Development of a Meteor Radar Wind Measuring Facility

A thesis
submitted in partial fulfilment
of the requirements for the Degree
of
Doctor of Philosophy in Physics
in the
University of Canterbury

by
Steven H Marsh

University of Canterbury
2000

To my family.

Abstract

This thesis details the development, construction and operation of a Doppler wind facility as an upgrade of the existing meteor radar, AMOR operated by the Department of Physics and Astronomy, Canterbury University, New Zealand. It also describes the initial interpretation of wind-field measurements so as to ascertain the potential for a more sustained survey.

An overview of atmospheric dynamics is presented in order to provide a summary of wind motions accessible to meteor radar techniques. Tidal analysis methods applied to the data confirm a dominant semidiurnal tide with seasonally varying amplitude. The height resolution of the radar enables analysis of vertical structure, i.e., the semidiurnal tide's vertical wavelength.

Echo analysis techniques which enable wind measurements with uncertainties $< 3\ \mathrm{m\,s^{-1}}$ to be achieved from meteor echoes having duration times down to only 0.03 s are discussed. The method allows a line of sight wind measurement to be made from 90% of echoes. A transmitted beam which is narrow in azimuth combined with a dual interferometer and range determination locates the echo point within a $8\ \mathrm{km^3}$ 3 dB box. Approximately 70% of the line of site wind measurements produce a spatially located horizontal wind speed.

A method of reducing wind speed measurement errors introduced by the magnetic field is given which relies on both meridional and zonal components of the wind-field being measured.

The AMOR winds data are compared with other wind measuring instruments that are geographically close and the results are discussed. Comparisons are also made with global model data. Analysis of the wind speeds as a function of ground range from the radar gives good evidence supporting the presence of gravity wave activity at meteor detection heights. A seasonal gravity wave dependence is also suspected.

Contents

Figures

Chapter 1

Introduction

The Earth's atmosphere is conveniently divided into height intervals based on the vertical structure of the temperature profile (figure 1.1). These intervals called the troposphere, stratosphere, mesosphere and thermosphere are separated by the tropopause, stratopause etc. The term middle atmosphere is used to describe the region between the tropopause and homopause (the atmosphere between vertical heights of about 10 and 110 km) and the term upper atmosphere often refers to the region above the homopause.

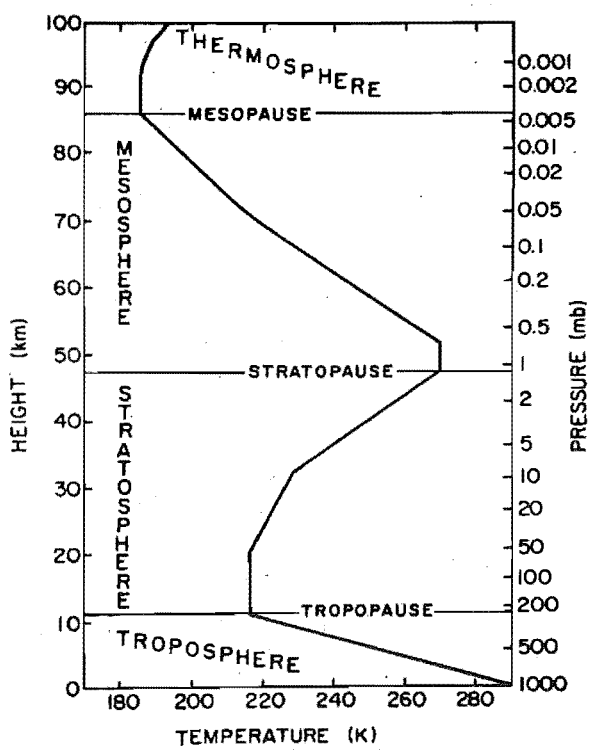

Figure 1.1: The atmosphere is divided up into layers based on the vertical temperature structure. (After Andrews et al., 1987)

As far back as Lord Kelvin, in the mid 19th century, physicists have tried to gain a greater understanding of the dynamics that govern the atmosphere's motion. More recently remarkable advances have been achieved in atmospheric physics primarily due to the greater general awareness prompted by phenomena such as global warming and the depletion of Antarctic ozone. Many methods for measuring the atmosphere's movement, or winds, have been developed ranging in complexity from simple anemometers for wind measurements at ground level, to instruments such as HRDI (High Resolution Doppler Interferometer) on the Upper Atmosphere Research Satellite (UARS). Winds measurements at heights ranging from the ground to the top of the troposphere have been made with balloons and in more recent times Very High Frequency (VHF) radars. Some of the higher power VHF radars are capable of probing the atmosphere to mesospheric heights and are called MST (Mesosphere Stratosphere Troposphere) radars. The Doppler shifts of night time airglow measured by Fabry-Perot Interferometers (FPI) are used to obtain winds at two distinct heights of approximately 90 and 100 km. Medium Frequency (MF) radars infer winds speeds at heights from 70–100 km from the ground level motion of diffraction patterns caused by partial reflections from upper middle atmosphere ionisation. A further technique, one which is the primary focus of this thesis, uses meteor radars and measures wind speed from the change in frequency of radio waves which have reflected back from meteor trains at altitudes between about 80 and 120 km.

There are many meteor radars measuring atmospheric winds at various locations in the world today. Some southern hemisphere meteor radars are situated at Grahamstown, South Africa; Adelaide, Australia (35°S 138°E); Jakarta, Indonesia and Amundsden-Scott Station South Pole. The South African meteor radar [Poole 1988] (33°S 26°E) operates at a frequency of near 28 MHz with a pulse repetition frequency (prf) of 500 Hz. The main lobe of the transmitted beam is at an elevation angle of 45° and radiation is transmitted in all azimuths. The system does not measure meteor detection heights; instead it assumes all echoes to have come from a height of 95 km. The Jakarta meteor radar [Tsuda 1995] is located at coordinates 6°S 107°E illuminating regions of atmosphere between 5°S 107°E and 7°S 107°E. Echo height is determined with an interferometer and approximately 800 meteors are detected each day. The meteor radar at the South Pole [Forbes et al. 1999] has been in operation since 1995 making wind measurements near 2° from the South Pole along the four meridians 0 degrees, 90 degrees E, 90 degrees W and 180 degrees.

As with the South African radar those heights are assumed to be at 95 km. Northern hemisphere radars include sites in Trivandrum, Durham in the USA, Sheffield in England, Christmas Island, the CLOVAR radar system in London Canada and a recently modified MU radar in Shigaraki, Japan. The system in Trivandrum (8.5°N, 77°E) operates at a frequency of 54.95 MHz generating a 280 μs pulse with a prf of 500 s^{-1} [Raghava Reddi et al. 1993]. The elevation angle of received echoes is measured allowing height determination for individual echoes. Wind measurements can be made in both the zonal (east-west) and meridional (north-south) directions. The Durham and Sheffield radar systems are both computer controlled coherent pulsed systems; the Durham radar [Salah et al. 1997] is at coordinates 43°N 71°W and the Sheffield radar [Mitchell et al. 1996] is at 52.5°N 2°W. For the Sheffield system echo heights are again assumed to be at 95 km whereas the Durham radar determines echo location with an interferometer. The meteor radar at Christmas Island [Avery et al. 1990] operates at a frequency of 50 MHz at location 2°N 158°W. Finally the CLOVAR (Canadian *at* London OntarioVHF atmospheric radar) is an instrument which has been in operation since 1993 at location coordinates 43°N 81°W. The system operates at a frequency of nearly 41 MHz and is operated by the University of Western Ontario in London, Canada [Hocking 1997]. The Middle and Upper atmosphere (MU) radar at coordinates 34.9°N 136.1°E in Shigaraki, Japan is a VHF MST radar recently modified to measure winds from the drift of meteor trains. This system generates 1 μs pulses with an interpulse period 400 μs (a PRF of 2500 s^{-1}). Interferometry techniques enable the measurement of echo arrival angles and sampling at 1 million samples/s enables a range resolution of 150 m [Nakamura et al. 1997].

There are a few other meteor radars operating around the world today and the list presented is not intended to be complete. It does however indicate that there are a large number of varied meteor radar systems. There are nevertheless subtle differences in the techniques adopted and each has its own advantages.

1.1 Thesis Layout

This thesis discusses the development of a meridional wind speed measuring component, at meteoric heights as an addition to the existing Advanced Meteor Orbit Radar (AMOR) meteor orbit radar at Birdlings Flat (172° 39´E, 43° 34´S) near

Christchurch in the South Island of New Zealand. The system uses a dedicated computer with specialised software to capture data from the radar's receivers during meteor events. Additional hardware between receivers and the new dedicated winds computer was required to transform the signal from the receivers into in-phase and phase-quadrature components that can be analysed to produce wind speeds. Wind speed is then inferred from the Doppler shift of the transmitted pulse having been reflected from the meteor train and the measurement position is determined from the elevation angle and range of the returned pulse (refer figure 1.2).

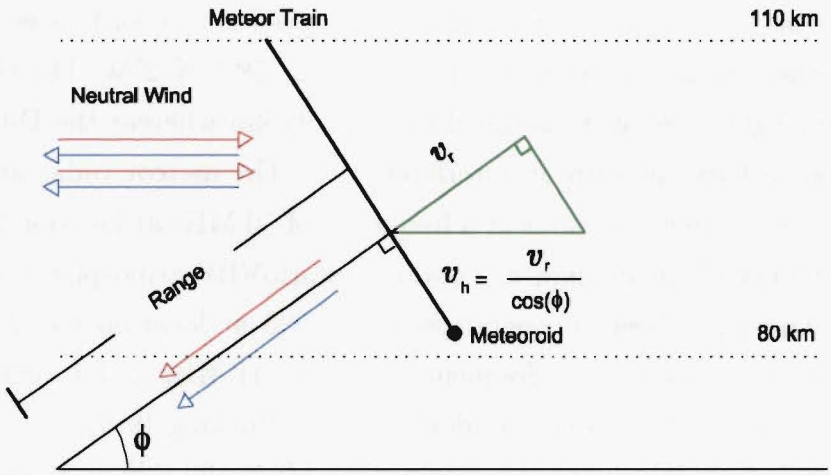

Figure 1.2: A meteoroid entering the Earth's atmosphere ionises the surrounding gases at heights between about 80 and 110 km. In the presence of wind this train of electrons moves at the wind's velocity and the radial wind speed (v_r) can be obtained from the Doppler shift of radiation reflected from the train. Assuming no vertical wind the horizontal wind speed can be simply derived. The location of the wind measurement is achieved from the range and elevation angle (ϕ).

The introductory chapter discusses areas of dynamics which are appropriate to, and whose features have been observed by the radar in, the atmosphere at meteoric heights (80–120 km). The formation of a meteor train, the atmospheric tracer probed by AMOR, is then considered and finally an overview of the AMOR system is given.

A description of additional hardware and software required to provide AMOR with wind speed measuring capability is given in detail in chapter 2. This chapter discusses the Doppler module; the unit which decomposes the AMOR signal into in-phase and phase-quadrature components, and the digital hardware which controls

the operation of accessing computer memory directly for the storage of meteor echoes. The approach to data analysis and the computer programmes which apply this analysis to derive wind speed measurements from meteor radar data are also explained in this chapter.

The effect of the Earth's magnetic field on the meteor train's drift is considered in chapter 3.

A discussion of data preparation for time series analysis and the effects of diurnal sampling are discussed in chapters 4 and 5 and results of harmonic analysis are presented in chapter 6. A comparison of these results with measurements made by other land-based instruments and model simulations is provided in chapter 7.

Finally chapter 8 discusses ideas for continued research with the winds component of the AMOR meteor radar.

The thesis has two appendices. The first provides a reference for the computer software which was developed during the AMOR winds project. These programmes have either been written in Borland Turbo Pascal (7.0) or standard ANSI C. A brief description is provided with each procedure or function to detail its purpose. The second appendix provides additional tables which were considered interesting yet would have interrupted the general flow of the thesis.

1.2 Introduction to Middle Atmosphere Dynamics

Measurements of the Earth's atmospheric motion, $U(u, v, w)$ (where u, v, w are components in the west-east, south-north, and vertical respectively) have shown it to be extremely complex, however Fourier analysis enables this motion to be approximated by a mean value and the expansion of a series of Fourier harmonics:

$$U = A_0(\phi, z) + \sum A_m(\phi, z) \cos\left[m^{-1}\lambda + \alpha_m(\phi, z)\right], \tag{1.1}$$

where $A_0(\phi, z)$ is the mean value at a particular latitude (ϕ) and height (z) and $A_m(\phi, z)$ is the amplitude of the mth harmonic where $\alpha_m(\phi, z)$ is its phase. Harmonics with periods greater than one day are called Planetary[1] waves [Salby 1996] whereas harmonics of the 24 hour oscillation with periods less than or equal to one day are referred to as atmospheric tides.

[1]Westward propagating planetary waves are generally called Rossby waves.

Atmospheric movements which cannot be attributed to tides, planetary waves or mean wind are likely to be due to gravity waves which have propagated into the meteor zone (80–110 km) depositing their momentum and creating turbulence (essentially the quasi-random movement of air particles).

1.2.1 Atmospheric Tides

An atmospheric tide is defined to be a world–wide pressure, density, temperature or wind velocity oscillation with a period of m^{-1} of a solar or lunar day (m = 1,2,3, ...). These oscillations are due to the gravitational and thermal forcing of both the Moon and Sun as the Earth rotates on its axis. Atmospheric forcing has been shown to be dominated by periodicities associated with the absorption of solar radiation by water vapor (H_2O) and ozone (O_3) in the lower stratosphere [Chapman & Lindzen 1970]. This is in contrast to oceanic tides which are a gravitational effect dominated by the lunar cycle. The O_3 and H_2O heating rates vary with season and latitude but an average height for the O_3 heating maximum is at 50 km with a FWHM of 20 km. The heating rate for H_2O is maximum near ground level and decreases with height.

Solar heating of the atmosphere can be approximated by the function shown in figure 1.3 which has a width of 12 hours centered at noon. Fourier analysis of this function gives a mean value, 24 hour and 12 hour components called in terms of atmospheric motion the mean wind, diurnal tide and semidiurnal tide. Other components are also present but they have much smaller amplitudes.

As the Earth rotates on its axis the region of maximum heating moves westward. Tides produced by this mechanism are called migrating tides.

There are several modes of atmospheric oscillation which can occur on the surface of the spherical Earth. These modes are denoted by the parameters (n,m). The zonal wavenumber, n, is the number of cycles per day i.e., n = 2 for the semidiurnal tide. The difference m − n (m $\geqslant$ n) gives the number of nodes between the Earth's poles (excluding the poles themselves). As an example the (2,6) mode would be a semidiurnal tide with 4 nodes between the north and south poles and graphs showing spherical harmonics for n=5 and m=0, 1, 2, 3, 4, 5 are given in figure 1.4. The modes which have a maximum at the equator are predominant since solar heating is greatest in the equatorial region.

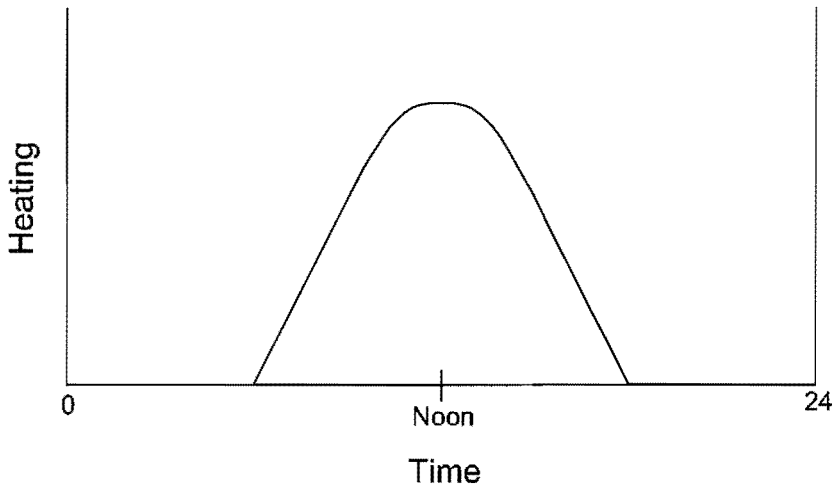

Figure 1.3: The solar heating can be approximated by such a function. (After Andrews et al., 1987)

Assuming that the earth is spherical, rotating at a constant angular velocity Ω and the acceleration due to gravity g is constant independent of altitude and latitude, the atmosphere can be described by the distribution of static pressure p_0, density ρ_0 and temperature T_0. With these assumptions Laplace [1799] shows tides to be solutions to Laplace's equation

$$L\hat{\Phi}_n + \gamma\hat{\Phi}_n = 0, \tag{1.2}$$

where L is a second order ordinary differential operator in $\mu \equiv \sin\phi$, dependent on zonal wavenumber s and σ (where the period is $2\sigma^{-1}$) taking the form

$$L = \frac{d}{d\mu}\left[\frac{1-\mu^2}{(\sigma^2-\mu^2)}\frac{d}{d\mu}\right] - \frac{1}{\sigma^2-\mu^2}\left[\frac{-s\left(\sigma^2+\mu^2\right)}{\sigma\left(\sigma^2-\mu^2\right)} + \frac{s^2}{1-\mu^2}\right], \tag{1.3}$$

and γ is Lamb's parameter

$$\gamma \equiv \frac{4\Omega^2 a^2}{gh}. \tag{1.4}$$

In this expression a is the Earth's radius and h is the atmosphere's equivalent depth.

The physically imposed condition that $\hat{\Phi}$ be bounded at the poles $\mu = \pm 1$ allows this eigenvalue problem to be solved numerically. Specifying s and σ as for the theory for thermally forced tides produces the eigenvalue solutions γ (or

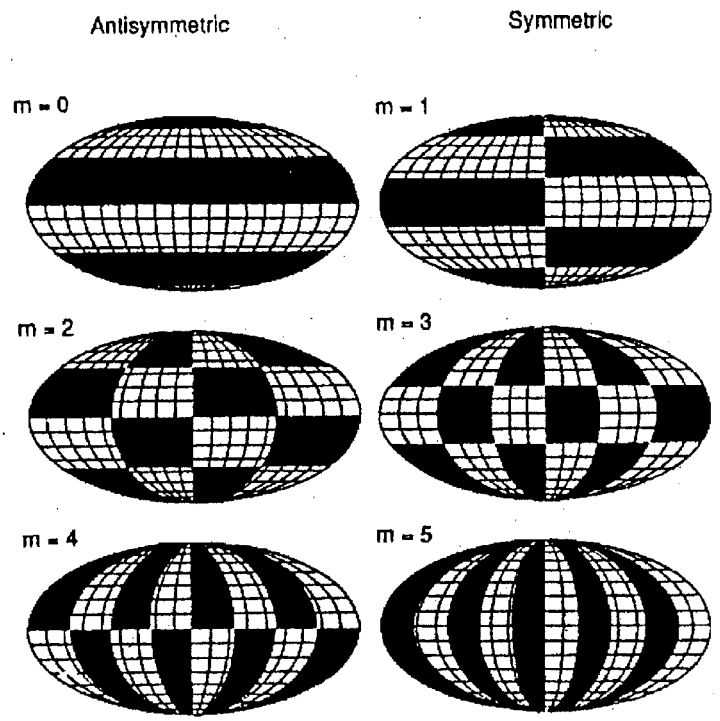

Figure 1.4: The patterns of positive and negative regions for the spherical harmonic functions with n=5 and m=0,1,2,3,4,5.

mean depths h) along with their corresponding eigenfunctions $\Theta_n^{(\sigma,s)}$ called Hough Functions. Properties of these Hough functions have been extensively documented [Longuet-Higgins 1968].

Some tidal modes propagate vertically and this can be observed by measuring a change of phase with height. The vertical wavelength of these tidal modes is the vertical distance through which the phase of the oscillation changes by 2π.

From the basic properties of the vertical structure equation derived in Andrews et al. [1987]

$$\frac{d^2W}{dz^2} + \left[\frac{N^2}{gh} - \frac{1}{4H^2}\right] W = \text{Forcing Terms}, \tag{1.5}$$

it can be seen that when $0 < h < 4N^2H^2/g$ a sinusoidal solution exists and Andrews et al. [1987]shows the wavelength of this sinusoid and hence vertical wavelength (λ_v) is given locally by

$$\lambda_v = 2\pi \left[\frac{N^2(z)}{gh} - \frac{1}{4H^2}\right]^{-\frac{1}{2}}, \tag{1.6}$$

where h is the equivalent depth parameter, N a buoyancy frequency and H is

the scale height. If h is negative (or $> 4N^2H^2/g$) then the wave is evanescent (vertically trapped) and the oscillation can only exist near the forcing region (lower stratosphere). The diurnal tide's vertically propagating modes are all evanescent poleward of 30° [Longuet-Higgins 1968, Chapman & Lindzen 1970] and hence should not be observed at meteoric heights over New Zealand. The equivalent depth h is positive for the semidiurnal tide and the (lower stratosphere) dominant mode (2,2) has $h \approx 7.85$ km. For this value $N^2/gh \approx 1/4H^2$ and large vertical wavelengths exist. At meteoric heights (80–110 km) the dominant mode of oscillation has been shown to be (2,4) with a vertical wavelength of ≈ 40 km [Lindzen & Hong 1974]. These higher modes tend to dominate due to the presence of vertically varying background winds and meridional temperature gradients causing mode coupling of the lower modes (2,2) and (2,3).

In general a number of modes, each with a different vertical wavelength, may exist and, as a consequence, the tide's vertical phase profile may not display a constant change with height and appear rather complicated. Reflection characteristics in the region, due to temperature or density changes, may cause a mode to propagate downwards further altering the apparent model content [Fellous et al. 1974].

Studies by Forbes [1982] and Manson et al. [1989] have shown strong seasonal variations in both the amplitude and phase of the semidiurnal tide.

1.2.2 Planetary Waves

Planetary (including Rossby) waves are a group of zonally propagating atmospheric waves of a global scale and hence are the most important for large-scale meteorological processes. These waves have periods of more than one day and are unlike the thermally driven tides of the previous section as they do not appear to be maintained by any forcing effects [Andrews et al. 1987]. Holton [1992] says that the Rossby wave is an absolute vorticity conserving motion, owing its existence to the latitudinal variation in the Coriolis force.

Holton [1992]describes Rossby wave propagation by considering a closed chain of eastward moving fluid parcels which are initially aligned along a circle of constant latitude (ϕ_0).

Defining absolute vorticity η to be given by $\eta = \zeta + f$ where ζ is the relative vorticity and f is the planetary vorticity, it is apparent that if the chain of air parcels

(in the northern hemisphere) undergoes a sinusoidal meridional displacement then, for displacement towards the equator the air parcel experiences reduced planetary vorticity. Therefore, in order to conserve absolute vorticity, the air parcel spins up cyclonically (refer figure 1.5).

The northward motion induced ahead of the parcel then deflects its motion poleward, through its undisturbed latitude (ϕ_0). Once poleward of latitude ϕ_0, the parcel spins up anticyclonically and southward motion ahead of the parcel deflects the parcel back towards, and eventually through, latitude ϕ_0. Thus the air parcel cycles back and forth about its undisturbed latitude due to the variation in f, with latitude, exerting a torque on the displaced air. The direction that Rossby waves propagate can be deduced from figure 1.5. The southward motion behind the cyclonically spinning parcel displaces that section of the contour equatorward and thereby shifts the wave trough westward. Increased northward motion west of the anticyclonic parcel generates the same effect.

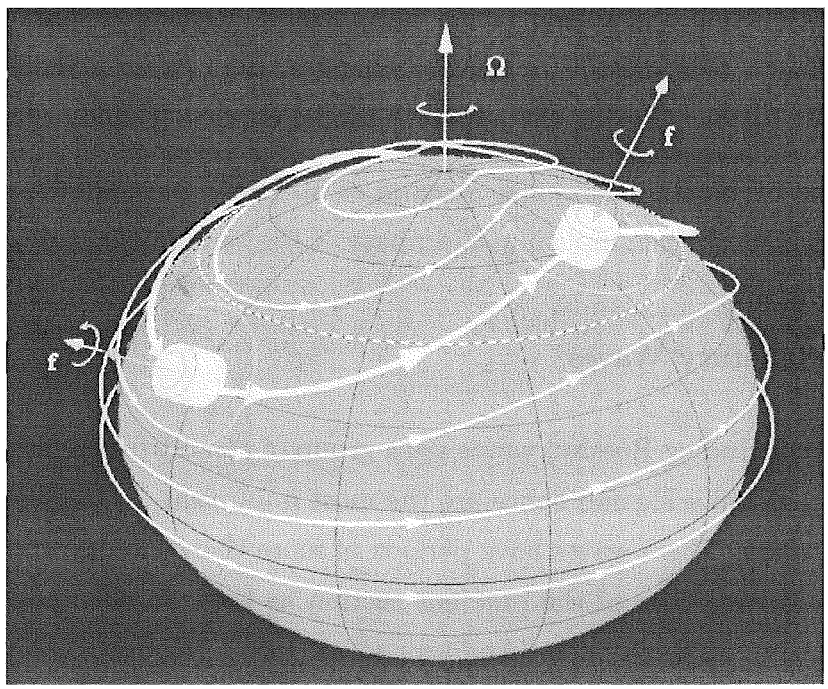

Figure 1.5: The reaction of an air parcel to meridional displacement. Displaced equatorward an eastward moving air parcel spins up cyclonically to conserve absolute vorticity. Northward motion ahead of the parcel then deflects the parcel polewardback toward and then through its undisturbed latitude (dashed line). The reverse process then occurs. (After Salby [1996])

Planetary waves which are observed by the AMOR winds system and discussed later in this thesis include the 2-day, 4-day, 5-day and 10-day oscillations.

1.2.3 Geostrophic Winds

The reference frame in which atmospheric motion is observed (the Earth) is rotational. In such frames scalar quantities (pressure, temperature etc.) appear the same as they would in an inertial reference frame, however vector quantities do not. The vectors which describe an air parcel's motion must be corrected to account for the acceleration of the Earth's reference frame [Salby 1996].

By considering a reference frame rotating with an angular velocity Ω it becomes apparent that a vector A which is constant in that frame must rotate when viewed in an inertial frame (figure 1.6).

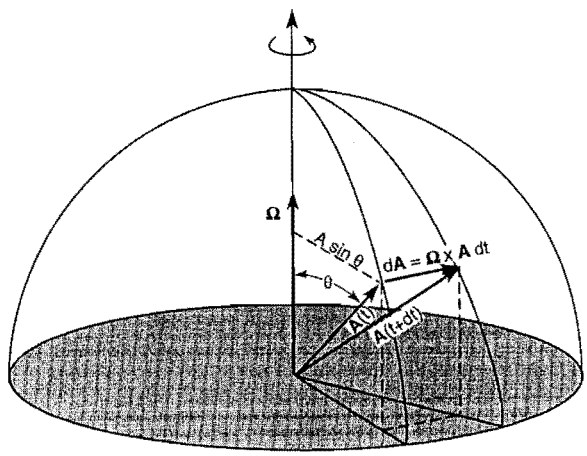

Figure 1.6: A vector A, which is fixed in a rotating reference frame, changes in an inertial reference frame. (After Salby, 1996)

During an interval dt, A will change by a vector increment dA, which is perpendicular to the plane of A and Ω and has a magnitude of

$$|dA| = A \sin\theta \cdot \Omega \, dt, \tag{1.7}$$

where θ denotes the angle between A and Ω. Therefore in an inertial reference frame, the vector A changes at a rate given by

$$\left|\frac{dA}{dt}\right| = A\Omega \sin\theta, \tag{1.8}$$

in a direction as described earlier. It is evident from this expression that the time rate of change of A apparent in an inertial reference frame is described by

$$\left(\frac{dA}{dt}\right)_i = \Omega \times A. \tag{1.9}$$

This expression can be expressed more generally to account for a time rate of change dA/dt in the rotating frame as

$$\left(\frac{dA}{dt}\right)_i = \frac{dA}{dt} + \Omega \times A, \tag{1.10}$$

in the inertial reference frame.

By considering the position r of an air parcel, from equation 1.10 it is apparent that the parcel's velocity $\mathbf{u} = dr\,/\,dt$ is given by

$$\mathbf{u}_i = \mathbf{u} + \Omega \times r, \tag{1.11}$$

and the acceleration apparent in the inertial frame is similarly given by

$$\left(\frac{d\mathbf{u}_i}{dt}\right)_i = \frac{d\mathbf{u}_i}{dt} + \Omega \times \mathbf{u}_i. \tag{1.12}$$

By incorporating the velocity apparent in the inertial frame equation 1.12 is

$$\begin{aligned}\left(\frac{d\mathbf{u}_i}{dt}\right)_i &= \left(\frac{d\mathbf{u}}{dt} + \Omega \times \mathbf{u}\right) + \Omega \times (\mathbf{u} + \Omega \times r)\,, \\ &= \frac{d\mathbf{u}}{dt} + 2\Omega \times \mathbf{u} + \Omega \times (\Omega \times r)\,.\end{aligned} \tag{1.13}$$

The expression given in equation 1.13 indicates that two correction terms for the acceleration of an air parcel arise as a result of the Earth's rotation. The term $2\Omega \times \mathbf{u}$ is called the Coriolis acceleration. This acceleration is perpindicular to both the motion of the air parcel and the Earth's angular velocity. The other correction term $\Omega \times (\Omega \times r)$ is called the centrifugal acceleration and is generally absorbed into effective gravity.

The Navier-Stokes equation (1.14), for describing the acceleration (with respect to a frame rotating with a frequency Ω) of an air parcel of velocity $\mathbf{u}$, considers all forces on the air parcel

$$\frac{D\mathbf{u}}{Dt} + 2\Omega \times \mathbf{u} = -\frac{1}{\rho}\nabla p - \Omega \times (\Omega \times \mathbf{r}) - g\hat{\mathbf{k}} + \mathbf{F}. \tag{1.14}$$

In addition to terms already described, $-\frac{1}{\rho}\nabla p$ arises from pressure gradients and $-g\mathbf{k}$ accounts for the influence of true gravity acting on the parcel. The final term $\mathbf{F}$ is a friction per unit mass term.

A scale analysis of equation 1.14 at midlatitudes reveals that the Coriolis and pressure gradient forces are in approximate balance. As a consequence the following

much simpler equations which are called the geostrophic wind expressions may be used in these regions.

$$fv \approx \frac{1}{\rho}\frac{\partial p}{\partial x}, \qquad fu \approx -\frac{1}{\rho}\frac{\partial p}{\partial y}, \tag{1.15}$$

where $f = 2\Omega \sin\phi$ is the Coriolis parameter and u and v are the mean winds in zonal (west-east) and meridional (south-north) directions respectively.

These equations show that, at midlatitudes, longitudinal and latitudinal temperature gradients generate meridional and zonal winds respectively.

1.3 Meteor Theory

Meteors can often be observed as streaks of light in the night's sky when a meteoroid impacts upon the Earth's atmosphere at a large velocity imparting some or all of its energy as heat, light and ionisation. The primary source of meteoroids are comets. At perihelion comets are heated by solar radiation and the volatile components are evaporated and ejected from the comet. Meteoroids are particles of dust which are dislodged from the comet as the gas evaporates from its surface. Some of these meteoroids eventually collide with the Earth, ablating in our atmosphere over a range of heights, from 70 – 110 km, depending on their size and velocity.

Meteor ionisation trains can be categorised into either of two types: overdense or underdense. Most of the echoes observed by the AMOR system belong to the underdense class where the electron density in the train is low enough that secondary radiative and absorptive effects on the incident wave can be neglected allowing the electrons to be thought of as independent scatterers. During ambipolar diffusion the diameter of the trail approaches the radar wavelength, interference occurs between the reflections from electrons at the near and far sides of the meteor train causes the returned power to be reduced.

If the electron density within the meteor train is large enough, then secondary scattering from electron to electron becomes important and the electrons can no longer be thought of as independent scatterers. As a result, the incident wave cannot penetrate freely and the dielectric constant, defined for an ionised gas as

$$\kappa = 1 - \frac{N\lambda^2}{\pi} r_e, \tag{1.16}$$

(where N is the number of electrons per cubic meter, $r_e = (\sigma_e/4\pi)^{1/2}$ is the classical electron radius and λ is the wavelength of the incident radiation) is negative. Such meteor trains are classified as overdense.

The transition from underdense to overdense meteor trains occurs at a critical line density, q_{tr}. The critical volume density, N_c, is defined to be that which gives $\kappa = 0$. From equation 1.16 and the following expression for the electron volume density:

$$N(r,t) = \frac{q}{\pi\left(4Dt + r_0^2\right)} \exp - \left[r^2/\left(4Dt + r_o^2\right)\right], \tag{1.17}$$

(where r_0 is the initial radius, at time $t = 0$, of the trail, D is the ionic diffusion coefficient and q is the electron line density) one obtains the expression

$$N_c = \frac{\pi}{\lambda^2 r_e} = \frac{q}{\pi\left(4Dt + r_0^2\right)} \exp - \left[r_c^2/\left(4Dt + r_o^2\right)\right], \tag{1.18}$$

and the transitional value of q, q_{tr}, is found by setting the square of the critical radius[2] $r_c^2 = 4Dt + r_0^2 = \lambda^2/4\pi^2$ to give

$$q_{tr} = \frac{\exp(1)}{4r_e} \simeq 2.4 \times 10^{14} \text{electrons } m^{-1}. \tag{1.19}$$

A meteor train with a line density $q > q_{tr}$ is defined to be an overdense trail and will yield overdense type echoes. Meteor trains with $q < q_{tr}$ produce underdense type echoes.

As underdense trails diffuse, the returned power decreases exponentially over a time period typically lasting a few tenths of a second. The echoes received from overdense trains may last for several seconds and generally become quite distorted as they are acted on by local wind and turbulence. As a probe for measuring winds the underdense echoes produced good results (because the ionisation trains retain a linear shape) however only wind velocities calculated from the initial, 0.3 seconds, of overdense type echoes were used in measurements later described.

1.3.1 The Formation of a Meteor Train

As the meteoroid enters the atmosphere it encounters the atmosphere's increasing density. Heat from atmospheric collision causes atoms to be vaporized from the surface of the meteoroid. In the case where the meteor's radius is less than the mean

[2]The radius within which $\kappa \leq 0$.

free path of the air particles, the impact momentum and energy are transferred to the meteor by direct hits of the air particles. As no air cushion is produced there is a relatively high coefficient of heat transfer and only a small percentage of the meteor's kinetic energy is needed for complete vaporisation. Hence these meteors disintegrate in a relatively short time. A second scenario relates to larger meteors whose radius is much larger than the mean free path of the air molecules. Under this condition a hydrodynamic cushion or air cap forms in front of the meteor which reduces the heat transfer coefficient[3] allowing the meteor to penetrate more deeply into the Earth's atmosphere.

Considering an ablating meteoroid with mass m and density ρ_m, it is possible to define a dimensionless shape factor, A, such that the effective cross-sectional area of the meteor is given by

$$A\left(\frac{m}{\rho_m}\right)^{\frac{2}{3}}. \tag{1.20}$$

$A \simeq 1.2$ for a sphere[4]. In general, irregularly shaped meteoroids may have a mean A similar to that for a sphere due to the meteoroid's rotation i.e., $A \approx 1.0$ for any meteor.

The meteoroid's momentum is transferred to the air particles that it encounters along its path. If a meteoroid moves with a velocity $\mathbf{v}$ for a time dt the mass of air (dm_a) intercepted by the body will be

$$dm_a = A\left(\frac{m}{\rho_m}\right)^{\frac{2}{3}} \rho_a \mathbf{v} dt, \tag{1.21}$$

where ρ_a is the density of air.

The air particles in this volume will therefore gain momentum at a rate of

$$\Gamma \mathbf{v}\frac{dm_a}{dt} = \Gamma A\left(\frac{m}{\rho_m}\right)^{\frac{2}{3}} \rho_a \mathbf{v}^2, \tag{1.22}$$

per second where Γ is the dimensionless drag coefficient, which generally lies between 0.5 and 1.0.

The meteoroid will lose momentum at a rate of $m\frac{d\mathbf{v}}{dt}$ per second and therefore the rate of momentum per unit mass gained by the air particles, is given by

[3]The heat transfer coefficient may decrease by as much as two orders of magnitude.

[4]for a sphere $A = \left(\frac{9\pi}{16}\right)^{\frac{1}{3}} \simeq 1.2$.

$$m\frac{d\mathbf{v}}{dt} = \Gamma A\left(\frac{m}{\rho_m}\right)^{\frac{2}{3}}\rho_a \mathbf{v}^2, \tag{1.23}$$

and dividing both sides by m gives

$$\frac{d\mathbf{v}}{dt} = \frac{\Gamma A}{\sqrt[3]{\rho_m}^2\sqrt[3]{m}}\rho_a \mathbf{v}^2, \tag{1.24}$$

which is commonly referred to as the Drag Equation.

Along the meteoroid's path much of the kinetic energy is converted to heat; however usually this energy also produces light and the ionisation of the surrounding atmosphere . There are many processes which govern the ablation of the meteoroid and these are dependent on the meteoroid's size, composition and height[5]. Independent of which process is relevant for a particular meteoroid at a particular time, equation 1.25, called the differential mass equation, can be obtained if it is assumed that the rate of loss of mass is proportional to the kinetic energy given to the surrounding air mass.

$$\frac{dm}{dt} = -\frac{\Lambda A}{2\xi}\left(\frac{m}{\rho_m}\right)^{\frac{2}{3}}\rho_a \mathbf{v}^3, \tag{1.25}$$

where ξ is the heat of ablation per unit mass of the meteor and Λ is the heat transfer coefficient.

As the ablated atoms are moving at the same velocity as the meteoroid, their kinetic energies will range from a few tens, up to several hundred electron volts (due primarily to their large range in masses). These energies are sufficiently large enough to ionise the surrounding gases and the energy of the ionisation created per second is

$$q\mathbf{v}\eta = -\frac{1}{2}\tau_q\frac{dm}{dt}\mathbf{v}^2, \tag{1.26}$$

where τ_q is a dimensionless ionisation efficiency factor and η is the mean ionisation potential per atom.

Again substituting for $\frac{dm}{dt}$ in the differential mass equation (1.25) the ionisation equation which describes the number of electrons produced per unit of path length, q, is obtained

$$q = \tau_q\frac{\Delta A}{4\xi\eta}\left(\frac{m}{\rho_m}\right)^{\frac{2}{3}}\rho_a v^4. \tag{1.27}$$

[5]The atoms which form the meteoroid may leave the body surface without significant interference when the mean free path of the air particles is greater than the meteor's radius.

It is this train of electrons that reflects the electro-magnetic pulse transmitted by the radar.

1.3.2 Radio Wave Reflection from a Meteor Train

Underdense meteor echoes are characterised by a rapid increase to maximum amplitude level, followed immediately by an exponential decay, often with Fresnel diffraction oscillations superimposed on the decay. Figure 1.7 shows the geometry of a meteor train relative to a transmitter and receiver assumed close together. The following analysis of radio wave reflection from meteor trains assumes they are composed of stationary electrons and the diameter of the train is much smaller than the wavelength of the radar.

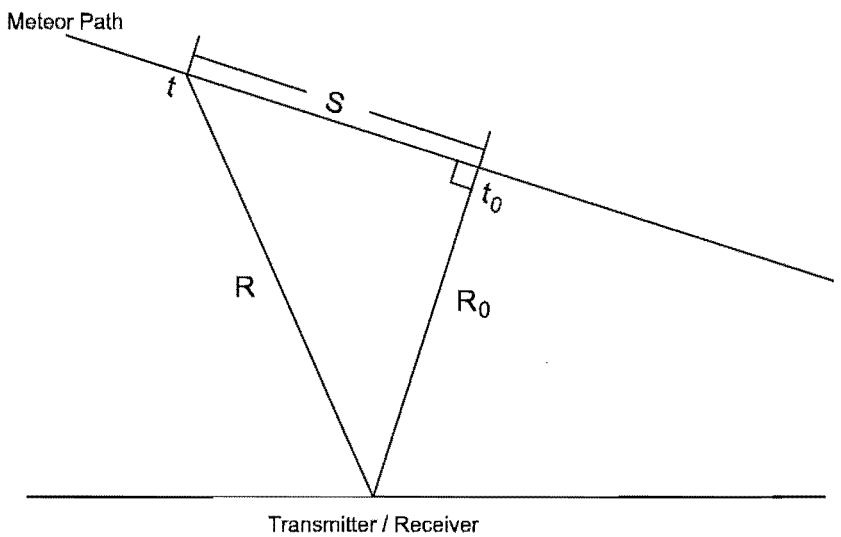

Figure 1.7: The geometry of a meteor train relative to the radar transmitter / receiver station.

The scattering cross-section of a free electron is $\sigma_e = \pi r_e^2 \sin^2 \gamma$ where r_e is the classical radius of the electron and γ is the scattering angle. For backscatter $\gamma = \pi/2$, implying $\sigma_e \approx 1 \times 10^{-28}$ m^2. The power flux incident on the train is $\Phi_i = P_T G_T / 4\pi R^2$ where P_T is the transmitted power, G_T is the antenna gain in that direction relative to an isotropic radiator and R is the range to the train. The effective absorbing area of the receiving antenna is $G_R \lambda^2 / 4\pi$, where G_R is the gain of the receiver relative to an isotropic radiator. Thus the power appearing at the

input of the receiver due to the backscattered signal from one electron is

$$P_e(R) = \frac{P_T G_T}{4\pi R^2} \frac{\sigma_e}{4\pi R^2} \frac{G_R \lambda^2}{4\pi} = \frac{P_T G_T G_R \lambda^2 \sigma_e}{64\pi^3 R^4}. \tag{1.28}$$

As all the electrons in a line element, ds, will scatter in phase (assuming the width of the train at formation is much smaller than λ) the field vectors, rather than the power fluxes must be added. At the receiver the peak amplitude of the field due to a single scattering electron is $(2rP_e)^{1/2}$, where r is the receiver input impedance. The absolute phase of the returning wave, which has covered a distance $2R$, is included in the time varying expression $\sin(2\pi ft - 4\pi R/\lambda)$, in other words the change in R produces a modulation in the phase of the returning wave. Now an expression for the instantaneous amplitude of the received signal from all the electrons in the line element, ds, may be written as

$$dA_R = (2r\,P_e)^{1/2}\, q(t) \sin\left(2\pi f t - \frac{4\pi R}{\lambda}\right) ds, \tag{1.29}$$

where $q(t)$ is the number of electrons per meter path (the electron line density).

From integration of the above expression the total field due to the electrons in the trail between s_1 and s can be obtained.

$$A_R = (2\,r\,P_e)^{\frac{1}{2}}\, q(t) \int_{s_1}^{s} \sin\left(2\pi ft - \frac{4\pi R}{\lambda}\right) ds. \tag{1.30}$$

For simplicity $q(t)$ is assumed constant along the train and R is approximated by $R \approx R_0 + s^2/2R_0$. In addition the following transforms are made $\chi = 2\pi ft - 4\pi R_0/\lambda$ and $2s = x\,(R_0\lambda)^{1/2}$ which gives

$$A_R = \frac{(2\,r\,P_e\,R_0\,\lambda)^{\frac{1}{2}}}{2} q(t) \int_{x_1}^{x} \sin\left(\chi - \frac{\pi x^2}{2}\right) dx. \tag{1.31}$$

The Fresnel integrals of optical diffraction theory are

$$C = \int_{-\infty}^{x} \cos\left(\frac{\pi\xi^2}{2}\right) d\xi \text{ and } S = \int_{-\infty}^{x} \sin\left(\frac{\pi\xi^2}{2}\right) d\xi. \tag{1.32}$$

Substituting these into equation 1.31 gives

$$A_R = \frac{(2rP_eR_0\lambda)^{\frac{1}{2}}}{2} q(t)\,[C\sin\chi - S\cos\chi]. \tag{1.33}$$

Because the maximum oscillation frequency of the Fresnel integrals is much less than the radio frequency f a time-average over an interval which is small compared

to the smallest oscillation period of C and S is taken. This gives an expression for the quasi-instantaneous power, P_R, received from the electrons in the trail between s and s_1

$$P_R = \frac{A_R^2}{2r} = \frac{P_e R_0 \lambda}{2} \left[\frac{C^2 + S^2}{2}\right] q^2(t), \tag{1.34}$$

which upon substituting equation 1.28 and parameters in SI units yields (in units of watts)

$$P_R = 2.5 \times 10^{-32} P_T G_T G_R \left(\frac{\lambda}{R_0}\right)^3 \left[\frac{C^2 + S^2}{2}\right] q^2(t). \tag{1.35}$$

This is the basic expression governing the echo power from an underdense train and it has exactly the same form as that describing the optical diffraction pattern produced from a Fresnel straight edge optical experiment (neglecting any time dependence of q).

