111

GROUND-BASED OBSERVATIONS IN SUPPORT OF THE ACTIVE SATELLITE MISSION

FINAL REPORT

FRANK T. BERKEY
PRINCIPAL INVESTIGATOR

1 JANUARY 1990 - 30 APRIL 1992

UTAH STATE UNIVERSITY
CENTER FOR ATMOSPHERIC AND SPACE SCIENCES
LOGAN, UTAH 84322-4405

NASA RESEARCH GRANT NAG 5-1341

(NASA-CR-190125) IONOSPHERIC GRAVITY WAVE MEASUREMENTS WITH THE USU DYNASONDE Final Report, 1 Jan. 1990 - 30 Apr. 1992 (Utah State Univ.) 111 p

N92-27434

Unclas
G3/46 0091245

ACKNOWLEDGMENTS

I would like to express my sincere appreciation to Dr. Gardiner S. Stiles, my major professor, for his patient guidance, thoughtfulness, and financial support. The trust he has placed in me throughout this research allowed this project to process soundly.

I would like to thank Dr. Frank T. Berkey, for his extensive aid, advice, and financial support. He read every revision of this report and made many valuable suggestions and corrections.

I would like to thank my graduate committee members, Dr. Richard W. Harris, the head of the department, Dr. Ronney, D. Harris, and Dr. Robert W. Gunderson for their critical review of this report.

I am deeply in debted to my parents, whose faith and inspiration have given me the strength to persevere. I would also like to thank my dear wife and son for their patience and love in fulfilling this report.

Jun Yuan Deng

This research has been supported in part by subcontract 90-184 from the University of Illinois (NSF ATM90-00171) and by NASA contract NAG 5-1341.

CONTENTS

LIST OF FIGURES

Figures Pages

LIST OF TABLES

ABSTRACT

IONOSPHERIC GRAVITY WAVE MEASUREMENTS

WITH THE USU DYNASONDE

by

Jun Yuan Deng, Electrical Engineer

Utah State University, 1991

Major Professor: Dr. Gardiner S. Stiles

Department: Electrical Engineering

A method for the measurement of ionospheric Gravity Waves (GW) using the Utah State University (USU) Dynasonde is outlined in this report. This method consists of a series of individual procedures, which includes functions for data acquisition, adaptive scaling, polarization discrimination, interpolation and extrapolation, digital filtering, windowing, spectrum analyses, GW detection and graphics display. Concepts of system theory are applied to treat the ionosphere as a system. An adaptive ionogram scaling method has been developed for automatically extracting ionogram echo traces from noisy raw sounding data. The method uses the well known Least Mean Square (LMS) algorithm to form a stochastic optimal estimate of the echo trace which is then used to control a moving window. The window tracks the echo trace, simultaneously eliminating the noise and interference. Experimental results show that the

proposed method functions as designed. Case studies which extract GW from ionosonde measurements have been carried out using the techniques described in this report. Geophysically significant events have been detected and the resultant processed results are illustrated graphically. This method has also been developed with the goal of real time implementation in mind. Execution time is evenly distributed between functional blocks so that a parallel computation can be easily implemented with the use of a pipeline mechanism.

CHAPTER I

INTRODUCTION

Since the introduction of digital ionosondes in 1960s, ground-based sweep frequency radars have been significantly improved. By means of such instruments, investigations of the relationship between radio wave propagation and the characteristics of the ionosphere, and the relationship between the variations or irregularities of the ionosphere and the relative change of the echo parameters, have been dominant in the area of upper atmosphere physics. The rapid development of VLSI during the 1970s has made it possible for modern digital radars to deal with multi parameter ionogram data and to extract the geophysically important ionospheric characteristics. With the help of these modern techniques, exploring the mystery of the Traveling Ionospheric Disturbance (TID) and ionospheric Gravity Waves (GW) have been made easier. Many theoretical and observational results have been achieved and several comprehensive reviews [Yeh and Liu, 1974; Testud, 1973; Hines, 1974; Francis, 1975; Hunsucker, 1982] and books [Beer, 1974; Hines et al., 1974; Gossard and Hooke, 1975] have been published in this area.

Newly developed Ionogram autoscaling techniques have dealt with predictions of smooth previous ionograms [Wright, 1972] and function approximation [Reinisch and Huang, 1983]; recently, they have dealt with pseudo-trace and multiparameters [Fox, 1988].

1.1 IONOSPHERIC GRAVITY WAVES

The phenomenon of the ionospheric gravity wave has been recognized as the manifestation of Traveling Ionospheric Disturbances. It has been widely and comprehensively studied in both the theoretical and experimental aspects. In general, GW's occur in three distinct forms: large scale, medium scale, and small scale. Large scale waves have periods of 30 minutes or longer with horizontal wavelengths of many hundreds to thousands of kilometers. Their phase and group velocities vary from about 300 to 900 meters per second. Medium scale waves have periods of about 5 to 30 minutes with less than a few hundred kilometers wavelength. Their phase and group velocities are less than 300 meters per second. Small scale waves have periods of about 2 to 5 minutes and phase and group velocities of about 300 to 3000 meters per second.

Studies indicate that GW generation is related to the Lorentz force and Joule heating. Many sources have been discovered, such as volcanic explosions, tropospheric turbulence, large man-made explosions, auroral phenomena, convective thunderstorms, etc. Other possible sources, such as the electron density irregularities, eclipses, and the Harang discontinuity [Maynard, 1974] have also been reported. Gravity Waves propagate from the source to other parts of the ionosphere as soon as they are generated. Large scale GWs usually have large amplitudes and propagate relatively farther [Richmond, 1978]. It has also been reported that heat conduction is usually more important than Joule force in attenuating GWs.

The most recent areas of ionospheric GW research are: (1) GW and TID cause-effect relations; (2) determination of the direction of propagation of TIDs; (3) "Fine-structure" of GW sources such as the duration and quantification of energy input into the ionosphere; (4) Relation and interaction of GWs generated in the ionosphere and middle atmosphere; (5) Effects of TIDs upon radio propagation; (6) Global and multi-technique investigations of the behavior of

ionospheric GWs, and (7) New techniques and methods in GW measurements.

1.2 TOOLS OF IONOSPHERIC GW MEASUREMENT

Many instruments are available to measure and observe the ionospheric GWs. The most common used methods are satellite and rocket sensors, ionosondes, incoherent radar, coherent radar, partial reflection radar, magnetometers, and photometric observation instruments (for thunderstorms).

Rockets and satellites with remote sensors and other space instruments provide the advantage of studying the behavior of the ionosphere in-situ and in horizontal directions. They have also greatly enhanced our knowledge of ionospheric dynamics, but they are limited in use because of their high cost.

The ground-based ionospheric sounding method is, perhaps, still one of the most important tools. It is able to measure all the observables contained in electromagnetic signals reflected from the ionosphere. There are four major techniques used for ground-based radio sounding. They are partial reflection, ionosondes, coherent radar, and incoherent radar. Each technique is based on a different reflection mechanism and has its own objective. (1) The partial reflection method is used to determine the D region electron density profile and to investigate horizontal stratified irregularities. The partial reflection measurement is used mainly to compare the amplitude and phase differences of the O-mode and X-mode traces. In general, partial reflection transmitters use 50-100kw peak power with a pulse width of 25μs, and a large aperture receiving antenna array. (2) The ionosonde, in contrast to the partial reflection technique, scans a large frequency band from below 1 MHz up to 20 MHz. It measures the ionization structure

from the E region to the peak of the F region. Most ionosondes use 50-600μs pulses for oblique and vertical sounding and transmitter peak powers between 1 and 10 kw. (3) The coherent radar is sensitive to coherent echoes from quasi-deterministic irregularity structures and is extremely sensitive normal to the geomagnetic field. Coherent radars usually work at high frequency (HF) or very high frequency (VHF) from 10 to 100 MHz. In the high latitude area, HF frequencies above the F layer critical frequency can be reflected if the radio wave is perpendicular to the magnetic field. For the E region, a VHF beam can be reflected perpendicular to the magnetic field. (4) The incoherent radar technique is more powerful than any of the above three techniques. It can measure electron density, electron and ion temperatures, ion compositions and ion and electron velocities in the E and F regions. Incoherent radars usually work at frequencies between 50 and 1000 MHz. Because of their extremely high cost, only a few incoherent radar facilities have been built.

Ground-based ionosondes armed with modern computer control and digital data processing techniques are able not only to measure the echo parameters such as range, angle of arrival, amplitude, phase, wave polarization, and doppler frequency, but are also to work globally with network, remote control, automatic electron density profile calculations, and centralized data collection.

1.3 THE OBJECTIVE OF THIS REPORT

The objective of this report is to design a feasible scheme to measure ionospheric GWs with the USU dynasonde. This report begins with a review of recent ionospheric GW studies, as well as the USU dynasonde configuration and its characteristics. From a system theory point of

view, the author views the ionosphere as an unknown system so that the system modeling and identification method is applied to investigate the ionosphere. By this method an adaptive ionogram scaling method is examined. This method, based on random signal processing theory, uses the well known Least Mean Square (LMS) algorithm to automatically track the radar echo traces and then extracts the O-mode and X-mode traces from the raw radar data. The theoretical analyses and the practical implementation of this method are discussed in detail. Case studies of the adaptive scaling application in the GW measurement process are carried out. After the adaptive scaling process, a full ionospheric GW detection scheme is designed. It includes the O-mode and X-mode trace separation, echo trace interpolation, digital filtering, h'(f) profile generation, power spectrum analyses and graphics presentation. Finally, the report ends with a discussion of some observations to demonstrate the capability of the designed scheme of GW measurement. It is worth while to point out that the designed scheme can be easily implemented in real time. Functional blocks can be processed in parallel with equal execution time in a pipeline fashion.

CHAPTER II

IONOSPHERIC GRAVITY WAVES AND THEIR PROPERTIES

It was discovered in the 1950s that a variety of waves exist in the atmosphere and ionosphere, of which the Gravity Wave is one. There are many known sources causing GWs, such as volcanic explosions, tropospheric turbulence (jet streams), auroral phenomena, large man-made explosions and thunderstorms. Once generated, the Gravity Waves carry with them energy and momentum from the source to other parts of the atmosphere. The literature shows that ionosphere electrodynamics and GWs are highly correlated [Yeh and Liu, 1974; Testud, 1972; Hines, 1974; Francis, 1975; Gossard and Hooke, 1975; Beer, 1974]. When a GW propagates through the ionosphere, it sweeps the ionization into a wavelike distribution through collisions between the charged and neutral particles. Generally, GWs occur in the ionosphere in three distinct forms: large scale, medium scale, and small scale. They are different in wavelength, period, and speed of propagation (see Table 2.1).

In order to understand GW, knowledge of the generation (source), propagation (period, direction, speed), and dissipation are essential.

2.1 GENERATION

Several formulas for the source functions have been presented in the literature [Chimonas and Hines, 1970; Kamide and Brekke, 1975]. Because of the complexity of the source and the events in the upper atmosphere, one has to make several assumptions to simplify his derivation. It has been established that the most likely high latitude ionospheric GW sources are: (1) Joule

heating and Lorentz forces associated with the auroral electrojet and; (2) intense particle precipitation events [Hunsucker, 1982]. Richmond [1978] also pointed out that thermospheric GW energies are generated either by the Lorentz acceleration interacting with a wind velocity in the same direction or else by a heat source interacting with a perturbation in the scale height.

Table 2.1 Properties of ionospheric GWs

Nomenclature	Horizontal velocity	Period	Wave length
large scale	400-1000m/s	30 min to 3 hours	>1000km
Medium scale	1000-250m/s	15 min to 1 hour	Several Hundred km
Small scale	300-3000m/s	2-5 min	

Beer [1974] also pointed out that F region TIDs can be categorized into three separate classes: (1) Very large types which usually follow magnetic storms, (2) medium scale TIDs which are common during the day, and (3) a type due to incoherent superposed gravity waves propagating to ionospheric heights from below.

It is also possible that electron density irregularities moving at supersonic speeds in the F region might, under certain conditions, generate GWs [Hunsucker, 1982].

In the analyses of GW generation, Richmond [1978] reached the following conclusions: (1) The amount of gravity wave energy generated is proportional to the square of the amplitude

of either the heat input or the momentum input of the source, so large amplitude disturbances are particularly effective in generating GW energy.

(2) The influence of the vertical distribution of the source on the spectral composition of vertical wavelengths generated can be estimated with reasonable accuracy, even in the absence of precise observations of the source parameters.

(3) The influence of the horizontal dimension and temporal scale of the source on the spectrum of GWs generated can be rather complex.

(4) The efficiency of GW energy generation associated with currents is independent of the electric field strength for the Lorentz force contribution, but is proportional to the horizontal component of the electric field for the Joule heat contribution.

(5) The relative importance of the Lorentz force to Joule heating in generating gravity waves depends primarily on the electric field strength.

It has also been shown that oscillations in electron and ion temperature or electron density and plasma velocity oscillations are manifestations of GWs [Testud, 1973].

2.2 PROPAGATION

The most recent theoretical advances in understanding GW propagation have been made by Testud [1972], Yeh and Liu [1974], Hines [1974], Francis [1975], and Richmond [1978]. The hypothesis that TIDs are the ionospheric manifestations of GWs has received nearly complete acceptance and often observations of TIDs are used to make statements about atmospheric wave generation and propagation. Early in 1975, Francis discussed the propagation mechanism for both large and medium scale TIDs produced by auroral zone sources. He predicted that the

atmospheric responses to an auroral zone source mechanism should consist of a discrete spectrum of guided modes (large scale TIDs) and a continuous spectrum of freely propagating internal waves (medium scale TIDs) in the upper atmosphere. The freely propagating internal waves may exist at any altitude. Francis also predicted that the average fluctuations of the auroral electrojet are sufficient to generate freely propagating GWs which should be detectable at large distances as TIDs. Although he went into considerable detail to investigate TID propagation in the middle High Latitude ionosphere, Francis points out because of the complexities of the propagation medium and source characteristics for the TIDs and GWs, it would still be a rather formidable problem.

In his conclusion, Richmond [1978] states that: (1) ray paths of GWs in the thermosphere tend to curve upward for horizontal wave velocities greater than 200m/s. Downward propagating waves are reflected upward in the lower thermosphere if their horizontal component of velocity is greater than 250m/s. (2) From a given source point to a given observation point, short-period GWs travel fast, and long-period GWs travel more slowly.

2.3 DISSIPATION

Different scales of ionospheric GW have different dissipative behavior. In the upper atmosphere wave amplitudes may be large but at the same time dissipative effects are also large. This has been shown by a variety of observational results that at any one height one particular gravity wave is dominant, but at higher or lower heights waves with some other frequency or wave number dominate.

Francis [1975] concluded that: (1) dissipative effects progressively remove more and more of the slower moving and shorter-period waves as the observation point is moved farther and farther from the source. Thus, long-period waves with high horizontal velocities are able to propagate farthest in the thermosphere before being dissipated; (2) Viscosity and heat conduction are usually more important than Joule dissipation in attenuating GWs.

CHAPTER III

THE DYNASONDE

Although modern techniques have provided a variety of means, such as satellites and rockets, incoherent scatter radars, and coherent radars, to study the atmosphere and the ionosphere, ionosondes still play an important role in exploring the physical characteristics of the ionosphere. Rapid progress in the integrated circuit market has led to new techniques for ground-based ionospheric sounding in the measurement of all the observables: amplitude, phase, doppler, incidence angle, and polarization. The USU Dynasonde located near Garden city, Utah (41.9N 111.4W, L=2.5) is a state-of-the-art fully computer controlled digital radar system, with broad frequency span, and high transmitter power (see Fig. 3.1). This system permits full digital processing of echoes and uses two quadrature receivers for rapid measurement of echo phase. The use of two receivers fed from four antennas allows determination of the echo angle of arrival. In addition, the system can make vernier group-path measurements by transmitting two closely spaced frequencies and measuring the phase difference between the returned echoes. This method increases the radar's height resolution, allowing much more detailed measurement of ionospheric changes than obtainable by time-of-flight methods. The dynasonde consists of five main sections: the RF transmitter, receivers, the data acquisition section, the signal processing section, and the system timer and controller.

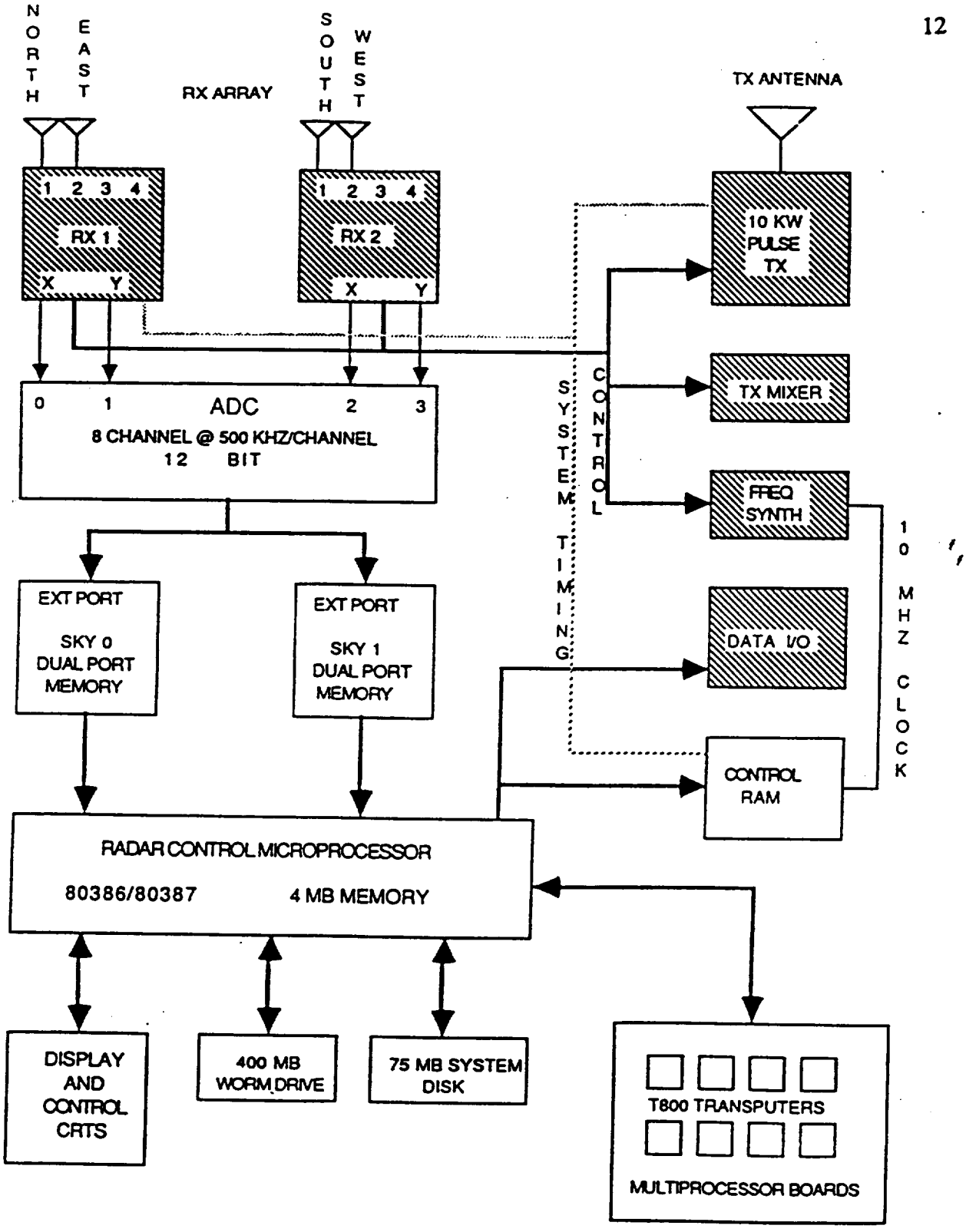


Figure 3.1 The Dynasonde system

3.1 THE TRANSMITTER

The transmitter subsystem consists of the frequency synthesizer, the TX mixer, a low power amplifier, and antennas. To create the source pulses, two oscillators work in a symmetric up and down conversion scheme. The first oscillator generates a frequency between 40.1 and 70 MHz for up-conversion of the receiver band (0.1 to 30 MHz) into an IF frequency of 40 MHz. The second oscillator at 40 MHz is down converted by the synthesizer to generate the output RF pulse with a frequency between 0.1 and 30 MHz. This oscillator also provides the reference signal for coherent quadrature detection of the received signal. A wide band solid-state driver amplifies the synthesizer's low-level output for input into the high-power amplifier, which generates a 10 kW peak pulse output to the transmit antenna with push-pull amplifier configuration.

3.2 RECEIVERS

This subsystem consists of two receivers, each of which can be multiplexed to one of four antennas. Four dipole antennas are configured in an L-shaped array, with the south-north pair and the east-west pair intersecting at the south-west corner [see Fig. 3.2]. In order to maintain accurate processing of the signals after digitization, the receiver system is linear. To obtain the necessary dynamic range, the system uses a wide band linear amplifier with a dynamic range greater than 140 dB, while a computer-controlled attenuator expands the range. The IF module is a linear amplifier with a bandwidth of 30 kHz. The quadrature outputs of the receivers are in the range of ±5V, which is compatible with the A/D input requirement.

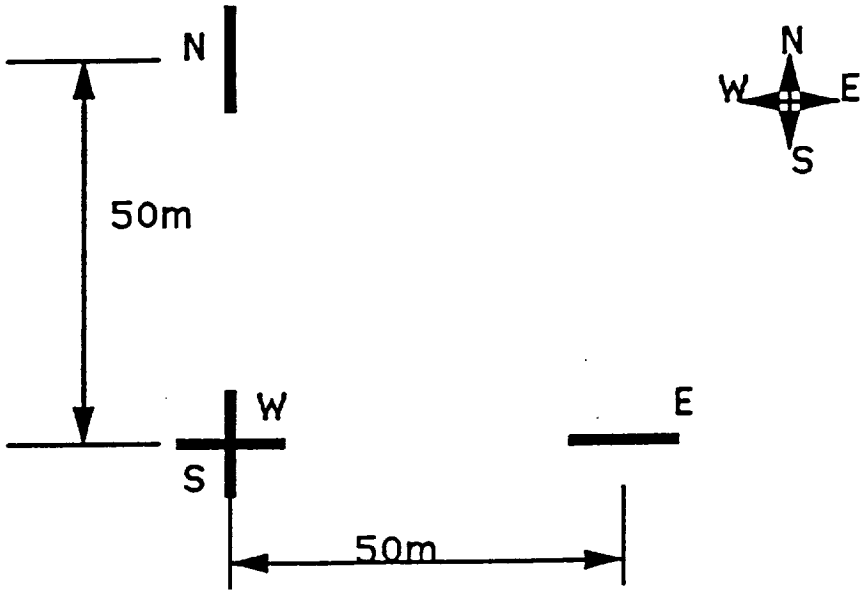


Fig. 3.2 L-Werpol receiving antenna array

Table 3.1 The pulse Configuration

Transmitted Pulse number	Time	Freq.	Receiver antenna	Phase	Receiver antenna	phase
P1	0	f	W	Φ1	S	Φ2
P2	t	f+df	W	Φ3	S	Φ4
P3	2t	f	W	Φ5	E	Φ6
P4	3t	f	N	Φ7	S	Φ8

3.3 DATA ACQUISITION AND SIGNAL PROCESSING

The A/D board is comprised of four Analog Devices Model 1204 converters, the input to each being the X or Y values from the two receivers. The converter chips digitize the ±5 volts input into 12-bit samples providing approximately 72-dB range between quantization noise and saturation. The maximum sample rate is 500KHz, which is much higher than the signal bandwidth. A 100 kHz (or 10 μs sampling interval) sampling rate is typically used in the system, so that the corresponding echo range resolution is 1.5km. Digital signal processing is carried out using two commercial TMS320 DSP boards (SKY 320). These boards use the TMS32010 chip, which features 16-bit fixed point arithmetic, a 32-bit accumulator and single-instruction 16x16 multiply. They also incorporate 64K words of dual-port memory, thus enabling both the PC and DSP boards to operate on data simultaneously.

Data from the A/D board goes to the DSP boards in DMA mode and main task of the DSP boards is to perform the data-reduction. For every single pulse transmission, the DSP boards receive four inputs, corresponding to the two receivers with two X-Y channels each. Data arrives from the A/D board at an aggregate rate of 4*100k samples/sec, usually for an interval of 5.12 msec corresponding to 720 km in range and yielding a raw sample size of 2k bytes. The Dynasonde uses a configuration of four pulses as a basic pulse set. For each pulse set there are about 8K bytes to be transferred from the DSP board to the PC. A peak-search algorithm running in the DSP boards reduces the volume of meaningful data by more than 90%. After the data reduction, a message passed through the communications register tells the PC to receive the data.

3.4 TIMING AND CONTROLS

The system timing breaks down into three distinct levels, the high speed, fast, and slow levels, depending on the speed at which operation must occur (see Fig. 3.3).

High speed system timing, such as coordinating the length and relative timing of pulse, strobing the A/D converters and turning receiver attenuators on and off, is performed on a microsecond basis and is too fast for a PC to control. Instead, the system uses a control RAM board (CRAM) which sends control vectors to the high speed hardware subsystems. The CRAM runs at a high speed after its control vectors are loaded by the PC through the I/O bus. The 10 MHz synthesizer signal is divided into 1 MHz and 100 Hz clock signals. The 100 Hz clock gates all pulse functions and keeps the pulses synchronous in time. The CRAM is armed under program control and is initiated on the next 100 Hz transition; it then clocks out control vectors with the 1 MHz clock. During the 5.12 msec that the CRAM is controlling data acquisition, the host can perform nonsynchronous tasks, such as plotting data or getting ready for the next pulse. The host PC frequently checks the status of the CRAM in order to rearm it after its completion.

