier is very important because its noise and drift are amplified by the later stages. The OP-27 chip is one of the best low noise amplifiers available, but for the drift problem, chopper stabilized operational amplifiers like ICL7650 show better performance. Therefore for the first stage of the amplifier, the circuit which conbines the low noise performance of OP-27 and low drift of ICL7650 chopper amplifier was designed as shown in Figure 2.6. The output of the chopper amplifier is conected to the offset adjust terminal of the OP-27 to compensate the offset drift of the amplifier. The second and third stages of the amplifier are simple non-inverting amplifiers using OP-27 as shown in Figure 2.7. All the circuits are constructed using printed circuit boards for the stability and low noise.

OP-AMP	INITIAL OFFSET		DRIFT		INPUT NOISE 0.1-10 Hz		THROUGH RATE	REF
	(μV)		(μV/°C)		(μV pp)		(V/μsec)	
	typ	max	typ	max	typ	max		
OP27-E	10	25	0.2	0.6	0.08	0.18	2.8	(1) L
OP07-E	30	75	0.3	1.3	0.35	0.6	0.17	(2) L
101-BM	50	250	3	5	1.0	1.3	6.5	(1) G
741-L	200	500	2	5	N.A.		0.5	(2) G
234-L		20		0.1		1.5	30	(2) C
ICL 7650	0.7	5	0.01	0.05		2	2.5	(3) C
AD612-C		200		5		1	1	(2) I

REF (1) : Burr-Brown
(2) : Analog Devices
(3) : Intersil

L : low noise low drift
G : general purpose
C : chopper stabilized
I : instrumantation amplifier

Table 2.3 Specifications of OP-amplifiers



$$\frac{V_o}{V_i} = 1 + \frac{R_2}{R_1}$$

Figure 2.6 Preamplifier circuit

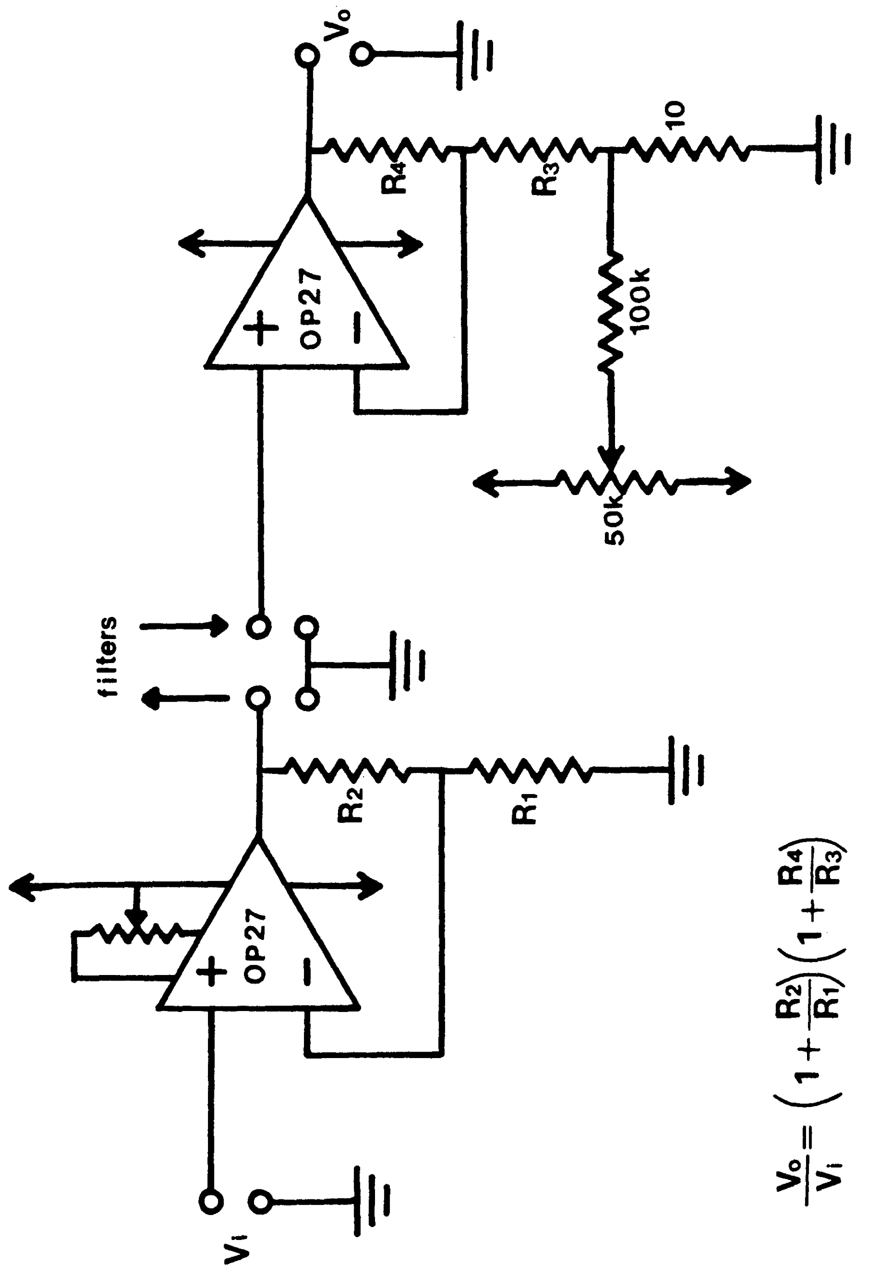


Figure 2.7 Amplifier Circuit

2.3.3 Filters

Sometimes, noise due to power lines (60 Hz) must be eliminated, especially for frequency domain experiments or field operations. A twin-T passoab svrnm uoirbh tmv½t wvvf band rejection performance but the Q-value is theoretically only 0.25. Figure 2.8 shows the measured amplitude response of a twin-T passive notch filter. To improve the Q-value, the active filter circuit shown in Figure 2.9 was used. The rejection frequency f_R and Q-value are obtained by the following equations :

$$f_R = \frac{1}{2\pi RC} \tag{2.8}$$

$$Q = \frac{1}{4(1-K)} \tag{2.9}$$

where K is the inverse of the gain of the amplifier.
If the Q-value is greater than 0.7, the step response of the notch filter causes oscillations. Figures 2.10 to 2.16 show the step response of the notch filter as then Q-value changes from 0.25 (twin-T) to 10. For the experiment, a notch filter with Q=0.7 was used.

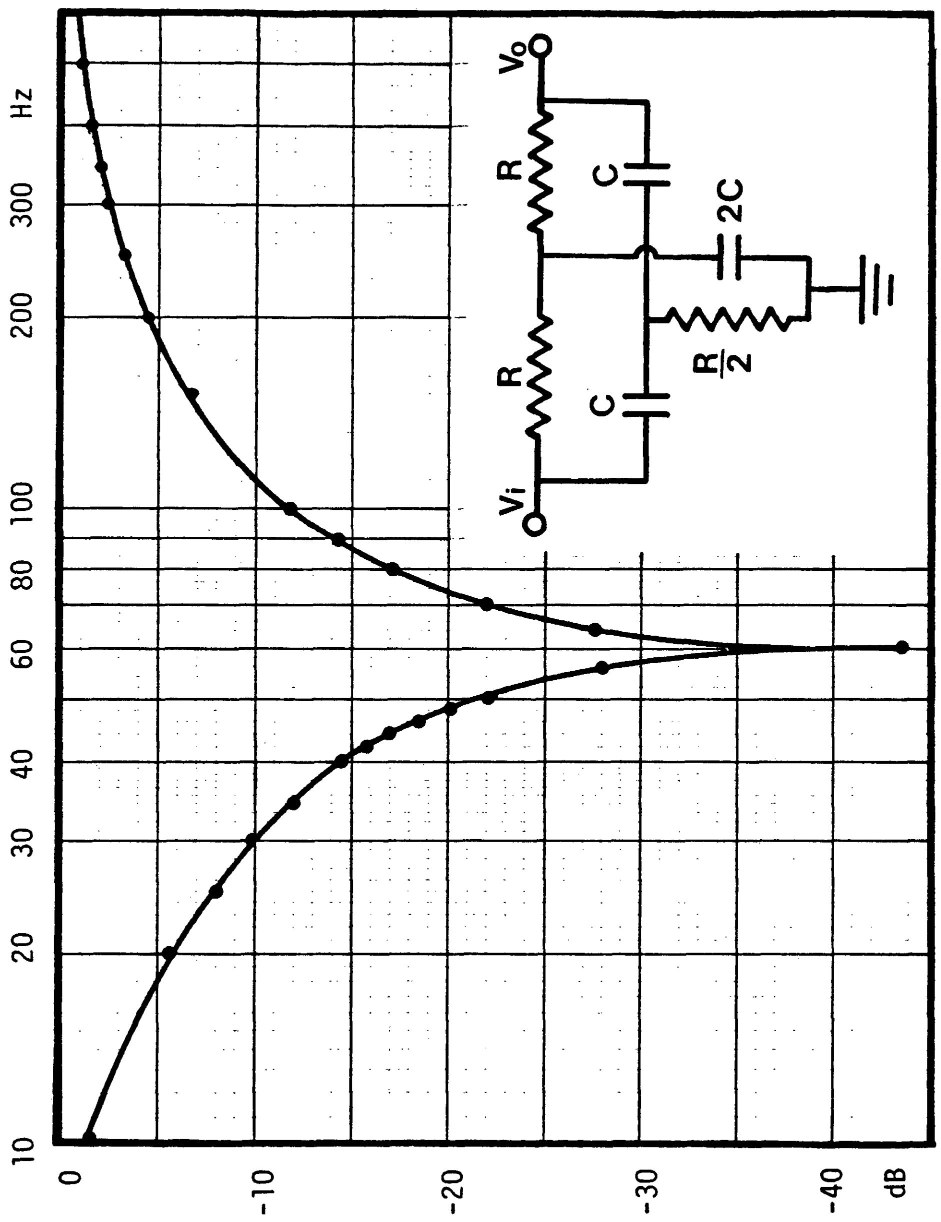
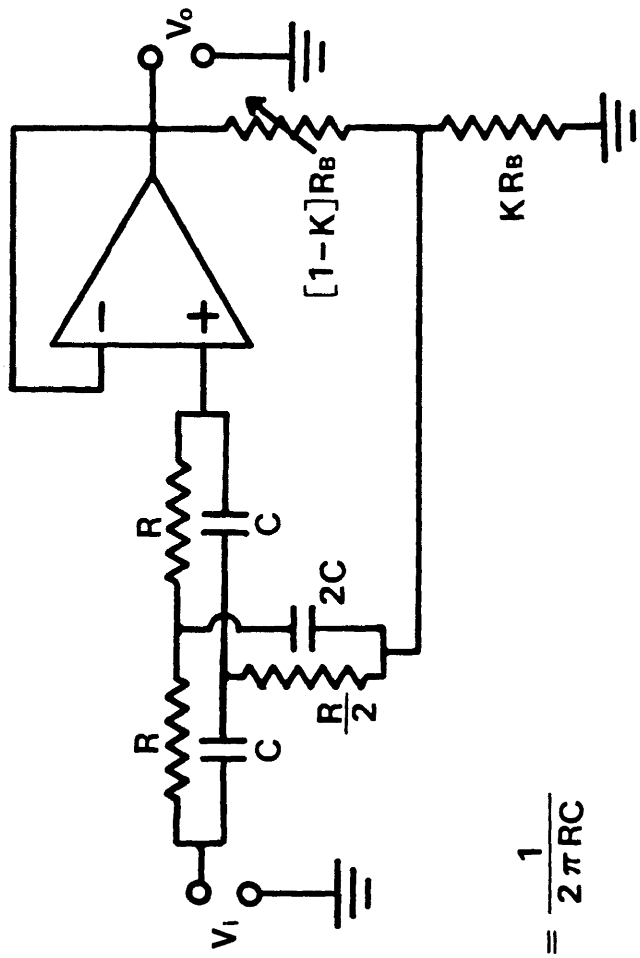


Figure 2.8 Amplitude Response of twin-T filter



$$f = \frac{1}{2\pi RC}$$

$$Q = \frac{1}{4(1-K)}$$

Figure 2.9 Active notch filter

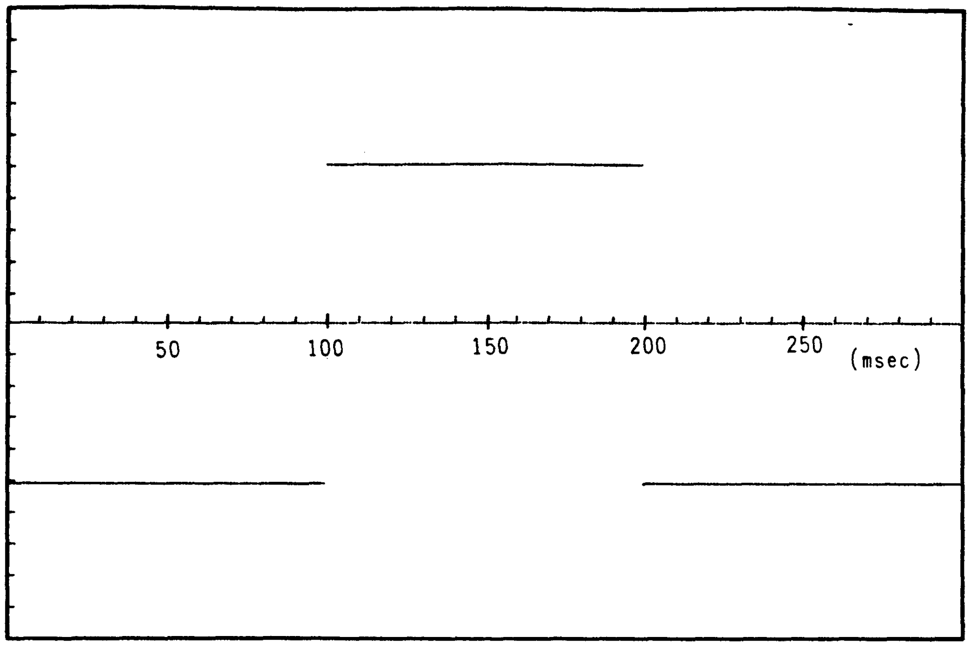


Figure 2.10 Input wave form (5Hz square wave)

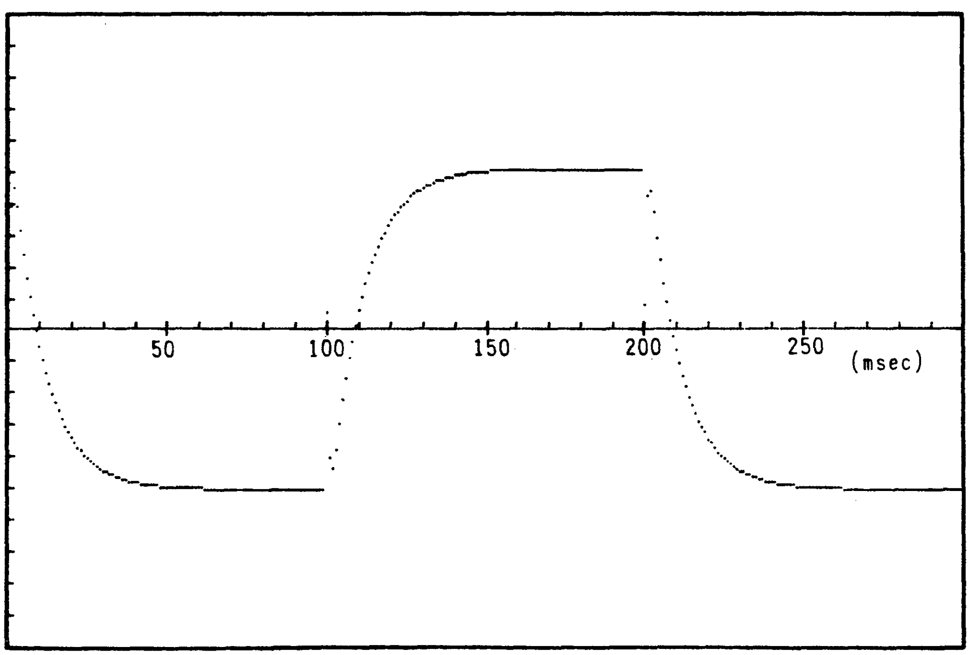


Figure 2.11 Notch filter (Q=0.25 twin-T)

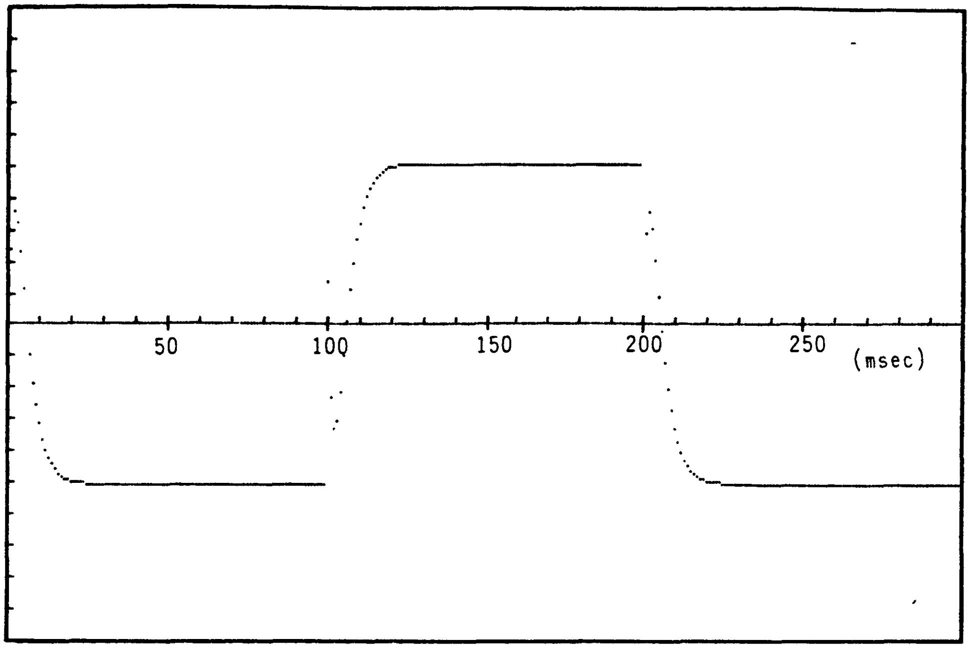


Figure 2.12 Notch filter (Q=0.5)

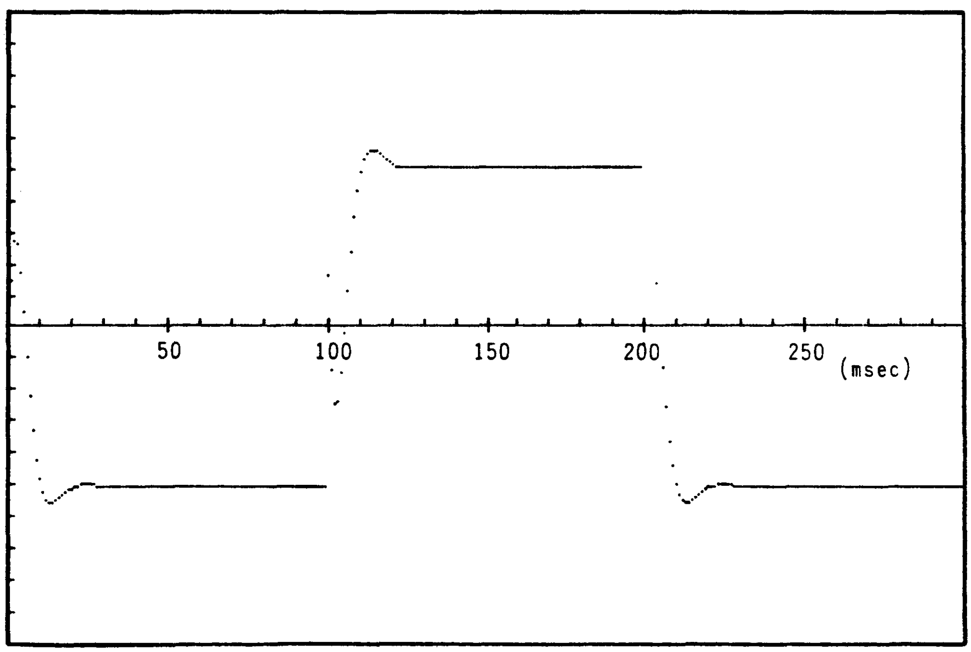


Figure 2.13 Notch filter (Q=0.75)

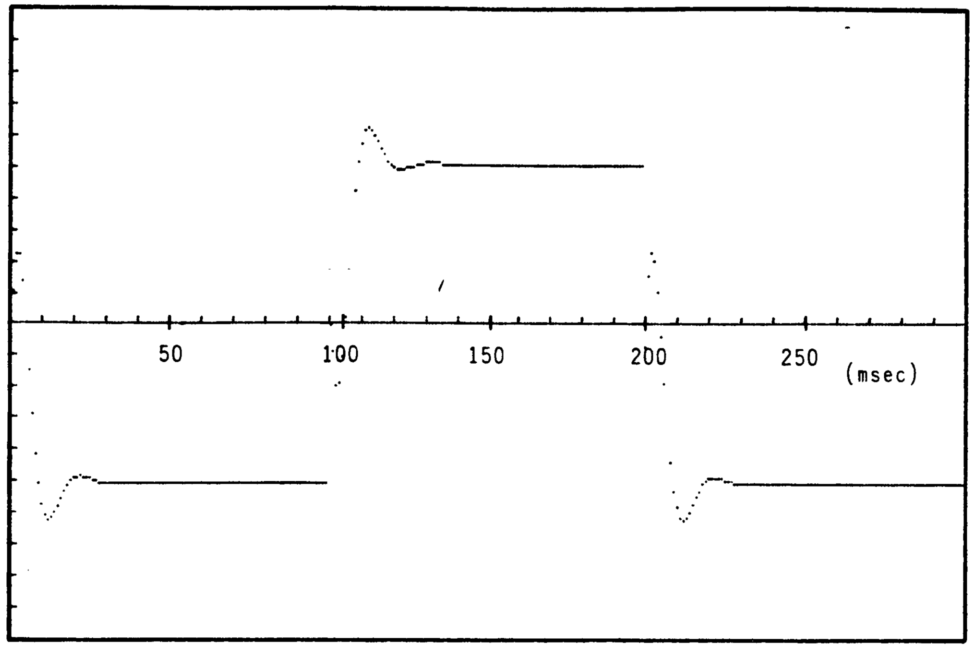


Figure 2.14 Notch filter (Q=1.0)

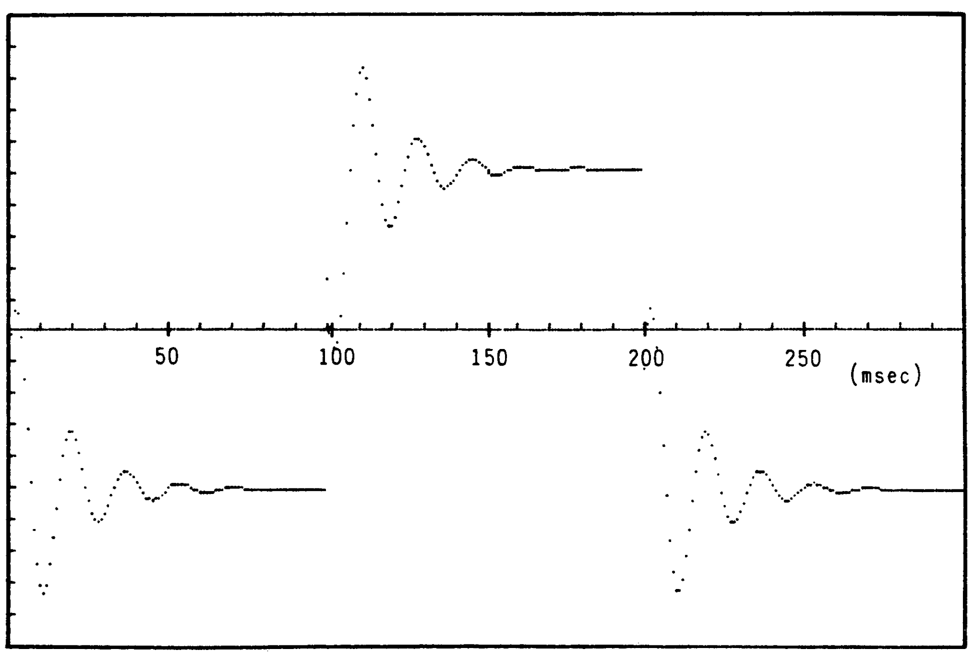


Figure 2.15 Notch filter (Q=2.8)

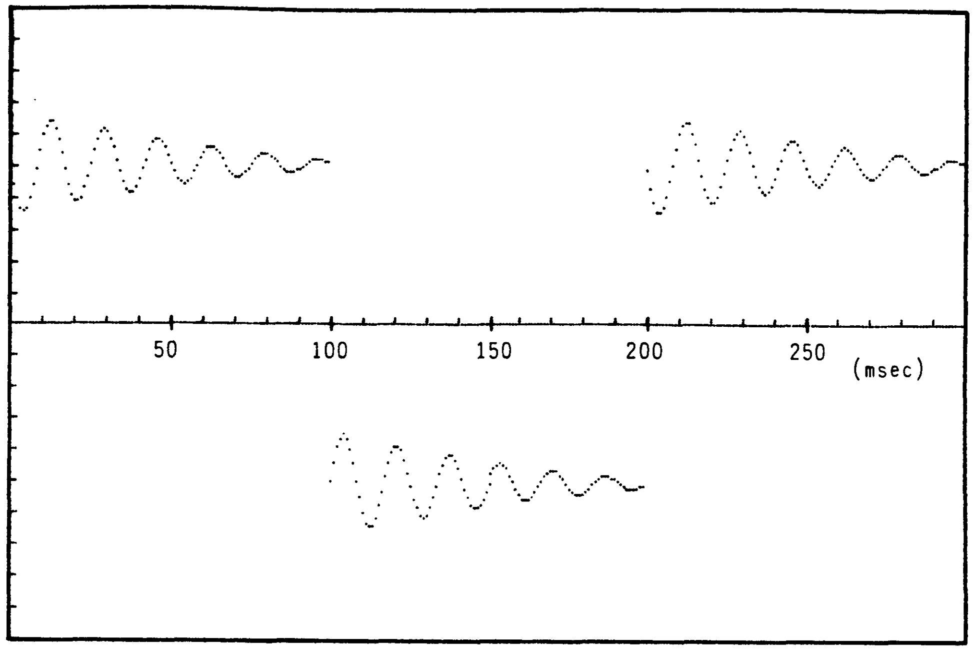


Figure 2.16 Notch filter (Q=10)

Lowpass filters are used to eliminate signal of unnecessary high frequency range. Simple first order RC-passive filters, for which the rolloff is -20 dB/decade, are used widely. Figure 2.17 shows the amplitude response of the passive filter, and the cutoff frequency f_c is calculated by

$$f_c = \frac{1}{2\pi RC} \quad (Hz) \tag{2.10}$$

Sometimes for the experiment or field measurement, when better lowpass filters are desired, state variable active filters were designed, as shown in Figure 2.18. State variable filters are widely used because cutoff frequency and Q-values are controlled easily by external resisters and lowpass, highpass, and bandpass response can be obtained simultaneously. A notch filter can be obtained by summing lowpass and highpass outputs of the state variable filters as shown in Figure 2.19. Higher order response filters can be obtained by cascading the second order state variable filters.

For electromagnetic transient measurements, Bessel response filter is the best due to its linear phase characteristics and minimum distortion of transient waveform.