The term

$$\left[\frac{C^2 + S^2}{2}\right], \tag{1.36}$$

has the value of unity when evaluated from $s = -\infty$ to $+\infty$. Although the approximation of R doesn't hold true for parts of the train that are far from the t_0 (specular reflection) point, this term is close to unity when taken over a few Fresnel zones either side of t_0. As a consequence the more remote sections of the train do not contribute significantly to the echo power.

The Fresnel behaviour of received echoes limits the resolution with which t_0 can be located to the length of one Fresnel zone. The length of the first Fresnel zone is given by

$$\sqrt{\frac{R\lambda}{2}}, \tag{1.37}$$

which equates to a length of $\approx$ 1 km for the AMOR system.

Hence the limitation in height resolution can be approximated by $\cos\phi$ (km) where ϕ is the elevation angle. For typical values of ϕ the associated maximum height resolution ranges between 0.5 and 0.9 km.

1.4 The AMOR Meteor Radar

The AMOR winds system is an additional development to the AMOR meteor orbit radar which has been in operation for nine years at Birdlings Flat near Christch-

urch in the South Island of New Zealand at 172° 39´E, 43° 34´S. To date AMOR has recorded the heliocentric orbits for almost one million Earth-intersecting meteoroids. Unique to the AMOR system is its narrow fan shaped radiation pattern which enables the implicit location of the meteor train target. The radar gain obtained from concentrating all the transmitted power into a narrow beam enables echoes to be recorded from meteors down to a limiting diameter of about (the value is speed dependent) 80 μm [Baggaley & Bennett 1996].

Timelags between the occurrence of echoes on spaced receivers allow the calculation of meteor velocities and gives a significant increase in the number of meteor velocity measurements when compared with previous methods which relied entirely on Fresnel diffraction analysis. Data from the remote sites (with an approximate separation of 10 km) are transmitted back to the Home site via FM Data links which operate at 39.0 MHz (vertically polarised) and 39.3 MHz (horizontally polarised) for the Spit and Nutt sites respectively. The meteor signal data from all receiving antennas are stored on the computer's hard disk for later reduction.

1.4.1 Transmitter

The AMOR transmitter building is located approximately 500 m from the building containing control equipment and receivers and generates a 60 kW 66 μs duration pulse at 26.2 MHz when triggered by the transmitter trigger signal sent from the receiver building every 2.64 ms.

1.4.2 Aerials

The transmission aerial is a 40 wavelength long co-linear broadside array (refer figure 1.9) which confines the transmitted radiation power to a $\approx 2°$ (FWHM) fan shaped pattern providing equal coverage in both the North and South directions. Although the radiation pattern is constrained in azimuth it is broad (15°– 75°) in elevation with the north/south symetrical radiated power maximum at elevation angles of about 30° and 120°.

There are five receiver aerials in total of which three are located at the Home site (refer figure 1.10) and provide interferometry for measuring the elevation angle of the returned signal. The remaining two receiver aerials are at the remote sites,

Nutt and Spit, and are used in making meteor orbit and time of flight speed measurements.

1.4.3 Receivers

There are five 26.2 MHz receivers in the AMOR radar system each of which has an input impedance of 50 Ω. The three Home-site elevation measuring receivers are fed via underground 50 Ω coaxial cables from the receiving antennas. Each FM receiver is fed via 50 Ω coax from a 39 MHz 6 element Yagi antenna. A 24.6 MHz crystal provides a local oscillator reference with which the received signals are mixed then low-pass filtered to produce signals at an Intermediate Frequency (I.F.) of 1.6 MHz. The receivers (with bandwidths of 30 kHz) are partially blanked during the transmitter pulse. This prevents damage to the receiver circuits whilst still allowing sufficient signal to capture the phase of the ground pulse.

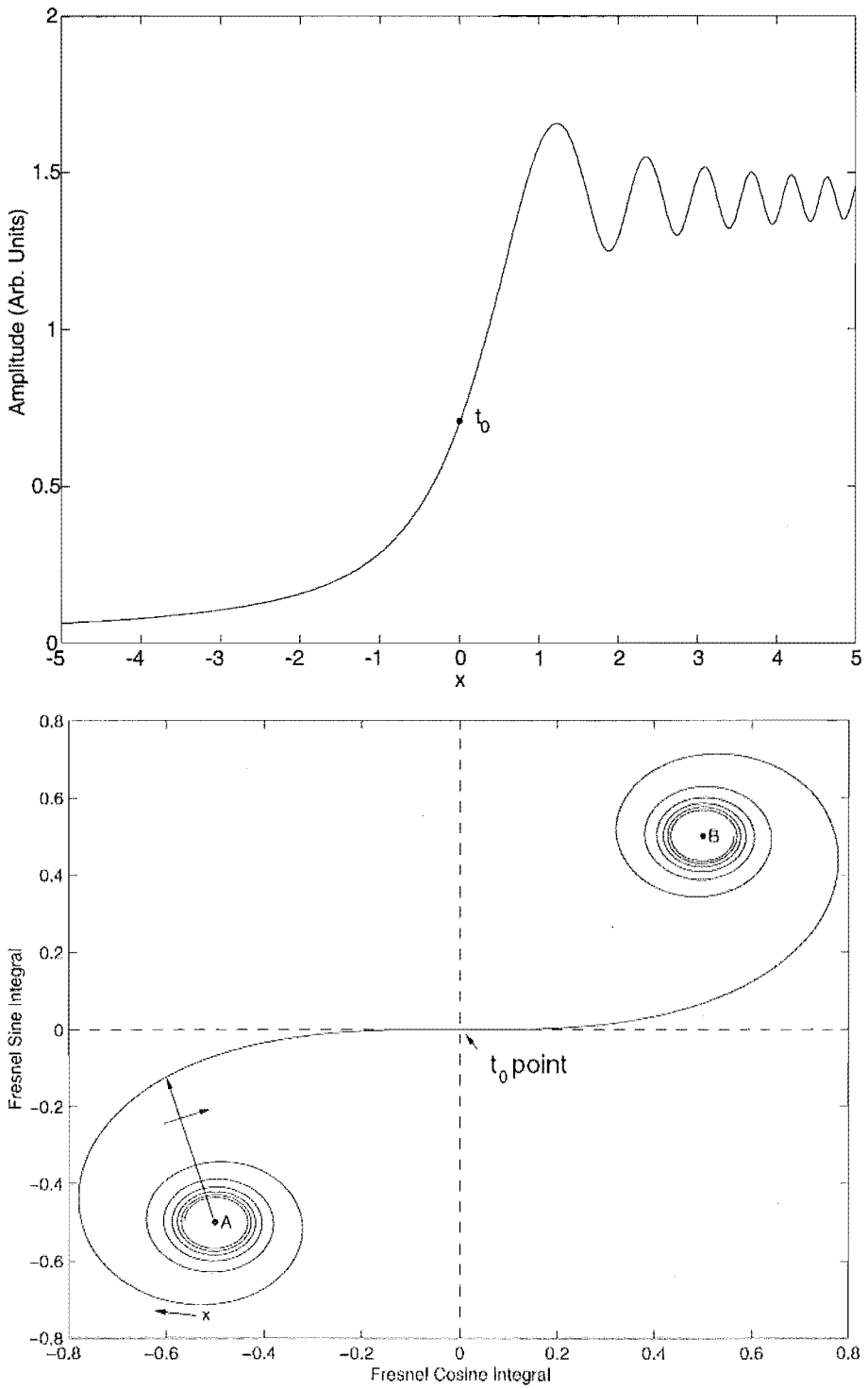

Figure 1.8: The amplitude of a meteor echo (top) from diffraction theory displaying a Fresnel oscillation (diffusion has been neglected). The amplitude and phase of the echo can be expressed simultaneously with a Cornu Spiral (bottom). The length of a vector anchored at point A with its head following the curve indicates the echo's amplitude and the vector's direction indicates the phase.

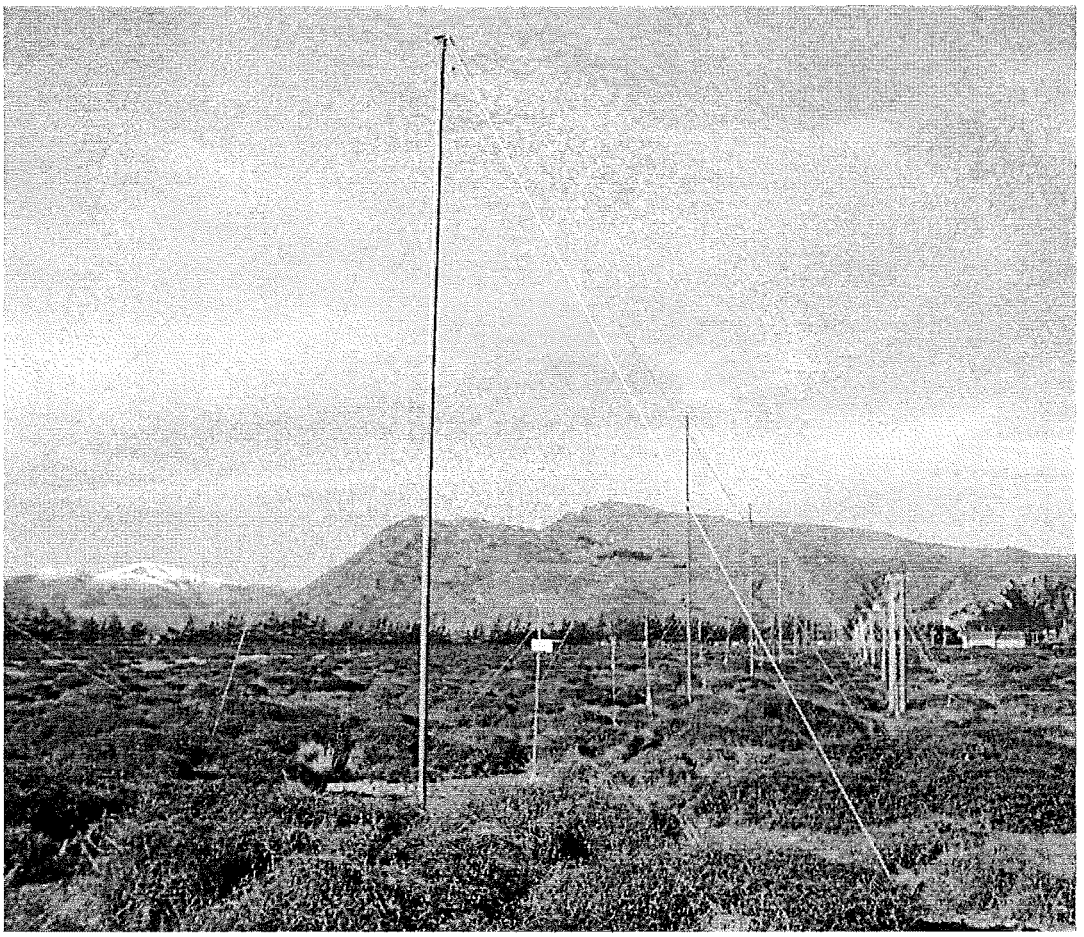

Figure 1.9: The transmitter array running east-west for approx 500 m.

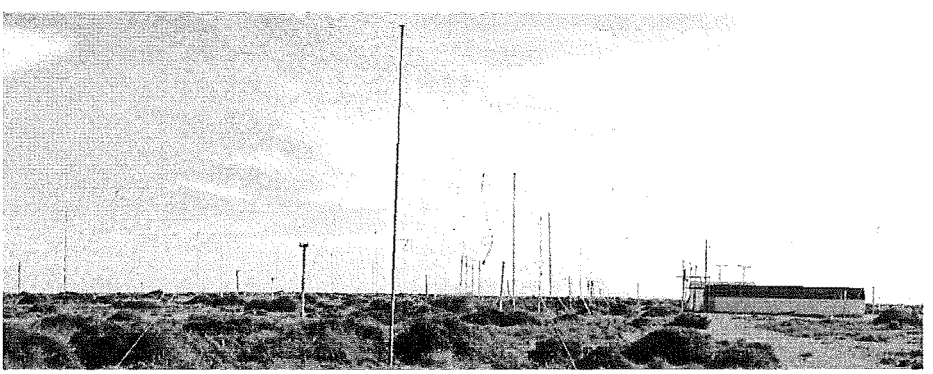

Figure 1.10: One of the three receiver antennas located at the Home site. Antenna 1 is in the foreground with the receiver building to its right. Two other identical antennas running parallel to antenna 1 can be seen in the left background.

Chapter 2

The AMOR Meteor Winds System

The AMOR meteor radar, located at Birdlings Flat near Christchurch on the South Island of New Zealand, has been measuring meteoroid orbits since 1990 (refer figure 2.1). There are two buildings at the Birdlings Flat Home site, one houses the radar control systems and receiver equipment while the other houses the HF transmitter. The transmitted signal is radiated in both northward and southward directions from a broadside array at the Home site and any subsequent echo can be detected on three local receiving antennas, forming a dual interferometer, located near the receiver building. For meteoroid orbit studies two remote sites called Nutt and Spit located approximately 10 km from the Home site enabled the measurement of parameters from which meteoroid orbits could be derived, namely the meteoroid's speed, direction and elevation angle.

Prior to the work presented in this thesis, measurement of meridional wind speeds at meteoric heights by AMOR was not possible as the phase and hence the frequency of the returned signal was not measured. Hence the Doppler shift of the transmitted signal, which is interpreted for a wind measurement, could not be obtained. To ensure that the current system remained unaffected by the addition of a meteor winds component, it was decided to implement the winds facility as a pseudo stand alone feature having as small an impact on AMOR as possible. A separate computer with its own dedicated software was required to control the data collection process.

The signal, taken from the intermediate frequency output of receiver 1 (refer figure 2.3), is decomposed into its in-phase and phase-quadrature components with the addition of new hardware referred to in the thesis as the Doppler module. Data from the Doppler module are then transferred into the winds control computer through a process called Direct Memory Access (DMA).

Where it seemed advantageous, use was made of the existing AMOR system. As

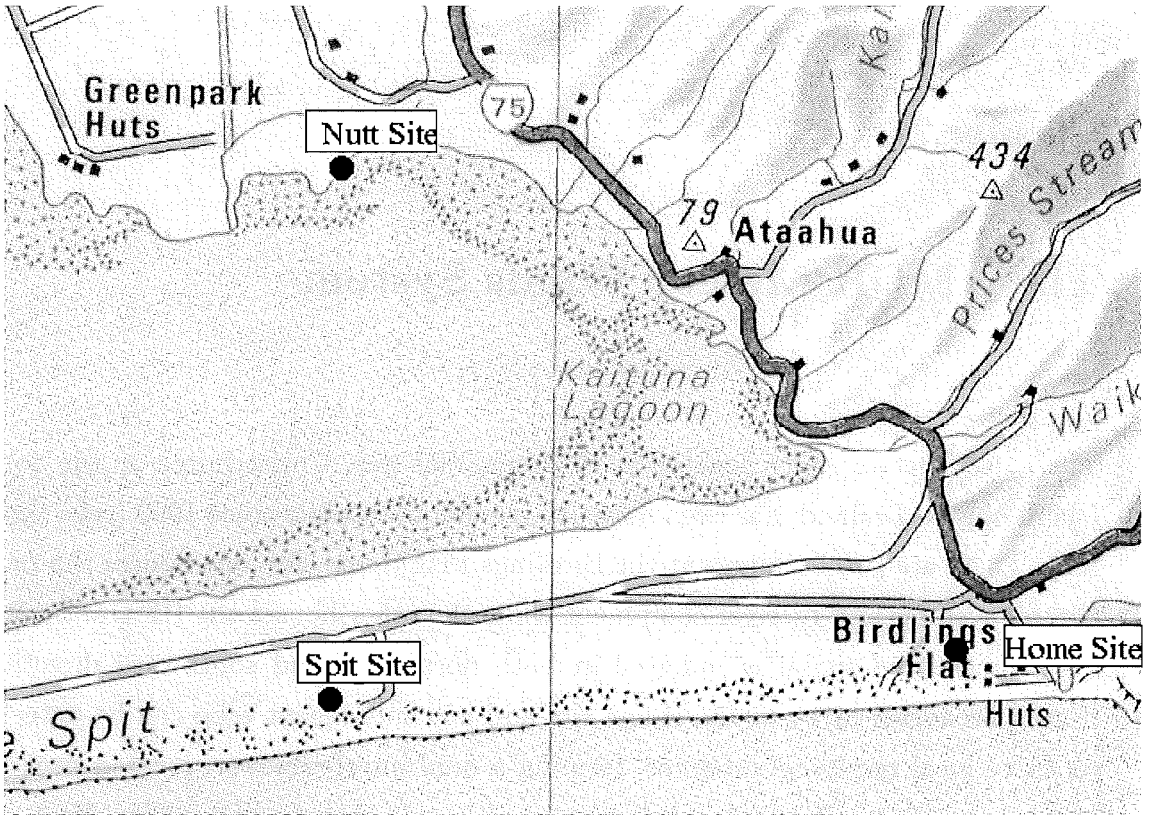

Figure 2.1: The local area of the AMOR meteor winds radar showing the location of the two remote sites relative to the Home site.

a result the detection of an echo and the determination of its spatial location is achieved with the original AMOR system; hence the winds system is reliant on the orbit system being operated simultaneously. Communication lines between the orbit and winds control computers enable synchronised data capture and the transference of an identification number which is stored along with the echo data to be used in the analysis process when matching the orbit and wind echo data.

The line of sight (radial) velocity of the wind measurement is directly proportional to the Doppler shift experienced by the transmitted pulse. By measuring the elevation angle of the wind measurement the horizontal wind velocity can be obtained, by projection (refer figure 1.2), if no vertical motion is assumed. This assumption is discussed in more detail in section 2.3.5.

2.1 Hardware for Doppler System

The new meteor winds system required the development of the following additional hardware to the existing AMOR radar system.

- A Doppler module which resides in the receiver rack and produces phase components of the received signal.
- An 80386 computer with 8 MBytes of RAM and a 650 MByte hard disk drive for data storage.
- An analogue to digital converter / direct memory access card fitted inside the computer enabling storage of data to computer memory.

The first and last components were constructed during this work and are discussed in detail in this chapter. The computers are housed inside metal boxes and stacked vertically in a frame (refer figure 2.2).

2.1.1 Doppler Module

The circuitry which takes as input the analogue signal from one of the AMOR receivers and produces digital phase and phase-quadrature values is referred to as the Doppler module. The Doppler module (refer figure 2.3) operates at the receiver intermediate frequency (I.F.) of 1.6 MHz and is located in the AMOR receiver rack (refer figure 2.4).

The Doppler module makes use of two phase sensitive detector (Motorola's MC1496) components to decompose the signal, as received by AMOR, into two orthogonal components, in-phase and phase-quadrature. Each phase sensitive detector (PSD) requires two input signals, producing output voltages which are proportional to the signal amplitudes and the cosine of the phase difference between them. Should the two signals be at the same frequency they would have a constant phase difference and the output voltage would be constant: however if the two input signals have different frequencies then the PSD produces a varying voltage at the frequency difference (figure 2.5).

The in-phase component is obtained by comparing the amplified signal at the I.F. from receiver 1 and a beat oscillator at a frequency near 1.6 MHz. To get the

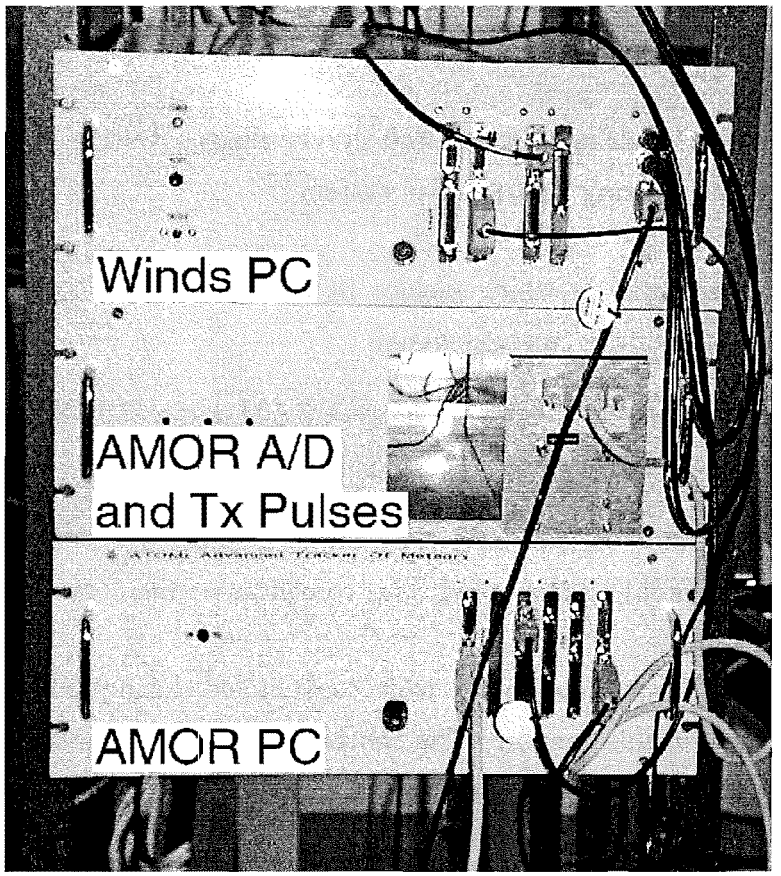

Figure 2.2: AMOR and Winds Control computers at the field station.

quadrature component the beat oscillator is moved in phase by 90° (figure 2.6).

The receiver local oscillator circuits are separate from the transmitter frequency control circuits and the effective reference frequency (the sum of the 24.6 MHz local oscillation and the 1.6 MHz beat oscillation frequencies) is adjusted to be close to that of the transmitter. Unless methods were adopted to ensure oscillator stability, drift could cause the frequency sum to vary significantly - up to a few hundred hertz, in which case the transmitter frequency (f_T) and the reference frequency (f_{ref}) may differ by an amount Δf which can be decomposed into an integer multiple of the sampling frequency (nf_s) and a residual Δf_r ($f_T = f_{ref} \pm nf_s + \Delta f_r$). Provided Δf_r is not near to the Nyquist frequency (f_N) the Doppler shift (f_D) is unambiguous (refer figure 2.7).

Prevention of wind speed ambiguities is accomplished by phase locking the 1.6 MHz beat oscillator to the transmitter frequency. This is achieved by comparing the intermediate frequency signal of a low gain receiver (discussed in section

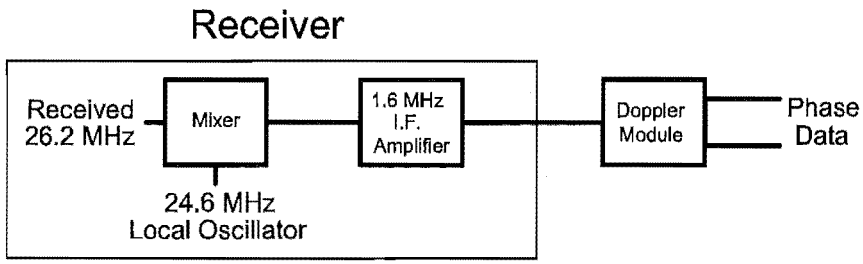

Figure 2.3: The mixer and low pass filter inside the radar receivers provide the signal at an intermediate frequency of 1.6 MHz to the Doppler module. The Doppler module produces phase and quadrature components.

2.1.3), connected to a low gain antenna with the 1.6 MHz beat oscillation using exclusive-or logic as a PSD. The output of this, during the transmitter pulse, is integrated and then applied to a variable capacitance diode which controls the 6.4 MHz oscillator. The 1.6 MHz beat oscillation is produced by frequency division of the 6.4 MHz oscillation using a Johnson counter. Further circuitry in the Doppler module provides a 95 Hz reference frequency offset to prevent wind speed ambiguities.

2.1.2 Analogue to Digital Converter and Direct Memory Access Card

The Analogue to Digital Converter (A/D) and Direct Memory Access (DMA) card (refer circuit diagram, figure 2.14) resides inside the winds control PC. The analogue data for both phase channels are converted into their 8-bit digital equivalent values, at 20 μs intervals, with two 7821 analogue to digital (A/D) converters. The conversion is applied to both channels simultaneously, the values are buffered before being read sequentially and stored in the computer's memory.

The microprocessor controlled Direct Memory Access (DMA) card reads data from the A/D cards for writing to memory. Before a DMA transfer can be achieved the microprocessor requires initialisation by the data capture software. Once initialised the DMA transfer process waits until bit #8 on the DMA card's output port goes high; this is for timing purposes. For each transmitter pulse 100 DMA transfers at 20 μs intervals are initiated for each channel. The 20 μs timing is achieved on the DMA card with the circuit shown in figure 2.13.

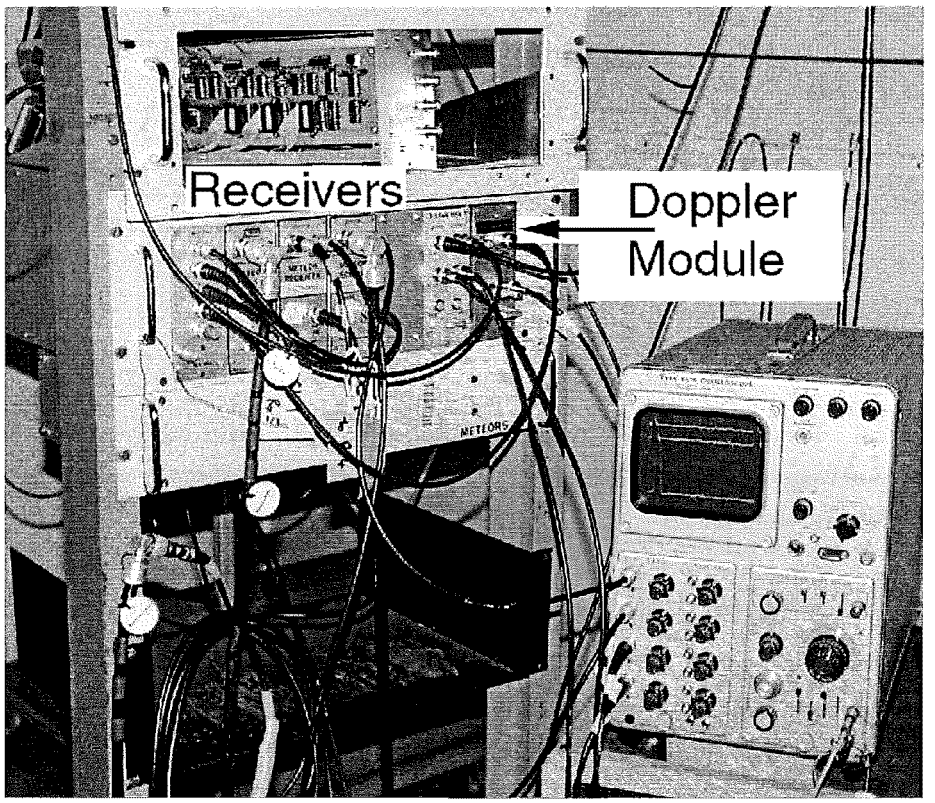

Figure 2.4: AMOR receiver rack showing the location of the Doppler module. Co-axial connections from the receivers video outputs are connected to the four channel oscilloscope for viewing echoes. From left to right the middle rack houses the local oscillator outputs, the three interferometer receivers, the two link receivers and finally the Doppler module.

2.1.3 Ground Pulse Receiver

The transmitted ground pulse signal is required by the AMOR winds system to lock the main receiver's 1.6 MHz beat oscillation and is received at the receiver building via a simple antenna (single folded dipole) and low gain receiver (refer circuit diagram figure 2.15). A MCL SBL-1 mixer produces the 1.6 MHz I.F. from the 26.2 MHz received transmitter ground pulse signal and the 24.6 MHz local oscillator. The output of this receiver is a square wave with a 1:1 mark to space ratio at the I.F (refer figure 2.8).

2.1.4 Oscillator Lock and Frequency Offset

It was found advantageous to prevent drift of the 1.6 MHz beat oscillator as it may cause the transmitter frequency to appear near the Nyquist frequency (f_N of figure

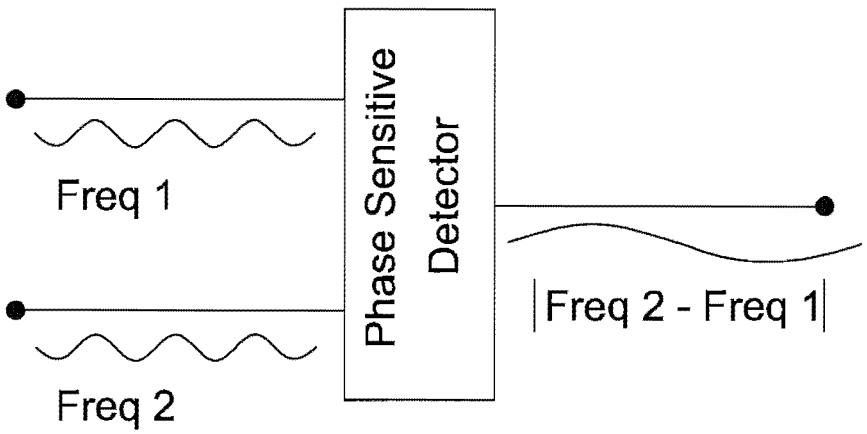

Figure 2.5: Schematic of a phase sensitive detector. The frequency of the output signal is equal to the frequency difference between the two input signals.

2.7) and subsequent wind measurements may become ambiguous. The solution required that the beat oscillator be phase-locked to the transmitter and this was achieved using the circuit detailed in figure 2.16. This section details how this circuit works.

The signal from the Ground Pulse Receiver (GPR) is applied to one input of an exclusive-or gate (S2) which acts as a phase sensitive detector. (The GPR output circuit is designed to give a square wave output with a 1:1 mark to space ratio.) The other input is derived by a complicated divide by 4 circuit from the 6.4 MHz controlled oscillator (the divider circuit is explained below). The Transmitter Trigger Pulse (TTP) is applied to the first of two monostable delay circuits (S6), which triggers the second (S7) approximately 100 μs after the leading edge of the TTP. The output of the second, 60 μs long, enables the tri-state inverter (S5) for this period, in the middle section of the transmitter pulse as received from the GPR. The initial ($\sim$100 μs) delay is to compensate for delays occurring in the cable to the transmitter, the transmitter itself, the ground path and the GPR and is adjustable.

The effect of this is to apply the output of the XOR gate (S2) to the integrator (S4) only during the TX pulse, at other times the integrator input is isolated. If the input to the integrator is not enabled at all the integrator output level will return to the central value of 2.5 V with a time constant of 0.1 s. The output of the XOR gate depends on the phase relationship between its two inputs. If these are exactly in phase the output is zero, if they are at 180° it is 5 V and at

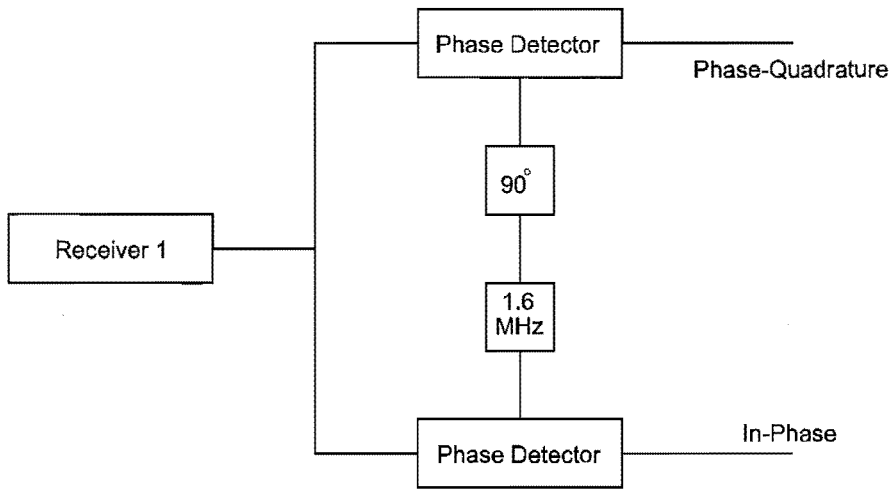

Figure 2.6: Phase detector arrangement as used by AMOR to provide wind data.

intermediate phase angles it consists of a train of pulses whose mark to space ratio varies, and the average level of which varies linearly with phase angle at values of between 0 and 5 V, i.e., at a phase relation of 90° the average output is 2.5 V (refer figure 2.11). At other phase angles (and hence average values) the output of the integrator is increased or decreased during the 60 μs sampling time and then remains approximately constant during the interval between pulses, relaxing towards 2.5 V with the 0.1 s time constant mentioned above. The circuit constants and timing intervals etc. mean that the output will vary by approximately $\pm$10 V over the range of phase angle if this is kept constant.

This output is applied through an inverting buffer (S5) to the variable capacitance diodes in the ~6.4 MHz crystal oscillator which provides a variation range of approximately 4 kHz. The 330 kΩ resistor is connected to cause the input signal to increase the gain of the oscillation transistor when positive voltages are applied to maintain the oscillation at a sufficient level as, under this condition, the small capacitance of the diodes leads to reduced feedback in the oscillator circuit. (The other two operational amplifiers, S8 & S9, connected to the buffer play no part during normal functioning and their operation is described below). The output of the oscillator is passed through buffers (S10 & S11) to the twisted ring counter (S19 & S20) which divides the frequency by four, but in addition supplies two signals having a 90° phase difference which are reference signals in quadrature to be used in the offset frequency generator (S21 & S22). In addition the buffer, S12, supplies a

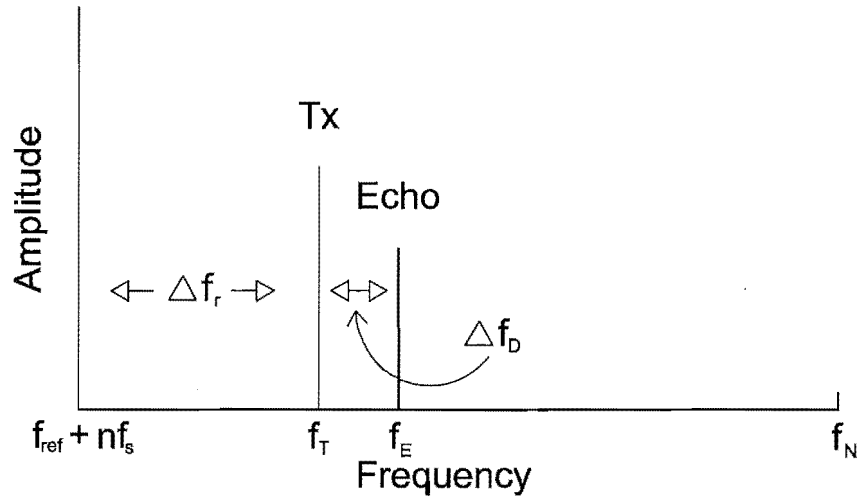

Figure 2.7: The transmitter frequency (f_T) and the reference frequency (f_{ref}) may vary. However f_T and echo frequency (f_E) will both be aliased preserving the Doppler frequency (Δf_D).

6.4 MHz reference signal to a similar twisted ring counter on the two channel phase and quadrature phase sensitive detector. The frequency offset generator comprises S15, S16, S17, S18, S21 and S22. Its function is to advance the phase of the 1.6 MHz signal that is applied to the XOR, S2, by 90° each TTP. The effect of this is that the reference signal as applied to S2 at the sampling intervals appears to be increased in frequency by one quarter of the transmitter pulse rate with the result that when the circuit locks the crystal oscillator to the 1.6 MHz output of the ground pulse receiver the 1.6 MHz reference frequency it provides is lower by this amount. This circuit may be disabled by taking pin 5 of S16 to ground so that the phase shifter

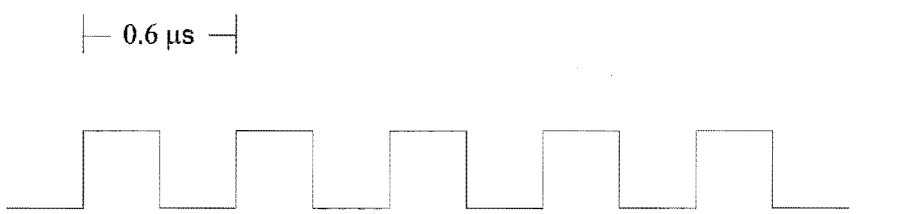

Figure 2.8: The output of the ground pulse receiver is a square wave with a frequency equal to the transmitter frequency (at an I.F. of $\sim$ 1.6 MHz).

S16, S17, S21(a-d) and S22(a-c) remains fixed in one state.

The integrator (S4) produces an average output dependent on the phase difference between the input pulses and the effective 1.6 MHz reference which, when applied to the varactor diodes, corrects the 6.4 MHz oscillator frequency so that it remains locked. The main integrator capacitor and resistor circuit (with a time constant of 0.1 s) is augmented by a short time constant (50 μs) circuit to provide a short correction pulse to assist maintaining lock. However the circuit will not capture the oscillation if more than a very small frequency difference exists. The circuit of S8 is provided for this purpose; S9 is just a buffer. The S8 circuit incorporates a "Wien Bridge" network in a way that makes it a broadband filter peaking at about 0.3 Hz, and high frequency and dc components of the input are not transferred to the output of S9. When the circuit is locked the output is essentially a dc signal with high frequency components which are too small to affect the circuit significantly when applied back to the input of S5. But when the circuit is not locked the phase relation is random and the average of the phase signals applied by S3 to the input of S4 is constant at 2.5 V. In this state the only effective input to S5 comes from S8 and the loop S5 to S8 is an oscillator circuit with a frequency of approximately 0.3 Hz. The oscillation builds up until limited by the diodes (at the expense of some distortion). The output of S5 sweeps the crystal oscillator across its frequency range of approximately 4 kHz at a slow rate, When the reference and received frequencies become nearly equal the phase differences are similar for several successive samples and the average outputs from S3 cause the integrator to produce a correction signal which maintains the oscillator frequency constant and gives rise to a locked condition in which the whole locking circuit reacts to reduce the gain of the buffer, S5, to zero so far as external "perturbing" signals from S8 are concerned, since the phase detector and integrator produce a signal to cancel them. The 0.3 Hz oscillation stops abruptly and the RC circuits discharge slowly.

If the transmitter pulses are not received for any reason, the circuit immediately begins searching back and forward and lock is reestablished in about 1 s once the signal re-appears.

2.1.5 Phase Sensitive Detectors

The Motorola MC1496 Phase Sensitive Detector (PSD, [Motorola 1990]) when operated under its normal conditions takes as input two signals and provides an output voltage in the range $0 - 5$ V which is proportional to the signal amplitude and the cosine of the phase difference between the signal and the reference oscillation. A constant voltage is produced if the signal is at the same frequency as the reference oscillation, however if they are different the PSD output voltage varies at the frequency difference of the two input signals. By having a second 1496 whose reference oscillation is at 90° to the other, two outputs are obtained which are respectively proportional to the in-phase and phase-quadrature components of the input.

The 6.4 MHz clock pulses from the oscillator locking circuit of the previous section are applied to a twisted ring (refer figure 2.12) which not only divides the frequency by four but produces two signals which are 90° out of phase with respect to each other. The outputs A & B are in phase with the clock pulses and lead the outputs C & D by 90°, providing reference signals for in-phase and phase-quadrature channels respectively. Pins 8 and 10 are the positive and negative inputs and pins 6 and 12 are the positive and negative outputs.

2.1.6 DMA

The PC offers Direct Memory Access (DMA) via the DMA controller chip (Intel 8237-5) on the PC motherboard, to allow high-speed data transfer of up to 476 kBytes per second. The analogue to digital converter along with the necessary DMA circuitry to operate in byte transfer mode by the winds PC is given in figure 2.14. To transfer a byte of data (assuming that the DMA devices have been initialised) the data are applied to the input port from the A/D converters and a DMA Request signal is sent via bit #4 of the output port to the DMA controller which sends a hold request to the CPU. At the end of the current bus cycle the CPU sends back a hold acknowledged signal to the DMA controller indicating that the bus is no longer being used. The DMA controller can now have control of the system bus to drive the address bus and control bus which enables the transfer of data from the A/Ds to memory. When the data has been copied to memory a Data Acknowledge (DACK) signal returns and the request is reset. There are two A/Ds on the board (one for each channel) so two DACK pulses are returned from the DMA controller for

every request sent and two bytes are stored in memory (refer figure 2.9). One sweep (pulse interval) consisting of 100 of these two byte transfers is initiated by sending DMA request pulses until the DMA counter reaches zero. The DMA Request pulse goes low at the negative edge of the second DACK pulse and stays low until driven high from the 20 μs clock pulse produced by the circuit in figure 2.13.

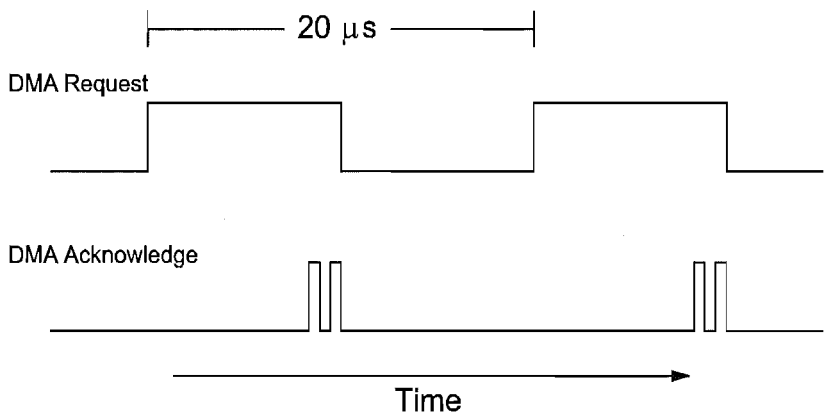

Figure 2.9: DMA Request timing diagram showing the two bytes transferred per DMA Request.

For DMA transfers to occur, first the DMA controller must be informed of the following:

- Whether data are to be written to or read from computer memory
- Type of data transfer i.e., burst, byte or demand transfer mode
- DMA channel to be used
- Memory address for start of transfer

This is achieved using I/O port out instructions from the CPU in the control software.

2.1.7 Communication Lines

There are four communication lines which go from the AMOR system into the winds PC and these are summarised in figure 2.10. A line between bit #0 on the AMOR

PC output port and bit #0 of the winds PC input port indicates the presence of a meteor echo. The transmitter trigger pulse from the AMOR A/D box connects to bit #1 of the wind PC input port for timing purposes. The BIOS clock is updated from an external GPS clock via an RS 232 connection into COM #1. A connection between COM #2 of the AMOR PC and COM #2 of the winds PC enables the transference of a meteor identification number (one byte).

The radar control software (to be described in Chapter 2.2) refers to the external GPS clock when it is first run, as well as at the start of every hour, to update the PC's BIOS clock. This was considered necessary as the BIOS time tended to drift. The AMOR PC also updates its BIOS clock from the same GPS source, synchronising the echo times on the AMOR and winds systems.