Fast level actions, such as setting frequencies and attenuation, arming the CRAM, and moving data into extended memory after each pulse set, occur at the millisecond level. The fast timing uses the 100 Hz clock. This fast timing interface, known as the Data Input/Output Bus (DIO), is an 8-bit unidirectional data path, comprised of 6 address lines and two control lines.

Operating in the slow domain is the radar's supervisory program that monitors the keyboard and the clock. This program runs on the PC under DOS. It first loads the CRAM and DSP boards with the appropriate configuration information and then starts sounding. At the end the supervisor program closes the DIO ports and stores data to the hard or optical drive. Fig. 3.4

LEVEL 1: Slow (0.1 to 1 sec)

PC Radar supervisory program under DOS

- Realtime clock
- Start sounding
- Check keyboard
- Copy sounding to disk

LEVEL 2: Fast (milliseconds)

Sounding program spawned from Level 1

- Pulse-to-pulse configuration
- Set frequency, attenuation, antenna
- Store PCTs

LEVEL 3: High speed (microseconds)

Uses CRAM and DSP boards

- Start pulse
- Control digitizing
- Peak detection
- Coincidence detection

Figure 3.3 Three levels of the system timing for the Dynasonde

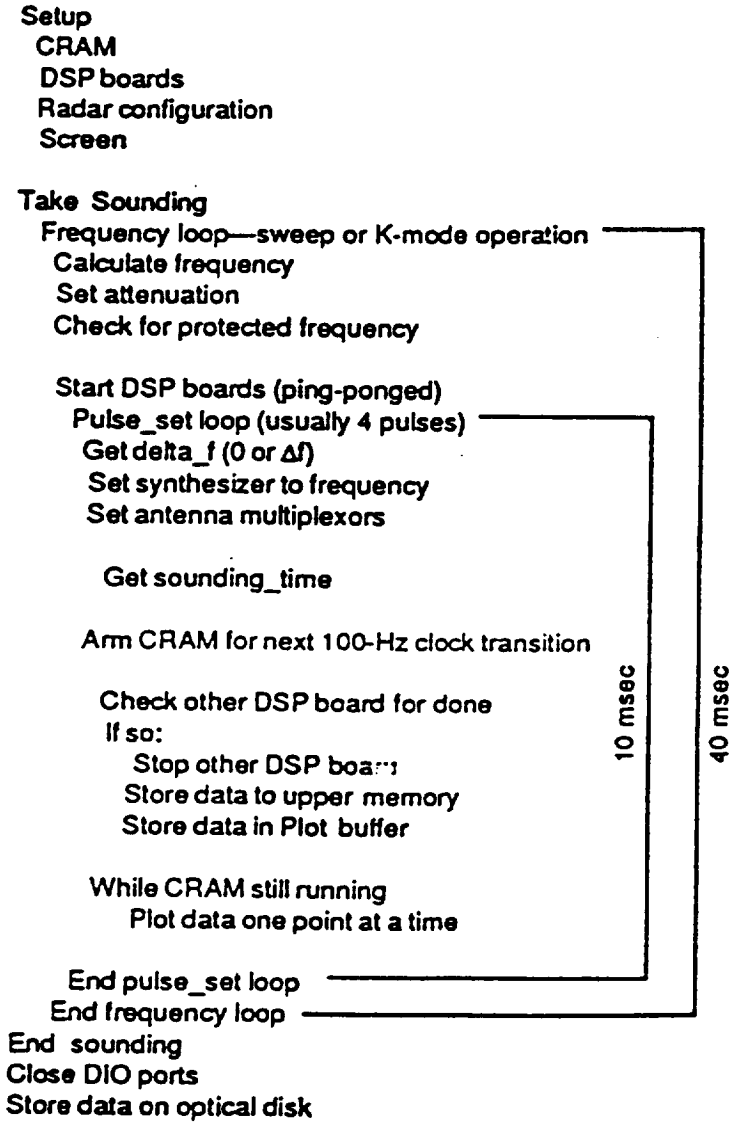


Figure 3.4 Sounding program functions

shows schematically how the sounding program functions.

3.5 PARAMETERS OF MEASUREMENTS

Parameters obtained directly from the radar system are

(1) The universal time, obtained from the translation of the computer local time.

(2) The A/D sampling rate, f_s= 100 kHz, or T_s = 10μs. The range resolution is determined by this parameter.

(3) The time of flight t; this parameter measures the time interval from the transmission of a pulse to the time of reception. It is usually expressed by $t=nT_s$, where n is the clock counter and T_s is the sample interval. The range r is obtained from this parameter, expressed by

$$r = \frac{1}{2}tc = \frac{1}{2}nT_sc \tag{3.1}$$

where c is the speed of light.

(4) Amplitude of the X-Y quadrature output of each receiver. The echo amplitude A is derived from

$$A = \sqrt{X^2+Y^2} \tag{3.2}$$

and the echo phase is derived from

$$\phi = \arctan\frac{Y}{X} \tag{3.3}$$

(5) The sounding frequency f; usually it varies from 1.6 MHz to 15 MHz with an increment of

approximately 20 KHz.

(6) The pulse frequency difference Δf in a pulse set. A pulse set is a sequence of four pulses separated by equal time intervals. Within the pulse set some pulses have a slightly high frequency ($f+\Delta f$) than the others (f). Δf is a constant during a pulse set. The selection of Δf will influence the group range resolution.

In addition to the direct parameters, there are some important parameters derived mainly from the echo phase. The USU Dynasonde uses an L-shaped antenna array of four elements with the South and West antennas forming a crossed dipole pair (Fig. 3.2). The pulse configuration is shown in Table 3.1.

The measured phase $\Phi(x,y,z,t,f)$ is a function of location, time, and frequency. The variation of the echo phase with the time and frequency can be obtained from a set of equations which result from the pulse configuration (Table 3.1):

$$(\phi_8-\phi_2) - (\phi_5-\phi_1) = 2\Delta t\frac{\partial\phi}{\partial t} \tag{3.4}$$

$$(\phi_5-\phi_3) - (\phi_4-\phi_2) = 2\Delta f\frac{\partial\phi}{\partial f} \tag{3.5}$$

In practice, a mean value is derived in order to reduce the background noise and the measurement error. By definition we can derive the following parameters:

(1) The doppler frequency f_d,

$$f_d \triangleq \frac{1}{2\pi}\frac{\partial \phi}{\partial t} \tag{3.6}$$

(2) The group range r_v,

$$r_v \triangleq \frac{1}{2}\frac{\partial}{\partial f}(fr) = \frac{c}{4\pi}\frac{\partial \phi}{\partial f} \tag{3.7}$$

where c is the speed of light. Comparing equations (3.1) and (3.7) we have ranges derived from different equations. Note that the maximum range resolution from equation (3.1) is 1.5km, but the group range resolution from equation (3.7) is 50m if the phase change over the interval Δf (=8kHz) can be measured with an accuracy of 1 degree. For a smaller Δf and the same echo phase, the accuracy of the range will decrease. On the other hand, for a large Δf and the same phase resolution the range ambiguity will increase due to the increased number of 2π phase rotations of Δ. In addition, a large Δf will cause inaccuracy in the linear approximation of the phase variation. 8 KHz is chosen for the Δf in the USU Dynasonde. For vertical transmission reflection the range is identical to the height.

CHAPTER IV

SOUNDING DATA AND IONOGRAMS

An obvious difference between the ionosphere (from 70 km to 800 km) and the other layers of the atmosphere is that the ionosphere consists of ions and reflects radio waves over a wide range of frequencies. This provides an opportunity to probe the ionosphere and to study geophysical events by using radar waves in the HF range. The ionogram still plays an important role in studying and visualizing the ionosphere, although a variety of new graphical presentation methods have been used in recent years.

A typical quiet daytime ionogram (called I-mode plot) is shown in Fig. 4.1 The vertical axis represents the virtual height (h') and the horizontal axis represents the transmitting frequencies. The E and F layer echoes, ordinary and extraordinary waves, and the second hop reflections are seen clearly. The shape of these curves on the ionogram vary with the geographic location, the season of the year, the time of day, and the solar activity.

Fig. 4.2 shows another form of data presentation which displays data similarly to the K-mode sounding. The horizontal and vertical axes represent the time and the echo range, respectively. In K-mode, the radar sends a pulse sequence at a fixed frequency and the received echoes form a time-series of data, representative of ionospheric variations as a function of time. The K-mode plot is particularly useful for the detection of GWs. Apart from the virtual height, there are other useful parameters directly or indirectly derivable from the radar echoes. They are

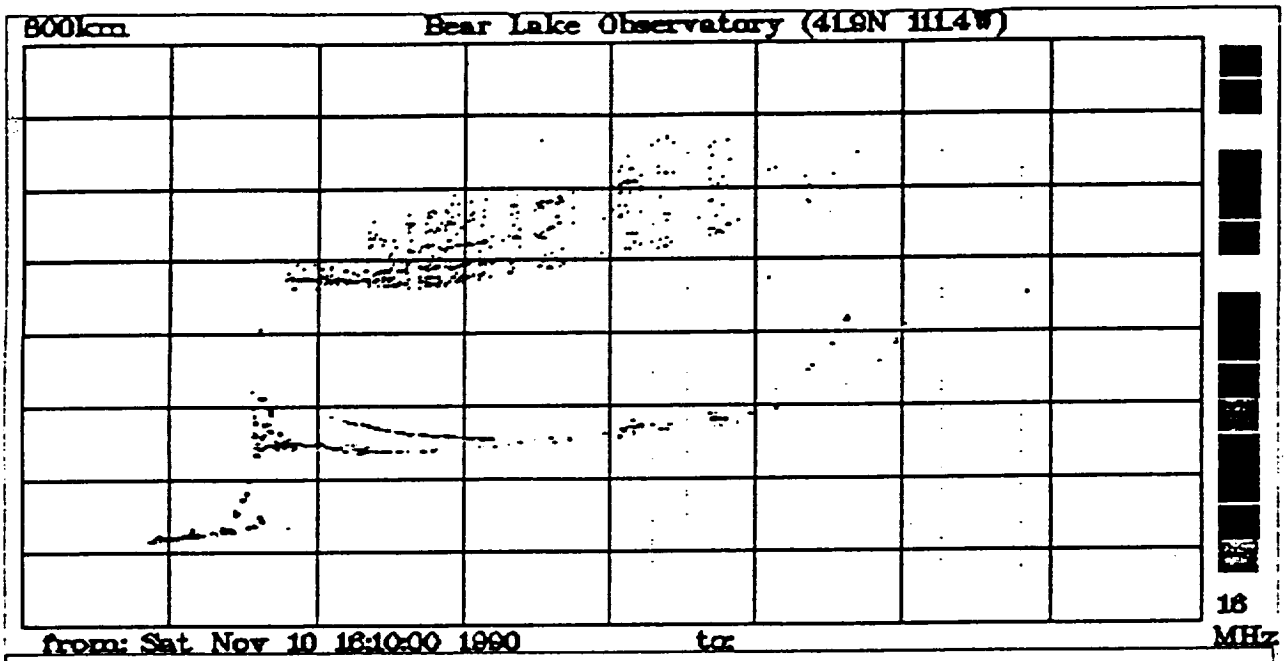


Fig4.1 I-mode ionogram

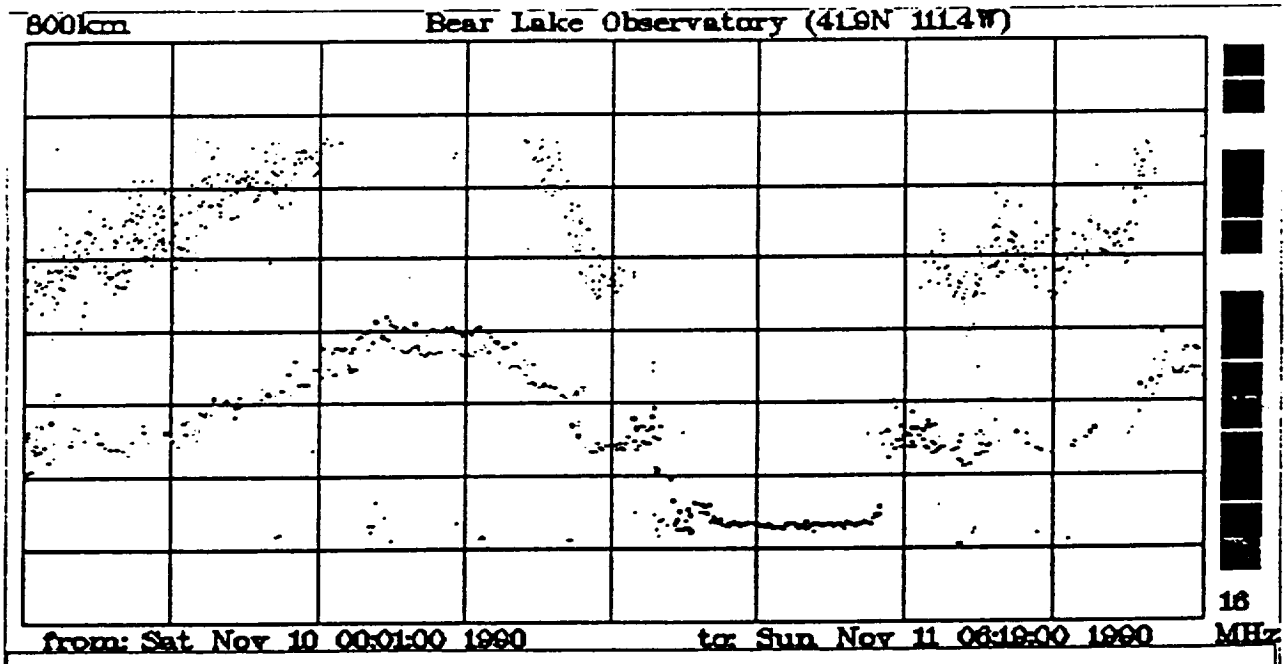


Fig4.2 The K-mode ionogram
f=3MHz.

amplitude, echo phase, angle-of-arrival, and the doppler velocity. These parameters, together with the virtual height, are the basic source of information for investigating GWs. In addition to the O-mode and X-mode echo traces, there can be interference, second hop reflections, and noise present in the ionogram. There are also some discontinuities in the frequency distribution of echoes, which result from missing data. It is desirable to extract as much information as possible from the noisy ionogram data before examining the behavior of the ionosphere.

4.1 SIGNAL-TO-NOISE RATIO

The Signal-to-Noise ratio (S/N) is usually large in the low frequency regime of the ionogram if the transmitter power is large. This results from the low altitude of the E and F layers. The S/N decreases as the sounding frequency increases and approaches the critical frequency. Frequently the S/N is less than 0 dB and echoes are not visible on the oscilloscope of the Dynasonde. In this circumstance, data loss occurs when the sampled data is processed by the peak search program in the SKY board. Because of the hardware configuration, it is not possible to increase the system bandwidth, however a feasible solution to the problem of low S/N is to use pulse coding and correlation detection techniques.

4.2 SECOND ORDER REFLECTIONS

The U.R.S.I. commission [1978] suggests that multiple echoes should always be examined and scaled when it is necessary to confirm or aid in the interpretation of the first order trace, but these values are not included in the basic U.R.S.I. summary tables or graphs. They are particularly valuable for showing whether the ionosphere is horizontally stratified, the assumption

implicitly made in most analyses of ionograms. In the detection of GWs we are primarily interested in the first order reflection. Because of the large height difference between the first and second order reflections one can easily separate them by using an appropriate range window. Since the range is a function of time, the window should be moved according to the variation of the range in time for an autoscaling system. At night the range of the first order reflection can be as large as the range of the second reflection during the day (Fig. 4.3). For a long duration K-mode analysis, the window position estimation is a difficult and time-consuming process. This problem can easily be solved by using the adaptive trace tracking algorithm which will be discussed in Chapter V.

4.3 INTERFERENCE

Careful study of the radar data collected using the USU Dynasonde shows that interference occurs mainly at night and most interference is due to strong transmissions by distant radio stations or by sferics associated with thunderstorms. Such interference appears in the ionogram as a line of 'echoes' randomly distributed over the whole 70-700 km range at particular frequencies. It has also been found that if both real echoes and interference exist in the data, the ratio of the maximum to minimum signal amplitude is greater than four. These two signatures of interference help to discriminate the echoes from interference. Further more, the discrete frequency of the interference shows an uncorrelated nature. Examining the data from the beginning frequency (I-mode) or from the first echo in the time series (K-mode) on the ionogram, we see that echoes are highly correlated and generally appear as continuous traces, whereas interference is uncorrelated and discontinuous. These features provide a possibility of using a

statistical method to remove interference.

4.4 O-MODE AND X-MODE DISCRIMINATION

If the ordinary and extraordinary echoes at a given frequency are well separated in virtual height, there should be a $+90^0$ or -90^0 phase difference between the North-South antenna pair and the East-West antenna pair, depending on the sense of rotation. One can easily separate the ordinary and extraordinary wave by comparing the echo phase at two orthogonal antennas. Then one can select one or the other component, depending on whether the phase difference is close to $+90^0$ or close to -90^0.

4.5 THE RELATION BETWEEN GWs AND VIRTUAL HEIGHT

The measured virtual height h' can be expressed by

$$h' = \frac{1}{2} c t_d \tag{4.1}$$

where c is the speed of light in a vacuum and t_d is the time delay defined by

$$t_d = 2\int_0^h \frac{1}{U} dz = 2\int_0^h \frac{1}{cn} dz \tag{4.2}$$

where h, the upper limit of the integration, represents the true height. U is the group velocity of the radio wave and n is the refractive index.

Substituting equation (4.2) into (4.1) the virtual height can be expressed as

$$h' = \int_0^h \frac{1}{n} dz \tag{4.3}$$

Equation (4.3) shows that virtual height is the integration of the reciprocal of the refractive index over the range from the ground to the true height.

The refractive index is a function of the electron concentration, radio frequency, electron collision frequency, geomagnetic field, and time. This relationship can be derived from Maxwell's equations and the equations of motion for charged particles [Liu, 1990]. For simplicity, the ionosphere is assumed to be homogeneous and isotropic. The refractive index can be expressed as

$$n^2 = 1 - \frac{f_N^2}{f_R^2} \tag{4.4}$$

where f_R is the transmitted radio frequency and f_N is the plasma frequency. The plasma frequency is a function of the electron concentration and can be expressed by

$$f_N = \frac{Ne^2}{4\pi^2 \varepsilon_0 m} \tag{4.5}$$

where N is electron concentration, e is the electron charge, ε_0 is the permittivity of free space, and m is the electron mass.

The contribution of the Lorentz force and Joule heating of a GW source to the electron concentration N will consequently influence the variation of the plasma frequency f_N, the refractive index n and virtual height h' at the time and location that the GW propagates over the observing site. Thus a measurement of the virtual height is representative of the movement of the GW disturbance.

By the same method, we are able to deduce the relationship between the GW and other radar parameters such as the echo phase, the amplitude, and the doppler frequency. In this report, we are restricting our study to the detection of GWs from measurements of the virtual height.

CHAPTER V

ADAPTIVE IONOGRAM SCALING

Accurately scaling and evaluating conventional ionograms is a tedious task which must be carried out in order to derive the electron density profile h(N). Because of the complexity of the ionosphere, especially the large variation in height and shape of the F layer, the search for an automatic scaling method is still incomplete. Reinisch and Huang [1983] proposed an Automatic Scaling algorithm using a simplified analytic function to approximate the E layer and an amplitude comparison to fit the F layer. Wright, et.al. [1972] proposed an autoscaling method by smoothing previously recorded ionograms and then using them to predict the next ionogram. But this method can neither eliminate the interference nor interpolate for missing data. Fox [1988] reported an automatic ionogram scaling software package in which a pseudo-trace is first formed by using all the expected echoes. Next, the program identifies the desired pixels as real echoes according to certain theoretical and practical criteria. The program also uses the amplitude information to distinguish between echoes and noise.

In this report, an alternate automatic ionogram scaling method is investigated using an adaptive filter. In the adaptive scaling process, signals (real echoes) and noise are input into a transversal filter consisting of a linear combiner. The coefficients of the linear combiner are adjustable and the output of the linear combiner forms an optimal estimate. An error signal, generated by subtracting the optimal estimate of the input signal from the input is fed back to adjust the linear combiner coefficients according to a specified optimal algorithm. The well

known LMS algorithm is used for the adaptive coefficient calculation in each step of the echo prediction. It can be shown that the LMS algorithm adaptive filtering technique yields a substantial reduction of noise and interference in both statistically stationary and nonstationary environments.

5.1 THE LINEAR COMBINER

An adaptive transversal filter consists of a tapped delay line connected to an adaptive linear combiner (Fig. 5.1). The input signal vector X_j of the adaptive linear combiner is defined by

$$X_j=[x_j\ x_{j-1}\ x_{j-2}\cdots x_{j-L}]^T \tag{5.1}$$

The subscript j is used as a time index. X_j is a column vector, and its elements are sequential samples taken at points j, j-1, ...going back in time through the sequence of data samples. The weighting coefficients W_j, or the weights, are adjustable and the output of the linear combiner y_j is equal to the inner product of X_j and W_j,

$$\begin{aligned} W_j &= [w_{0j}\ w_{1j}\ w_{2j}\ \cdots w_{(L-1)j}]^T \\ y_j &= X_j^T W_j = W_j^T X_j \end{aligned} \tag{5.2}$$

The output of the linear combiner is an estimate of a "desired response" d_j. In the adaptation process using performance feedback, the weight vector of the linear combiner is adjusted to cause

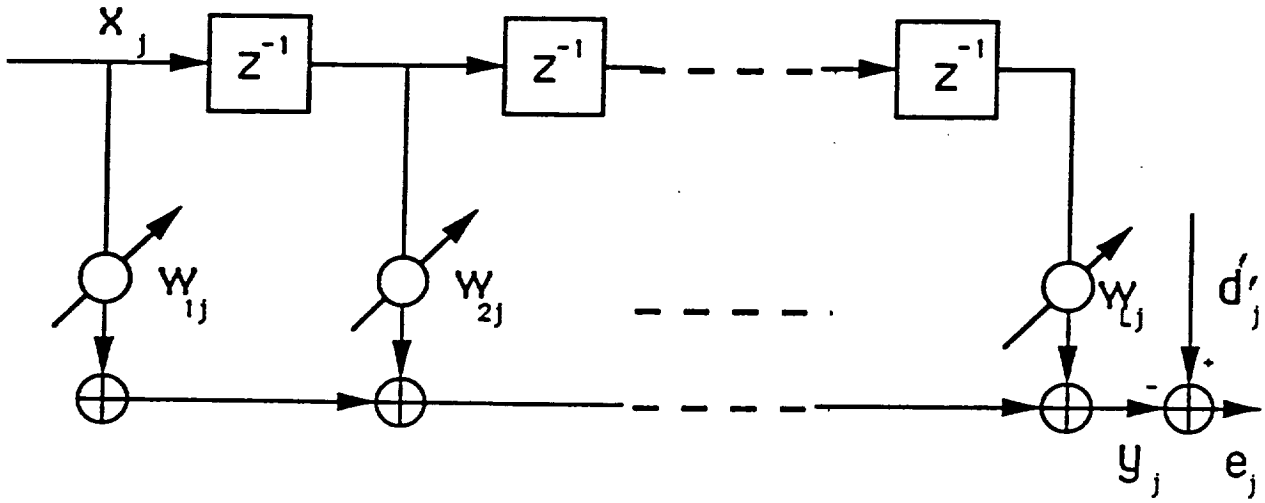


Fig. 5.1 Adaptive transversal filter

the output, y_j, to agree as closely as possible with the desired response signal. This is accomplished by comparing the output (y_j) with the desired response (d_j) to obtain an "error" signal (ε_j) and then adjusting or optimizing the weight vector to minimize this signal. The source of the desired response signal, d_j, depends on the utilization of the adaptive linear combiner.

From Fig. 5.1 the error signal with the time index can be expressed by

$$e_j = d_j - y_j \tag{5.3}$$

Substituting equation (5.2) into equation (5.3) yields

$$e_j = d_j - X_j^T W_j = d_j - W_j^T X_j \tag{5.4}$$

In most practical instances, the adaptive process is oriented toward minimizing the mean square error. We now square equation (5.4) to obtain the instantaneous squared error,

$$e_j^2 = d_j^2 + W_j^T X_j X_j^T W_j - 2 d_j X_j^T W_j \tag{5.5}$$

Assuming that ε_j, d_j, and X_j are statistically stationary, we take the expectation of equation (5.5)

$$E[e_j^2] = E[d_j^2] + W_j^T E[X_j X_j^T] W_j - 2E[d_j X_j^T] W_j \quad (5.6)$$

Let **R** be defined as the input correlation matrix

$$R = E[X_j X_j^T] \quad (5.7)$$

where **R** is a symmetric square matrix.

Let **P** be similarly defined as the column vector

$$P = E[d_j X_j] = E[d_j x_{0j}\ d_j x_{1j} \ldots d_j x_{Lj}]^T \quad (5.8)$$

This vector is the cross correlation vector between the desired response and the input components.