For the design of Bessel response filters, the normalized frequency f_n and Q-values are selected from the Table 2.4, and necessary parameters are calculated by the following equations:

$$f_s = f_n \times f \qquad (2.11)$$

$$R_F = 1.592 \times 10^8 / f_s \qquad (2.12)$$

$$A_{BP} = Q \times A_{LP} = Q \times A_{HP} \qquad (2.13)$$

$$R_G = 5.0 \times 10^4 / A_{BP} \qquad (2.14)$$

$$R_Q = \frac{5.0 \times 10^4}{2Q - A_{BP} - 1} \qquad (2.15)$$

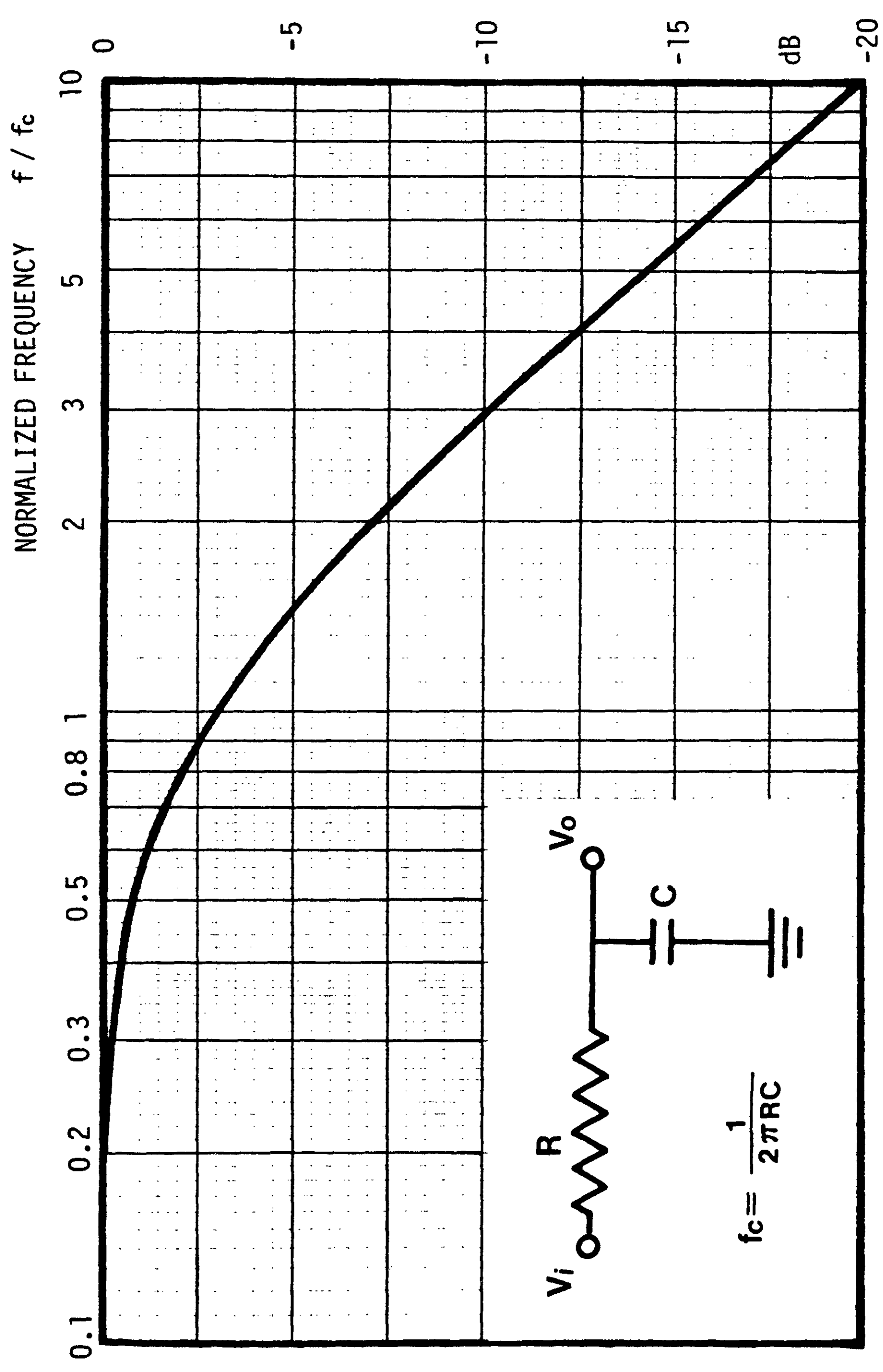


Figure 2.17 Amplitude Response of RC lowpass filter

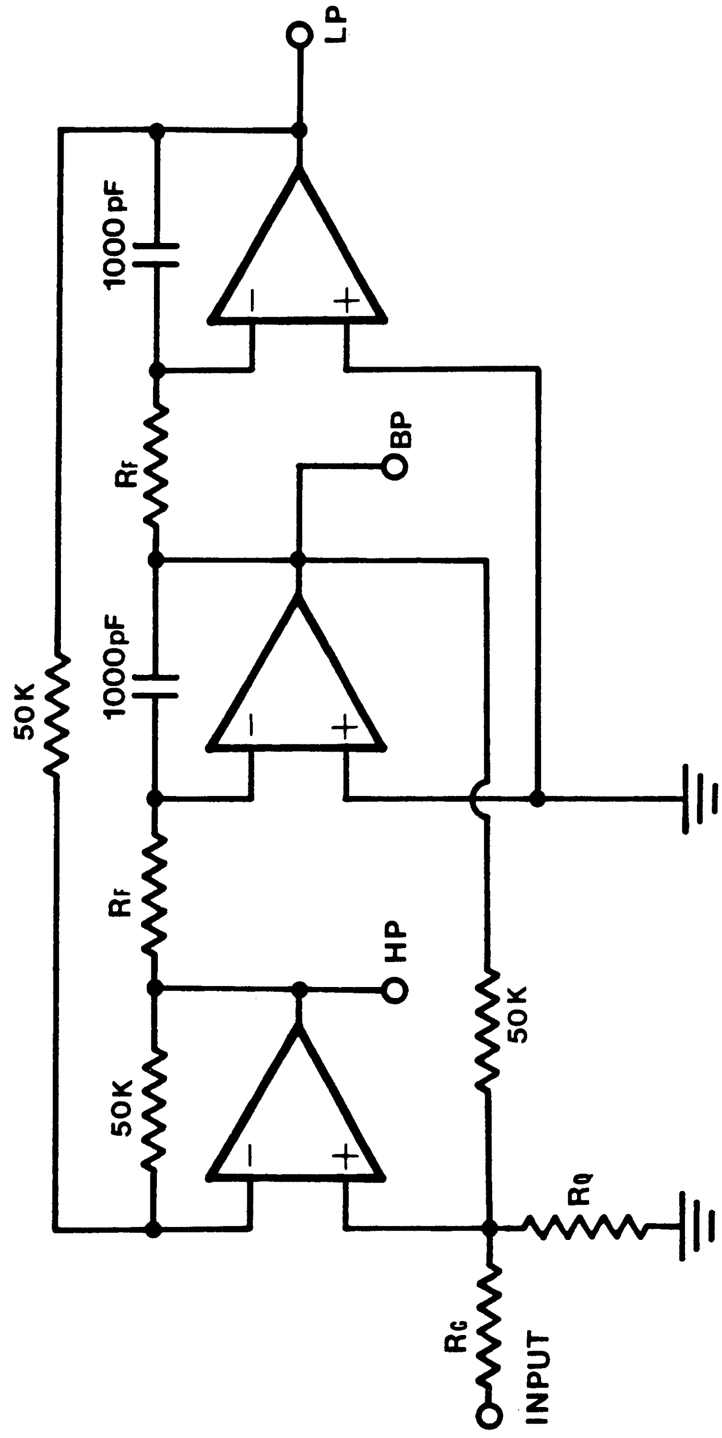


Figure 2.18 The state variable filter circuit

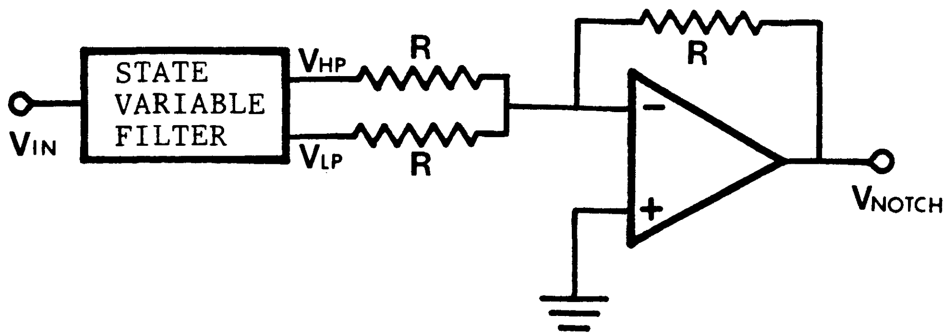


Figure 2.19 Notch response using state variable filter

number of poles	Butterworth fn	Butterworth Q	Bessel fn	Bessel Q
2	1.0	0.70711	1.2742	0.57735
3	1.0	-------	1.32475	-------
	1.0	1.0	1.44993	0.69104
4	1.0	0.54118	1.43241	0.52193
	1.0	1.3065	1.60594	0.80554
5	1.0	-------	1.50470	-------
	1.0	0.61805	1.55876	0.56354
	1.0	1.61812	1.75812	0.91652

Table 2.4 Parameters for Butterworth and Bessel Filters

2.3.4 Analog to digital converter

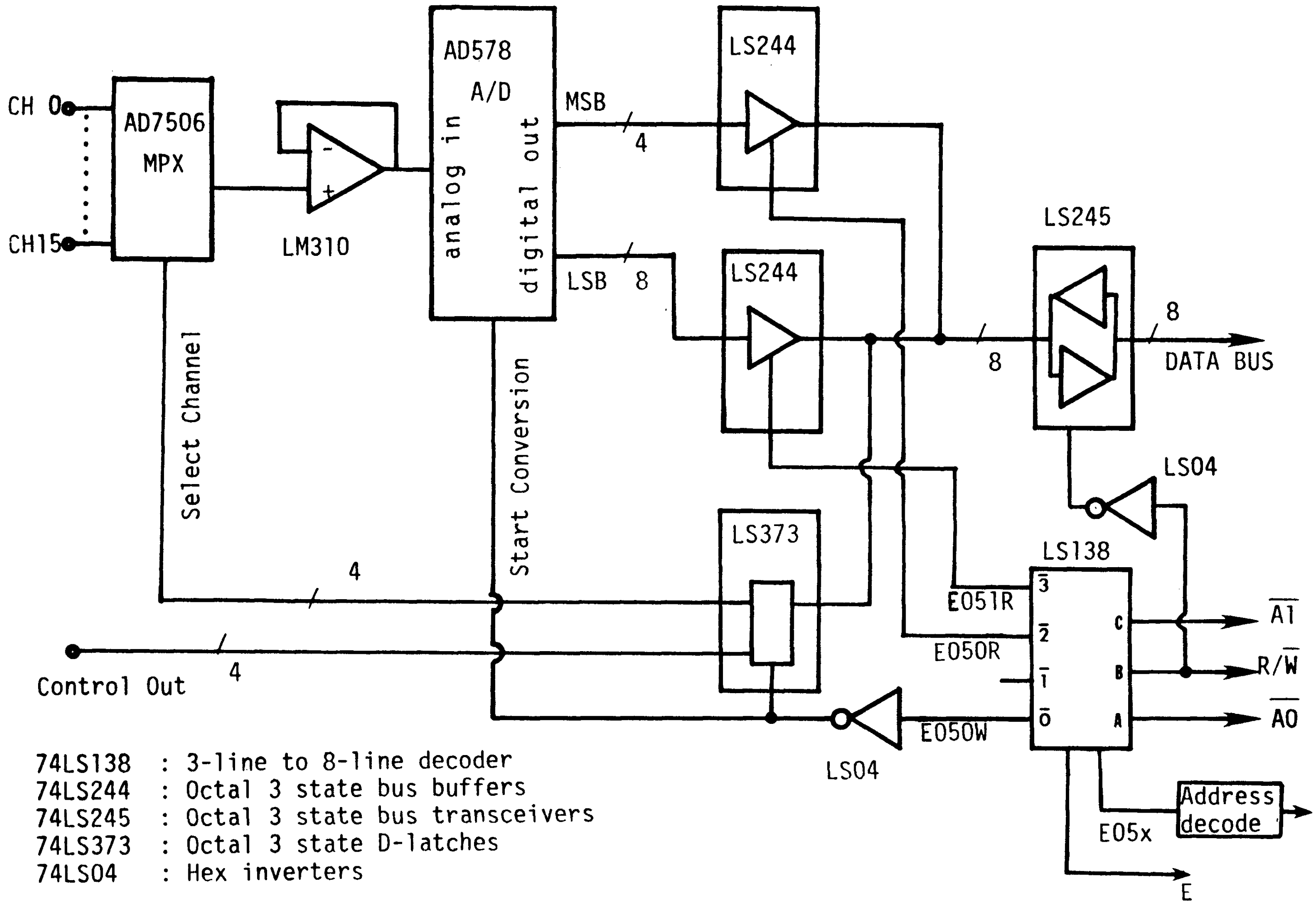
The analog voltage output of the amplifier is transformed to a digital number by the analog to digital converter. A very high speed 16 channel input analog to digital converter was prepaired using the Analog Devices AD578 analog to digital converter chip, as shown in Figure 2.20. The specifications of AD578 are as follows :

(1) Conversion time : 3 microseconds (grade L)
(2) Resolution : 12 bit
(3) Internal refrence : 10 volts ±0.1 %
(4) Gain error : ±0.1 % full scale
(5) Output : offset binary or two's complement
(6) Input impedance : 10 kilo-ohm

This analog to digital converter board was designed to act as a memory board from the central processing unit. Writing a channel number to the address of the board selects the channel of the multiplexer (AD7506) and also sends a start conversion signal to AD578 chip. Then, reading the data from the same address transfers the digitized number to the accumulator of the CPU. Table 2.5 shows the assembly language program of the A/D converter driver, and the numbers in the parenthesis indicate the machine cycles to execute the instruction. Since the clock speed of the M6809 is 1.5 MHz, it takes 2/3 microseconds to execute one machine cycle, and a total of 30 machine cycles are necessary to get one value, store to memory and incliment the counter. Therefore, the through put of the A/D conversion is maximum 50 KHz (20 microseconds/ sample).

```
*       A/D converter driver program

ADADR   EQU     $E050                   address of A/D board
TIMER   EQU     $E042                   address of clock


BEGIN   LDX     #MEMSTART               set IX to base address
        LDY     #NSAMPLE                set IY to number of samples
        BSR     TIMSET                  initialize and start clock

ADSTRT  LDA     CHN             (3)     set channel number
        STA     ADADR           (2)     and start conversion
        LDD     ADADR           (5)     load data to accumulator D
        STD     ,X++            (8)     save data to memory and inc. IX

WAIT    SYNC                    (2)     wait for time up
        LDA     TIMER           (2)     clear intrupt flag

        LEAY    -1,Y            (5)     decliment IY
        BNE     ADSTRT          (3)     loop if not finished all sample points
```

Table 2.5 Program of A/D converter Driver



74LS138 : 3-line to 8-line decoder
74LS244 : Octal 3 state bus buffers
74LS245 : Octal 3 state bus transceivers
74LS373 : Octal 3 state D-latches
74LS04 : Hex inverters

Figure 2.20 Block diagram of A/D converter

2.3.5 Main Processing unit and peripherals

The main processing unit, which controls the whole experimental system, consists of the central processing unit (CPU), memory (58K RAM, 2K ROM), programmable clock, serial and parallel I/O ports. The peripheral units, such as CRT terminal (Televideo 950), graphic printer (Epson MX-80 with graphtrax), digital X-Y plotter (Graphtec MIPLOT-J), dual floppydisk drive and telephone modem, are also controlled by the MPU. Figure 2.1 shows the block diagram of the MPU and peripherals.

For the CPU, a Motorolla M6809 8-bit microprocessor was selected because of its 16 bit internal registers and strong instruction set and addressing mode which are similer to PDP-11 mini-computer (Artwick, 1980). Internal registers and addressing mode are shown in Figure 2.21 and Table 2.6 respectively.

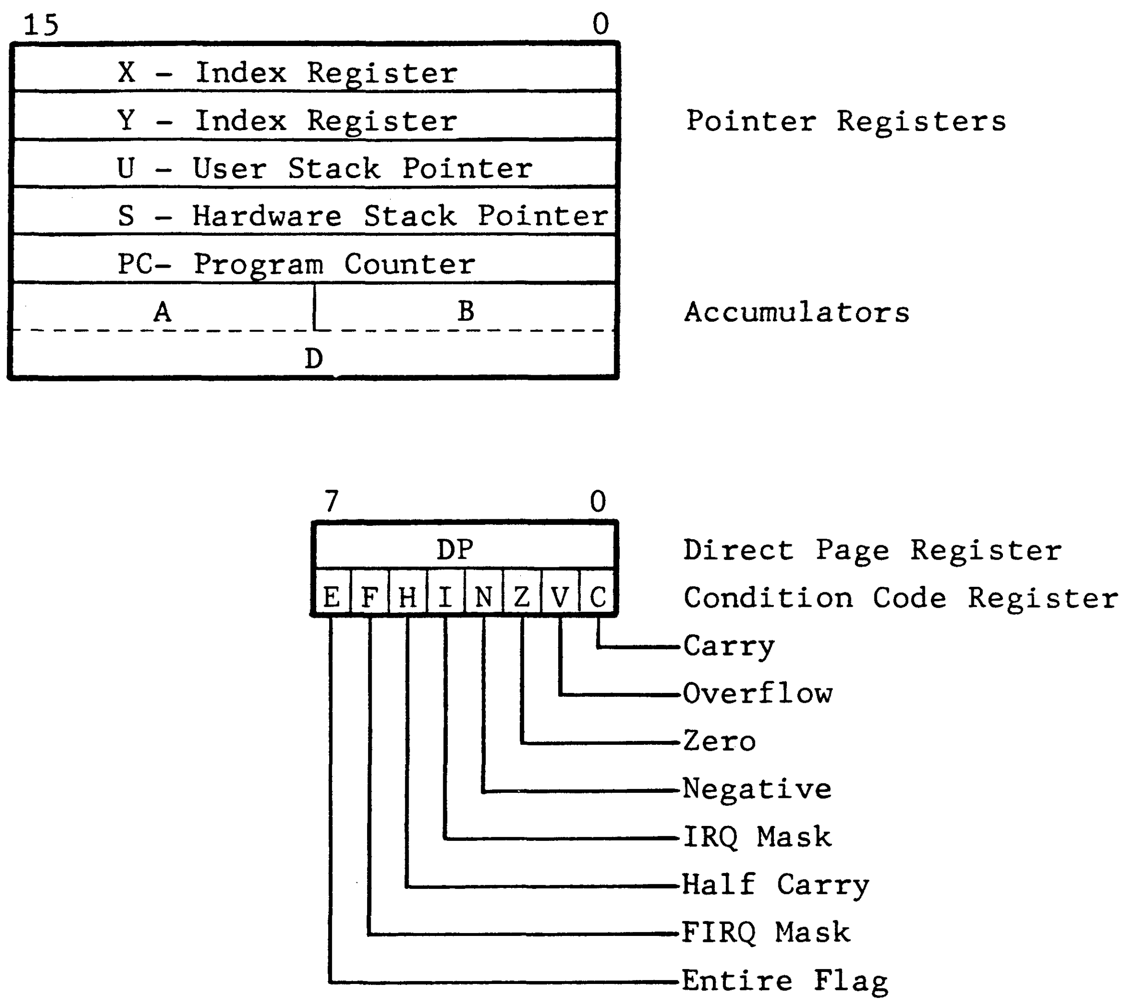


Figure 2.21 Internal registers of M6809

Type	Forms	Non Indirect				Indirect			
		Assembler Form	Postbyte OP Code	+ ~	+ #	Assembler Form	Postbyte OP Code	+ ~	+ #
Constant Offset From R	No Offset	,R	1RR00100	0	0	[,R]	1RR10100	3	0
(2's Complement Offsets)	5 Bit Offset	n, R	0RRnnnnn	1	0	defaults to 8-bit			
	8 Bit Offset	n, R	1RR01000	1	1	[n, R]	1RR11000	4	1
	16 Bit Offset	n, R	1RR01001	4	2	[n, R]	1RR11001	7	2
Accumulator Offset From R	A Register Offset	A, R	1RR00110	1	0	[A, R]	1RR10110	4	0
(2's Complement Offsets)	B Register Offset	B, R	1RR00101	1	0	[B, R]	1RR10101	4	0
	D Register Offset	D, R	1RR01011	4	0	[D, R]	1RR11011	7	0
Auto Increment/Decrement R	Increment By 1	,R+	1RR00000	2	0	not allowed			
	Increment By 2	,R++	1RR00001	3	0	[,R++]	1RR10001	6	0
	Decrement By 1	,–R	1RR00010	2	0	not allowed			
	Decrement By 2	,– –R	1RR00011	3	0	[,– –R]	1RR10011	6	0
Constant Offset From PC	8 Bit Offset	n, PCR	1xx01100	1	1	[n, PCR]	1xx11100	4	1
(2's Complement Offsets)	16 Bit Offset	n, PCR	1xx01101	5	2	[n, PCR]	1xx11101	8	2
Extended Indirect	16 Bit Address	–	–	–	–	[n]	10011111	5	2

R = X, Y, U or S
x = Don't Care

RR:
00 = X
01 = Y
10 = U
11 = S

$\overset{+}{\sim}$ and $\overset{+}{\#}$ indicate the number of additional cycles and bytes for the particular variation.

Table 2.6 Indexed Addressing Mode

2.3.6 Data acquisition software

Data acquisition software was developed using the OmegaSoft Pascal compiler and assembler under the FLEX (trademark of Technical Systems Consultants, Inc) operating system, and consists of the following three parts :

(1) Analog to digital conversion program
(2) Graphic plot program using MX-80 printer
(3) Data processing program using digital plotter

Programs (1) and (2) are combined into one program, ADPLT6 in Appendix B. The main part of the analog to digital conversion program is written using assembler language and explained in the section 2.3.4. The graphic plot program is written in Pascal and its main part is the procedure PIXEL(x,y) which sets a pixel(x,y) in the buffer memory frame consisting of 300 x 240 pixels (can be changed easily). A straight line is drawn using the DAA algorithm (symmetric digital differential analyzer, Harrington 1983), for which program is shown in Table 2.7.

The data processing program uses a digital X-Y plotter, and to drow a line or curve is very simple. Drawing a straight line from (x1,y1) to (x2,y2) is performed by the following Pascal sentence :

```
Procedure DAA( x1,y1,x2,y2 : integer ) ;
 (* Connect between two points (x1,y1) and (x2,y2) by a
    straight line using digital differential analyzer *)

  var
      i      : integer ; (* counter *)
      length : integer ; (* length of x or y direction *)
      x,y    : real    ; (* coordinate to set a pixel *)
      xinc,yinc : real ; (* value of incremant *)
      dx,dy  : real    ;

  begin
     dx := x2 - x1 ;
     dy := y2 - y1 ;
     length := abs(dx) ;
     if abs(dy)   length then length := abs(dy) ;
     xinc := dx / length ;
     yinc := dy / length ;
     x := x1 + 0.5 ;
     y := y1 + 0.5 ;
     for i := 1 to length do
        begin
           PIXEL( trunc(x), trunc(y) ); (* set a pixel *)
           x := x + xinc ;
           y := y + yinc
        end
  end;  (* of DAA *)
```

Table 2.7 DAA program to draw a straight line

```
Writeln(auxout,'D',x1,y1,x2,y2);
        where x1,y1,x2,y2 are integer numbers.
```

The data processing program WPLOT, in Appendix C, has also a cubic spline interpolation procedure to draw a smooth curve connecting the data points. The spline fitting algorithm is used to minimize the following integral (Ruckdeschel 1981) :

$$D = \int \left(\frac{d^2g}{d\,x^2} - \frac{d^2f}{d\,x^2} \right) dx$$

where f(x) : true function

g(x) : fitted curve

and tends to minimize net curvature while passing through all the data points. Figure 2.25 shows a comparison of curve fitting programs using cubic spline and Lagrange interpolation. In case of common Lagrange interpolation, the fitted curve tends to oscillate, especially near the edges of the data area.

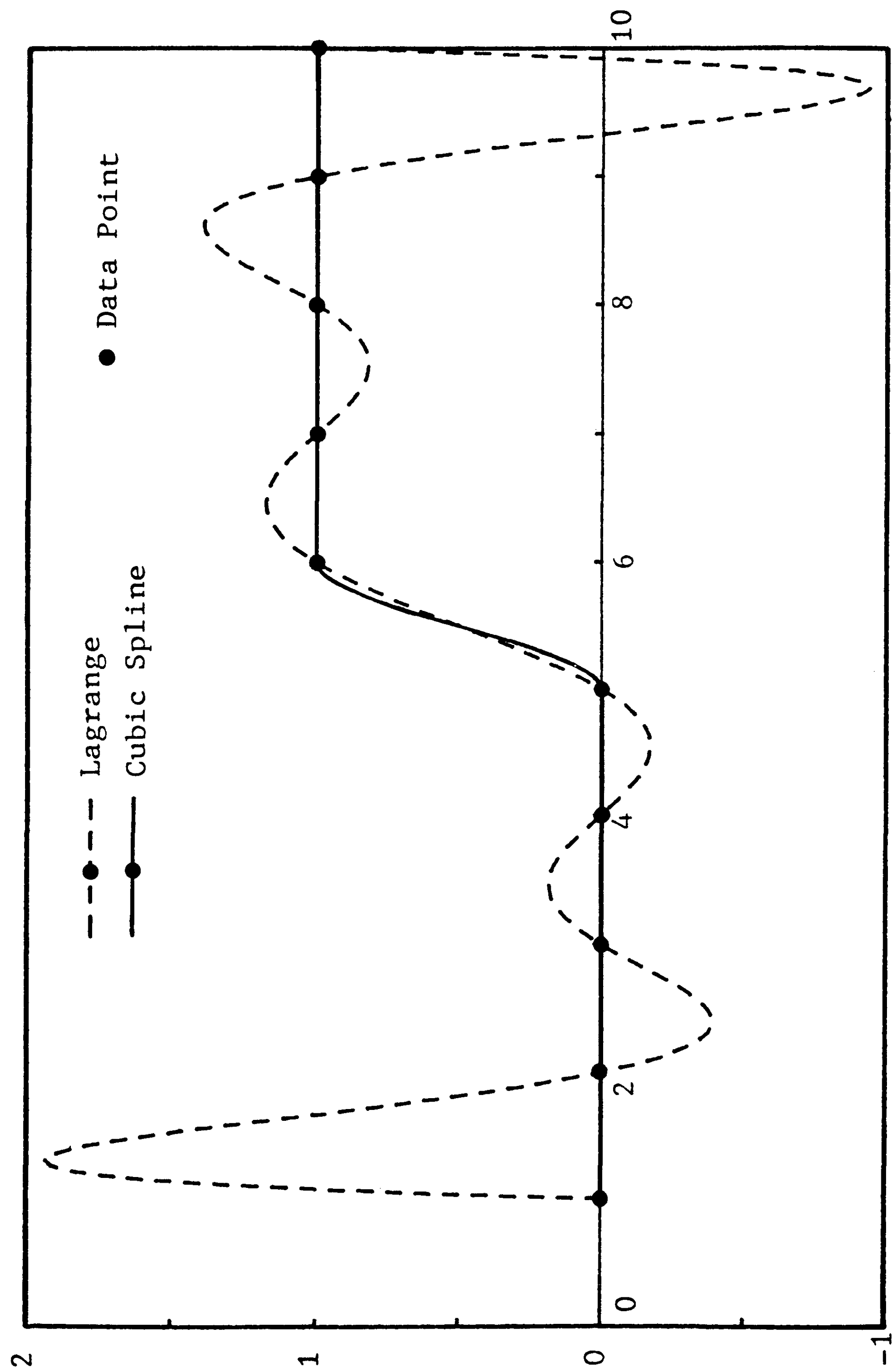


Figure 2.22 Cubic Spline and Lagrange interpolation.

CHAPTER 3

EVALUATION OF THE ACCURACY OF THE SYSTEM

3.1 Introduction

To perform physical modeling, the accuracy of the system must at first be examined, since the results of modeling contain various errors caused by instrument, array and model. Those errors can be summarized as follows:

(1) Errors due to instruments

1. Error and non-linearity of the gain with amplifier.
2. Non-linearity of the A/D converter.
3. Drift of frequency in frequency domain measurement.
4. Drift of sampling time in time domain measurement.
5. Measurement error of voltage, current and phase.

(2) Errors due to array

1. Measurement error of linear dimensions including transmitter-receiver separation.
2. Error of the alignment of transmitter and receiver.
3. Error of transmitter and receiver moments.

(3) Error due to model

1. Error of the size and orientation of model.
2. Error of conductivities of structures.

Therefore, it is important to check the accuracy of the system by comparing the experimental results to exact solutions. A conducting sphere model was selected as the standard model for this purpose, since the response of the conducting sphere for the geophysical problems has been studied by many authors. Wait has studied the frequency and time domain responce for a uniform field (1951), for a dipole field (1953), and for a two-layer sphere (1969). Negi also considered the frequency response of a sphere in free space.

The electromagnetic response of a conducting sphere is explained in following sections. Frequency and transient responses were also measured by the developed system and compared with the theoretical results.

3.2 Conducting sphere in a uniform magnetic field

3.2.1 Frequency domain response

If a conducting sphere is situated beneath the large transmitter loop (Figure 3.1), the electric field caused by the loop does not intersect the surface of the sphere, so no charges appear on the surface of the body. The electromagnetic field can be expressed by Maxwell's equations as follows if the displacement currents are neglected :

$$\text{curl}\ \overline{H} = \sigma \overline{E} \tag{3.1}$$

$$\text{curl } \overline{E} = i\omega\mu\overline{H} \qquad (3.2)$$

$$\text{div } \overline{H} = 0 \qquad (3.3)$$

$$\text{div } \overline{E} = 0 \qquad (3.4)$$

where $\overline{E}$ and $\overline{H}$ are the complex amplitudes of the electrical and magnetic fields respectively.

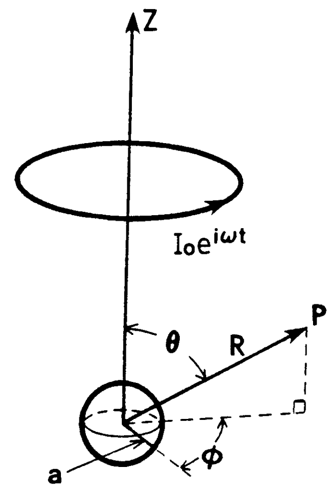


Figure 3.1 Conducting sphere

From equations (3.1) and (3.2), we have

$$\text{curl curl } \overline{E} = i\omega\mu\sigma\overline{E} = k^2\overline{E} \qquad (3.5)$$

where

$$k = \sqrt{i\omega\mu\sigma} = \frac{1+i}{\delta} \qquad (3.6)$$

and δ is called skin depth and expressed as :

$$\delta = \sqrt{\frac{2}{\sigma\mu\omega}} \qquad (3.7)$$

Then, the electric field inside and outside of the conducting sphere can be written as :

$$\text{curl curl } \overline{E} - k^2\overline{E} = 0 \qquad R < a \qquad (3.8)$$

$$\text{curl curl } \overline{E} = 0 \qquad R > a \qquad (3.9)$$

where a is the radius of the sphere. On the surface of the sphere, the following boundary conditions can be used :

$$E^i_\phi = E^e_\phi \qquad R = a \tag{3.10}$$

$$H^i_\theta = H^e_\theta \qquad R = a \tag{3.11}$$

where E^i_ϕ, H^i_θ and E^e_ϕ, H^e_θ are the electric and magnetic field inside and outside the sphere respectively.