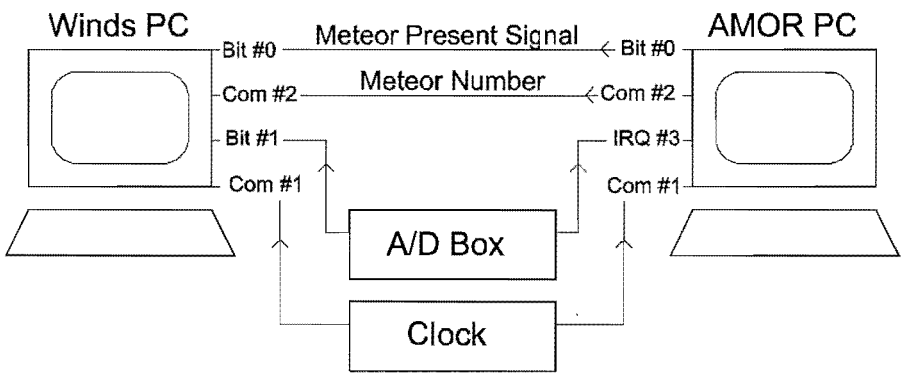

Figure 2.10: Communication lines between the AMOR and winds PCs in the control section in the receiver building. Clock and transmitter pulse connections are also shown.

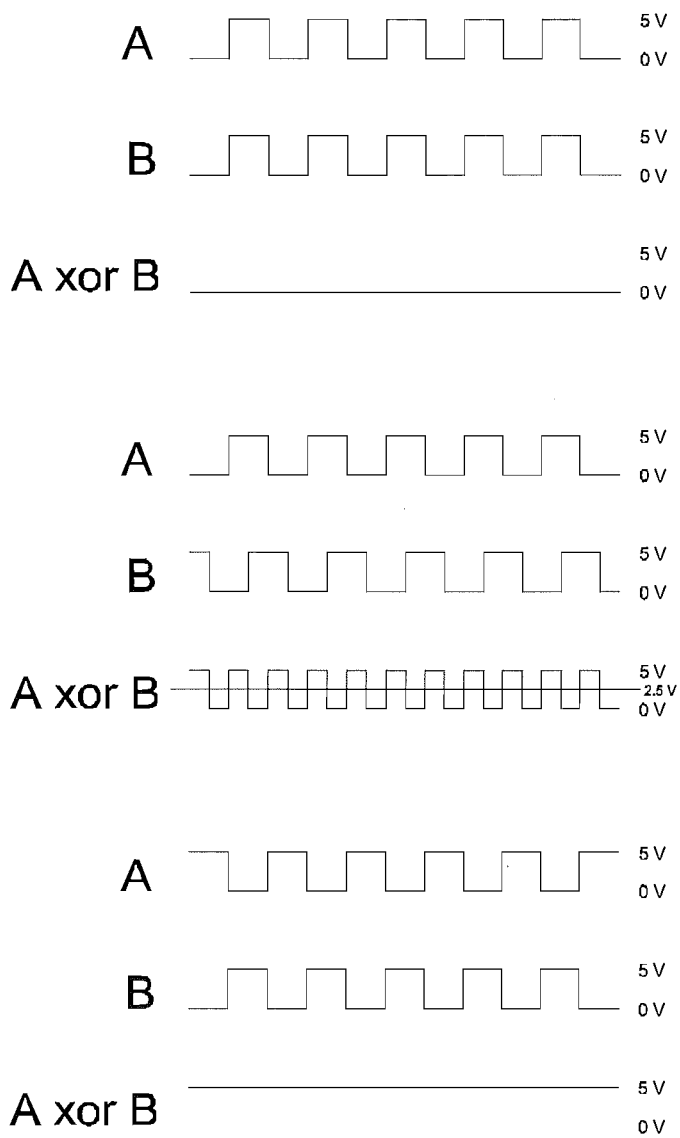

Figure 2.11: A represents the I.F. signal from the GPR and B is the local 1.6 MHz oscillation. The average voltage (produced by an integrator) of the series of output pulses is indicative of the phase difference between the two input signals (A and B). For A and B in-phase the average output is 0 V (top), for A and B 180° out of phase the average output is 5 V (bottom). There is a linear average output voltage dependence with phase difference between the two extremes, i.e., 90°phase difference results in a 2.5 V average output (middle).

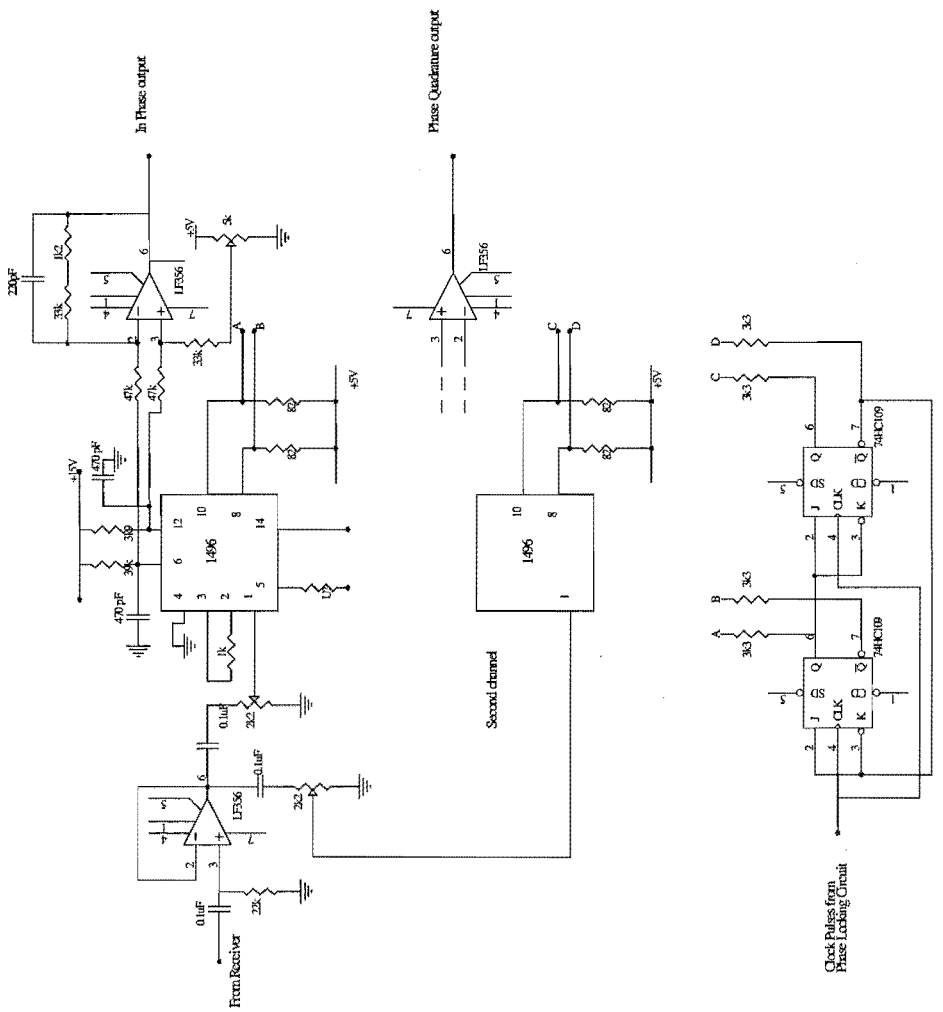

Figure 2.12: Two channel phase and phase-quadrature circuit.

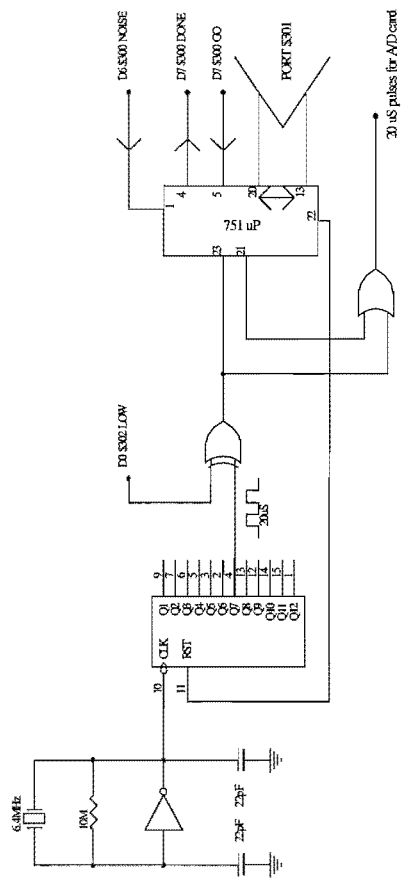

Figure 2.13: Generates 20 μs pulses and houses microprocesser for DMA transfers.

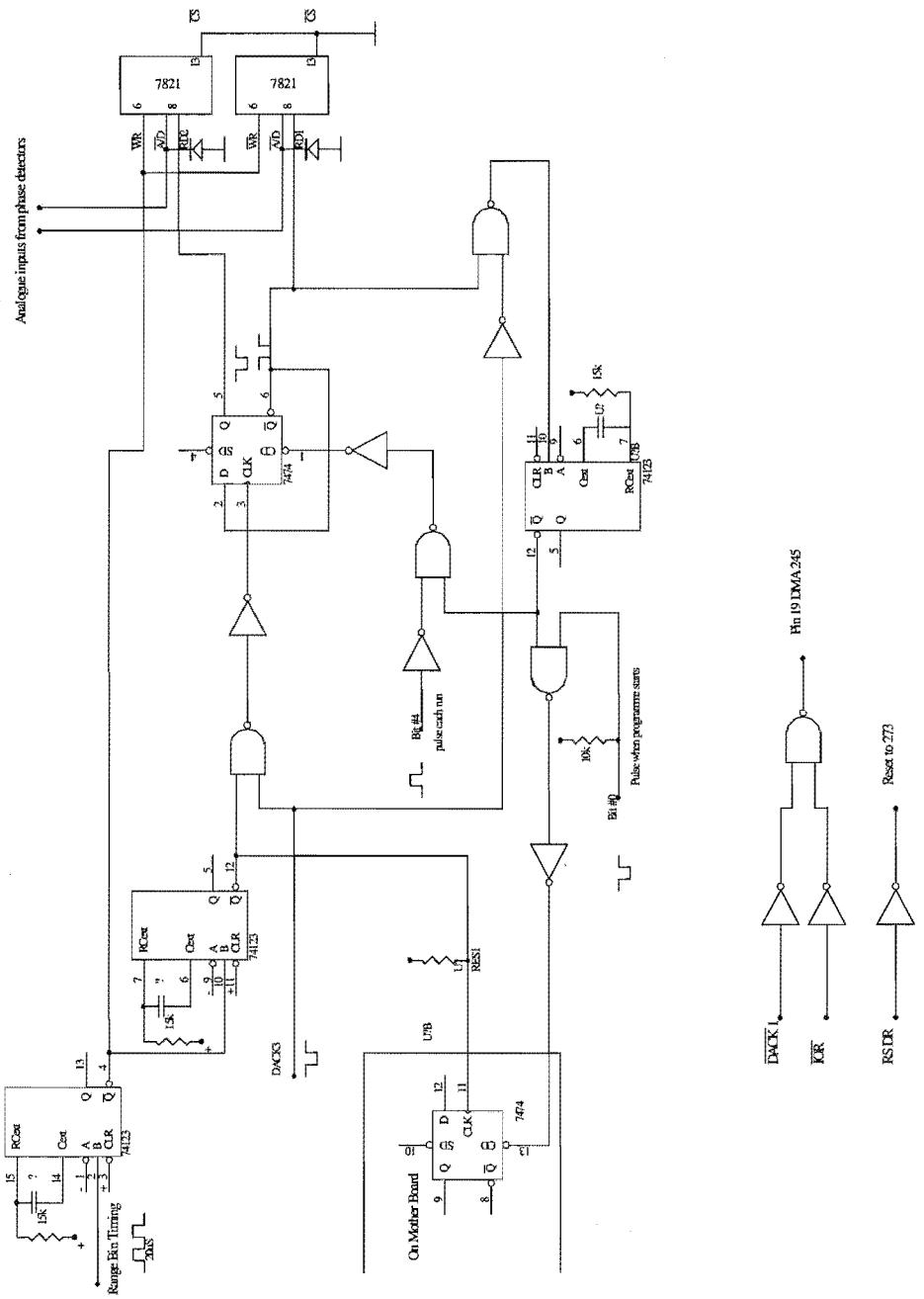

Figure 2.14: Circuitry for anologue to digital conversions.

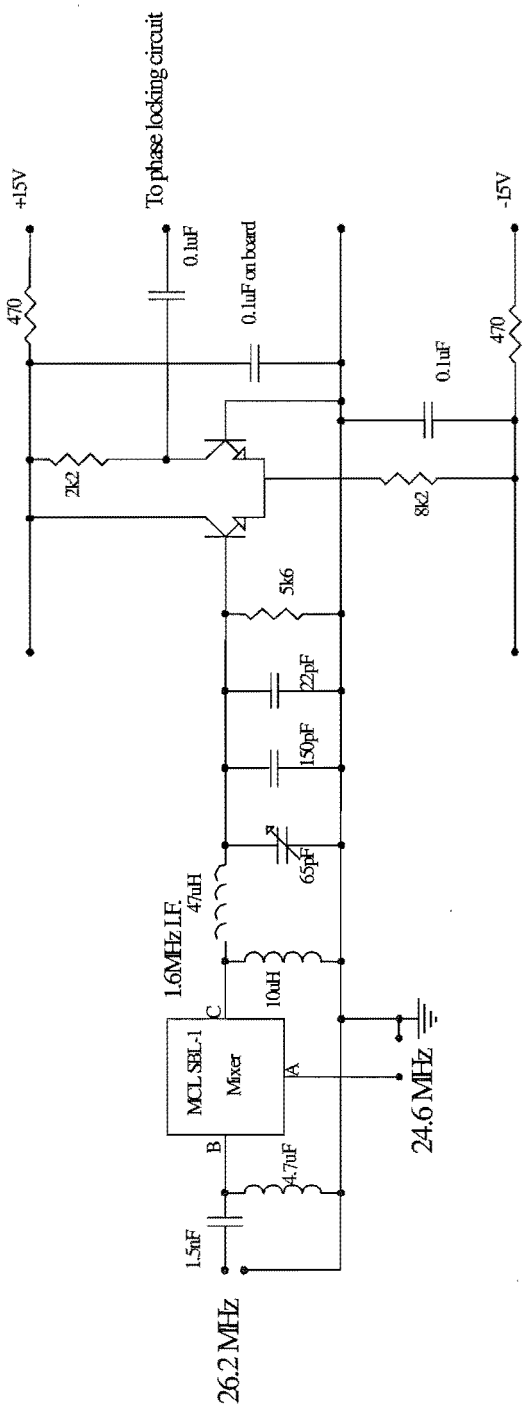

Figure 2.15: Ground pulse receiver circuit.

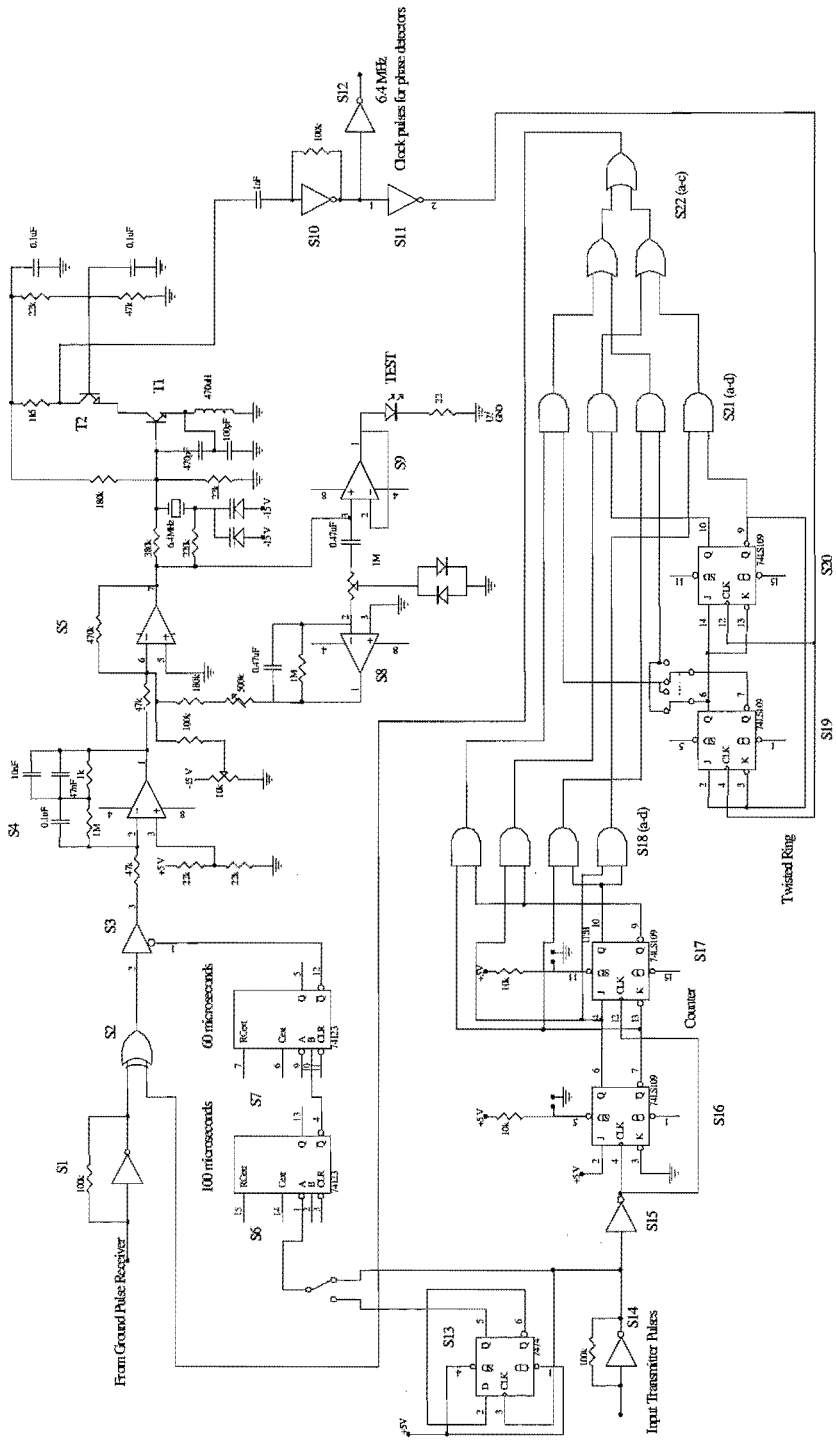

Figure 2.16: 1.6 MHz oscillator lock and offset circuit.

2.2 Data Collection Software

The meteor winds PC control software is called DopRad and can be run from the root directory by typing DopRad<RETURN>. Should the winds control computer accidentally be turned off, then, upon restarting the system, the control programme is automatically restarted (via AutoExec.Bat). The control programme can be halted at any time during the data collection process by pressing the 'S' key, whereupon control of the computer is returned to DOS.

While data collection is in progress the monitor display indicates the system is running along with details of the last recorded echo. The data file being currently accessed and the AMOR defined meteor number are also displayed on the monitor. Displaying this information is advantageous as it can indicate if there are any problems with the system.

The identification of a meteor echo's presence is done by the AMOR observation software. AMOR then informs the winds system that an echo exists by raising the voltage on bit #0 of the AMOR output port; this is connected to bit #0 of the winds input port. The winds control software, DopRad observes this line and when this echo present signal is detected, 128 DMA start pulses are sent to bit #7 of the DMA cards output port to signal the start of DMA transfers. The timing of these DMA start pulses is controlled by the transmitter trigger pulse from bit #1 of the DMA input port. Each start pulse initiates one sweep of 100 DMA transfers 20 μs apart for each of the two channels (in-phase and phase-quadrature). Both the echo and transmitter ground pulse are extracted from the DMA memory storage and written to hard disk.

2.2.1 DMA Transfers

Initialising the DMA Controller

As was discussed in section 2.1.6 the process of DMA data transference requires that the DMA controller be initialised prior to any transfers being carried out.

The winds control software configures the DMA controller to enable DMA transfers from the A/Ds on DMA channel 3 in byte transfer mode. The DMA counter is set to expect the transfer of 200 bytes before requiring re-initialisation. The

memory location of the first byte is defined by the values in variables Page, MSB and LSB and subsequent bytes are automatically written to the next address in memory.

DMA transfers are initialised by the procedure DMASetup with the following lines of Pascal.

```
procedure DMASetup;
begin
with SweepAddresses[SweepNumber] do begin
port[$82] :=Page;                    { Page Reg set}
port[$0c] :=0;                       { Reset the Byte Ptr F/F }
port[$06] :=LSB;                     { Set Base address LSB}
port[$06] :=MSB;                     { Set Base address MSB}
port[$07] :=199;                     { Set DMA Count LSB to 199 }
port[$07] :=0;                       { Set DMA Count MSB to 0 }
end;
```

As each sweep is sampled 100 times at 20 μs intervals, referred to as "range bins", there is only 0.6 ms available for the next sweep of DMA transfers to be initialised (to ensure there are no missed transmitter pulses). Tests[1] have shown that DMASetup takes much less than the available time and no transmitter pulses are being missed.

Initialising A/D Memory

The memory locations for DMA transfers are calculated only once and this is at the beginning of the data collection run by Procedure SetAtoDMemory. This reduces initialisation time between transmitter pulses and prevents any sweeps from being missed. A DMA address table contains the 128 starting addresses that are required to store an echo in memory. These locations are reused for all echoes that are detected during the run. The Pascal type SweepAddr is defined as a record of the DMA start address specifiers Page, Most significant byte (MSB) and Least significant byte (LSB), and SweepAddresses (refer figure 2.17) is an array of SweepAddr holding all the start memory locations for an echo. To facilitate the extraction of data to disk the memory page segment and offset for these DMA start memory locations were assigned at this time as type AtoDLocations. The AtoDLocations for a sweep are

[1]The time to initialise DMA can be measured by raising the voltage on one of the unused bits on the output port at the beginning of DMASetup then lowering the voltage at its completion.

referenced from the array `AtoDMemIRQ`.

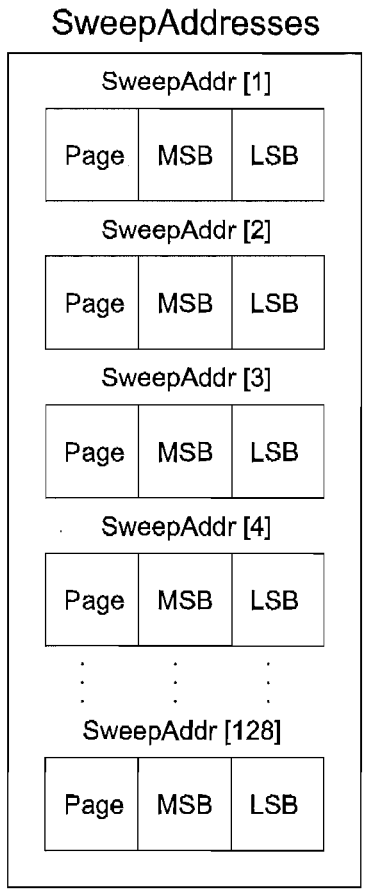

Figure 2.17: The variable SweepAddresses is an array of SweepAddr which contain `Page`, `MSB` and `LSB` values.

The following Pascal Procedure calculates the 128 start memory locations for the DMA transfers.

```
Procedure SetAtoDMemory;
Begin   For sweepnumber := 1 to NoOfSweeps do
   begin
      Getmem(P,200);
      Pseg    := seg(P^);
      Offset := ofs(P^);
      AtoDMem[SweepNumber]:=ptr(Pseg,Offset);
      with AtoDMemIRQ[SweepNumber] do begin
         Segm := Pseg;
         Offsm := Offset;
      end;
      L := seg(P^)*sixteen+ofs(P^);
      with SweepAddresses[sweepnumber] do
         begin
         Page :=  L shr 16;
         LSB  :=  L and $FF;
         MSB  := (L shr 8) and $FF
      end;
   end;
end;
```

2.2.2 Identifying a Meteor Echo in Phase Data

The collection software scans through the 100 range bins recorded per sweep detecting those range bins which contain the ground pulse and echo. The echo and ground pulse phase data are then extracted to be stored on the hard disk.

As echoes are extremely unlikely to be detected in a range bin with a bin number less than #15, only range bins numbered 15 to 100 are inspected for an echo. Likewise the locality of the transmitter array results in a ground pulse which can not be in a range bin later than #14[2]. To locate the range bins containing signal information the signal amplitude in each range bin for the first 25 sweeps is summed.

[2]Although the ground pulse is consistently found in bin #7 it was considered advantageous to search for it as the back-up transmitter may have a different response time to the transmitter trigger pulse and hence the ground pulse bin may change.

Only the first 25 sweeps are used as it is during this time that the amplitude of the echo is likely to be at its largest and hence most distinct from noise.

The range bin above #14, the sum of whose signal amplitudes is the greatest, contains the echo and the range bin below #15 with the largest amplitude sum contains the ground pulse.

2.2.3 Data Storage Format

All echoes for date `yymmdd` detected in hour <hh>(NZST) are written to disk in a file named `Drive:\Data\yymmdd\Wind_<hh>`. The hourly binary files contain many echoes and the 521 bytes which define each echo are described in figure 2.18.

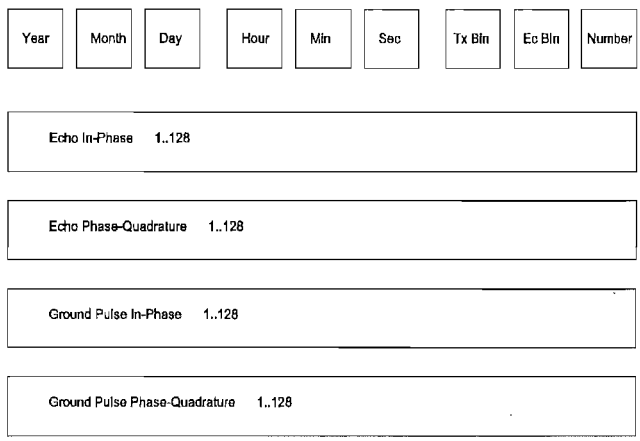

Figure 2.18: The record structure for each meteor echo stored on the AMOR winds PC.

Each record is time-stamped and contains the AMOR defined meteor number (read from the communications port) for matching with AMOR data. The Echo range bin and the ground range bin from which the data are extracted are also written to the echo record.

2.3 Obtaining Wind Velocities

In this work methods have been developed which allow a wind speed to be produced from a meteor echo which has a lifetime as small as 1/40 of a second. As the echo is reflected from a very localised meteor train, any wind measurements made from it can be associated with a small region of atmosphere and the radar's narrow beam

width ($\approx 2^\circ$ FWHM) enables the location of this ionisation to be determined within a region bounded by a box of approximately 8 km^3 (at typical ranges of 200 km).

The location of the measurement is found from the echo range and elevation angle; these are obtained from analysis of the interferometer data recorded in the AMOR NZST_<hh> data files. The horizontal wind velocity is measured by first calculating the line of sight (radial) velocity from the change in echo phase with time (using data in the Wind_<hh> files) and then projecting it on to the horizontal.

System is the name of the programme which does the analysis of AMOR wind data which has been collected at the Birdlings Flat field station.

2.3.1 Determining Radial Velocity

When radar pulses (of frequency f) are reflected from stationary targets they return at the transmitted frequency. If however the reflector is moving then, as a result of the Doppler effect, the frequency of the returning pulse is altered by an amount Δf given by

$$\Delta f = f\left[\frac{1+\beta}{1-\beta}\right] - f. \tag{2.1}$$

Here $\beta = v_s/c$, where c is the speed of light and v_s is the relative speed of separation of the moving target and the detector (β positive for target moving towards observer). For a typical wind speed of 50 m s^{-1} and at the radar's frequency of 26.2 MHz, equation 2.1 indicates the Doppler shift is $\sim$ 9 Hz. From the same expression it is apparent that a velocity precision of 1 m s^{-1} (such precision is required as the speed of atmospheric motion can be of the order of only a few metres per second) requires a frequency resolution of less than 0.2 Hz and this can only be obtained from Fourier analysis if the echo lasts for 5 seconds or more. As a typical meteor echo lasts only a fraction of a second the Fourier spectrum would provide harmonics with a wind speed resolution of >15 m s^{-1} and hence it can not be applied directly to echo data to obtain a precise wind measurement.

Methods of interpolating between spectral harmonics were developed based on the premise that the Fourier spectrum of the echo, when sampled over a short period of time, is the convolution of the data's Fourier spectrum with a sinc function (refer figure 4.1), to give a better estimate of the spectrum's peak. If a is the value of the largest harmonic (at a distance x_0 from the maximum of the sinc function) and b the value of the larger neighbouring harmonic (which is implicitly at a distance

$1 - x_0$ from the maximum of the sinc function as the harmonics are separated by a distance 1) then by putting $\eta = \frac{a}{b}$ it is apparent from figure 2.19 that

$$\eta = \frac{\mathrm{sinc}(x_0)}{\mathrm{sinc}(1 - x_0)}, \tag{2.2}$$

which is equivalent to

$$\eta = \frac{\frac{\sin(\pi x_0)}{\pi x_0}}{\frac{\sin \pi(1 - x_0)}{\pi(1 - x_0)}} \tag{2.3}$$

and rearranging gives

$$\eta = \frac{\sin(\pi x_0)}{\sin \pi(1 - x_0)} \frac{\pi(1 - x_0)}{\pi x_0}, \tag{2.4}$$

which can be simplified to give

$$\eta = \frac{1 - x_0}{x_0} = \frac{1}{x_0} - 1 \tag{2.5}$$

and further rearranging finally gives

$$x_0 = \frac{1}{1 + \eta}, \tag{2.6}$$

and hence a shift $x_0 = \frac{1}{1+\eta}$ from the main harmonic towards the larger neighboring harmonic is a better estimate of the signal frequency. Although this method of harmonic interpolation provided a good method for improving frequency resolution, the use of Fourier analysis still required that the echo exist for a large fraction of a second (more than 0.25 s) and as a large proportion of the meteor echoes recorded by the AMOR system are shorter lived than this many might still be rejected. Naturally, echoes which exhibit a rapid decay then the interpolating function should not be the sinc function but one derived from the envelope of the echo pulses. However it was decided that this approach would suffer the same limitation and was not persued.

The method of analysis used here, which when tested did not have this limitation, measures frequency from the time rate of change in phase over the duration of the echo. With it, a Doppler shift and hence radial velocity can be measured from echoes lasting only a few transmitter pulses. Computationally this method also offers significant speed advantages over using a Fourier transform.

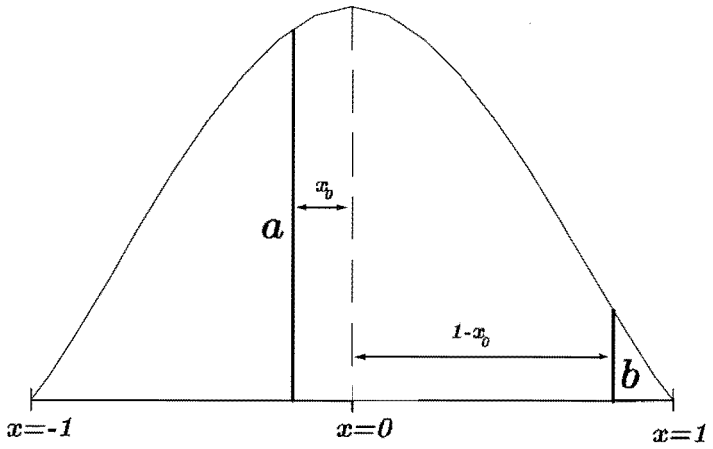

Figure 2.19: The above diagram is part of a sinc function and two harmonics of a spectrum produced from Fourier analysis of finite time evenly sampled data. The true Doppler frequency is at $x = 0$. The sinc function defined as $\frac{\sin(\pi x)}{\pi x}$ equals zero at $x = \pm 1$ where x, in units of frequency, is equivalent to the Fourier harmonic frequency spacing and is determined from $1/\tau$ where τ is the sampling duration.

Radial wind speeds (v_r) can be interpreted from the echoes' rate of change of phase (ϕ) with equation 2.7

$$v_r = -\frac{\lambda}{4\pi}\frac{d\phi}{dt}, \tag{2.7}$$

where

$$\phi = \tan^{-1}\frac{\text{Phase-Quad Comp}}{\text{In-Phase Comp}}. \tag{2.8}$$

This equation can be easily derived from the change in phase of the received signal by noting that the signal shifts by 2π radians for a train drift of $\lambda/2$.

The AMOR receivers directly observe and record the transmitter ground pulse and its phase profile enabling a direct comparison between transmitted and received signals. If ϕ_E and ϕ_T are the echo and transmitter ground pulse phase respectively a modified expression for obtaining radial wind speeds is

$$v_r = \frac{\lambda}{4\pi}\left(\frac{d\phi_E}{dt} - \frac{d\phi_T}{dt}\right), \tag{2.9}$$

providing the advantage that drift in the transmitter frequency does not affect the calculated wind speed. The expression retains the convention of northward

meridional wind speeds having a positive sign for echoes detected in the southern beam only[3].

For a wind speed to be obtained `System` removes the DC level inherited from the A/D conversion process then the phase of each pulse is found for both transmitter ground pulse and echo. Phase data are unwrapped (a process that changes absolute phase jumps that are greater than π to their 2π complement) and the gradients of this unwrapped echo and transmitter data are determined by linear regression (these are the values $\frac{d\phi_E}{dt}$ and $\frac{d\phi_T}{dt}$ of equation 2.9 respectively).

Phase Profile Analysis Window

The phase profile, calculated from equation 2.8, is always 128 transmitter pulses long. Not all data in the profile will necessarily contribute to a wind measurement as data towards the end of the profile may have been recorded when the echo had decayed into noise and data from the beginning of the profile may have been contaminated by reflections off the meteoroid itself[4]. These contaminating data are rejected by only accepting the data within a window which bounds data of steadily changing phase (refer figure 2.20).

The constant phase region is determined by differentiating the phase profile and searching for a region which, on analysis, is judged to have remained constant. This method proved more successful than simply bounding the accepted region to coincide with the echo amplitude maximum and a pre-determined amplitude minimum as the chosen method compensates if the noise level varies.

Tests showed that an accurate measurement can be usually made if the echo duration is longer than 6 transmitter pulses. Echoes shorter than this tended to produce inaccurate results and have large uncertainties associated with them. In practise only echoes which exist longer than 10 pulses were used for a wind measurement as they provided a more reliable data set.

A temporary file called `PhVels.dat` is generated in the `E:\Analysed\yymmdd` subdirectory into which the calculated wind velocities are written.

The algorithm was tested thoroughly with test data to ensure that expected

[3]A sign change is required for echoes detected in the northern beam.

[4]Hence a wind velocity measurement would be contaminated by an aliased $\sim 30\ \mathrm{km\,s^{-1}}$.

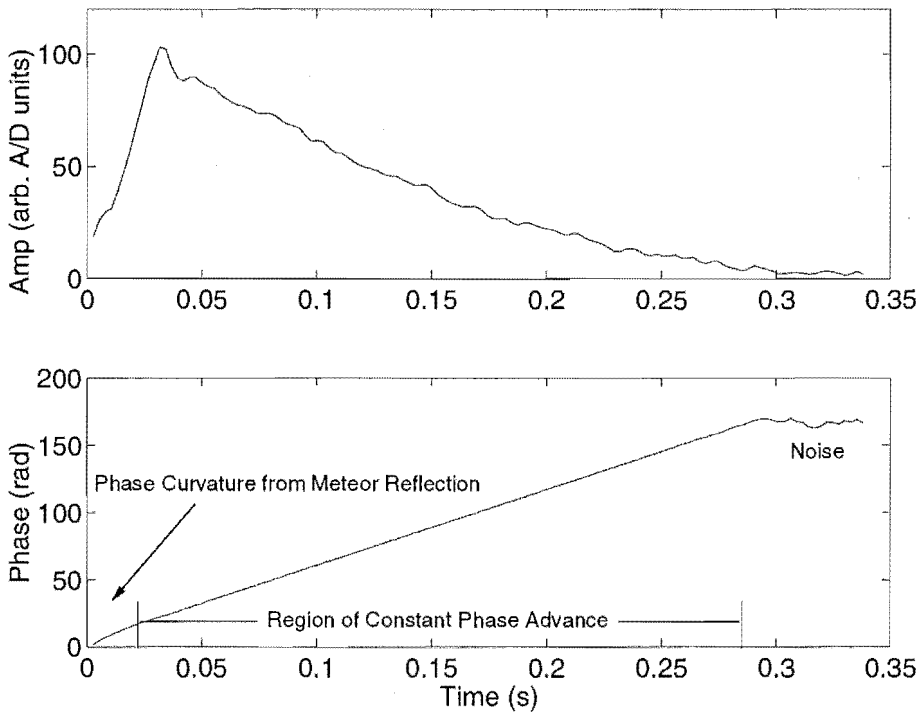

Figure 2.20: Amplitude and phase profile of echo. Measurements of wind speed can only be made during the region of constant phase advance. Careful inspection of the phase during the initial stages of the echo shows curvature of the echo's phase and the phase wanders randomly when the echo amplitude has decayed into noise.

results were obtained. Additionally, the AMOR winds system occasionally observed a persistent meteor and more than one wind measurement was made from it. It could be expected that two or more wind measurements that were spatially and temporally similar should give near-identical results and analysis supported this. Note that in practice only the initial signal from persistent echoes are used in winds as wind shear can deform the train resulting in multiple reflection points which contaminate the measurement[5].

2.3.2 Spatial Location of the Wind Measurement

The fan shaped radiation pattern of the AMOR system confines echoes to $\approx 2°$ (FWHM) in the meridian and this has the advantage that the reflector's position

[5]The speed of expansion of the overdense train is superimposed on the background wind resulting in an error in the wind measurement. However, as few overdense echoes are detected and the induced error is only a few meters per second the effect is minimal.

can be easily determined from its range and elevation angle only. The meteor train's specular reflection point (refer figure 2.21) cannot be measured precisely but rather it can be determined as being within a volume, and at a typical range of 200 km, approximately 80% of echoes would be located in a cell with sides approximately 1 km in height 1 km in latitude and 8 km in longitude (refer figure 2.22). Approximately 97% of echoes (those remaining in the main lobe plus echoes in the first side lobe) would be confined to a 27 km^3 cell.

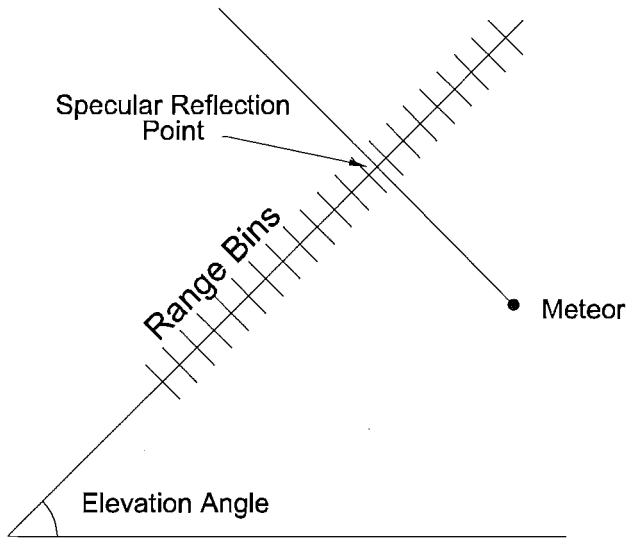

Figure 2.21: The returning pulse is sampled every 40 μs which provides range bins that are 6 km wide. Determining the range bin with maximum echo amplitude provides an estimate of range and analysis of the maximum amplitude and amplitude of neighbouring range bins improves range precision.

The range bins for the AMOR orbit system are 40 μs long providing a coarse initial range measurement with an uncertainty of ± 3 km, however techniques used allow the range to be measured with an accuracy of ± 0.5 km. This improvement is achieved by applying a fitting function to the values in the range bin with the largest echo amplitude and its neighbours to estimate where the true echo maximum is [Taylor 1991].

Elevation angle ϕ, with an uncertainty of 0.3° at a typical value of 30° [Baggaley & Bennett 1996], is measured with a dual interferometer of spacings 3.0 and 10.5 wavelengths (figure 2.23) and is the solution to the expressions

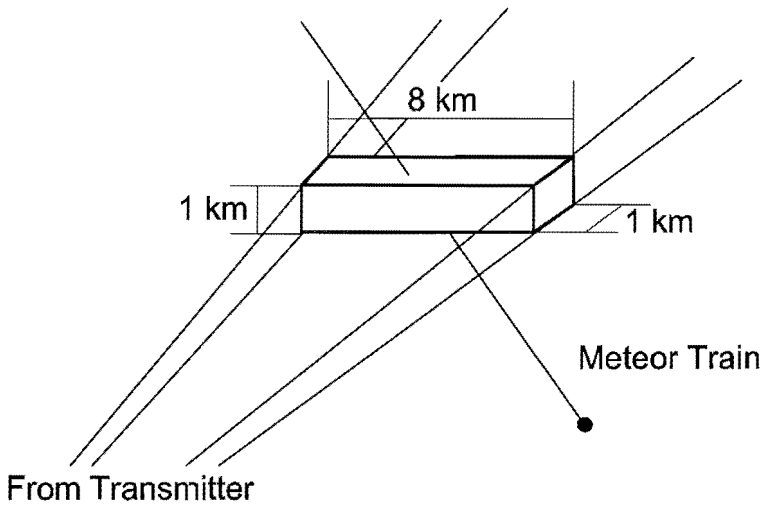

Figure 2.22: AMOR's narrow beam width combined with interferometry enables the location of the echoes specular reflection point to be determined within a 3dB volume of 8 km^3 with the above dimensions.

$$\cos\phi = \frac{n_1 + \frac{\phi_\alpha}{2\pi}}{3.0} \tag{2.10}$$

and

$$\cos\phi = \frac{n_2 + \frac{\phi_\beta}{2\pi}}{10.5}, \tag{2.11}$$

where n_1 and n_2 are integer. The phase differences measured on the 3.0 and 10.5 wavelength interferometers are ϕ_α and ϕ_β respectively.

The lines of zero phase difference, between the two received signals, for each interferometer are shown in figure 2.24 where the light and dark lines represent the 10.5 λ and 3.0 λ spacing interferometers respectively. From this figure it can be seen than a more precise measurement of ϕ can be made with the 10.5 λ interferometer, however it offers many more possible solutions, i.e., values of n_2 which satisfy equation 2.11.

The unique elevation angle solution is achieved by comparison with the possible solutions obtained by the 3.0 λ spacing interferometer. Should an ambiguity still exist, then often the correct solution can be obtained with the a priori knowledge

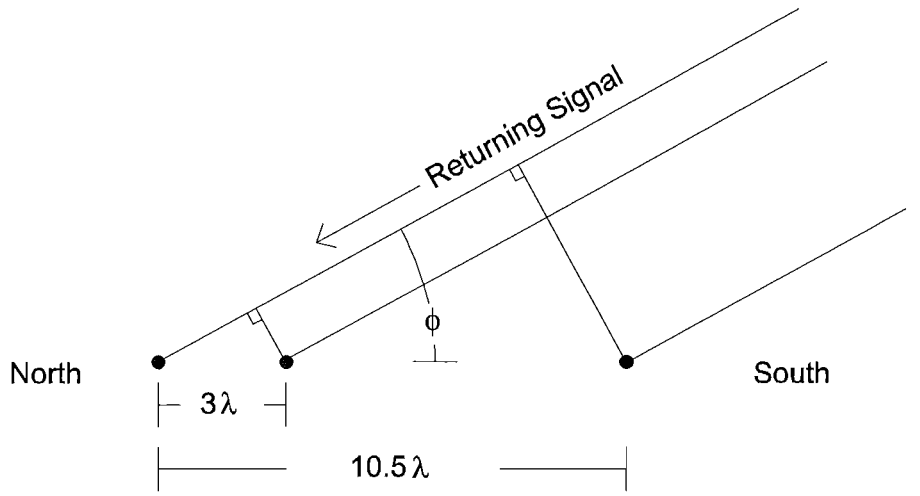

Figure 2.23: The dual interferometer antenna spacings of 3.0 λ adn 10.5 λ as viewed from the west side of the Home site.

that the meteor height must be in the meteor zone.