Let ξ be the mean-square error, rewriting equation (5.6)

$$\xi = E[e_j^2] = E[d_j^2] + W_j^T R W_j - 2P^T W_j \quad (5.9)$$

This equation shows that the mean-square error ξ is a quadratic function of the components of the weighting vector W_j when the input components and the desired response signal are stationary stochastic variables. A typical two-dimensional mean-square-error function is illustrated in Fig. 5.2. The bowl-shaped quadratic error function, usually called the performance surface,is

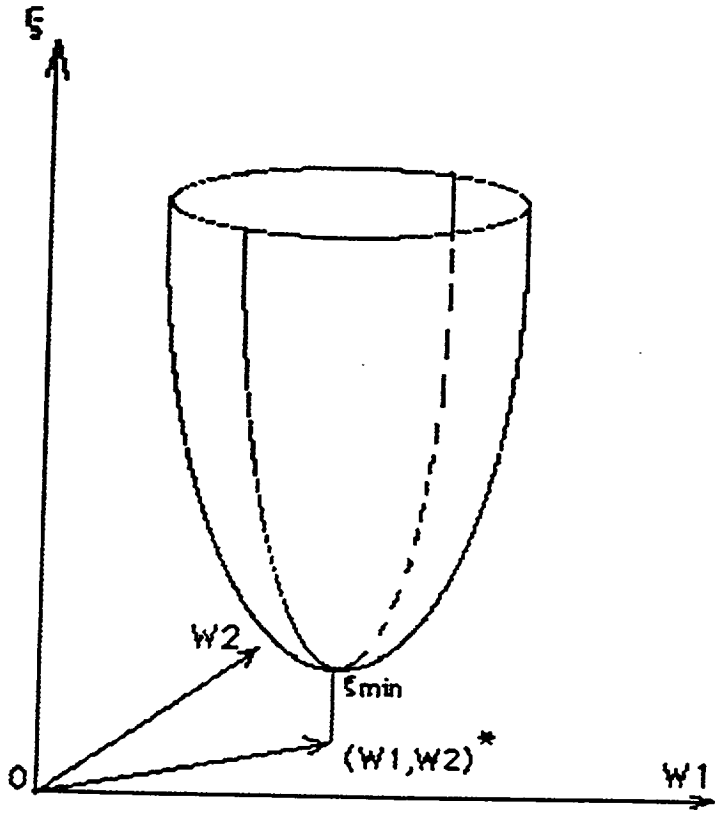


Fig.5.2 Two dimensional mean square function

a paraboloid. The value of ξ at the point of the "bottom of the bowl" represents the minimum mean-square error. The projection of this point on the weight-vector plane represents the optimal weight vector W^*.

To find the minimum mean-square error ξ_{min} and the optimal weighting vector W^*, we can use the gradient method. The gradient of the mean-square error is obtained by differentiating equation (5.9)

$$\nabla_j \triangleq \frac{\partial \xi}{\partial W_j} = 2RW_j - 2P \tag{5.10}$$

where R and P are given by equations (5.7) and (5.8), respectively. The optimal point on the performance surface should be the point where the gradient equals zero

$$\nabla = 0 = 2RW^* - 2P \tag{5.11}$$

It can be shown that R is nonsingular. Finally, we can find the optimal weight vector W^*

$$W^* = R^{-1}P \tag{5.12}$$

Equation (5.12) is the well known Wiener-Hopf equation [Widrow, 1985] in matrix form.

The minimum mean-square error is obtained by substituting $\mathbf{W}^*$ for $\mathbf{W}_j$ in equation (5.9) and noting that **R** is a symmetric matrix:

$$\xi_{min} = E[d_j^2] - P^T R^{-1} P = E[d_j^2] - P^T W^* \tag{5.13}$$

The Wiener filter theory has an important result: the error signal and the input signal are orthogonal, which also applies to the adaptive linear combiner. It can be shown by multiplying both sides of equation (5.4) by $\mathbf{X}_j$ that we have

$$e_j X_j = d_j X_j - X_j X_j^T W_j \tag{5.14}$$

Taking the expected value of (5.14) we obtain

$$E[e_j X_j] = P - R W_j \tag{5.15}$$

Letting $\mathbf{W}_j$ take its optimum value $\mathbf{W}^*$, we finally obtain

$$E[e_j X_j]_{W=W^*} = P - R W^* = 0 \tag{5.16}$$

5.2 THE METHOD OF STEEPEST DESCENT AND THE LMS ALGORITHM

In most practical cases, parameters of the performance surface are unknown, and an analytical description of it is not available. The solution to this problem is to develop proper algorithms, capable of searching the performance surface and finding the optimal weight vector by using the measured or estimated data. The steepest descent method has been shown to be the most widely applicable among the various gradient search methods. By definition, the steepest descent algorithm can be expressed

$$W_{j+1} = W_j + \mu(-\nabla_j) \tag{5.17}$$

where μ is a constant which regulates the step size and $-\nabla_j$ is the negative gradient at jth step. Substituting equations (5.10) and (5.12) into equation (5.17) yields

$$W_{j+1} = (I-2\mu R)W_j + 2\mu R W^* \tag{5.18}$$

By making (I-2μR) orthogonal, we have

$$(I-2\mu R) = Q(I-2\mu\Lambda)Q^T \tag{5.19}$$

where Q is a unit matrix and Λ is the eigenvalue matrix of R.

It can be shown [Widrow, 1985] that the steepest descent algorithm is stable and convergent if, and only if,

$$\lim_{j\to\infty}(I-2\mu\Lambda)^j = 0 \qquad (5.20)$$

Equation (5.20) may also be expressed

$$\lim_{j\to\infty}(1-2\mu\lambda_l)^j = 0 \quad l=0,1 \ldots L \qquad (5.21)$$

From equation (5.21) we can show that the convergence condition is satisfied by choosing μ so that

$$0 < \mu < \frac{1}{2\lambda_{max}} \qquad (5.22)$$

where λ_{max} is the largest eigenvalue of R.

Equations (5.17) or (5.18) are not useful in practical applications, because it is difficult to calculate the gradient for each search step. The LMS algorithm uses a special estimate of the gradient that is valid for the linear combiner. Recalling equation (5.9)

$$\xi = E[e_j^2], \qquad (5.23)$$

let ε_j^2 itself be the estimation of mean-square error ξ, then at each iteration in the adaptive

process, we have a gradient estimate of the form

$$\hat{\nabla}_j = \frac{\partial e_j^2}{\partial W_j} = 2e_j \frac{\partial e_j}{\partial W_j} \qquad (5.24)$$
$$= 2e_j X_j$$

The derivatives of ε_j with respect to the weights follow directly from equation (5.4). Substituting equation (5.24) into equation (5.17) we have

$$W_{j+1} = W_j - \mu\hat{\nabla}_j = W_j + 2\mu e_j X_j \qquad (5.25)$$

This is the LMS algorithm. From its form in equation (5.25), we see that the LMS algorithm can be implemented in a practical system without averaging, squaring, or differentiation and is elegant in its simplicity and efficiency.

It can be easily proved that ∇_j is an unbiased estimate of ∇_j by taking the expected value of equation (5.24) and noting equation (5.10),

$$E[\hat{\nabla}_j] = -2E[e_j X_j] = -2E[d_j X_j - X_j X_j^T W_j]$$
$$= 2(RW_j - P) = \frac{\partial \xi}{\partial W_j} = \nabla_j \qquad (5.26)$$

where we assume that the weight vector W_j is independent of the input X_j. Since the expected

value of the estimated gradient is the true gradient, the estimate of the gradient is unbiased. For stationary input processes, the expected value of the weight vector $E[W_j]$ of the LMS algorithm converges to the Wiener optimal solution, that is $W^*=R^{-1}P$. Taking the expected value of both sides of equation (5.25) we have

$$\begin{aligned} E[W_j+1] &= E[W_j]+2\mu E[e_jX_j] \\ &= E[W_j]+2\mu(E[d_jX_j]-E[X_jX_j^TW_j]) \\ &= E[W_j]+2\mu(P-RE[W_j]) \end{aligned} \tag{5.27}$$

After a sufficient number of iterations, we have $W_{j+1}=W_j \rightarrow W^*$ and then equation (5.27) becomes $W^*=R^{-1}P$.

Two important parameters for an iterative algorithm are the condition of convergence and the speed of convergence or the time constant. It has been shown that the LMS algorithm has the same convergence restriction as the steepest descent algorithm [Widrow,1985]

$$0 < \mu < \frac{1}{2\lambda_{max}} \tag{5.28}$$

where λ_{max} is the largest eigenvalue of R. Note that

$$\begin{aligned} \lambda_{max} &\le tr[\Lambda] \\ tr[\Lambda] &= tr[R] \\ &= (L+1)E[X_j^2] \\ &=(L+1)(\textit{signal power}) \end{aligned} \tag{5.29}$$

Equation (5.29) can be written as

$$0 < \mu < \frac{1}{2(L+1)(\textit{signal power})} \tag{5.30}$$

where L is the dimension of R.

This is a more restrictive bound on μ than equation (5.22), but it is much easier to apply because the elements of R and the signal power can generally be estimated more easily than the eigenvalue of R.

The time constant, in terms of iterations for the LMS algorithm, is given by

$$\tau_l \approx \frac{1}{2\mu\lambda_l} \quad l=0,\ 1,\ 2,\ \ldots,\ L \tag{5.31}$$

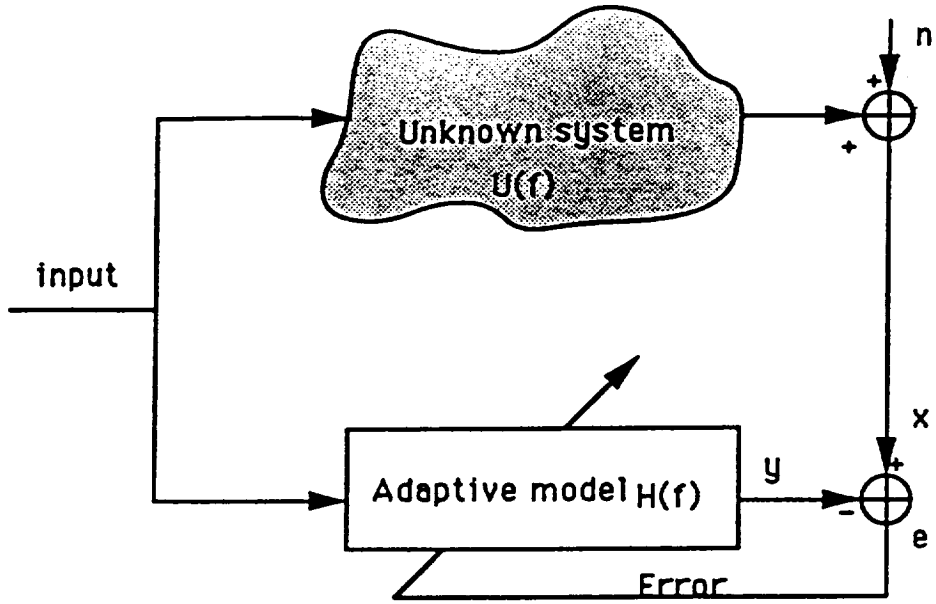
Equation (5.31) shows that each weight element converges to its optimal value at a different rate and finally, that the system convergence rate depends on the smallest eigenvalue of R.

5.3 CONSIDERATION OF THE IONOSPHERE AS A DYNAMIC SYSTEM

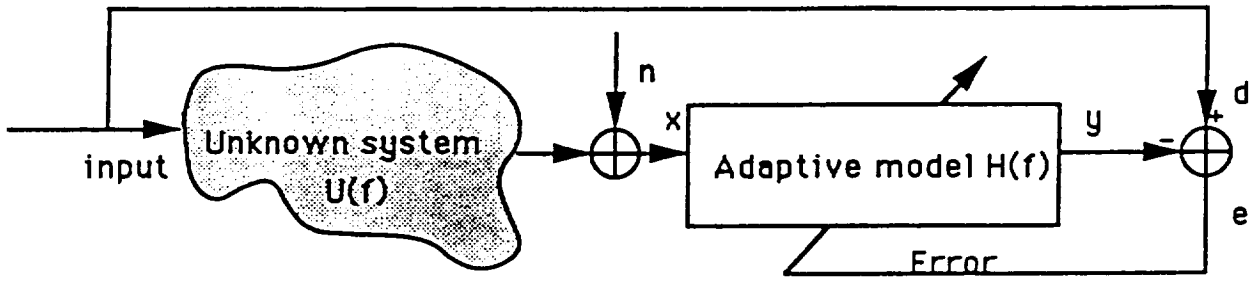
Scientists are constantly seeking a precise analytic form in which to describe the dynamics of the ionosphere. However, because of its complexity it is impossible to use a single closed form to represent dynamics that are strongly influenced by their surrounding environment. Scientists have to make a series of assumptions and simplifications before reaching a reasonable conclusion. From the point of view of system theory, we may look at the ionosphere as a physical dynamic system or an unknown "black box" and the received radio echo trace a random signal. Then we can apply system modeling and identification theory to investigate the behavior of the ionosphere in a particular time period or dimension. We also can apply random signal processing theory to examine the radar echo. A fundamental method of modeling and identification is to adjust a modeling system to simulate the unknown system by minimizing the output of the error of the two systems. Next, the modeling system is used to represent the unknown system to be analyzed.

An adaptive system identification scheme is illustrated in Fig. 5.3a. A signal is input into the unknown system and the adaptive filter simultaneously. Then the output of the adaptive filter and the noisy output of the unknown system are compared and an error signal is generated. This error signal is used to adjust the weights of the adaptive filter, so that the outputs of the adaptive filter and the unknown system have a best match in the sense of a least mean square error.

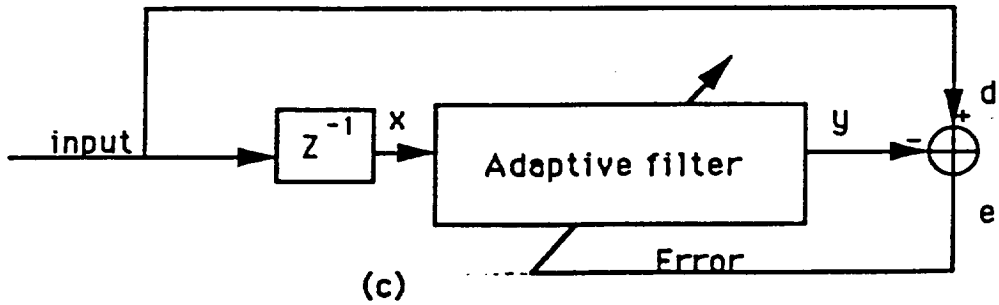
An alternate adaptive modeling configuration is illustrated in Fig. 5.3b. The unknown system and the modeling system are linked in a cascade form and the final transfer function of the modeling system is the inverse of the unknown system. This kind of structure has found



(a)



(b)



(c)

Fig. 5.3 (a) Direct system modelling
(b) Inverse system modelling
(c) Linear predictor

application to communication channel equalization [Cowan, 1987]. Considering the present problem, the ionosphere is the "black box"; the input to the system is the radar signal; and the noisy output of the "black box" is the received signal. By choosing proper input signals and slightly modifying the system, we are able to obtain the unknown system parameters and more importantly, the dynamic behavior of the system. Fig. 5.3c is a linear prediction structure, where the input to the adaptive filter is a delayed version of the desired output from the filter itself. Thus, the filter output forms the prediction of its future inputs. If the input of the system is the radar echo sequence, then the output of the adaptive filter is an optimal prediction of the echo trace itself.

5.4 ADAPTIVE IONOGRAM SCALING

In order to detect ionospheric gravity waves we need to extract the real echo trace from the Dynasonde data and to obtain a reliable profile in the h-N or h'-f planes. As described in Chapter IV, obtaining real echo traces from noisy radar data is a tedious and difficult task. In this Chapter, we investigate an adaptive autoscaling method to obtain the echo trace using the method described in section 5.3. To apply the adaptive ionogram autoscaling method, we make the following assumptions:

(1) The ionosphere is stable over a time period comparable to the frequency sweep. This ensures that the received signal or the echo sequence are highly correlated. In practical situations, this is generally true. Here "stable" means no fast random variations that cause decorrelation of the received signal. The TID, irregularity, drift motion, and GWs are correlated disturbances.

(2) The echo sequence forms a continuous trace as a function of sounding frequency. In other

words, there should be no large gaps along the echo trace. Usually, however there are some discontinuities because of interference, protected frequencies, and irregularities. This problem can be solved by using a range window, if the gap widths are tolerable (we will see that the E and F layer gap can be easily detected by the adaptive system).

With the above assumptions we build our adaptive scaling system as illustrated in Fig. 5.4. A set of radar "echoes" received at time j pass through a moving range window whose position is controlled by the output of an adaptive predictor. After windowing, there are one or more echoes left, which are regarded to be the real echoes. They can be expressed as

$$x_j(n) \quad n = 1, 2, 3, ..., N \tag{5.32}$$

where j denotes time and n the echo number. These echoes are averaged and the average value is represented by

$$\bar{x}_j = \frac{1}{N}\sum_{n=1}^{N} x_j(n) \tag{5.33}$$

In this case, the input vector of the transversal filter has the following form

$$X_j = [\bar{x}_j \; \bar{x}_{j-1} \; \bar{x}_{j-2} \; \cdots \; \bar{x}_{j-L}]^T \tag{5.34}$$

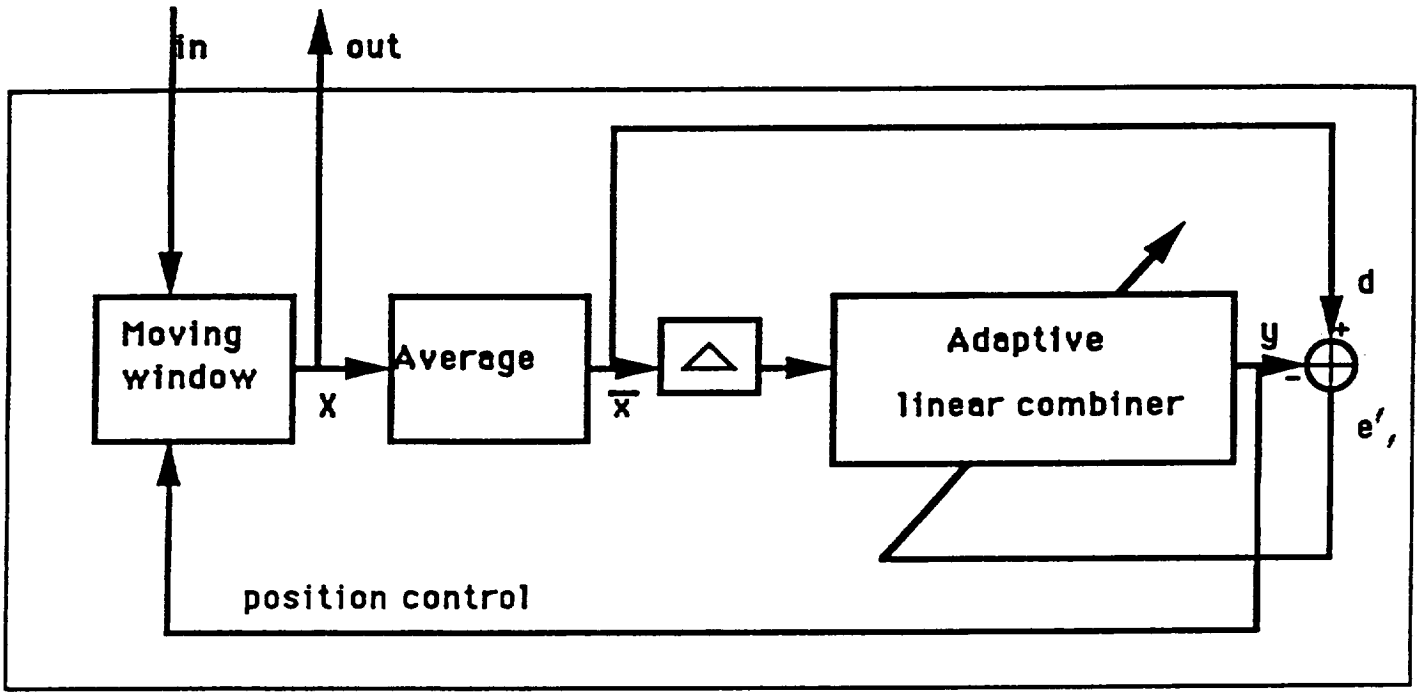


Fig. 5.4 The adaptive scaling system

Before the linear combiner, a one-step delay operation is carried out so that the output of the linear combiner(y_j) forms an optimal estimation of the average value of the next windowed data set x_{j+1}.

The system now becomes an adaptive predictor. In this case, the desired response d_j is the signal itself with a one step delay

$$d_j = \bar{x}_{j+1} \tag{5.35}$$

and we then have

$$P = E[d_j X_j] = E[\bar{x}_{j+1} X_j] \tag{5.36}$$

$$R = E[X_j X_j^T] \tag{5.37}$$

$$W^* = R^{-1}P \tag{5.38}$$

In practical situations, the input value x_j is a sum of the echo range and the noise. We have shown in Chapter IV that the noise and interference are uncorrelated and they are also uncorrelated with the echo range. Thus, the correlation vector P and matrix R consist of a contribution due only to the echo range. The weighting vector will converge to its optimal value so as to form an optimal estimate of the echo range.

The use of a moving window will not only remove the second order reflection but also the noise and interference outside the window. Under control of the optimal range estimate y_j, the moving window has the ability to track the echo trace at any time during the day or night.

Often, due to the presence of oblique or spread echoes there can be more than one echo received for each individual pulse transmission. Reflection from oblique directions and perturbations due to tilts or travelling disturbances can cause multiple traces on the ionogram. Since echoes and traces are likely to appear more close together, the average range is chosen as the adaptive predictor input and all traces and echoes within the windows will be accepted as signals to be displayed on the ionogram.

Some simulation and experimental results of the adaptive ionogram scaling are presented by using analytic input signals and ionospheric sounding data. Figure 5.5 illustrates a simulation using the adaptive scaling method with a sine wave input (x, the blue curve). The red curve is an estimate of the sine wave (y), while the green curve represents the error signal ε. It takes about 20 steps to converge when the parameter L equals 8 and μ equals 0.5E-6.

Figure 5.6 shows another simulation using the adaptive scaling method with an exponential wave input (x, the blue curve). The estimated value (y, the red curve) is nearly coincident with the input (x) for this ideal noiseless and no-data-loss situation. The system converges after 16 points when the parameters L is 8 and μ is 0.1E-6.

Figure 5.7 shows an I-mode ionogram processed by the adaptive scaling system. The red and yellow waves are the O-mode and X-mode traces respectively. The O-mode and X-mode traces are completely separated from the second hop and noise by the adaptive scaling system.

Figure 5.8 shows the same ionogram as in Figure 5.7 except for an additional curve

shown in blue, which is the estimate of the average range of the O-mode and X-mode traces. Steps of the adaptive tracking process are clearly seen. The parameters used adaptive scaling system are listed in the Figures.

Figures 5.9 and 5.10 show another example of an I-mode plot processed by the adaptive scaling system. Although there are several frequencies at which interference occurs and large amount of missing data at frequencies between 10 MHz and 12 MHz, the adaptive scaling system still works very well.

Figures 5.11 and 5.12 illustrates the application of the adaptive scaling method to single frequency plot derived from 600 ionograms. These data were obtained between November 10 0001:00 UT, 1990 and November 11 0619:00 UT, 1990. Curves with red, yellow and blue colors are the O-mode trace, X-mode trace and the estimate (y) of the average echo traces respectively. The chosen frequency is 4 MHz and the other parameters are shown on the figures. There is strong interference and discontinuarities in the O-mode and X-mode traces during the night but the adaptive scaling system readily follows the real echo trace.

Figures 5.13 and 5.14 show another example of a single frequency plot processed by the adaptive scaling system. The data is chosen from a different time interval and the K-mode frequency is lower than it is in Figures 5.11 and 5.12 so that less noise is seen on the plot.