By introducing spherical coordinate system (R,θ,ϕ), equation 3.11 can be rewritten using electrical field as follows:[N]

$$\frac{1}{\mu_i}\frac{\partial(RE^i_\phi)}{\partial R} = \frac{1}{\mu_e}\frac{\partial(RE^e_\phi)}{\partial R} \tag{3.12}$$

Curl $\overline{E}$ has only R and θ components since electric field has only a ϕ component :

$$\text{curl}_R\overline{E} = \frac{1}{R\sin\theta}\frac{\partial}{\partial\theta}(\sin\theta E_\phi) \tag{3.13}$$

$$\text{curl}_\theta\overline{E} = \frac{1}{R}\frac{\partial}{\partial R}(RE_\phi) \tag{3.14}$$

$$\text{curl}_\phi\overline{E} = 0 \tag{3.15}$$

Then, curl curl $\overline{E}$ has only a ϕ component :

$$\text{curl curl }\overline{E} = -\frac{1}{R}\frac{\partial^2}{\partial R^2}(RE_\phi) - \frac{1}{R}\frac{\partial}{\partial\theta}\frac{1}{\sin\theta}\frac{\partial}{\partial\theta}\sin\theta E_\phi \tag{3.16}$$

and substituting to equations (3.8) and (3.9), we have the following differential equations for the electric fields inside and outside of the sphere :

$$\frac{1}{R}\frac{\partial^2}{\partial R^2}(RE^i_\phi)+\frac{1}{R^2}\frac{\partial}{\partial\theta}\frac{1}{\sin\theta}\frac{\partial}{\partial\theta}(\sin\theta E^i_\phi)+k^2E^i_\phi = 0 \qquad R<a \qquad (3.17)$$

$$\frac{1}{R}\frac{\partial}{\partial R}(RE^e_\phi)+\frac{1}{R}\frac{\partial}{\partial\theta}\frac{1}{\sin\theta}\frac{\partial}{\partial\theta}(\sin\theta E^e) = 0 \qquad R>a \qquad (3.18)$$

These equations can be solved using the separation of variables method, by assuming that the electric field E_ϕ can be expressed as the multiplication of two independent functions $T(R)$ and $\Phi(\theta)$:

$$E_\phi = T(R)\Phi(\theta) \qquad (3.19)$$

Substituting this equation into equation (3.17) and multiplying $R^2/T\Phi$, we have the following equation :

$$\frac{R}{T}\frac{\partial}{\partial R}(RT) + \frac{1}{\Phi}\frac{\partial}{\partial\theta}\frac{1}{\sin\theta}\frac{\partial}{\partial\theta}(\sin\theta\Phi) + k^2R^2 = 0 \qquad (3.20)$$

The first and third terms of the equation are a function of R, and the second term is a function of θ, so equation (3.20) is equivalent to the following two ordinary differential equations :

$$\frac{R}{T}\frac{d^2(RT)}{dR^2} + k^2R^2 = m \tag{3.21}$$

$$\frac{1}{\Phi}\frac{d}{d\theta}\frac{1}{\sin\theta}\frac{d}{d\theta}(\sin\theta\Phi) = -m \tag{3.22}$$

From equation (3.21) we have the following Bessel's differential equation :

$$\frac{d^2T}{dR^2} + \frac{2}{R}\frac{dT}{dR} + (k - \frac{m}{R^2})T = 0 \tag{3.23}$$

and the solution is expressed by the modified Bessel function Z_s as follows:

$$T(k,R) = R^{-\frac{1}{2}} Z_s\ (ikR) \tag{3.24}$$

where

$$s = (\frac{1}{4} + m)^{\frac{1}{2}} \tag{3.25}$$

From equation (3.22), by introducing a new variable $u=\cos\theta$, we have the following Legendre's differential equation :

$$(1-u^2)\frac{d^2\Phi}{du^2} - 2u\frac{d\Phi}{du} - \frac{\Phi}{1-u^2} + m\Phi = 0 \tag{3.26}$$

The solution can be expressed through the associate Legendre function P_n and Q_n, and have a solution only if m satisfies the following equation :

$$m = n(n+1) \tag{3.27}$$

so the general solutions take the following form :

$$T_n(k,R) = R^{-\frac{1}{2}}\left[A_n I_p(ikr) + B_n K_p(ikr)\right] \tag{3.28}$$

$$\Phi(\theta) = C_n P_n^{(1)}(u) + D_n Q_n^{(1)}(u) \tag{3.29}$$

where
$$p = \sqrt{\frac{1}{4} + n(n+1)} \tag{3.30}$$

I_p, K_p : modified Bessel functions of the first and second kinds

$P_n^{(1)}, Q_n^{(1)}$: associate Legendre functions

A_n, B_n, C_n, D_n : constants

The functions $Q_n^{(1)}$ and K_p cannot be used to describe the field because $Q_n^{(1)}$ goes to infinity on the z-axis (θ=0), and K_p goes to infinity at the center of the sphere (R=0). Then, the general solution for the electric field can be written using equation (3.19) as :

$$E_\phi^i = R^{-\frac{1}{2}} \sum_{n=1}^{\infty} B_n I_p(ikr)\, P_n^{(1)}(\cos\theta) \tag{3.31}$$

Outside of the sphere, equation (3.21) can be written as :

$$\frac{d}{dR}(R^2 T') - n(n+1)T = 0 \tag{3.32}$$

and the solutions of this equation are elementary functions of the form :

$$T_n(R) = C_n R^n + F_n R^{-n-1} \qquad (3.33)$$

The field due to the currents inside the sphere should go to zero when R goes to infinity, so the first term of the right-hand side of equation (3.33) cannot be used. The general solution for the electric field ($E^e_{1\phi}$) outside of the sphere due to the currents inside the sphere can be written as:

$$E^e_{1\phi} = \sum_{n=1}^{\infty} F_n R^{-n-1} P_n^{(1)}(\cos\theta) \qquad (3.34)$$

and the primary field $E_{o\phi}$ is :

$$E_{o\phi} = \frac{i\omega\mu H_{oz}}{2} R \sin\phi$$

$$= \frac{i\omega\mu H_{oz}}{2} R P_1^{(1)}(u) \qquad (3.35)$$

Then, the electric field inside and outside of the sphere can be expressed as :

$$E^e_{\phi} = \frac{i\omega\mu_e H_o}{2}\left[R P^{(1)}(u) + \sum_{n=1}^{\infty} D_n R^{-n-1} P_n^{(1)}(u)\right] \qquad (3.36)$$

$$E^i_{\phi} = \frac{i\omega\mu_i H_o}{2} R^{-\frac{1}{2}} \sum_{n=1}^{\infty} C_n I_p(ikR) P_n^{(1)}(u) \qquad (3.37)$$

Since these equations must satisfy the boundary conditions

(3.10) and (3.12), the coefficients C_n and D_n can be determined. Due to the orthogonality of the associated Legendre functions, coefficients C_n and D_n are zero except C_1 and D_1, so we have two equations with two unknowns :

$$a + D_1 a^{-2} = a^{-\frac{1}{2}} C_1 I_{\frac{3}{2}}(ika) \tag{3.38}$$

$$\mu_i(2a^3 - D_1) = \mu_e a^{\frac{3}{2}} C_1 \left[\frac{1}{2} I_{\frac{3}{2}}(ika) + ika I'_{\frac{3}{2}}(ika)\right] \tag{3.39}$$

Then coefficients C_1 and D_1 can be determined as :

$$C_1 = \frac{(2\mu_i+\mu_e)\, x\, I_{-\frac{1}{2}}(x) - [\mu_e(1+x^2) + 2\mu_i]\, I_{\frac{1}{2}}(x)}{(\mu_i-\mu_e)\, x\, I_{-\frac{1}{2}}(x) + [\mu_e(1+x^2) - \mu_i]\, I_{\frac{1}{2}}(x)} a^3 \tag{3.40}$$

$$D_1 = \frac{3\mu_i x\, a^{\frac{3}{2}}}{(\mu_i-\mu_e)\, x\, I_{-\frac{1}{2}}(x) + [\mu_e(1+x^2) - \mu_i]\, I_{\frac{1}{2}}(x)} \tag{3.41}$$

where $x=ika$, and a is the radius of the sphere .

The complex amplitude of the electric field inside and outside of the sphere can be written as :

$$E^i_\phi = \frac{i\omega\mu_e}{2} H_{oz} C_1 R^{-\frac{1}{2}} I_{\frac{3}{2}}(ikR)\sin\theta \qquad a \geqq R \tag{3.42}$$

$$E^e_\phi = E_{o\phi} + \frac{i\omega\mu_e}{2} H_{oz} D_1 R^{-2}\sin\theta \qquad a \leqq R \tag{3.43}$$

and the complex amplitude of the magnetic field due to the currents in the sphere can be written as :

$$H_{1R}^{e} = \frac{D_1}{R^3} H_{oz} \cos\theta \qquad (3.44)$$

$$H_{1\theta}^{e} = \frac{D_1}{2R^3} H_{oz} \sin\theta \qquad (3.45)$$

The magnetic field due to a dipole at the origin of spherical coordinate system can be expressed as :

$$H_R = \frac{M}{2\pi R^3} \cos\theta \qquad (3.46)$$

$$H_\theta = \frac{M}{4\pi R^3} \sin\theta \qquad (3.47)$$

Comparing these equations with equations (3.44) and (3.45), the magnetic field created by the currents inside the sphere is same as the field due to a dipole located at the center of the sphere and the magnitude of moment $M = 2\pi D_1 H_{oz}$.

When the sphere is made of non-magnetic materials ($\mu_e = \mu_i$), rewriting the associated Bessel functions using hyperbolic functions, we have :

$$I_{\frac{1}{2}}(x) = \sqrt{\frac{2}{\pi x}} \sinh x$$

and $$I_{-\frac{1}{2}}(x) = \sqrt{\frac{2}{\pi x}} \cosh x$$

then we have : $$D = \frac{3x\cosh x - (3+x^2)\sinh x}{x^2 \sinh x}$$

$$= \frac{3 \coth x}{x} - \frac{3}{x^2} - 1 \qquad (3.48)$$

and $$D_1 = D\, a^3 \qquad (3.49)$$

where $x = ika = p(1-i)$

$p = \frac{a}{\delta}$

δ : skin depth

A computer program written in the Pascal language to calculate the frequency domain response of a conducting sphere is listed in Appendix D.

The complex function D has inphase and quadrature components, which have their own frequency responses. If we consider the low frequency part of equation (3.48), the function coth x can be expressed through the series :

$$\coth x = \frac{1}{x} + \frac{x}{3} - \frac{x^3}{45} + \frac{2\,x^5}{945} - \cdots\cdots \qquad (3.50)$$

Then, D can be written as :

$$D = i\frac{\sigma\mu a^2}{15}\omega - \frac{2}{315}(\sigma\mu a^2)^2\omega^2 + i\frac{1}{1575}(\sigma\mu a^2)^3\omega^3 - \ \ldots. \qquad (3.51)$$

Therefore, at low frequencies, the magnetic field can be written as follows if we neglect the terms higher than third order :

$$H_R = \frac{a^3 H_o}{R}\cos\theta \left[i\frac{1}{15}\sigma\mu a^2\omega - \frac{2}{315}(\sigma\mu a^2)^2\omega^2 \right] \quad (3.52)$$

$$H_\theta = \frac{a^3 H_o}{2R}\sin\theta \left[i\frac{1}{15}\sigma\mu a^2\omega - \frac{2}{315}(\sigma\mu a^2)^2\omega^2 \right] \quad (3.53)$$

At high frequencies ($p \gg 1$), the complex function D can be expressed as :

$$D = (-1 + \frac{3}{2p}) + \frac{3}{2p}i$$

Then, the magnetic fields can be expressed as :

$$H_R = \frac{a^3 H_o}{R^3} \cos\theta \left[-1 + \frac{3\sqrt{2}}{2}\frac{1}{\sigma\mu\omega a^2}(1+i) \right] \quad (3.54)$$

$$H_\theta = \frac{a^3 H_o}{2R^3} \sin\theta \left[-1 + \frac{3\sqrt{2}}{2}\frac{1}{\sigma\mu\omega a^2}(1+i) \right] \quad (3.55)$$

Therefore, if frequency increases, quadrature components approach zero, and only inphase components remains.

3.2.2 Frequency domain experiment

Frequency domain experiments were performed and results were compared with those of theoretical calculations in order to examine the accuracy of the system.

A schematic of the frequency domain physical modeling

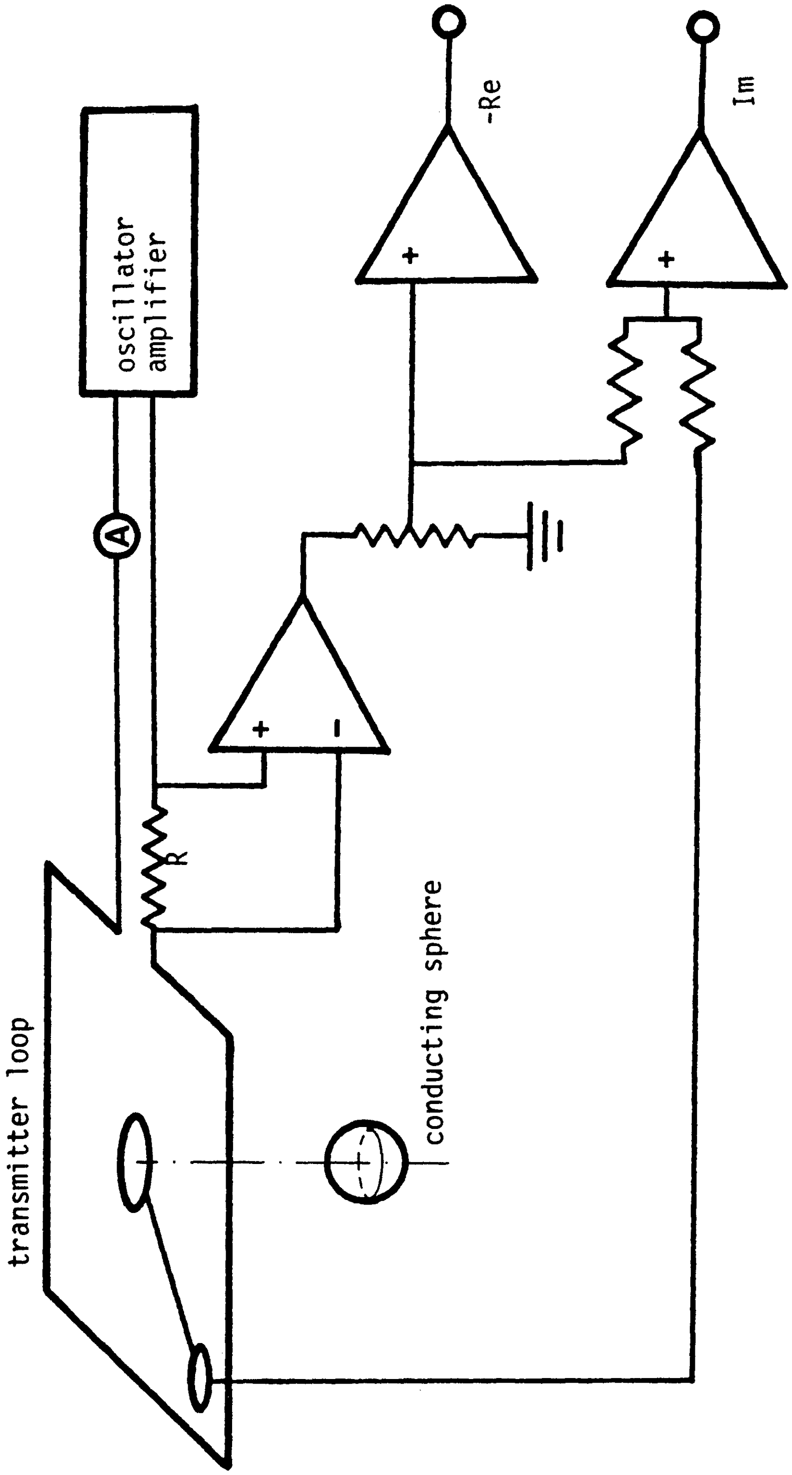


Figure 3.2 Block diagram of frequency domain experiment

system is shown in figure 3.2. One receiver coil is placed at the center of the transmitter loop, and electromotive force is measured. The other reference coil which has a smaller moment is connected in series and placed near the transmitter loop (far from the sphere). The primary field is cancelled by moving the reference coil near the transmitter loop to obtain the minimum voltage reading without the model.

Then the sphere is placed on the axis of the transmitter loop. A variable resister VR1 is used to adjust the magnitude of voltage A, which has the opposite direction to the real part (Re V) as shown in figure 3.3.

When the magnitude of A is the same as Re V, the sum of A and V is the imaginary part of the voltage (Im V), and this value is obtained as the minimum reading of the voltmeter VM1.

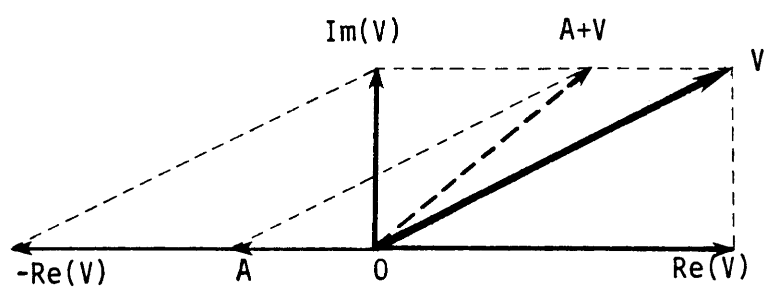


Figure 3.3 Measurement of real and imaginary parts

Fig 3.4 shows an example of the comparison of the results of the experiment and numerical calculation. Continuous curves show the calculated results and curves with square marks show the results of experiment.

Since the conductivity of the sphere is not known accurately, we should investigate how to determine that conductivity. From equation (3.52), at low frequencies, the z-component of the magnetic field can be written as follows :

$$Hz = \frac{a^3}{Z^3} Hoz \left[i\frac{1}{15}\sigma\mu a^2\omega - \frac{2}{315}(\sigma\mu a^2)^2\omega^2 \right] \qquad (3.55)$$

where $$Hoz = \frac{M_T}{2\pi(r_T^2 + Z^2)^{\frac{3}{2}}} \qquad (3.56)$$

and $M_T = \pi r_T^2 In$

I : current of transmitter loop

r_T : radius of the transmitter loop

n : number of terns of transmitter loop

From equation (3.55), if we take the ratio of Inphase and Quadrature components, we have :

$$\sigma = \frac{1}{\omega\mu a^2} \frac{315}{30} \frac{In(H_Z)}{Q(Hz)} \qquad (3.57)$$

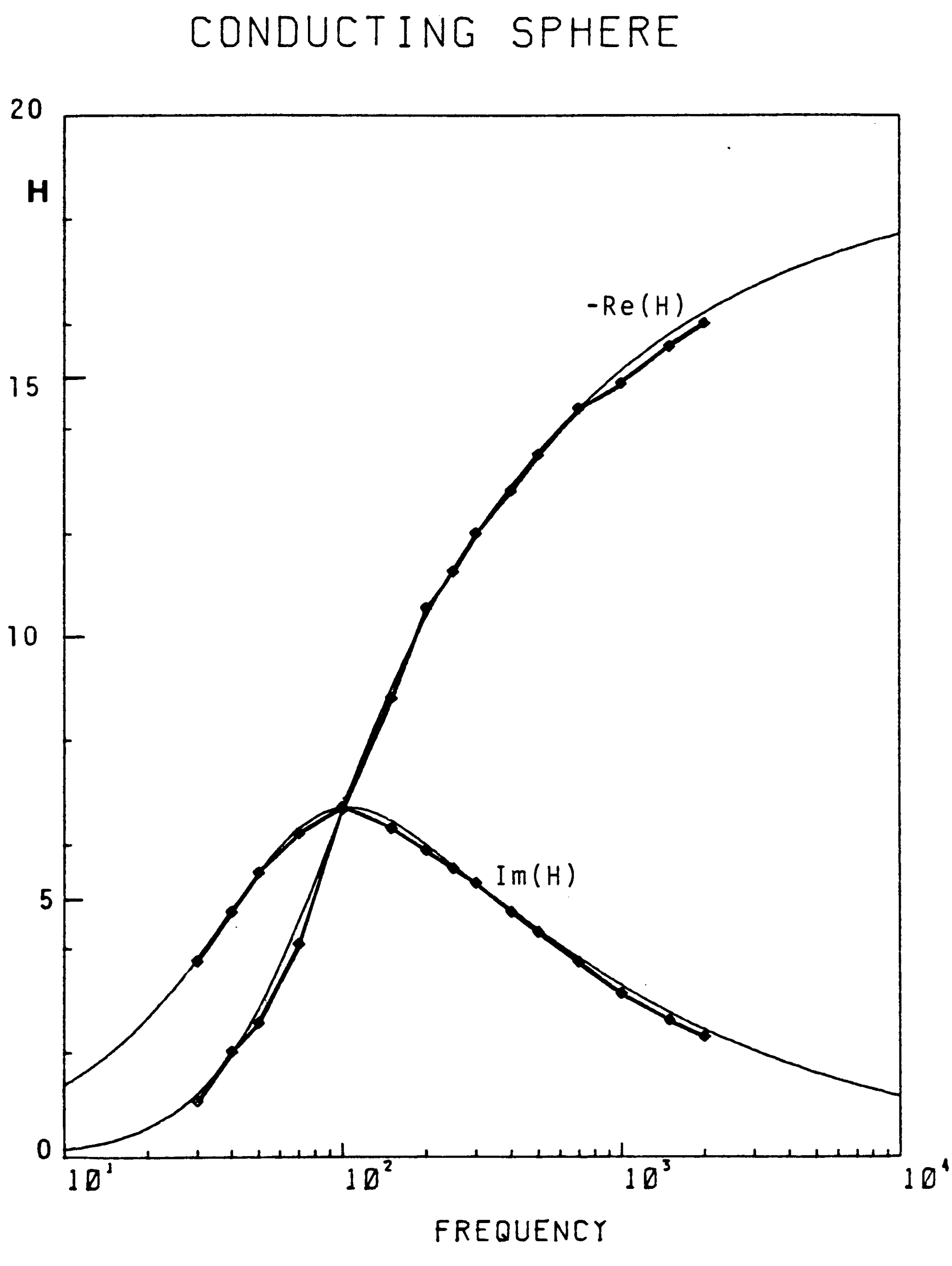


Figure 3.4 Frequency response of conducting sphere

From Figure 3.4, the values In(Hz)=2.01 and Q(Hz)=4.71 were obtained at a frequency of 40 Hz. The conductivity of the sphere can be calculated as :

$$\sigma = \frac{1}{2\pi \cdot 40 \cdot 0.052 \cdot 4\pi \cdot 10^{-7}} \; \frac{315}{30} \; \frac{2.01}{4.17}$$
$$= 5.24 \times 10^{6}$$

Then, we can define the conductivity of the sphere from the frequency domain experiment.

3.2.3 Time domain response of conducting sphere

Time domain response can be obtained by taking a Fourier transform of the frequency domain response obtained in the section 3.2. The function coth x used in equation (3.48) can be expressed as :

$$\coth x = \frac{1}{x} + 2x \sum_{n=1}^{\infty} \frac{1}{\pi^2 n^2 + x^2} \tag{3.58}$$

and if we substitute to equation (3.48), we have :

$$D = 6 \sum_{n=1}^{\infty} \frac{1}{\pi^2 n^2 + x^2} - 1$$

$$= 6\alpha \sum_{n=1}^{\infty} \frac{1}{p_n - i\omega} - 1$$

where

$$p_n = \pi^2 n^2 \alpha$$

and

$$\alpha = \frac{1}{\sigma \mu a^2} \tag{3.59}$$

Then if we use the following relation :

$$\sum_{n=1}^{\infty} \frac{1}{n^2} = \frac{\pi}{6}$$

we have

$$D = 6\alpha \sum_{n=1}^{\infty} \left(\frac{1}{p_n - i\omega} - \frac{1}{p_n} \right)$$

$$= 6\alpha \sum_{n=1}^{\infty} \frac{i\omega}{p_n (p_n - i\omega)} \tag{3.60}$$

So, the magnetic field H_R^e can be written as :

$$H_R^e(\omega) = \frac{a^3 H_o}{R^3} \cos\theta \left(6\alpha \sum_{n=1}^{\infty} \frac{1}{p_n} \frac{i\omega}{p_n - i} \right) \tag{3.61}$$

and by taking the Fourier transform for the step response, we have the following equations :

$$H_R^e(t) = \frac{a^3 H_o}{R^3} \cos\theta \left(6\alpha \sum_{n=1}^{\infty} \frac{1}{2\pi p_n} \int_{-\infty}^{\infty} \frac{e^{-i\omega t} d\omega}{p_n - i\omega} \right)$$

$$= \frac{a^3 H_o}{R^3} \cos\theta \frac{6}{\pi^2} \sum_{n=1}^{\infty} \frac{1}{n^2} e^{-\pi^2 n^2 \alpha t} \tag{3.62}$$

In the same way, θ-component of magnetic field can be obtained as :

$$H^e_\theta(t) = \frac{a^3 H_o}{R^3} \sin\theta \frac{3}{\pi^2} \sum_{n=1}^{\infty} \frac{1}{n^2} e^{-\pi^2 n^2 \alpha t} \quad (3.63)$$

and for the vertical component of the magnetic field H :

$$H_Z(t) = \frac{a^3}{Z^3} H_{oZ} \frac{6}{\pi^2} \sum_{n=1}^{\infty} \frac{1}{n^2} e^{-\pi^2 n^2 \alpha t} \quad (3.64)$$

For the electromotive force to the vertical receiver coil, we have :

$$V(t) = - \frac{6 a^3 M_R \alpha}{Z^3} \mu H_{oZ} \sum_{n=1}^{\infty} e^{-\pi^2 n^2 \alpha t} \quad (3.65)$$

where z is the depth of the sphere and is expressed by equation (3.59). H_{oZ} is the primary magnetic field at the center of the sphere and expressed by equation (3.56). A computer program is shown in appendix E to calculate time domain response of the conducting sphere.

When $\pi^2 \alpha t > 1$ (late time), the higher order terms decay quickly due to the exponential decay and only the first term remains :

$$H_Z(t) = \frac{a^3}{Z^3} H_{oZ} \frac{6}{\pi^2} e^{-q_1 \alpha t} \quad (3.66)$$

where $q_1 = \pi^2$

and for the electromotive force :

$$V(t) = -\frac{6a^3 M_R \alpha}{Z^3} \mu H_{\sigma z} e^{-\pi^2 \alpha t} \tag{3.67}$$

If we plot the field using a logarithmic ordinate (linear abscissa), the curve approaches a straight line at late time since the field can be expressed by only one exponential term. The slope of the line is :

$$\text{slope} = -q_1 \alpha = -\frac{\pi^2}{\sigma \mu a^2} \tag{3.68}$$

Then, by measuring the slope of the straight line, we can determine the conductivity of the body.

Calculated results are shown in figure 3.5 to compare the equations (3.65) and (3.67). If the conductivity of the sphere is smaller, the electromotive force of the receiver coil decays faster. At the late time, curves approach straight lines, the slope of which is expressed by equation (3.68).

3.2.4 Time domain experiment

A time domain experiment was performed using conducting sphere, to check the accuracy of the system. A block diagram of the system is shown in figure 2.1, where the receiver coil was placed at the center of the transmitter loop.

Figure 3.6 shows the comparison of the theoretical calculation and measured values. The sampling interval of the data is 100 microseconds and the data were stacked ten times. Dashed lines show the calculated values and continuous lines with square symbols show the measured data. The first data point is sampled 50 microseconds after the turn off of the current.

Measured data coincide well with the results of theoretical calculation. The curve approaches a straight line, and the slope can be obtained using the exponential regression procedure of WPLOT program (Appendix C) :

$$\text{slope} = -0.559 \text{ volts } / \text{ 1 millisecond}$$

Then, from equation (3.68), the conductivity of the sphere can be calculated as :

$$\sigma = 5.20 \cdot 10^{6} \text{ mho/m}$$

This result is very close to that of the frequency domain experiment (about 1%).

For both the frequency domain and time domain experiment, obtained data are very close to the results of theoretical calculations, and the accuracy of the system has been confirmed.