The direction in which the meteor echo was observed (either north or south) is generally determined from the difference in the relative timing of the echo's detection at the Homes site and the two remote sites. For echoes that were not detected at the remote sites this approach cannot be used. Instead, techniques using the interferometer are used to determine the meteor direction, as well as the elevation angle, of the meteor echo.

Range calibration required the accurate measurement of system delays and was achieved by measuring the time interval between generating a transmitter trigger pulse and receiving the ground pulse. Calibration of the elevation angle is achieved with calibrated cables of known phase delay in conjunction with a signal detector [Baggaley & Bennett 1996]. An overall calibration is also regularly carried out by astronomical means: comparing shower radiant coordinates with those published in catalogues.

A temporary file called `Heights.dat` is generated in the `E:\Analysed\yymmdd` subdirectory into which the calculated meteor elevation angles, ranges and heights are written.

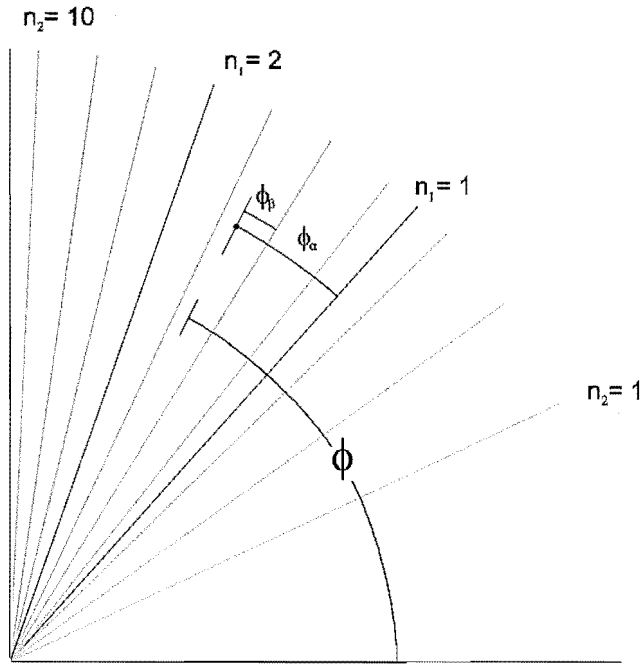

Figure 2.24: The elevation angle for meteor echoes can be unambiguously obtained from a dual interferometer. The elevation angle (ϕ) is the unique solution to the expression: $\cos\phi = \frac{n_1 + \frac{\phi_\alpha}{2\pi}}{3.0} = \frac{n_2 + \frac{\phi_\beta}{2\pi}}{10.5}$.

2.3.3 Combining AMOR and Wind Files

Radial wind speeds in the `PhVels.dat` file are merged with their corresponding spatial locations in the `Heights.dat` file primarily by the meteor's AMOR assigned meteor number. This number has a value between 0 and 255 and is sent from the AMOR observation PC to the WIND's computer via a serial connection included in the WIND echo record. Echoes in both `WIND_hh` and `NZST_hh` files are said to be from the same meteor event if they have the same meteor number and occurred at similar times[6]. A range check is also done to ensure both echoes occurred at similar ranges (in the event of simultaneous echoes at different ranges). Finally the radial velocity (v_r) is projected onto the horizontal to get the horizontal wind velocity (v_h) with the relationship

$$v_h = \frac{v_r}{\cos\phi}, \tag{2.12}$$

[6]The meteor identification number is the key to ensuring the correct matching. As the period for meteor number repetition is large (tens of minutes) the echo's time plays only a minor role.

where ϕ is the elevation angle (refer figure 1.2). The temporary files `PhVels.dat` and `Heights.dat` are replaced with the file `Graphs.dat` in the `E:\Analysed\yymmdd` subdirectory before being deleted. Echoes which exist for fewer than ten transmitter pulses are rejected at this stage as the uncertainties associated with such short-lived echoes tended to be high (2–10 $\mathrm{m\,s^{-1}}$).

2.3.4 Uncertainty in Speed Measurement

The radial velocity is obtained from the following expression

$$v_r = \frac{\lambda}{4\pi}\left(\frac{d\phi_E}{dt} - \frac{d\phi_T}{dt}\right). \tag{2.13}$$

The values $\frac{d\phi_E}{dt}$ and $\frac{d\phi_T}{dt}$ are determined from a linear least squares fit to the phase data and uncertainties for these values are determined by multiplying the residual's standard deviation by the factor $\sqrt{\frac{\chi^2}{N-2}}$ where χ^2 is computed from $\chi^2(a,b) = \sum_{i=1}^{i=n}(\frac{y_i-a-bx_i}{\sigma_i})$ [Bevington 1969]. From standard techniques it can be shown that the uncertainty in the radial velocity is given by

$$\Delta v_r = \sqrt{\Delta\frac{d\phi_E}{dt}^2 + \Delta\frac{d\phi_T}{dt}^2}. \tag{2.14}$$

The horizontal wind velocity v_h is then obtained by projecting the radial velocity onto the horizontal,

$$v_h = \frac{v_r}{\cos\phi}, \tag{2.15}$$

where ϕ is the elevation angle and has an uncertainty of $\sim$0.3°. Therefore the uncertainty in v_h is fully described by

$$\frac{\Delta v_h}{v_h} = \sqrt{\left(\frac{\Delta v_r}{v_r}\right)^2 + (\tan\phi\ \Delta\phi)^2}, \tag{2.16}$$

where, for this expression, $\Delta\phi$ must be expressed in radians.

From equation 2.16 it is apparent that as ϕ tends to 90° the uncertainty in v_h increases dramatically. Horizontal wind velocity uncertainties for a horizontal velocity of 40 $\mathrm{m\,s^{-1}}$, obtained from a radial velocity with an uncertainty of 1 $\mathrm{m\,s^{-1}}$ are shown in figure 2.25.

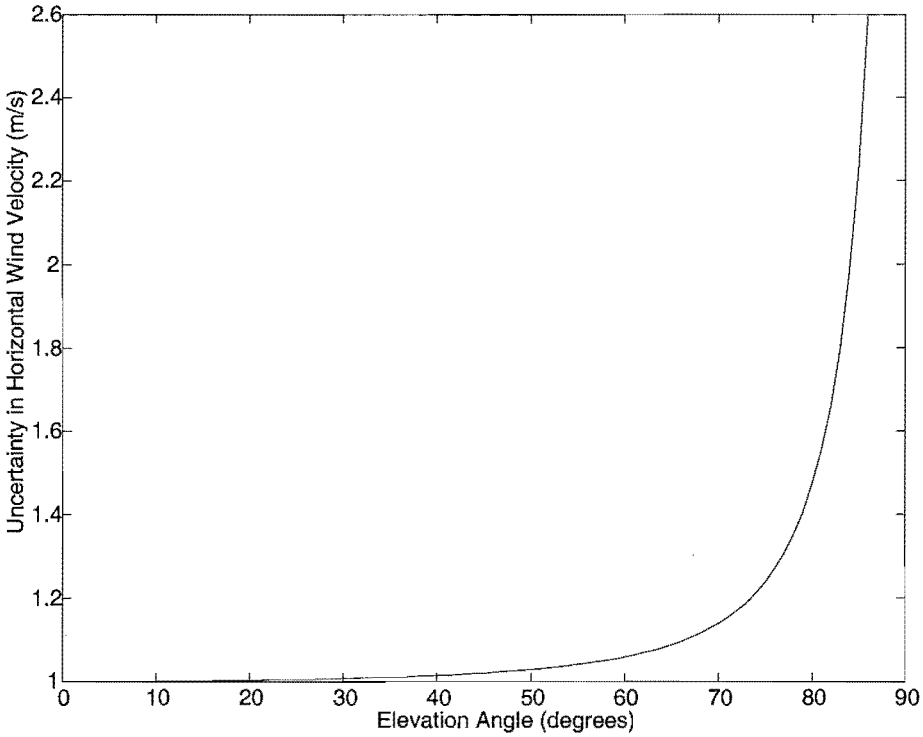

Figure 2.25: The uncertainty in the derived horizontal wind varies as a function of elevation angle. This graph displays trend and values for a calculated horizontal velocity of 40 $\mathrm{m\,s^{-1}}$ obtained from a radial velocity with an uncertainty of 1 $\mathrm{m\,s^{-1}}$. Typically the AMOR meteor radar detects meteor echoes from elevation angles between 15 and 55 degrees.

2.3.5 Errors Introduced by Assumptions

All Echoes From Within the Main Beam

From a theoretical antenna radiation pattern of the AMOR system (essentially a sinc function) it can be estimated that approximately 10% of echoes detected will be outside of the main beam. As AMOR has no means, i.e. an azimuthal interferometer, for identifying these echoes they are treated as if they were detected in the main lobe introducing a wind speed error (wind speed magnitudes of echoes in the side lobes will be consistently underestimated). If the measured wind speed v_m is actually at an angle θ in azimuth with respect to the centre of the main beam then the true wind speed, v_t, could be found from the equation

$$v_t = \frac{v_m}{\cos\theta}. \tag{2.17}$$

The difference between v_m and v_t has been expressed (as a percentage of v_m) in figure 2.26 for the first four side lobes. As virtually no echoes are detected out-

side the second side lobe, the largest introduced error is only 0.6% and considered insignificant.

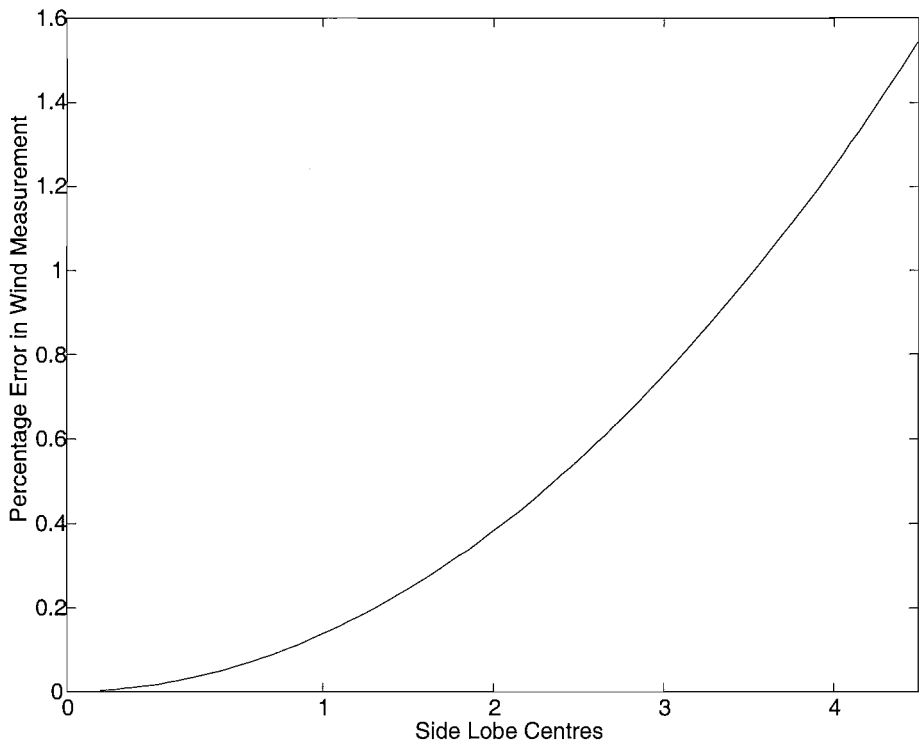

Figure 2.26: All meteors are assumed to have been detected in the main beam. Wind measurements from meteors outside this beam contain percentage errors given by this graph. For example the absolute value of wind speed measured from meteors detected in the centre of the first side lobe are underestimated by 0.13%.

No Vertical Wind

Analysis throughout this work has assumed no vertical wind [Smith 1998]. However studies have indicated the presence of a vertical component of the wind field at meteoric heights which can occasionally be as large as 2–3 $\mathrm{m\,s^{-1}}$[Mitchell & Howells 1998]. As a result, measurements of wind directed away from the radar during large vertical velocity episodes are either overestimated (when the vertical motion is upward) or underestimated when the motion is downward. For winds directed toward the radar the reverse applies.

Referring to the upper diagram of figure 2.27 if the true wind vector, v_{t1} or v_{t2}, contains a horizontal component v_{ht} and a vertical component v_{vt} (directed upwards or downwards) then the radial velocity observed by the meteor radar at elevation

angle ϕ is given by v_r and is simply the projection of the wind vector onto the line of observation. The method adopted in this work then projects the measured radial velocity on to the horizontal and obtains horizontal components v_{h1} during upward moving vertical winds and v_{h2} during downward propagating vertical winds.

From the bottom diagram of figure 2.27 it is apparent that for an echo detected at an elevation angle ϕ the difference (δ in units of $\mathrm{m\,s^{-1}}$) between the true horizontal velocity v_{ht} and the interpreted horizontal velocity v_{h1} (or v_{h2}) in the presence of a vertical wind v_{vt} is given by

$$\delta = \frac{v_{vt}}{\tan(90 - \phi)}. \tag{2.18}$$

Figure 2.28 shows the dependence of δ on ϕ and v_v for values of v_v between 0.5 $\mathrm{m\,s^{-1}}$ and 3 $\mathrm{m\,s^{-1}}$. The error introduced can be seen to climb rapidly for larger elevation angles. For a typical elevation angle of 30° and a large vertical velocity of 3 $\mathrm{m\,s^{-1}}$ δ equates to a value of 1.7 $\mathrm{m\,s^{-1}}$.

This value provides an indication of the upper limit for the uncertainty associated with horizontal wind speeds from individual echoes of 3 $\mathrm{m\,s^{-1}}$.

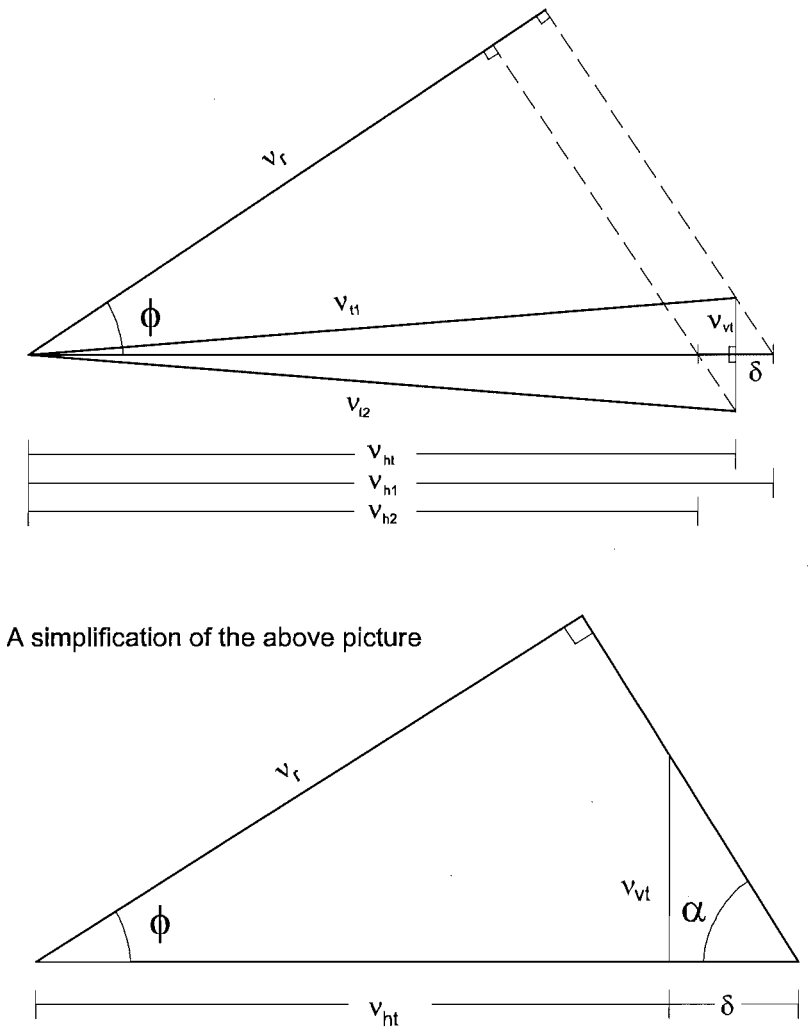

Figure 2.27: Analysis for the horizontal component of the wind assuming no vertical motion incorrectly estimates the true horizontal component v_{ht} in the presence of a vertical component v_{vt} by an amount δ as shown in the diagram. In the case when v_{vt} is directed upwards and the wind direction is away from the radar the obtained horizontal velocity v_{h1} is overestimated and conversely for v_{vt} downwards v_{h2} is underestimated. For wind towards the radar the opposite applies. ϕ is the echo's elevation angle.

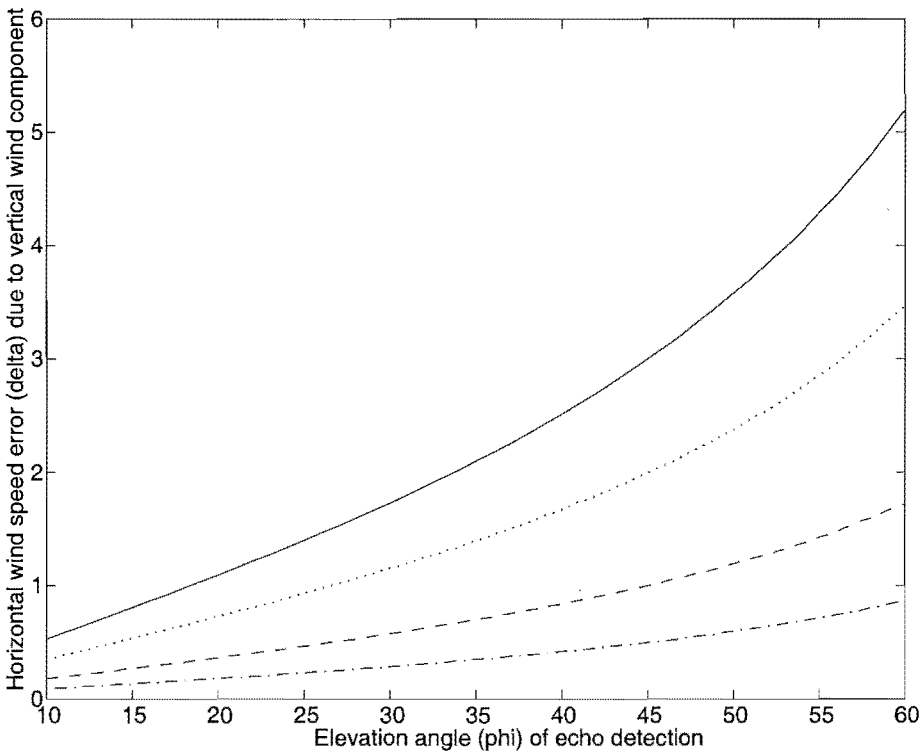

Figure 2.28: The response of the error induced into horizontal winds at elevation angles 10–60 degrees due to the assumption that no vertical winds (v_{vt}) exist for v_{vt} = 3 $\mathrm{m\,s^{-1}}$(solid line), 2 $\mathrm{m\,s^{-1}}$ (dotted line), 1 $\mathrm{m\,s^{-1}}$(dashed line) and 0.5 $\mathrm{m\,s^{-1}}$(dash dot).

Chapter 3

The Drift of a Meteor Train in the Earth's Atmosphere

The meteor train is only useful as a wind tracer if the electron train movement is directly indicative of the wind's motion. This chapter discusses the train's motion under the influence of a magnetic field and background wind. It is shown that the train movement is directly indicative of the wind at heights below 100 km. Above 100 km the introduced errors, due to the train drifting in a direction orthogonal to the wind, become increasingly larger.

An algorithm is derived which, it is proposed, can correct for errors introduced by the orthogonal drift. This algorithm requires that both zonal and meridional wind measurements are made and hence has not been applied to the AMOR data set.

3.1 Parallel and Orthogonal Drift Components

At meteoric heights particles in the meteor train are subjected to the influence of a weak electric field $\mathbf{E}$, a geomagnetic field, $\mathbf{B}$, and friction due to collisions, $m\nu\mathbf{v}$. The general motion of a particle of charge q and mass m in a background wind $\mathbf{u}$ can be described by the relation

$$m\left(\frac{d\mathbf{v}}{dt}\right) = q\left(\mathbf{E} + \mathbf{v} \times \mathbf{B}\right) + m\nu\left(\mathbf{v} - \mathbf{u}\right), \tag{3.1}$$

where ν is the particle's collision frequency. Forces on these particles accelerate them until the increased friction produces equilibrium at a velocity, $\mathbf{v} = (v_x, v_y, v_z)$.

The meteor train is a quasi-neutral weak plasma and external $\mathbf{E}$ fields have negligible effect. Should charge separation within the train occur, an internal $\mathbf{E}$ field is generated which opposes this separation. Thus the meteor train undergoes

ambipolar diffusion and forces on the particles from the **E** field are negligible and neglected in any further analysis.

Particle drift is collision dominated when $|q(\mathbf{v} \times \mathbf{B})| \ll m\nu v$, or, assuming **B** and **v** orthogonal

$$\frac{q\mathrm{B}}{m} \ll \nu. \tag{3.2}$$

The quantity on the left hand side of the above expression is called the particle's gyrofrequency and denoted ω.

At equilibrium equation 3.1 equates to zero and Kaiser et al. [1969] show that electrons, moved by a wind of velocity **u**, would drift so the components parallel, $v_{\parallel}$, and orthogonal, $v_{\perp}$, to the wind vector are given by

$$v_{\parallel} = \frac{\mathrm{u}}{1 + \dfrac{\omega^2}{\nu^2}} \tag{3.3}$$

and

$$v_{\perp} = \frac{\mathrm{u}\dfrac{\omega}{\nu}}{1 + \dfrac{\omega^2}{\nu^2}}. \tag{3.4}$$

As meteor trains experience ambipolar diffusion the train will drift at a rate which is controlled by the heavier ions. Only a few *ms* after formation the ions and electrons are at similar temperatures, so, if one neglects any Coulomb interaction, the ratio, β, of ion diffusion D_+ to electron diffusion D_- will be height independent,

$$\beta = \frac{D_+}{D_-} = \frac{m_- \nu_-}{m_+ \nu_+}, \tag{3.5}$$

and the components for meteor train drift will be given by

$$v_{\parallel} = \frac{1}{1 + \beta^2 \dfrac{\omega^2}{\nu^2}}\,\mathrm{u} = \varepsilon_{\parallel}\,\mathrm{u} \tag{3.6}$$

and

$$v_{\perp} = \frac{\beta \dfrac{\omega}{\nu}}{1 + \beta^2 \dfrac{\omega^2}{\nu^2}}\,\mathrm{u} = \varepsilon_{\perp}\,\mathrm{u}, \tag{3.7}$$

where ω and ν are for electrons.

Kaiser et al. [1969] determine a value for β from experimental measurements of ν_- [Hall & Bullough 1963], temperatures from the CIRA standard atmosphere [Kallman-Bijl et al. 1961] and D_+ from observations of meteor echo train decay [Greenhow & Hall 1960]. The dimensionless quantity, β, was shown to be 3.1×10^{-4}.

The height dependence of $\varepsilon_\parallel$ and $\varepsilon_\perp$ for January (solid line) and July (dashed line) are shown in figure 3.1, the seasonal difference being due to the height at which particular electron collision frequencies occur decreasing in winter. The electron collision frequencies were derived from a model fit to CIRA 86 data [von Biel 1995].

Wind measurements are likely to incur error if either $v_\parallel$ is not representative of the wind speed or $v_\perp$, of particles blown by a wind perpendicular to the line of observation, is large.

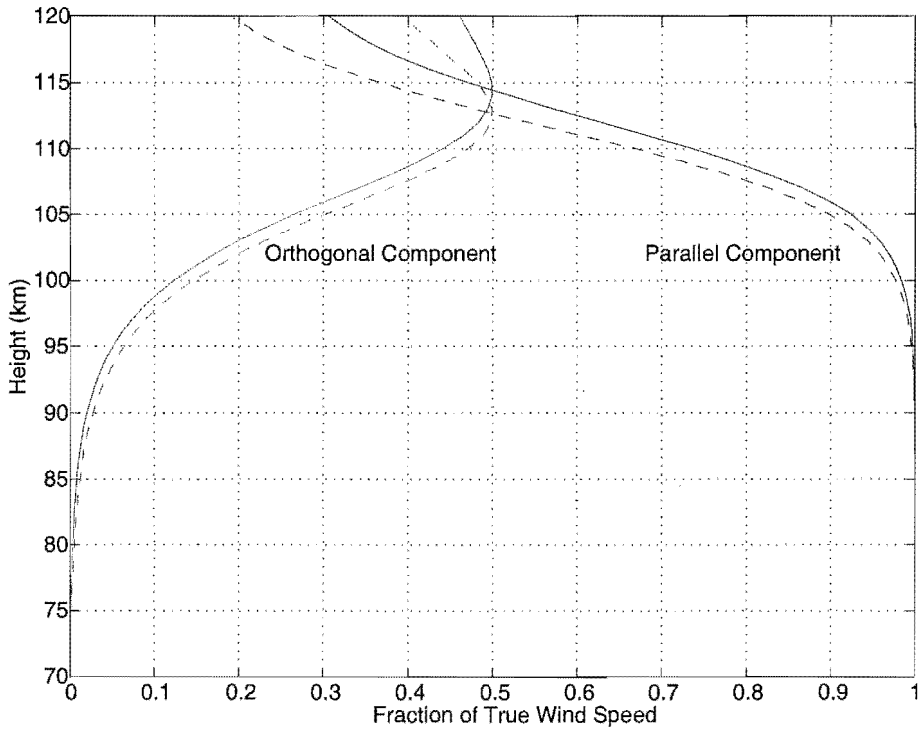

Figure 3.1: The magnetic field has an increasing influence on the motion of a blown meteor train with increasing height. This graph shows that the component of the train parallel to the direction of the background wind, $\varepsilon_\parallel$, is approximately unity at heights up to 90 km. Between 90 and 105 km, $\varepsilon_\parallel$ reduces to 0.9. The component of the train velocity orthogonal to the wind vector, $\varepsilon_\perp$, increases from 0 at heights above about 75 km and by 105 km has a value of approximately 0.25. This graph details the worst possible scenario of perpendicular wind and magnetic field vectors.

As $\varepsilon_\parallel$ and $\varepsilon_\perp$ assume $\mathbf{u}$ and $\mathbf{B}$ orthogonal, for $\mathbf{u}$ and $\mathbf{B}$ at an angle Ψ, $\varepsilon_\perp$ needs to

be multiplied by $\sin\Psi$ and $\varepsilon_{\|}$ needs to be multiplied by $\eta(\Psi) = 1+(1/\varepsilon_{\|}-1)\cos\Psi$. The angle between the meridional component of a wind field and the magnetic field is denoted α where from figure 3.2

$$\alpha = \cos^{-1}(\cos\theta\cos\sigma). \tag{3.8}$$

The angle between the zonal component and the magnetic field is denoted γ and again from figure 3.2 this is seen to have a value of

$$\gamma = \cos^{-1}(\cos\theta\sin\sigma). \tag{3.9}$$

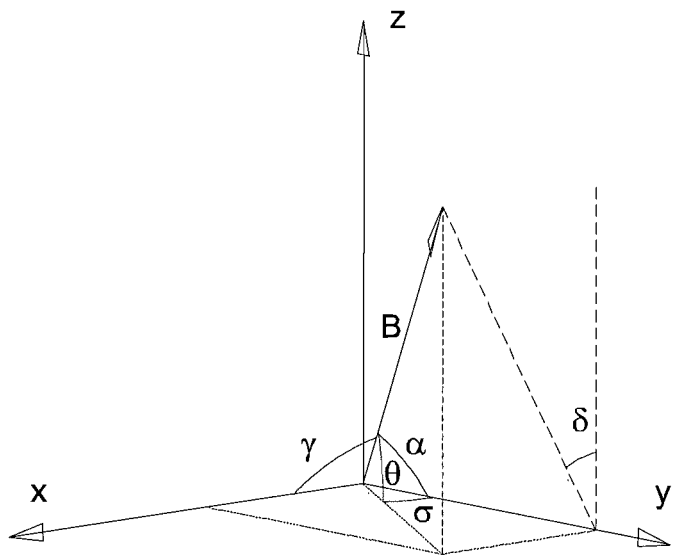

Figure 3.2: The magnetic field line vector **B** is at angles θ with respect to the x-y plane and σ with respect to the y-z plane. The angle between the meridional component (aligned with the y-axis) of the wind vector and **B** is shown on the diagram as α. The angle between the zonal component (aligned with the x-axis) of the wind vector and **B** is shown on the diagram as γ. The vector orthogonal to both the meridional component and the magnetic field is at an angle δ to the x-axis.

Decomposing a general wind vector in to meridional, u_y, and zonal, u_x, components allows consideration of each independently. Considering the meridional component first, the motion of the train, due to this component only, v_{mm}, is given by

$$v_{mm} = \eta(\alpha)\varepsilon_{\|}u_y. \tag{3.10}$$

As the radar observes the train's drift at an angle, ϕ_m, called the elevation angle (where m indicates a meridional observation and z a zonal observation), the drift observed by the radar is its projection on to the line of observation. The line of sight drift is given by

$$v_{mm} = \eta(\alpha)\varepsilon_{\parallel} \mathrm{u}_y \cos\phi_m. \tag{3.11}$$

A meridional component in a northward direction will generate an orthogonal component, v_{zm}, to the train's drift which moves eastward and downwards, while a southward meridional component will cause a drift which is westward and upwards. The direction of $\mathbf{u}_y \times \mathbf{B}$, is at an angle, δ, to the horizontal where δ (refer figure 3.2) is given by

$$\delta = \tan^{-1}\left[\frac{\sin\sigma}{\tan\theta}\right]. \tag{3.12}$$

From figures 3.3 and 3.4 the line of sight observation of this drift is given by

$$v_{zm} = \varepsilon_{\perp} \mathrm{u}_y \sin\alpha \cos\left(\phi_z \pm \delta\right), \tag{3.13}$$

where the sign in the brackets is '$-$' for a westward observation and '$+$' for an eastward observation.

When the train is blown by wind with a zonal component, the expression for the parallel drift of the train due to this component, v_{zz}, is given by

$$v_{zz} = \eta(\gamma)\varepsilon_{\parallel} \mathrm{u}_x \cos\phi_z. \tag{3.14}$$

The orthogonal drift of the train due to an eastward component is upwards and towards the south. For a westward component the drift is downwards and towards the north. Synonymous with equation 3.13 the magnitude of this motion is $\varepsilon_{\perp} \mathrm{u}_x \sin\gamma$. As can be seen from figures 3.5 and 3.6 the influence of the train's drift on the measurement velocity is different for each observation direction. The line of sight orthogonal drift velocity of the zonal component is given by

$$v_{mz} = -\varepsilon_{\perp} \mathrm{u}_x \sin\gamma \sin\left(\alpha \pm \phi_m\right), \tag{3.15}$$

where the sign in the brackets is '$-$' for a northward observation and '$+$' for a southward observation. This equation has a negative sign to comply with the convention that northward and eastward winds have positive sign.

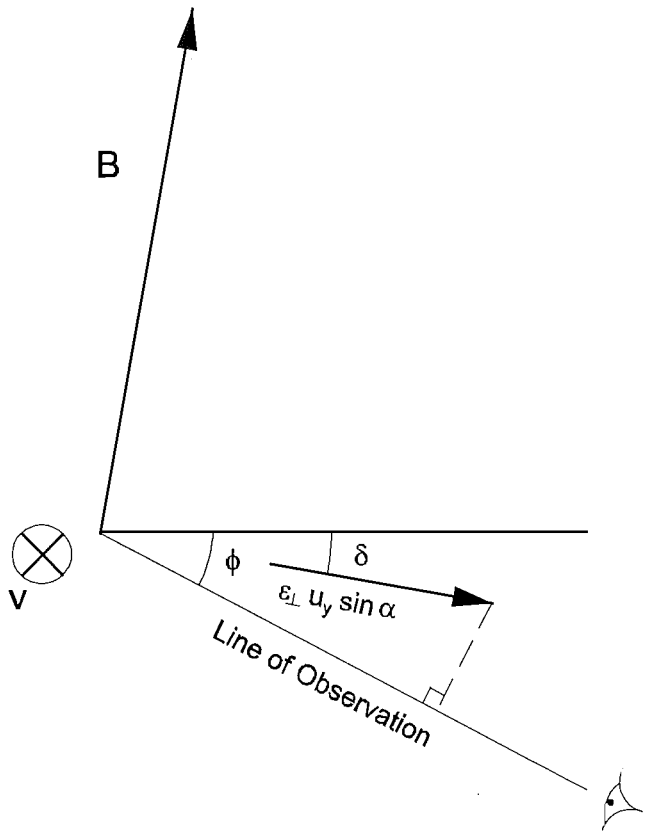

Figure 3.3: A northward wind (into the page) component will cause the train to move in the direction of $\mathbf{u}\times\mathbf{B}$ with magnitude $\varepsilon_\perp u_y \sin\alpha$ thereby altering the radial velocity of the meteor train. The magnitude of this introduced error is given by the projection of the vector $\varepsilon_\perp u_y \sin\alpha$ onto the line of observation which is inclined at an angle ϕ. A southward wind would introduce an error of the same magnitude but opposite direction. This diagram details the geometery for an echo observed in a westward direction.

3.2 Correction Algorithm

It is apparent from equations 3.13 and 3.14 that the measured (line of sight) zonal component, u_M, is a function of the true zonal component, u_T, and the true meridional component, v_T, by the relation

$$u_M = \eta(\gamma)\varepsilon_\parallel u_T \cos\phi_z + \varepsilon_\perp v_T \sin\alpha\cos(\phi_z \pm \delta), \tag{3.16}$$

and from equations 3.11 and 3.15 the measured (line of sight) meridional component, v_M, is a function of v_T and u_T by the following relation

$$v_M = \eta(\alpha)\varepsilon_\parallel v_T \cos\phi_m - \varepsilon_\perp u_T \sin\gamma\sin(\alpha \pm \phi_m). \tag{3.17}$$

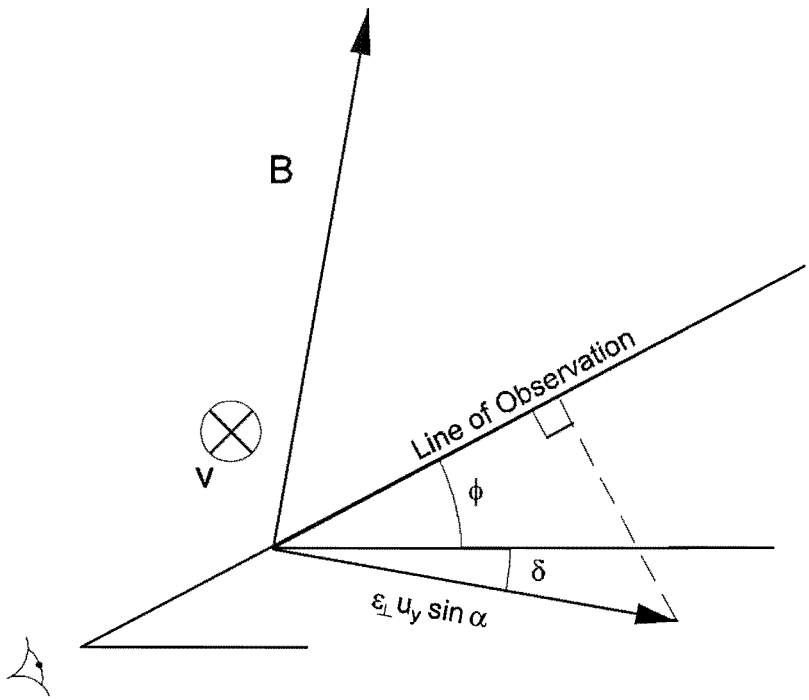

Figure 3.4: Same as for previous diagram however this details the geometery for an echo observed in an eastward direction. Again if the zonal wind was directed out of the page the magnitude of $\varepsilon_{\perp} u_y \sin\alpha$ projected onto the line of observation would remain the same, however its direction changes by 180°.

From equations 3.16 and 3.17 it can be seen that

$$\frac{u_{MH}}{u_T} \propto \frac{v_T}{u_T} \tag{3.18}$$

and

$$\frac{v_{MH}}{v_T} \propto -\frac{u_T}{v_T}, \tag{3.19}$$

where v_{MH} and u_{MH} are the horizontal projection of the observed meridional and zonal components respectively.

To illustrate the effect of the wind field's zonal component on the measured meridional component, figure 3.7 shows simulations of the ratio of measured and true meridional values as a function of the ratio of true zonal to true meridional values. The influence of a meridional component on measurements of a zonal component are displayed in figure 3.8. In this simulation values for the orientation of the Earth's magnetic field have been extracted from the geomagnetic reference field model (IGRF)[1]. The value used for the inclination of the Earth's magnetic field over Christchurch, θ, is 68° at an angle of $\sigma = 22°$ to geographic north .

[1]Program which generate magnetic field values are available to the public from the World Wide

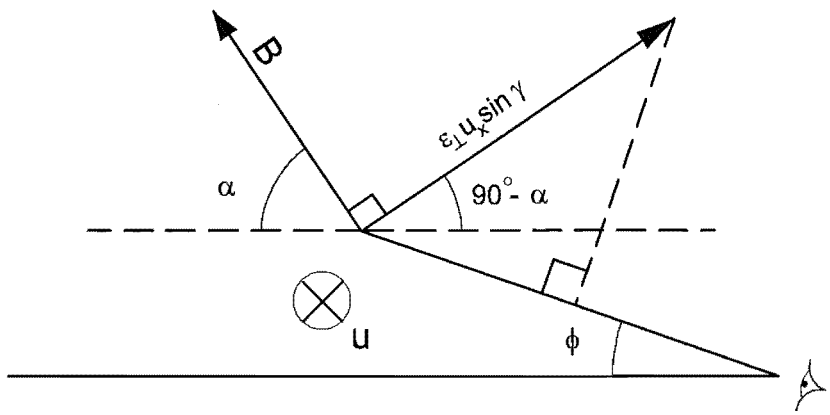

Figure 3.5: An eastward wind (into the page) component will cause the train to move in the direction of $\mathbf{u}\times\mathbf{B}$ with magnitude $\varepsilon_\perp u_x \sin\gamma$ thereby altering the radial velocity of the meteor train. The magnitude of this introduced error is given by the projection of the vector $\varepsilon_\perp u_x \sin\gamma$ onto the line of observation which is inclined at an angle ϕ. A westward wind would introduce an error of the same magnitude but opposite direction. This diagram details the geometery for an echo observed in a northward direction.

Considering the effect of the zonal component on the measured meridional component first, only a small effect is apparent at 90 km and that is only when zonal and meridional components are of the same sign and the zonal component is of comparatively large amplitude. Increasing altitude sees this effect become more pronounced. Large amplitude zonal winds with the same sign as the meridional wind cause the measured meridional component to be underestimated. Conversely when the signs are opposite, the meridional component is measured too large.

The meridional wind influence on measured zonal winds displays a similar height behaviour. However, for this component, comparatively large amplitude meridional winds of same or opposite sign cause the measured zonal component to be overestimated or underestimated respectively.

The following simultaneous solution to equations 3.16 and 3.17 suggest that wind speed components obtained from a meteor observed simultaneously from orthogonal

Web server of the World Data Center at
http://www.ngdc.noaa.gov/seg/potfld/geomag.html

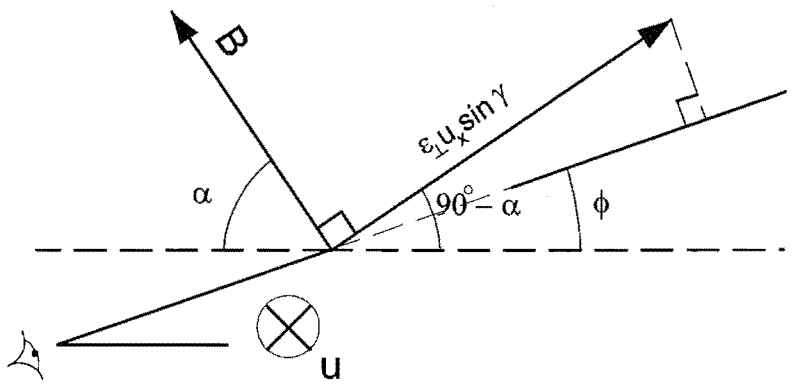

Figure 3.6: Same as for previous diagram however this details the geometery for an echo observed in a southward direction. Again if the zonal wind was directed out of the page the magnitude of $\varepsilon_\perp u_x \sin\gamma$ projected onto the line of observation would remain the same, however its direction changes by 180°.

directions (refer figure 3.9) could be corrected to account for magnetic field effects.

$$u_T = \frac{\eta(\alpha)\varepsilon_\parallel \cos\phi_m u_M - \varepsilon_\perp \sin\alpha \cos(\phi_z \pm \delta) v_M}{\eta(\gamma)\eta(\alpha)\varepsilon_\parallel^2 \cos\phi_m \cos\phi_z + \varepsilon_\perp^2 \sin\gamma \sin\alpha \cos(\phi_z \pm \delta) \sin(\alpha \pm \phi_m)} \tag{3.20}$$

and

$$v_T = \frac{\eta(\gamma)\varepsilon_\parallel \cos\phi_z v_M + \varepsilon_\perp \sin\gamma \sin(\alpha \pm \phi_m) u_M}{\eta(\gamma)\eta(\alpha)\varepsilon_\parallel^2 \cos\phi_m \cos\phi_z + \varepsilon_\perp^2 \sin\gamma \sin\alpha \cos(\phi_z \pm \delta) \sin(\alpha \pm \phi_m)}, \tag{3.21}$$

which can be expressed in matrix form

$$\begin{bmatrix} u_T \\ v_T \end{bmatrix} = \frac{1}{\Gamma} \begin{bmatrix} \eta(\alpha)\varepsilon_\parallel \cos\phi_m & -\varepsilon_\perp \sin\alpha \cos(\phi_z \pm \delta) \\ \varepsilon_\perp \sin\gamma \sin(\alpha \pm \phi_m) & \eta(\gamma)\varepsilon_\parallel \cos\phi_z \end{bmatrix} \begin{bmatrix} u_M \\ v_M \end{bmatrix}, \tag{3.22}$$

where $\Gamma = \eta(\gamma)\eta(\alpha)\varepsilon_\parallel^2 \cos\phi_m \cos\phi_z + \varepsilon_\perp^2 \sin\gamma \sin\alpha \cos(\phi_z \pm \delta) \sin(\alpha \pm \phi_m)$.

Such observation conditions are difficult to obtain and therefore the full benefit of this approach may seldom be achieved. Nevertheless, intuition suggests that improved hourly wind velocity averages may be obtained, from the application of this correction algorithm, by meteor radar systems which observe in both meridional and zonal directions (refer figure 3.10).

To test this theory a simple Monte-Carlo simulation was used. Components, v_T and u_T, at a height of 110 km, were randomly generated with values between

0 ± 40 m s^{-1}. Meridional and zonal elevation angles, ϕ_m and ϕ_z, associated with these wind measurements were randomly assigned values between 15 and 60°. For each pair of zonal and meridional components, equations 3.16 and 3.17 predict the radar line of sight measured velocities, v_M and u_M. Calculating averages of v_M and u_M, for n of these measurements, provides the simulated hourly wind speed averages generally calculated from individual measurements. These averages are then compared with the average values of v_T and u_T as an indicator of good or poor agreement.