5.5 PARAMETER SELECTION

Three parameters in the adaptive scaling system need to be carefully selected. These are the convergence constant μ, the weight vector length L, and the size of the moving window. Recalling equations (5.28), (5.29) and (5.31),

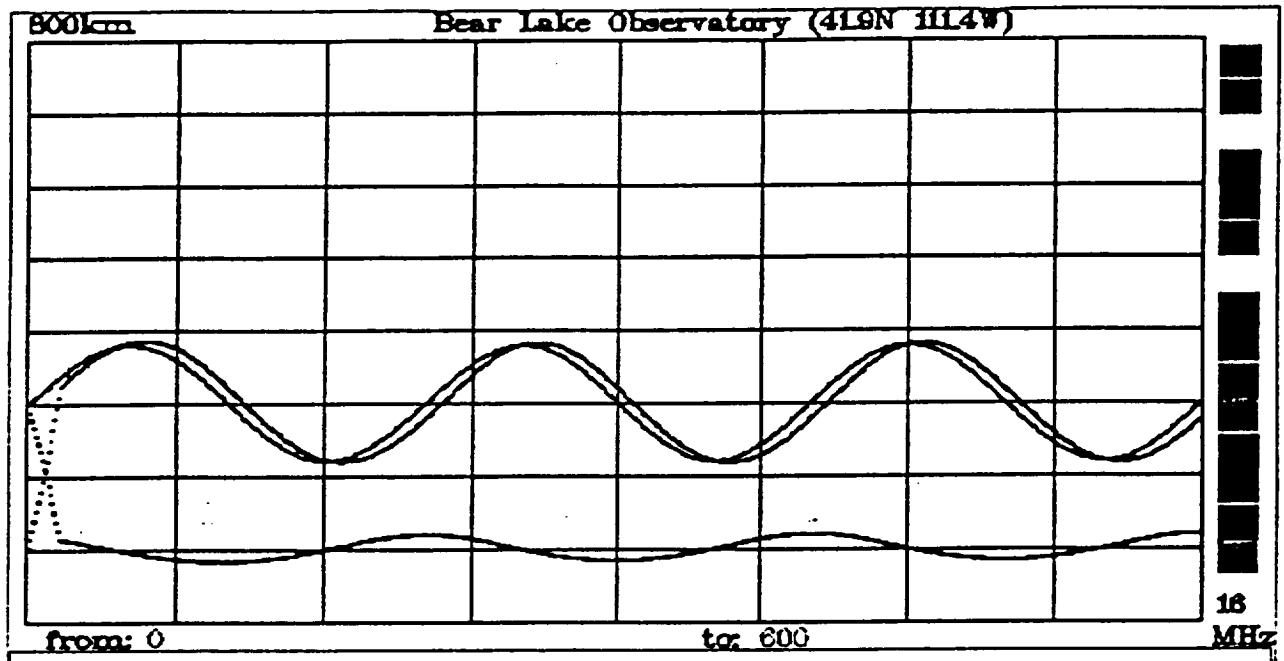


Fig. 5.5 Adaptive predictor simulation
Blue: x, sine function; Red: y, estimate of x; Green: e, the error.
L=8, mu=0.10E-6, data length=600

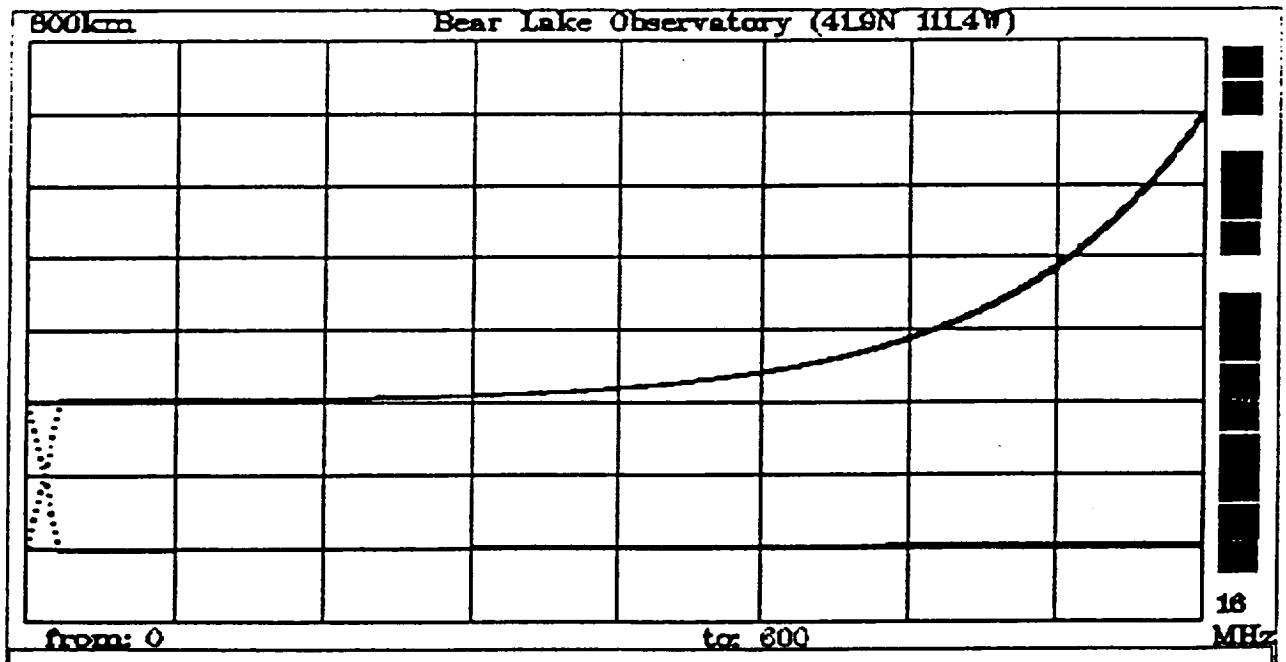


Fig. 5.6 Adaptive predictor simulation
Blue: x, exponential function; Red: y, estimate of x; Green: e, the error.
L=8, mu=0.10E-6, data length=600

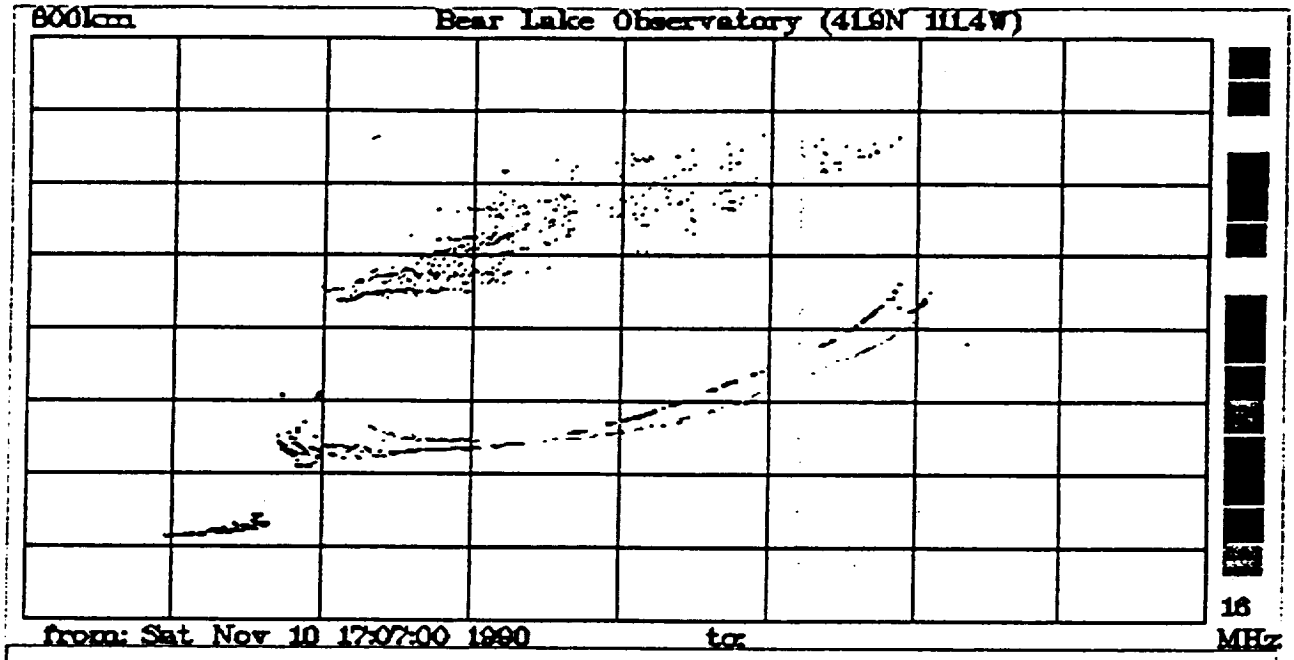


Fig.5.7 I-mode ionogram using adaptive scaling method
moving window size=122
LMS algorithm, L=8, mu=0.1E-7

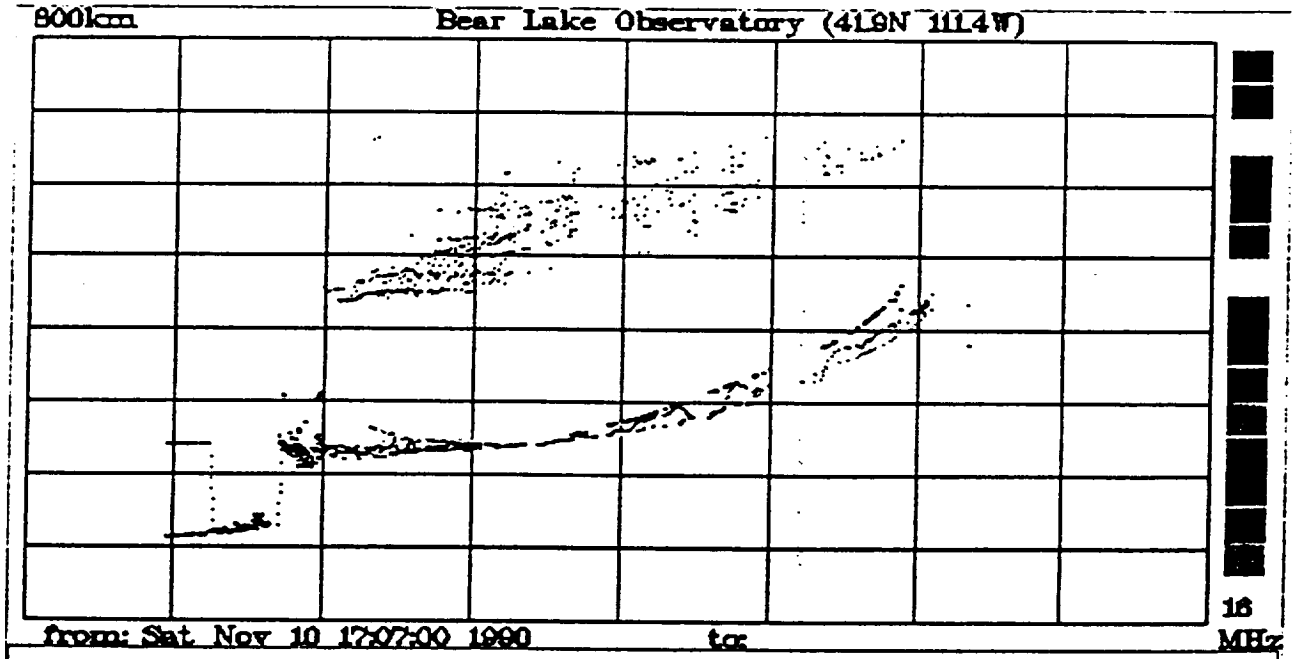


Fig.5.8 I-mode ionogram using adaptive scaling method
moving window size=122
LMS algorithm, L=8, mu=0.1E-7

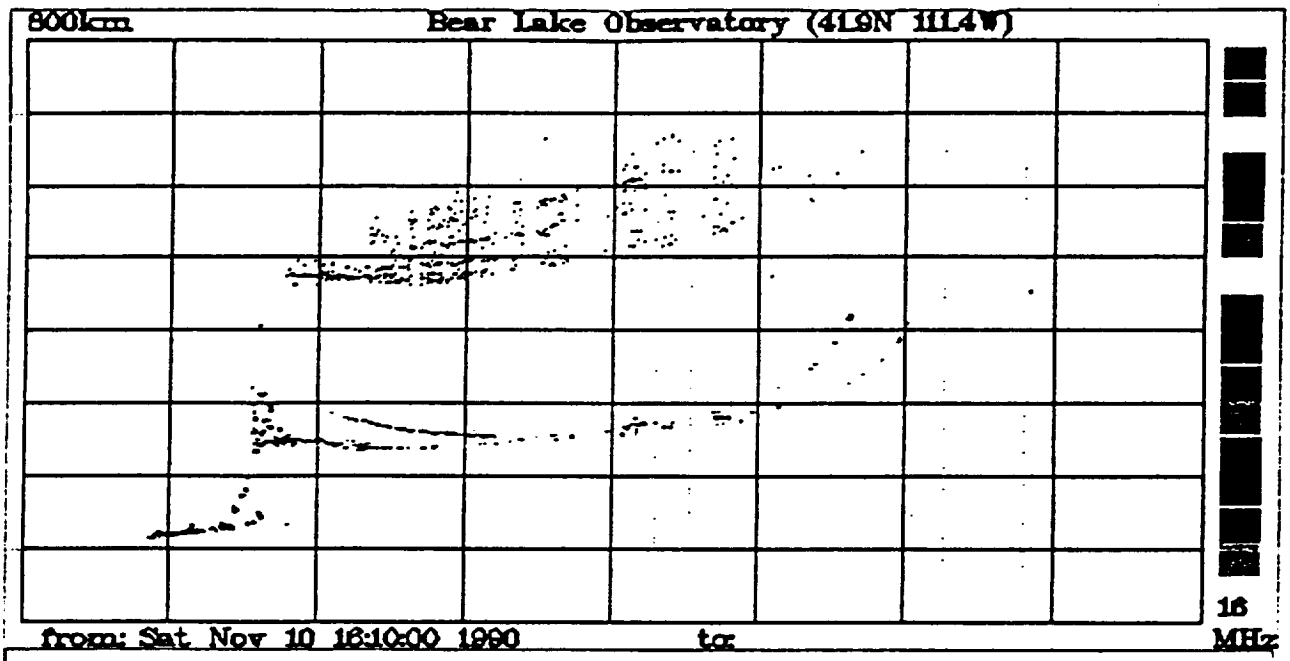


Fig.5.9 I-mode ionogram using adaptive scaling method
moving window size=122
LMS algorithm, L=8, mu=0.1E-7

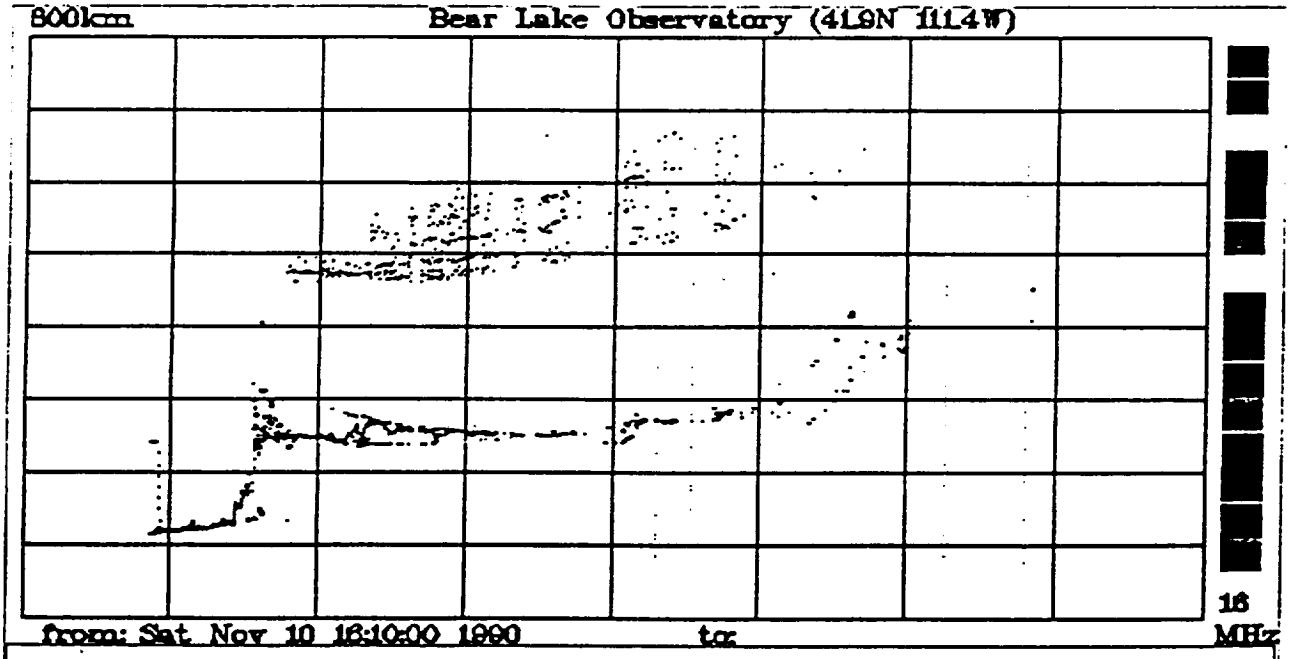


Fig.5.10 I-mode ionogram using adaptive scaling method
moving window size=122
LMS algorithm, L=8, mu=0.1E-7

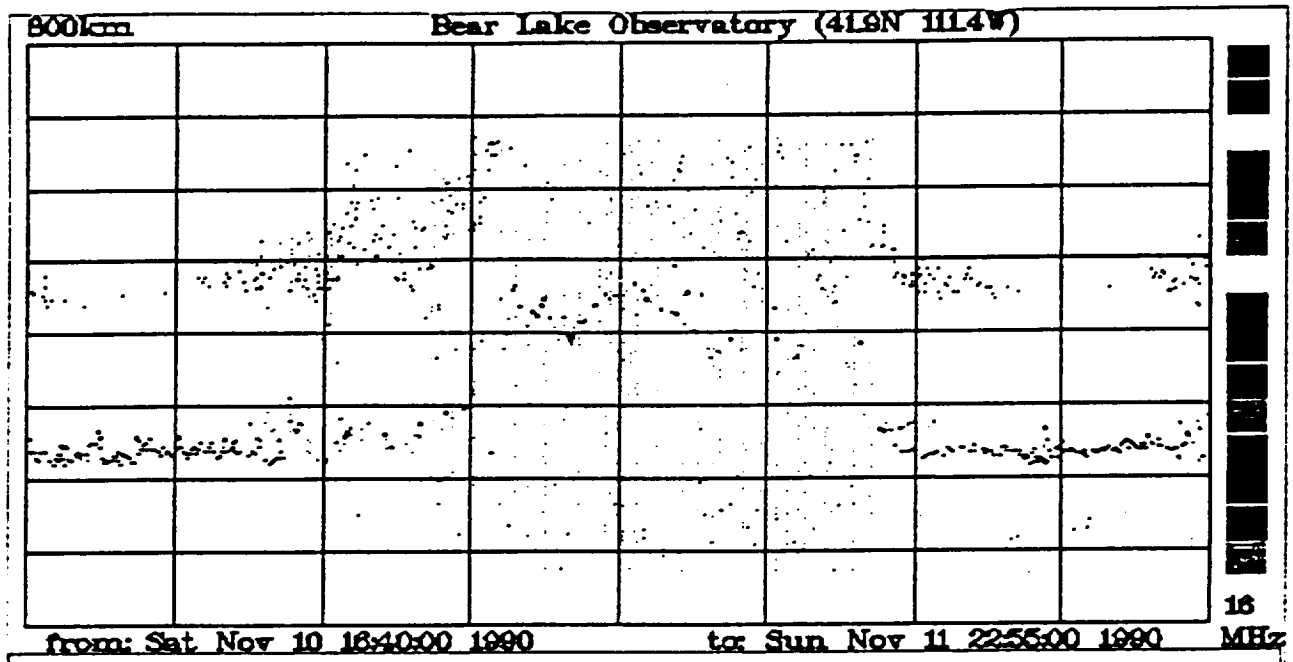


Fig.5.11 The K-mode ionogram using adaptive scaling
f=4MHz, L=8, mu=0.1E-6,moving window size=90.

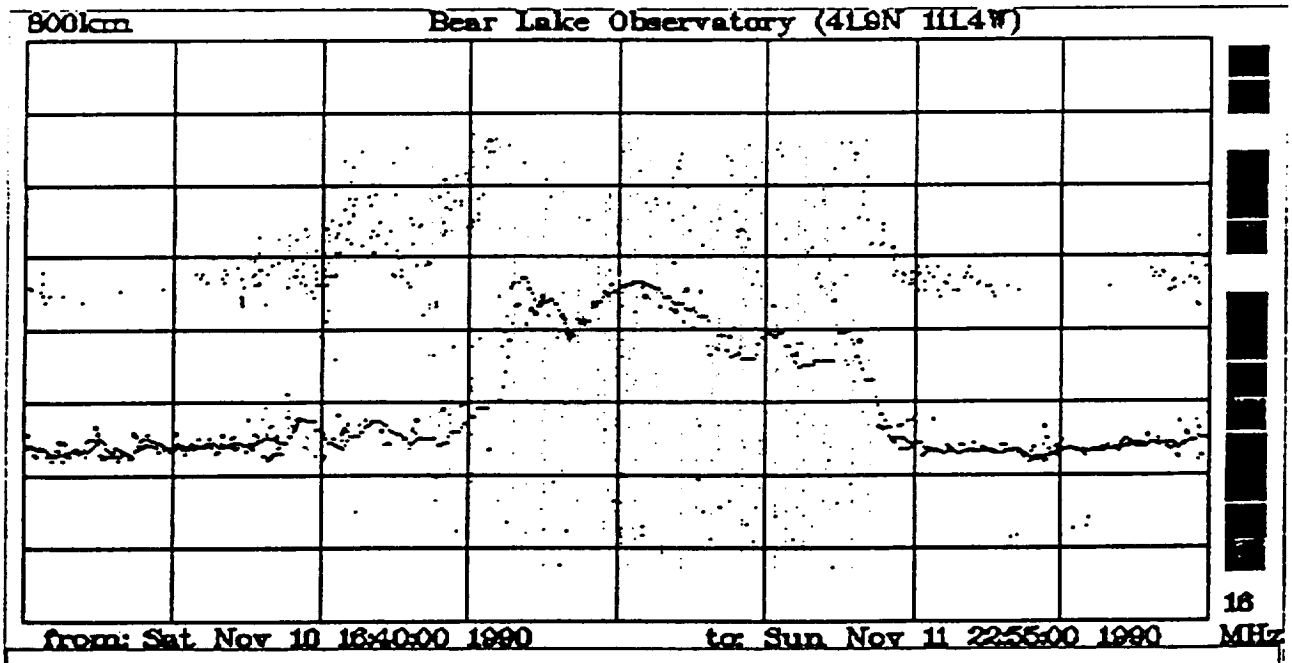


Fig.5.12 The K-mode ionogram using adaptive scaling
f=4MHz, L=8, mu=0.1E-6,moving window size=90.

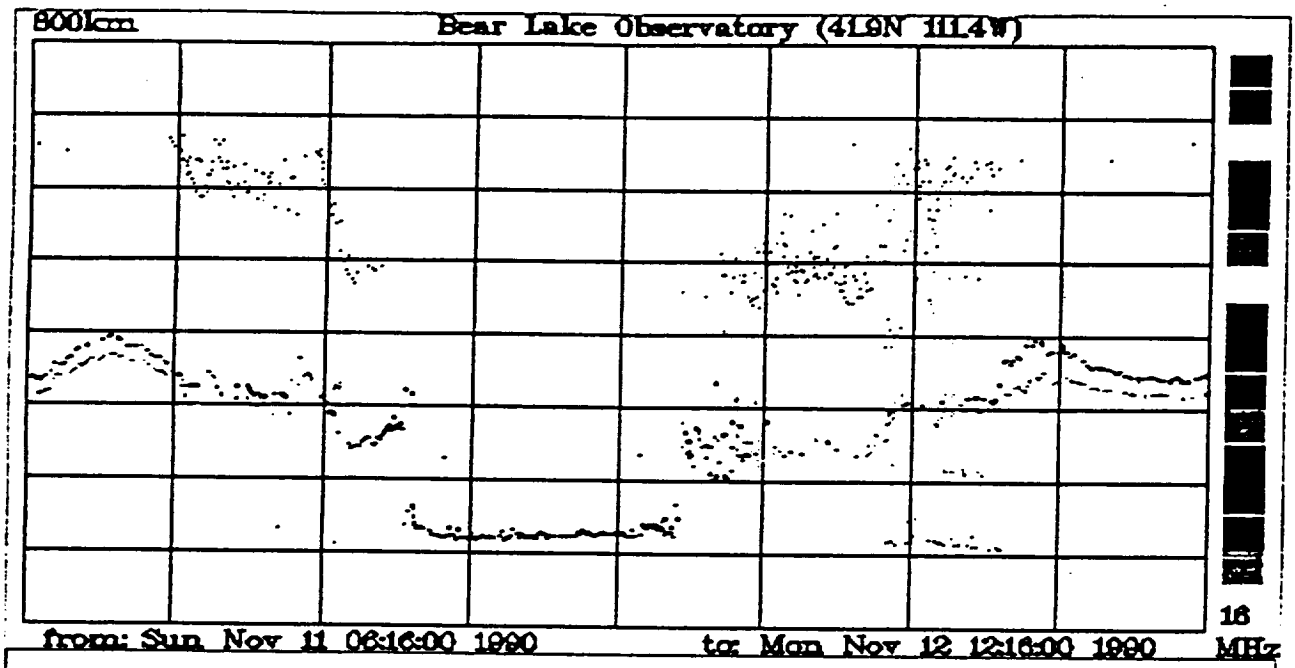


Fig.5.13 The K-mode ionogram using adaptive scaling f=2.8MHz, L=8, mu=0.1E-6,moving window size=122.

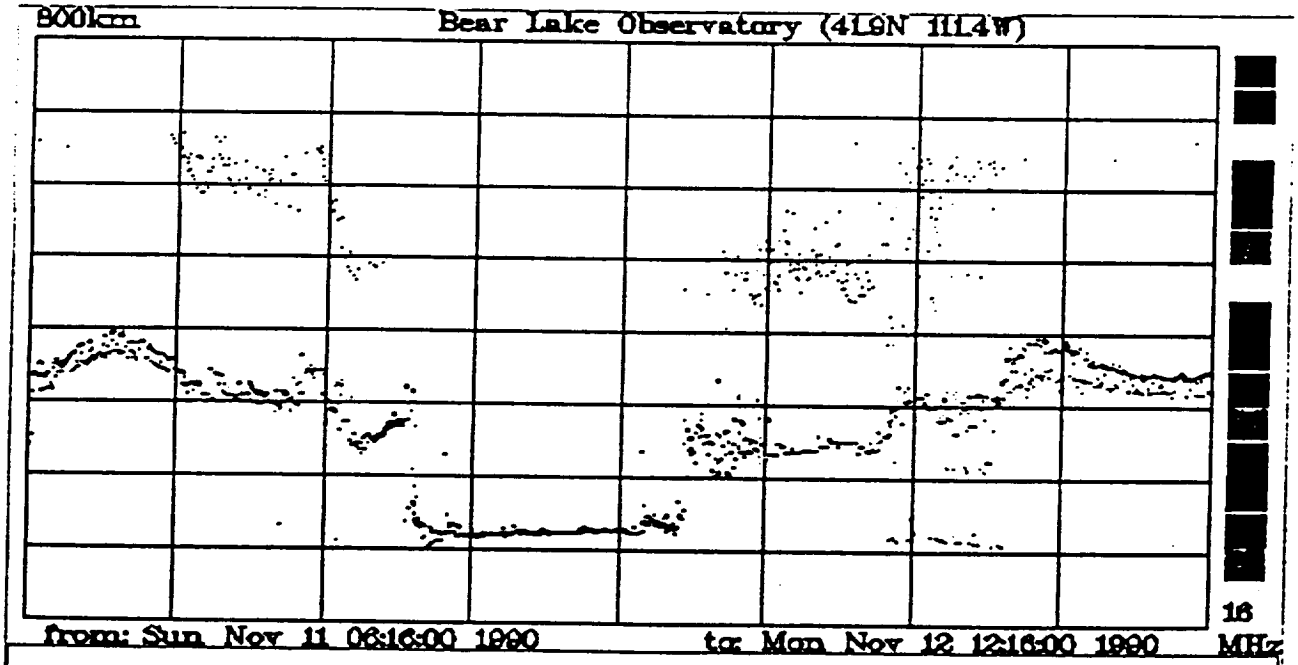


Fig.5.14 The K-mode ionogram using adaptive scaling f=2.8MHz, L=8, mu=0.1E-6,moving window size=122.

$$0 < \mu < \frac{1}{2\lambda_{max}} \tag{5.39}$$

$$\lambda_{max} \leq (L+1)(signalpower) \tag{5.40}$$

$$\tau_l = \frac{1}{2\mu\lambda_l} \quad l = 0, 1, 2, ..., L \tag{5.41}$$

we see that μ, τ, and L are related to each other. Large values of L will cause slow convergence and a large μ increases the convergence speed, but could cause violation of the convergence condition. In real applications μ and L are selected experimentally, where μ ranges from 0.00001 to 0.000001, and L lies between 8 and 20.