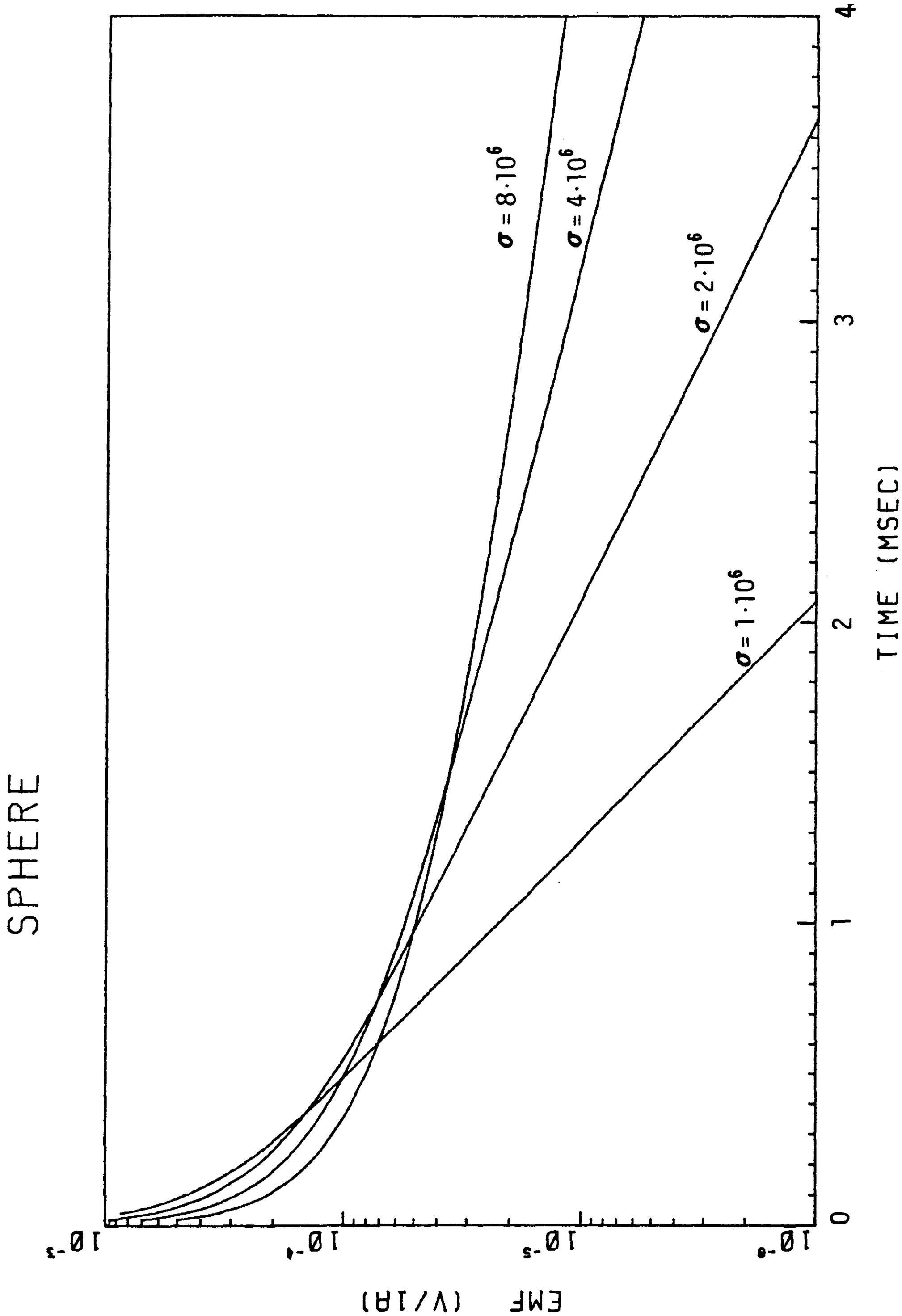


Figure 3.5 Time domain response of conducting sphere

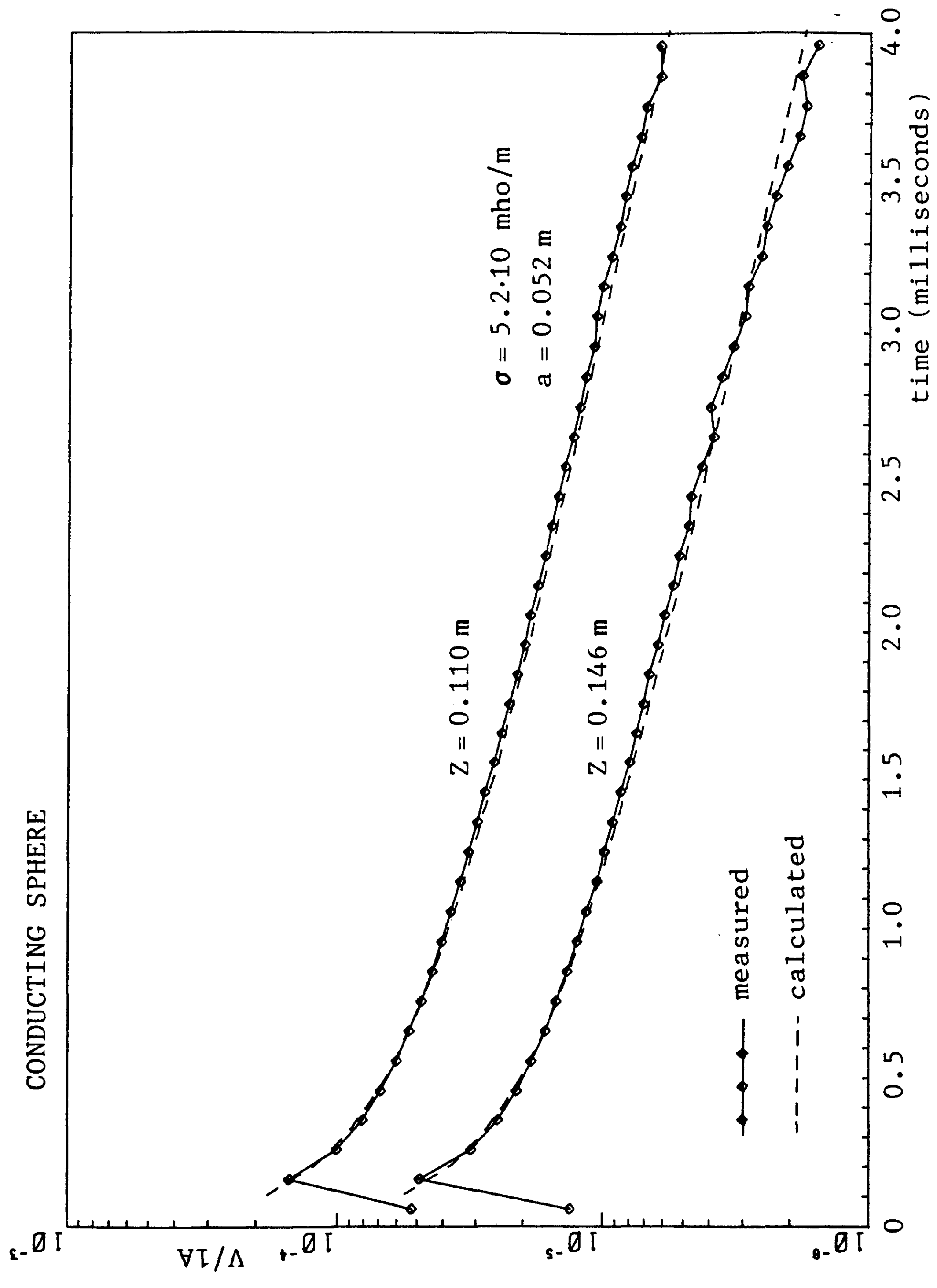


Figure 3.6 Results of time domain measurements

CHAPTER 4

EXAMPLES OF PHYSICAL MODELING

4.1 Conducting plate

4.1.1 Introduction

In the case of a confined ore body, the transient field decays exponentially at late time :

$$V(t) \propto e^{-\frac{t}{\tau_0}} \tag{4.1}$$

where V : electromotive force

$\tau_0 = \frac{1}{q_1 \alpha}$ (time constant)

q_1 : constant depends on the shape of the body

$\alpha = \frac{1}{\sigma \mu a^2}$

For mining or geothermal exploration, the time constant is very important in estimating the size and conductivity of the ore body or geothermal reservoir.

The q values for spheroids and elliptical cylinders were obtained for various major and minor axis ratio (Kaufman 1978), and shown in table 4.1.

Since the time constant of a conducting rectangular plate is not known, physical modeling experiments were performed, as explained in following sections.

a/b	1	2	4	8	1	32
spheroid	9.87	3.45	1.35	0.575	0.261	0.122
cylinder	5.78	2.16	0.889	0.396	0.185	
disk	5.5					

Table 4.1 Pole Q1 of spheroid, elliptical cylinder and disk

4.1.2 Time constant of conducting rectangular plate

To determine the time constants of the rectangular plate, the conductivities of the materials must at first be measured. Since the q_1 value of the circuler disk is known to be 5.5, by measureing the time constants of the disk, we can obtain the conductance of the disk :

$$S = \frac{5.5\tau_0}{\mu a} \tag{4.2}$$

where S : conductance (σh)

σ : conductivity of the disk material

h : thickness

a : radius of the disk

Therefore, by measuring the time constants of the disks made from the same materials as the rectangular plates, the conductivities can be obtained.

Figure 4.1 shows examples of the measured transient response of a conducting disk, of which radius is 5 cm and thickness is 0.953 cm, for various hight. From the slope of the data, the conductance S of the disk is calculated to be $1.33 \cdot 10^5$ mho, and the conductivity to be $1.4 \cdot 10^7$ mho/m.

Then, the time constants of rectangular plates of various shape were measured. Figure 4.2 shows the responses of various rectangular plates.

The q_1 values obtained as a function of the ratio a/b are shown in figure 4.3, where a and b are the length of the sides of the rectangular plate, and $a > b$.

The next step is to fit a curve to data to obtain an empirical formula. Linear or exponential regression methods are widely used in case of simple curve fitting, but for the obtained data, more sophisticated methods are needed. For this purpose, the Marquardt algorithm is widely used (Marquardt 1963). More recently, the Simplex method has been introduced, which can handle virtually any functions without divergence of the solutions (Nedler 1965, Caceci 1984). The function fitted by the Simplex method is shown as a continuous line in figure 4.3, and expressed as :

$$q_1 = -\frac{1.56}{r^3} + \frac{5.06}{r^2} - \frac{1.11}{r} + 8.02 \qquad (4.3)$$

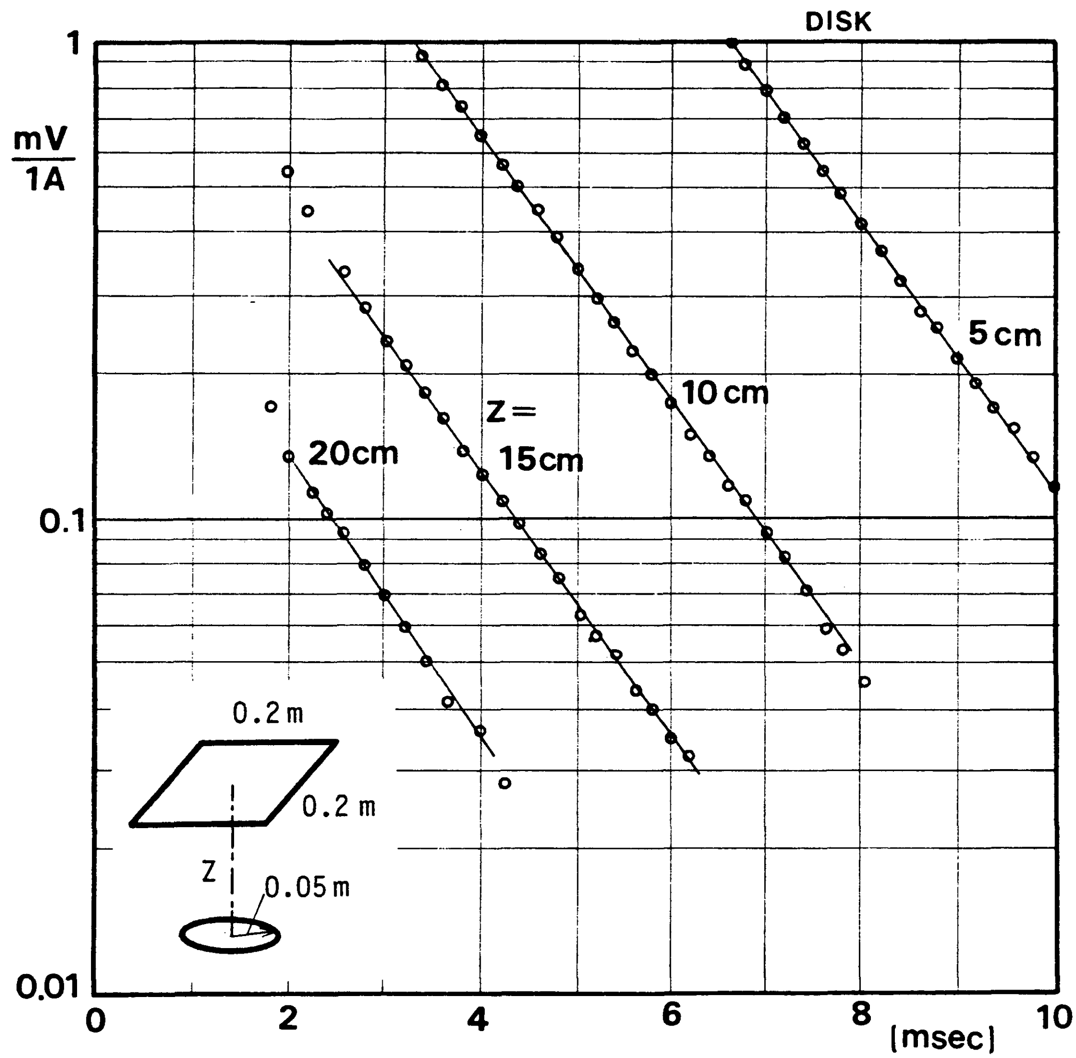


Figure 4.1 Transient response of conducting disk

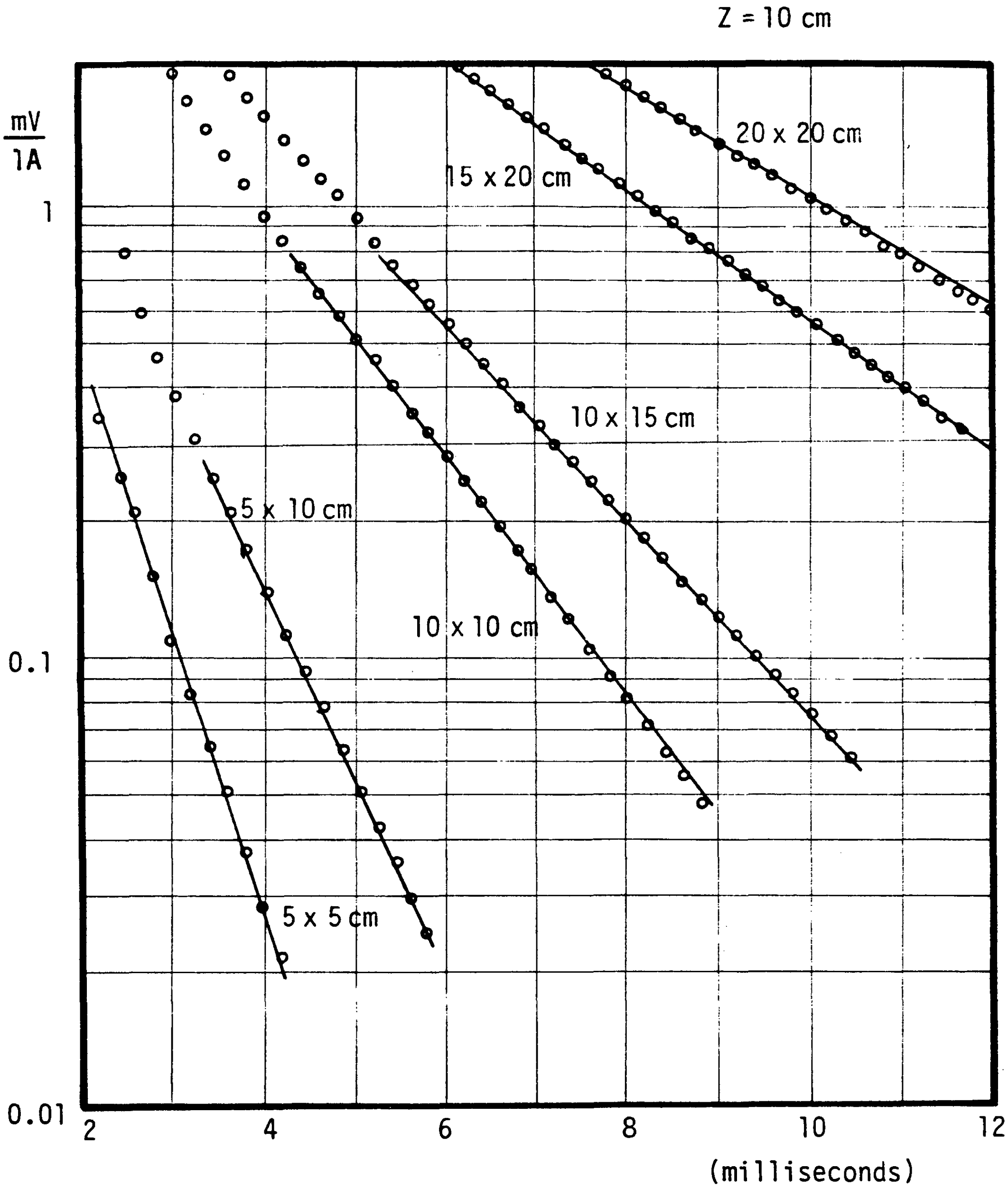


Figure 4.2 Transient response of rectangular plate

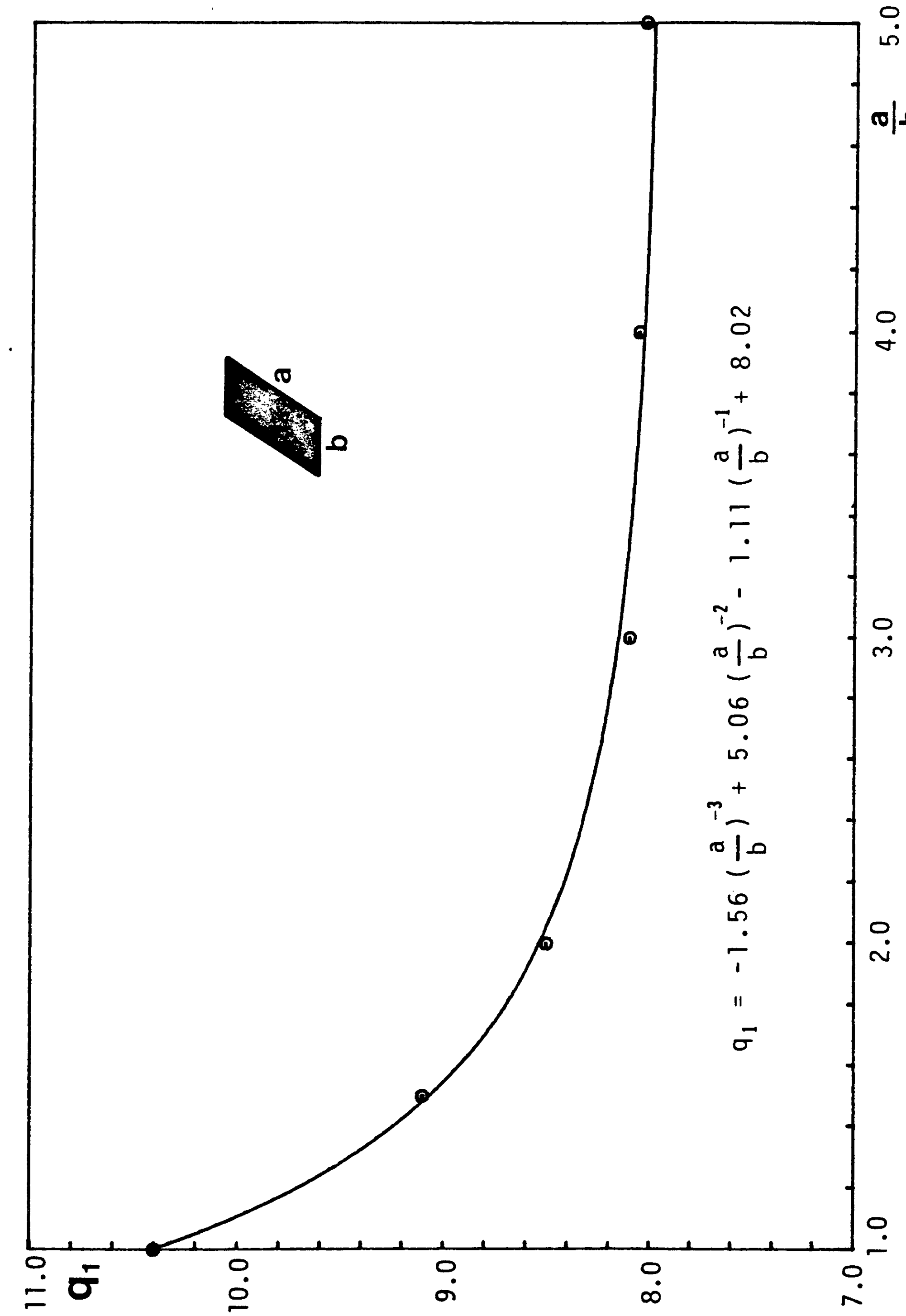


Figure 4.3 Q1 value as a function of a/b

And the time constant of the rectangular plate can be expressed as :

$$\tau_0 = \frac{\mu S\, b}{C_1 r^{-3} + C_2 r^{-2} + C_3 r^{-1} + C_4} \quad \text{(seconds)} \tag{4.4}$$

where $r = \frac{a}{b}$

$C_1 = -1.56$

$C_2 = 5.06$

$C_3 = -1.11$

$C_4 = 8.02$

4.1.3 Moment of conducting rectangular plate

At late time, induced currents in the rectangular plate flows uniformly. If the field due to the currents is measured far from the plate, it can be assumed to be a field due to a current loop of moment M_P located at the center of the plate. The voltage in the receiver coil due to this moment can be expressed as :

$$V(t) = \frac{\mu H_o M_P M_R}{Z^3} \frac{q_1}{\mu S b} e^{-\frac{q_1}{\mu S b} t} \tag{4.5}$$

where H_o : primary magnetic field at the center of the plate normal to the surface

M_R : moment of receiver coil

S : conductance of the plate

$$V(t) = \frac{\mu H_o M_p M_r}{Z^3} \frac{q_1}{\mu S b} e^{-\frac{q_1}{\mu S b} t}$$

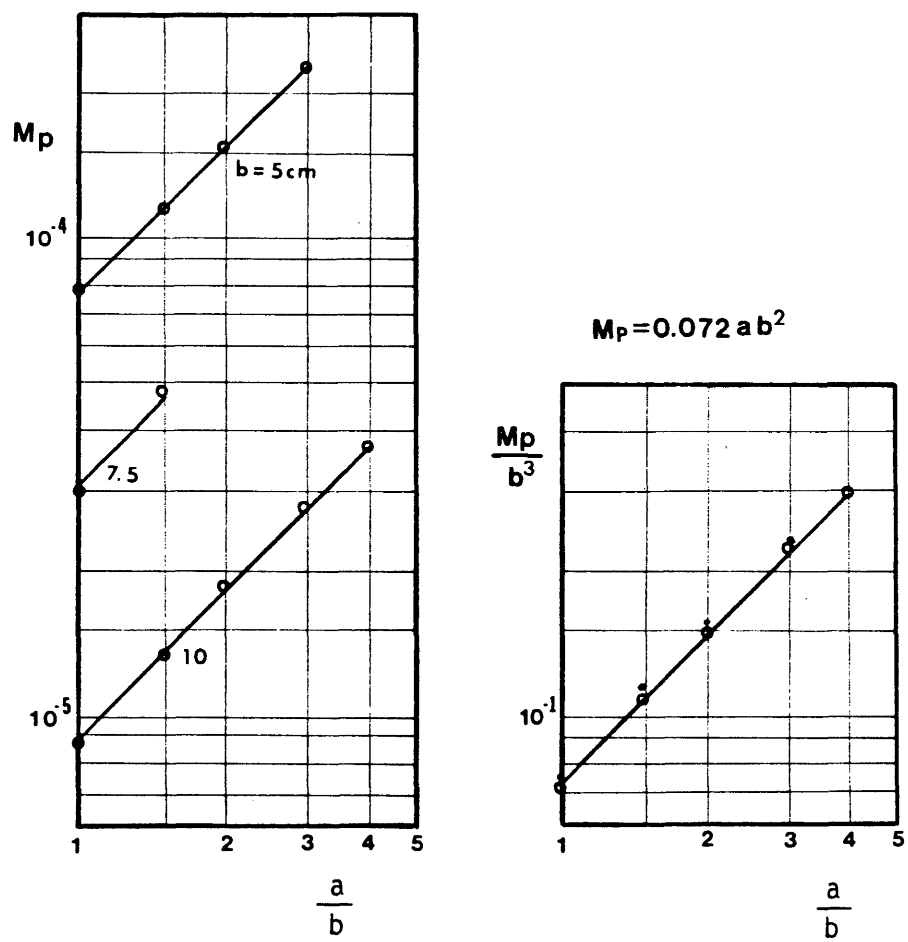


Figure 4.4 Moment of rectangular plate

b : length of smaller side of the plate

z : hight of the receiver coil

Then, from the experiments, the moment of the plate can be obtained as :

$$M_P = \frac{Z^3 V(t) \mu S b}{\mu H_o M_R q_1} e^{\frac{q_1}{\mu S b} t} \qquad (4.6)$$

Figure 4.4(a) shows the moment M_P values as a function of a/b. For each values of b, straight lines were obtaind. If M is normalized by b , all the data are expressed by only one line as shown in figure 4.4(b). Therefore, from this experiment, the moment of the rectangular plate M_P is expressed as :

$$M_P = 7.2 \cdot 10^{-2} a\ b \qquad 1 \leqq a/b \leqq 4 \qquad (4.7)$$

And if we substitute to equation 4.5, we have :

$$V(t) = \mu H_o \frac{7.2 \cdot 10^{-2} ab}{Z^3} M_R \frac{q_1}{\mu S b} e^{-\frac{q_1}{\mu S b} t} \qquad 1 \leqq a/b \leqq 4 \qquad (4.8)$$

Then, the empirical relation of voltage at the late time was obtained.

4.1.4 Transient profile of conducting plate

For mining exploration, the conducting plate model is one of the most widely used models. Some examples of

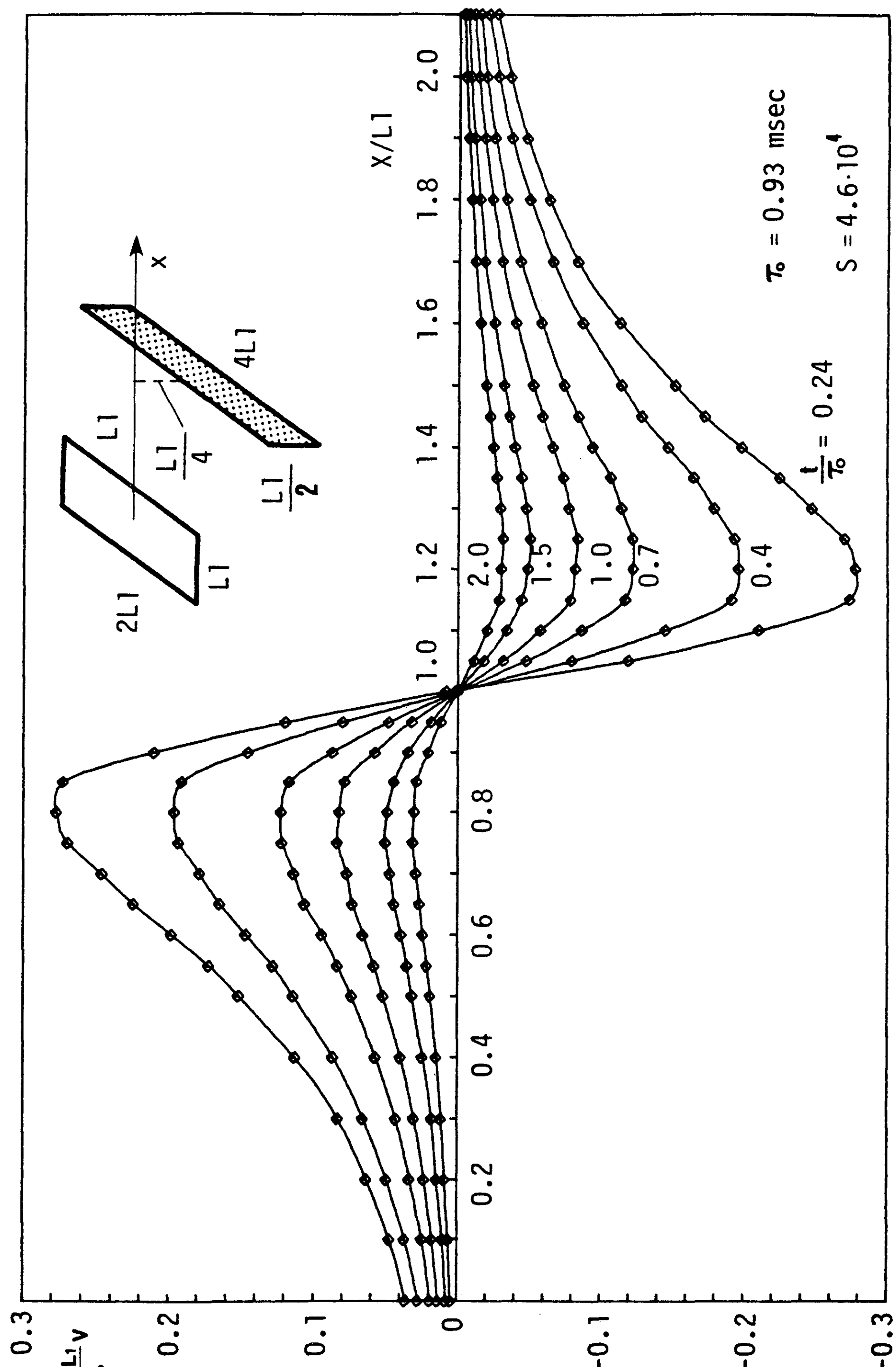


Figure 4.5 Vertical thin plate

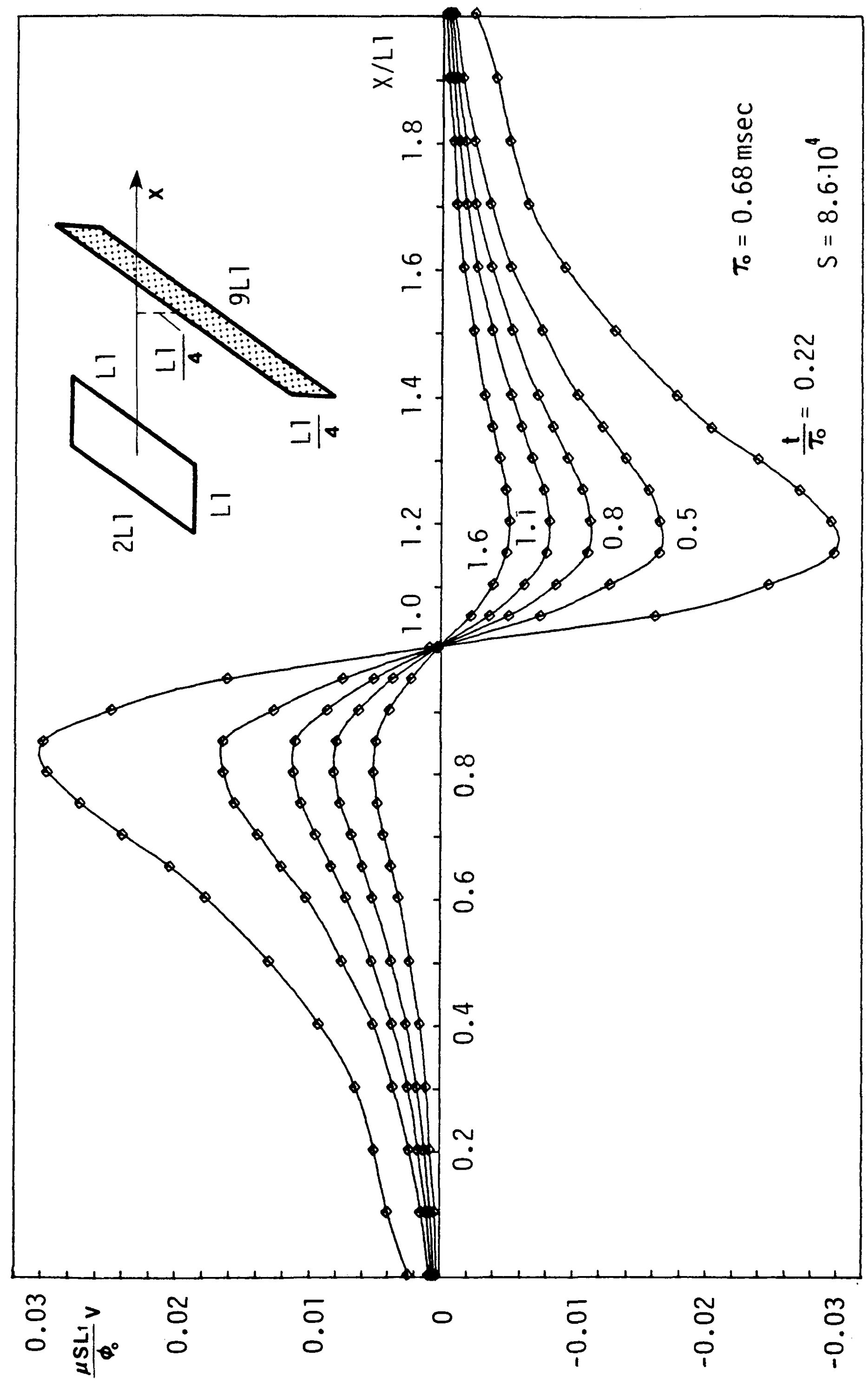


Figure 4.6 Vertical thin plate

vertical conducting plate modeling are shown in figure 4.5 and 4.6.