The corrected values of v_M and u_M are then obtained by applying equation 3.22 to hourly averages v_M, u_M, ϕ_m and ϕ_z (and the remaining values appropriate for a height of 110 km). These were also compared with v_T and u_T.

Graphs showing the distribution of the variation from the true values (error) are shown in figure 3.11. From these it is immediately apparent that the spread in the error is greatly reduced when the correction is applied. This is confirmation that the correction algorithm can be applied to hourly averaged values in an effort to reduce the effect of magnetic field induced train drift at greater heights.

At the time of writing this thesis the AMOR radar system measured the meridional component only and hence the correction algorithm could not be applied[2]. In response to the results presented in this section, wind speed measurements presented in this thesis are confined to heights less than 105 km.

3.3 Mutual Influence by Zonal and Meridional Semidiurnal Tides

As has been discussed, the meteor radar observed meridional wind velocity is affected by zonal winds at large heights. It was also shown that the relative affect was proportional to the ratio of the true zonal and meridional velocities.

Section 1.2 states that the semidiurnal tide is, in general, the most prominent feature of mid-latitude wind measurements. The phase of the zonal semidiurnal tide leads the meridional tide by near 3 hours and consequently, the times of maximum zonal motion due to the semidiurnal tide coincide precisely with zero velocities in

[2]Work has just begun constructing an orthogonal array which will eventually provide zonal wind measurements. It will also enable the testing/application of this method.

the meridional component of that tide. The effect, as a function of height, of a zonal semidiurnal tide on an equal amplitude meridional semidiurnal tide is displayed in figure 3.12.

As expected measured values are most affected at greatest heights. There is virtually no effect at 90 km, however as height increase from 100 to 110 km, the profiles show an increasing change in the phase of the tide; this can be explained from work detailed earlier. Regions where the zonal component is non-zero, and of the same sign as the meridional component, will have meridional wind speeds which are underestimated. Conversely, when the zonal component is of opposite sign to the meridional component the meridional component will be overestimated. The measured and true wind speeds are most similar when the ratio of the true meridional and zonal velocities is greatest. It is also apparent that the measured amplitude is virtually unaffected.

The effect of a meridional semidiurnal tide on the, same amplitude, zonal semidiurnal tide is shown in figure 3.13. Again the phase is shown to be delayed at greater heights: however this time, the amplitude is also shown to reduce.

If the amplitude of one of the components is much larger than the other then it has a greater influence on the other tide and both the phase difference and measured amplitude become larger.

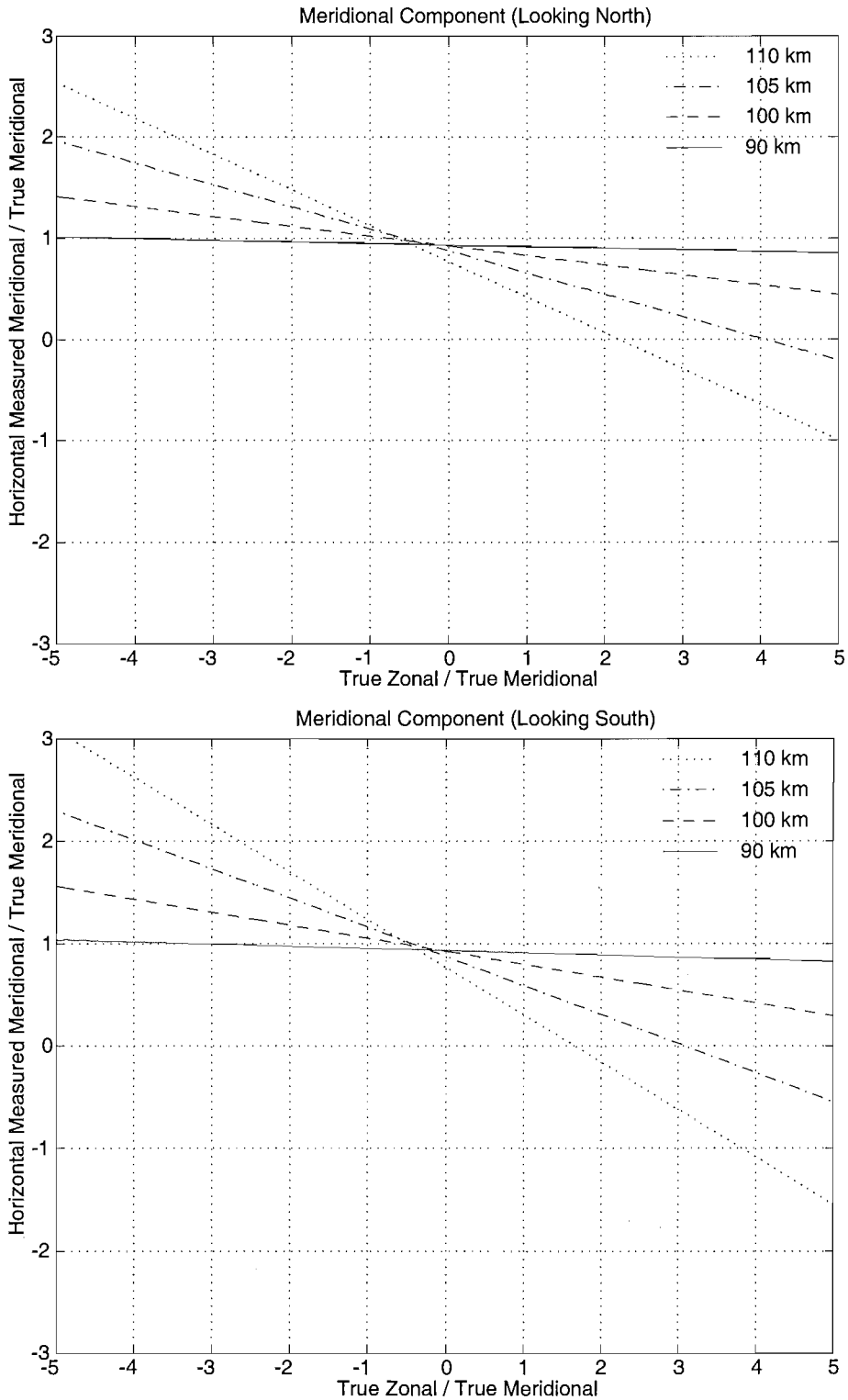

Figure 3.7: The simulated ratio of measured and true meridional values are graphed as a function of the ratio of true zonal to true meridional values. In this model the elevation of the Earth's magnetic field, θ, is 68° and it is at an angle of 22° to geographic north.

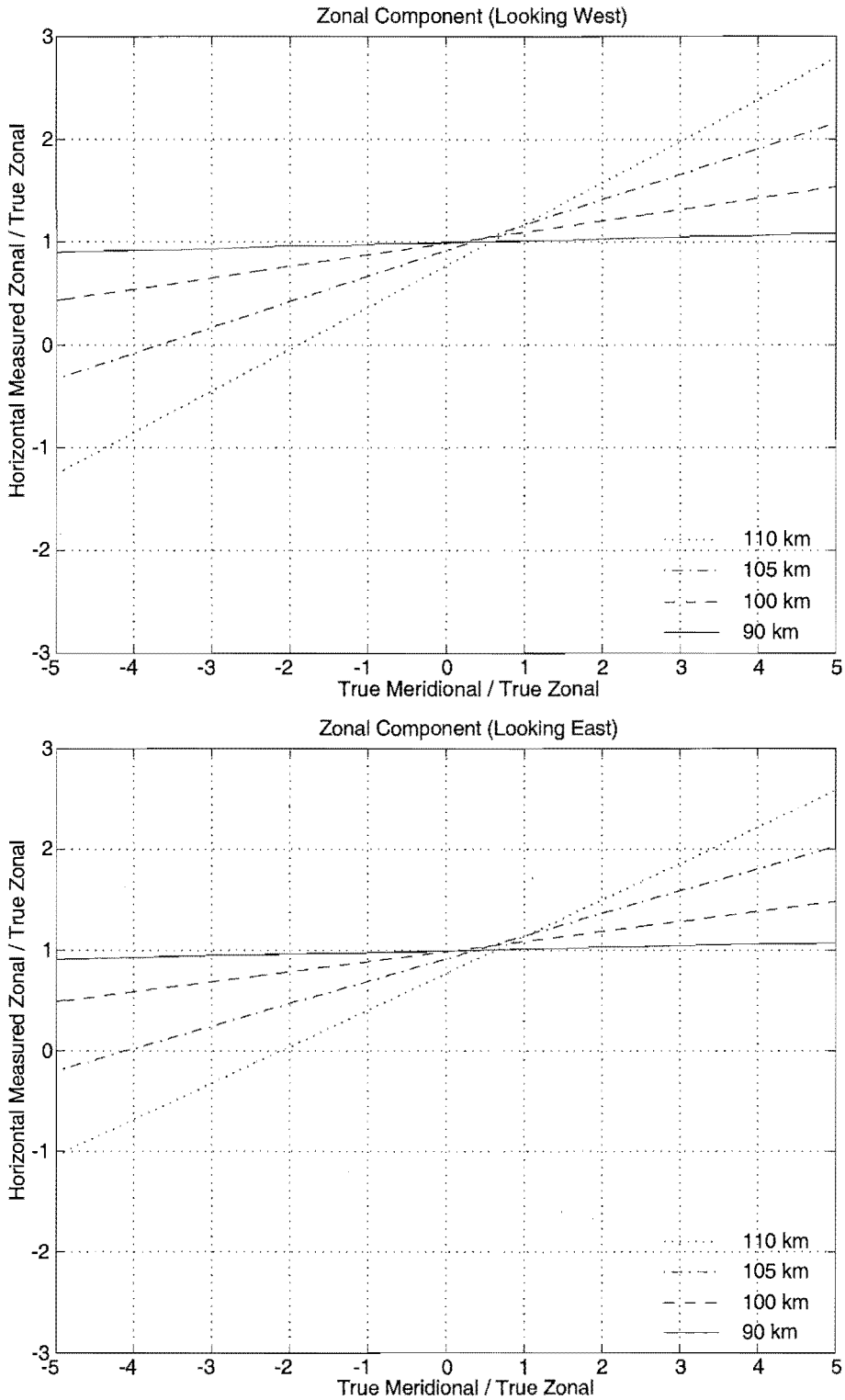

Figure 3.8: The simulated ratio of measured and true zonal values are graphed as a function of the ratio of true meridional to true zonal values. In this model the elevation of the Earth's magnetic field, θ, is 68° and it is at an angle of 22° to geographic north.

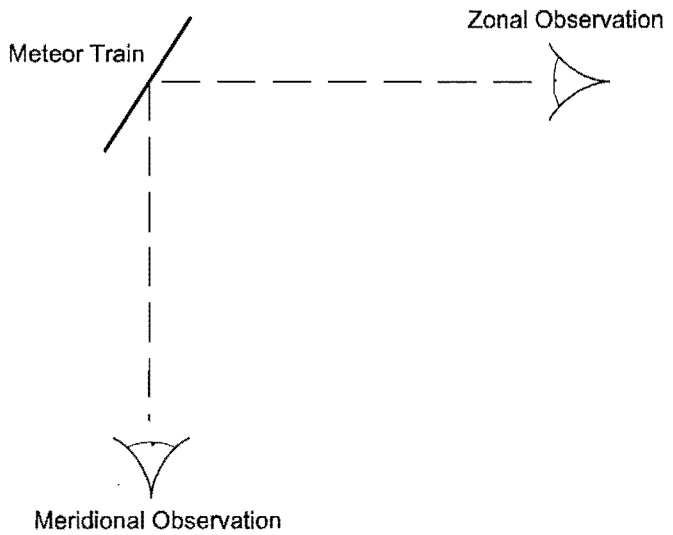

Figure 3.9: The simultaneous observation of the same meteor train from orthogonal stations would enable the calculation of corrected wind component velocities at large heights.

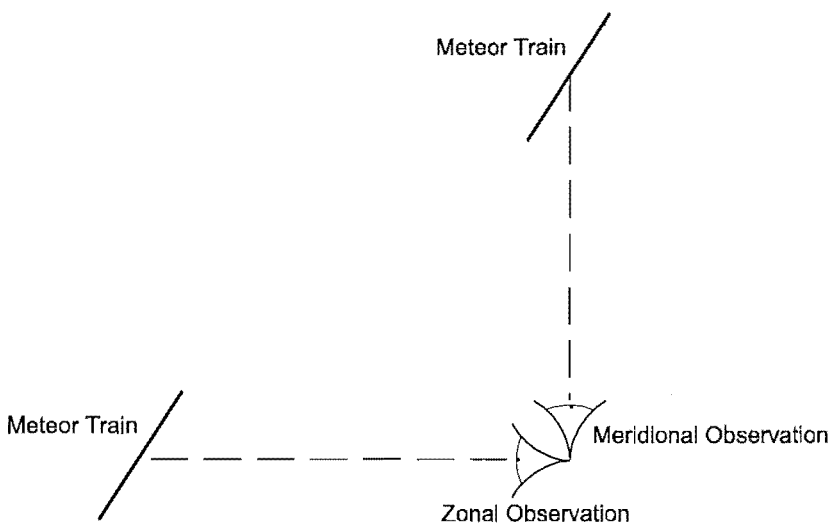

Figure 3.10: In general, meteor radars which measure both zonal and meridional components make these measurements by sampling orthogonal regions of space.

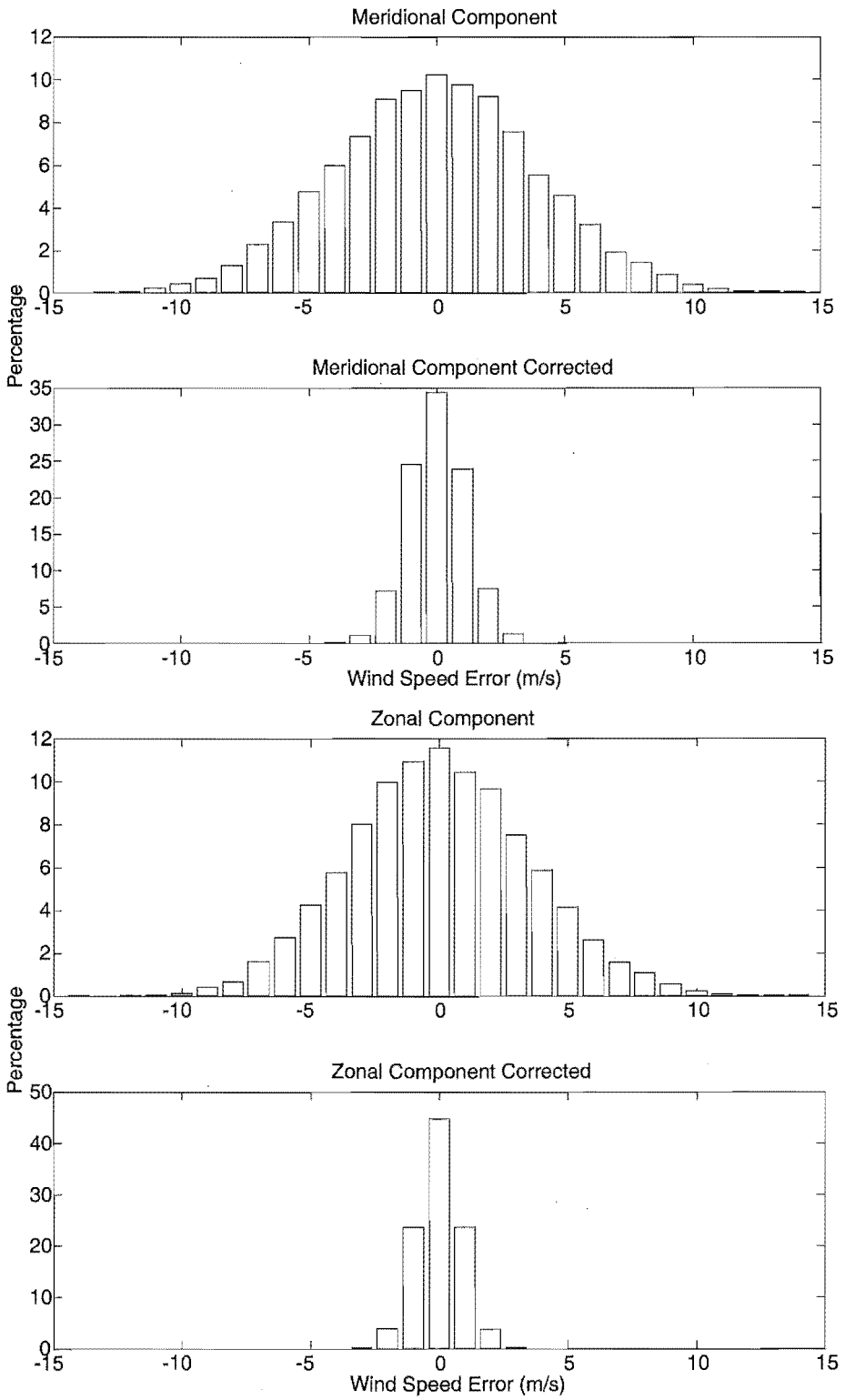

Figure 3.11: Simulated velocity measurements for meridional and zonal winds components at 110 km show a spread about the true wind speed due to magnetic field effects. A technique described in the text reduces this spread considerably.

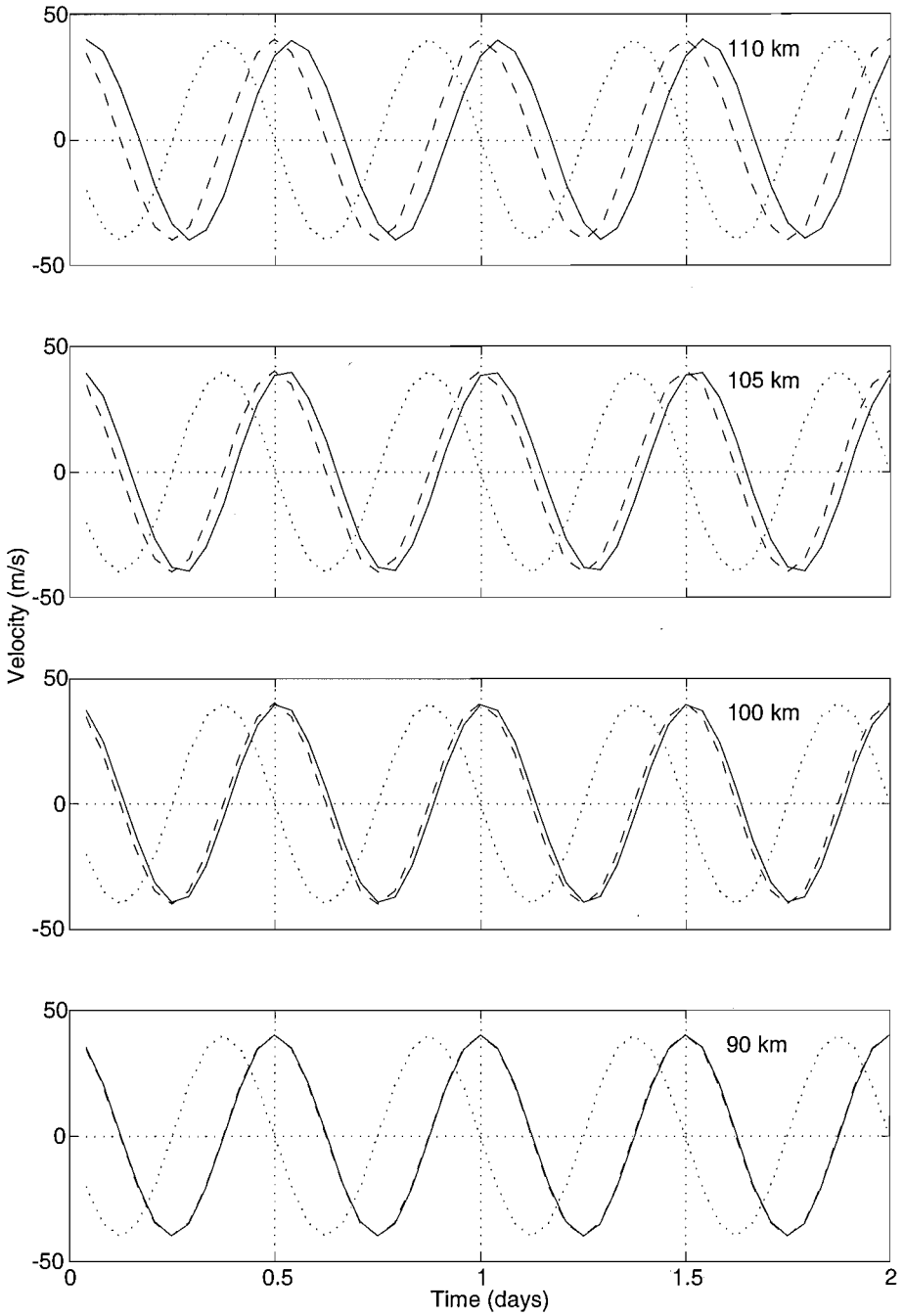

Figure 3.12: The effect of the zonal component of the semidiurnal tide on the meridional semidiurnal tide. The dashed line shows the undisturbed wind velocities, describing a 12 hour period semidiurnal tide. The solid line shows the wind velocities one would expect to measure if the measurements were being affected by a zonal semidiurnal tide of equal amplitude. The dotted line indicates the zonal semidiurnal tide profile.

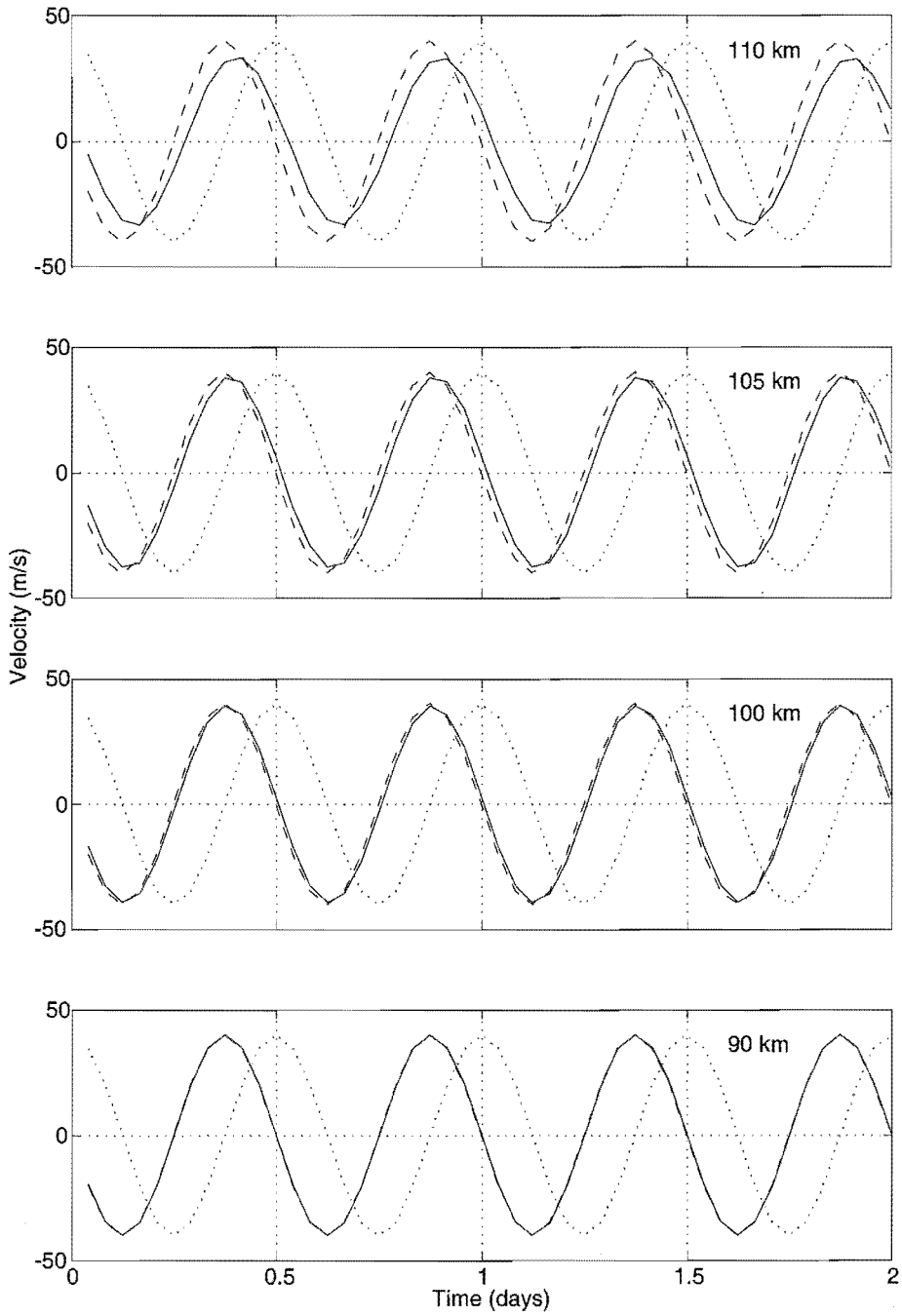

Figure 3.13: The effect of the meridional component of the semidiurnal tide on the zonal semidiurnal tide. The dashed line shows the undisturbed wind velocities and the solid line shows the wind velocities one would expect to measure if the measurements were being affected by a meridional semidiurnal tide of equal amplitude. The dotted line indicates the meridional semidiurnal tide profile.

Chapter 4

Time Series Analysis

Harmonic analysis is the process of transforming data from the time to the frequency domain, and as the atmosphere's motion contains many features which are periodic (refer section 1.2.1), is an important tool for analysing the AMOR winds data set.

4.1 Data Preparation for Harmonic Analysis

For analysis the wind data are grouped according to temporal and spatial location and the dimensions of these data bins are varied depending on the atmospheric parameters being studied.

With the AMOR system it is beneficial to group data which are either similar in height yet varied in ground range, or similar in ground range yet varied in height. The first of these two schemes is applied in this work when searching for phenomena that are height critical, such as the vertical wavelength of the semi-diurnal tide, and the second approach is used when observing ground range variations, such as the influence of orographic gravity waves on wind measurements.

The accuracy with which range and elevation angle can be determined by the AMOR system enables the height and ground range of echoes to be found to the nearest kilometre, thus setting a lower limit on spatial resolution. Although the event time for individual meteors can be measured accurate to seconds, harmonic analysis is applied to hourly mean velocities primarily as this reduces any bias introduced into the harmonic analysis caused by the uneven distribution of meteor events across each day. The mean wind speed is then associated with the mean event time of meteors during that hour, as opposed to associating the mean wind speed with the middle of the hour.

Hourly averaging also reduces any high frequency (<1 hr) variation in the wind

speed measurements which is considered noise in tidal analysis. For this purpose no benefit was observed by reducing the averaging interval, however it may prove better to perform 10 (or so) minute averages when using the AMOR data set for studying gravity wave (high frequency) processes.

4.2 Fourier Analysis

An approach for extracting any periodic nature in the meteor winds data set is to apply Fourier analysis to the data. However this method cannot be directly used as the Discrete Fourier transform (DFT, or FFT) requires that the data are both complete and regularly spaced. To fulfil these requirements it is possible to construct an equivalent day which consists of 24 hourly averaged values from the data period. If gaps still exist in the data then linear interpolation across the gap is acceptable; however should more than one hour be vacant then other methods, which are discussed in this chapter, may need to be considered.

4.3 Lomb-Scargle Periodogram and Least Squares

A combination of the Lomb-Scargle Periodogram and Linear Least Squares techniques enables the extraction of harmonic amplitude and phase from data which are unevenly spaced. The Lomb-Scargle Periodogram [Scargle 1982] has been widely used in astronomy and identifies the dominant period in the data while linear least squares calculates the amplitude and phase of the period found.

Linear least squares provides a method of summarising a given set of observations (x_i, y_i) by fitting it to a model of the form

$$y(x) = \sum_{k=1}^{m} a_k X_k(x), \tag{4.1}$$

that depends on adjustable parameters a_k where $X_k(x)$ are m specified functions called basis functions which, for harmonic analysis, are sines and cosines. The parameters of the model are adjusted to achieve a minimum in the sum of squared differences between measured and modelled values thereby obtaining a best fit estimate for the values a_k.

Once the dominant frequency component of the data has been determined by

the Lomb-Scargle Periodogram and parameterised with linear least squares, it is extracted from the data and this modified data set is then re-searched for the next dominant component. This process is continued while periodicities are being found at the 95% confidence level. Having found all significant frequencies, a best fit linear least squares process is applied to the original data (along with a mean value) to obtain the best estimates of the tidal parameters.

4.4 The Lomb-Scargle Fourier Transform

The Lomb-Scargle Fourier Transform (LSFT) [Scargle 1989] provides a further method of obtaining a Discrete Fourier Transform (DFT) for data which are unevenly spaced. Unlike the Lomb-Scargle Periodogram, amplitude and phase information can be obtained directly from its spectrum. Like the DFT, frequencies analysed by this process are integer multiples of the fundamental frequency, however for reasons of speed, slight modifications were made to the LSFT algorithm so that it analyses a pre-determined set of frequencies only, producing an algorithm called LSFT2 (detailed in Appendix A). This modification was only a minor adjustment, the frequencies for the harmonics returned by LSFT are effectively pre-calculated depending on the time duration over which the data were sampled and the number of data points. LSFT2 relies on the user supplying these frequencies in advance. Naturally, when using LSFT2 care needs to be exercised to ensure that only frequencies which would have been members of the frequency set calculated by LSFT are requested.

4.5 Sliding Window

If any of the above harmonic analysis techniques were applied directly to a time series of data, the returned spectra would only detail the average amplitudes and average phases of the Fourier components, and the presence of any short lived (albeit possibly large amplitude) oscillations may go undetected. A sliding window (or moving average) is an effective method for observing such short duration events, and also provides a record of the changing tidal parameters present in the time series.

The sliding window method selects only a fraction of the available data (centered at a time t_0) for harmonic analysis and the spectrum returned from the data subset

is likely to be a more accurate representation of the conditions around t_0 than the spectrum obtained from the complete data series. By sliding the window so it is centered at different values of t_0 different spectra are calculated, and an examination of the varying spectra can be used to detail changes in tidal amplitude and phase with time.

Because returned spectra contain average harmonics within the window, its length is an important consideration. The window length must be chosen to contain an integer multiple of the periods of interest (c.f. Fourier analysis which only selects integer multiples of the fundamental frequency) otherwise amplitudes for the harmonics returned will be incorrectly estimated.

Shorter windows (around 10–30 days) provide good resolution for observing short term data fluctuations such as the changing semidiurnal tide amplitude but the presence of any long period planetary waves will remain undetected possibly influencing the mean value. Conversely longer windows (approximately 60 days) are necessary for observing large planetary waves [Beard et al. 1999], but details concerning the short period oscillations will be reduced. Naturally a window 10 days wide cannot resolve fluctuations with periods larger than 10 days, however Vial [1989] suggests that tidal analysis on data within shorter length windows may contain effects which are non-global (gravity waves etc.) and also that for a climatological study 30 day (monthly) long windows are most appropriate.

The presence of frequencies within the data window which are not an integer multiple of the window length tend to spread power throughout the spectrum. This can be appreciated by considering an infinite time series of data points which are sampled from a signal of single frequency f with amplitude A. After harmonic analysis, the spectrum, from $f = 0$ Hz to the Nyquist frequency, would consist of a single delta function at frequency f with amplitude A. However, harmonic analysis of a finite (τ seconds) section of these data would produce a spectrum which is the convolution of the infinite series transform with a sinc function (defined as $\frac{\sin(\pi x)}{\pi x}$, see figure 4.1) which has a maximum at $x = 0$ and nulls at frequencies which are integer multiples of $1/\tau$.

From this it can be seen that if the signal frequency f is an integer multiple of $1/\tau$ then convolving the two spectra will result in all power being at that frequency's component of the signal spectrum, if however f does not equal an integer multiple of $1/\tau$ then the transform of the signal will produce a delta function which lies between

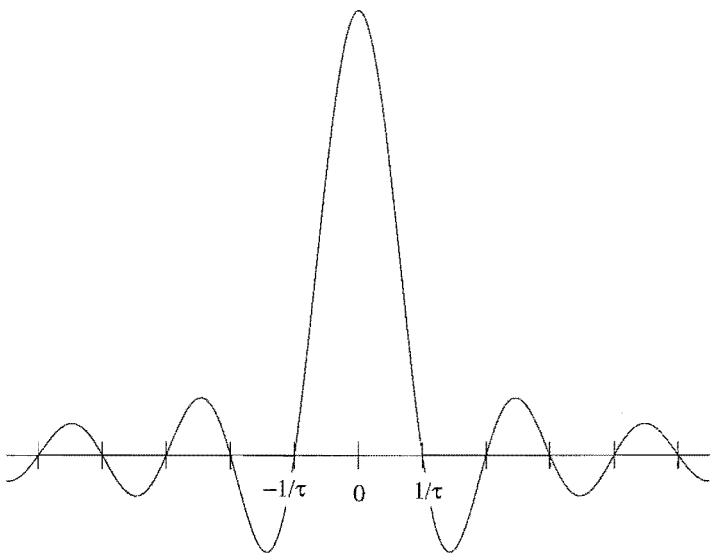

Figure 4.1: The sinc function as defined in section 4.5 has a maximum at $x = 0$ and nulls at frequencies which are integer multiples of $1/\tau$.

nulls of the sinc function and convolution which will spread power throughout the spectrum.

The power deposited into incorrect harmonics can be reduced by applying, to the finite data series, a window function which has the property of reduced side lobes in the frequency domain. For this work a tapered cosine window has been applied to winds data. The window has a cosine lobe width of $(\alpha/2)N$ where N is the number of data points and α a value between 0 and 1. (This function evolves from a rectangular to a Hanning window as α varies from 0 to 1). In this work $\alpha = 0.5$ (refer figure 4.2) was selected providing reduced sidelobes, the first three of which are 15 dB, 25 dB and 35 dB down on the main lobe [Harris 1978].

Sliding the tapered cosine window across the wind data in hourly steps and performing harmonic analysis on these data at each step produces a series of spectra which can be inspected for tidal variation.

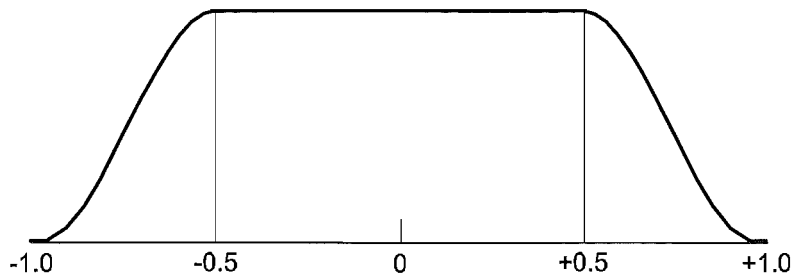

Figure 4.2: The tapered cosine window applied to data prior to harmonic analysis to reduce power incorrectly appearing in harmonics. The ends are tapered in the shape of a cosine hence its name . The region of tapering displayed is for an α value of 0.5.

4.6 Comparing the Analysis Methods

A comparison of the harmonic analysis techniques (Lomb-Scargle Fourier Transform, Fourier Transform and Lomb-Scargle Periodogram-Least Squares) was performed by generating a model function which contained a tidal period of 12 hours with an amplitude which varied throughout the year, along with a diurnal component and a random (noise) value. The data were gradually deteriorated in steps by randomly removing 240 data points. The data were then analysed for harmonic components before removing a further 240 points. This process was continued until effectively all data were removed. By correlating the tidal amplitudes returned by the three harmonic analysis techniques with the known amplitude values which were entered into the simple model, as a function of missing data points, an indication of how well each method performed was obtained.

Figure 4.3 displays results from this test and indicates that all methods produced calculated amplitude values which correlated well (coefficient > 0.8) with actual values at the 95% confidence level provided more than 50% of the data points were available for analysis.

The Lomb-Scargle Periodogram and least squares technique produced correlation coefficients slightly higher than the other two methods, especially when the number of removed data was between about 40 and 80%, however computation speed for this method was very slow.

Although the equivalent day method calculated amplitudes which correlate well

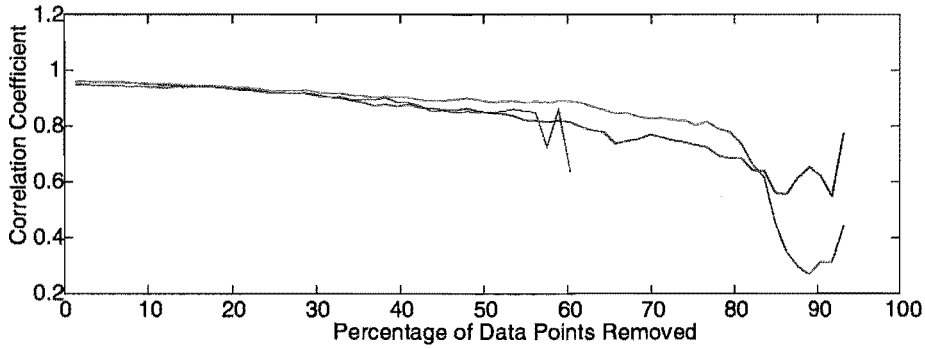

Figure 4.3: A comparison showing how well various harmonic analysis techniques perform as increasing amounts of data are removed from a time series. The red trace is for the Lomb-Scargle Fourier Transform, green is for the Lomb-Scargle Periodogram and Least Squares technique and the blue line arises from a Fast Fourier Transform on an equivalent day.

with those of the other methods when the data was reasonably complete, it failed altogether when the number of removed data increased beyond about 60%. This, combined with the fact that this technique does not resolve periods which are greater than one day[1] meant that this method was not used in this work.

The Lomb-Scargle Fourier Transform technique performed consistently well. Although it performed slightly poorer than the Lomb-Scargle Periodogram and least squares technique in the 40 and 80% removed data region, the algorithm ran extremely quickly and speed was an important consideration. Hence this technique was utilised, for this work, as the preferred technique for harmonic analysis. Nevertheless on occasion the Lomb-Scargle Periodogram and least-squares method was considered to have advantages over the Lomb-Scargle technique and the thesis indicates when this is so.

Based on the poor performance of all techniques when the majority of data has been removed, tidal parameters are only calculated if fewer than 70% of the data are missing.

[1]Although perhaps an equivalent week could be used.

4.7 LSFT Harmonics Uncertainties

As has been discussed, the majority of harmonic analysis performed in this work uses the Lomb-Scargle Fourier Transform (LSFT or LSFT2) based on the algorithm given in Scargle [1989]. This section details the derivation of uncertainties for the returned parameters which arise due to uncertainties in the input hourly mean values. Uncertainties for the input values are the standard error of the mean for time and velocity in each time-height bin. These values are propagated through the LSFT algorithm providing uncertainties for the amplitude and phase of each harmonic.

The following uncertainty calculations have been derived using the standard approach to find the uncertainty, $\Delta f(x,y)$, of a function $f(x,y)$ due to uncertainties in both x and y, given by

$$\Delta f(x,y) = \sqrt{\left(\frac{\partial f}{\partial x}\right)^2 \Delta x^2 + \left(\frac{\partial f}{\partial y}\right)^2 \Delta y^2}. \tag{4.2}$$

Uncertainty analysis of the LSFT routine (where the variables in this analysis are defined in Scargle [1989] and the C translation upon which this analysis is done has been provided in Appendix A) gives the uncertainty in the amplitude, ΔA, of a harmonic to be

$$\Delta A = \sqrt{\frac{ftrp^2 \Delta ftrp^2 + ftip^2 \Delta ftip^2}{ftrp^2 + ftip^2}}, \tag{4.3}$$

with a phase uncertainty ($\Delta\phi$) of

$$\Delta\phi = \sqrt{\frac{ftip^2 \Delta ftrp^2 + ftrp^2 \Delta ftip^2}{(ftrp^2 + ftip^2)^2}} + \ldots \tag{4.4}$$

$$\sqrt{\left[\left(\frac{\Delta wrun}{wrun}\right)^2 + \left(\frac{\Delta tzero}{tzero}\right)^2\right] phase^2 + \Delta wtnew^2}. \tag{4.5}$$

Equations 4.3 and 4.5 have been derived through the definition of a variable *work* which is the complex exponential defining the harmonic at the shift point[2]

[2]A time shift of τ is applied to ensure the data are time invariant, i.e., shifting the time origin would produce spectra which are unchanged except for a phase factor. Lomb [1976] also introduces the shift parameter, τ from a least-squares approach to ensure that cosine and sine terms are orthogonal at the frequency of interest.

and *tmp* which describes the position to which the shift is made. To obtain the correct phase for the particular harmonic it is necessary to transform from the shift point back to the origin and this is achieved by multiplying *work* by *tmp*. If *work* and *tmp* are defined as $(W \pm \Delta W)\, e^{i(\phi_W \pm \Delta\phi_W)t}$ and $e^{i(\phi_T \pm \Delta\phi_T)t}$ respectively then the product can be expressed as

$$(W + \Delta W)e^{i(\phi_W + \phi_T + (\Delta\phi_W + \Delta\phi_T))t}. \tag{4.6}$$

From this expression it can be seen that the harmonic amplitude uncertainty is solely dependent on ΔW and the harmonic's phase uncertainty is given by the sum $\Delta\phi_W + \Delta\phi_T$. To obtain the value of ΔW it is convenient to revert back to complex notation in which *work* is defined as:

$$work = (ftrp + \Delta ftrp) + i(ftip + \Delta ftip), \tag{4.7}$$

where $ftrp$ means Fourier Transform Real Part etc. The amplitude of *work*, W, is then given by $\sqrt{ftrp^2 + ftip^2}$ and its uncertainty can be shown to be

$$\Delta W = \sqrt{\frac{ftrp^2 \Delta ftrp^2 + ftip^2 \Delta ftip^2}{ftrp^2 + ftip^2}}, \tag{4.8}$$

and the uncertainty in phase due to uncertainties in calculating *work* with equation 4.7 can also be shown to be

$$\Delta\phi_W = \sqrt{\frac{ftip^2 \Delta ftrp^2 + ftrp^2 \Delta ftip^2}{(ftrp^2 + ftip^2)^2}}. \tag{4.9}$$

The uncertainty in the phase at the shift point is given by

$$\Delta\phi_T = \sqrt{\left[\left(\frac{\Delta wrun}{wrun}\right) + \left(\frac{\Delta tzero}{tzero}\right)^2\right] phase^2 + \Delta wtnew^2}, \tag{4.10}$$

however this expression simplifies to $\Delta\phi_T = \Delta wtnew$ when using the LSFT2 algorithm as both $\Delta wrun$ and $\Delta tzero = 0$. The more complicated expression arises as a result of the calculation of harmonic frequencies in LSFT.