The selection of the size of the moving window is flexible because of the adaptive trace tracking ability. A window too small (less than 20km) could cause loss of tracking of an I-mode trace when the radio frequency is close to the critical frequency and the slopes of the O-mode and X-mode traces are large. On the other hand, a window too large (greater than 150 km) will include more noise and interference inside the window, and in the worst situation, could lead to an incorrect tracking of the second order reflection. In practice, the window size is chosen within the range from 20km to 100km.

5.6 THE MODIFIED LMS ALGORITHM

In the I-mode plot, the slope of the echo trace curve becomes larger and larger as the sounding frequency approaches the critical frequency of the F-region. This kind of fast variation in range at frequencies near the critical frequency affects the tracking ability of the adaptive predictor. A modified LMS algorithm is used in the I-mode plot to improve the adaptive system performance. The modified LMS algorithm has the following form

$$W_{j+1} = W_j + 2\mu g(f) e_j X_j \tag{5.42}$$

where g(f) is a positive function of the sounding frequency. Together with μ, it acts as a variable convergence factor which increases its value as the sounding frequency increases. A simplified mode of the modified LMS algorithm is the use of g(f) as a linear function of the sounding frequency f:

$$g(f) = kf \tag{5.43}$$

where k is a constant used to adjust the slope of the linear function. Thus, the modified LMS algorithm is given by

$$W_{j+1} = W_j + 2\mu k f e_j X_j \tag{5.44}$$

It is obvious that the modified LMS algorithm has the same convergence constraint as the LMS if μ is replaced by $\mu k f_{max}$ in equations (5.30) and (5.31). However, the modified LMS algorithm

greatly improved the tracking ability of the adaptive scaling system. A comparison of the LMS algorithm and the modified LMS algorithm is given in Figures 5.15 and 5.16. Figure 5.15 gives the adaptive scaled ionogram using the LMS algorithm. When the sweeping frequency increases up to 11 MHz the adaptive system output (y) no longer follows the average value of the echo traces. It follows only the X-mode trace which is varying slowly. There is one echo undetected at the far top of the O-mode trace (a point with black color). Figure 5.16 shows the advanced tracking performance of the modified LMS algorithm as a comparison with the LMS algorithm. Although there is data missing and the slope of the echo trace is large, the adaptive scaling output (y) still follows the average value of the echo traces. The echo undetected in Figure 5.15 is extracted and assigned a red color in Figure 5.16.

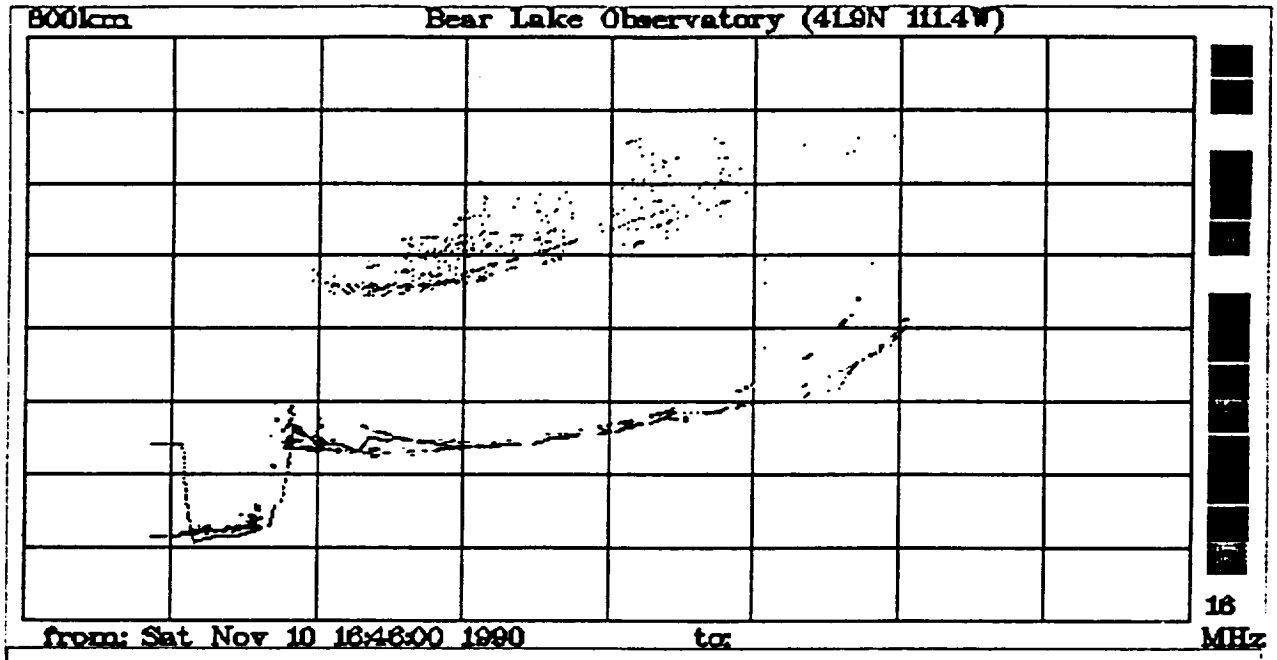


Fig.5.15 I-mode ionogram using adaptive scaling method
moving window size=122
LMS algorithm, L=32, mu=0.1E-7

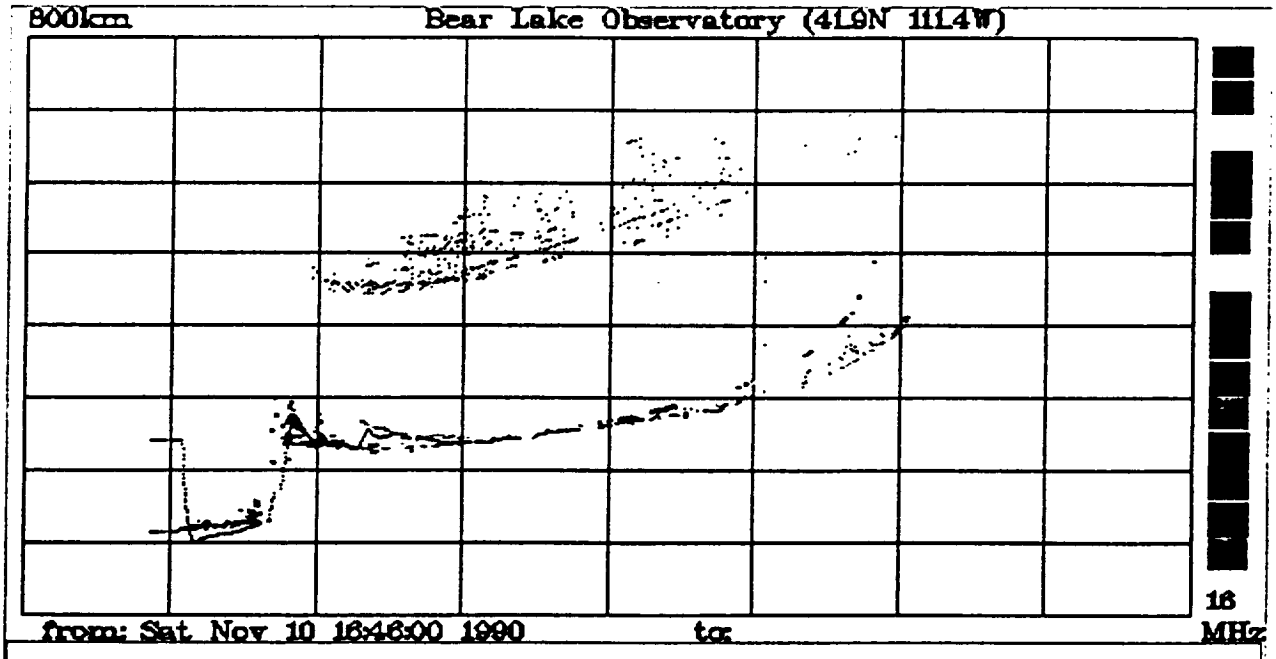


Fig.5.16 I-mode ionogram using adaptive scaling method
moving window size=122
MLMS algorithm, L=32, mu=0.1E-7

CHAPTER VI

THE MEASUREMENT OF IONOSPHERIC GRAVITY WAVES

6.1 A METHOD FOR THE MEASUREMENT OF IONOSPHERIC GRAVITY WAVES

In this chapter a method for extracting information about GWs from data acquired using the USU Dynasonde is discussed. This method can be subdivided into several procedures and a flow chart illustrating this method is shown in Fig. 6.1. This method is designed to be applicable for all parameters which can be measured by the Dynasonde. In the following discussion, the range parameter is used as an example. The Dynasonde transmits radio waves into the ionosphere where they are reflected and the reflected signal is termed an echo. The received echoes are stored by the computer as I-mode or K-mode data files depending on which sounding program has been executed.

We can consider the ionosphere as an unknown system, or a "black box" and the information relating to disturbances or geophysical events occurring in the ionosphere is "hidden" in the noisy data. In this method, the data is first input into an adaptive scaling system, which has been discussed in Chapter V. The adaptive scaling system uses a moving window which automatically tracks the echo trace and eliminates the second order reflection and interference. The selected echoes within the window consist of an O-mode and a X-mode trace. The O-mode and X-mode traces are readily separated because of the intrinsic 90^0 phase difference. Frequently, because of missing data, the recorded O-mode and X-mode traces are not continuous curves. Therefore, an interpolation and extrapolation procedure is applied after the O-mode and X-mode separation. As discussed in Chapter III, since it is impossible to use a single function to describe

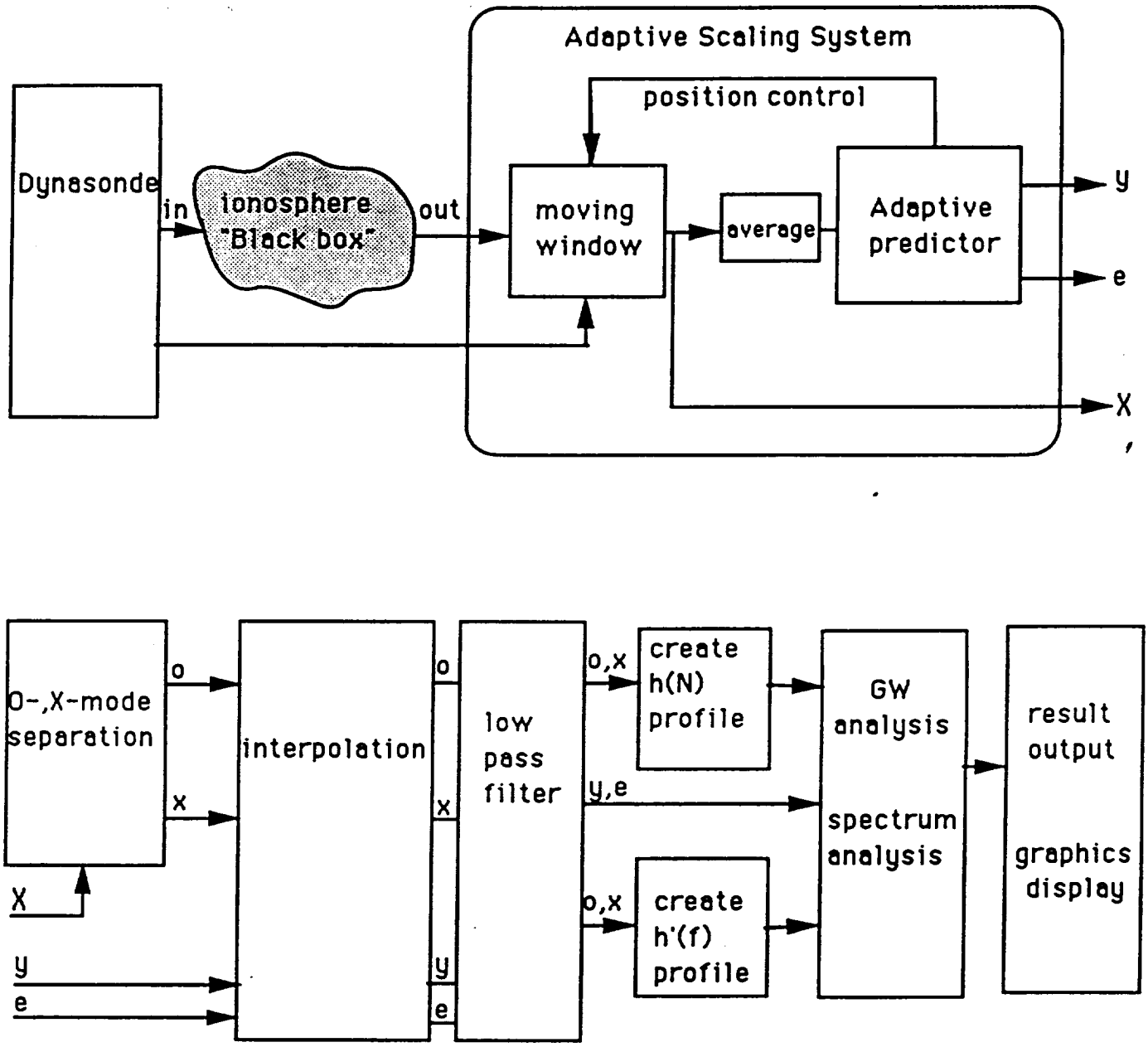


Fig. 6.1 A schemetic of the method of Gravity Wave measurements

an entire O-mode or X-mode trace, we divide the echo traces into several segments. For the I-mode plot these segments can represent the E, F1, and F2 layers and for the K-mode plot they are time segments. For any individual segment of the echo trace, a first order or second order polynomial formula can be used to approximate the missing points. Higher order interpolation does not necessarily increase the accuracy. In practice, if the signal is weak there will be soundings that have long segments of missing data, in which case the data are ignored. U.R.S.I [1978] suggests that traces due to very weak reflections should be ignored, since these traces seldom represent phenomena which can be studied effectively. In addition to the O-mode and X-mode traces, the estimated range value y and the error signal ε are also passed through the subsequent subprocedures as reference signals.

During the GW analysis we are interested only in the low frequency part of the signal spectrum since GWs have long period which has been discussed in Chapter II. Noise due to quantization, finite register length effects, and interpolation exist in the high frequency band. A low pass filter is applied after the interpolation in order to smooth the echo traces. We use an IIR digital filter designed by using a bilinear transformation from a second order Butterworth analog low pass filter. Medium scale GWs have periods of about 20-60 minutes. The filter pass band is chosen as f_p=4 cycle/(60 minutes). The digital low pass filter transfer function is given by

$$H(z) = \frac{0.206572(1+2z^{-1}+z^{-2})}{1-0.3695273z^{-1}+0.1958157z^{-2}} \qquad (6.1)$$

This function has two poles, $Z_{1,2} = 0.1847637 \pm j0.4020921$ lying within the Z-plane unit circle, hence the filter is stable. The output of the interpolation and the extrapolation subprocedure contains a clean and smoothed O-mode and X-mode traces, which are used to generate the h(N) and h'(f) profiles treated in the next subprocedure.

The detection and parameter estimation of GWs can be done in both the time and frequency domains, since an important characteristic of the GW is its periodicity. Power Spectral Density (PSD) estimation is helpful to detect the period of the GW. PSD can also be used to separate the GW from noise and interference, and to find the power in the GW. The power in the GW is an important parameter in deriving the GW source and direction of propagation. The PSD is obtained by computing a periodgram using an FFT program; since the data length is usually short (64-128 points), the PSD analysis can be implemented in real time.

Several data output and graphics programs have been developed to assist in the analysis of GWs using this method.

6.2 REMARKS ON APPLICATIONS OF THE DIGITAL SIGNAL PROCESSING

Some fundamental Digital Signal Processing (DSP) techniques are introduced in the GW measurement method. Implementations of the DSP techniques are discussed in this section.

6.2.1 THE DIGITAL FILTER

In the GW analysis, a digital filter is used to reduce the noise and interference. The frequency components of interest in GWs range from zero to about 1 cycle per hour. Therefore, a low pass filter is suitable for reducing the noise and interference in the data. In terms of the

impulse response, there are two kinds of digital filters which can be applied: the Finite Impulse Response (FIR) filter and the Infinite Impulse Response (IIR) filter. A FIR filter has an output which is dependent on its present and past input,

$$Y(Z) = \sum_{n=0}^{N-1} a_n X(Z) Z^{-n} \tag{6.2}$$

where a_n, n=0, 1, 2 ..., N-1 are the filter coefficients. X(Z) and Y(Z) are the input to and output from the filter.

The FIR filter has both advantages and disadvantages in this application. A FIR filter can be designed with a linear phase, which is preferable for ionogram trace signal filtering. A phase distortion can cause errors in the measurement of the speed and wavelength of the GW.

A FIR filter is inherently stable because there is no feedback in the filter structure. A disadvantage of the FIR filter is that a high order is needed in order to achieve a specified system amplitude response. For a two hour measurement obtained with a sampling rate of 2 minutes (the period between soundings), there are only 60 samples. A FIR filter usually has a length of more than several tens of coefficients. Thus, for short data lengths, the filter will cause serious signal distortion.

An IIR filter has a feedback loop and its output depends on the present and the past inputs as well as the past output. A general form of the IIR filter can be expressed by

$$Y(Z) = \sum_{n=0}^{N-1} a_n X(Z) Z^{-n} + \sum_{m=1}^{M-1} b_m Y(Z) Z^{-m} \qquad (6.3)$$

where a_n and b_m are the filter coefficients; X(Z) and Y(Z) are the system input and output, respectively. The advantage of the IIR filter is that only a few orders (first or second order) are needed to obtain the same amplitude response as the FIR filter. Therefore, it is suitable for short data samples. However, normally an IIR filter is not a linear phase filter and when used in the detection of GWs, it could effect the measurement of velocity and frequency of the GW. In addition, an unstable IIR filter could introduce a false oscillation in the data.

6.2.2 THE TIME DOMAIN AND FREQUENCY DOMAIN ANALYSIS

Both time and frequency domain analyses are necessary in the study of GWs. Frequency domain analysis yields the power spectral density of the signal, which is a more precisely quantitative measurement of the frequency than the time domain analysis. It is valuable to interpret irregularities or to compare the theoretical analysis with the experimental measurements in the frequency domain. There are two main approaches to estimate the frequency of the spectral lines in a time series: the Fourier method and the "exponential spectrum estimators" [Marple, 1980]. The Fourier method can be easily implemented in real time by using the Fast Fourier Transformation (FFT). It provides equally spaced frequency components. However, for very short data samples, the discrete Fourier transform can cause serious frequency leakage and poor

frequency resolution. The "exponential spectrum estimator" fits arbitrarily-spaced frequency components to the data. An example of such a technique is the Maximum Entropy Method (MEM). The MEM technique has no limit on spectrum resolution other than that imposed by signal-to-noise constraints. It was shown by Marple [1982], for example, that above approximately -20db S/N the MEM had better spectral resolution than the Fourier method. But the MEM has two disadvantages: (a) MEM assumes sharp, narrow, spectral lines. For a non signal or a single "pulse" GW, MEM can produce a false spectrum with several sharp peaks. (b) MEM is susceptible to noise. Noise can cause splitting or shifting of the spectral lines.

Time domain analysis is more efficient in observing signals with only a few cycles of oscillation. If there are data gaps or if the signal noise ratio is low and the data is not long enough, the frequency is barely detectable in the frequency domain. Large uncertainties in the frequency estimate could also occur if the continuous measurement is of the order of a few hours and the GW period is several tens of minutes. Time domain analysis could be applied to such data with fewer problems being encountered in the frequency domain analysis. Recent research shows that GWs have multiple frequency components and their wave-length varies as it propagates. Time domain analyses can not distinguish these kinds of variation.

6.2.3 THE WINDOW FUNCTION

A DC removal operation and window function is generally applied before the power spectrum calculation. The window function is used to modify the characteristics of the DFT frequency output response. Windows are used in conjunction with a DFT to reduce the spectral

leakage, reduce scalloping loss, and to provide variable resolution. The windowed DFT output is the convolution of the DFT of the window function and the DFT of the input signal. It is necessary to use a window in processing the echo trace to reduce the frequency leakage caused by short data segments. The Hanning window is used in this application; its weighting function is defined by

$$a(n) = 0.5[1 - \cos(2\pi\frac{n}{N})] \qquad n = 0, 1, 2, ..., N-1 \tag{6.4}$$

The frequency response for this function is shown in Fig. 6.2. Note that the highest sidelobe level is -32db and the sidelobe fall off is approximately -18 db per octave.

6.3 ONE TRACE TRACKING

In practice, the adaptive scaling system is very sensitive to the system parameters, such as the convergence constant μ, the adaptive linear combiner length, and the moving range window size. The system is also dependent on the signal itself. Large gaps in the echo traces and low signal-to-noise ratio usually cause a loss of tracking. A modified adaptive scaling system and a schematic of GW measurements are presented in Figures 6.3 and 6.4. In this case, the adaptive linear combiner output y is used to estimate the O-mode or X-mode trace, rather than an average