The obtained voltage was normalized to be a non-dimensional quantity as :

$$Y = V(t)\ \frac{\mu S L_1}{\phi_o}$$

where $\phi_o = S_o H_T$

S_o : effective area of receiver coil

H_T : primary magnetic field at the center of transmitter loop

L_1 : length of smaller side of the transmitter loop

In the case of a vertical thin plate, the polarity of the vertical component changes at the top of the plate, and the position of the plate can be determined easily.

4.1.5 Conducting thick plate

Sometimes, an ore body can be relatively thick. Figure 4.7 shows the profile of vertical thick plate measured by physical modeling. Curves are very similar to the curves for a thin plate, but at early time, reversal of the polarity can be observed. Figure 4.8 shows the measured voltage as a function of time, where sign reversal was indicated when

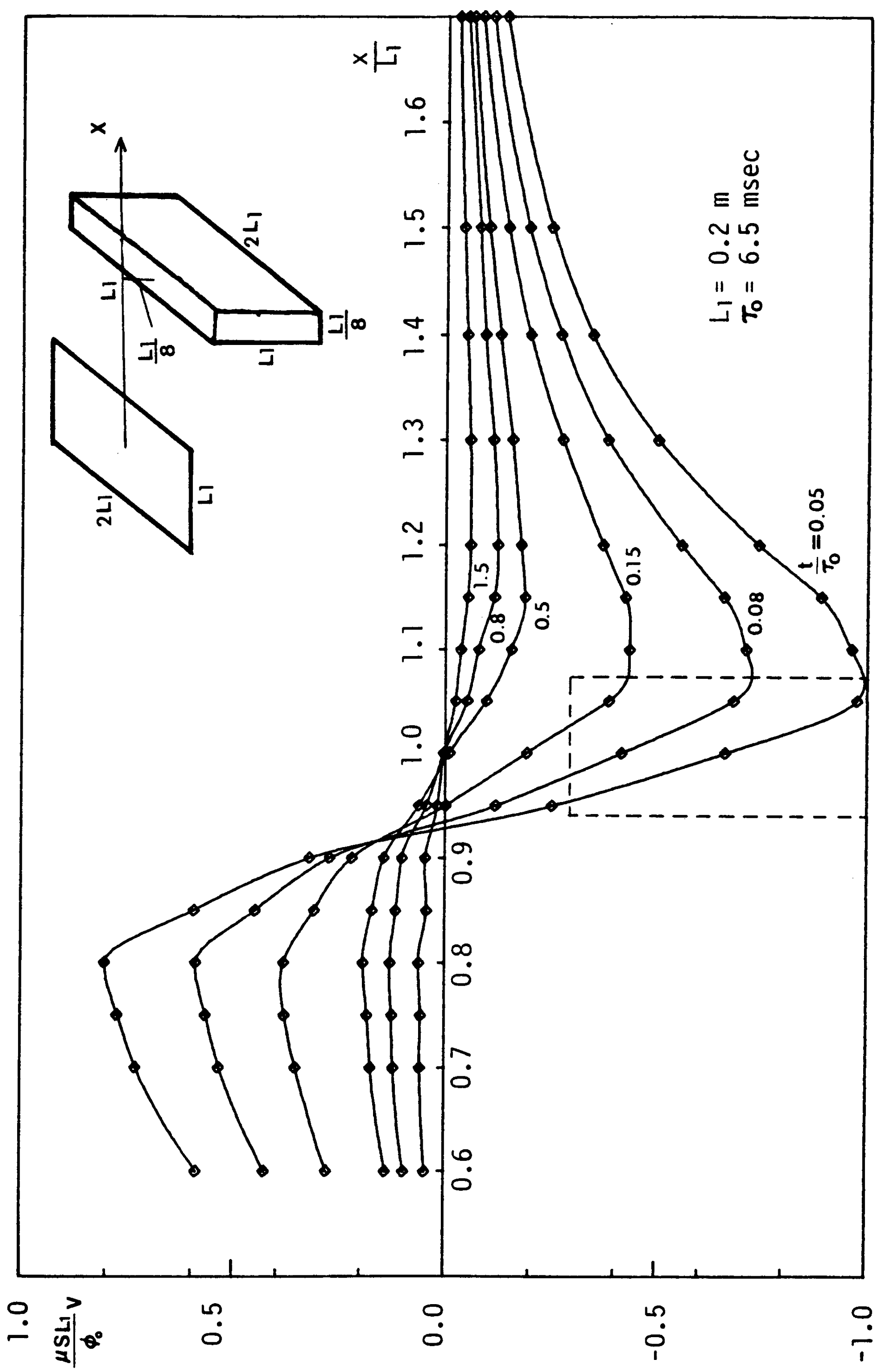


Figure 4.7 Conducting Thick Plate

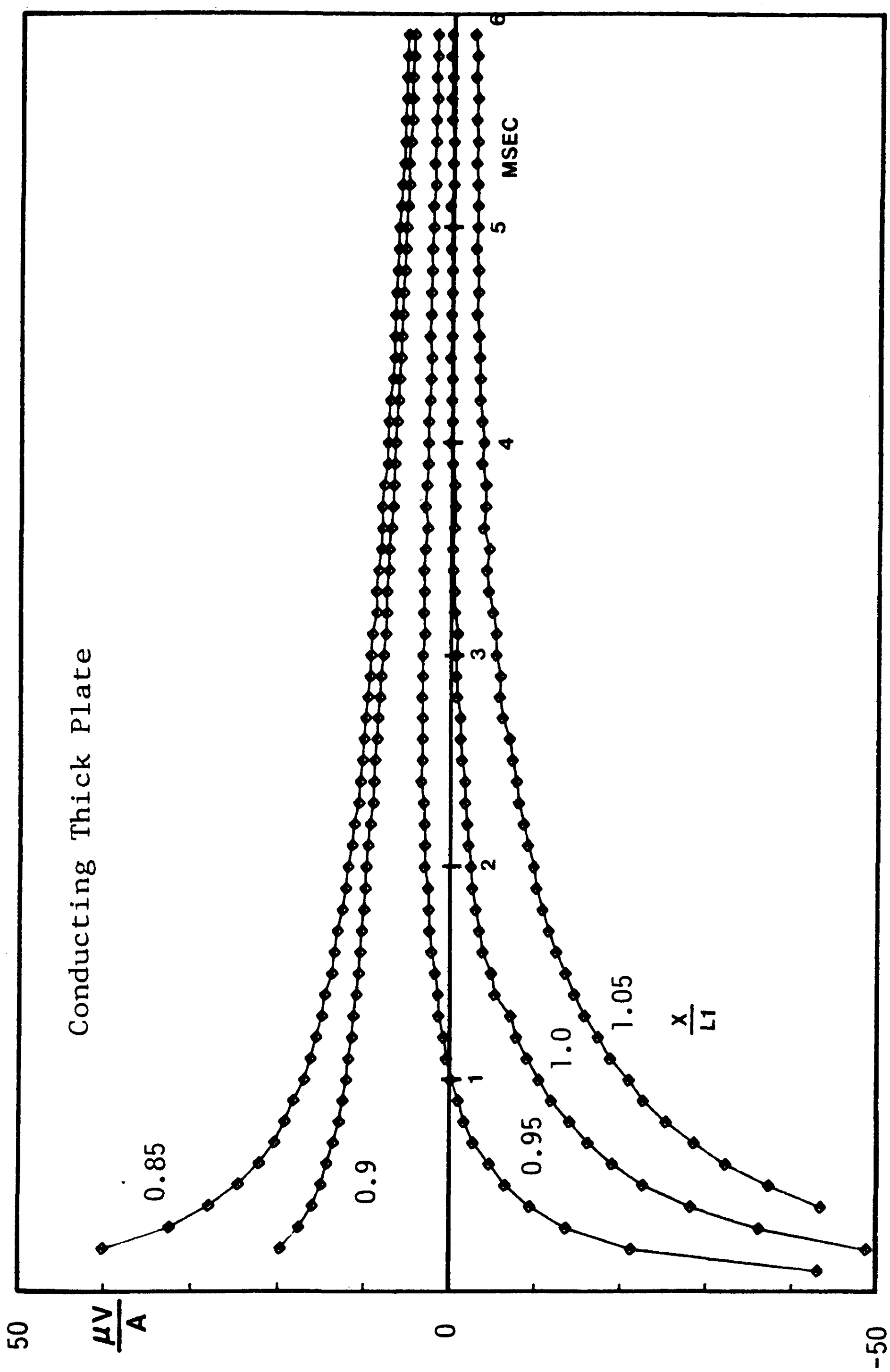


Figure 4.8 Transient response of conducting thick plate

X/L1 is 0.95.

The zero crossing of the profile (figure 4.8) coincides with the front surface of the thick plate at early time, and at late time, it is at the middle of the plate. The reason for this is that at early time, currents are on the surface of the orebody due to the skin effect, and at late time currents have diffused inside the plate and the zero crossing is shifted to the middle of the thick plate. From this shuft of the zero crossing, we can estimate the thickness of the plate.

Figure 4.9 shows the field example for mining exploration, and data were obtained by Geonics EM37 system. By considering the results of previous section, the ore bodies seemed to be two vertical plates, and by the physical modeling shown in figure 4.10, it was confirmed. But sign reversals were observed at the station 0.25N and 5.75S in figure 4.9. If we plot the data as a function of time for the stations around 0.25N (figure 4.11), we have the same behavior of sign reversal as in figure 4.8. Therefore, the ore bodies of figure 4.9 were estimated as the vertical thick plate, and the thickness of them were estimated to be about 100 m each by the shift of zero crossing.

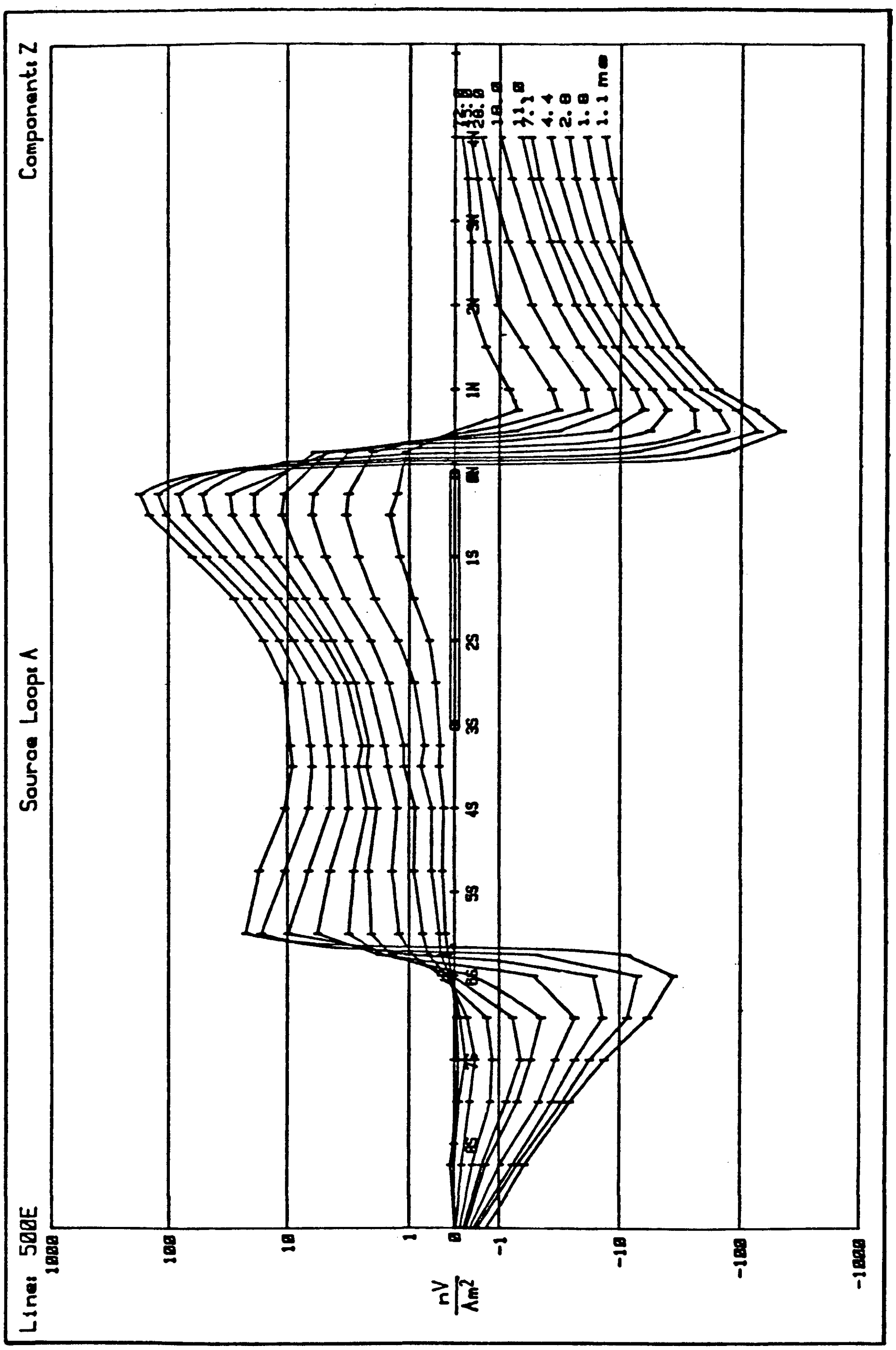


Figure 4.9 Field data

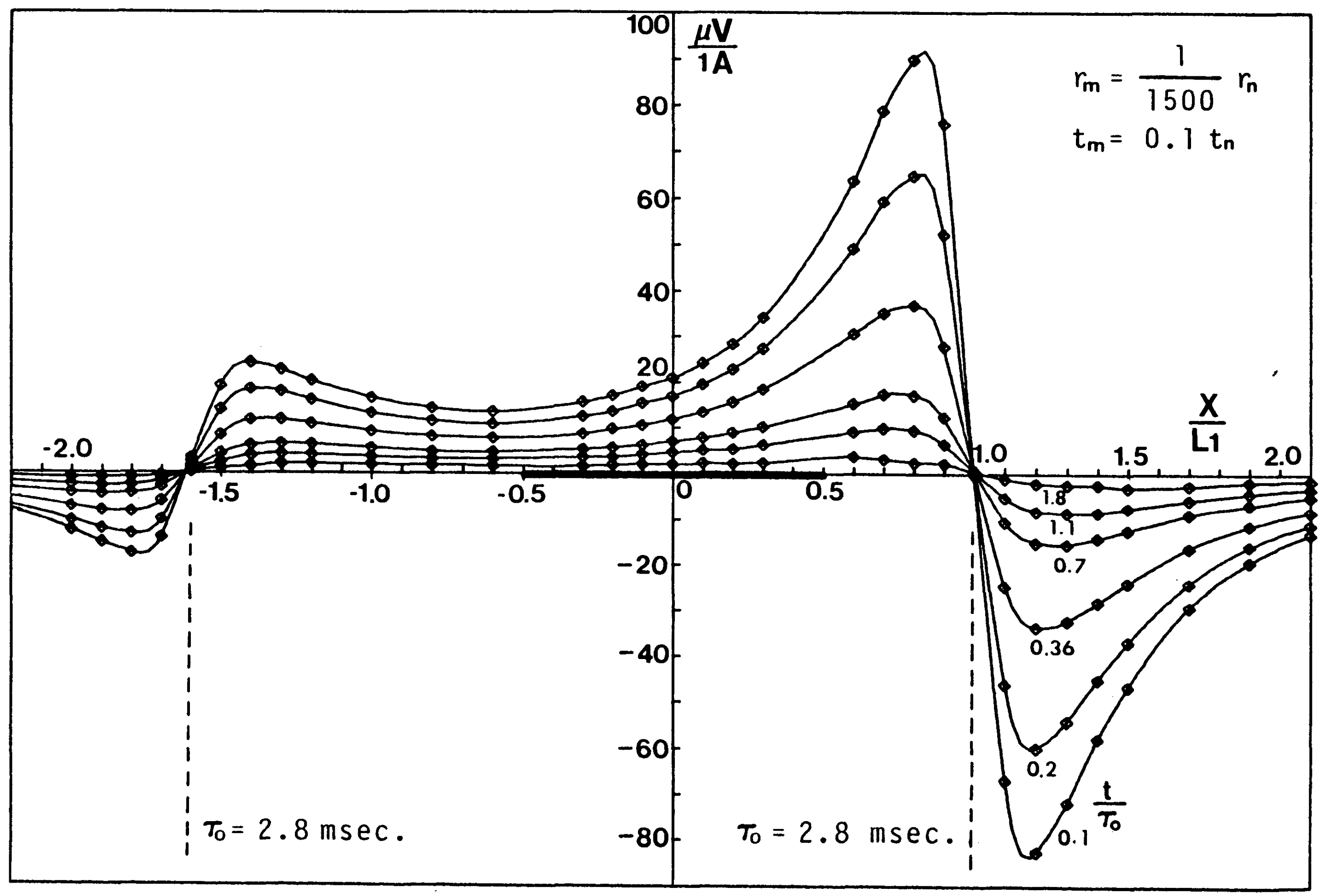


Figure 4.10 Physical modeling for two vertical plates

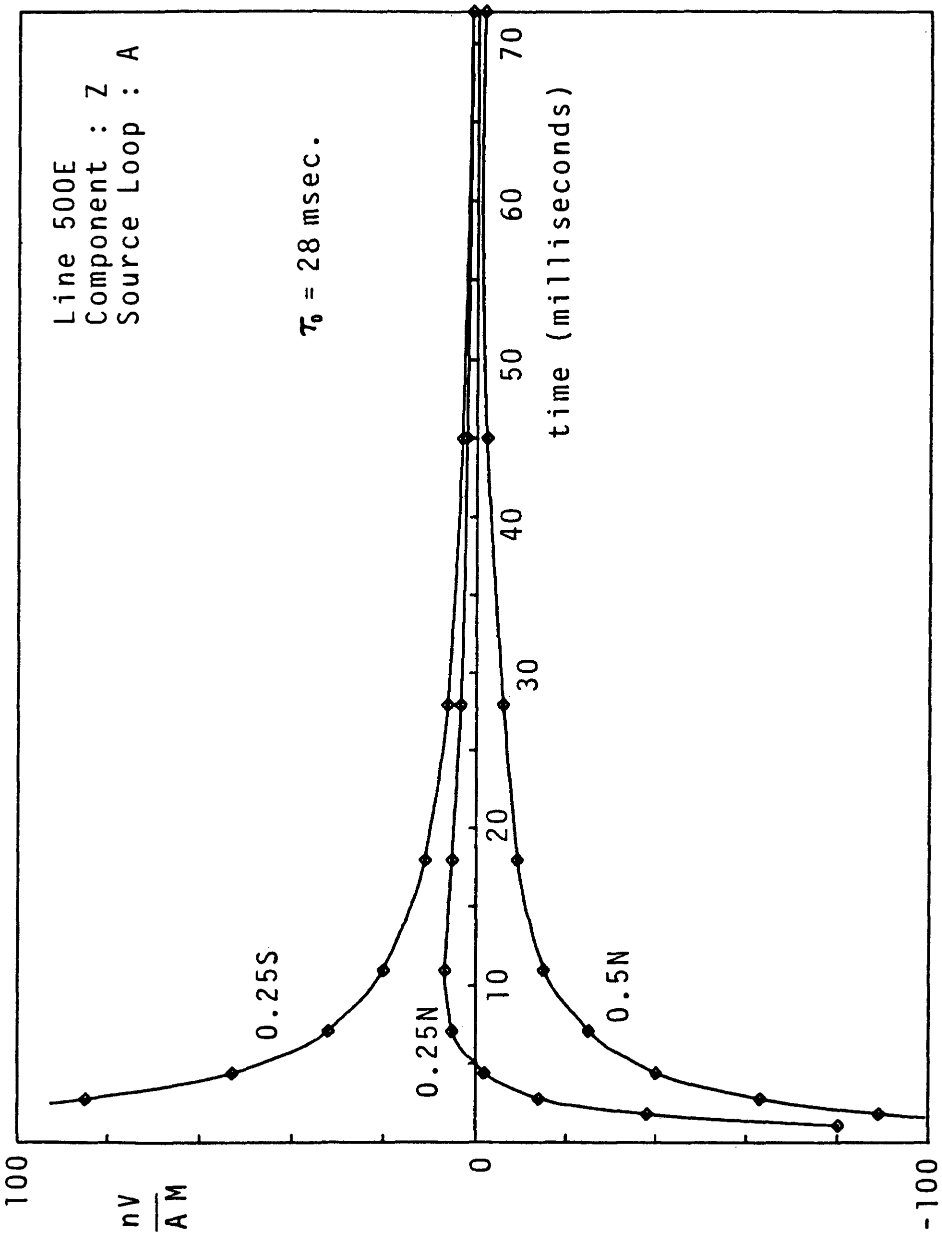


Figure 4.11 Transient response

CONCLUSIONS

An electromagnetic physical modeling system has been developed, and using the system some of the practical results were obtained for a rectangular thin plate and a thick plate.

The basic principle of physical modeling, called electromagnetic similitude, was derived from the well known equations for an electromagnetic field in a conducting medium. The following conditions must hold for nature and the model :

$$\sigma_n \mu_n \omega_n r_n^2 = \sigma_m \mu_m \omega_m r_m^2 \quad \text{(frequency domain)}$$

$$\frac{\sigma_n \mu_n r_n^2}{t_n} = \frac{\sigma_m \mu_m r_m^2}{t_m} \quad \text{(time domain)}$$

The relations of the voltages in receiver coils for nature and the model are :

$$\frac{H_n}{H_{on}} = \frac{H_m}{H_{om}} \quad \text{(frequency domain)}$$

$$V_n = V_m \frac{H_{on}}{H_{om}} \frac{\mu_n}{\mu_m} \frac{M_{R_n}}{M_{R_m}} \frac{t_m}{t_n} \quad \text{(time domain)}$$

Due to the progress in the integrated circuit technique, very capable and flexible physical modeling systems were developed. The ramp time of the inverter, which was developed by using a high voltage transistor with a zener diode protection circuit, was about 35 microseconds. Special high gain stable amplifier was designed by using OP-27 operational amplifier chips and ICL7650 chopper-stabilized amplifiers.

A data acquisition system was developed by using a Motorolla M6809 microprocessor and an Analog devices AD578 A/D converter chip. Maximum through put of 50 KHz (20 micro-seconds/sample) was established for 12 bit resolution.

By using the developed modeling system, the time constant of rectangular thin plate was measured and an empirical equation (equation 4.4) was obtained. If a field is measured far from the plate, the plate can be assumed as a dipole, and the empirical equation for the moment was obtained as :

$$M_P = 7.2 \cdot 10^{-2} a \cdot b^2 \qquad 1 \leqq \frac{a}{b} \leqq 4$$

The empirical equation to calculate the electromotive force was then obtained as equation 4.8.

The transient response of a conducting thick plate was also measured, where the reversal of the polarity of the measured voltage was observed.

REFERENCES

Adby, P.R. (1974), Introduction to Optimization Method, pp 42-73, Chapman and Hall, London.

Allaud, Louis A. (1977), SCHLUMBERGER - The history of a Technique, John Wiley & Sons, New York.

Analog Devices (1982), Data acquisition databook, Analog Devices, Inc, Norwood, Massachusetts.

Annan, A.P. (1974), The equivalent Source Method for Electromagnetic Scattering Analysis and its Geophysical Application, Ph.D. Thesis, Memorial University of Newfoundland.

Artwick, Brace A. (1980), Microcomputer Interfacing, pp 74-77, Prentice-Hall, Englewood Cliffs, N.J.

Barringer, A.R. (1962), New approach to exploration - the input airborne electrical pulse prospecting system, Mining Congress Journal, Vol 48, pp 49-52.

Becker,A. et al. (1972), Scale model study of time domain electromagnetic response of tabular conductors, Canadian Mining and Metallurgical Journal, Vol 65, pp 90-96.

Berlin, Howard M. (1977), Design of active filters with experiments, pp 55-64, 175-189. Howard W. Sams & Co., Indianapolis, Indiana.

Boyd, D., Roberts, B.C. (1961), Model experiments and survey results from a wing-tip-mounted electromagnetic prospecting system, Geophysical Prospecting, Vol 9-3, pp 411-420.

Box, George E. (1978), Statistics for Experimenters - An Introduction to Design, Data Analysis, and Model Building, pp 510-555, John Wiley & Sons, New York.

Brigham, E.Oran (1974), The fast Fourier transform, pp 75-109, Prentice-Hall, Englewood Cliff, N.J.

Brown, H.G., and King, R.(1934), High-frequency models in Antena Investigations, Proceedings of the Institute of Radio Engineers, Vol 22-4, pp 457-480.

Brubaker, D.G. (1957), Apparatus and Procedure for Electromagnetic Prospecting, Trans. A.I.M.E., pp 777-780.

Bruckshaw, J.M. (1936), Experiments on conducting laminae in periodic magnetic fields, Proc. Physical Society of London, Vol 48, pp 63-74.

Burr-Brown (1982), Burr-Brown product data book, Burr-Brown Research Corp., Tucson, Arizona.

Conklin, H.R. (1917), Prospecting with Electricity, Engineering and Mining Journal, Vol 104-8, pp 339-340.

Doll, H.G. (1949), Introduction to Induction Logging and Application to Logging of Wells Drilled with Oil Base Mud, J. Petrol. Technol., June, (AIME) TP2641, pp 142-162.

Doll, H.G, and Martin M. (1950), Recent developments in electrical logging and auxiliary methods, Quarterly of Colorado School of Mines, pp 49-78, Golden, Colorado.

Dosso, H.W. (1966), Analogue model measurements for electromagnetic variations near vertical faults and dykes, Canadian J. of Earth Science, Vol 3, pp 287-303.

Douloff Artel A. (1961), The response of a disk in a dipole field, Geophysics, Vol 26-4, pp 452-464.

Einstein, P.A. (1951), Factors limiting the accuracy of the electolytic modelling tanks, British J. Applied Physics, Vol 2, pp 49-55.

Eve, A.S. (1932), Absorption of Electromagnetic Induction and Radiation by Rocks, Trans A.I.M.E., pp 160-168.

Focken, Charles M.(1937), The Sundberg Inductive Method of Electrical Prospecting, Quarterly of the Colorado School of Mines, pp 225-252, Golden Colorado.

Fraser D.C., and Ward S.H. (1967), Analytic and model studies of a rotatable field electromagnetic prospecting system, Geophysics, Vol 32-5, pp 899-917.

Frischknecht, F.C. (1971), Electromagnetic Scale Modeling, In :J.R. Wait(Editor),Electromagnetic Probing in Geophysics, Golem Press, pp 265-320, Boulder, Colorado.

Fuller, B.D. (1971), Electromagnetic Response of a Conductive Sphere Surrounded by a Conductive Shell, Geophysics, Vol 36-1, pp 9-24.

Goldman, M.M., and Stoyer, C.H. (1983), Finite-difference Calculations of the Transient Field of an Axially Symmetric Earth for Vertical Magnetic Dipole Excitation, Geophysics, Vol 48-7, pp 953-963.

Goldman, M.M. (1983), The Integral-finite-difference Method for Calculating Transient Electromagnetic Fields in a Horizontally Stratified Medium, Geophysical Prospecting, Vol 31, 664-686.