The uncertainties in the real and imaginary components of each harmonic ($\Delta ftrp$ and $\Delta ftip$) are obtained from the following two calculations:

$$\Delta ftrp = const1 \sqrt{\left(\frac{\Delta sumr^2}{scos2}\right) + \left(\frac{sumr^2}{4scos2^3}\right) \Delta scos2^2} \tag{4.11}$$

and

$$\Delta ftip = const2\sqrt{\left(\frac{\Delta sumi^2}{ssin2}\right) + \left(\frac{sumi^2}{4ssin2^3}\right)\Delta ssin2^2}, \tag{4.12}$$

or under certain circumstances in the LSFT algorithm the simpler equation is obtained

$$\Delta ftip = \left(\frac{\Delta sumx}{sumx}\right) ftid. \tag{4.13}$$

Table 4.1 summarises the miscellaneous calculations required for the previous expressions.

$$\Delta wtnew = \Delta wtau = \frac{\Delta watan}{watan} wtau,$$

$$\Delta scos2 = \sqrt{4 \cos\phi^2 \sin\phi^2 \Delta\phi},$$

$$\Delta ssin2 = \sqrt{4 \cos\phi^2 \sin\phi^2 \Delta\phi},$$

$$\Delta sumr = \sqrt{\sum_i \cos\phi^2 xstddev[i]^2 + xx[i]^2 \sin\phi^2 \Delta\phi^2},$$

$$\Delta sumi = \sqrt{\sum_i \sin\phi^2 xstddev[i]^2 + xx[i]^2 \cos\phi^2 \Delta\phi^2},$$

$$\Delta sumx = \sqrt{\sum_i xstddev[i]^2},$$

$$\Delta watan = \sqrt{\frac{\Delta csum^2 ssum^2 + \Delta ssum^2 csum^2}{(ssum^2 + csum^2)^2}} \text{ or}$$

$$\Delta watan = \sqrt{\frac{\Delta sumtc^2 ssum^2 + \Delta sumts^2 csum^2}{(ssum^2 + csum^2)^2}},$$

$$\Delta\phi = \Delta arg = \sqrt{\left(\frac{tstddev[i]}{tsamp[i]}\right)^2 arg^2 + \Delta wtnew^2},$$

$$\Delta sumtc = \sqrt{\cos\phi^2 \Delta t^2 + t^2 \sin\phi^2 \Delta\phi^2},$$

$$\Delta sumts = \sqrt{\sin\phi^2 \Delta t^2 + t^2 \cos\phi^2 \Delta\phi^2},$$

$$\Delta arg = \sqrt{\left(\frac{\Delta wrun}{wrun}\right) + \left(\frac{\Delta ttt}{ttt}\right)^2} arg,$$

$$\Delta sumt = \sqrt{\sum_i tstddev[i]^2},$$

$$\Delta ftrp[0] = \frac{\Delta sumx}{sumx} ftrp[0],$$

Table 4.1: Miscellaneous equations required in the calculation of uncertainties for the Lomb-Scargle Fourier Transform.

Chapter 5

Atmospheric Sampling by AMOR

The meteor radar's temporal and spatial sampling of the atmosphere is determined by the combination of the radar's sampling volume along with the quasi-randomness of meteor events. Typically the radar detects around 3000 meteor echoes per day and from these an average of 2250 wind measurements are made. The main reasons for echo rejection are difficulty in unambiguously locating the meteor train or poor echo phase information. During Sporadic-E episodes the AMOR system records multiple echoes which are not easily useful; hence data from these times are rejected.

5.1 Sampling Volume

The AMOR broadside transmitter array operating with horizontal polarisation is at a height of 0.5 wavelengths above ground which should produce a main beam at an angle of 30° to the horizontal (elevation angle). To reduce power in azimuthal sidelobes a nonuniform current distribution is provided along the array[1] which has an additional effect of slightly broadening, in azimuth, the main antenna beam by about 17%. Approximately 0.3% of echoes can be associated with reflections in the sidelobes which, at 2° from the main beam and at typical meteor train ranges, can be treated as being in the main beam without producing significantly different (<0.5%) wind speeds[2]. As meteors only ionise in the atmosphere to form meteor trains in the relatively narrow band of heights from 70 – 120 km, the AMOR atmospheric sampling volume can be represented by figure 5.1.

Figure 5.2 shows the number of echoes recorded as a function of ground range

[1]The angular distribution of field is the Fourier Transform of the antenna feed current distribution across the aperture.

[2]If necessary it would be possible to detect sidelobe echoes by splitting one of the home site receiver antennas in the middle making an azimuth measuring interferometer.

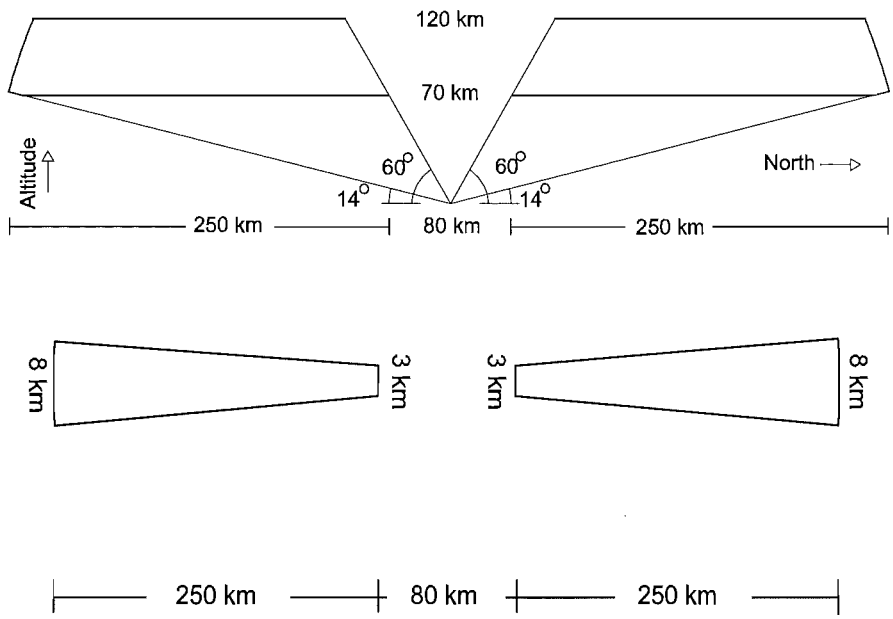

Figure 5.1: The radar sampling volume as viewed from east to west (upper) and a layer at 70 km from above (lower).

for 2.5 years of data. Ground range (GR) is the projection of the range, to the echo reflection point, onto the ground. The zero ground range point is located at the Home site, negative Ground Ranges (GRs) are echoes northward of whereas positive GRs are southward of the Home site.

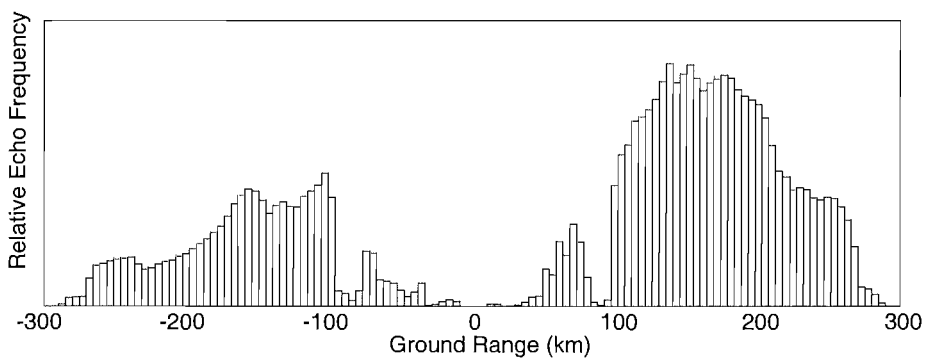

Figure 5.2: Ground range distribution of echoes recorded by AMOR. Data are grouped into 5 km ground range bins.

There is an obvious difference in the total number of echoes recorded from each direction and this is attributed to the location of the main meteor source. For detection by the AMOR radar, the transmitted pulse must undergo specular

reflection from the meteor train. To satisfy this condition echoes detected in the southern beam must have come from meteors which have a source northward of the Birdlings Flat field station and vice versa. As meteors are predominantly caused by meteoroids which have orbits lying near the ecliptic plane, more meteors arrive from north of the radar than from the south, and hence more echoes are detected in the southern beam.

Another feature of figure 5.2 is a reduction in data at ranges of about -100 and 100 km. Two reasons could explain this; either echoes are being rejected due to difficulties in unambiguously determining elevation angles at these ground ranges, or, the antennas are at an effective height of 0.7 λ, thereby producing a null at 45°.

The latter explanation was rejected as the height of the antennas was measured as 0.6 λ from physical ground and previous work in the department indicated that electrical ground was at the same height at the Birdlings site. Additionally, if the data reduction were due to the radiation pattern it would seem unlikely that the data rate would change as dramatically as indicated in figure 5.2. Therefore the sudden reduction in data at ground ranges of about -100 and 100 km is deliberate to remove ambiguous elevation angle measurements.

The reduction of recorded echo numbers with distance is consistent with that expected from the radar equation (1.35) which suggests that the power received from a meteor train at a distance R_0 is proportional to $1/R_0^3$. The received power dependence on range makes the detection of echoes from smaller meteors less likely at large distances, and hence the number of echoes detected reduces with such distance.

The explanation for why there are few echoes close to the Home site (within 100 km) is evident from AMOR's theoretical antenna radiation pattern where the relative electric field (E) at a large distance and angle (θ) is given by

$$E(\theta) = 0.5 \left[1 - \cos\left\{2\pi\left(\frac{2h}{\lambda}\right)\sin\theta\right\}\right],$$

where h is half the distance between the antenna and its image (this is equivalent to the antenna's height if it is situated above ground which is a perfect conductor) and λ is the wavelength of the transmitted radiation. Assuming ground to be a perfect conductor, from figure 5.3 it is evident that for an antenna such as AMOR's located at a height $h = 0.6\ \lambda$ above ground, very little radiation is transmitted in

the main lobe at elevation angles greater than 60°. This implies that few echoes are going to be detected at ground ranges less than approximately 50 km.

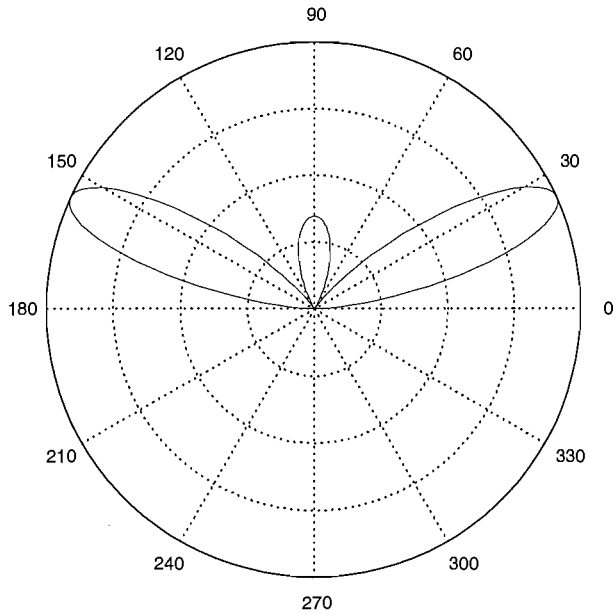

Figure 5.3: The theoretical radiation pattern for the AMOR transmitter antenna. The pattern assumes the antenna to be 0.6 wavelengths above a perfectly conducting ground plane.

The effective antenna height of 0.6 λ also introduces a minor lobe radiating vertically (between 60° and 90°). To detect a meteor in this lobe specular refection requires the statistically less favourable condition of the train being near horizontal to the ground[3]. In addition, the first 16 range bins are not sampled and the minimum range at which echoes can be detected is approximately 96 km. In the rare event that a vertical echo is detected it is often rejected due to the possibility of range alliasing combined with elevation angle ambiguities. Utilising echoes received from horizontal trains could be a valuable means for measuring the atmosphere's vertical motion which has been assumed negligible throughout this work.

A polar plot showing the relative number of echoes detected as a function of elevation angle is given in figure 5.4. Although this diagram cannot be taken directly to imply the antenna's radiation pattern as the number density is biased towards being larger for smaller elevation angles[4], and there are more echoes detected to

[3]This is due to ionisation density being proportional to the cosine of the angle to the zenith at which the meteor enters the atmosphere.

[4]This is evident from figure 5.5 which shows that for a range of elevation angles $\Delta\theta$ about an elevation angle θ the horizontal sampling region is greatest for small θ.

the south, it does indicate that very few echoes are recorded from elevation angles greater than about 50°. There is an obvious absence of echoes which have elevation angles that are less than about 15°. This is due to such echoes requiring a range which is beyond that sampled by the winds system[5].

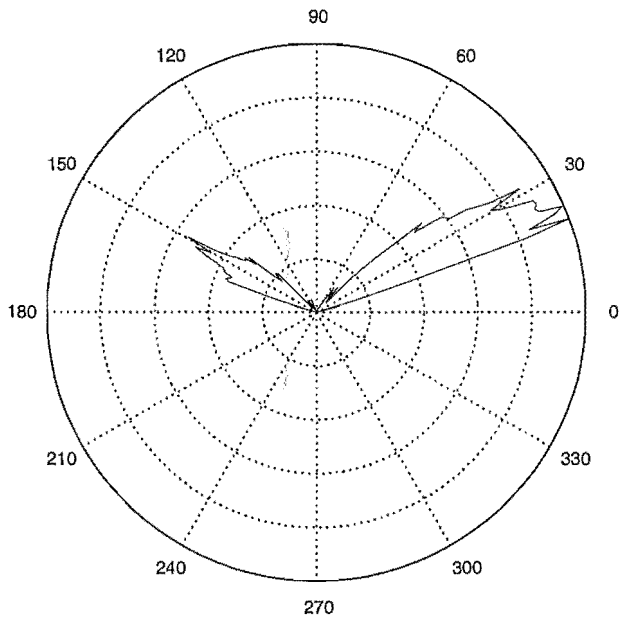

Figure 5.4: The relative number of meteor echoes detected as a function of elevation angle.

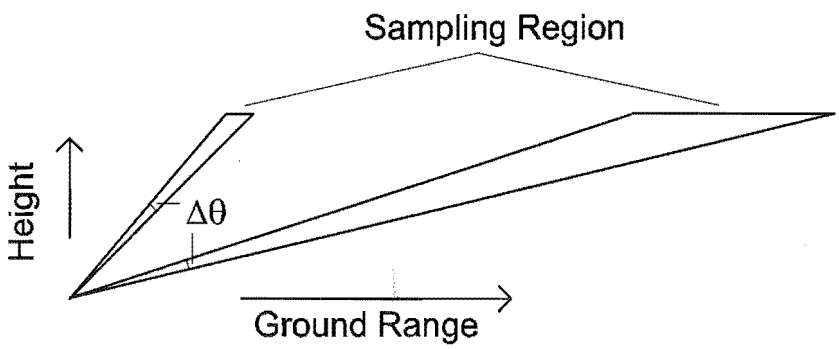

Figure 5.5: The number of meteors detected at smaller elevation angles will be biased larger due to the larger sampling regions associated with them

The spatial number density of meteor wind measurements made between 19 April 1997 and 21 June 1999 is displayed in figure 5.6 and shows a distribution

[5]The AMOR orbit system does record echoes from ranges greater than 300 km. However, as very few echoes are detected at these extreme ranges the extra effort of enabling the winds system to utilise these echoes was not considered.

similar to that predicted by the top diagram of figure 5.1. The height distribution of meteor echoes varies between the two observation directions. Echoes measured in the southern beam are on average detected at greater heights than those in the northern beam due to meteors detected in the southern beam having, on average, higher speeds[6].

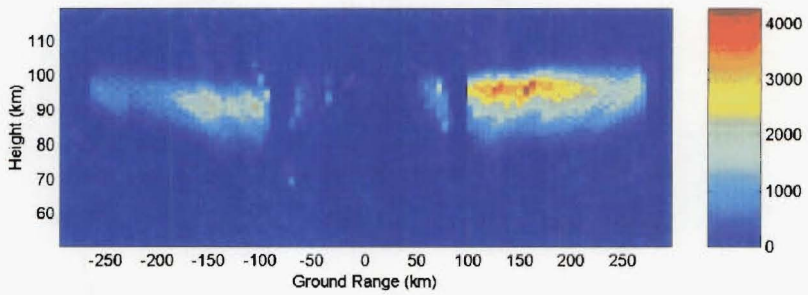

Figure 5.6: The spatial number density of echoes from which wind measurements were derived by the AMOR winds radar. The region of largest echo count is at approximately 97 km in the southern beam and 92 km in the northern beam. This height difference is explained by differences in meteor speeds for each direction.

5.2 Sampling Times

The sampling times for atmospheric measurements with the meteor radar are determined by the random occurrence of meteor events. In addition there is a diurnal variation in the data rate which is maximum in the morning. This is when the apex of the Earth's orbital motion is near the local prime meridian. To establish whether the sampling characteristics (sampling function) of the AMOR meteor winds radar influence the tidal parameters that are calculated with harmonic analysis, semi-diurnal tide amplitude and phase were calculated from Stratospheric Mesospheric Model [Lawrence 1997] data. These values were then compared with amplitude and phase values obtained when these data were filtered to coincide with the meteor radar's atmospheric sampling function.

[6]Meteors detected in the southern beam are generally headed in a direction opposed to the Earth's motion. Hence they have higher speeds in the atmosphere's reference frame and ionise higher up in the atmosphere.

To access the data within the Stratospheric Mesospheric Model's (SMM) binary output files, routines modified from the Middle Atmosphere Dynamics Project at Oxford (MADPO) program suite were generated. These routines produced a text file called `ModelDataC.dat` (C for continuous sampling) which contained meridional wind speeds produced every four minutes at 80 km and $45°S$ for a simulation of approximately 2 years.

A second file called `ModelDataS.dat` is produced from the `ModelDataC.dat` file. For a wind speed in `ModelDataC.dat` to also be in `ModelDataS.dat` (S for sampled) the time of the measurement has to coincide within two minutes of an AMOR detected meteor event. The selection process can also require that the height of the meteor be within a certain range. Thus, the number of SMM wind speeds in the file `ModelDataS.dat` will reflect how well the AMOR radar sampled the atmosphere, as a function of time, over a particular height range.

A reference file containing AMOR meteor event times and heights, called `Amor_WND.dat`, was produced in advance and enabled the SMM model data selection.

A 30 day sliding window LSFT harmonic analysis of the data in `ModelDataC.dat` produces a reference amplitude and phase profile of the SMM's semidiurnal tide. These profiles are then compared with the profiles obtained from the identical analysis applied to the data in the `ModelDataS.dat` files.

Sampling the data in `ModelDataC.dat` to coincide with AMOR meteor events which occurred over the range 95 ± 0.5 km, and applying LSFT harmonic analysis produced the semidiurnal tide amplitude profile shown as a light line in figure 5.7. The dark line is the calculated semidiurnal amplitude profile when all data within `ModelDataC.dat` were analysed. The phase comparison is presented in figure 5.10. The good agreement shown between profiles generated from sampled and continuous data indicates that the AMOR sampling function is acceptable for this type of analysis at 95 ± 0.5 km.

From figure 5.7 it can be seen that there are times when agreement between the two profiles is better than at other times. Some of these differences can be attributed to the diurnal nature of the sampling function; this is explained in section 5.4.

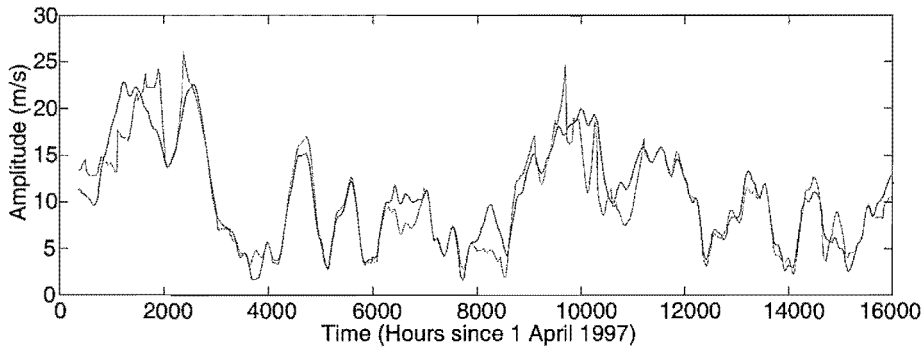

Figure 5.7: Amplitude profile of semidiurnal tide present in SMM at 80 km. The dark line shows the amplitude profile obtained with harmonic analysis for the complete hourly averaged data set whereas the light line is the amplitude profile obtained from hourly averaged data which have been selected to coincide with times for actual meteor wind velocity measurements from 95±0.5 km.

5.3 Height Dependent Sampling

Due to the decrease in atmospheric density with height and the mass distribution function of meteors, the number of echoes varies with height. The height distribution of wind measurements made by the AMOR radar is shown in figure 5.8. Echoes are rejected by the system if their heights are calculated outside of the range of 70 – 120 km. The figure displays the height distribution by giving the percentage of echoes for both the northward (dashed line) and southward (solid line) beams for each at one kilometre height intervals.

The previous section concluded by showing that the semidiurnal tide's amplitude and phase were well recovered from the SMM when the SMM data were sampled at times consistent with the meteor radars sampling of the atmosphere at 95±0.5 km. However, at heights away from 95±0.5 km the number of AMOR detected meteor echoes reduces and hence a repetition of the previous section's analysis at other heights would produce a `ModelDataS.dat` file with fewer SMM mean wind measurements. This section shows that harmonic analysis of the `ModelDataS.dat` data for sampling heights between 80 and 110 km gives good agreement.

To determine whether AMOR samples adequately at other heights, the SMM meridional wind measurements were sampled to be consistent with AMOR wind measurements over a series of heights at 5 km intervals from 75±0.5 km to 110±0.5 km.

The semidiurnal tide amplitude and phase profiles for these heights are given

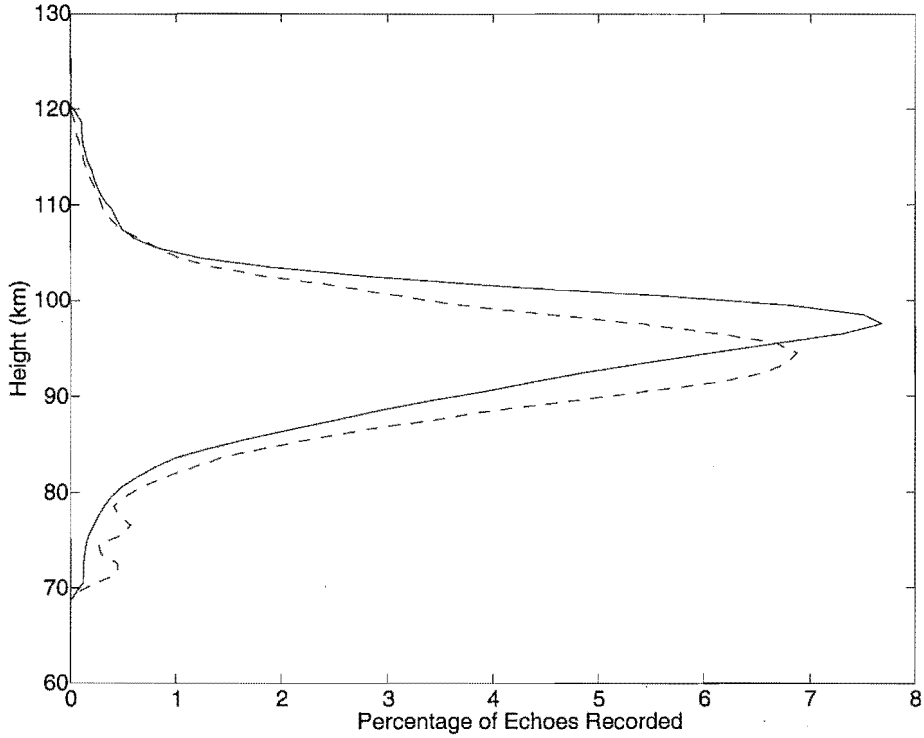

Figure 5.8: The percentage of wind measurements recorded by the AMOR system as a function of height. The solid line peaking at 97 km shows the distribution of wind measurements detected in the southern beam and the dashed line peaking at 92 km shows the distribution detected in the northern beam. Echoes which appear to be outside of the 70–120 km height interval are rejected by AMOR.

in figures 5.9 and 5.10 The dark lines are profiles from the continuous SMM data and the lighter lines are from SMM data sampled relative to meteor activity. The top graphs of figures 5.9 and 5.10 are produced from data sampled at times which correspond to meteor events at 110±0.5 km, the remaining graphs step down in intervals of 5 km to the lowest height of 75±0.5 km. The acceptance layer thickness for each height is 1 km.

These graphs show that the atmospheric sampling by the meteor radar at heights between 80 and 110 km is high enough to produce semidiurnal tide amplitude and phase profiles which are similar to profiles produced from continuous SMM data. Correlation coefficients (at a 95% confidence level) between semidiurnal tide parameters for sampled and continuous data are given if table 5.1. The amplitude correlation values are maximum with a value of 0.92 at heights between 90 and 100 km. The amplitude correlation values reduce with increased height to a value of 0.82 at 110 km. The amplitude correlation values also decrease with decreasing

height. The poorest amplitude correlation is 0.80 at 75 km.

The trend for the semidiurnal tide phase correlation coefficients is similar to that for the amplitude correlations. Maximum correlations of 0.96 and 0.97 are obtained between heights of 90 and 100 km. The phase correlation reduces to 0.85 as the height increases to 110 km. Again the poorest correlation is at 75 km with a value of 0.67.

As semidiurnal tide parameters for the meteor radar sampled data display high correlation with those for the homogeneous data, it is proposed that the meteor radar sampling function is sufficient for analysing oscillations with periods equal to, or greater than, the semidiurnal tide for heights between 80 and 110 km.

Height (km)	**Amplitude Corr Coeff**	**Phase Corr Coeff**
110	0.82	0.85
105	0.83	0.92
100	0.92	0.97
95	0.92	0.97
90	0.92	0.96
85	0.90	0.92
80	0.82	0.86
75	0.80	0.67

Table 5.1: This table provides values for the correlation coefficient between full model data set and meteor event sampled model data set as a means of quantifying how well the meteor radar retrieves amplitude and phase values for the semidiurnal tide at various heights.

5.4 Influence of Diurnal Sampling on Tidal Amplitudes

This section investigates whether tidal amplitudes are affected by the inherent diurnal variation in the sampling rate of the AMOR meteor radar. This is achieved by defining a sampling function for hourly averaged data containing either a value of one or zero. Hours for which a mean wind velocity has been calculated are associ-

ated with the value one, and hours for which no wind measurement has been made are associated with a value of zero.

Harmonic analysis (LSFT) was applied to this sampling function representing 2.5 years AMOR winds data to identify whether any sampling periodicities were evident. The power spectrum from this is shown in figure 5.11 and indicates a strong mean wind along with a diurnal component. As this analysis was applied to all available data the returned spectrum shows average values for sampling periodicities. These values possibly indicate that there are times where the sampling function is highly periodic and that there are other times when it is not.

Further inspection reveals that there are also harmonics of the diurnal component, however their amplitudes are not as significant and are not considered further.

The effect of sampling the atmosphere in a periodic fashion is that each periodicity (of frequency f_1 with units of cycles per day, where $\omega_1 = 2\pi f_1$) within the sampling function will combine with periodicities in the hourly averaged data (of frequency f_2) and put power into the sampled wind speed spectrum at frequencies $f_1 + f_2$ and $f_1 - f_2$. In other words beat frequencies will be generated; this is evident from the trigonometric relationship

$$\cos\omega_1 t \cos\omega_2 t = \frac{1}{2}\cos\left(\omega_1 + \omega_2\right) t + \frac{1}{2}\cos\left(\omega_1 - \omega_2\right) t. \tag{5.1}$$

Therefore, in the presence of a semidiurnal tide ($f_2 = 2$) and diurnal ($f_1 = 1$) sampling, equation 5.1 indicates that harmonic analysis would introduce (spurious) components at frequencies of 1 cycle per day and 3 cycles per day. Likewise, diurnal sampling in combination with a diurnal tide ($f_2 = 1$) would produce harmonics at "dc" (mean wind) and 2 cycles per day.

Should harmonic analysis of the sampling function reveal only a mean value, i.e. the data are continuously sampled, then $f_1 = 0$ and the frequencies f_2 appear correctly with no false harmonics.

The spectrum obtained from data of a periodic nature which has been sampled with a sampling function which is periodic also can be represented with the expression

$$\text{Spectrum} = \sum_{m=1}^{q}\sum_{n=1}^{p}\cos\omega_m t \cos\omega_n t, \tag{5.2}$$

where p and q are the number of significant harmonics in the sampling function and wind speed data respectively.

As discussed in section 1.2.1 the dominant signature of mid-latitude atmospheric motion is the semidiurnal tide with a frequency of 2 cycles per day. In the presence of diurnal sampling it would be expected that the terdiurnal component and mean value would be affected. Additionally, should the diurnal tide be of a reasonable amplitude then spurious power would contaminate the semidiurnal tide. To observe these effects, the amplitudes of the semidiurnal and terdiurnal tides are calculated from meteor event time sampled and continuous data sets from the SMM for comparison (refer graphs 4 and 6 of figures 5.13 to 5.16).

Taking as an example results from 95 km, the top graph in figure 5.15 shows the amount of diurnal sampling. This was achieved with a sliding window Lomb-Scargle Fourier Transform of the sampling function. The greatest influence on the terdiurnal tide, due to diurnal sampling and the semidiurnal tide's amplitude, is likely to occur when their product is greatest. Graph 5 of figure 5.15 is the product of the diurnal sampling component with the semidiurnal tide amplitude (sampled data) which, when large, should indicate regions of terdiurnal contamination and when small should identify where the terdiurnal tide remains unaffected. Graph 5 of figure 5.15 suggests that terdiurnal contamination should be relatively small between elapsed hours 4000 and 7500 and graph 6 reflects this. In contrast graph 5 indicates that the influence on the amplitude on the terdiurnal tide should be large in the region 9500 – 11000 elapsed hours. Again this is observed in graph 6.

Similarly the semidiurnal tide's amplitude, calculated from the sampled data, is seen to differ most from that calculated from the continuous data when the product of the diurnal component of the sampling function and the diurnal tide's amplitude is greatest. This is apparent at times about elapsed hours 8000 and 9700 in graph 4 of figure 5.15.

The effect of diurnal sampling can be reduced by either increasing the sampling volume,thereby increasing the likelihood of detecting a meteor echo and hence determining a mean value, or increasing the time interval. Naturally, each of these methods results in a reduced resolution of the parameter increased.

Since the number of echoes varies with height as a result of the meteor ionisation process (refer figure 5.8), it is expected that diurnal sampling would increase at

	Influence on S_2		Influence on S_3	
Height (km)	Corr Coef (r)	% Explained (r^2)	Corr Coef (r)	% Explained (r^2)
75	0.1696	2.9	0.3021	9.1
85	-0.0302	.09	0.4223	17.8
95	0.1033	1.1	0.3514	12.3
105	0.2582	6.7	0.5921	35.1

	Influence on Mean Wind	
Height (km)	Corr Coef (r)	% Explained (r^2)
75	-0.2461	6.1
85	0.2368	5.6
95	-0.0694	0.5
105	-0.1413	2.0

Table 5.2: The top table presents correlation coefficients for calculated tidal amplitude differences and product of sampling function with lower frequency tide. The lower table presents the correlation with mean wind differences.

heights where there are fewer echoes. Figure 5.12 shows the harmonic spectra of AMOR's sampling function at heights of 105, 95, 85 and 75 km. Diurnal sampling is shown to be greatest at the extreme ranges of 105 and 75 km and therefore it is expected that the greatest tide contamination will be evident at these heights.

Figures 5.13 to 5.16 show the influence of diurnal sampling in combination with diurnal and semidiurnal tides on semidiurnal and terdiurnal tides for heights between 75 and 105 km. The effect on calculated mean wind values is shown in figure 5.17. By inspection it can be seen that poor tidal (of frequency n cycles per day) amplitude agreement occurs at times consistent with a large product of diurnal sampling function and amplitude of tide with frequency $n-1$ cycles per day.

Correlating continuous and sampled amplitude difference values for tide n with the diurnal sampling and tide $n-1$ product, produces the correlation coefficients presented in table 5.2. Values are significant at the 95% confidence level.

It is evident from these results that diurnal sampling has a greater affect on the calculated amplitude for the terdiurnal tide as such sampling explains more of the differences in the profiles. Diurnal sampling explains over 35% of the differences in amplitude profiles for the terdiurnal tide at 105 km. It is interesting to notice that

this effect only explains 9% of the profile variation at 75 km. It was expected that a value nearer to that obtained at 105 km would be returned as the diurnal sampling power indicated in figure 5.12 shows a value comparable with that at 105 km.

The diurnal sampling/semidiurnal tide interaction is not evident in calculated mean wind comparisons as only data from 85 km show a positive correlation.

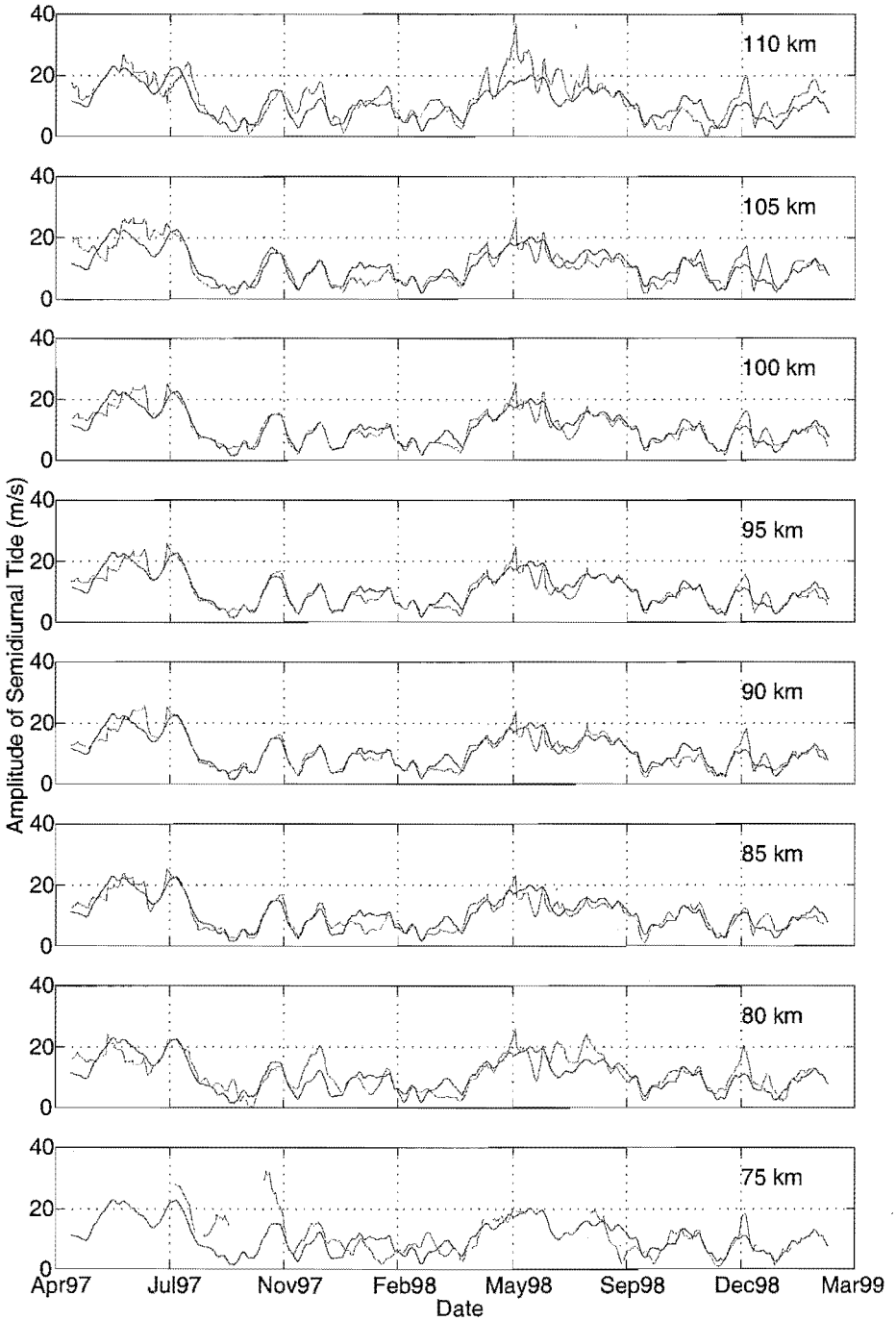

Figure 5.9: The dark lines show, for comparison, the amplitude of the semidiurnal tide as calculated for the complete model generated data set. The light lines show the semi-diurnal tide amplitude when the model data set is sampled according to timing of meteor events. The sampling heights, with a spread of ± 0.5 km, are indicated on each graph.

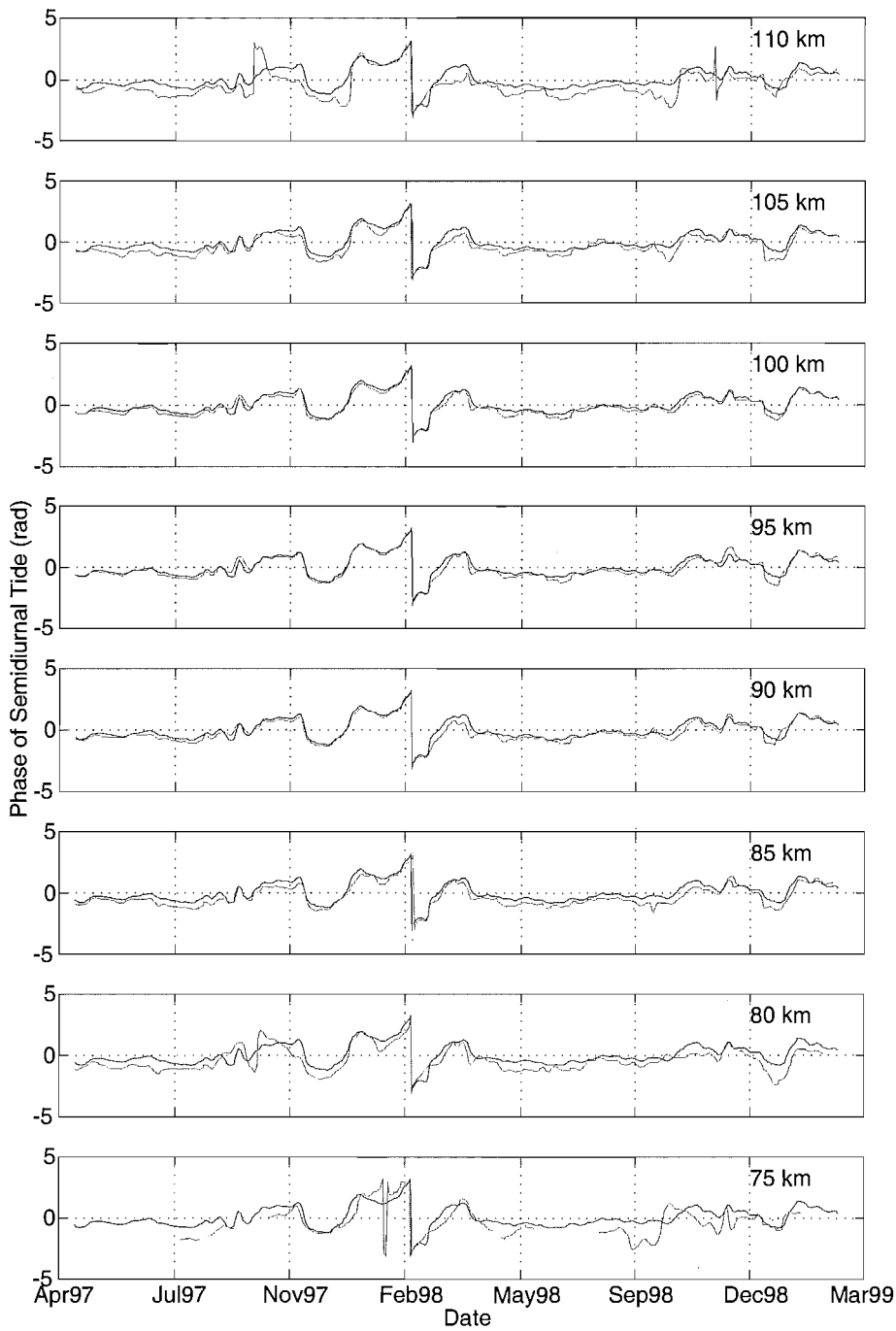

Figure 5.10: The dark lines show, for comparison, the phase of the semidiurnal tide as calculated for the complete model generated data set. The light lines show the semi-diurnal tide phase when the model data set is sampled according to timing of meteor events. The sampling heights, with a spread of ± 0.5 km, are indicated on each graph.

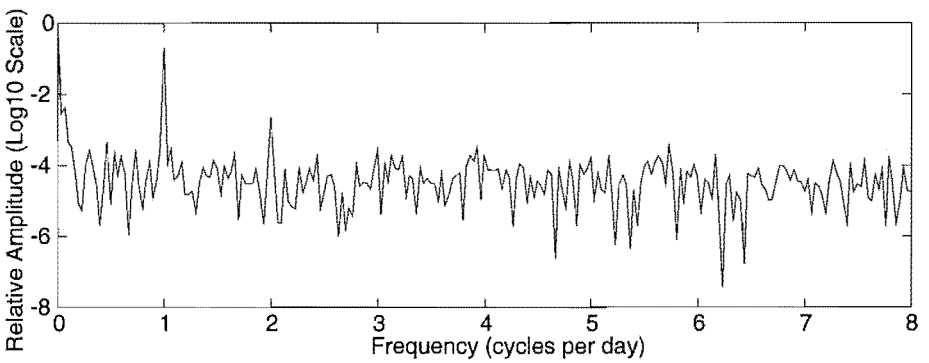

Figure 5.11: A harmonic transform for the 2.5 years of hourly averaged sampling times present in the AMOR winds data set. The dominant mean value exists although a diurnal component is also present.

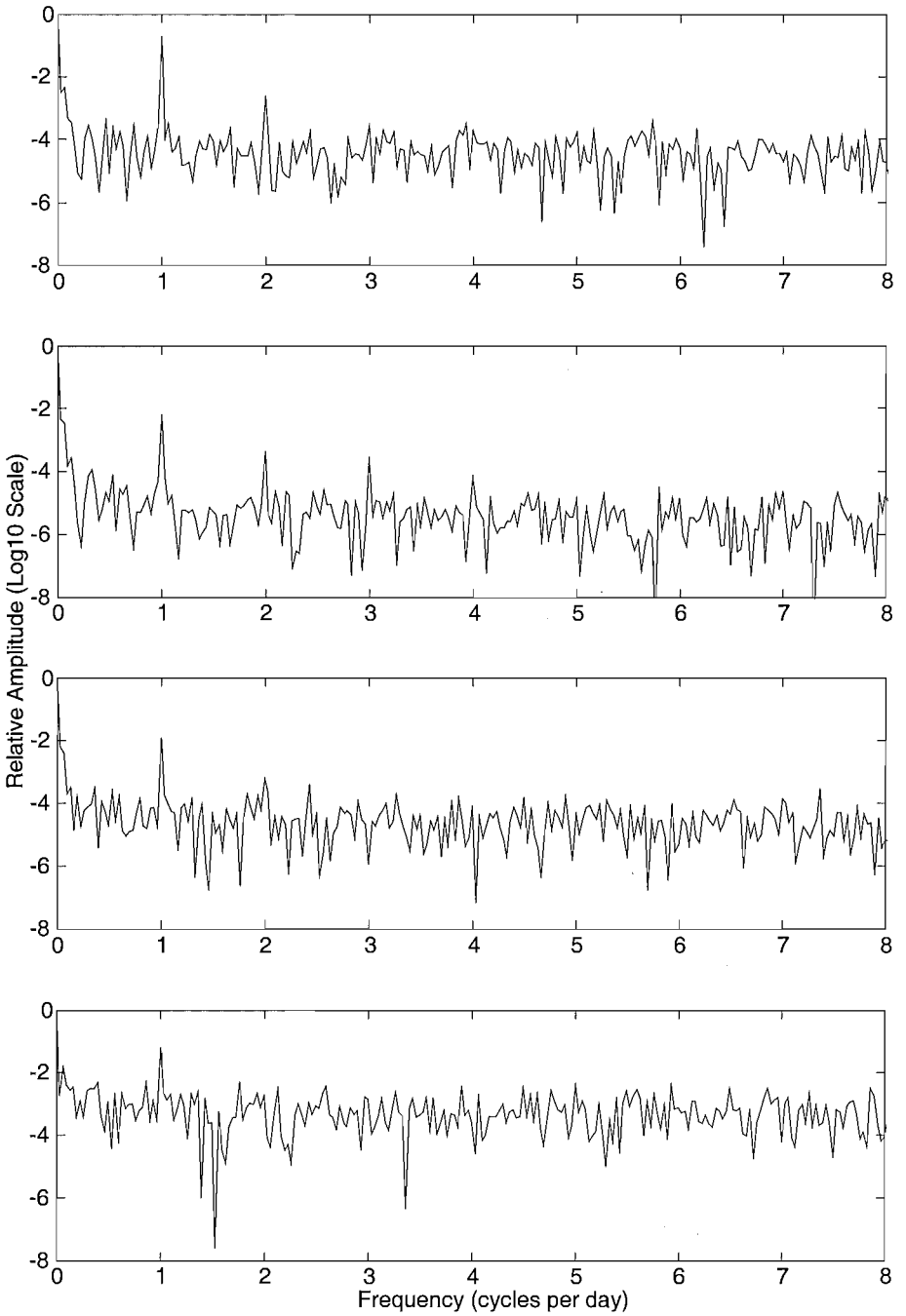

Figure 5.12: Harmonic analysis of AMOR's sampling function at 10 km height intervals. The top graph is at 105 km and the bottom panel is at 75 km. The dominant (non dc) term at each height is the diurnal sampling rate.