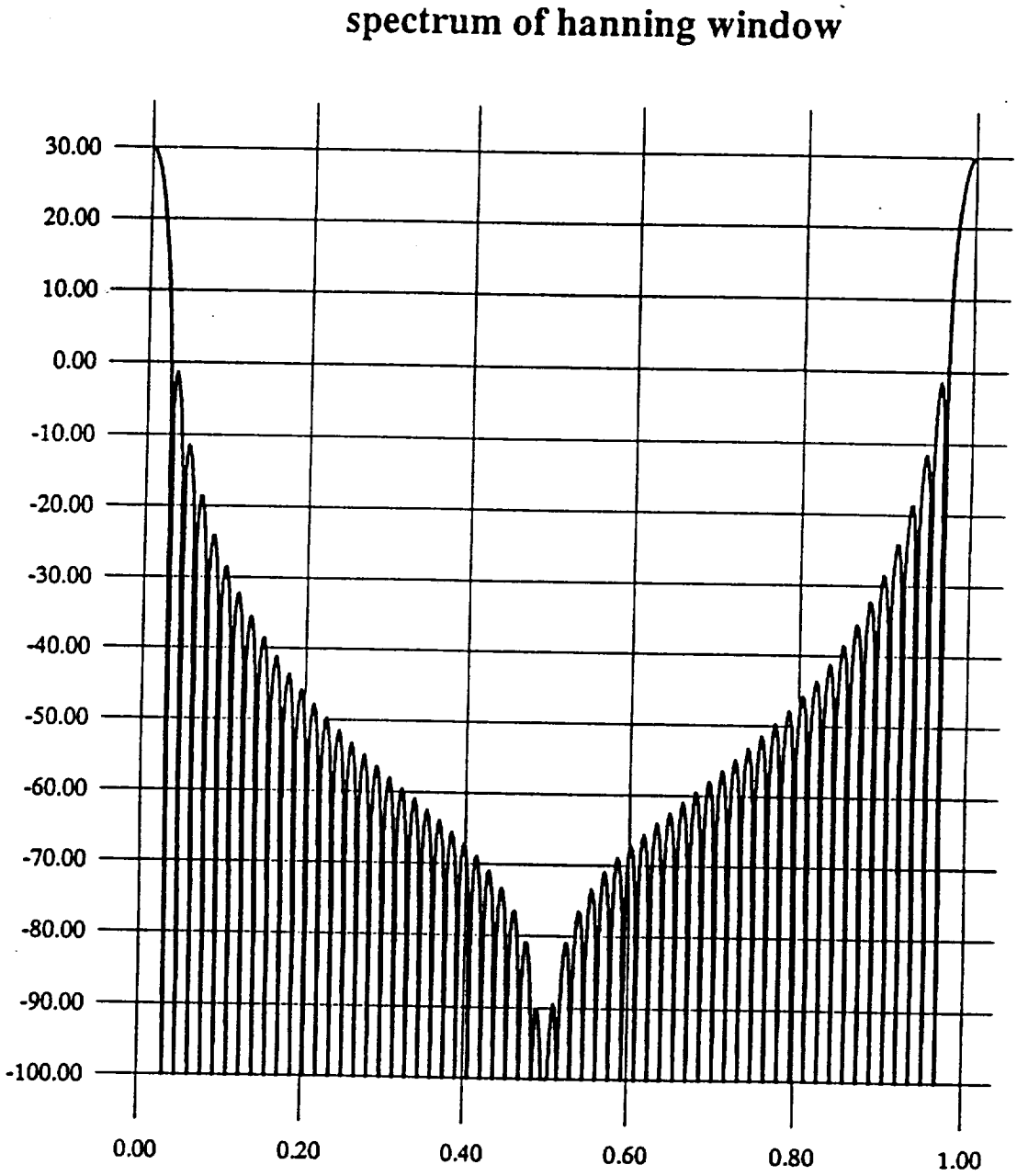


Figure 6.2 The frequency response of the Hanning window function

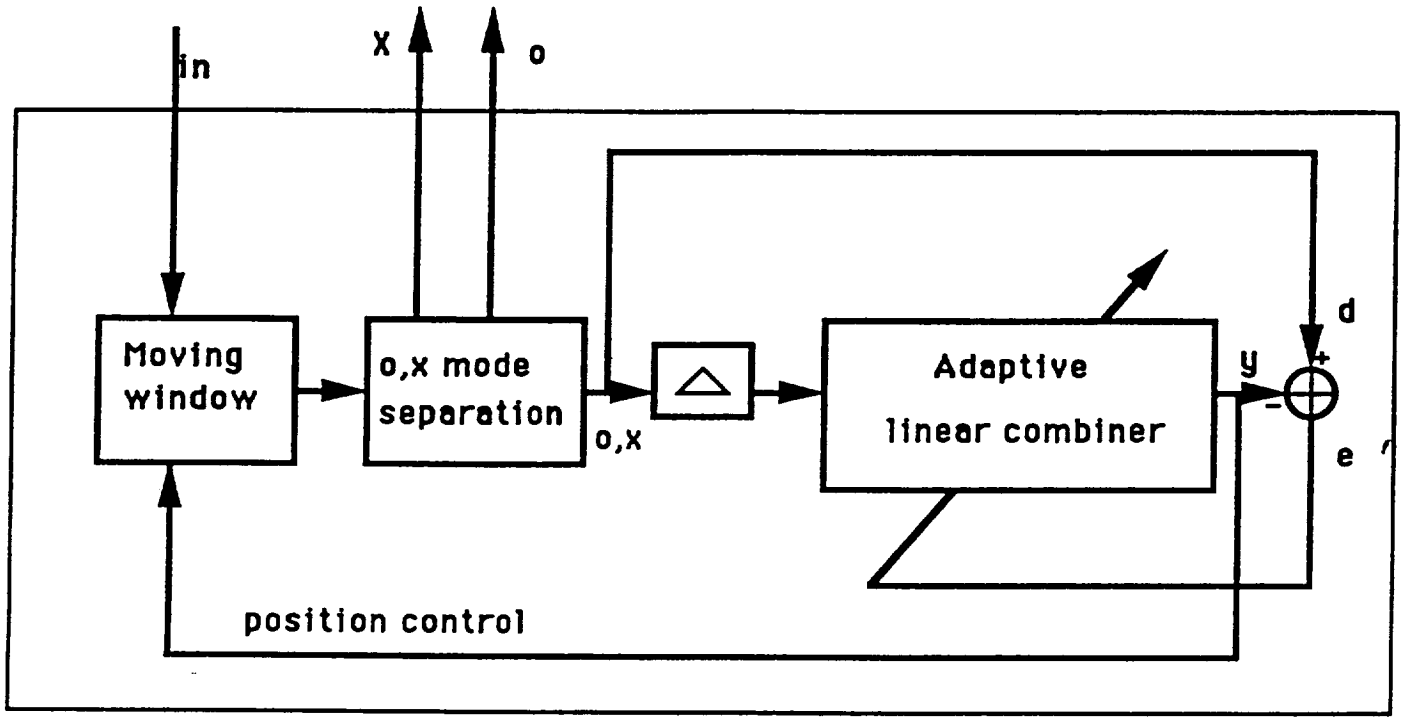


Fig. 6.3 Modified Adaptive scaling system for one trace tracking

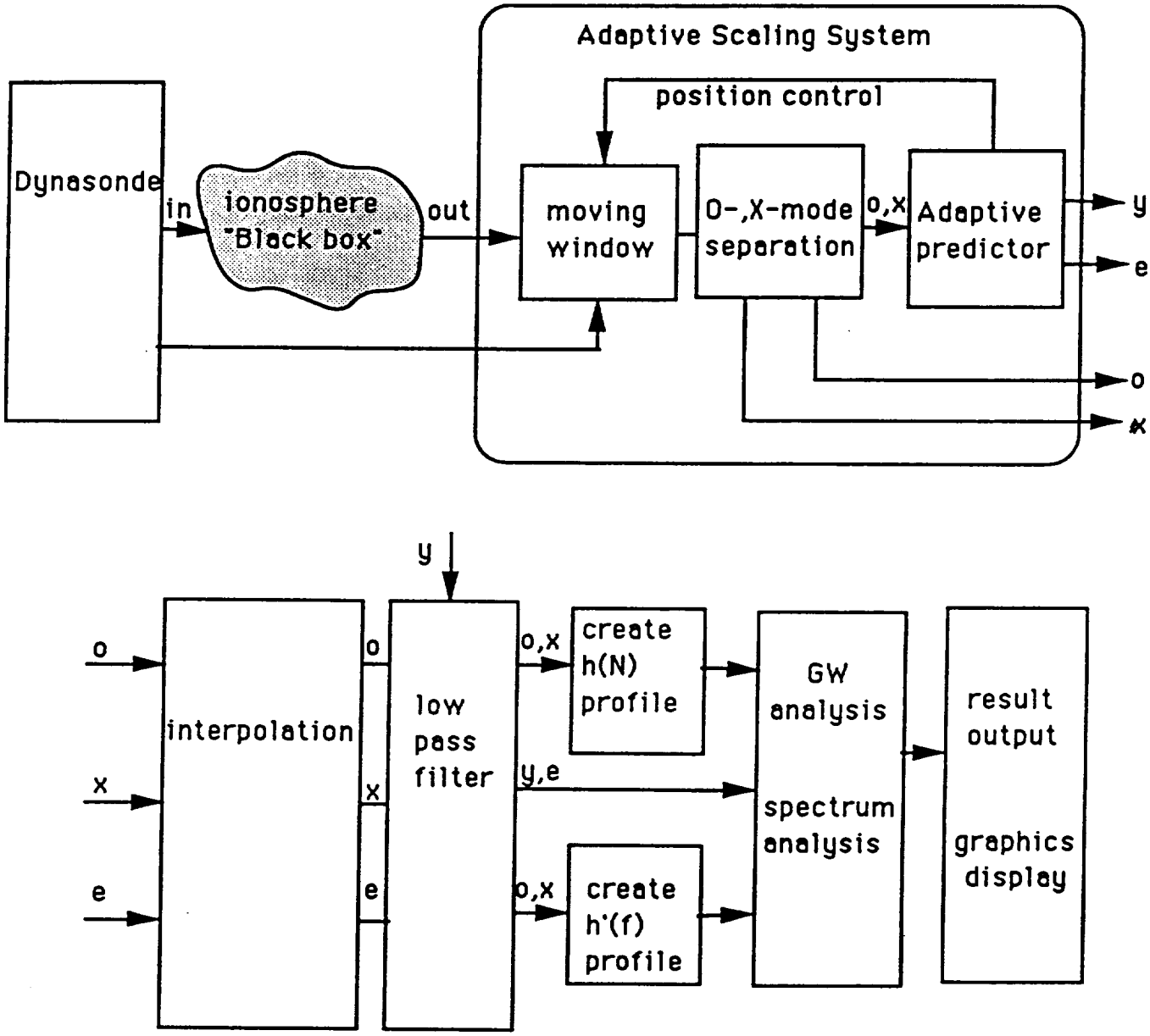


Fig. 6.4 The Modified method of Gravity Wave measurements for one trace tracking.

trace passing through the range window. The O-mode and X-mode separation sub procedure is carried out immediately after the range windowing operation and only one trace, the O-mode or X-mode trace, is input into the adaptive predictor. Hence, this method is called one trace tracking. The modified adaptive scaling system improves the system performance, especially in the high frequency segment, and at frequencies near the critical frequency where the O-mode and X-mode have a large range difference. The modified adaptive scaling system permits use of a small window size so that less noise and interference are allowed to pass through the range window and to be input into the predictor. In other words, the adaptation performance is improved due to less noise. Another advantage of the one trace tracking method is that the input signal of the predictor is preselected by an added phase detection process, the O-mode and X-mode separation. For the averaged input signal of the predictor, loss of tracking occurs if the signal-to-noise ratio is low, if there are missing segments of data, or if the window size is too small or too large. These kinds of problems are rarely occur in the modified, more robust adaptive scaling system. Moreover, the signal y, which is the optimal estimate of the O-mode or X-mode trace is made more useful. The y signal is a continuous signal so no interpolation is needed. The wave form and the PSD output of the optimal estimate value y should be identical to that of the O-mode or X-mode signal. Thus, y is an ideal signal for the GW analysis in both the time and frequency domains.

Figures 6.5 and 6.6 illustrated the one trace tracking of the O-mode and X-mode curves respectively.

6.4 GRAVITY WAVE OBSERVATIONS

The methods discussed in this report are examined by using the data collected by the USU Dynasonde during the fourth Worldwide Atmospheric Gravity Wave Study (WAGS) campaign November 10-24, 1990.

A landscape of the ionogram over a 30 hour interval with full frequency band from 1.6MHz to 16MHz is shown in Figure 6.7 in order to overview the dynamic behavior of the ionosphere. It records the variation of the virtual height (h') with the time from 0001:00 Nov. 10, 1990 UT to 0619:00 Nov. 11, 1990 UT. The color bar on the right hand side of the Figure represent frequency bands and each color corresponds to a frequency band of 1 MHz. Examining this "map" we found that there were frequently wave motions throughout this long time period. Most wave motions occur in the F layer within the range from 200km to 500km. The best frequencies for observing GWs were found to be 1 MHz-3 MHz during the night and 5 MHz-12 MHz during the day.

Figure 6.8 illustrates a multi-frequency (2.0, 2.5, 3.0, 3.5 and 4.0 MHz) K-mode ionogram using the same data in Figure 6.7. Very definite wave structure on the h' variation was observed: (1) between 0500:00 Nov. 10, 1990 UT and 1300:00 Nov. 10, 1990; (2) between 0420:00 Nov. 11, 1990 and 0619:00 Nov. 11, 1990 UT. Both events were observed at night and the first interval lasted for 8 hours.

Magnified plots of the event which occurred between 0500:00 Nov. 10, 1990 UT and 1300:00 Nov. 10, 1990 is shown in Figures 6.9a and 6.9b, where the separation of O-mode and X-mode traces have been plotted respectively. The sounding frequencies are listed in the legend. Waves appear to have a single frequency sine wave form with a period of approximately 50 min. Waves are seen clearly when the sounding frequency is less than 4 MHz in Figure 6.9a, but only

few waves appear on the ionogram and only within a small frequency range near 3 MHz in Figure 6.9b.

Figure 6.10 shows the PSD of GWs observed in Figure 6.9. The measured signal period is approximately 50 min. Two additional PSD are presented in this Figure, which are the PSD of the echo trace estimate (y) and the error signal (e). The GW is detected from the PSD of the y signal and the PSD of the error signal depicts an uncorrelated wide band noise. These results agree with the theoretical analyses in Chapter IV.

Figures 6.11a and 6.11b show the magnified O-mode and X-mode ionograms of Figure 6.7 over the time period from 0416:00 Nov. 11, 1990 UT to 0916:00 Nov. 11 1990. The wave-form appears similar to the pulse response of a dynamic system. A strong "pulse" started at 0416:00 UT, caused a fast range increasing with a slope of approximately 125km/hour while the "response" lasted for two hours with very little attenuation. This event suggests a large instantaneous energy input to the ionosphere. The period of the oscillation of the "response" increases every cycle and the first cycle takes about 50 min. Similarly, as illustrated in Figure 6.9, O-mode traces appear over a wider frequency range than do the X-mode traces as shown in Figures 6.11a and 6.11b.

Figure 6.12 illustrates the PSD of the wave-forms shown in Figure 6.11. The detected period of the GW is 50 min. which agrees with the results measured in the time domain.

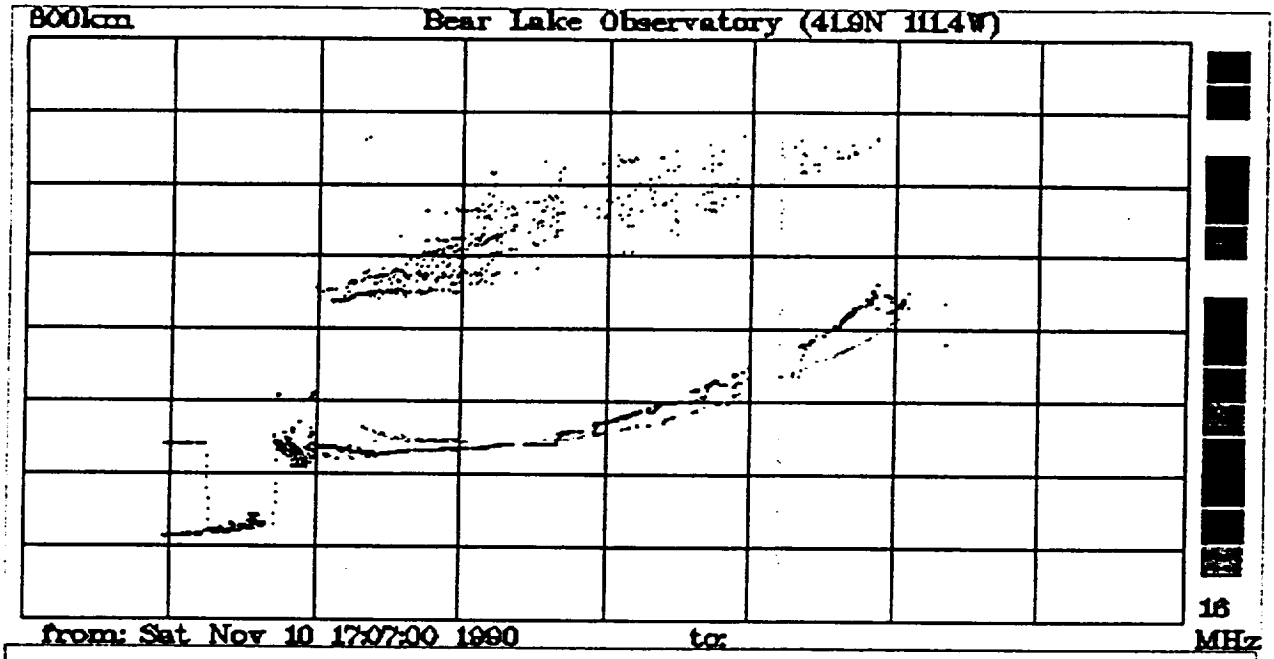


Fig.6.5 I-mode ionogram using adaptive scaling method
moving window size=122, L=8, mu=0.1E-7
One trace tracking(O-mode), MLMS algorithm

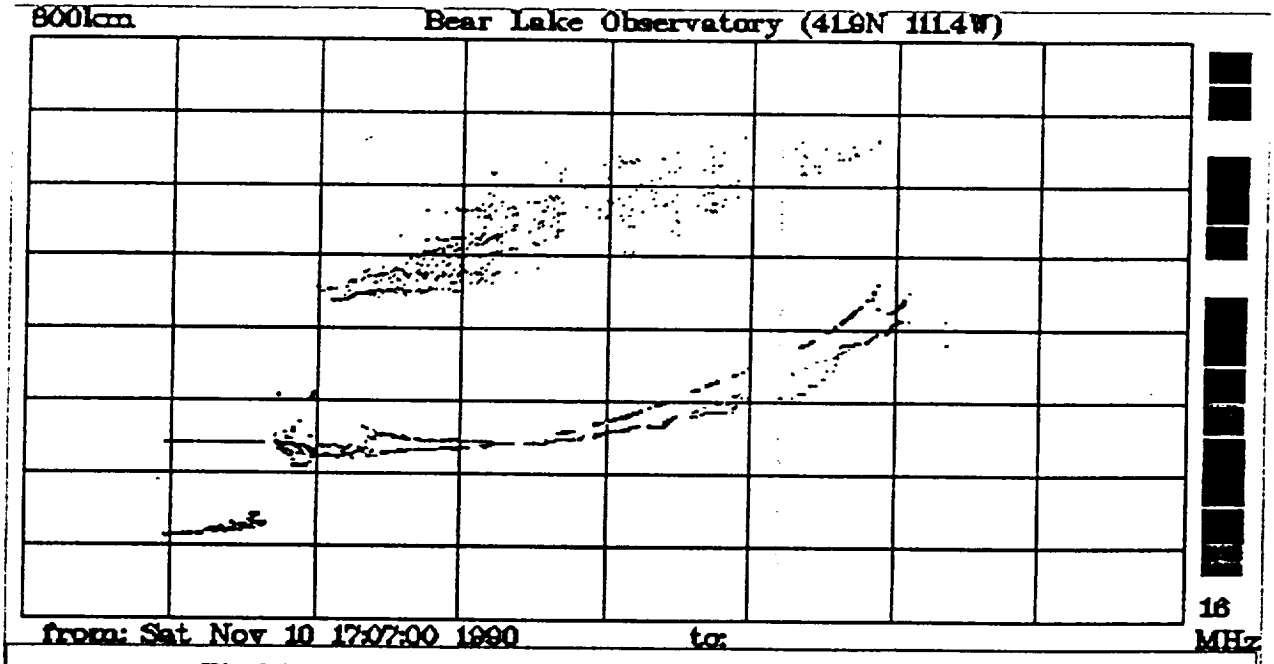


Fig.6.6 I-mode ionogram using adaptive scaling method
moving window size=122, L=8, mu=0.1E-7
One trace tracking(X-mode), MLMS algorithm

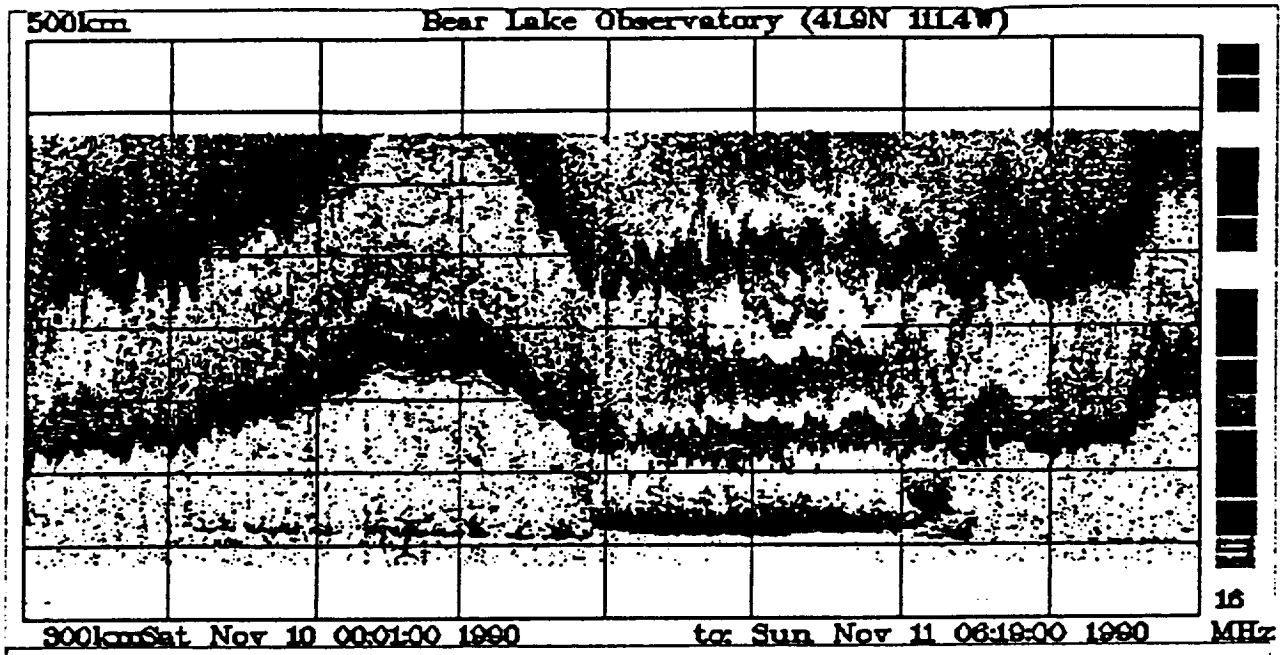


Fig.6.7 30 hours ionogram
WAGS1 from #1 to #600

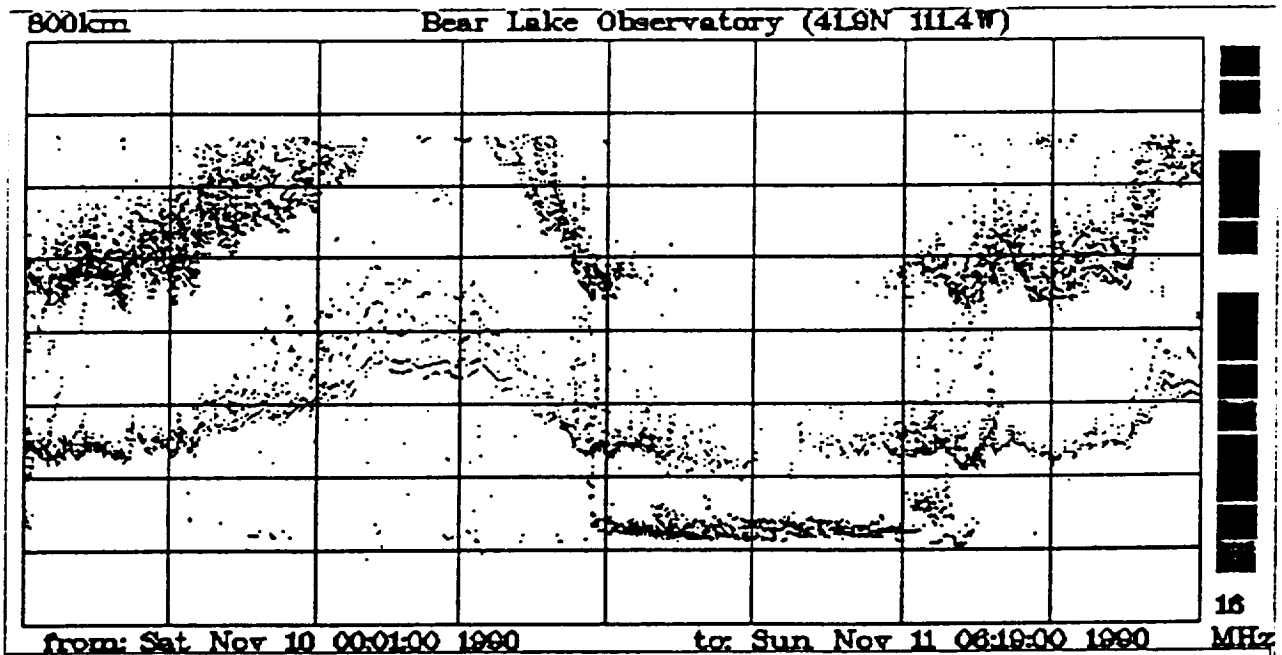


Fig.6.8 30 hours ionogram
WAGS1 start from #1

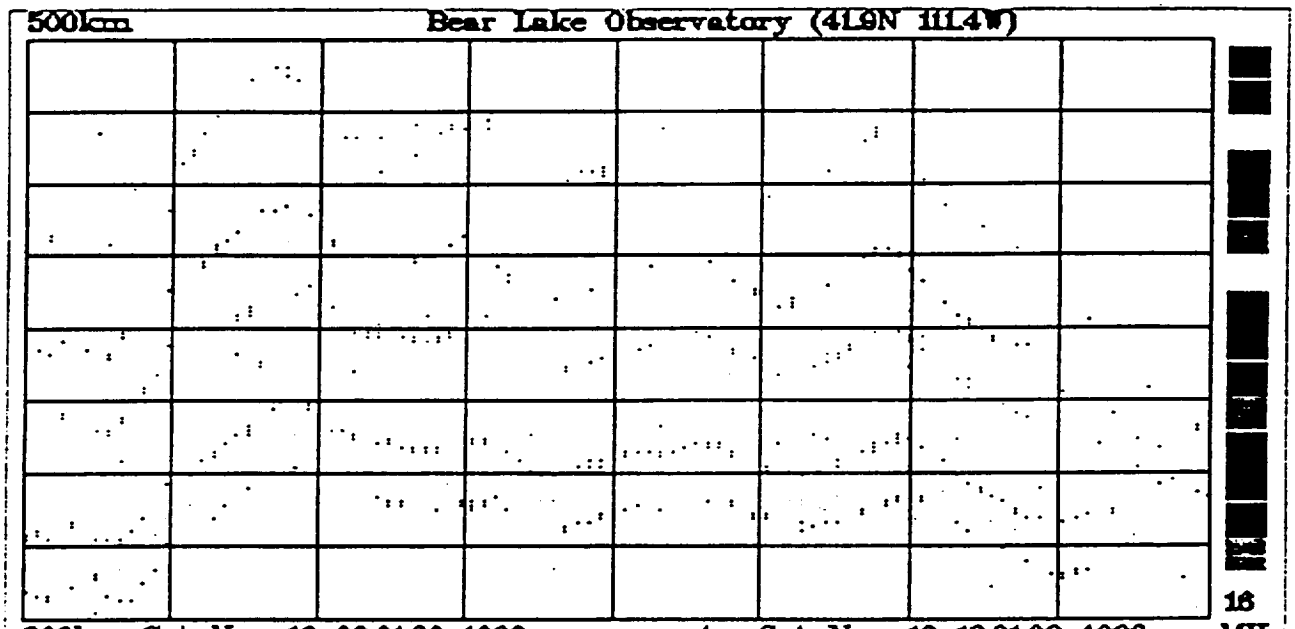


Fig.6.9a Gravity Waves observed
K-mode o-trace, file dir: WAGS1 from #160 to #260
frequencies: 1.9 2.1, 2.3, 2.9, 3.2, 3.5MHz.