Harrington, Steven (1983), Computer Graphics, pp 11-16, McGraw-Hill, New York.

Hedstrom, Helmer (1937), Phase Measurements in Electrical Prospecting, Trans. A.I.M.E., Vol 138, pp 456-472.

Hedstrom, E.H., Parasnis, D.S.(1958), Some model experiments relating to electromagnetic prospecting with special refrence to airborne work, Geophysical Prospecting, Vol 6-4, pp 322-341.

Heiland, C.A. (1929), Geophysical Methods of Prospecting - Principles and recent successes, Quarterly of the Colorado School of Mines, Vol 24-1, pp 99-129, Golden, Colorado.

Heiland, C.A. (1938), Electrical Prospecting, Colorado School of Mines, pp 20-31, Golden, Colorado.

Hermance, John F. (1968), Model studies of the coast effect on geomagnetic variations, Canadian Journal of Earth Science Vol 5-3, pp 515-522.

Hohmann, G.W. (1975), Three-dimensional Induced Polarization and Electromagnetic Modeling, Geophysics, Vol 40-2, pp 309-324.

Inman, Joseph R., et al.(1973), Resistivity Inversion, Geophysics, Vol 38-6, pp 1088-1108.

Intersil (1982), Intersil Catalog 1982, Intersil Inc., Cupertino, California.

Jakosky, J.J. (1928), Operating Principles of Inductive Geophysical Process, A.I.M.E. TP-134, pp 1-39.

Johnk, Carl T.A. (1975), Engineering Electromagnetic Field and Waves, pp 311-336, John Wiley & Sons, New York.

Kaufman, A.A., and Shapiro, R.P. (1959), Operation of a ring induction logging integrator, Geologiya i Razbedka, NO 10.

Kaufman, A.A. (1978), Frequency and Transient Responses of Electromagnetic Fields Created by Currents in Confined Conductors, Geophysics, Vol 43-5, pp 1002-1010.

Kaufman, A.A. (1978), Resolving Capabilities of the Inductive Methods of Electroprospecting, Geophysics, Vol 43-7, pp 1392-1398.

Kaufman, A.A. (1979), Harmonic and Transient Fields on the Surface of a Two-layer Medium, Geophysics, Vol 44-7, pp 1208-1217.

Kaufman, A.A., Keller, G.V. (1981), The Magnetotelluric Sounding Method, pp 157-174, Elsevier, Amsterdam.

Kaufman, A.A., Keller, G.V. (1983), Frequency and Transient Soundings, pp 315-450, Elsevier, Amsterdam.

Keller, G.V., and Frischknecht, F.C. (1966), Electrical Method in Geophysical Prospecting, Pergamon Press, New York.

Kelly, Sherwin F. (1950), The Rise of Geophysics, Canadian Mining Journal, July Vol 71, pp 47-53.

Kelly, Sherwin F. (1950), Spontaneous polarization, or self-potential method, in Method and case histories in mining geophysics, Canadian Institute of Mining and Metallurgy.

Ketola, Matti, and Puranen M. (1967), Type curves for the interpretation of slingram (horizontal loop) anomalies over tabular bodies, Geological Survey Finland Rep. Invest 1.

Koefoed, O., and Struyk, A.P. (1969), The electrical current pattern induced by an oscillating magnetic dipole in a semi infinite conductive thin plate, Geophysical Prospecting, Vol 17-2, pp 182-195.

Koenigsberger, J.G. (1937-39), Elektrische Vertikal-sondierung von der Erdoberflache aus mit der Zentral-induktionsmethode, Beitruge Zur Geophysik, Vol 7, pp 112-161

Lowrie, W., West, G.F. (1965), The effect of a conducting overburden on electromagnetic prospecting measurements, Geophysics, Vol 30-4, pp 624-632.

Lundberg, Hans (1950), Current trend and progress in mining geophysics, Quarterly Colorado School of Mines, 75th Aniv. pp 41-47, Golden, Colorado.

Marquardt,Donald W.(1963), An algorithm for least-square estimation of nonlinear parameters, Journal for the Society of Industrial and Applied Mathematics, Vol 11-2, pp 431-441.

Mueser, E.E. (1926), The Elbof Method of Electrical Prospecting, Canadian Institute of Mining and Met. Bulletain, pp 1148-1154.

Nabighian, Misac N. (1984), Time-Domain Electromagnetic Methods of Exploration, Geophysics, Vol 49-7, pp 849-853.

Nair, S.K. et al. (1968), Experimental studies on the electromagnetic response of tilted conducting half-planes to a horizontal-loop prospecting system, Geoexploration, Vol 6, pp 207-244.

Nedler, J.A., and R. Mead (1965), A Simplex Method for Function Minimization, Computer Journal, Vol 7, pp 308.

Negi, Janardan G., Gupta, Chandra P. (1968), Models in applied geo-electromagnetics, Earth Science Review, Vol 4-3, pp 219-241.

Newman, William M., and Sproull, Robert F. (1979), Principles of interactive computer graphics, pp 20-27, McGraw-Hill, New York.

Ogilvy, R.D. (1983), A model study of the transient electromagnetic coincident loop technique, Geoexploration, Vol 21-4 pp 231-264.

Parkhomenko, E.I. (1967), Electrical Properties of Rocks, Translated from Russian and edited by Dr. G.V. Keller, Plenum Press, pp 59-184, New York.

Paterson, Norman R. (1961), Experimantal and field data for the dual-frequency phaseshift method of airborne electromagnetic prospecting, Geophysics, Vol 26-5, pp 601-617.

Patra, H.P., Mallick, K. (1980), Geosounding Principles 2 (Time-Varying Geoelectric Soundings), Elsevier, pp 184-189, Amsterdam, Netherlands.

Poddar, M., Bhattacharya P.K. (1966), On the Electromagnetic Response of Conductor in the inductive method of Prospecting, Geophysical Prospecting, Vol 14-4, pp 470-486.

Pritchett,W.C. (1955), A low frequency electrical earth model, Geophysics, Vol 20-4, pp 860-870.

Ramaprasada Rao, I.B., Bhimasankaran, V.L.S. (1973), Electromagnetic Modelling of Sheet-like Bodies by the Combined-loop Version of the Transient Pulse Induction Method, Geoexploration, Vol 11, pp 87-95.

Ruckdeschel F.R. (1981), BASIC Scientific Subroutines Vol 2, pp 305-310, McGraw-Hill, New York.

Saito, A., and Ohya, T. (1984), Differences of timedomain measurement and frequency domain measurement for electromagnetic prospecting methods, S.E.G.Japan meeting abstract (1984 spring), pp 131-132.

Saito, A., Sassa, K., and Yoshizumi, E.(1977), Central Induction Vertical Sounding Method, S.E.G.Japan 37th meeting abstract, pp 52.

Sinclair, G. (1948), Theory of Models of Electromagnetic Systems, Proceedings of the I.R.E., Vol 36-11, pp 1364-1370.

Slichter, L.B. (1932), Observed and Theoretical Electromagnetic Model Response of Conducting Spheres, Transaction A.I.M.E. 97, pp 443-459.

Smythe, William R.(1968), Static and Dynamic Electricity, Third Edition, pp 368-408, McGraw-Hill, New York.

Spies, Brian (1976), The transient electromagnetic method in Australia, BMR Journal of Australian Geology and Geophysics, Vol 1, pp 23-32.

Spies, Brian (1976), The derivation of absolute units in electromagnetic scale modeling, Geophysics, Vol 41-5, pp 1042-1047.

Spies, Brian (1980), A fielf occurrence of sign reversals with the transient electromagnetic method, Geophysical Prospecting, Vol 28, pp 620-632.

Strangway, David W. (1966), Electromagnetic Scale Modeling, in Methods and techniques in geophysics ed. by S.K. Runcorn, Interscience Publishers, London.

Stratton, J.A. (1941), Electromagnetic Theory, pp 488-490, McGraw-Hill, New York.

Sweet, George Elliott, (1956), The history of Geophysical Prospecting, Everette Lee De Golyer Memorial Edition, Science Press, Los Angeles.

Swanson, H.E. (1961), Model studies of an apparatus for electromagnetic prospecting, Trans.A.I.M.E. 220, pp 234-238.

Verma, S.K., and Singh, R.N. (1970), Transient electromagnetic Response of an Inhomogeneous Conducting Sphere, Geophysics, vol 35-2, pp 331-336.

Ward, S.H., Gredhill, T.R. (1957), Electromagnetic Surveying - Ground methods, in Method and case histories in mining geophysics, Canadian Institute of Mining and Metallurgy.

Ward, S.H. (1957),Airborne Electromagnetic Surveying, in Method and case histories in mining geophysics, Canadian Institute of Mining and Metallurgy.

Ward, S.H. (1980), Electrical, electromagnetic, and magnetotelluric methods, Geophysics, Vol 45-11, PP 1659-1666.

Wait, J.R. (1969), Electromagnetic Inductions in a Solid Conducting Sphere enclosed by a Thin Conducting Spherical Shell, Geophysics, Vol 34-5, pp 753-759.

Wilson John H. (1928), Geophysical Prospecting, Colorado School of Mines Magazine, Golden Colorado.

Yost, W.J. (1952), The Interpretation of Electromagnetic Reflection Data in Geophysical Exploration - Part 1 General Theory, Geophysics, Vol 17-1, pp 89-106.

Yost, W.J., Caldwell, and R.L., et al.(1953), The interpretation of Electromagnetic reflection Data in Geophysical Exploration - Part 2 Metallic Model Experiments, Geophysics, Vol 17-4, pp 806-826.

Zuschlag, T. (1932), Mapping Oil Structures by the Sundberg Method, Trans. A.I.M.E., pp 144-159.

APPENDIX A

Electromagnetic similitude

Electromagnetic similitude explained in Chapter 1.2 can be derived by following Maxwell's equations :

$$\text{curl } \overline{E}_n = -\mu_n \frac{\partial \overline{H}_n}{\partial t_n} \tag{A.1}$$

$$\text{curl } \overline{H}_n = \sigma_n \overline{E}_n + \epsilon_n \frac{\partial \overline{E}_n}{\partial t_n} \tag{A.2}$$

$$\text{div } \overline{E}_n = \delta \tag{A.3}$$

$$\text{div } \overline{H}_n = 0 \tag{A.4}$$

where $\overline{E}$: electric field (V/m)

$\overline{H}$: magnetic field (A/m)

μ : magnetic permeability (H/m)

σ : conductivity (mho/m)

ϵ : dielectric constant (F/m)

and subscript n for nature.

Then the geometrical scale factor p can be introduced as follows:

$$r_n = p r_m \tag{A.5}$$

where subscript m is for model

r_m : Characteristic length for model

r_n : characteristic length for nature

In order to keep Maxwell's equations invariant under the scaling of the length above, it is necessary to scale $\overline{E}$, $\overline{H}$ and t in general. Then, we have :

$$\overline{E}_n = a\overline{E}_m \tag{A.6}$$

$$\overline{H}_n = b\overline{H}_m \tag{A.7}$$

$$t_n = ct_m \tag{A.8}$$

where a,b and c are scale factors for electrical field, magnetic field and time respectively. Then equations (A.1) and (A.2) can be expressed for the model as follows :

$$\text{curl } \overline{E}_m = -\mu_m \frac{\partial \overline{H}_m}{\partial t_m} \tag{A.9}$$

$$\text{curl } H_m = \sigma_m \overline{E}_m + \epsilon_m \frac{\partial \overline{E}_m}{\partial t_m} \tag{A.10}$$

Then if we substitute equations (A.6) through (A.8) to equations (A.9) and (A.10) :

$$\text{curl } \overline{E}_m = \frac{p}{a} \text{ curl } \overline{E}_n \tag{A.11}$$

$$\text{curl } \overline{H}_m = \frac{p}{b} \text{ curl } \overline{H}_n \tag{A.12}$$

And from equations (A.6) and (A.8), we have :

$$\frac{\partial \bar{E}_m}{\partial t_m} = \frac{\partial \bar{E}_m}{\partial t_n}\frac{\partial t_n}{\partial t_m} = \frac{c}{a}\frac{\partial \bar{E}_n}{\partial t_n} \quad (A.13)$$

and equations (A.7) and (A.8) yield :

$$\frac{\partial H_m}{\partial t_m} = \frac{\partial H_m}{\partial t_n}\frac{\partial t_n}{\partial t_m} = \frac{c}{b}\frac{\partial H_n}{\partial t_n} \quad (A.14)$$

Substituting equations (A.11) through (A.14) to (A.9) and (A.10) yields :

$$\frac{p}{a}\text{ curl } E_n = -\mu_m \frac{c}{b}\frac{\partial H_n}{\partial t_n} \quad (A.15)$$

$$\frac{p}{b}\text{ curl } H_n = \sigma_n \frac{E_n}{a} + \frac{c}{a}\epsilon_m \frac{\partial E_n}{\partial t_n} \quad (A.16)$$

These equations must be equivalent to equations (A.1) and (A.2), then following conditions can be obtained :

$$\sigma_m = \frac{pa}{b}\sigma_n \quad (A.17)$$

$$\epsilon_m = \frac{pa}{bc}\epsilon_n \quad (A.18)$$

$$\mu_m = \frac{pb}{ac}\mu_n \quad (A.19)$$

The fields can be simulated precisely through scale model by selecting a,b,c and p to satisfy all three relations above at the same time. Theoretically, a model can be constructed through arbitrary choice of scale factors a, b, c and p, but in practice it may not be possible to obtain materials of required electrical properties.

In most geophysical problems, dielectric constant can be neglected, therefore, equation (A.18) can be neglected.

And from equation (A.8), we have :

$$\omega_m = c\,\omega_n \qquad (A.20)$$

By eliminating the ratio a/b from equations (A.17) and (A.19), we have

$$\sigma_m \mu_m \omega_m = p^2 \sigma_n \mu_n \omega_n \qquad (A.21)$$

And using the characteristic length r, we have the following relations for frequency and time domain :

$$\sigma_m \mu_m \omega_m r_m^2 = \sigma_n \mu_n \omega_n r_n^2 \qquad (A.22)$$

$$\frac{\sigma_m \mu_m r_m^2}{t_m} = \frac{\sigma_n \mu_n r_n^2}{t_n} \qquad (A.23)$$

These relations are widely used for physical modeling, but since we neglected scale factors of electrical and magnetic fields a and b, electrical and magnetic field intensity can not be simulated. Sinclair(1948) called this model as geometric model, and the model which satisfy equations (A.17) through (A.19) is called absolute model.

APPENDIX B

DATA ACQUISITION PROGRAM (ADPLT6)

```
{******************************************************
 *Graphic plotting program for MX-80 with graphics opt. *
 *     using Omegasoft Pascal compiler                  *
 ******************************************************}
{*12bit A/D conversion program NO TRIGGER CA2 IS USED TO CONTROL TRANSMITTER }
{ DIRECTION OF CURRENT REVERSED, 2'S COMPLIMENT DATA, USE WITH ADSUB6.RO AND SUB1.RO }
{$B+}
Program ADPLT6(input,output,Auxout);
{$B-}
{ const and variables for plotting subprograms }
  const Home = $8000; { 16k for plot}
        Xmax = 300 ; {number of dots for x axis}
        Ymax = 240 ; {number of dots for y axis}
TYPE ARY=ARRAY[1..1800] OF REAL ;
  var   I,K,Hmax,x,y,flag : integer;
        addr,data,hdata : hex ;
        AX,BX,AY,BY,XR1,XR2,YR1,YR2,XAmin,XAmax,YAmin,YAmax : real;
{ variables for main program }
        a,b : real;
         FL1 :INTEGER;
         FL2,flag1 :CHAR;
         XX,YY,XX1,XX2,YY1,YY2,sum : REAL;
         V : ARRAY[1..1800] OF INTEGER at $A000 ;
         AV,VSQR : ARY ;
         Hstack,channel,CLKCHN : hex ;
         Nstack : integer ;
         Nsample,dt,II,KK : integer;
         C,unit  : char;
         Hclok,Hsample,clkrate : hex ;
         interval,Nclok,I1,K1,LL,I9 : integer ;
         rint,gain : real ;
         name : text;
         fname,title,comment : string ;
         auto:boolean;
         current,ctimesg,min,max,Maxtime : real ;
{ assembler routines and should be linked with }
Procedure ORPOKE(addr,data : hex);external ;
Procedure POKE(addr,data :hex);external ;
Function PEEK(addr:hex):hex;external ;
FUNCTION CREAD:CHAR;external;
Procedure ADCON6(Hsample,Hclok,CLKCHN : hex);external ;
Procedure PSET; external ;
```

```
Procedure PAUSE; external ; { pause on at subl.ro }
Procedure NOPAUS; external;
{***************** PLTOUT ****** print out results,  called at the end ***********}
Procedure PLTOUT;
  var n1,n2 : integer ;
  BEGIN
  if Xmax >=256  then begin
                       n1 := Xmax-256; n2 := 1
                     end
              else begin
                       n1 := Xmax; n2 := Ø
                     end;
     writeln(Auxout);writeln(Auxout); write(Auxout,chr(27),'A',chr(8));
     addr := Home ;
     for I := 1 to (Ymax div 8) do
     begin
       write(Auxout,chr(27),chr(75),chr(n1),chr(n2));
       for K := 1 to Xmax do
       begin
          data := PEEK(addr); write(Auxout,chr(data)); addr := addr + $1
       end;{of K loop}
       writeln(Auxout)
     end; {of I loop}
     write(Auxout,chr($1B),chr($32)); {set to 6line/inch}
     writeln(Auxout);writeln(Auxout)
  END;  {of Procedure PLTOUT}
{***************************** PLTINI  initialize memory, called beginning ******}
Procedure PLTINI;
  Begin
    data := $Ø ; Hmax := Ymax div 8 ; addr := Home ; K := Hmax*Xmax;
    for I := Ø to K do
    Begin
       POKE(addr,data); addr := addr + $1
    end ;
    writeln(' plotting initialized !',Home,addr );
  END;
{***************** CASEAX  *** convert to log scale if necessary ***********}
Procedure CASEAX(var x1,x2,y1,y2 : real);
  begin
   if (FLAG=3)OR(FLAG=4) then begin
                              x1 :=log(x1); x2 :=log(x2)
                            end;
   if (FLAG=2)OR(FLAG=4)then begin
                              y1 := log(y1); y2 := log(y2)
                            end
  end;
{****************************************************************************}
```

```
Procedure PIXEL(x,y : integer) ;
{ set a pixel, and usually called by other subprograms }
  var h,nh,bit : integer ;
  BEGIN
    h := Ymax-y-1 ; nh := h div 8 ; bit := h-nh*8;
    hdata := HEXINT(nh*Xmax+x) ; addr :=Home +hdata ;
    case bit of
      0 : data := $80;
      1 : data := $40;
      2 : data := $20;
      3 : data := $10;
      4 : data := $8 ;
      5 : data := $4 ;
      6 : data := $2 ;
      7 : data := $1
      otherwise data := $0
    end;
    ORPOKE(addr,data)
  END;
{**************************************************************************}
Procedure CONECT(x1,y1,x2,y2 : integer );
{ draw a line, usually called by other subprograms }
   var length:integer ;
       dx,dy,rx,ry : real ;
  BEGIN
    length := abs(x2-x1);
    if abs(y2-y1) > length then length:=abs(y2-y1);
    dx := (x2-x1)/length; dy := (y2-y1)/length; rx := x1 ; ry := y1;
    for I := 0 to length do
    begin
       PIXEL(round(rx),round(ry)); rx := rx+dx; ry := ry+dy
    end
  END; {of CONECT}
{**************************************************************************}
Procedure WINDOW(FL:integer;X1,X2,Y1,Y2: real);
{ define a window and called next to the initialize }
  BEGIN
    FLAG := FL ; CASEAX(X1,X2,Y1,Y2); XAmin:=X1;XAmax:=X2; YAmin:=Y1;YAmax:=Y2;
    AX := (Xmax-1)/(X2-X1); BX:=X1*(Xmax-1)/(X2-X1); AY:=(Ymax-1)/(Y2-Y1); BY:=Y1*(Ymax-1)/(Y2-Y1)
  END;
{**************************************************************************}
Function PX(X:real): integer;
{ transform a real number to pixel number of x axis }
  BEGIN
     PX :=ROUND(AX*X-BX)
  END;
Function PY(Y:real): integer;
```

```
{ transform a real number to pixel number of y axis }
  BEGIN
     PY := ROUND(AY*Y-BY)
  END;
{*****************************************************************************}
Procedure NPLOT(RX,RY:real);
{ set a dot at point (x,y :real) }
  var NX,NY: integer;
  BEGIN
    NX := ROUND(AX*RX-BX); NY := ROUND(AY*RY-BY);
    if (NX<=Xmax)and(NX>=0)and(NY<=Ymax)and(NY>=0)then PIXEL(NX,NY)
                                             else writeln('  window exceeded at',RX,RY)
  END;
{*****************************************************************************}
Procedure PLOT(rx,ry : real);
{transform to log(x,y) if necessary and call NPLOT}
  BEGIN
     if(FLAG=3)OR(FLAG=4)THEN rx:=log(rx);
     if(FLAG=2)OR(FLAG=4)THEN ry:=log(ry);
     NPLOT(rx,ry)
  END;
Procedure LINE(X1,Y1,X2,Y2 : real);
{ draw a strait line between two points }
  var IX1,IX2,IY1,IY2 : integer ;
  BEGIN
     CASEAX(X1,X2,Y1,Y2); IX1:=PX(X1);IY1:=PY(Y1); IX2:=PX(X2);IY2:=PY(Y2); CONECT(IX1,IY1,IX2,IY2)
  END;
{*****************************************************************************}
Procedure FRAME;
{ draw a frame around plotting area }
  var J:integer;
  BEGIN
     J := Xmax-1; K := Ymax-1;
     for I:=0 to J do
       begin
          PIXEL(I,0); PIXEL(I,K)
       end;
     for I:=0 to K do
       begin
          PIXEL(0,I); PIXEL(J,I)
       end;
   END; { of FRAME }
{*****************************************************************************}
Procedure NX(x1,y1,dx:real);
  var IY: integer;
      ax1: real;
  BEGIN
```

```
    IY := PY(y1); CONECT(0,IY,Xmax-1,IY); ax1:= x1+dx ;
    while ax1 <= XAmax do
      begin
         NPLOT(ax1,y1+1/AY); NPLOT(ax1,y1+2/AY); ax1:= ax1+ dx
      end;
    ax1:= x1-dx ;
    while ax1>= XAmin do
      begin
         NPLOT(ax1,y1+1/AY); NPLOT(ax1,y1+2/AY); ax1 := ax1-dx
      end
  END;
{****************************************************************************}
Procedure NY(x1,y1,dy:real);
  var IX: integer;
      ay1: real;
  BEGIN
    IX := PX(x1); CONECT(IX,0,IX,Ymax-1); ay1 := y1+dy;
    while ay1<=YAmax do
      begin
         NPLOT(x1+1/AX,ay1); NPLOT(x1+2/AX,ay1); ay1 := ay1+dy
      end;
    ay1 := y1-dy;
    while ay1>=YAmin do
      begin
         NPLOT(x1+1/AX,ay1); NPLOT(x1+2/AX,ay1); ay1 := ay1-dy
      end
  END;
{****************************************************************************}
Procedure LX(y0:real);
  var t,xt,x1,y1 : real;
      cycle : integer ;
  BEGIN
    cycle := round(XAmax-XAmin); t := 10.0**(XAmin);
    for I := 1 to cycle do
      begin
         for K := 1 to 9 do
           begin
              xt := K*t; x1 := LOG(xt); y1 := y0+1/AY;
              NPLOT(x1,y1); NPLOT(x1,y1+1/AY)
           end;
           t := 10.0*t
       end; {of I}
    I := PY(y0);
    for K := 0 to Xmax-1 do
        PIXEL(K,I);
  END;
Procedure LY(x0:real);
```

```
  var t,xt,x1,y1 : real;
      cycle : integer ;
  BEGIN
    cycle := round(YAmax - YAmin);
    t := 10.0**(YAmin);
    for I := 1 to cycle do
    begin
       for K := 1 to 9 do
       begin
          xt := K*t; x1 := x0+1/AX; y1 := LOG(xt);
          NPLOT(x1,y1); NPLOT(x1+1/AX,y1);
       end;
       t := 10.0*t
    end;
    I := PX(x0);
    for K:= 0 to Ymax-1 do
       PIXEL(I,K)
  END;
{***************************************************************************}
Procedure AXIS(x1,y1,dx,dy : real);
  BEGIN
    if (FLAG=2)or(FLAG=4) THEN y1:=log(y1); if (FLAG=3)OR(FLAG=4) THEN x1:=log(x1);
    if (FLAG=1)OR(FLAG=2)THEN NX(x1,y1,dx); if (FLAG=1)OR(FLAG=3)THEN NY(x1,y1,dy);
    if (FLAG=3)OR(FLAG=4)THEN LX(y1); if (FLAG=2)OR(FLAG=4)THEN LY(x1)
  END;
{***************************************************************************}
Procedure PAXDAT(X1,X2,Y1,Y2 : real ; title:string);
  BEGIN
    writeln(Auxout); writeln(Auxout,' ',title);
    writeln(Auxout,' X-axis ',X1,' to ',X2);writeln(auxout,' Y-axis ',Y1,' to ',Y2);writeln(Auxout);
  END;
Procedure ADDAT(gain,rint:real;unit:char;Nstack:integer);
   begin
    write(Auxout,' sampling interval is ',rint:6:1);
    if unit='M' then writeln(Auxout,' millisecond ')
               else writeln(Auxout,' second ');
     writeln(Auxout,' stacking ',Nstack:4,' times');
     writeln(Auxout,' gain of amp ',gain:8:1); writeln(Auxout)
   end;
{******************************************************************}
  Procedure CRTPLT ;
  var Vmin,Vmax : real ;
      cline : array[0..70] of char ;
      I0,J,N : integer ;
  Function C( v:real ): integer ;
     begin
      C:=round(70*(v-Vmin)/(Vmax-Vmin))
```

```
    end ;
  begin
    writeln(' input min and max to plot '); readln(Vmin,Vmax);
    for I0:=I1 to K1 do
     begin
       for J:=0 to 70 do cline[J]:=' ';
       cline[C(Vmin)]:='|'; cline[C(Vmax)]:='|';
       if (Vmin*Vmax)<=0.0 then cline[C(0.0)]:=':';
       N:=C(AV[I0]) ;
       if ((N>=0)and(N<=71)) then cline[N]:='*' ;
       write( rint*(I0-1):9:4);
       for J:=0 to 70 do write(cline[J]) ;
       if break then exit;
       writeln
     end
  end;
{*****************************************************************************}
 Procedure ADSTRT ;
   begin
       ADCON6(Hsample,Hclok,CLKCHN); write(chr(7)); min:=V[1]/2.0; max:=min ;
       for KK:=1 to Nsample do
         begin
           a:=V[KK]/2.0 ;
           if min>a then min:=a
                   else if max<a then max:=a ;
           AV[KK]:=AV[KK]+a ; VSQR[KK]:=VSQR[KK]+a*a
         end;
       max:=max*0.0049 ;{ 2'S COMPLIMENT }
       min:=min*0.0049 ;
       writeln(II:3,' Min = ',min,'  Max = ',max)
    end;
{***************************************************}
BEGIN { of main program }
  flag1 := 'R' ; NOPAUS; { no pause at endpage }
  while flag1='R' do
  begin {while flag1}
   writeln;writeln(' 12-bit A/D conversion and plot program');
   writeln('  CA2 high 1 sec before sampling ');
   writeln('  Direction of current is reversed for each transient');
   writeln('  maximum number of sample is 2000');
   writeln('  minimum sampling rate is 0.1 millisecond');
   writeln;
   writeln(' input the title');
   readln(title);
   repeat
       write(' input Max-time to measure ! '); readln(Maxtime);
       write(' input time unit ( M = millisecond, S = second ) ! '); readln(unit);
```

```
  until (unit='M') or (unit='S') ;
  write(' input sampling interval ? '); readln(rint);
  Nsample := round(Maxtime/rint); Hsample := HEXINT(Nsample);
  write(' how many stacking ? '); readln(Nstack); Hstack := HEXINT(Nstack );
  if unit='S' then clkrate:=$05;
  if unit='M' then clkrate:=$02;
{ write(' input the channel number in hex (0-F) ');
  readln(channel);}
  channel := $0 ; { fix to channel 0 for voltage input }
  Nclok := round(rint*10.0); Hclok := HEXINT(Nclok) ;
  CLKCHN := clkrate * $100 + channel ;
  I1:=1 ; K1:=Nsample+1 ;
  WRITE(' hit a key to start conversion'); C := CREAD;writeln; II:=1 ;
  repeat
   FOR I9:=1 to 100 do
     begin
       ADSTRT ;
       if break then exit
     end;
   write(' OK (Y/N/D) ? D=display to CRT '); C:=CREAD ;writeln;
   if C='D' then
         begin
          for LL:=1 to Nsample do AV[LL]:=V[LL]*0.00245 ;
          CRTPLT
         end;
   II:=II+1
  until (C='Y') or (C='y');
  for II:=1 to Nsample do AV[II]:=0.0 ;
  for II:=1 to Nsample do VSQR[II]:=0.0 ;
  for II := 1 to Nstack do
    begin
     ADSTRT ;
     if break then exit
    end ; { of stacking}
  writeln;
  writeln(' A/D conversion finished !');
  write(' Input current  (Ampare) '); readln(current);
  write(' Input the Amplifire gain '); readln(gain);
  ctimesg:=current*gain ;
  min:=100.0 ; max:=-100.0;
  for II := 1 to Nsample do
    begin
      a := AV[II]/Nstack;
      VSQR[II]:=(VSQR[II] / Nstack)-SQR(a) ;
      a :=(a*0.0049)/(ctimesg);
      AV[II] :=a ;
      if min>a then min:=a
```

```
              else if max<a then max:=a
      end ;
   writeln;
   writeln(' Min = ',min,'   Max = ',max);
    XX1:=0.0;XX2:=Maxtime;I1:=1;K1:=Nsample+1;
    FL2 := 'D' ;
    while (FL2<>'R') and (FL2<>'E') do
    begin { display and plot }
      writeln; writeln(' Time range set (T)');
      writeln(' Display data (D)'); writeln(' Printout data (P)');
      writeln(' Graphic plot (G)'); writeln(' Save to disk (S)');
      writeln(' Restart A/D (R)'); writeln(' Information (I)');
      writeln(' PLOT to CRT (L)'); writeln(' Exit (E)');
      write(' input command ! ',chr(7)); FL2 := CREAD ;writeln;
{ Time range set}
      if FL2='T' then
         begin
           writeln(' input time range in two real numbers ? '); readln(XX1,XX2);
           I1 := Round(XX1/rint)+1 ; K1 := Round(XX2/rint)+1 ;
           if K1>Nsample then K1:=Nsample
         end;
{ Display }
   if FL2='D' then for II:=I1 to K1 do
      begin
        if break then exit ;
        writeln(II,(II-1)*rint,AV[II],VSQR[II])
      end ;
{*******************************************************}
   if FL2='I' then
      begin
        writeln; writeln(' TITLE :   ',title);
        writeln('  Current :   ',current:6:2,' Ampare');
        writeln('  Gain    :   ',gain:12:2,' times');
        writeln('  minimum voltage : ',min,'   Max voltage : ',max);
        write('  time range  ',XX1:12:2,'to',XX2:12:2);
        if unit='M' then writeln(' millisecond')
                    else writeln(' second');
        writeln
      end; {I}
{*******************************************************}
    if FL2='F' then
      for II:=1 to Nsample do AV[II]:=-AV[II];
    if FL2='A' then
      for II:=1 to Nsample do AV[II]:=abs(AV[II]);
{*******************************************************}
    if FL2='L' then CRTPLT ;
{ Printout data }
```

```
    if FL2='P' then
      begin {if}
           PSET ;
           writeln(Auxout);writeln(Auxout,'  ',title);
           writeln(Auxout,chr(27),'P');
           for II:= I1 to K1 do
             writeln(Auxout,II,rint*(II-1),AV[II],VSQR[II]);
           writeln(Auxout,chr(12),' ',chr(27),'@')
        end ; {if}
{  GRAPHIC PRINT }
  if FL2='G' then
  begin {Graphic}
  WRITELN;
  WRITELN(' PLOTTING PROGRAM USING MX-80 ');
  WRITELN(' INPUT 1=Lin-Lin, 2=Lin-Log, 3=Log-Lin, 4=Log-Log, 5=End');
  readln(FL1);
  if FL1<>5 THEN
     BEGIN
       PSET ;
       write(' Auto scale  (Y/N) ? '); readln(auto);
       if auto then
          begin
            YY1:=min ;  YY2:=max
          end
          else begin
            writeln(' input Ymin and Ymax ? ');
            readln(YY1,YY2)
          end;
       PLTINI;
       WINDOW(FL1,XX1,XX2,YY1,YY2);
     repeat
       WRITELN(' INPUT AXIS DATA X0,Y0,DX,DY ');
       READLN(XR1,YR1,XR2,YR2)
     until (XR1>=XX1)and(XR1<=XX2)and(YR1>=YY1)and(YR1<=YY2);
       AXIS(XR1,YR1,XR2,YR2);
       FRAME;
        for II:= I1 to K1 do
          begin
            a := rint*(II-1);
            b := AV[II] ;
            PLOT(a,b)
          end;
       repeat
         WRITE('  plot to the Printer (Y/N) ?');
         FL2 := CREAD;writeln
       until (FL2='Y') or (FL2='N') ;
       IF FL2='Y' THEN
```

```
         begin
            PLTOUT;
            PAXDAT(XX1,XX2,YY1,YY2,title);
            ADDAT(gain,rint,unit,Nstack);
            writeln(auxout);
            writeln(' input comment lines ');
            repeat
              readln(comment);
              writeln(auxout,comment)
            until comment='';
            writeln(auxout,chr(12))
         end
      end {of FL1}
    end; {of Graphic}
{ save data to disk }
     if FL2='S' then
       begin
         writeln(' what is the name of the file ?');
         readln(fname);
         CREATE(name,fname,output);
          writeln(name,title);
          writeln('  input comment lines  (only CR to end)');
          repeat
            readln(comment);
            writeln(name,comment)
          until comment='';
          writeln(name,'$DATA');
         for II:=I1 to K1 do
          begin
           a := rint*(II-1) ;
           b := AV[II] ;
           writeln(name,a,b)
          end;
        CLOSE(name)
       end;{ of if}
{ exit}
  if FL2='E' then flag1:=FL2;
  end {of FL2 loop}
   end;{of while flag1}
  PAUSE
END.
```