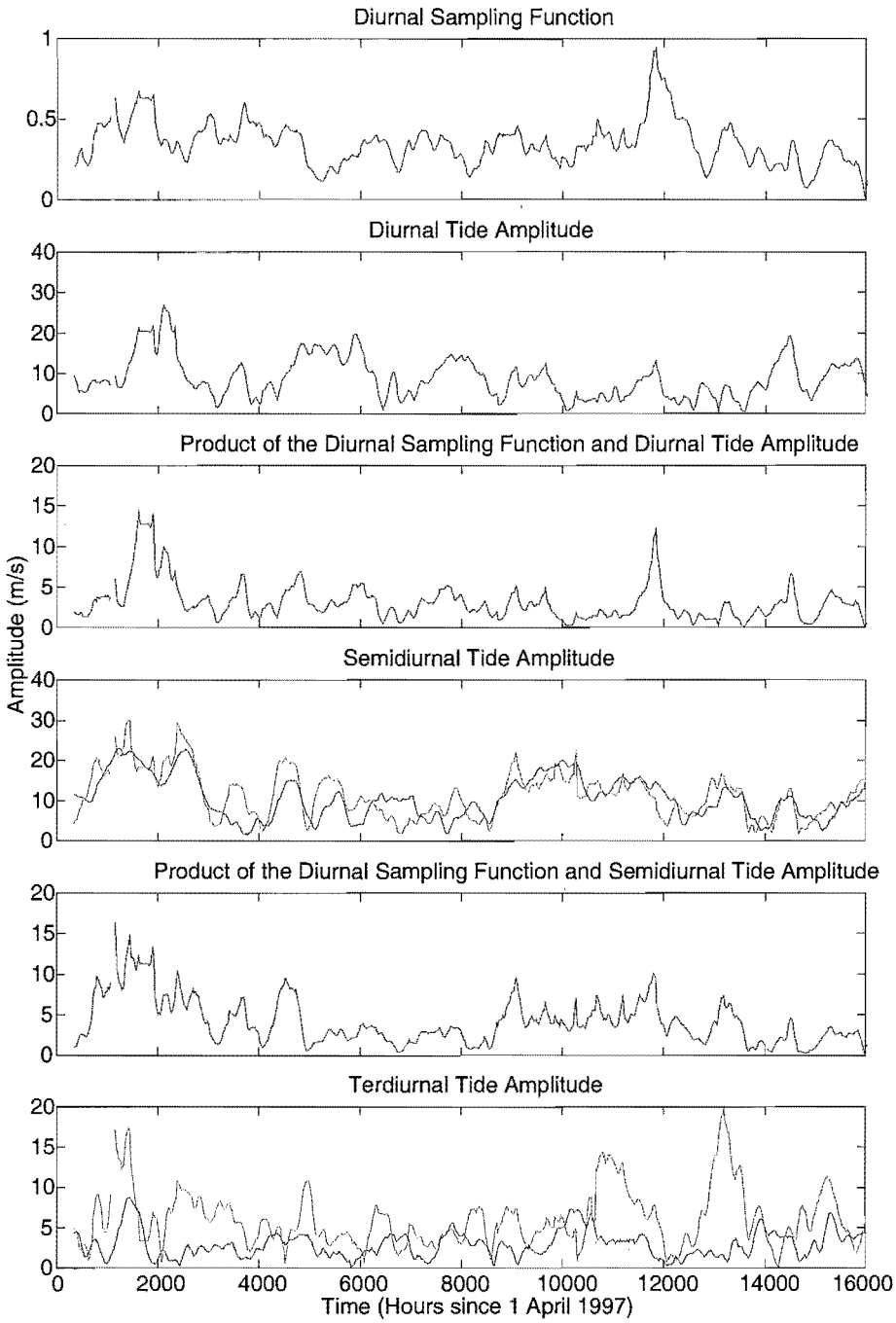

Figure 5.13: The effect of diurnal variation in data sampling on the harmonic analysis of model data which has been sampled at times consistent with meteor events at 75±0.5 km. For reference the graphs are numbered from 1 to 6 (upper to lower). Dark and light lines represent analysis for the homogeneous and meteor event sampled data respectively. 0 hours is 0:00:00 1 April 1997.

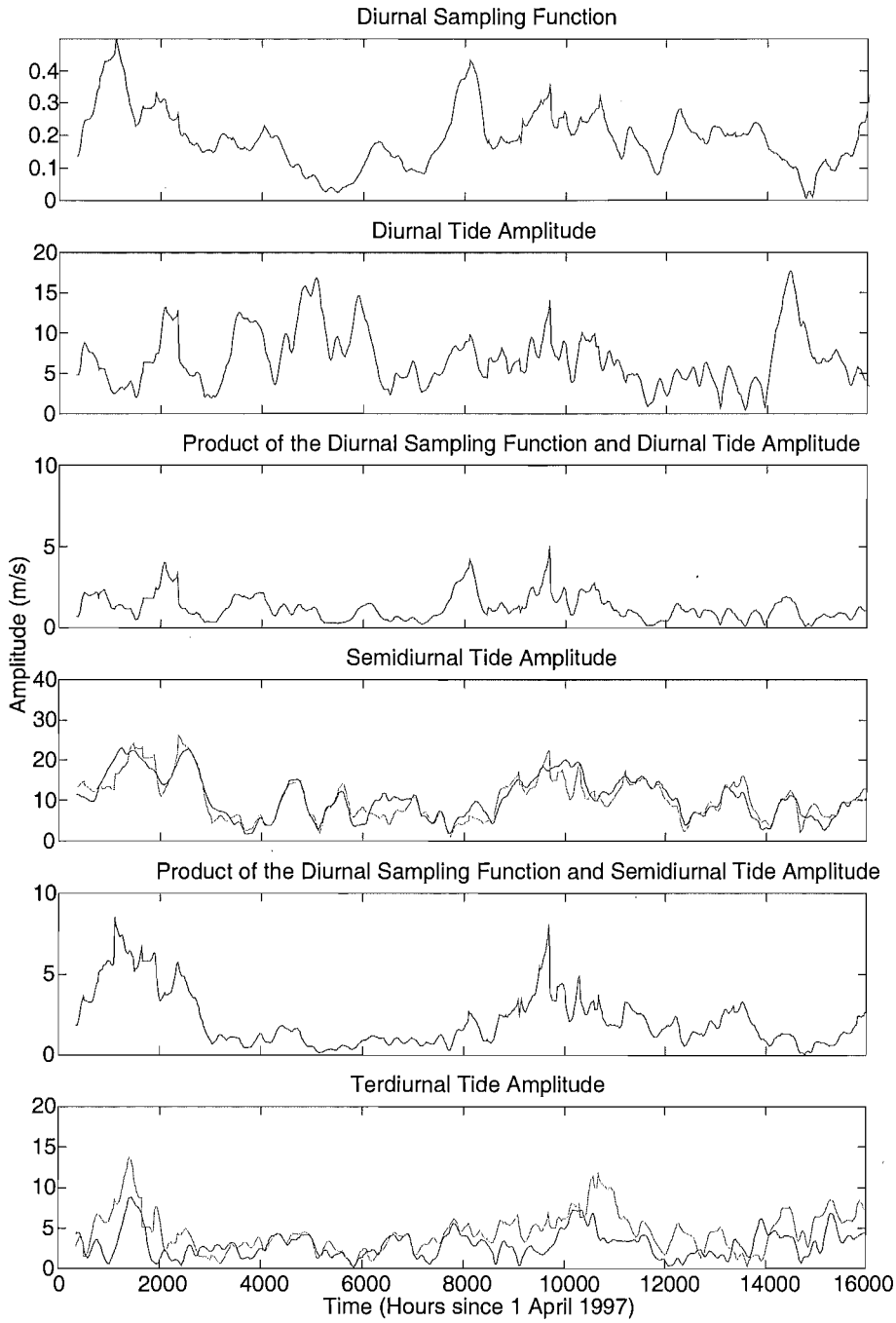

Figure 5.14: The effect of diurnal variation in data sampling on the harmonic analysis of model data which has been sampled at times consistent with meteor events at 85±0.5 km. For reference the graphs are numbered from 1 to 6 (upper to lower). Dark and light lines represent analysis for the homogeneous and meteor event sampled data respectively. 0 hours is 0:00:00 1 April 1997.

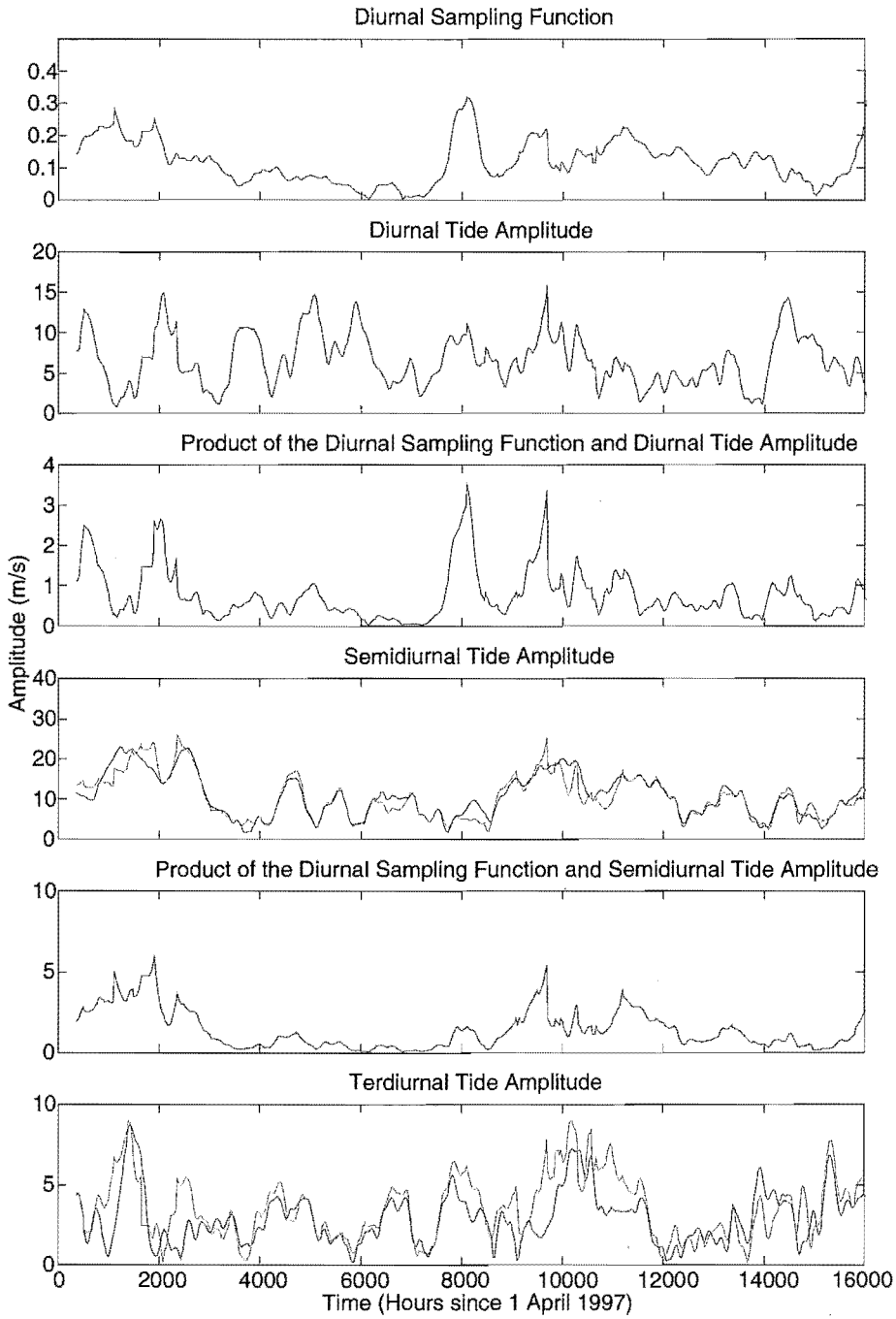

Figure 5.15: The effect of diurnal variation in data sampling on the harmonic analysis of model data which has been sampled at times consistent with meteor events at 95±0.5 km. For reference the graphs are numbered from 1 to 6 (upper to lower). Dark and light lines represent analysis for the homogeneous and meteor event sampled data respectively. 0 hours is 0:00:00 1 April 1997.

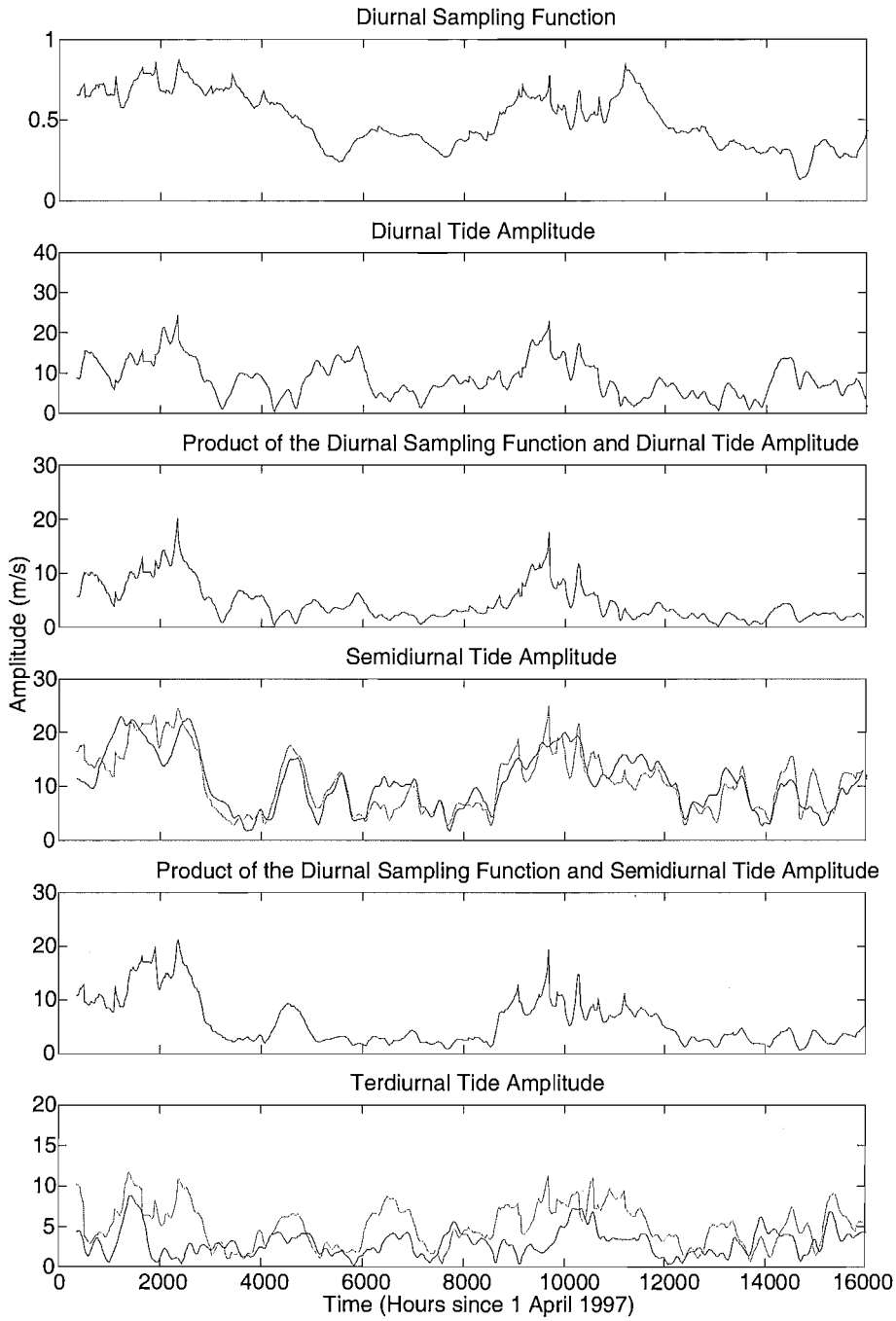

Figure 5.16: The effect of diurnal variation in data sampling on the harmonic analysis of model data which has been sampled at times consistent with meteor events at 105±0.5 km. For reference the graphs are numbered from 1 to 6 (upper to lower). Dark and light lines represent analysis for the homogeneous and meteor event sampled data respectively. 0 hours is 0:00:00 1 April 1997.

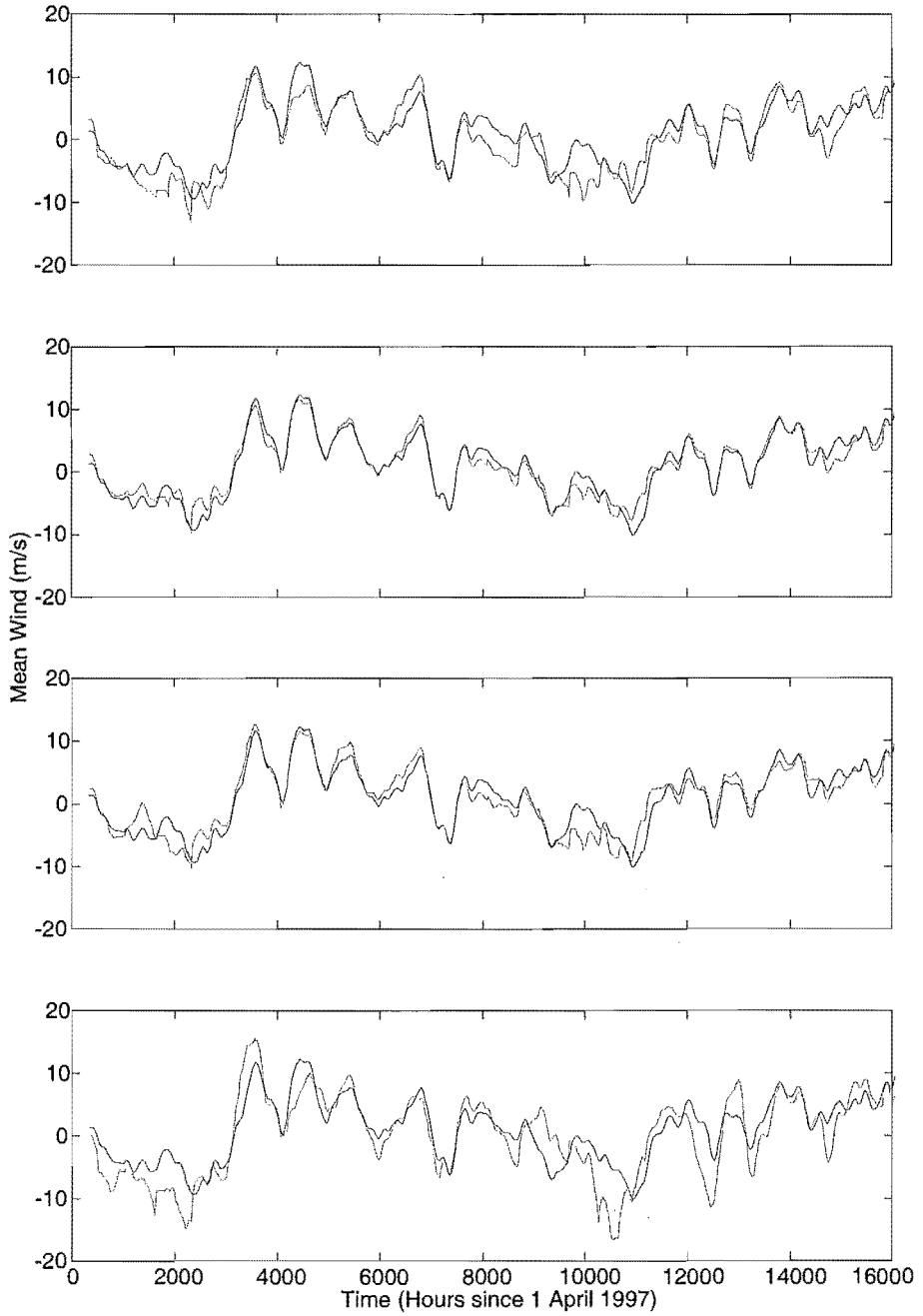

Figure 5.17: Mean wind comparisons for complete (dark) and meteor sampled (light) data SMM at heights from 105 (top) to 75 (bottom).

Chapter 6

AMOR Winds

In this chapter results from the analysis of wind measurements during the period 19 April 1997 to 21 June 1999 from the AMOR meteor winds radar are presented. These results are included to indicate the capabilities of the instrument; they are not intended as a detailed survey or analysis of dynamics in the meteor zone of the Earth's atmosphere.

A selection of raw wind measurements are presented in figure 6.1 from the period 2–7 May 1997. From these data two observations can be made. The first observation is that the data density appears to vary throughout the day; this diurnal variation, and its implications on tidal analysis, has been discussed in section 5.2. The second observation is that the data clearly follow a sinusoid with a period of approximately 12 hrs. This oscillation, the semidiurnal tide (S_2), is generally the strongest signature in AMOR wind speed measurements and is the initial focus of this chapter.

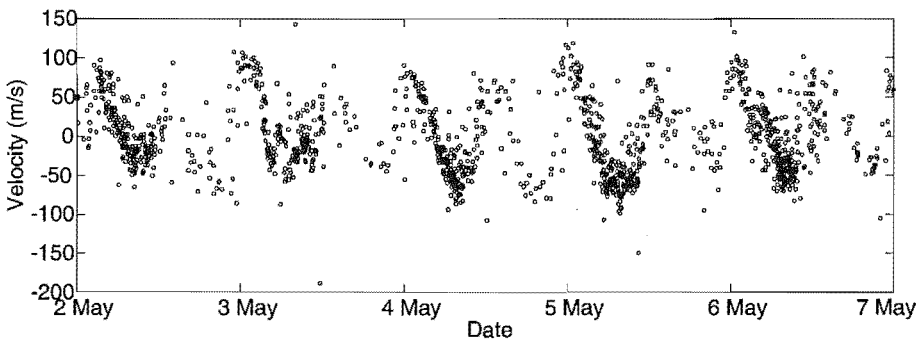

Figure 6.1: Individual wind measurements for the period 2–7 May from 95±2 km are presented. The semidiurnal tide is clearly seen along with a diurnal variation in data rate. The uncertainties for each measurement are typically 1–3 $\mathrm{m\,s^{-1}}$.

6.1 Tides in Meteor Winds Data

Normalising the Lomb Scargle Fourier Transform (LSFT) power spectrum of hourly averaged AMOR meteor winds data at 95±0.5 km from the period 19 May 1997 to 21 June 1999 produces the spectrum given in figure 6.2. The dotted line, at the level of Siegel's test statistic [Siegel 1980], indicates which harmonics (up to a maximum of three) are statistically significant at a 99% level. The spectrum in figure 6.2 shows the 2 cycles per day semidiurnal tide to be the most significant oscillation. From Siegel's statistic the 3 cycle per day terdiurnal tide and a 1 cycle per day diurnal tide also appear as significant features in the AMOR winds data set.

Tentative evidence, from the broadening of the harmonic peak at 2 cycles per day, also suggests that the frequency of the semidiurnal tide (S_2) may change slightly throughout the year.

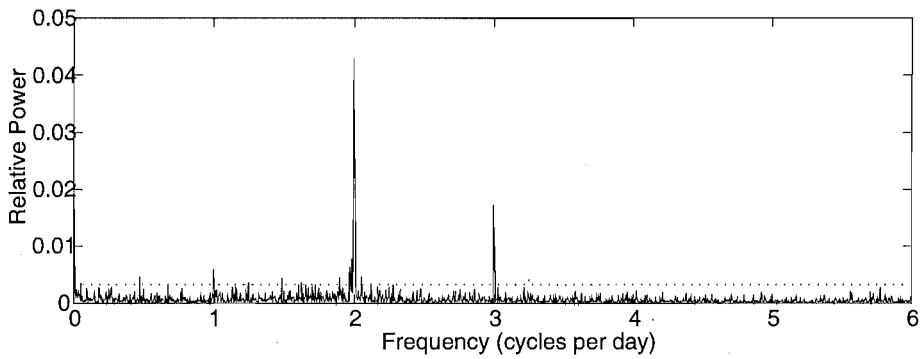

Figure 6.2: Spectral analysis of meteor winds data from 95±0.5 km. The dotted lines show Siegel's statistic calculated for the spectrum and indicates a cutoff for significant frequencies.

6.2 Amplitude of the Semidiurnal Tide

Changes in the amplitude of the semidiurnal tide were observed by applying a 30 day sliding window to hourly averaged AMOR data at 1 km height intervals in the range 91±0.5 km up to 104±0.5 km. This ensured that each meteor echo wind measurement could only be included in the analysis at one of the analysed heights, and data at each height were independent of data at other heights.

It is evident from the amplitudes of the semidiurnal tide derived from the meteor winds data as given in figure 6.4 that, at any particular time, they all displayed sim-

ilar amplitudes. This obviously important result was very positive as it confirmed that consistent results were being obtained from the analysis of wind measurements derived from independent sets of meteors.

For heights between 93 and 103 km there was a cyclic repetition in the semidiurnal tide's amplitude. The most obvious feature of these graphs was the large amplitude consistently observed during autumn months.

Inspection of the semidiurnal tide's amplitude during autumn 1997 revealed that the peak amplitude of this tide was maximum at 99 km and steadily decreased at heights away from 99 km; this feature may be explained by the presence of more than one tidal mode (see section 6.4).

The amplitude of the semidiurnal tide obtained from AMOR winds data when analysed with a 30 day window has been summarised in figure 6.3.

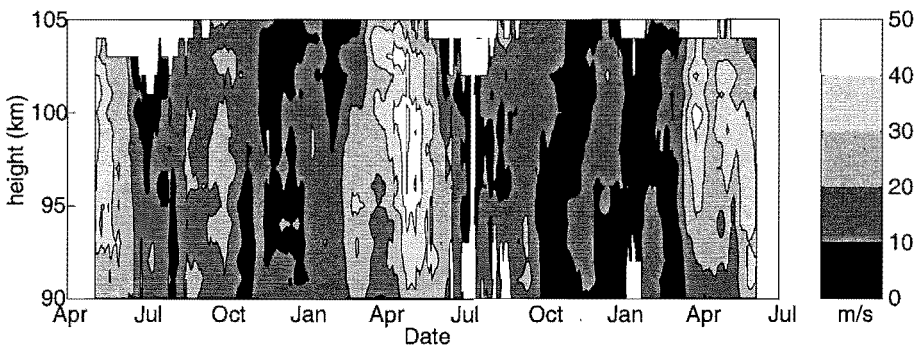

Figure 6.3: Amplitude of the semidiurnal tide measured by AMOR with a 30 day window.

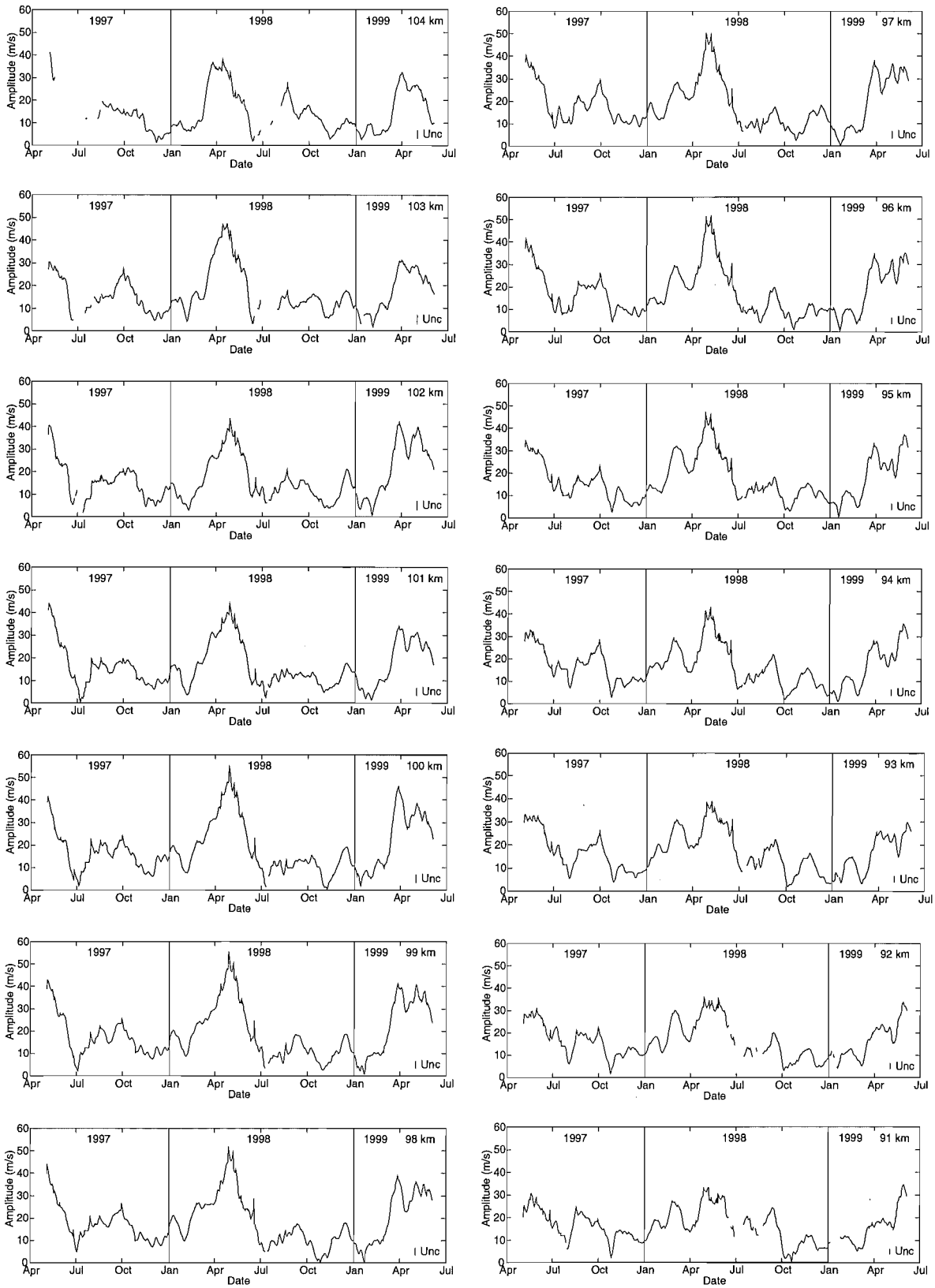

Figure 6.4: Amplitude of semidiurnal tide as determined at the heights indicated in each graph. The mean uncertainty at each height (2 s.d.) is also indicated.

6.3 Phase and Period of the Semidiurnal Tide

The phase (defined as time of maximum northward motion) of the semidiurnal tide was seen to be height dependent (refer figure 6.6). The phase was generally measured as being earlier at greater heights, i.e., for May 1997 the phase at 104 km the phase was measured as 11.5 hours NZST[1] whereas at 91 km the phase was 15 hours NZST.

The phase of the semidiurnal tide also exhibited seasonal variation throughout the year. Inspection of the available data from the year 1997 showed the phase gradually increased to a maximum in winter months. The phase then tended to decrease slowly towards the 1997/1998 summer. Late summer 1998 phase decreased throughout February, to a yearly low for heights above 99 km, before increasing again in March. The phase then steadily increased until late April or early May where it remained near constant until June. During June the phase decreased slowly before a dramatic phase increase during July. This sudden phase increase was also seen in the 1997 data. For the period of July 1998 through to the end of September 1998 the phase below 100 km was seen to steadily increase; above 100 km it steadily decreased. At heights below 99 km, October 1998 results showed the phase of the semidiurnal tide increasing to the extent that it went through 12 hours (2π) and appears as a smaller phase; this is possibly as a consequence of the S_2 amplitude being particularly small during this time (refer figure 6.4). At heights above 99 km the 1998 phase slowly decreased and continued to decrease through till the end of that year. The phase was then seen to increase rapidly throughout October to values consistent with those of the previous year. A sudden temporary phase increase was observed in December 1998 at all heights. January 1999 phase at heights between 93 and 103 km showed unusual behaviour. It is not certain why the phase should experience such a height dependent shift. However, this time of year does coincide with the two-day planetary wave which may have caused the effect (note however, that such an event was not observed the previous year) or it may have originated as a result of several days missing data (the missing data is evident from figure 6.8).

The measured phase values tended to lie within the values 0800 and 1600 hours NZST and were rarely seen to exist outside of this range. Occasions when phase

[1]New Zealand Standard Time, equivalent to UT + 12 hours.

values did go beyond this range briefly were generally coincident with small semidiurnal tide amplitudes and may be evidence for interference between more than one tidal mode coexisting with similar amplitudes and different phases. Annual phase variations with similar seasonal trends to those described here have also been reported from radar measurements by Vincent et al. [1988] and optical wind measurements by Fauliot et al. [1995].

The phase of the semidiurnal tide obtained from AMOR winds data when analysed with a 30 day window has been summarised in figure 6.5.

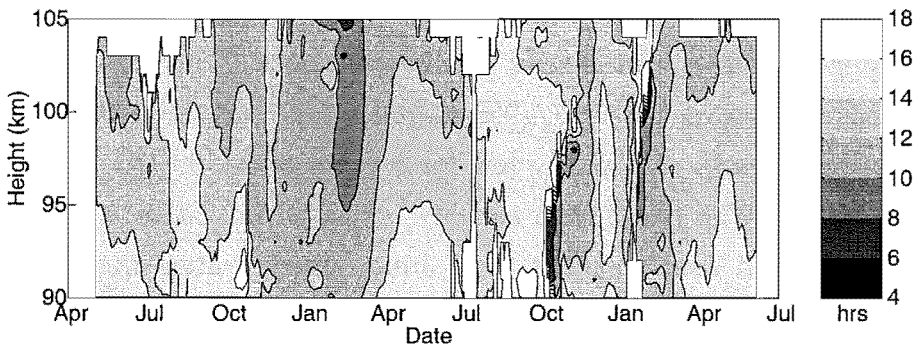

Figure 6.5: Phase (NZST) of the semidiurnal tide as measured by AMOR with a 30 day window.

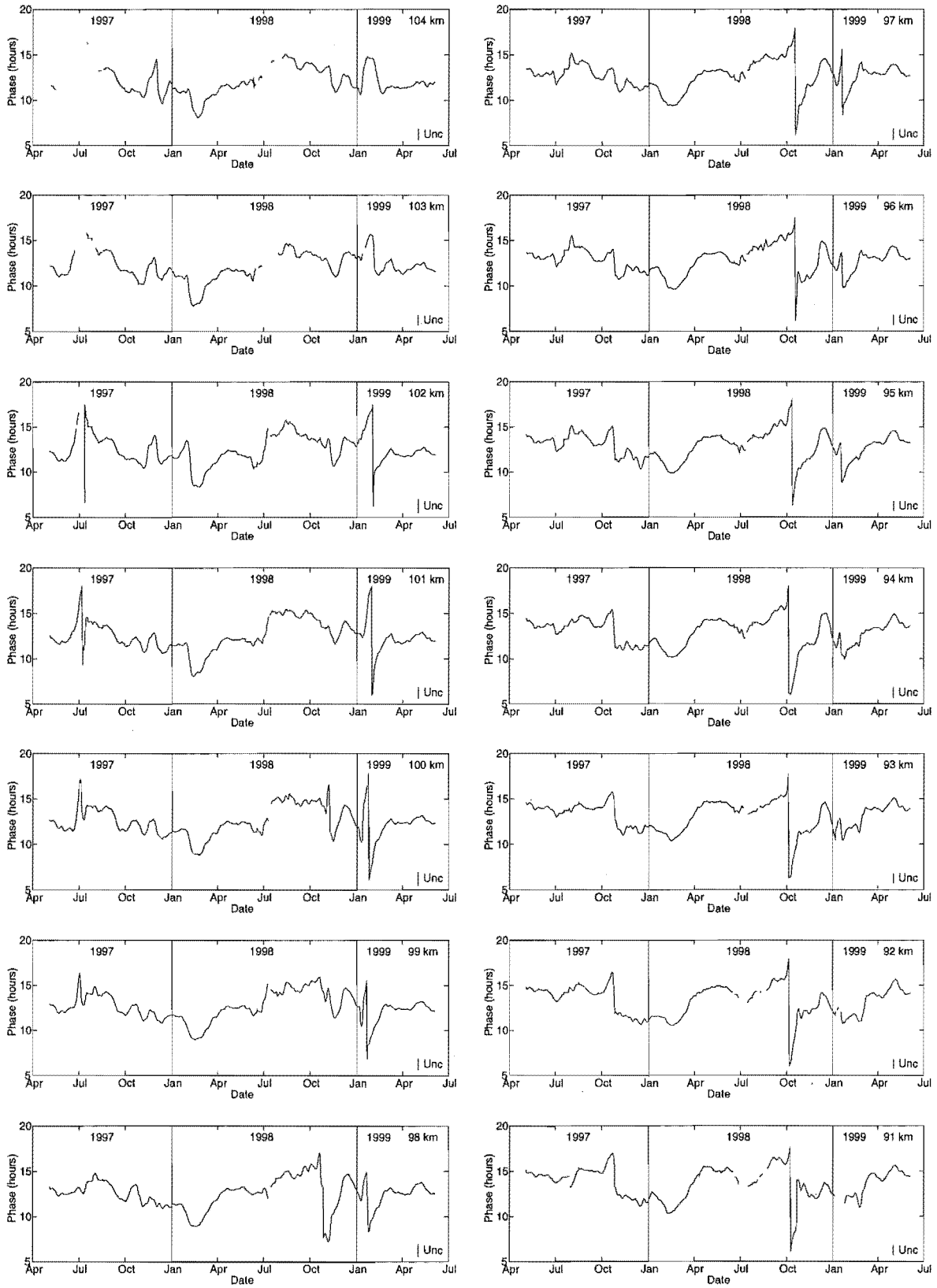

Figure 6.6: Phase (NZST) of semidiurnal tide as determined at the heights indicated in each graph. The mean uncertainty at each height (2 s.d.) is also indicated.

As the phase of the semidiurnal tide gradually changed over successive 24 hour periods, its period couldn't have remained constant at 12 hours. To observe this frequency variation the Lomb Periodogram [Scargle 1982] was applied to 30 day windowed hourly averaged wind data and the largest amplitude period, near 12 hours, was taken as the semidiurnal tide's period at that time.

Variation in the period of the semidiurnal tide within AMOR data at heights from 91–104 km is shown in figure 6.7. For consistency the phases in figure 6.6 should increase during episodes when the period of the tide is greater than 12 hours, and conversely the phases should decrease when its period is less than 12 hours. Regions of constant phase only occur when its period is precisely 12 hours. Inspection of figures 6.6 and 6.7 confirms this consistency.

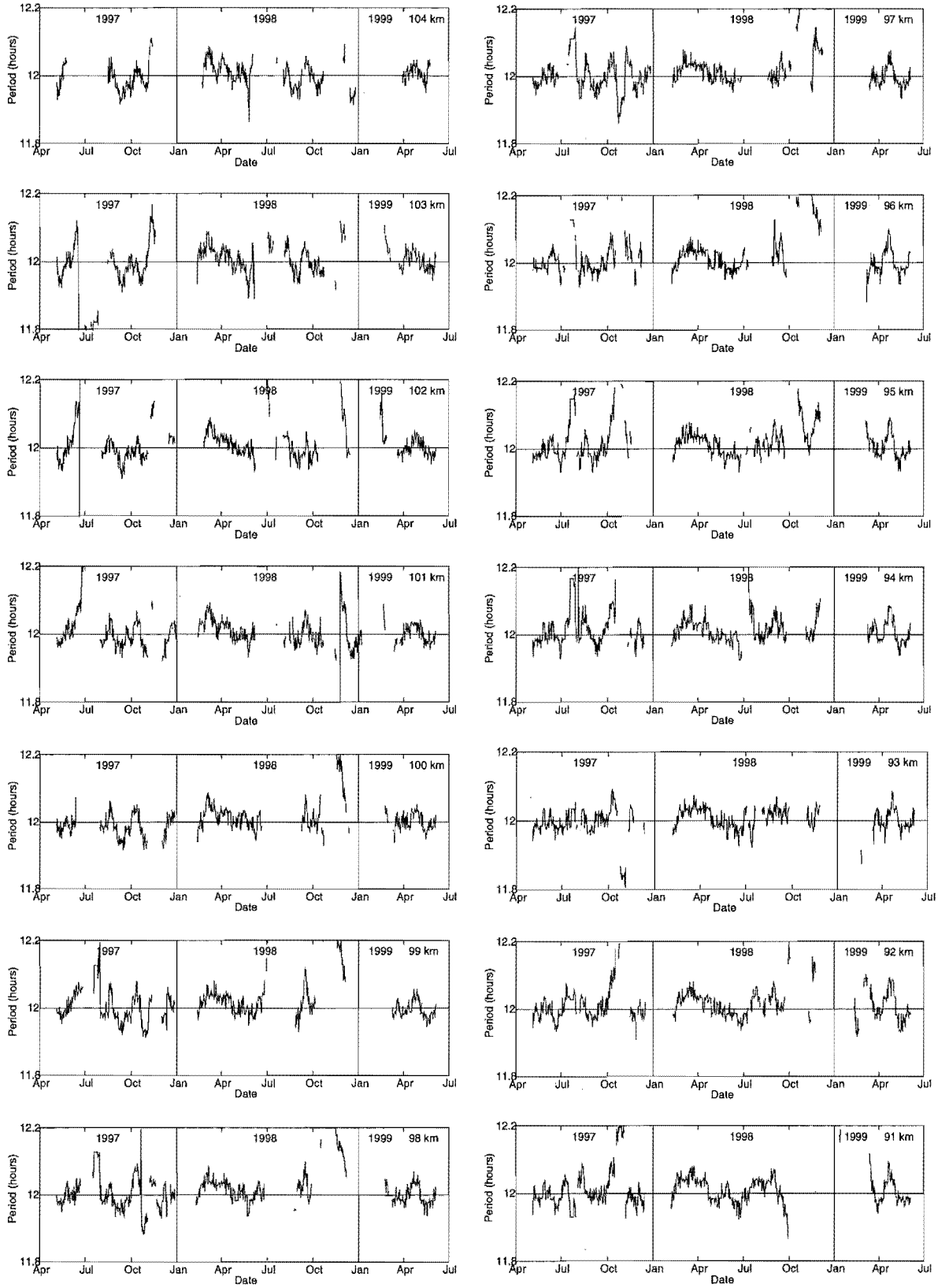

Figure 6.7: Period of the semidiurnal tide as determined at the heights indicated in each graph.