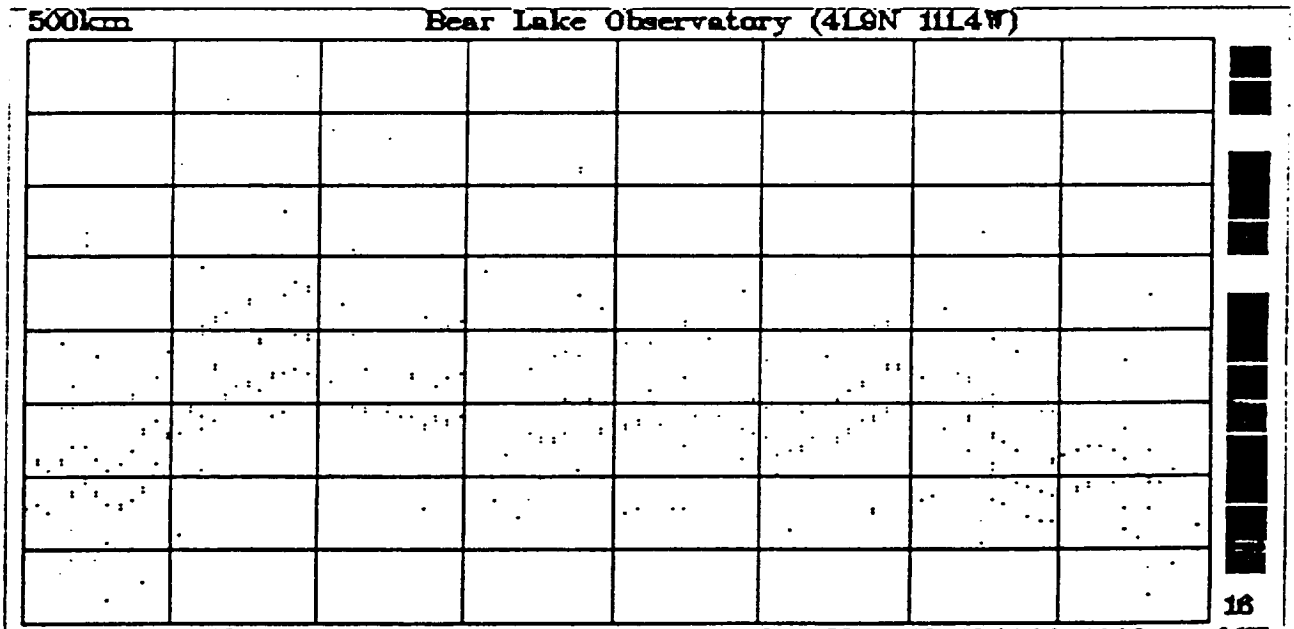


Fig.6.9b Gravity Waves observed
K-mode x-trace, file dir: WAGS1 from #160 to #260
frequencies: 1.9 2.1, 2.3, 2.9, 3.2, 3.5MHz.

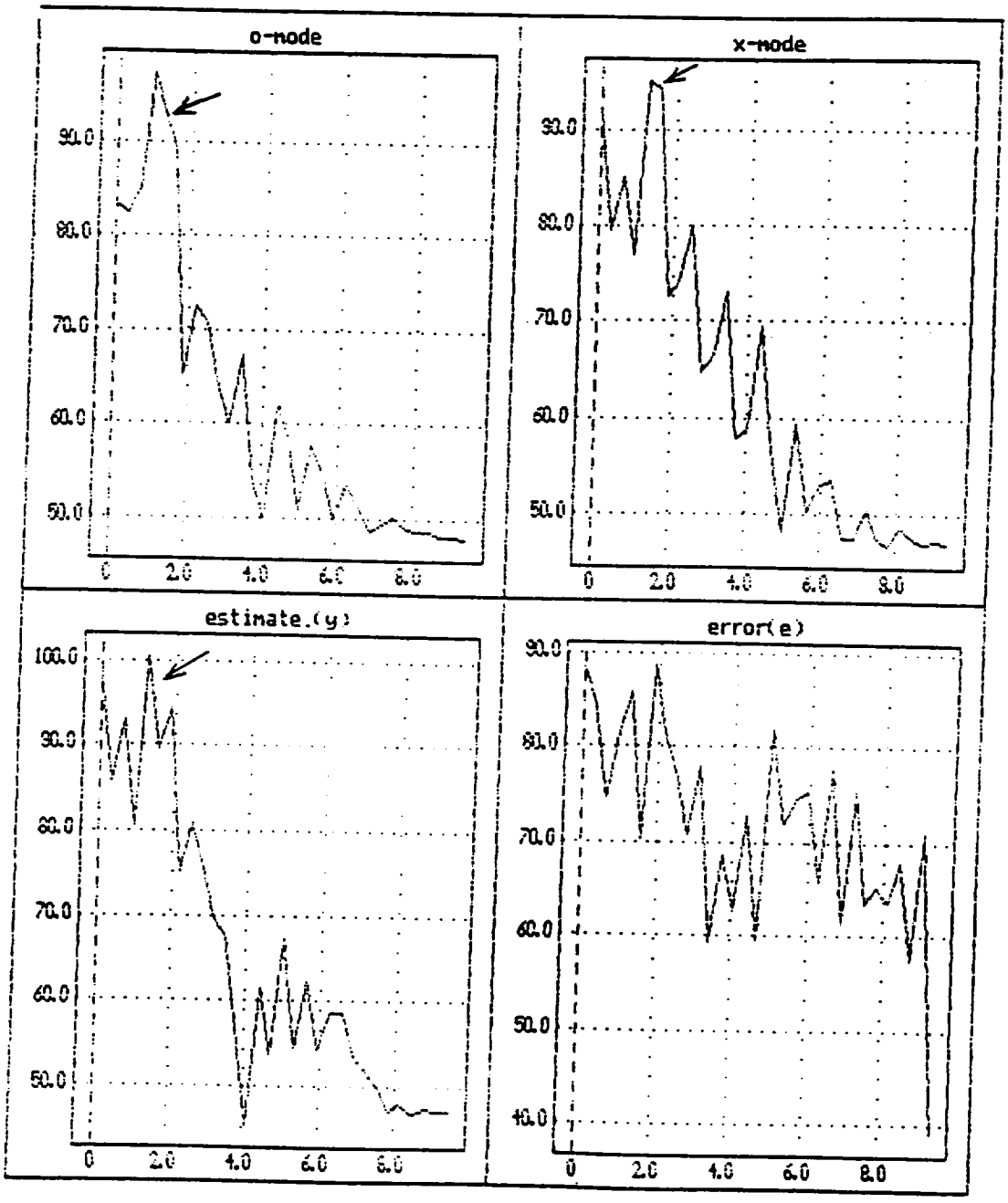


Figure 6.10 The PSD of the Gravity Wave observed

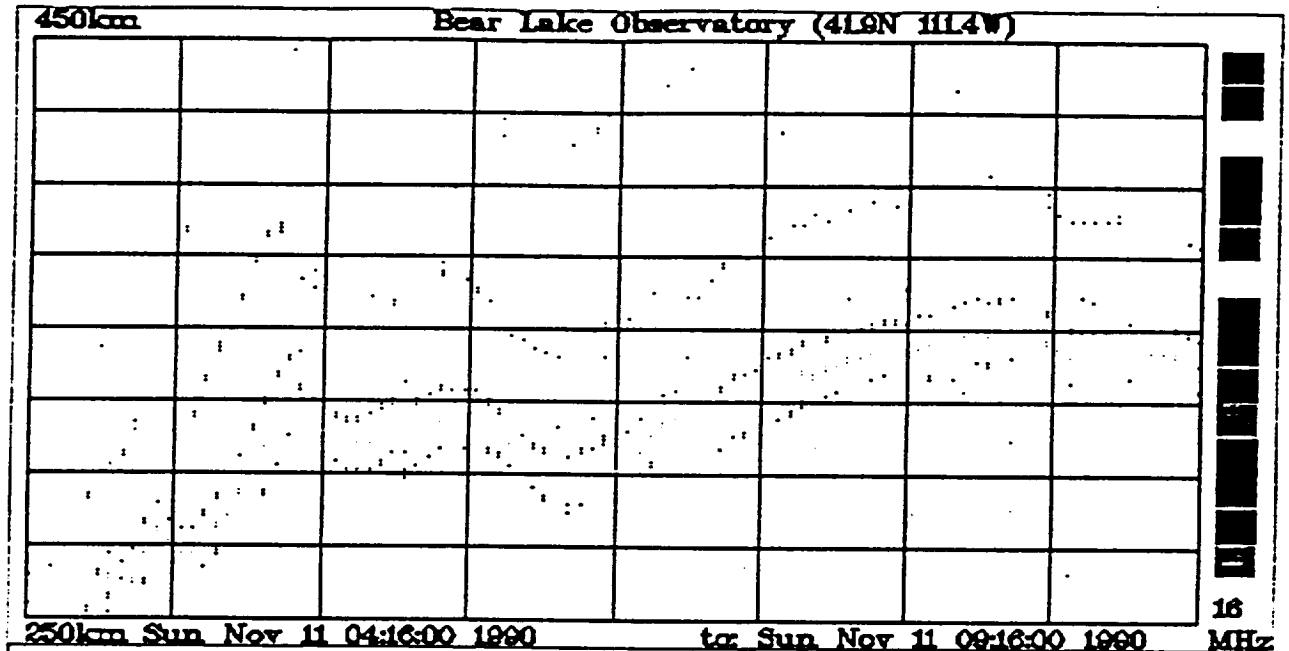


Fig.6.11a Gravity Waves observed
K-mode O-trace, file Dir: WAGS1 from:#560 to #660
frequencies: 1.9, 2.1, 2.3, 2.9, 3.2, 3.5MHz

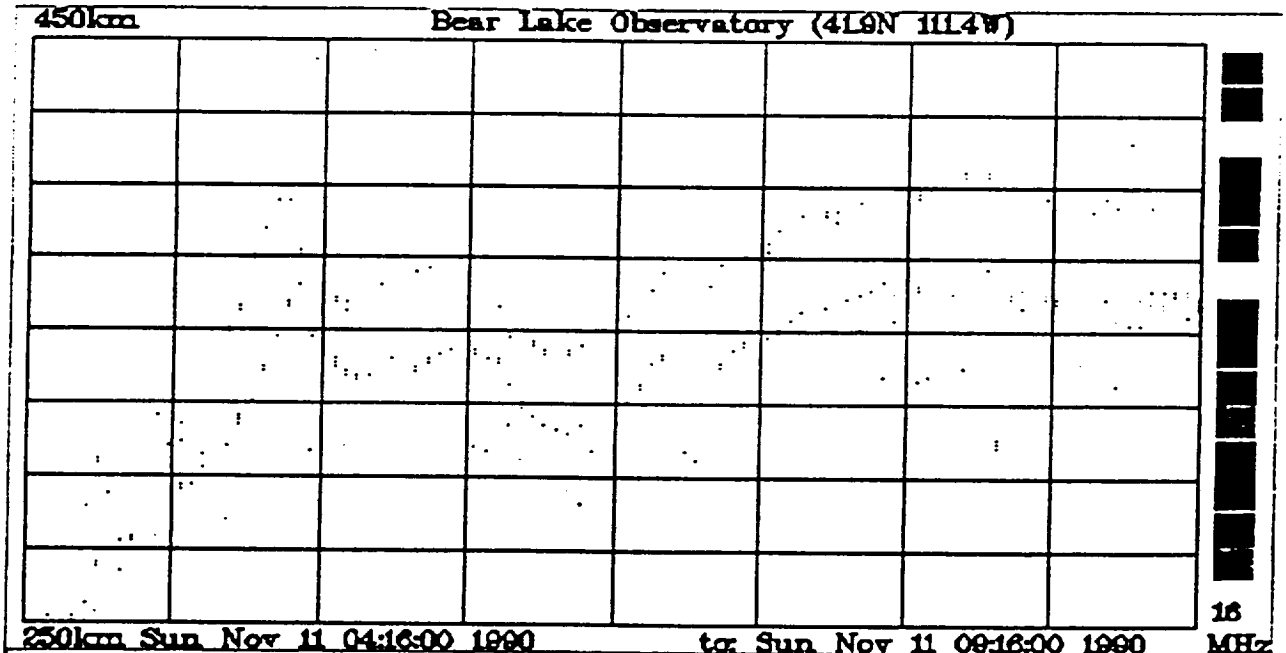


Fig.6.11b Gravity Waves observed
K-mode X-trace, file Dir: WAGS1 from:#560 to #660
frequencies: 1.9, 2.1, 2.3, 2.9, 3.2, 3.5MHz

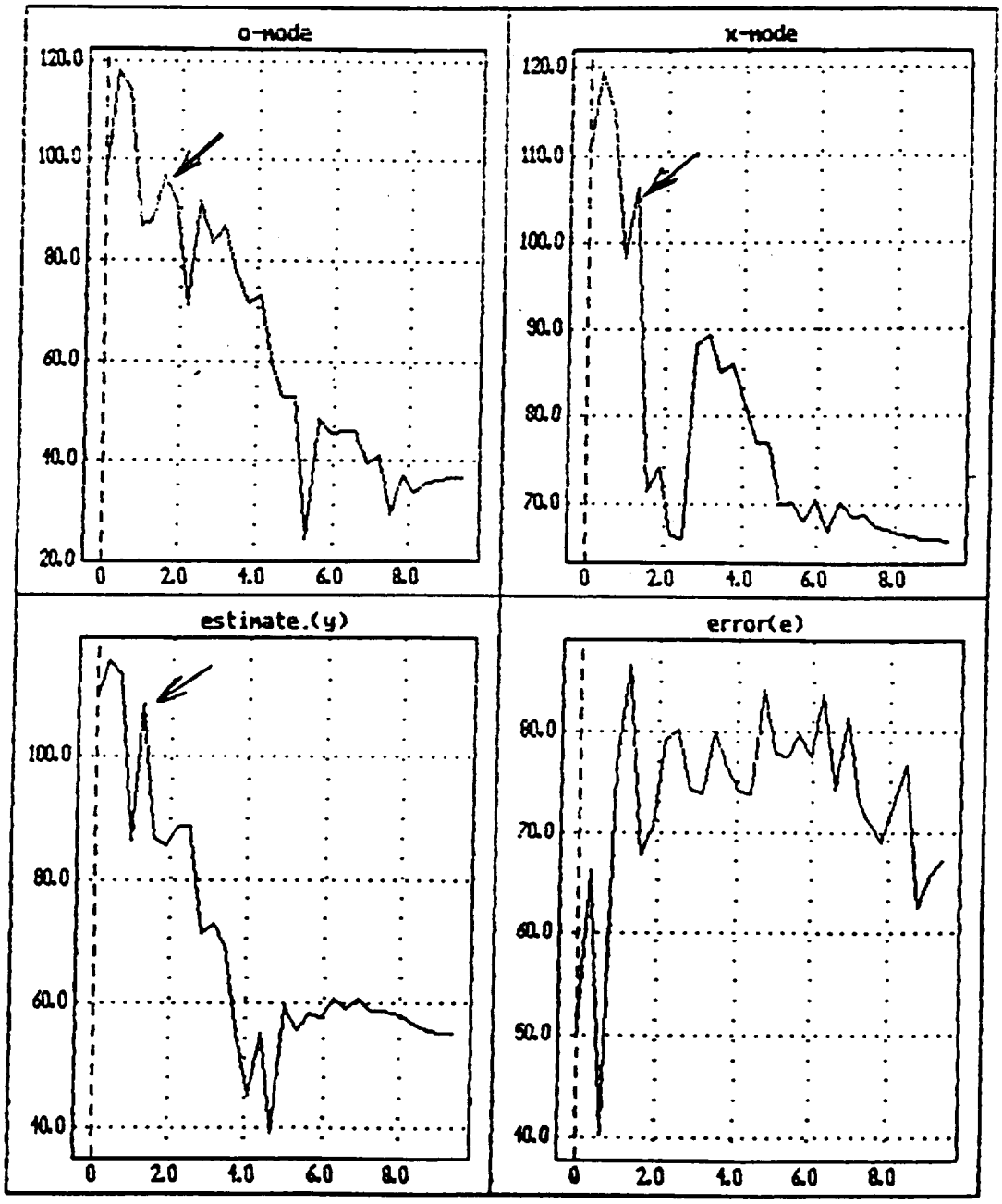


Figure 6.12 The PSD of the Gravity Wave observed

CHAPTER VII

CONCLUSIONS AND FURTHER CONSIDERATIONS

1. A new method of automatic ionogram scaling, called the *adaptive scaling* method, is proposed in this report. Based on system theory, the adaptive scaling method uses the optimal estimation technique to deal with the random signal in a dynamic system. The theoretical analysis has been presented in detail and the technique of implementation of this method is discussed. The adaptive scaling method, which can extract the echo trace from noisy radar data with few constraints, has provided a new means of autoscaling for input to N(h) profile derivation. Simulation results have shown good performance of this method.

2. A data processing technique designed for the measurement of ionospheric gravity waves using the USU Dynasonde has also been presented in this report. This method includes several sub procedures: the sounding data acquisition, adaptive scaling, O-mode and X-mode trace separation, data interpolation and extrapolation, digital filtering, h'-f and h-N profile generation, windowing, PSD analysis, and color graphics presentation.

3. Several applications of digital signal processing in the measurement for extracting echo traces are implemented. Both time and frequency domain analyses for the detection of the ionospheric gravity waves have shown to be valuable analysis tools. But it is necessary to seek a suitable PSD estimation algorithm to increase the frequency resolution for short data samples.

4. An advantage of the proposed method is that it can be easily implemented in real time. All subsequence subprocedures are independent function blocks and can easily be parallelized by distributing each of them into a single processor in a pipeline configuration. The Speed Up of execution can reach approximately up to the number of the subprocedures.

5. Data obtained by the USU Dynasonde near Garden city (41.9N, 111.4W) from November 10, 1990 to November 24, 1990 has been analyzed for the presence of GWs. Most waves were observed in the F layer and within the frequency range between 2 MHz and 3.5 MHz (during the night) and between 5 MHz and 12 MHz (during the day).

6. Two interesting events were observed during the data processing. The first event occurred from 0500:00 Nov. 10, 1990 UT to 1300:00 Nov. 10, 1990 UT and is a long lasting sinusoidal wave with a period of approximately 50 min.. The second event was observed between 0420:00 Nov. 11, 1990 UT and 0619:00 Nov. 11, 1990 UT and was system-pulse-response-like wave with a large instantaneous energy input into the ionosphere.

7. The use of other parameters of the Dynasonde will provide much valuable information in the study of GWs. Particularly, the echo phase can be the first variable to be put into the analysis because it provides high range resolution.

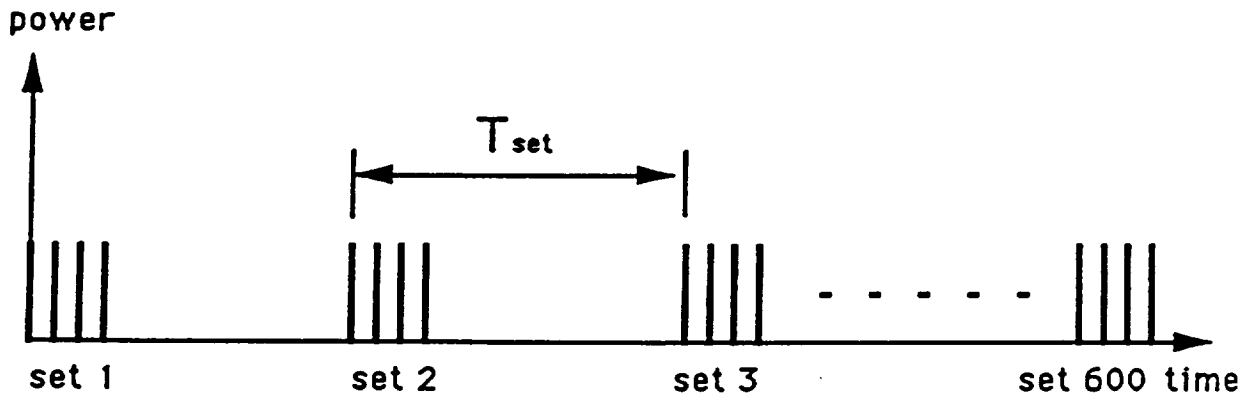
APPENDIX A

THE DATA STRUCTURE

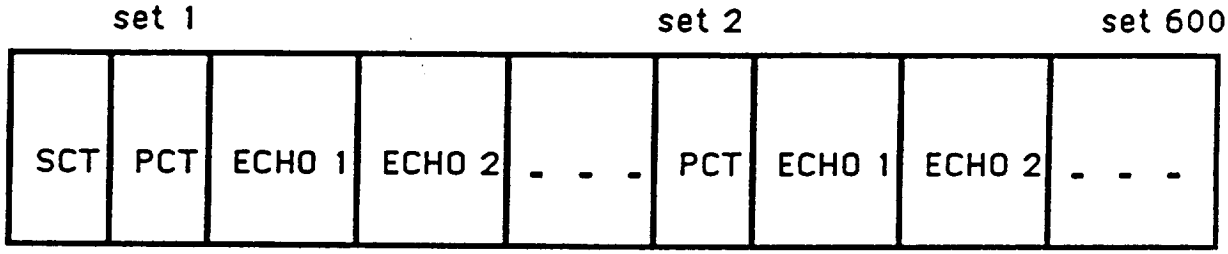
This appendix is a description of the data structure used in data acquired with the USU Dynasonde.

The Dynasonde transmits a pulse set every T_{set} where each pulse set consists of four pulses whose configuration is shown in table 3.1. T_{set} is a constant that is selected before the sounding program starts, which also defines the sampling rate of the I-mode and K-mode soundings. Large T_{set} will cause short data problem as discussed in Chapter VI. Small T_{set} helps to increase the data size and to improve the frequency resolution in the PSD analysis. However, the penalty is large storage requirements and low processing speed. The smallest T_{set} is the time interval that a radio wave travels twice the distance of 720 km.

A sounding data file is stored in the computer and it consists of a Sounding Configuration Table (SCT), multiple Pulse Configuration Tables (PCT) followed by a sequence of echoes (see Figure A1). A SCT contains the system information, transmitter and receiver parameters, sounding type, CRAM parameters, and SKY board parameters. The PCT contains parameters such as PCT index, frequency, IF and RF attenuation, and the SKY header which records the pulse and echo parameters in the SKY board. Following each PCT is a sequence of X, Y values for each echo. Several parameters are stored: the echo range bin and eight quadrature amplitude values of the X-Y echo pair. The sounding data structure is shown in figure A1 and the data type definitions of the PCT and echoes are listed in Figure A2.



(a)



(b)

Fig . A1 (a) The time sequence of pulse sets.
(b) The sounding data structure stored in the computer.

Figure A2 The data type definition of the PCT and echoes

```
typedef struct {          /* set of XY pairs */
    word RgBn;
    word NmPrs;
    word X0,Y0,X1,Y1,X2,Y2,X3,Y3,X4,Y4,X5,Y5,X6,Y6,X7,Y7;
    } echo_t;


typedef struct {        /* sky data header */
    word Nine;              /* 9 if sky done */
    word NmPs;              /* Number of pulses */
    word TOf;               /* Time Offset     */
    word MxA;               /* Max Amplitude          */
    word MnT;               /* Mean Threshold         */
    word NmEs;              /* Number of echoes */
    } sky_h;


typedef struct {        /* Pulse configuration table */
    word Len; /* Length of table in bytes */
    word Indx;          /* Index of PCT in sounding */
  freq_t Freq;
    word IFAttn;      /* dB */
    word RFAttn;
    sky_h SH; /* SKY header */
    } pct_t;
```

APPENDIX B

PROGRAMMING TOOLS

The Microsoft C and FORTRAN languages have been used for software development. The C language is a middle-level language between assembly and high level languages. It is much more flexible to use the C language to interface the system hardware and the application software. Thus, most of the subprocedure used in this report are C programs.

The Windowing, PSD, and FFT programs are carried out by using the FORTRAN language which is more suitable for scientific computation. Data transfer between C and FORTRAN is accomplished by writing to and reading from data files.

The Microsoft Programmer's Work Bench (PWB) was used for program development in this report. The PWB provides an convenient edit, compile, and link environment to deal with large codes.

Make files called BROWS1.MAK and BROWS2.MAK were created to join the main program BROWS1.C or BROWS2.c with the other programs (SCHADUE.C, SCREEN.C, PLOT.C, WINDOW.C, HARDWARE.C, MISC.C). Programs BROWS1.C and BROWS2.C are used for the I-mode and K-mode trace analyses respectively.

The main program BROWS1.C is listed in appendix C.