Assembly language procedure, linked to ADPLT6

Relocatable Assembler version 1.20

```
0000                     NAM   ADSUB6
0000                     XDEF  TIMINI,TIMSET,ADCON6
0000               *
0000               * A/D CONVERSION AND PLOT TO EPSON MX-80
0000               *    5/30/83  FOR PASCAL PROGRAM ADPLT6.TXT AND SUB1.RO
0000               *    7/7/83   TIME TO 0.5 SEC
0000               *
0000  A0 00        STADR  EQU   $A000
0000  F8 08        INCHEK EQU   $F808
0000  F8 10        PSTRNG EQU   $F810
0000  E0 50        ADADR  EQU   $E050     A/D BOARD PORT#5
0000  E0 40        PORT4  EQU   $E040     A-PORT OF PIA FOR TRIGGER INPUT
0000  E0 42        TIMER  EQU   $E042     TIMER BOARD PORT #4 FOR PSP
0000  BF F0        NCLOKX EQU   $BFF0        SAVE IX FOR NUMBER OF CLOCK CYCLE
0000  BF F2        NSAMPL EQU   $BFF2        SAVE IY FOR NUMBER OF SAMPL
0000  BF F4        CLK    EQU   $BFF4        SAVE CLOCK RATE (1SEC=$06)
0000  BF F5        CHN    EQU   $BFF5        SAVE CHANNEL NUMBER
0000               *
0000  34 30        ADCON6 PSHS  Y,X          SAVE IY AND IX
0002  37 36               PULU  Y,X,D        IY=NSAMPLE, IX=NCLOCK, D=CL
0004  B7 BF F4            STA   CLK
0007  F7 BF F5            STB   CHN
000A  BF BF F0            STX   NCLOKX       SAVE IX
000D  10 BF BF F2         STY   NSAMPL
0011  8E A0 00            LDX   #STADR
0014  1A 10               ORCC  #$10         NO INTRUPT
0016  17 00 AD            LBSR  TIMINI       INITIALIZE TIMER
0019  B6 E0 40            LDA   PORT4
001C  86 34               LDA   #$34         I OFF
001E  B7 E0 41            STA   PORT4+1
0021  86 10               LDA   #$10         CURRENT (-)
0023  B7 E0 50            STA   ADADR
0026  17 00 BF            LBSR  T05S         WAIT 0.5 SEC
0029  86 3C               LDA   #$3C
002B  B7 E0 41            STA   PORT4+1      CA2 TO HIGH TO I ON (-)
002E  17 00 B7            LBSR  T05S         WAIT 0.5 SECOND
0031  86 34               LDA   #$34
0033  B7 E0 41            STA   PORT4+1      CA2 TO LOW I OFF
0036  B6 E0 40            LDA   PORT4        CLEAR FLAG
0039  17 00 9D            LBSR  TIMSET       TIMER SET FOR A/D
```

```
003C  B6 BF F5            LDA   CHN
003F  B7 E0 50            STA   ADADR       SET CHANNEL
0042  B6 BF F5    ADSTRT  LDA   CHN
0045  B7 E0 50            STA   ADADR       START CONVERSION
0048  FC E0 50            LDD   ADADR       LOAD DATA TO ACC D
004B  85 08               BITA  #$08        BIT3 OF ACCA SET?
004D  26 04               BNE   POSI1       IF ZERO NEGATIVE
004F  8A F8       NEG1    ORA   #$F8
0051  20 02               BRA   SAVE1
0053  84 07       POSI1   ANDA  #$07        CHANGE TO 2'S COMLIMENT
0055  ED 81       SAVE1   STD   ,X++        SAVE DATA TO MEMORY
0057  36 10               PSHU  X
0059  BE BF F0            LDX   NCLOKX
005C  13          WAIT    SYNC
005D  B6 E0 42            LDA   TIMER       CLEAR INTRUPT FLAG
0060  30 1F               LEAX  -1,X        CLOCK LOOP FINISHED ?
0062  26 F8               BNE   WAIT
0064  37 10               PULU  X           RESTORE IX (MEMORY INDEX)
0066  31 3F               LEAY  -1,Y        DEC Y
0068  26 D8               BNE   ADSTRT
006A  B6 E0 40            LDA   PORT4
006D  86 00               LDA   #$00        CURRENT (+) DIRECTION
006F  B7 E0 50            STA   ADADR
0072  17 00 73            LBSR  T05S        WAIT 0.5 SEC
0075  86 3C               LDA   #$3C        I ON (+)
0077  B7 E0 41            STA   PORT4+1
007A  8E A0 00            LDX   #STADR
007D  10 BE BF F2         LDY   NSAMPL      RESTORE # OF SAMPLE
0081  17 00 64            LBSR  T05S        WAIT 0.5 SEC
0084  86 34               LDA   #$34        I OFF (+)
0086  B7 E0 41            STA   PORT4+1
0089  B6 E0 40            LDA   PORT4
008C  8D 4B               BSR   TIMSET
008E  B6 BF F5            LDA   CHN
0091  B7 E0 50            STA   ADADR
0094  B6 BF F5    ADST2   LDA   CHN
0097  B7 E0 50            STA   ADADR       A/D START
009A  FC E0 50            LDD   ADADR       DATA TO ACC D
009D  85 08               BITA  #$08
009F  26 04               BNE   POSI2       CHANGE TO 2'S COMPLIMENT
00A1  8A F8       NEG2    ORA   #$F8        MASK TO UPPER BIT 1
00A3  20 02               BRA   SAVE2
00A5  84 07       POSI2   ANDA  #$07
00A7  A3 84       SAVE2   SUBD  ,X
00A9  ED 81               STD   ,X++        SUBTRUCT AND STORE
00AB  36 10               PSHU  X
00AD  BE BF F0            LDX   NCLOKX
```

```
00B0  13            WAI2   SYNC
00B1  B6 E0 42             LDA   TIMER
00B4  30 1F                LEAX  -1,X
00B6  26 F8                BNE   WAI2
00B8  37 10                PULU  X
00BA  31 3F                LEAY  -1,Y       DEC Y
00BC  26 D6                BNE   ADST2
00BE  8D 06                BSR   TIMINI     STOP THE CLOCK
00C0  B6 E0 42             LDA   TIMER      I-FLAG CLEAR
00C3  35 30                PULS  Y,X
00C5  39                   RTS              RETURN TO MAIN
00C6                *
00C6  86 FF         TIMINI LDA   #$FF       INITIALIZE TIMER
00C8  B7 E0 42             STA   TIMER
00CB  86 3D                LDA   #$3D
00CD  B7 E0 43             STA   TIMER+1
00D0  86 34                LDA   #$34
00D2  B7 E0 41             STA   PORT4+1  CA1 NEG.EDGE ACTIVE
00D5  B6 E0 40             LDA   PORT4
00D8  39                   RTS
00D9                *
00D9  86 80         TIMSET LDA   #$80       SET TIMER
00DB  B7 E0 42             STA   TIMER
00DE  B6 BF F4             LDA   CLK
00E1  B7 E0 42             STA   TIMER      TIMER SET AND START
00E4  B6 E0 42             LDA   TIMER       I-FLAG CLEAR
00E7  39                   RTS
00E8  C6 80         T05S   LDB   #$80       0.5 SEC DELAY
00EA  F7 E0 42             STB   TIMER
00ED  86 05                LDA   #$05       SET T0 0.1 SEC
00EF  B7 E0 42             STA   TIMER
00F2  7D E0 42             TST   TIMER
00F5  13                   SYNC
00F6  7D E0 42             TST   TIMER
00F9  13                   SYNC
00FA  7D E0 42             TST   TIMER
00FD  13                   SYNC
00FE  7D E0 42             TST   TIMER
0101  13                   SYNC
0102  7D E0 42             TST   TIMER
0105  13                   SYNC
0106  7D E0 42             TST   TIMER
0109  39                   RTS
010A                       END
Total errors : 0   Psct size : 010A   Table usage : 25
```

APPENDIX C

DATA PROCESSING PROGRAM (WPLOT)

```
 {$B+  Data processing program using Watanabe DA6000 digital plotter }
Program WPLOT(input,output,auxout);
 {$B-}

 var fname,title,command : string ;
  `  name        : text ;
     Npoint,NP   : integer ;
     X,X1,Y,Y1   : array[1..1024] of real ;
     abort,first,dataok : boolean ;
     K,I1,I2,Ncomment   : integer ;
     value       : real ;
     comment     : array[1..20] of string;
{*****************************************************************************************}
  Procedure PLOT(var X,Y:array[1..1024] of real;Npoint,FL:integer; first:boolean);external;
  Procedure PSET   ; external ; { printer set }
  Procedure PLTSET ; external ; { plotter set }
  Procedure PAUSE  ; external ;
  Procedure NOPAUS ; external ;
  Procedure EXPREG(var X,Y:array[1..1024] of real;var Npoint:integer); external ;
{*****************************************************************************************}
  Procedure INCMD( var FL : string);
    begin
     REPEAT
      writeln;write(' Command ?      H=help ',chr(7)); readln(FL); FL:=upshift(FL);
      if FL='H' then
      begin
        write(' Create new data file.....(CRE)'); writeln(' Type data to CRT.........(TYP)');
        write(' Flip data ...............(FLI)'); writeln(' Absolute value ..........(ABS)');
        write(' Add offset Value to Y....(YAD)'); writeln(' add value to X...........(XAD)');
        write(' Multiple a value to Y....(YMU)'); writeln(' Multilpe a value to X....(XMU)');
        write(' Range Set of X to plot...(RAN)'); writeln(' Regression ..............(REG)');
        write(' Plot to CRT..............(CRT)'); writeln(' Reset to Initial value...(RES)');
        write(' Initial plot.............(IPL)'); writeln(' Output to plotter........(PLO)');
        write(' Print data to printer....(PRI)'); writeln(' save data to Disk........(SAV)');
        write(' Get data from disk.......(GET)'); writeln(' check data contents......(CHE)');
        write(' Edit data................(EDI)'); writeln(' Curvilinear Interpolate..(CUR)');
        write(' Smoothing................(SMO)'); writeln(' Exit.....................(EXI)')
      end
     UNTIL FL<>'H'
    end;
{*************************************************************}
  Procedure Create ;
```

```
     var Xinc,a,b : real ;
         I    : integer ;
   begin
     I1:=1 ;
     write(' input X[1] value and X-increment ! '); readln( X[1],Xinc);
     X1[1]:=X[1];
     writeln('  to stop input, type -9999 ');
     if Xinc>0 then
         begin
           I:=1 ;
           repeat
             write('  I=',I:4,' , X[I]= ',X[I]:10:4,'  input Y[I] ');
             readln( Y[I] ); Y1[I]:=Y[I] ; I:=I+1 ; X[I]:=X[I-1]+Xinc; X1[I]:=X[I]
           until Y[I-1]=-9999
         end
         else begin
             I:=1 ;
             repeat
               write(' I= ',I:4,'  input X[I] and Y[I] ');
               readln(X[I],Y[I]); X1[I]:=X[I] ; Y1[I]:=Y[I] ; I:=I+1
             until X[I-1]=-9999
         end ;
      Npoint:=I-2; NP:=Npoint ; I2 := Npoint;
      writeln('  Input comment for the data ');
      I:=0;
      repeat
        I:=I+1; readln(comment[I])
      until comment[I]='';
      Ncomment:=I
   end ;
{*******************************************************************}
  Procedure CRT;
    var I: integer ;
   begin
     for I:=1 to NP do begin
                         if break then exit;
                         writeln( X[I],'  ',Y[I])
                      end;
     writeln
   end ;
{*******************************************************************}
 Procedure Printout;
   var I: integer ;
   begin
     PSET; { print program load }
     writeln(auxout,title); writeln(auxout);
     for I:=1 to NP do begin
```

```
                    writeln(auxout,X[I],Y[I] );
                    if break then exit
                  end ;
     writeln(auxout, chr(12));
     PLTSET   { PLOTTER SET }
   end ;
{****************************************************************}
  Procedure SAVEDISK ;
    var I: integer ;
    begin
      write(' input the file name ? '); readln(fname);
      write(' input the comment of the data ! '); readln(title);
      rewrite(name,fname) ;
      writeln(name,title); writeln(name); writeln(name,'$DATA');
      for I:=1 tl NP do writeln(name,X[I],Y[I]);
      close(name)
    end;
{****************************************************************}
  Procedure READDISK ;
    var I,I1: integer ;
        title : string ;
    begin
      writeln(' What is the name of the Input file ? '); readln(fname);
      OPEN(name,fname,input);
      title:=''; I1=1 ;
      while (title<>'$DATA') and (not EOF(name))and(I1<20) do
          begin
            readln(name,title); comment[I1]:=title ;
            I1:=I1+1 ; writeln(title)
          end;
      Ncomment:=I1-1;
      if I1>19 then writeln(' too many comment line ');
      if EOF(name) then
            writeln(' Data format error !, use $DATA before 1st data',chr(7));
      I:=1 ;
      while not EOF(name) do
        begin
          readln(name,X[I],Y[I]);
          X1[I]:=X[I] ; Y1[I]:=Y[I] ; I:=I+1
        end ;
      Npoint:=I-1 ; NP:=Npoint ; I1:=1 ; I2:=Npoint ;
      CLOSE( name)
   end ;
{**********************************************************}
  Procedure Range ;
   var R1,R2 : real ;
       I     : integer ;
```

```
  begin
   repeat
    writeln;writeln(' current data range ',X1[1],'  to',X1[NP]);
    write(' input two real numbers for new range '); readln(R1,R2)
   until (R1>=X[1])and(R2<=X[NP])and(R1<R2) ;
    I:=0 ;
    repeat
     I:=I+1;
    until R1<=X[I] ;
   I1:=I ;
   repeat
    I:=I+1
   until X[I]>=R2 ;
   I2:=I; NP:=I2-I1+1 ;
   for I:=1 to NP do
    begin
      X[I]:=X[I+I1-1] ; Y[I]:=Y[I+I1-1]
    end
 end ;
{************************************************************}
 Procedure CRTPLT ;
  var Vmin,Vmax : real ;
      line : array[0..70] of char ;
      I,J,N : integer ;
  Function C( v: real):integer ;
    begin
      C:=round(70*(v-Vmin)/(Vmax-Vmin))
    end;
  begin (CRTPLT)
    writeln(' input min and max to plot '); readln(Vmin,Vmax);
    for I:=1 to NP do
    BEGIN
     for J:=0 to 70 do line[J]:=' ';
     line[ C(Vmin) ]:= '|'; line[ C(Vmax) ]:='|';
     if (Vmin*Vmax)<=0.0 then line[C(0.0)]:=':';
     N:=C(Y[I]) ;
     if((N>=0) and (N<=71)) then line[N]:='*' ;
     WRITE(X[I]:9:4);
     for J:=0 to 70 do write(line[J]);
     if break then exit ;
     writeln
    end
   end;
{*******************************************************}
 Procedure REGRESSION ;
   var plt : boolean ;
 begin
```

```
    RANGE;
    EXPREG(X,Y,NP);
    PLTSET ;
    write(' Plot to the plotter (Y/N) ? '); readln(plt);
    if plt then PLOT(X,Y,NP,5,false)
  end;
{************ EDIT **********************************}
  Procedure EDIT;
    var Kedit : integer ;
        Xedit : real;
        CMD   : char ;
    begin
      repeat
        write('  Input X-value ');readln(Xedit)
      until (Xedit>=X[1]) and (Xedit<=X[NP]);
      K:=0;
      repeat
        K:=K+1
      until X[K]>Xedit*1.001;
      Kedit:=K-1;
      repeat
       repeat
        writeln(Kedit,'  X=',X[Kedit],'  Y=',Y[Kedit]);
        write('   OK ?  Y=change,  U=up 1 line, D=down 1 line , A=abort ');
        readln(CMD)
       until CMD in ['Y','U','D','A'];
       if CMD='Y' then
          begin
            if Kedit<=NP then
              begin
                write(' New Y value ? '); readln(Y[Kedit]);
              end
              else begin
                write(' Input new X-value and Y-value to add ');
                readln(X[Kedit],Y[Kedit]); NP:=Kedit
              end;
            Kedit:=Kedit+1
          end;
       if CMD='U' then Kedit:=Kedit-1;
       if CMD='D' then Kedit:=Kedit+1
      until CMD='A'
   end;
{**************************************************}
  Function Lagrange(xint : real ):real ;
     var alpha, beta : real ;
         I,J : integer ;
     begin
```

```
      beta :=0.0 ;
      for I:=1 to NP do
        begin
          alpha:=1.0 ;
          for J :=1 to NP do
            if I<>J then alpha:=alpha*(xint-x[J])/(x[I]-x[J]);
          beta :=beta + (alpha*y[I])
        end ; {I}
      Lagrange:=beta
    end; {Lagrange}
{****************************************************}
{  Akima Spline fitting from basic scientific subroutines vol.2 p308}
  Function Spline(xint : real):real;
    var M,Z : array[0..250] of real ;
        I,V : integer ;
        a,b,c,dx: real ;
    begin
      V:=NP;
      if (xint < x[1]) or (xint>x[NP-3]) then
        begin
           dx:=x[Np]-x[NP-1]; x[NP+1]:=x[NP]+0.1*dx; x[NP+2]:=x[NP]+0.2*dx;
           y[NP+1]:=y[NP]; y[NP+2]:=y[NP]
        end;
      abort:=false;
      if not abort then
        begin
          for I:=1 to V-1 do M[I+2]:=(y[I+1]-y[I])/(x[I+1]-x[I]);
          M[V+2]:=2*M[V+1]-M[V]; M[V+3]:=2*M[V+2]-M[V+1];
          M[2]:=2*M[3]-M[4]; M[1]:=2*M[2]-M[3];
          for I:=1 to V do
            begin
              a:=abs(M[I+3]-M[I+2]); b:=abs(M[I+1]-M[I]);
              if (a+b)= 0.0 then Z[I]:=(M[I+2]+M[I+1])/2
                             else Z[I]:=(a*M[I+1]+b*M[I+2])/(a+b)
            end;
          I:=0 ;
          repeat
            I:=I+1
          until xint<x[I] ;
          I:=I-1;
          if I=0 then begin
                        b:=x[1]-(2*x[1]-x[2]); a:=xint-(2*x[1]-x[2])
                      end
                      else begin
                             b:=x[I+1]-x[I]; a:=xint-x[I]
                      end;
          c:=y[I]+z[I]*a+(3*M[I+2]-2*z[I]-z[I+1])*a*a/b;
```

```
        Spline:=c+(z[I]+z[I+1]-2*M[I+2])*a*a*a/(b*b)
      end
    end;
{*****************************************************************}
  Procedure Curve ;
    var L,M,nint : integer ;
        plt : boolean ;
        flag : char ;
        dx,xm : real ;
    begin
      if NP>100 then
        begin
          writeln('  too many sample point '); abort:=true
        end;
      if not abort then
        begin
          repeat
            write('  Lagrange (L), or Spline (S), or Abort (A)  ? '); readln(flag)
          until flag in ['L','S','A'] ;
          if flag = 'A' then abort := true
        end;
      if not abort then
        begin
          write(' How many points for entire curve ? '); readln(nint) ;
          dx:=(x[NP]-x[1])/nint ;  xm:=x[1] ;
          for L:=1 to nint+1 do
            begin
              x[L+100]:=xm ;
              if flag='L' then y[L+100]:=lagrange(xm)
                          else y[L+100]:=Spline(xm);
              xm:=xm+dx
            end;
          for L:=1 to nint+1 do begin
                                  x[L]:=x[L+100]; y[L]:=y[L+100]
                                end;
          NP:=nint+1;
          write(' plot ? ');readln(plt);
          if plt then PLOT(x,y,NP,5,false)
        end
    end; { curve}
{***************************************************}
  Procedure Smooth;
     var a,b,c : real ;
         N1,N2 : integer ;
         flg : char ;
     begin
       write(' input start and end # of point to smooth ') ; readln(N1,N2);
```

```
      repeat
        write(' Linear (L), or Exponent (E), or abort (A) ');  readln(flg)
      until flg in ['A','L','E'];
      if flg<>'A' then
        begin
          if N2>NP then N2:=NP ;
          N1:=N1+1 ; a:=y[N1-1]; b:=y[N1]; c:=y[N1+1];
          if flg='E' then begin
                           a:=ln(a) ; b:=ln(b) ; c:=ln(c)
                         end;
          repeat
            y[N1]:=(2.0*b+a+c)/4.0 ;
            if flg='E' then y[N1]:=exp(y[N1]);
            N1:=N1+1; a:=b ; b:=c ; c:=y[N1+1];
            if flg='E' then c:=ln(c)
          until N1+1=N2
        end {not A}
    end;
{*****************************************************}
  begin { main }
    writeln;  writeln('  Watanabe Plotter plot program');
    PLTSET ; NOPAUS ;
    dataok := false ; abort:=false ; first:=true ;
    repeat
      INCMD(command);
      Case command of
       'CRE' : begin Create;dataok:=true end;
       'TYP' : CRT ;
       'FLI' : for K:=1 to NP do Y[K]:=-Y[K] ;
       'ABS' : for K:=1 to NP do Y[K]:=abs(Y[K]) ;
       'YAD' : begin
                write(' Input the offset value to be added '); readln(value);
                for K:=1 to NP do Y[K]:=Y[K]+value
               end ;
       'PRI' : if dataok then Printout else writeln(' No data !');
       'XAD' : begin
                write(' Input the offset value to add to X '); readln(value);
                for K:=1 to NP do X[K]:=X[K]+value
               end;
       'YMU' : begin
                write(' Input the value to Mult. to Y '); readln(value);
                for K:=1 to NP do Y[K]:=Y[K]*value
               end;
       'XMU' : begin
                write(' Input the value to Mult. to X '); readln(value);
                for K:=1 to NP do X[K]:=X[K]*value
               end ;
```

```
'SAV' : if dataok then SAVEDISK else writeln(' No data !');
'REG' : REGRESSION ;
'CUR' : Curve ;
'GET' : begin READDISK ;dataok:=true end;
'IPL' : begin
         if dataok then begin
                         PLOT(X,Y,NP,5,true); first:=false
                        end
                   else writeln(' No data !')
        end;
'PLO' : if dataok then PLOT(X,Y,NP,5,false);
'RAN' : if dataok then Range ;
'CRT' : if dataok then CRTPLT ;
'SMO' : if dataok then Smooth ;
'EDI' : if dataok then EDIT;
'RES' : if dataok then begin
                         for K:=1 to Npoint do begin
                                                 X[K]:=X1[K] ; Y[K]:=Y1[K]
                                               end;
                         I1:=1 ; I2:=Npoint ; NP:=Npoint
                       end;
'CHE' : begin
        writeln('**Original Buffer**  ',Npoint,' points ');
        writeln(' X axis range ',X1[1],'  to',X1[Npoint]); writeln ;
        writeln('**Current data**     ',NP,' points ');
        writeln('    X axis range ',X[1],'  to',X[NP]); writeln ;
        if Ncomment>1 then
             for K:=1 to Ncomment do writeln(comment[K])
      end;
'EXI' : abort:=true
else writeln(' invalid command ')
end {case}
until abort;
PAUSE
end.
```

```
Module REGSUB1(input,output,auxout);
  { data regresion program, link to WPLOT }
{*************************************************************************}
Procedure PSET; external ;
Procedure  EXPREG(var X,Y:array[1..1024] of real;var Npoint:integer);ENTRY;
   var Xcoor,Ycoor,sumx,sumy,coeff,error : real ;
       sumxsqr,sumysqr,sumxy,a,b,x1,x2,step  : real ;
       cont : boolean ;   I : integer ;
       command,flag : char ;
   Procedure Getdata ;
    var i : integer;
    begin
      sumx:=0; sumy:=0; sumxsqr:=0; sumysqr:=0; sumxy:=0;
      for i:=1 to Npoint do
         begin
            if (command='L')or(command='E') then Xcoor:=X[i] else Xcoor:=ln(abs(X[i]));
            if (command='L') then Ycoor:=Y[i] else Ycoor:=ln(abs(Y[i]));
            sumx:=sumx+Xcoor ; sumy:=sumy+Ycoor ;
            sumxsqr:=sumxsqr+sqr(Xcoor); sumysqr:=sumysqr+sqr(Ycoor); sumxy:=sumxy+Xcoor*Ycoor;
         end;
      if Npoint*sumxsqr=sumx*sumx then
         begin
            cont:=false; writeln(' Regression cannot be calculated !')
         end
       else begin
              b:=(Npoint*sumxy-sumy*sumx)/(Npoint*sumxsqr-sumx*sumx);
              a:=(sumy-b*sumx)/Npoint
            end
    end;
{*************************************************************************}
 Procedure CRA ;
    var temp : real;
    begin
      temp:=b*(sumxy-sumx*sumy/Npoint);
      coeff:=abs(temp/(sumysqr-sqr(sumy)/Npoint));
      error:=sumysqr-sqr(sumy)/Npoint-temp
    end;
{*************************************************************************}
 Procedure Interpolate;
      var i : integer;
          a1: real ;
```

```
      begin
        if command='E' then begin
                             a1:=exp(a);
                             for i:=1 to Npoint do Y[i]:=a1*exp(b*X[i])
                           end;
        if command='L' then
                         for i:=1 to Npoint do  Y[i]:=b*X[i] + a ;
        if command='G' then begin
                             a1:=exp(a);
                             for i:=1 to Npoint do Y[i]:=a1*(X[i]**b)
                           end
      end;
{***********************************************************************}
  Procedure Writeit(Pout:boolean);
     var Outfile : text ;
     begin
       PSET;
       if Pout then Open(Outfile, auxout) else Open(Outfile, output); writeln(Outfile);
       if command='E' then begin
                            writeln(outfile,'  Exponential Regression ');
                            writeln(outfile,'        Y = A * EXP(B*X)');
                            writeln(outfile,'             A =',exp(a):12:6);
                            writeln(outfile,'             B =',b:12:6)
                          end;
       if command='L' then begin
                            writeln(outfile,' Linear Regression ');
                            writeln(outfile,'         Y = A * X + B');
                            writeln(outfile,'            A = ',b:12:6);
                            writeln(outfile,'            B = ',a:12:6)
                          end;
       if command='G' then begin
                            writeln(outfile,' Geometric Regression ');
                            writeln(outfile,'          Y = A * X**B');
                            writeln(outfile,'             A = ',exp(a):12:6);
                            writeln(outfile,'             B = ',b:12:6)
                          end;
       writeln(outfile); writeln(outfile,'  Coefficient of determination (R-sqr)=',coeff:9:6);
       writeln(outfile,'  Coefficient of correlation=',sqrt(abs(coeff)));
       if Npoint=2 then Npoint:=3 ;
       writeln(outfile,' Standard error of estimate=', sqrt(abs(error/(Npoint-2))):9:6);
       writeln(outfile); Close(outfile)
    end;
{***********************************************************************}
  Procedure Result;
     var Pout : boolean;
     begin
```

```
      Pout:=false;
      writeln; writeit(Pout);writeln; {write results}
      write('   Output to Printer (Y/N) ? '); readln(Pout);
      if Pout then Writeit(Pout)
    end;
{*************************************************************************}
      begin { EXREG }
         repeat
           writeln; writeln(' Regression Subprogram ');
           writeln('  Linear Regression...........(L)');
           writeln('  Exponential Regression......(E)');
           writeln('  Geometric Regression........(G)');
           writeln('  Abort.......................(A)');
           write('   Command ? '); readln(command)
         until command in ['L','G','E','A'] ;
         if command<>'A' then
           begin
             cont:=true; Getdata ;
             if cont then
               begin
                 CRA; Result ;
                 repeat
                   writeln;
                   writeln(' Replace with calculated data ...(R)');
                   writeln(' Calculate for the range ........(C)');
                   writeln(' Abort ..........................(A)');
                   write('   Command ? '); readln(flag)
                 until flag in ['R','C','A'] ;
                 if flag = 'R' then Interpolate ;
                 if flag = 'C' then
                   begin
                     write('  Input X axis range '); readln(x1,x2);
                     write('  Input Number of point to calculate ');readln(Npoint);
                     step:=abs(x2-x1)/Npoint ; I:=1 ; X[I]:=x1 ;
                     repeat
                       I:=I+1 ; X[I]:=X[I-1]+step
                     until X[I]>x2*1.001 ;
                     Npoint:=I-1;
                     Interpolate
                   end {C}
               end {cont}
           end { not A }
      end;
MODEND.
```

```
Module PLTSUBW(input,output,auxout);
{ Procedures for Watanabe Plotter driver }

{  X[1..Npoint] : X-axis data
   Y[1..Npoint] : Y-axis data
         Npoint : Number of point
             FL : flag  1=linear-linear
                        2=linear-log
                        3=log-linear
                        4=log-log
                        5=not defined yet }

{********* Procedure PLOT ***********************************************}
Procedure PLOT(var X,Y:array[1..1024] of real;Npoint,FL:integer; first:boolean ); entry ;
  var Xmin,Xmax,Ymin,Ymax : real ;
      ax,bx,ay,by : real ;  { coefficients}
      x1,x2,y1,y2 : real ; { plot window }
      x0,y0       : real ; { origin of axis }
      logx1,logy1,xi,yi,xm,ym : real ;
      xcycle,ycycle,I,N,I1,I2 : integer ;
      ch  : char ;
      name : text ;
      fname : string ;
      range,abort : boolean ;
{******** Procedure MinMax *** find minimum and maximum of X & Y array ***}
  Procedure MinMax( P: boolean );
    var a,b : real ;
        I : integer ;
    begin {MinMax}
      Xmin:=X[1] ; Xmax:=X[Npoint] ; Ymin:=Y[1] ; Ymax:=Y[Npoint] ;
      for I:=1 to Npoint do
        begin
          a:=X[I] ; b:=Y[I] ;
          if Xmin > a then Xmin:=a else if Xmax<a then Xmax:=a ;
          if Ymin > b then Ymin:=b else if Ymax<b then Ymax:=b
        end;
    if P=true then begin
                     writeln(' Xmin= ',Xmin,' Xmax=',Xmax ); writeln(' Ymin= ',Ymin,' Ymax=',Ymax )
                   end { if P}
    end; {MinMax}
{***************** Procedure Cycle **************************}
  Procedure Cycle(var R1,R2:real ; var ncycle:integer);
```

```
    var I1,I2 : integer ;
    begin
      I1:=floor(log(R1)+0.01) ; I2:=floor(log(R2)) ;
      if abs(log(R2)-I2)>0.001 then I2:=I2+1 ;
      R1:=10**I1 ; R2:=10**I2 ; ncycle := I2-I1
    end ;
{********* Procedure Window ********************************}
  Procedure Window;
    var Lx,Ly,           { length of plot area in (mm) }
        Ox,Oy : integer ; (coordinate of lower left corner }
        ld    : integer ; { length of one cycle of log axis }
        speed,K   : integer ;   flag : char ;
        OK ,cont  : boolean ;
        tlx,tly:integer ; { temporary Lx,Ly }
    begin { Window }
      Lx := 200 ; Ly := 150 ; Ox := 20 ; Oy := 15 ;
      repeat
        write(' Plot size parameter S=standard, T=thesis, I=input '); readln(flag)
      until flag in ['S','T','I'];
      if flag='T' then begin
                         Lx:=180 ; Ly:=120 ; Ox:=20 ; Oy:=25
                       end;
      if flag='I' then
        begin
          repeat
            write(' Input X-axis and Y-axis length in (mm) '); readln(Lx,Ly);
            write(' Input Coordinate of lower left corner in (mm) '); readln(Ox,Oy);
            if(Lx+Ox<=250)and(Ly+Oy<=180) then OK:=true
                                     else begin
                                            writeln(' Too big ! '); OK:=false
                                          end
          until OK
        end ;
      repeat
        write(' input pen speed (10=max, ...., 1=min) ? '); readln(speed)
      until ( speed in [1..10]) ;
      writeln(auxout,'T',speed:2);
      REPEAT   {  until OK }
        OK:=true ; MinMax(true); write(' auto scale (Y/N) ? ');readln(ch);
        if (ch='Y')or(ch='y') then begin
                                 x1:=Xmin ; x2:=Xmax ; y1:=Ymin ; y2:=Ymax
                               end
                          else begin
                                 repeat
                                   write(' input X-axis range  ? ');readln(x1,x2) ;
                                   write(' input Y-axis range  ? ');readln(y1,y2);
                                   write(' OK (Y/N) ? '); readln(cont)
```

```
                              until cont
                             end ; {else}
     if FL=5 then
       begin
         repeat
           write(' input lin-lin=1, lin-log=2, log-lin=3, log-log=4 '); readln(FL)
         until ( FL in [1..4] )
       end; {FL=5}
     if ((FL=2)or(FL=4)) and (y1<=0.0) then
      begin
         write(' Negative Y-value, use ABS(Y) (Y/N) ? '); readln(ch);
         if (ch='Y') or (ch='y') then for K:=1 to Npoint do Y[K] := ABS(Y[K])
                              else FL:=5 ;
         OK:=false
      end ;
 IF OK THEN
  BEGIN
   if (FL<>1) then
      begin
         write(' Input the 1 cycle lenght of log axis in (mm) '); readln(ld)
      end;
   if(FL=1)or(FL=2)then ax:=Lx*10/(x2-x1)
                  else begin
                         ax :=ld*10; Cycle( x1,x2,xcycle); logx1:=log(x1);
                         Lx:=xcycle*ld ;
                         if Lx>200 then
                           begin
                             writeln(' X-axis cycle is ',xcycle,'  Reduce length !');
                             OK:= false
                           end
                       end;
   if(FL=1)or(FL=3) then ay:=Ly*10/(y2-y1)
                  else begin
                         ay:=ld*10 ; Cycle(y1,y2,ycycle); logy1:=log(y1); Ly:=ld*ycycle;
                         if Ly>150 then
                            begin
                              writeln(' Y-axis cycle is ',ycycle,' Reduce length !');
                              OK:= false
                            end
                       end
 end;{ if ok }
 if OK then begin
             write(' Continue (Y/N) ? '); readln(OK)
           end ;
 if not OK then FL:=5
until OK ;
   bx:=Ox*10 ; by:=Oy*10 ;
```

```
      writeln(auxout,'M',Ox*1Ø,Oy*1Ø,chr($Ød),'LØ');
      writeln(auxout,'IØ',Ly*1Ø,Lx*1Ø,Ø,Ø,-Ly*1Ø,-Lx*1Ø,Ø); writeln(auxout,'RØ,Ø')
    end ; {Window}
{************ Function Px *** calc. integer value for plot **********************}
  Function Px( xvalue : real):integer ;
    begin
      if (FL=1)or(FL=2) then Px := round(ax * (xvalue-x1) + bx)
                        else Px:=round(ax*(log(xvalue)-logx1)+bx)
    end ; { Px }
  Function Py( yvalue : real ) : integer ;
    begin
      if (FL=1)or(FL=3) then Py := round(ay * (yvalue-y1) + by)
                        else Py:=round(ay*(log(yvalue)-logy1)+by)
    end ; { Py }
{************* Procedure Xtic  and  Ytic ***********************}
  Procedure Xtic(var xn : real ;var len : integer ) ;
    begin
      writeln(auxout,'M',Px(xn):4,',',Py(yØ):4) ; writeln(auxout,'IØ,',len:3)
    end ;
  Procedure Ytic(var yn : real ;var len : integer ) ;
    begin
      writeln(auxout,'M',Px(xØ):4,',',Py(yn):4) ; writeln(auxout,'I',len:3,',Ø')
    end ;
{*********** Procedure Logtic ****************************}
  Procedure Logtic( LL,ncycle : integer ) ;
    var J1,J2,length,len : integer ;
        rØ,tic : real ;
    begin
      length:=1Ø ;
      if LL=1 then rØ:=x1 else rØ:=y1 ;
      for J1:=1 to ncycle do
        begin
          for J2:=1 to 9 do
            begin
              if J2=1 then len:=length*2 else len:=length ;
              tic :=rØ*J2*(1Ø**(J1-1)) ;
              if LL=1 then Xtic(tic,len) else Ytic(tic,len)
            end
         end
      end ;
{************ Procedure Line(var xa,ya,xb,yb : real);************}
  Procedure Line( xa,ya,xb,yb : real );
    begin
      writeln(auxout,'M',Px(xa),Py(ya)); writeln(auxout,'D',Px(xb),Py(yb))
    end ;
{************* PROCEDURE PLTEX ********************************}
  Procedure PLTEX(rx,ry : real; ex,iflag : integer );
```

```
    begin
      iflag:=iflag-1 ;{ Ø=xaxis , 1=y axis}
      writeln(auxout,'S3');
      if iflag=Ø then writeln(auxout,'M',Px(rx)+5,Py(ry)-45)
                 else writeln(auxout,'M',Px(rx)-1Ø,Py(ry)+5);
      writeln(auxout,'P1Ø');writeln(auxout,'S1');
      if iflag=Ø then writeln(auxout,'RØ,25')
                 else writeln(auxout,'R-25,Ø');
      writeln(auxout,'P',ex:1); writeln(auxout,'S3')
   end ;
{************************************************************}
  Procedure PEXX ;
    var ncycle,I1 : integer ;
        rx,ry : real ;
    begin
      I1:=round(logx1) ; ncycle:=xcycle+1; rx:=x1 ; ry:=yØ ;
      repeat
         PLTEX(rx,ry,I1,1); rx:=rx*1Ø.Ø ; I1:=I1+1 ; ncycle:=ncycle-1
      until ncycle=Ø
    end ;
  Procedure PEXY ;
    var ncycle,I1 : integer ;
        rx,ry : real ;
    begin
      I1:=round(logy1); ncycle:=ycycle+1 ; rx:=xØ ; ry:=y1 ; writeln(auxout,'Q1');
      repeat
        PLTEX(rx,ry,I1,2); ry:=ry*1Ø ; I1:=I1+1 ; ncycle:=ncycle-1
      until ncycle=Ø ;
      writeln(auxout,'QØ')
    end;
{***************** PROCEDURE PLNUM ******************************}
  Procedure PLTNUM( rx,ry,number:real;iflag:integer );
    begin
      iflag:=iflag-1 ; writeln(auxout,'Q',iflag); writeln(auxout,'S3');
      if iflag=Ø then
         begin
           writeln(auxout,'M',Px(rx)+5,Py(ry)-45)
         end
         else begin
           writeln(auxout,'M',Px(rx)-1Ø,Py(ry)+5)
         end;
      if ((abs(number)>Ø.1) and (abs(number)<99.Ø)) or(number=Ø.Ø)
                        then writeln(auxout,'P',number:4:1)
                        else writeln(auxout,'P',number)
   end;
{***************** Inrange ************************************}
  Procedure Inrange(a,b:real;var OK : boolean );
```

```
  begin
    if ((1.001*a<x1)or(0.999*a>x2)or(1.001*b<y1)or(0.999*b>y2))
        then OK:=false else OK:=true
  end;
{************ Procedure Linaxis ************************************}
 Procedure Linaxis ;
  var length : integer ; dx,dy,xn,yn: real ;
      xtitle,ytitle,mtitle:string ;
  begin
    length := 10 ; write(' Origin at lower left corner (Y/N) ? '); readln(ch);
    if (ch='Y')or(ch='y')then begin
                               x0:=x1 ; y0:=y1
                             end
        else begin
          repeat
            write(' input the origin (x0 y0) ? ') ; readln(x0,y0); Inrange(x0,y0,range);
            if not range then writeln(' range  X : ',x1,x2,'  Y : ',y1,y2)
          until range
        end;
    if (FL=1) or (FL=2) then { lin-x }
      begin
        write(' input tick distance of the X-axis ? '); readln(dx) ;
        xn:=x0-dx*trunc((x0-x1)/dx);
        if (y1<>y0)and(y2<>y0) then Line(x1,y0,x2,y0) ;
        while xn<x2 do
          begin
            if (xn<>x0)and(xn>x1) then Xtic(xn,length) ; xn:=xn+dx
          end;
          if x0<>x1 then PLTNUM(x1,y0,x1,1);
          PLTNUM(x0,y0,x0,1);
          if x2<>x0 then PLTNUM(x2,y0,x2,1)
       end
       else begin { Log X axis }
         if (y1<>y0)and(y2<>y0) then Line(x1,y0,x2,y0);
         Logtic(1,xcycle); PEXX
       end ;
    if (FL=1)or(FL=3) then  { lin Y axis }
     begin (if lin-Y }
       write(' input tick distance of Y-axis ? '); readln( dy );
       yn:=y0-dy*trunc((y0-y1)/dy) ;
       if (x1<>x0)and(x2<>x0) then Line(x0,y1,x0,y2) ;
       while yn<y2 do
         begin
           if(yn<>y0)and(yn>y1) then Ytic(yn,length); yn:=yn+dy
         end;
         if y0<>y1 then PLTNUM(x0,y1,y1,2);
         PLTNUM(x0,y0,y0,2);
```

```
        if y2<>yØ then PLTNUM(xØ,y2,y2,2)
      end {lin Y}
      else begin { Log Y axis}
        if (x1<>xØ)and(x2<>xØ) then Line(xØ,y1,xØ,y2) ;
        Logtic(2,ycycle); PEXY
      end;
      writeln(auxout,'RØ,Ø'); write(' input X-axis title '); readln(xtitle);
      write(' input Y-axis title '); readln(ytitle);
      write(' input main title '); readln(mtitle);
      if xtitle<>'' then
          begin
            writeln(auxout,'QØ'); writeln(auxout,'M',Px(xØ)+2ØØ,Py(yØ)-12Ø);
            writeln(auxout,'P',xtitle)
          end;
      if ytitle<>'' then
          begin
            writeln(auxout,'M',Px(xØ)-1ØØ,Py(yØ)+1ØØ);
            writeln(auxout,'Q1',chr($ØD),'P',ytitle); writeln(auxout,'QØ')
          end;
      if mtitle<>'' then
          begin
            writeln(auxout,'S5'); writeln(auxout,'M',Px(xØ)+1ØØ,Py(y2)+1ØØ);
            writeln(auxout,'P',mtitle,chr($ØD),'S3')
          end;
    end ; {Linaxis}
{****************************************************************}
   Procedure Interpolate(point:integer; var xint,yint:real);
      var xp1,xp2,yp1,yp2 : real ;
      begin
          xp1:=X[point] ; xp2:=X[point+1] ;
          yp1:=Y[point] ; yp2:=Y[point+1] ;
          if (yp1>y2) or (yp2>y2) then yint:=y2
                                  else yint:=y1 ;
          xint:=(xp2-xp1)*(yint-yp1)/(yp2-yp1) + xp1
      end; { interpolate}
{******************************************************************}
  begin { plot }
   if first then
      begin
        Window; Linaxis ; fname:='1.WPLOT.DAT'; rewrite(name,fname);
        writeln(name,FL); writeln(name,ax,bx,ay,by); writeln(name,x1,x2,y1,y2); close(name)
      end
      else begin
        fname:='1.WPLOT.DAT'; open(name,fname,input); readln(name,FL); readln(name,ax,bx,ay,by);
        readln(name,x1,x2,y1,y2); close(name); logx1:=log(x1) ; logy1:=log(y1)
      end;
    writeln(auxout,'H'); writeln(' Change pen color, then');
```

```
    write(' input line type, 0=solid, 1..4=dot, 5..6=one point ,', '7..8=two point, -1=no line ');
    readln(N); I:=0;
    repeat
       I:=I+1; Inrange(X[I],Y[I],range)
    until range ;
    I1:=I;
    if N>=0 then
      begin
        writeln(auxout,'B40',chr($0D),'L',N:1); writeln(auxout,'M',Px(X[I]),Py(Y[I]) );
        I:=I+1;
        repeat
          if break then exit ;
          xi:=X[I] ; yi:=Y[I] ; Inrange(xi,yi,range);
          if range then writeln(auxout,'D',Px(xi),Py(yi))
                   else begin
                          Interpolate(I-1,xm,ym);
                          writeln(auxout,'D',Px(xm),Py(ym)); abort:=false ;
                          repeat
                            I:=I+1;
                            if (X[I]>x2) or (I>Npoint) then abort:=true;
                            Inrange(X[I],Y[I],range)
                          until range or abort ;
                          if not abort then
                            begin
                              Interpolate(I-1,xm,ym); writeln(auxout,'M',Px(xm),Py(ym));
                              writeln(auxout,'D',Px(X[I]),Py(Y[I]))
                            end
                        end;
          I:=I+1
        until (X[I]>x2)or(I>Npoint);
        I2:=I-1;
       end; {if N>=0 }
    writeln(auxout,'H'); write(' plot mark ? 0 = no mark ');
    readln(N);
    if N>0 then
       begin
         I:=I1;
         repeat
           if break then exit ;
           Inrange(X[I],Y[I],range);
           if range then writeln(auxout,'M',Px(X[I]),Py(Y[I]),chr($0D),'N',N:2);
           I:=I+1
         until (I>Npoint)or(X[I]>x2)
       end; {N>0}
      writeln(auxout,'H')
   end ; { plot }
MODEND.
```

APPENDIX D

PASCAL PROGRAM FOR CONDUCTING SPHERE IN FREE SPACE (FREQUENCY DOMAIN)

```
 {$B+  calculate frequency domain response of sphere }
Program FSPHERE(input,output,auxout);
 {$B-} { for break }

 type complex = record
                      re : real ;
                      im : real
                end; {record}
     ary = array[1..1024] of real ;
 var D,Hz,K,X     : complex ;
     Q1,Q2,Q3,Q4 : complex ;
     r1,r2,r3,r4 : real ;
     s : real ;     { consuctivity of sphere }
     Mt,Mr : real ; { moment }
     Rt :  real ;   { radius of T-loop }
     z  : real  ;   { depth of the sphere }
     mu,omega,freq,Freq1,Freq2,Lt,H0 : real ;
     Npoint,Nt,I : integer ;
     a : real ; { radius of sphere }
     continue,first : boolean ;
     df,coeff : real ;
     Qi,SQRI : complex ;
     FRQ,REHZ,IMHZ : ary ;
{**********************************************************************}
 Procedure PLOT(X,Y:ary;Npoint,FL:integer;first:boolean);external;
 Procedure PLTSET ; external ;  { initialize digital plotter }
{**********************************************************************}
 Function CMPLX( realpart,imagipart : real ): complex ;
   begin
     CMPLX.re := realpart ; CMPLX.im := imagipart
   end ;
{**********************************************************************}
 Function CRMUL( z1 : complex ; realnumber : real ):complex ;
   begin
     CRMUL.re :=z1.re * realnumber ; CRMUL.im :=z1.im * realnumber
   end ;
{**********************************************************************}
 Function CADD(z1,z2 : complex):complex ;
  begin
    CADD.re:=z1.re+z2.re ; CADD.im:=z1.im+z2.im
  end;
{**********************************************************************}
```

```
 Function CSUB(z1,z2 : complex):complex ;
  begin
    CSUB.re:=z1.re-z2.re ; CSUB.im:=z1.im-z2.im
  end;
{*********************************************************************}
 Function CMUL(z1,z2: complex):complex ;
  begin
     CMUL.re:=z1.re*z2.re-z1.im*z2.im ;
     CMUL.im:=z1.re*z2.im+z1.im*z2.re
  end;
{*********************************************************************}
 Function CABS(z1:complex):real  ;
   begin
     CABS:=sqrt( sqr(z1.re) + sqr(z1.im) )
   end;
{*********************************************************************}
 Function CDIV(z1,z2:complex):complex;
  var r : real ;
  begin
     if abs(z1.re) > 1e7 then
        begin
          z1.im:=z1.im/z1.re ; z2.re:=z2.re/z1.re ; z2.im:=z2.im/z1.re ; z1.re:=1.0
        end;
     r := sqr(z2.re)+sqr(z2.im) ;
     CDIV.re:=(z1.re*z2.re+z1.im*z2.im)/r ; CDIV.im:=(z1.im*z2.re-z1.re*z2.im)/r
  end;
{*********************************************************************}
 Function CINV(z1:complex):complex ;
  var r : real ;
  begin
    r :=sqr(z1.re) + sqr(z1.im);
    CINV.re:=z1.re/r ; CINV.im:=-z1.im/r
  end;
{*********************************************************************}
 Function CEXP(z1:complex):complex ;
  begin
    CEXP.re:=cos(z1.im)*exp(z1.re) ; CEXP.im:=sin(z1.im)*exp(z1.re)
  end;
{*********************************************************************}
 FUNCTION CCOTH(z1: complex):complex ;
    var z2,z3,z4 : complex ;
   begin
     z2.re:=-z1.re ; z2.im:=-z1.im ;
     z3 := CADD(CEXP(z1),CEXP(z2)) ; z4 := CSUB(CEXP(z1),CEXP(z2));
     CCOTH := CDIV(z3,z4)
   end;
{*********************************************************************}
```

```
 Function Dx(z1:complex ): complex ;
   var z2 : complex ;
   begin
    z2:=CSUB(CCOTH(z1),CINV(z1)); z2 := CDIV(z2,z1);
    Dx.re:=3.0*z2.re - 1.0 ; Dx.im:=3.0*z2.im
   end;
{************************ MAIN ****************************************}
  begin { MAIN }
   mu:=pi*4.0E-7 ;
   a := 0.052 ; { radius 5cm }
   Mr :=0.65;   { moment of receiver coil }
   Lt := 0.8 ;  { 0.8 x 0.8 m x 100 turn transmitter loop }
   Rt := Lt/sqrt(pi) ;
   Freq1 :=10.0 ; { Min freq to calcu }
   Freq2 :=10000. ;
   Npoint := 100 ; { 100 point to calcu }
   SQRI := CMPLX( 1.0/sqrt(2) , 1.0/sqrt(2) );
   Nt := 100 ; Mt := sqr(Lt)*Nt ;
   Qi := CMPLX(0.0 , 1.0);  { imaginary number unit }
   repeat
    writeln ; writeln(' Frequency Domain Responce of Conductive Sphere ');
    writeln;write(' input depth and conductivity '); readln( z,s );
    H0 := Mt/ (2.0*pi*sqrt(Rt*Rt + z*z)); coeff :=((a/z)**3 ) * H0 ;
    df:=(Freq2/Freq1)**(1.0/(Npoint-1)) ;
    writeln('  freq      Hz.real        Hz.imag  ');
    freq:=Freq1 ; r1 := sqrt(s*mu) ;
    for I:=1 to Npoint do
      begin
         if break then exit;
         omega:=2.0 * pi * freq ;
         K := CRMUL(SQRI,r1*sqrt(omega)); { k=sqrt(i*s*mu*omega)}
         X := CRMUL( CMUL(K,Qi) , a) ; { x=i*k*a }
         D := Dx(X) ; Hz:= CRMUL(D,coeff); FRQ[I]:=freq; REHZ[I]:=-Hz.re ; IMHZ[I]:=Hz.im ;
         writeln( freq,Hz.re, Hz.im );
         freq:=freq*df
      end;
      write(' Plot the data ? '); readln(continue);
      if continue then
        begin
         PLTSET;
         write(' first plot ? ') ; readln(first) ;
         PLOT(FRQ,REHZ,Npoint,5,first);
         PLOT(FRQ,IMHZ,Npoint,5,false);
        end;
    write(' Continue ? ');readln(continue)
  until not continue
end.
```

APPENDIX E

PASCAL PROGRAM FOR CONDUCTING SPHERE IN FREE SPACE

(TIME DOMAIN)

```
Program TSPHERE(input,output,auxout);
   { Calculation of time domain response of conducting sphere }

  type ary = array[1..1024] of real;
  var Mt,Mr,B0,alpha,d1 : real;
      a,z,L1,L2,Rt,emf : real;
      sigma,t,mu : real ;
      I,J,K,Npoint : integer ;
      x,y   : ary ;
      first,plt,continue : boolean;
      t1,t2,dt,coef : real;
{*********************************************************************}
Procedure PLTSET; external;
Procedure PLOT(var x,y : ary;Npoint,fl:integer;first:boolean);external;
{*********************************************************************}
  Function Esum(time : real ):real;
    var sum,u,w: real;
        i  : integer;
    begin
      sum:=0.0; i:=1; u:=-pi*pi*alpha*time;
      repeat
         w:=u*sqr(i); sum:=sum+exp(w); i:=i+1
      until w<-20. ;
      Esum:=sum
    end;
{*********************************************************************}
  Procedure  Getparameter;
    const  current=1.0;
    var flg : char;
        Nt : integer;
    begin
      write(' enter conductivity of the sphere (mho/m) : ');readln(sigma);
      write(' enter radius and depth of the sphere (m) : ');readln(a,z);
      repeat
        write(' Circuler t-loop (C), or Rectanguler t-loop ( R) '); readln(flg)
      until flg in ['R','C'];
      if flg='C' then
        begin
           write(' input radius of transmitter loop : '); readln(Rt)
        end
        else begin { Rect. loop }
          write(' enter L1 and L2 of transmitter loop : '); readln(L1,L2);
```

```
        Rt:=sqrt(L1*L2/pi)
      end;
    write(' how many turns ? '); readln(Nt);
    Mt:=pi*sqr(Rt)*current*Nt; mu:=pi*4.0e-7; alpha:=1.0/(sigma*mu*sqr(a));
    B0:=Mt*mu/(2.0*pi*((sqr(z)+sqr(Rt))**1.5));
    write(' enter Moment of receiver loop Mr : ');readln(Mr);
    d1:=6.0*a*sqr(a)/(sqr(pi)*sqr(z)*z)
  end;
{*********************************************************************}
 Procedure Savedisk;
  var NP : integer ;
      title,fname: string;  name : text ;
  begin
    write('  enter file name ? '); readln(fname);
    write('  enter a comment line : ');readln(title);
    rewrite(name,fname);
    writeln(name,fname); writeln(name,title); writeln(name); writeln(name,'$DATA');
    for NP:=1 to Npoint do writeln(name,x[NP],y[NP]);
    close(name)
  end ;{ Savedisk}
{******************************* MAIN ********************************}
 begin { main }
  repeat
    writeln; writeln(' Time domain response of conducting sphere');
    Getparameter;
    write('  enter time range to calculate (millseconds) : '); readln(t1,t2);
    t1:=t1/1000 ; t2:=t2/1000;
    write('  enter time incliment in milliseconds   : '); readln(dt);
    dt:=dt/1000; coef:=B0*Mr*alpha*d1*sqr(pi); t:=t1; J:=1;
    repeat
      emf:=coef*Esum(t); writeln(t,emf);
      x[J]:=t*1000 ; y[J]:=emf; { X MILLISEC,  Y VOLTS/1A}
      J:=J+1; t:=t+dt
    until t>t2;
    Npoint:=J-1;
    write(' Plot (Y/N) ? '); readln(plt);
    if plt then
      begin
        PLTSET;
        write(' first plot (Y/N) ? '); readln(first);
        PLOT(x,y,Npoint,5,first)
      end;
    write(' Save data to disk (Y/N) ? ');readln(plt);
    if plt then Savedisk;
    write(' Continue ? ');readln(continue)
  until not continue
 end.
```