APPENDIX C
MAIN PROGRAM LIST

```
/*****************************************************************/
/* This is the main program to simulate the adaptive scaling   */
/* method and to creat the I-mode and K-mode ionogram.         */
/* It reads data from the Optic disk (H:), parameters from     */
/*  brows1.dat file and output brows1.plt file for results     */
/* ploting. The following programs are needed to link with this*/
/* program:     SKY_IO.OBJ,HARDWARE.OBJ,MISC.OBJ,WINDOW.OBJ,   */
/* SCREEN.OBJ, PRINTER.OBJ;                                    */
/* Large memory mode for compiling and /ST:40000 switch for    */
/* linking;                                                    */
/* The MS PWB is used as the program development environment.  */
/*****************************************************************/
#include <dos.h>
#include <conio.h>
#include <graph.h>
#include <gprintms.h>
#include <malloc.h>
#include <stdlib.h>
#include <string.h>
#include <stdio.h>
#include <ctype.h>
#include <stdarg.h>
#include <sys\types.h>
#include <sys\stat.h>
#include <math.h>
#include "window.h"
#include "misc.h"
#include "fais.h"
#include "schedule.h"
#include "config.h"
#include "screen.h"
#define size_of_pct sizeof(pct_t)
#define size_of_echos sizeof(echo_t);
#define F1 59
#define F2 60
#define F3 61
#define DirDef "h:\\wags1\\"
int   adapsize,o_trace,x_trace,multi_trace,MLMS,
LMS,plot_y,printswitch;
int analyze_size,tom_number=18;
char fig1[80],fig2[80],fig3[80];
int freq,whichplot=4;
double start_time,sec_per_frame,end_time,mid_time;
int first_data_file=1,in_data,n_pulse,time_int;
struct tm *newtime;
void plot_one_set(int file_indx ,fpos_t Pos,pct_t *P,FILE *F);
void arange(echo_t E,double *Pol0,double *amp);
```

```
void pulse_window(void);
void screen_phase(void) ,dialog(void);
int  tom_alg(double echo_amp[21],word number_of_echos);
void screen_1pulse(void),screen_4pulse(void);
void point(int ipulse, double  x,double y,int i_color,int i);
void print_handler(void);
double pmean(double(x),double(y));
double    prange(double p) ;
sct_t *S;
struct stat Dstat;
word LastPS,PulseSet = 0;
int NFMax=3000;
struct fili *file_list_addr;
int color[16]={
     15,1,8,3,4,5,11,14,2,9,10,11,12,13,6,7};
int f_band[16]={
     0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0};
double mean_range=160.,old_range=140.,new_range,delta_range;
double mean_range1,sum_range1,vari_range;
double    range_window=50.;
int F2ragen=0;
double    adapw[64],adapx[64],adapy,adap_init
,adape,adapmu=0.00000001;
/*0.000001*/
FILE *image, *brows;
unsigned numberbytes;
void main(void)
{
     char *CFD, FBuff[MaxLen],Buff[MaxLen],*Def = DirDef;
     struct WindowDef *W1,W2 = {
          0,1,0,80,Magenta,White,Black," Select a data file "    }
     FILE *F;
     int j,i,err,P=0;
     pct_t PCT;
     fpos_t Pos;
     int NotDone = 1;
     char ch[40],cch=' ';
     int size,pct_length=0;
     struct find_t Finfo;
     int Status;
     int i_pulse=SINGLE_PULSE;
     short videomode=_VRES16COLOR;
     if((image = fopen("I.DAT","w")) == NULL){
     wwprintf(1,"Unable to open %s","I.DAT");
     exit(1);

     }
     if((brows = fopen("brows1.DAT","r")) == NULL){
     wwprintf(1,"Unable to open %s","brows1.DAT");
     exit(1);
```

```
}
fscanf(brows,"%s %d",ch,&printswitch);
fscanf(brows,"%s %d",ch,&analyze_size);
fscanf(brows,"%s %d",ch,&adapsize);
fscanf(brows,"%s%lf",ch,&adapmu);
fscanf(brows,"%s%lf",ch,&adap_init);
fscanf(brows,"%s %d",ch,&tom_number);
fscanf(brows,"%s %d",ch,&o_trace);
fscanf(brows,"%s %d",ch,&x_trace);
fscanf(brows,"%s %d",ch,&multi_trace);
fscanf(brows,"%s %d",ch,&plot_y);
fscanf(brows,"%s %d",ch,&LMS);
fscanf(brows,"%s %d",ch,&MLMS);
fscanf(brows,"%s",fig1);
fscanf(brows,"%s",fig2);
fscanf(brows,"%s",fig3);
for(i=0;i<80;i++){
if(fig1[i]=='~')fig1[i]=' ';
if(fig2[i]=='~')fig2[i]=' ';
if(fig3[i]=='~')fig3[i]=' ';
          }
printf("printswitch%d\n",printswitch );
printf("analyze_size%d\n",analyze_size );
printf("adapsize%d\n",adapsize );
printf("adapmu %lf\n",adapmu );
printf("adap_init%lf\n",adap_init );
printf("tom_number%d\n",tom_number );
printf("o_trace %d\n",o_trace );
printf("x_trace %d\n",x_trace );
printf("multi_trace%d\n",multi_trace );
printf("plot_y%d\n",plot_y );
printf("LMS%d\n",LMS );
printf("MLMS%d\n",MLMS );
printf("%s\n",fig1 );
printf("%s\n",fig2 );
printf("%s\n",fig3 );
getch();
W1 = findwindow(WinCoords);
if (W1 == NULL) W1 = &DefaultWindow;
W1->Bcol = Green;
W1->Fcol = BWhite;
W1->Tcol = Magenta;
W1->Title = " Data file browser ";
CFD = getenv(DataDir);
W2.br = W1->br;        /* bottom of screen */
W1->br = W1->tr +2;
W2.tr = W1->br +1;
makewindow(2,&W2);
makewindow(1,W1);
if (!CFD){
     wprintf("  DATA directory ? > ");
```

```
        CFD = getstr(Def);
}
clearwindow(1);
if(!CFD){
        wprintf("No directory selected");
        exit(0);
}
shiftwindow(2);
sprintf(Buff,"%s*.*",CFD);
_setvideomode(videomode);
_print_switch(printswitch);
_settextjustify(LEFT_TEXT,TOP_TEXT);
_setfontpath( "c:\\gp" );
_setfontstyle( TRIPLEX_FONT );
_settextscaling( 2, 4, 2, 4 );
_settextcolor(0);
_clearscreen(_GCLEARSCREEN);
screen_phase();
dialog();
/****start for each file ********/
if ((Status = _dos_findfirst (Buff,
     _A_NORMAL|_A_ARCH|_A_RDONLY,&Finfo)) != 0){
        wprintf("No abble to find first file");
        exit (0);
}
NFMax=first_data_file+analyze_size ;
for(i=1;i<=NFMax;i++){
        if(i<first_data_file){
                if(( Status = _dos_findnext(&Finfo))!=0)exit(0);
                continue;
        }
        sprintf(FBuff,"%s%s",CFD,Finfo.name);
        if(kbhit()) break;
        if((F = fopen(FBuff,"rb")) == NULL){
                wprintf("Unable to open %s",FBuff);
                exit(1);
        }
        S = (sct_t *)malloc(sizeof(sct_t));
        if(!S){
                wprintf("Memory allocation error !!\n");
                exit (1);
        }
        fread(S,sizeof(sct_t),1,F);
        if(S->Len != sizeof(sct_t)){  /* check SCT from file */
        wprintf("The SCT is an unexpected size\n");
                exit(0);
        }
        stat (FBuff,&Dstat);
        /***plot  sounding time ***/
        _settextcolor(0);
        _settextwindow(1,1,30,80);
```

```
if(i==first_data_file){
     _setviewport(0,0,639,479);
     _grtextxy(70,385,ctime(&Dstat.st_atime));
     _settextposition(25,8);
     _outtext(ctime(&Dstat.st_atime));
}
if((i-first_data_file)==analyze_size){
      _setviewport(0,0,639,479);
      _grtextxy(380,385,ctime(&Dstat.st_atime));
}
_settextposition(25,50);
_outtext(ctime(&Dstat.st_atime));
_setviewport(2,398,637,477);
_setwindow(1,0.0,0.0,WINDOW_W,WINDOW_H);
_setcolor(11);
_rectangle_w(_GBORDER,0.0,0.0,WINDOW_W,WINDOW_H);
for(j=0;j<adapsize;j++){
     adapw[j]=1.0/(double)adapsize;
     adapx[j]=adap_init;
}
adapy=adap_init;
mean_range=220.;
old_range=210.;
Pos=S->Len;
PulseSet = 0;
NotDone = 1;
F2ragen=0;
_setviewport(10,20,600,380);
_setwindow(1,0.0,0.0,WINDOW_W,WINDOW_H);
while(NotDone){
     fsetpos(F,&Pos);
     size=fread(&PCT,size_of_pct,1,F);
     if(size<=0){
          clearerr(F);
          NotDone = 0;
     }
     if((int)((&PCT) ->SH.NmEs)<tom_number)
     plot_one_set(i-first_data_file,Pos,&PCT,F);
     Pos += PCT.Len;
     PulseSet++;
     if((int)(PCT.Freq/10000)>1000)F2ragen=1;
}
fclose(F);
free((void *)S);
if(( Status = _dos_findnext(&Finfo))!=0)break;
_settextwindow(26,2,30,80);
_settextposition(3,51);
sprintf(ch,"   %d",i);
_outtext(ch);
if(whichplot==4){
```

```
if(( cch =getch())=='p')  {  /*print_handler(); */
err = _open_drw( "browsl.plt" );
   if( err )
   {
      _setvideomode( _DEFAULTMODE );
      printf("\n _open_drw() error=%s", gperrmsg(err));
      exit(1);
   }
   err = _save_drw();
   if( err )
   {
      _setvideomode( _DEFAULTMODE );
      printf("\n _save_drw() error=%s", gperrmsg(err));
      exit(1);
   }
   err = _close_drw();
   if( err )
   {
      _setvideomode( _DEFAULTMODE );
      printf("\n _close_drw() error=%s", gperrmsg(err));
      exit(1);
}
     }
         _clearscreen(_GVIEWPORT);
         pulse_window();
         cch=getch();
     }
         if(   cch ==27)break;
 /*  rewind(image);  */
}

     if(whichplot!=4){
       /* if(( cch =getch())=='p')*/ {
err = _open_drw( "browsl.plt" );
   if( err )
   {
      _setvideomode( _DEFAULTMODE );
      printf("\n _open_drw() error=%s", gperrmsg(err));
      exit(1);
   }
   err = _save_drw();
   if( err )
   {
      _setvideomode( _DEFAULTMODE );
      printf("\n _save_drw() error=%s", gperrmsg(err));
      exit(1);
   }
   err = _close_drw();
   if( err )
   {
      _setvideomode( _DEFAULTMODE );
```

```
            printf("\n _close_drw() error=%s", gperrmsg(err));
            exit(1);
        }
          }
          }
          _setvideomode(_DEFAULTMODE) ;
}
/********    subroutines     *****************/
void plot_one_set(int file_index ,fpos_t Pos,pct_t *P,FILE *F)
{
     word  j,number_of_echos;
     echo_t E;
     double amp2,echo_Po,Pol[21],echo_amp[21],echo_range[21];
     int plot_color,frequency,color_band ,sizeofechos ;
     int i,echo_in_window=0,plot;

     double    sum_range=0,xx1,xx2,yy1,yy2;
     frequency=(int)((P->Freq)/10000);
     if(f_band[frequency]!=1)return;
     color_band=color[(15-frequency)%16];
     _setcolor(color_band);
     number_of_echos= P->SH.NmEs;
     if(number_of_echos>=20)number_of_echos=20;
     Pos += size_of_pct;
     fsetpos(F,&Pos);
     sizeofechos=size_of_echos ;
     delta_range=100.0;
     echo_in_window=0;
     sum_range=0.0;
     for(j=0;j<number_of_echos;j++){
          fread(&E,sizeofechos,1,F);
          arange(E,&echo_Po,&amp2);
          echo_amp[j]=amp2;
          Pol[j]=echo_Po;
          echo_range[j]=1.5*(double)E.RgBn;
     }
     switch(whichplot){
     case 0:
               xx1=(double)(file_index);

               for(j=0;j<number_of_echos;j++){
               yy1=echo_range[j];
               _setpixel_w(xx1,yy1);
          }
          break;
     case 1: /* for fig611a,b.plt  */
               xx1=6.0*(double)(file_index);
          if (((P->Freq)>=60000)&&((P->Freq)<=60200)) plot_color=0;
          if (((P->Freq)>=65000)&&((P->Freq)<=65200)) plot_color=7;
          if (((P->Freq)>=70000)&&((P->Freq)<=70200))plot_color=13;
          if (((P->Freq)>=75000)&&((P->Freq)<=75200)) plot_color=1;
```

```
      if (((P->Freq)>=80000)&&((P->Freq)<=80200)) plot_color=0;
      if (((P->Freq)>=85000)&&((P->Freq)<=85200)) plot_color=4;
      _setcolor(plot_color);
      if((((P->Freq)>=60000)&&((P->Freq)<=60200))
      || (((P->Freq)>=65000)&&((P->Freq)<=65200))
      || (((P->Freq)>=70000)&&((P->Freq)<=70200))
      || (((P->Freq)>=75000)&&((P->Freq)<=75200))
      || (((P->Freq)>=80000)&&((P->Freq)<=80200))
      || (((P->Freq)>=85000)&&((P->Freq)<=85200)))
      for(j=0;j<number_of_echos;j++){
         if(o_trace==1)  if(((Pol[j])> (pi*0.25))&&((Pol[j])
            <(pi*0.75)))  /* o-mode only*/
           {
           yy1=echo_range[j];
           if((yy1>200.0)&&(yy1<300.0)){
           yy1=(yy1-200.0)*8.0;
           _setpixel_w(xx1,yy1); }
             }

          if(x_trace==1)if(((Pol[j])<(-pi*0.25))
                          &&((Pol[j])>(-pi*0.75)))
           {
           yy1=echo_range[j];
           if((yy1>200.0)&&(yy1<300.0)){
           yy1=(yy1-200.0)*8.0;
           _setpixel_w(xx1,yy1); }
             }


      }
      break;
case 2:  /* for fig69a.plt and fig69b.plt*/
           xx1=6.0*(double)(file_index);
      if (((P->Freq)>=21000)&&((P->Freq)<=21200)) plot_color=7;
      if (((P->Freq)>=23000)&&((P->Freq)<=23200))plot_color=13;
      if (((P->Freq)>=32000)&&((P->Freq)<=32200)) plot_color=0;
      if (((P->Freq)>=35000)&&((P->Freq)<=35200)) plot_color=4;
      if (((P->Freq)>=29000)&&((P->Freq)<=29200)) plot_color=1;
      if (((P->Freq)>=19000)&&((P->Freq)<=19200)) plot_color=0;
      _setcolor(plot_color);
      if((((P->Freq)>=21000)&&((P->Freq)<=21200))
      || (((P->Freq)>=23000)&&((P->Freq)<=23200))
      || (((P->Freq)>=32000)&&((P->Freq)<=32200))
      || (((P->Freq)>=35000)&&((P->Freq)<=35200))
      || (((P->Freq)>=29000)&&((P->Freq)<=29200))
      || (((P->Freq)>=19000)&&((P->Freq)<=19200)))
      for(j=0;j<number_of_echos;j++){
         if(o_trace==1)  if(((Pol[j])> (pi*0.25))&&((Pol[j])

           <(pi*0.75)))  /* o-mode only*/
           {
```

```
                yy1=echo_range[j];
                if((yy1>250.0)&&(yy1<450.0)){
                yy1=(yy1-250.0)*4.0;
                _setpixel_w(xx1,yy1); }
                  }

                    i f ( x _ t r a c e = = 1 )
if(((Pol[j])<(-pi*0.25))&&((Pol[j])>(-pi*0.75)))
                {
                yy1=echo_range[j];
                if((yy1>250.0)&&(yy1<450.0)){
                yy1=(yy1-250.0)*4.0;
                _setpixel_w(xx1,yy1); }
                  }
          }
          break;
     case 3:
          xx1=(double)(file_index*600/analyze_size);
          if((((P->Freq)>=10000)&&((P->Freq)<=10200))
          || (((P->Freq)>=20000)&&((P->Freq)<=20200))
          || (((P->Freq)>=30000)&&((P->Freq)<=30200))
          || (((P->Freq)>=40000)&&((P->Freq)<=40200))
          || (((P->Freq)>=50000)&&((P->Freq)<=50200))
          || (((P->Freq)>=60000)&&((P->Freq)<=60200)))
          for(j=0;j<number_of_echos;j++){
               if(((Pol[j])> (pi*0.25))&&((Pol[j]) <(pi*0.75)))
                {
               yy1=echo_range[j];
               _setpixel_w(xx1,yy1);
                }
          }
          break;
     case 4:
          if(!F2ragen) for(j=0;j<number_of_echos;j++){
               plot_color=8;
               plot=1;
               if(number_of_echos>=5){
                    plot=tom_alg(echo_amp,number_of_echos);
                    if(plot==0) plot_color=11;
                    else plot_color=9;
               }
                if(plot&&(adapy-range_window<echo_range[j])
                    &&(echo_range[j]<adapy+range_window))

               {
                     if(multi_trace==1){
                       sum_range+=echo_range[j];
                         echo_in_window++;       }
                    plot_color=14;
               if(((Pol[j])> (pi*0.25))&&((Pol[j]) <(pi*0.75)))
                    {           /* adaptive ordinary only*/
```

```
                        plot_color=1;
                    if(o_trace==1){     sum_range+=echo_range[j];
                        echo_in_window++;       }
                    }

if(((Pol[j])<(-pi*0.25))&&((Pol[j])>(-pi*0.75)))
                    {               /* X-mode  */
                        plot_color=7;
                     if(x_trace==1){
                                    sum_range+=echo_range[j];
                        echo_in_window++;     }
                    }
                }
                _setcolor(plot_color);
                xx1=(double)(P->Freq/250);
                yy1=echo_range[j];
                if((plot_color==1)||(plot_color==7))
                _rectangle_w(_GFILLINTERIOR,xx1,yy1+1.,xx1+1., yy1);
                 _setpixel_w(xx1,yy1);
                 if(plot_y==1){       _setcolor(13);
                 _setpixel_w(xx1,adapy);    }
                if((j==number_of_echos-1)&&(echo_in_window!=0)){
                     mean_range=sum_range/(double)(echo_in_window);
                     for(i=adapsize-1;i>0;i--)adapx[i]=adapx[i-1];
                     adapx[0]=mean_range;
                      if(MLMS==1)for(i=0;i<adapsize;i++)adapw[i]+=
                      adapmu*(double)frequency*adape*adapx[i];
                      if(LMS==1)for(i=0;i<adapsize;i++)adapw[i]+=
                      adapmu*adape*adapx[i];
                      adapy=0.0;

for(i=0;i<adapsize;i++)adapy+=adapw[i]*adapx[i];
                     adape=mean_range-adapy;
                }
           }
           else if(F2ragen) for(j=0;j<number_of_echos;j++){
                plot=1;
                plot_color=4;
                if(number_of_echos>=5){
                     plot=tom_alg(echo_amp,number_of_echos);
                     if(plot==0) plot_color=11;
                     else plot_color=9;
                }

                if(plot&&(delta_range>(echo_range[j]-old_range))&&
                    (old_range<echo_range[j])){
                     delta_range=echo_range[j]-old_range ;
                     mean_range=echo_range[j];
                     plot_color=14;
                     if(((Pol[j])> (pi*0.25))&&((Pol[j])
                     <(pi*0.75)))plot_color=1;
```

```
                    if(((Pol[j])<(-pi*0.25))&&
                    ((Pol[j])>(-pi*0.75))) plot_color=7;
               }
               _setcolor(plot_color);
               xx1=(double)(P->Freq/250);
               yy1=echo_range[j];
               _setpixel_w(xx1,yy1);
               if(j==number_of_echos-1)old_range=mean_range;
          }
          break;
     }
}
void dialog(void)
{
     void screen_1pulse(void),screen_4pulse(void);
     char ch;
     int i;
     _setviewport(2,398,637,477);
     _clearscreen(_GVIEWPORT);
     _setwindow(1,0.0,0.0,WINDOW_W,WINDOW_H);
     _rectangle_w(_GBORDER,0.0,0.0,WINDOW_W,WINDOW_H);
          _settextcolor(0);
          _setviewport(0,0,639,479);
          _grtextxy(100,410, fig1);
          _grtextxy(100,430, fig2);
          _grtextxy(100,450, fig3);
     _settextwindow(26,2,30,80);
     _settextposition(1,1);
     _outtext("Plot mode (0,1,2,3,4)? ");
     scanf("%d",&whichplot );
     n_pulse=1;
     if(n_pulse < 4) screen_1pulse();
     else screen_4pulse();
     _settextwindow(26,2,30,80);
     _settextposition(2,1);
     _outtext("                                                  ");
     _settextposition(2,1);
     _outtext("Frequency Bands 1MHZ-16MHZ (1...f or g(global))?");
     while((int)(ch=getche())!=13){
          switch(ch){
          case '0':
               f_band[0]=1;
               break;
          case '1':
               f_band[1]=1;
               break;
          case '2':
               f_band[2]=1;
               break;
          case '3':
               f_band[3]=1;
```

```
                    break;
               case '4':
                    f_band[4]=1;
                    break;
               case '5':
                    f_band[5]=1;
                    break;
               case '6':
                    f_band[6]=1;
                    break;
               case '7':
                    f_band[7]=1;
                    break;
               case '8':
                    f_band[8]=1;
                    break;
               case '9':
                    f_band[9]=1;
                    break;
               case 'a':
                    f_band[10]=1;
                    break;
               case 'b':
                    f_band[11]=1;
                    break;
               case 'c':
                    f_band[12]=1;
                    break;
               case 'd':
                    f_band[13]=1;
                    break;
               case 'e':
                    f_band[14]=1;
                    break;
               case 'f':
                    f_band[15]=1;
                    break;
               case 'g':
                    for(i=0;i<16;i++)f_band[i]=1;
                    break;

                    break;
               }

          }
          _settextposition(3,1);
          _outtext("
");
          _settextposition(4,1);
          _outtext("
");
```

```
        _settextposition(3,1);
        _outtext("Data file # you want to start with(1,2,3...)? ");
        scanf("%d",&first_data_file);
        _settextposition(4,1);
        _outtext("predict window size(1...100)?");
        scanf("%d",&i);
        range_window=(double)i;
        _settextposition(4,35 );
        _outtext("<Press:p(plot),ESC(stop),others(restart)>");

}
```

PRECEDING PAGE BLANK NOT FILMED

REFERENCES

Beer, Tom, "Atmospheric Waves", John wiley & sons, New York, 1974.

Chimonas, G.and Hines, Atmospheric gravity waves launched by auroal current, Planet. space Sci., 18, 565, 1970.

Cowan, C.F.N., Performance Comparision of Finite Linear Adaptive Filter, IEE Proceedings Part F, Vol. 134(3), pp.211-216, June, 1987.

DeFatta, D. J., Lucas J. G., and Hodgkiss W.S. "Digital signal processing: A system design Approach, John Wiley & sons, New York, 1988.

Dudeney, J.R. and Kressman R.I., Empirical models of the electron concentration of the ionosphere and their value for radio communications purposes, Radio Science, Vo.21, No.3 pp319-330, May 1986.

Fox, M. W., Automatic ionogram scaling at IPS, Technical report, PS-TR-88-04, IPS Radio and Space Services,Australian, March 1988.

Francis, S.H., Global propagation of atmospheric gravity wave: A review, Journal of Atmospheric and Terrestrial Physics, Vol.37, pp.1011-1054, 1975.

Gossard, E. E. and Hooke W. H., " Wave in the atmosphere", Elsevier, New York, 1975.

Hines, C. O., Propagation velocities and speeds in ionospheric waves: A review, J. Atmos. Terr. Phys., 36, 1179, 1974.

Hines, C. O. et al. "The upper atmosphere in Motion", Geophys. Monogr. Ser., Vol. 18, AGU, Washington, D. C., 1974.

Hunsucker, R.D., Atmospheric Gravity Waves generated in the high-latitude ionosphere: a review, Reviews of Geophys. and Space Phys., Vol.20 No.2, pp. 293-315, May 1982.

Jarvis, M.J. and Dudeney J.R., Reduction of ambiguities in HF radar results through a revised receiving antenna array and sounding pattern, Radio Science, Vo.21, No.1, January. 1986.

Kamide, Y. and Brekke, Auroral electrojet current density deduced from the Chatanika radar and from Alaska meridian chain of magnetic observations, J. Geophys. Res., 80, 587,1975.

Kelley, M.A., M.F.Larsen, C.LaHoz,and J.P.McCure, Gravity wave initiation of equatorial spread F: A case study, J.Geophys. Res., 86, 9087, 1981.

LaHoz, C., Larsen M.F., and Kelley M.C., Gravity Wave initiation of equatorial spread F: A case study, journal of Geophysical Research, Vol.86, No.A11, October 1, 1981.

Liu, J. Y., A study of the relationship between ionospheric oscillation and geomagnetic pulsations, Ph.D dissertation, Utah State University, 1990.

Liu, C.H. and Yeh K. C., Acoustic gravity wave in the upper atmosphere, Rev. Geophys.Space Phys., 12(2), 193, 1974.

Marple, S. L., Frequency resolution of Fourier and maximum entropy spectral estimates, Geophysics, 47(9), 1303-1307, 1982.

Marple, S. L., A new autoregressive spectrum analysis algorithm, IEEE Trans. Acoust. Speech Signal Proc. ASSP-28(4), 441-454, 1980.

Meehan, D.H., Modelling of complicated ionograms using several echoes, Journal of Atmospheric and Terrestrial Physics, Vol.52, No.9, pp.771-779, 1990.

Mulgrew, B. and Cowan C.F.N., "Adaptive filter and equalizers", Kluwer Academic Publishers,

Boston, 1988.

Piggott, W.R., Rawer K.,World data center A for solar-terrestrial physics, U.R.S.I. handbook of ionogram interpretation and reduction, July, 1978.

Paul, E. A., Research instrument combines custom A/D, commercial PSD boards, Personal Engineering & Instrumentation News, pp.49-57, August 1987.

Paul, A.K., Fregion tilts and ionogram analysis, Radio Science, Vo.20, No.4 pp959-971, July 1985.

Reinisch, B. W., New techniques in ground-based ionospheric sounding and stadies, Radio Science,Vl. 21, No.3, pp.331-341, 1986.

Reinisch, B.W. and X. Huang, Automatic calculation of electronic density profiles from digital ionograms 3. Processing of bottom side ionograms, Radio Science, Vo.18, No.3 p477-492, May 1983.

Richmond, A. D., Gravity Wave Generation, Propagation,and Dissipation in the Thermosphere, J. Geophys. Res., 83, 4131-4145, 1978.

Stiles, G.S., Berkey F.T., and Doupnik J.R.,Digital ionosonde studies of the ionosphere from Siple Station and Roberval, Quebec, Antarctic Journal, pp.213-215, 1981.

Testud, J., Interaction between gravity waves and ionization in the ionosphere F region, Space Res., XII, 1163, 1972.

Treichler, J. R., Johnson C.R. and Larimore M. G., " Theory and design of adaptive filters", John Wiley & Sons, New York, 1987.

Walker, G. O., Wong Y. W., etc., Propagating ionospheric waves observed throughout east Asia during the WAGS October 1985 campaign, Radio Science, Vo.23, No.6, pp.867-878,

Nov. 1988.

Widrow, B. and Stearns S.D., "Adaptive signal processing", Printice-Hall,Inc. USA, 1985.

William, H. Press, Brian P. Flnnery,Saui A. Teukolsky,William T.Vetterling, Numerical recipes in C, The art of scientific computing, Cambridge University Press, Cambridge, New York, 1990.

Wright, J.W., Laird A.R., etc., Automatic N(h,t) profile of the ionosphere with a digital ionosonde, Radio Science, Vo.7, No.11, pp.1033-1043, Nov. 1972.