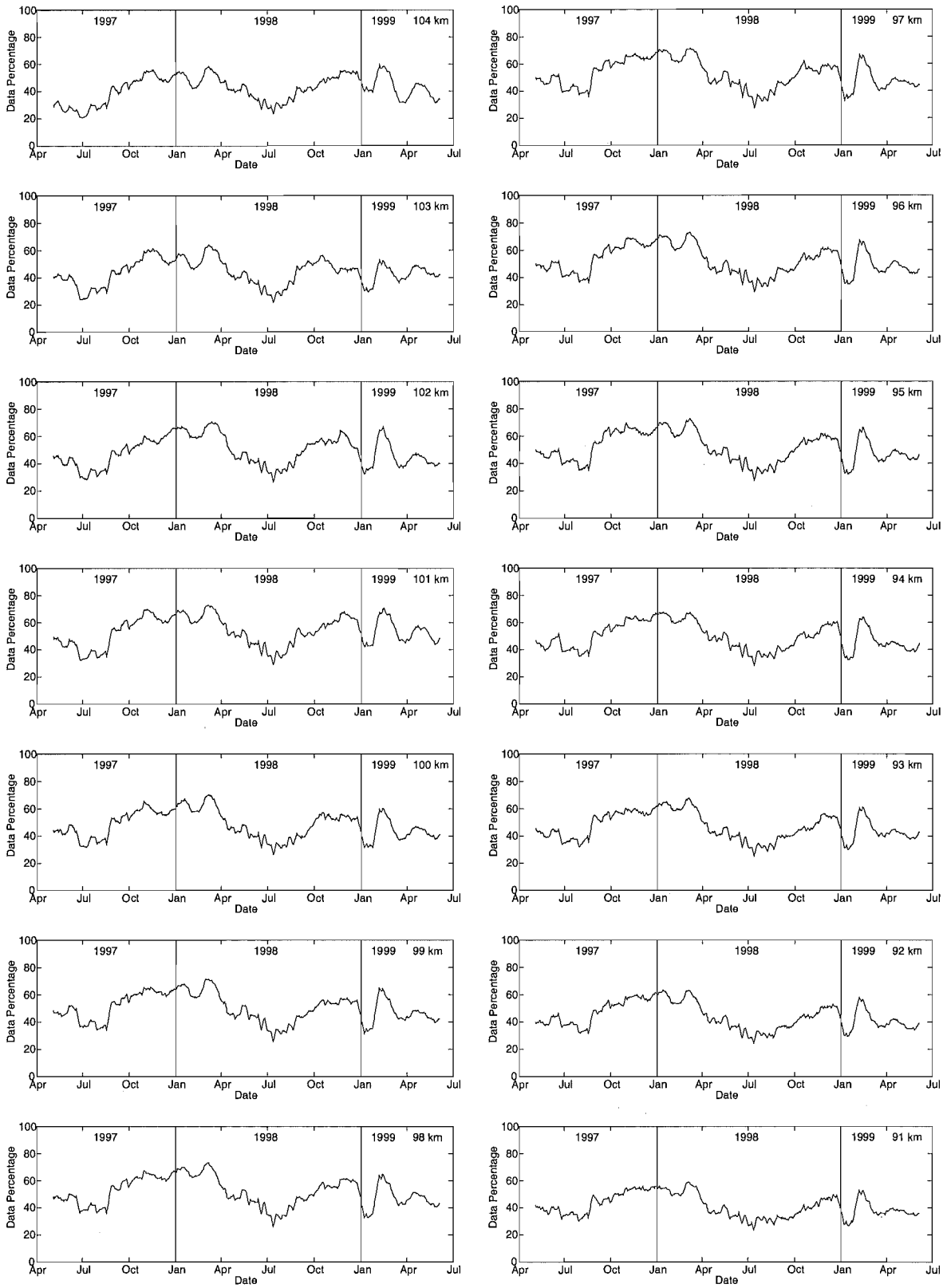

Figure 6.8: Percentage of observation period for which hourly means were determined.

6.4 Semidiurnal Tide Vertical Wavelength

The AMOR system measures a change in the phase of the semidiurnal tide with height. This is indicative of a vertically propagating component of the tide with a vertical wavelength equal to the distance through which the phase changes by 2π. The vertical wavelength (λ_v, km) has been obtained by calculating the gradient (m, hr km^{-1}) of the semidiurnal tide's phase with height, using a linear least squares fit (refer figure 6.9) and the relationship

$$\lambda_v = \left| \frac{12}{m} \right| .$$

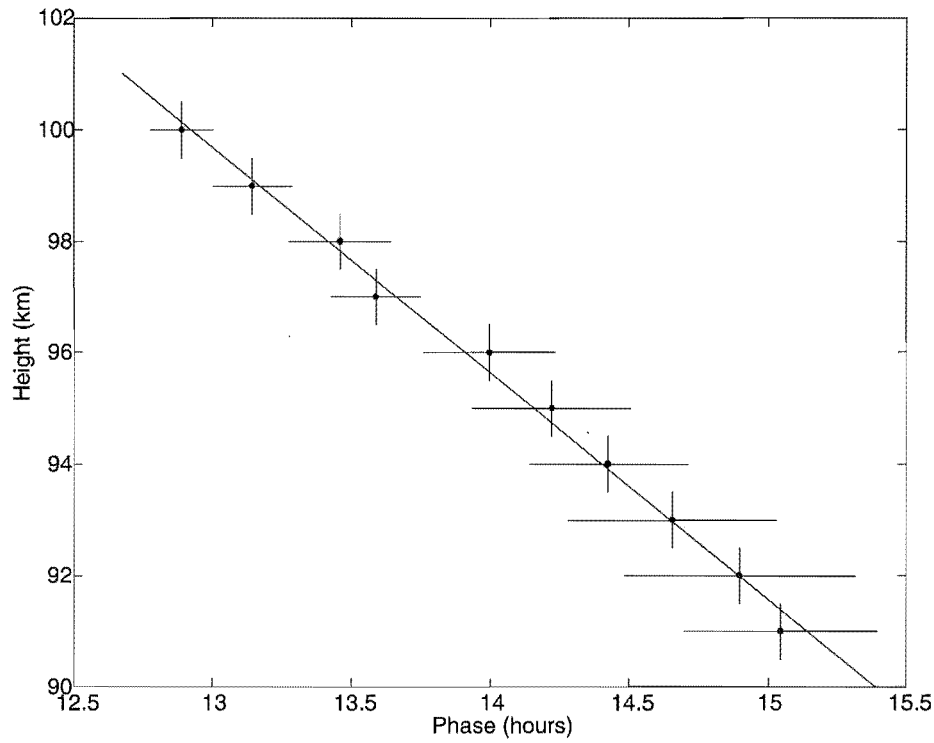

Figure 6.9: An example of the linear fit to semidiurnal tide phase as a function of height. In this example the straight line gradient is -3.9±0.1 km hr^{-1} ($m = -0.26$ hr km^{-1}) which implies a vertical wavelength of 47±1 km.

A negative value for m occurred more than 70% of the time and simply indicated that the wave was propagating upwards. The vertical wavelength of the semidiurnal tide obtained from AMOR meteor wind data during the period 19 March 1997 to 21 June 1999 is shown in figure 6.10.

Figure 6.11 from Andrews et al. [1987] shows the relationship between the

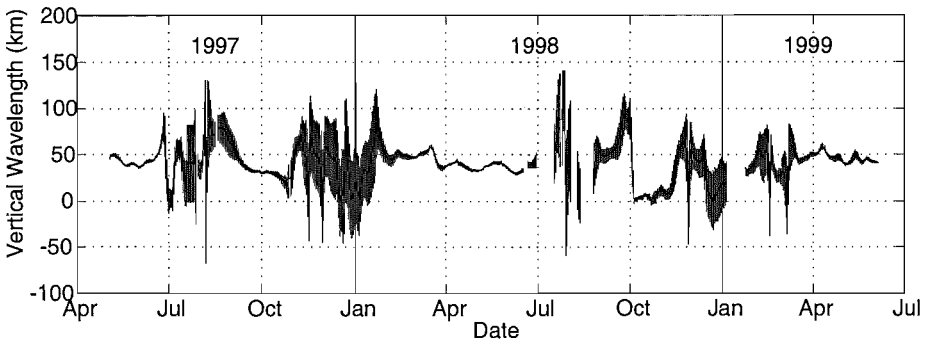

Figure 6.10: The vertical wavelength of the semidiurnal tide. Uncertainties in the wavelength are drawn on the graph and appear as a thickening of the line. From these values dominant tidal modes can be interperated.

semidiurnal tide's vertical wavelength (λ_v) and the atmosphere's equivalent depth or scale height (h). In addition table 6.1 [Lindzen & Chapman 1969] shows the dependence of the dominant tidal mode on h. Therefore a measurement of the semidiurnal tide's vertical wavelength can be used to interpret the dominant tidal mode at that time. From figure 6.10 the vertical wavelength was in the order of 35–45 km during autumn and winter months and reference to figure 6.11 and table 6.1 suggests that the most probable modes present were (2,4) and (2,5). On occasions λ_v increased to a value nearer 80 km (Aug 1997, Oct 1998) and indicates that (2,3) was the dominant mode for the semidiurnal tide during those times.

Interestingly, at no stage was λ_v measured as being large enough (>200 km) to indicate the presence of the preferred state due to thermal excitation, mode (2,2); this suggests that at lower heights the exponential growth of (2,2) was being interrupted and 'coupled'(Lindzen & Hong [1974], Walterscheid & Venkateswaren [1979a],Walterscheid & Venkateswaren [1979b], Walterscheid et al. [1980]) into higher order modes [Forbes 1982].

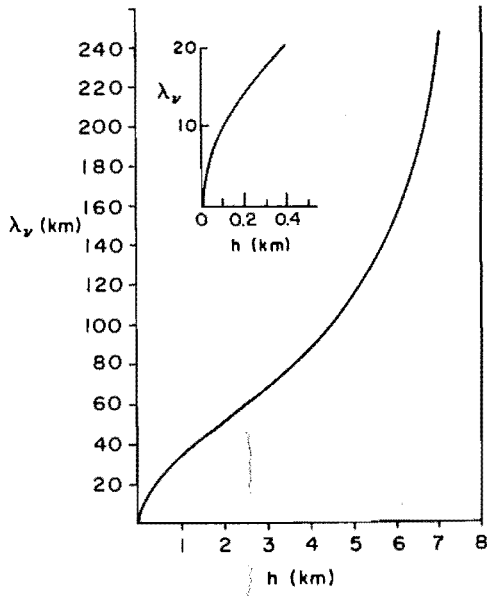

Figure 6.11: Vertical wavelength as a function of equivalent depth (After Andrews et al. [1987]).

Mode	h (km)	Mode	h (km)
(2,2)	7.85	(2,3)	3.67
(2,4)	2.11	(2,5)	1.37
(2,6)	0.96	(2,7)	0.71
(2,8)	0.54	(2,9)	0.43
(2,10)	0.35	(2,11)	0.29
(2,12)	0.24	(2,13)	0.21
(2,14)	0.18	(2,15)	0.16
(2,16)	0.14	(2,17)	0.12

Table 6.1: Equivalent depths for different modes of semidiurnal tide (After Lindzen & Chapman [1969]).

6.5 Acceptance Layer Thickness

As a consequence of the semidiurnal tide generally having such a short vertical wavelength, analysis of its amplitude can produce values which are smaller than they should be if the thickness of the height layer from which the data are hourly averaged is significant in relation to the vertical wavelength.

Amplitudes obtained for the semidiurnal tide at 95 km from meteor winds data averaged over a variety of height layer thicknesses are shown in figure 6.12. The blue line shows the amplitude obtained when the acceptance layer is 1 km; the highest resolution possible with the AMOR system. This amplitude is constantly greater than that obtained when the acceptance height layer thickness was broadened to 9 km (the red line). Extending the acceptance layer such that all meteors (70–120 km) are included in the calculation of the hourly mean values results in the green amplitude profile.

From these data it can be seen that the 9 km and 1 km acceptance layer thicknesses did give good general agreement, however when all data were included the calculated amplitude was greatly reduced; occasionally down to a factor of 0.5. There are occasions indicated in figure 6.12 when the three layer acceptance thicknesses produced amplitudes which were consistent with each other, e.g. November 1997, July 1998 and March 1999. These appeared to be when the tide's amplitude was small and, more importantly, coincided with times of large vertical wavelength (refer figure 6.10). Conversely the largest differences in calculated semidiurnal tide amplitude occurred when the vertical wavelength was calculated as small, e.g., April – June 1998.

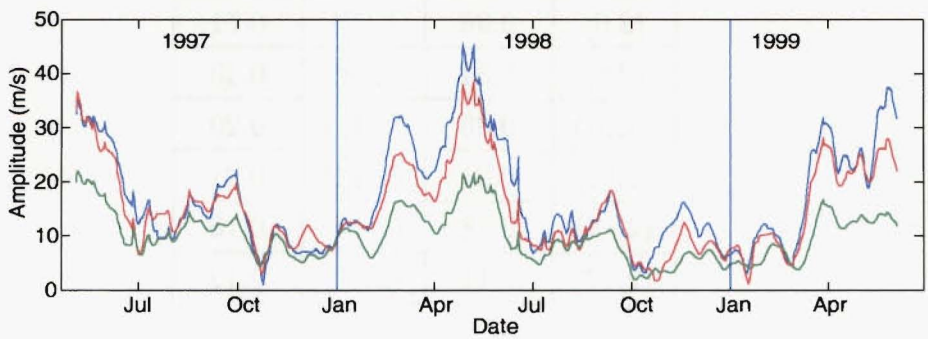

Figure 6.12: Amplitude of the semidiurnal tide as measured from layers of various thicknesses at a height centered at 95 km. The blue line has been obtained from an acceptance layer only 1 km thick, the red line is the amplitude measured if the acceptance layer is extended to 9 km thick and the green line from analysis of data at all heights.

6.6 Planetary Waves

Planetary or Rossby waves are global scale oscillations with periods of a few days. Possible planetary wave activity has been detected in the AMOR winds data set from analysis with the Lomb-Scargle Periodogram on 20 day windowed data[2]. Least squares was used to parameterise and remove the dominant component and then the process was repeated to obtain the second most dominant (subdominant) component. Figure 6.13 shows the dominant (top graph) and second most dominant (bottom graph) periodicities obtained from this process and identifies the presence of occasional short lived atmospheric oscillations which have periods of 2, 4 and 5 days.

From this figure it can be seen that in general the dominant behaviour was the semidiurnal tide, however, during November 1997 four-day wave activity was seen gradually to gain dominance over the semidiurnal tide; appearing first as a subdominant signal prior to becoming the clearly dominant signal for about one week. Its amplitude then receded and the semidiurnal tide returned as the dominant oscillation.

During January 1998 the two-day wave was clearly evident for about one month (The apparent length of this event was extended by the window width). The onset of the two-day wave appeared to occur very quickly as it gained dominance over the semidiurnal tide in a short period of time, however there is a suggestion that the decay of the two-day wave was slower as it appeared as a subdominant once the semidiurnal tide regained dominance.

There was a possibility that a six or eight-day wave was in a formation phase during November 1998 but the period of any signal appeared erratic and further analysis would be required to confirm its existence. Furthermore, weak five-day wave activity was also apparent as a subdominant in February 1998. This feature was seen again, though slightly later, the following year.

The amplitude profiles for possible planetary waves with periods of 2, 4, 5, 6, 8 and 10 days were calculated with a sliding window LSFT2 and are shown in figure 6.14. The two-day wave was clearly evident during 1998 however its existence was doubtful for 1999; this was most likely caused by reduced data, due to radar

[2]In this case a 20 day window was used because of the a priori knowledge of four and five day wave existence.

down-time, coinciding with its expected maximum.

The amplitude of the four-day component was shown in figure 6.14 to generally vary quite erratically over short time periods. However during November 1997, the time identified as containing a four-day component in figure 6.13, the amplitude profile shows a slowly amplitude varying signal. This is good evidence for the existence of a stable four-day wave during this time and appeared to be the only four-day wave event detected by the AMOR system during the observation period.

The five-day wave amplitude profile shown in figure 6.14 shows large stable amplitudes at times consistent with times identified in figure 6.13 and confirmed the possibility of a five-day wave occurring during February 1998 and March 1999.

Amplitude profiles for six and eight-day components have been produced in figure 6.14, however, from inspection, it appears most unlikely that these planetary waves exist in the AMOR winds data set as there isn't any clear indication that either of the six or eight-day wave amplitude profiles ever show any stability[3]. The possible ten-day event identified in July 1997 by figure 6.13 seems unlikely to be physically real, considering this evidence alone, as it appears to occur over such a short time period. However, an inspection of the ten-day amplitude profile in figure 6.14 does suggest a large amplitude was stable during that time and provides tentative support for its presence.

[3]Six and eight-day wave amplitude profiles were obtained with 18 and 24 day windows respectively.

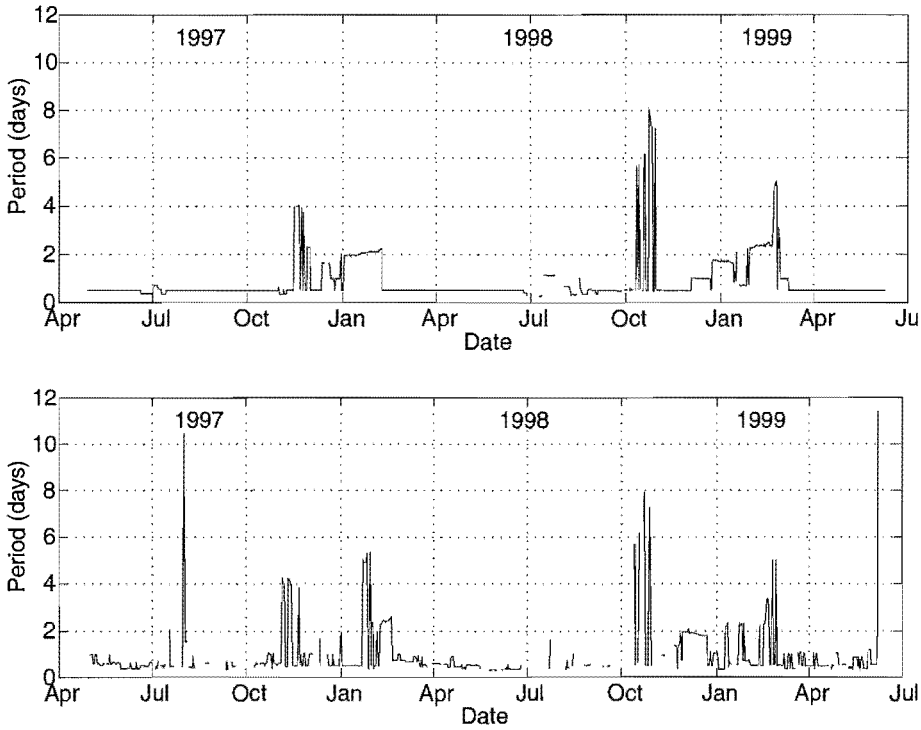

Figure 6.13: Most dominant periods in upper graph and next dominant in lower graph.

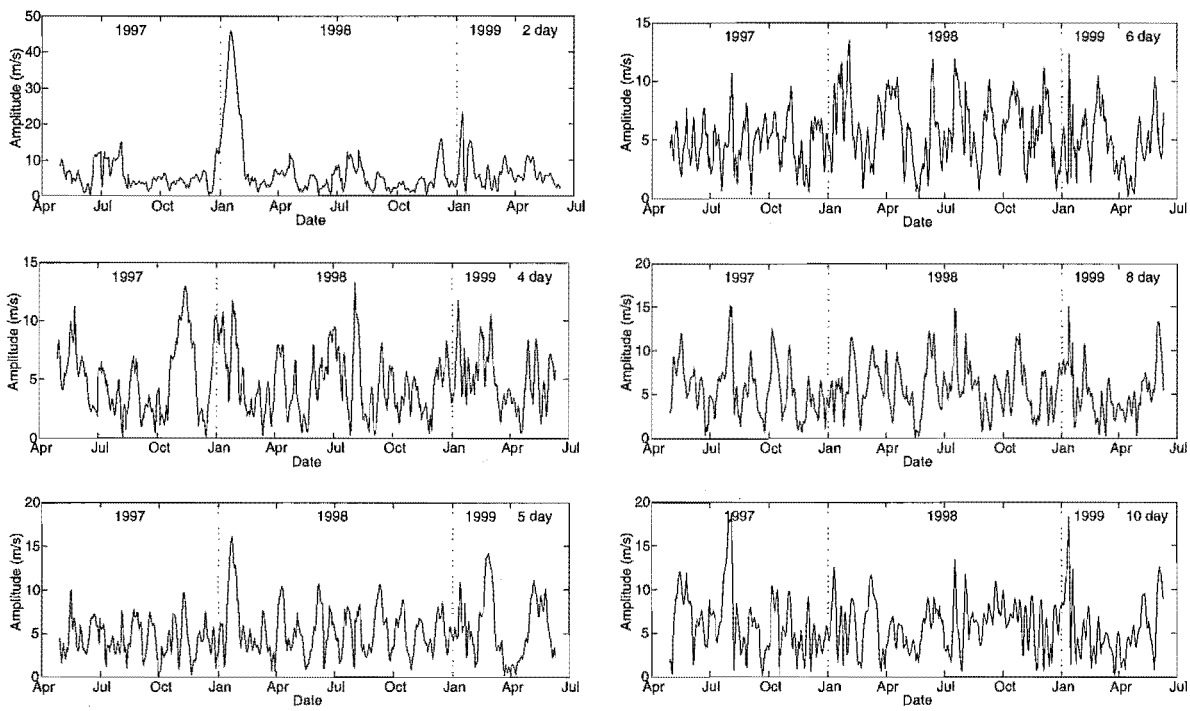

Figure 6.14: Amplitude of planetary waves.

6.7 Mean Wind

The mean wind profiles obtained from the AMOR wind data set at heights between 91±0.5 and 104±0.5 km are shown in figure 6.16. These graphs indicate that the mean wind has no significant height dependency as all graphs show good general agreement. A general description of the measured mean wind would be that it was generally directed towards the north and that the amplitude of this motion was largest in summer and smallest in winter. During winter months the mean wind was small and appeared to change direction frequently.

The maximum northward mean wind (~15 $m s^{-1}$) was measured during the summer months of December and January each year. The mean then decreased in February and March down to a value of ~-5 $m s^{-1}$ in late March or early April. The mean wind was then seen to exhibit fairly erratic behaviour during the months of April through to October before returning to a stable large northerly. The transition from the erratic winter behaviour to the large amplitude stable summer behaviour occurred around late October in both 1997 and 1998 and the transition back occurred during March for both 1998 and 1999.

The mean wind obtained from AMOR winds data, when analysed with a 30 day window, has been summarised in figure 6.15.

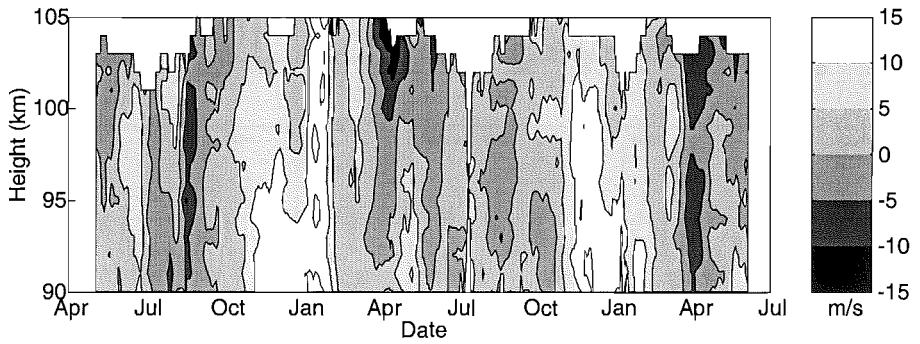

Figure 6.15: The mean wind as measured by the AMOR meteor radar with a 30 day sliding window.

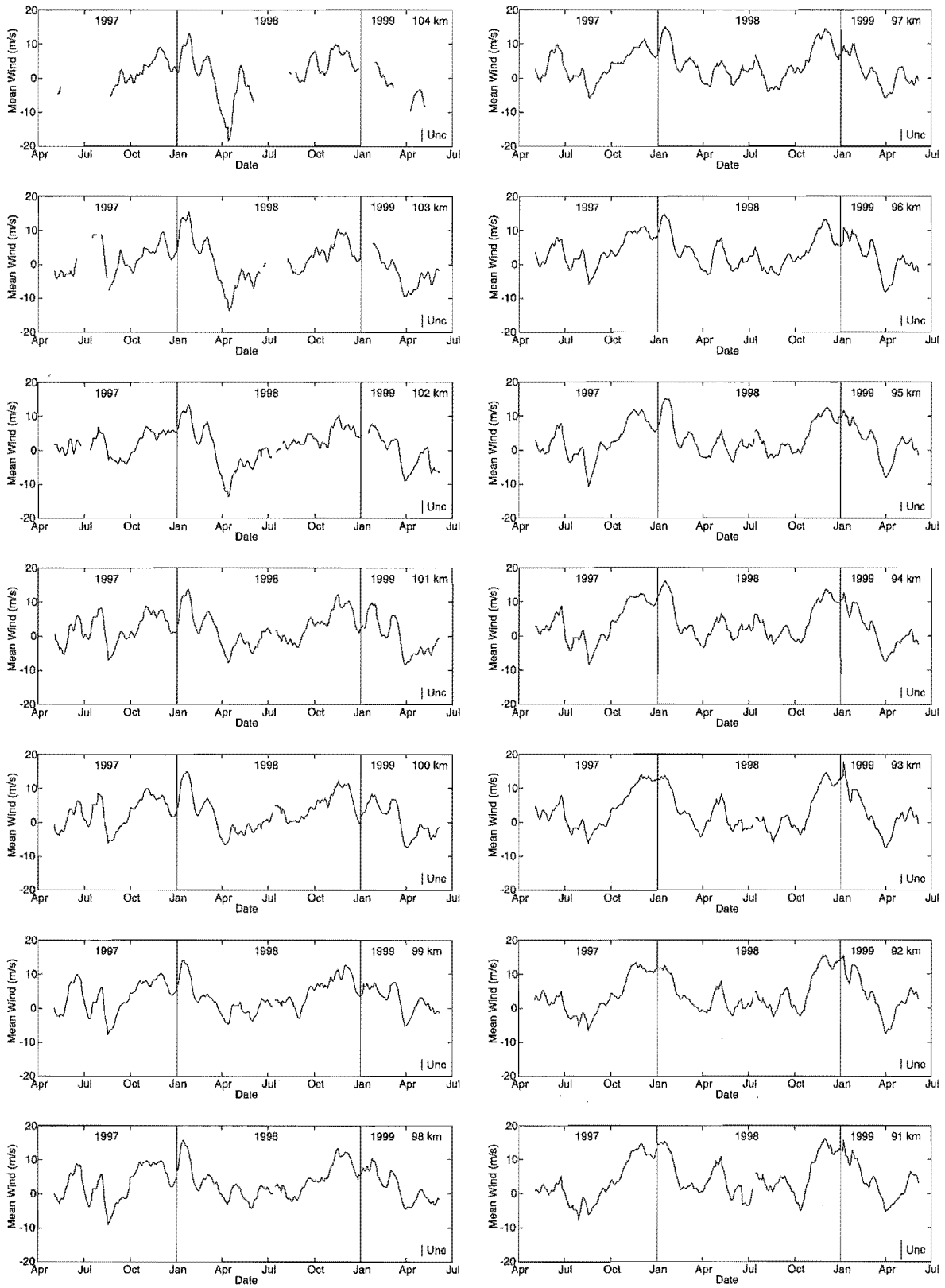

Figure 6.16: Value of the mean wind calculated from AMOR meteor radar data with a 30 day window at various heights.

Chapter 7

Data Comparisons with the AMOR Winds Data Set

In this chapter data from the Fabry-Perot Interferometer (FPI) at Mount John and the Medium Frequency (MF) Radar at Birdlings Flat are compared with data from the AMOR meteor radar. The three instruments sample different volumes of the atmosphere (refer figure 7.1) using different methods.

The FPI [Hernandez 1986] measures wind by observing the Doppler shift of the night-time sky emission of the OH(6–2)$P_1(2)$ transition at 840 nm and the atomic oxygen line (OI) at 557.7 nm. The instrument observes in the four cardinal directions at an angle of 20° as well as in the zenith.

The MF radar measures wind by the partial-reflection technique [Fraser 1984]. Winds are calculated from the ground level observed motion of diffraction patterns caused by partial reflections from ionisation in the 70–105 km height region of the atmosphere. The observation field for the MF radar is a vertical cone with a half-power full-beam width of 32° by 18° perpendicular and parallel to the north/south direction resulting in a sampling volume of 48 by 37 km at a height of 95–100 km located directly above the Birdlings Flat site [Smith 1996].

This chapter also details the intra-comparison of AMOR meteor radar data, which are grouped into bins by ground range, thus removing any systematic differences present in the inter-experiment comparisons. The number of Ground Range Bins (GRBs) is variable and results for sixteen GRBs, 32.5 km wide, eight in each direction ranging from 40 km either side of the Home site to a maximum of 300 km in both north and south directions, are presented here. The few echoes detected at ranges less than 40 km from the Home site are included in the bins nearest the Home site.

Finally, semidiurnal tide amplitude and phase values are compared with data

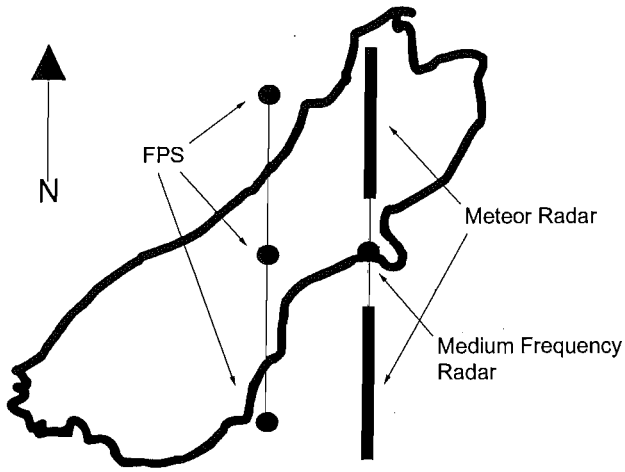

Figure 7.1: The South Island of New Zealand showing the relative sampling volumes of the three wind measuring methods.

produced by the Global Scale Wave Model (GSWM) [Hagan et al. 1999] at equinox.

7.1 Meteor Winds and FPI Comparison

7.1.1 FPI Height Measurements

As shown in Chapter 6 wind speed varies with height and hence good agreement from wind measurement comparisons requires that the data are sampled from similar heights. As the FPI cannot directly measure air-glow layer heights, data from the green (557.7 nm) line of atomic oxygen and the red (840 nm) hydroxyl (OH) line are typically assumed to be from heights of 97 km and 87 km respectively. For the wind comparisons with data from the FPI in this work [Plagmann et al. 1998], a more accurate estimate of the airglow layer heights has been obtained by comparing the phase of the semidiurnal tide calculated at each airglow height from FPI data with the phase of the semidiurnal tide over a range of heights from 80–110 km (with a 1 km height resolution) obtained from the AMOR meteor radar winds data. This technique takes advantage of the semidiurnal tide's phase being height dependent (refer section 6.4) and the airglow layer height is defined as being the height, measured by AMOR, for which the semidiurnal phases are nearest.

Heights produced by this method were seen to vary slightly on a daily basis.

Heights fluctuated around 100 km for the atomic oxygen layer and 92 km for the OH layer. The heights obtained were slightly higher than is usually quoted (+3 km and +5 km for the green and red lines respectively) however they do lie within the range of heights obtained from rocket measurements [Baker & Stair 1988]. In addition, increased heights measured in this way were to be expected as these airglow layer heights are indicative of the layer's average emission height and this is greater than the height of the profile peak: the value which is most often quoted in the literature.

As an FPI wind measurement is made from the received emission integrated over a layer several kilometres thick (refer figure 7.2), data are selected from the AMOR data set for comparison weighted to approximate this. O'Brien et al. [1965]describes the airglow layers as similar in shape to a Pascalian triangle with a FWHM of 6–10 km. Therefore, a Pascalian weighted vertical profile function which is 9 km thick and centred at the height identified with semidiurnal phase matching is applied to the AMOR meteor winds data. For comparison the 557.7 nm night airglow profile of O'Brien et al. [1965]and the applied weighting function are presented in figure 7.3.

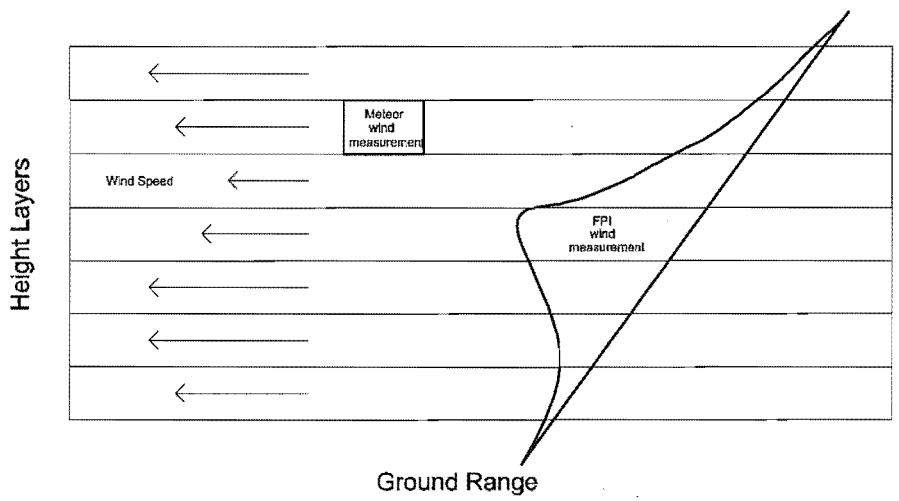

Figure 7.2: Sampling regions for meteor radar and FPI. The radar makes a localised wind measurement (within an 8 km^3 3 dB box) whereas the FPI measurement represents the wind speed integrated throughout a volume.

Wind speed measurements cannot be made by the FPI during daytime hours due to saturation of the interferometer by solar radiation and as a consequence comparisons for full day's data cannot be achieved. FPI wind measurements are

further reduced during cloudy periods due to clouds diffusing the airglow emissions.

The data comparison presented in this work has been confined to data collected during the period 2–6 May 1997 during which there were five consecutive clear sky nights. Comparisons of AMOR and FPI data from an additional eleven days between June 1997 and April 1998 were presented and discussed in Plagmann et al. [1998].

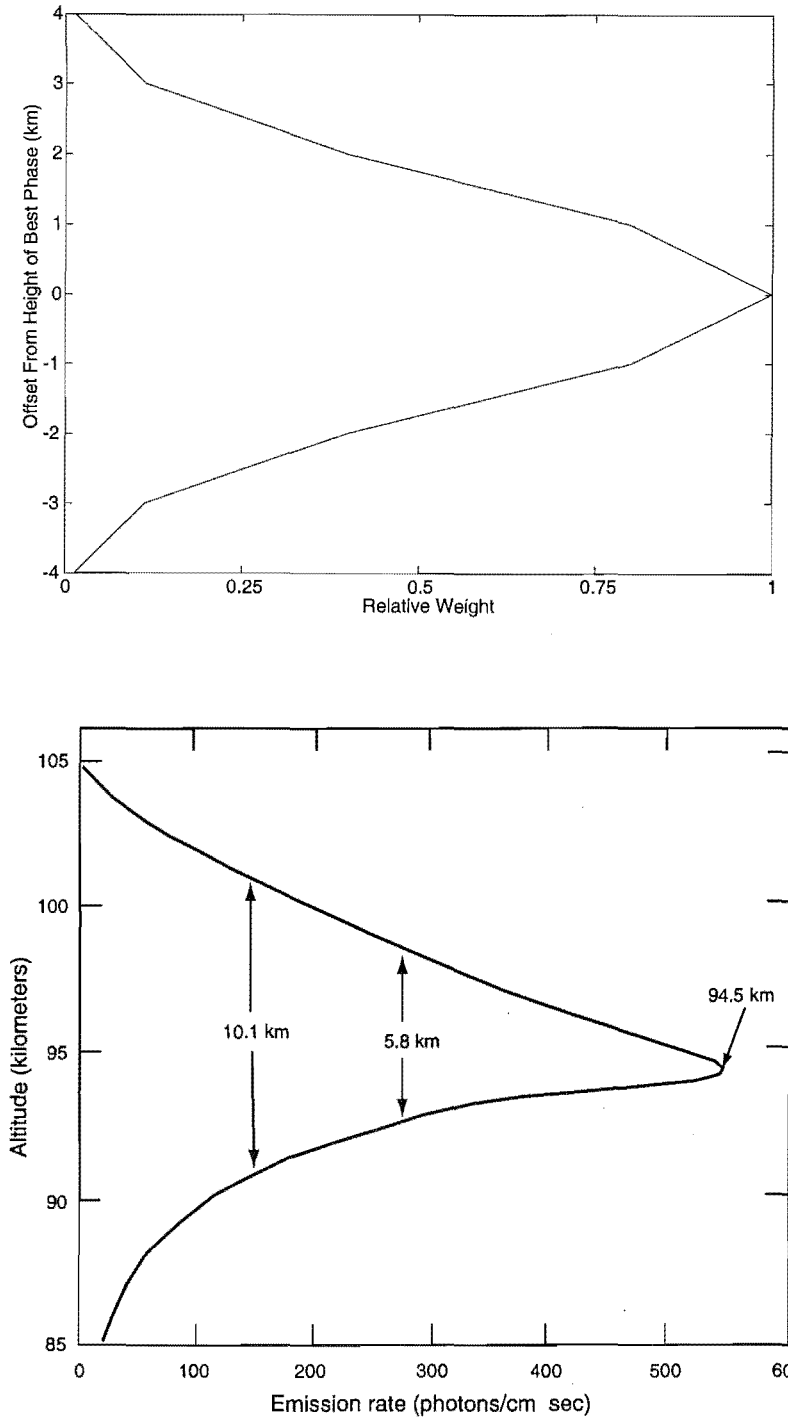

Figure 7.3: The top figure shows the weighting function applied to the meteor winds data. The bottom figure shows altitude profile of 557.7 nm night airglow [O'Brien et al., 1965].

7.2 Meteor Winds and MF Radar

Like the AMOR meteor radar, the MF radar is capable of measuring winds 24 hours per day, however poorer signal to noise is experienced at night (the noise level is 20dB greater at night [Smith 1996] and is due to increased interference coupled with less ionisation) and a poorer data rate results. As a result the best agreement between the AMOR meteor winds and the MF radars is to be expected during daytime hours and much poorer agreement may be expected at night.

Similar to the AMOR system, the MF radar can measure winds over a range of heights (70–105 km) with height measuring ability. The MF radar is capable of measuring winds in both zonal and meridional directions; a feature which is shared with the FPI (the AMOR meteor radar can measure the meridional component only). Again, for comparison with the FPI, an airglow emission profile weighting function has been applied to the MF data.

7.3 Results

Wind speeds as measured by the two Canterbury University radars and the FPI at Mount John are shown in the following figures. The data presented are hourly means where blue traces are from the meteor radar, the green traces the MF radar and red traces are from the FPI. Time of day is given in hours UT to centralise the FPI data set in the time period.

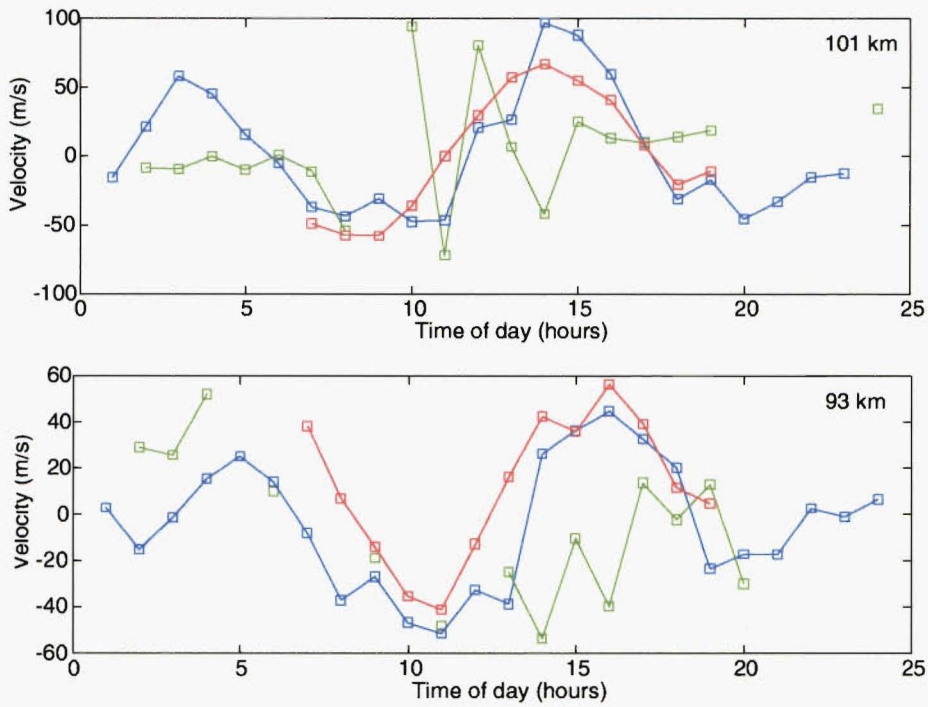

Figure 7.4: Comparison of AMOR (blue line) FPI (red line) and MF radar (green line) hourly (NZST) mean winds for 2 May 1997.

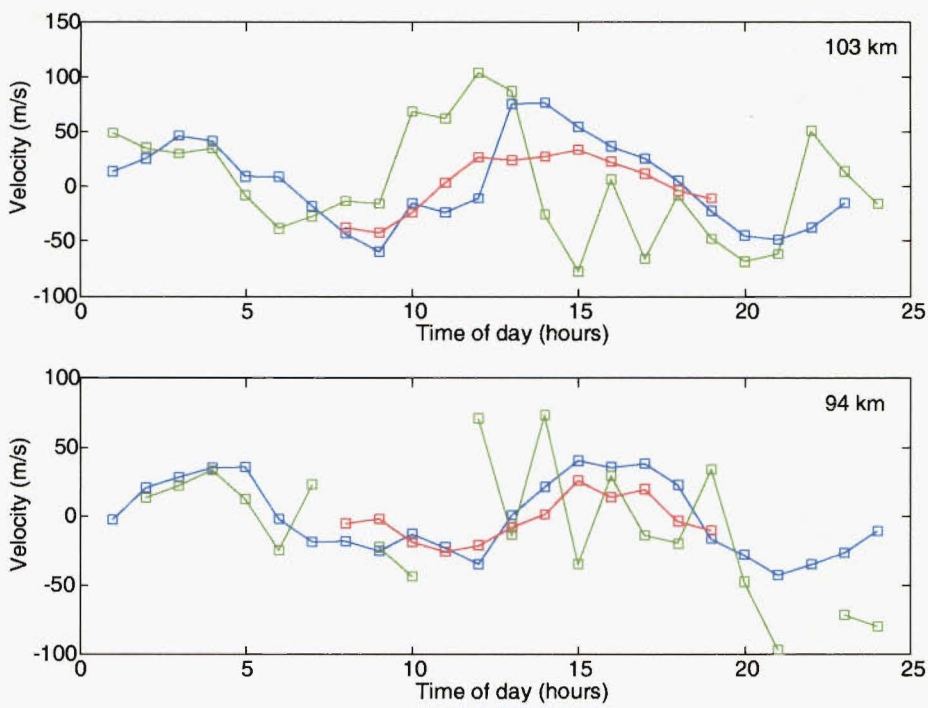

Figure 7.5: Comparison of AMOR (blue line) FPI (red line) and MF radar (green line) hourly (NZST) mean winds for 3 May 1997.

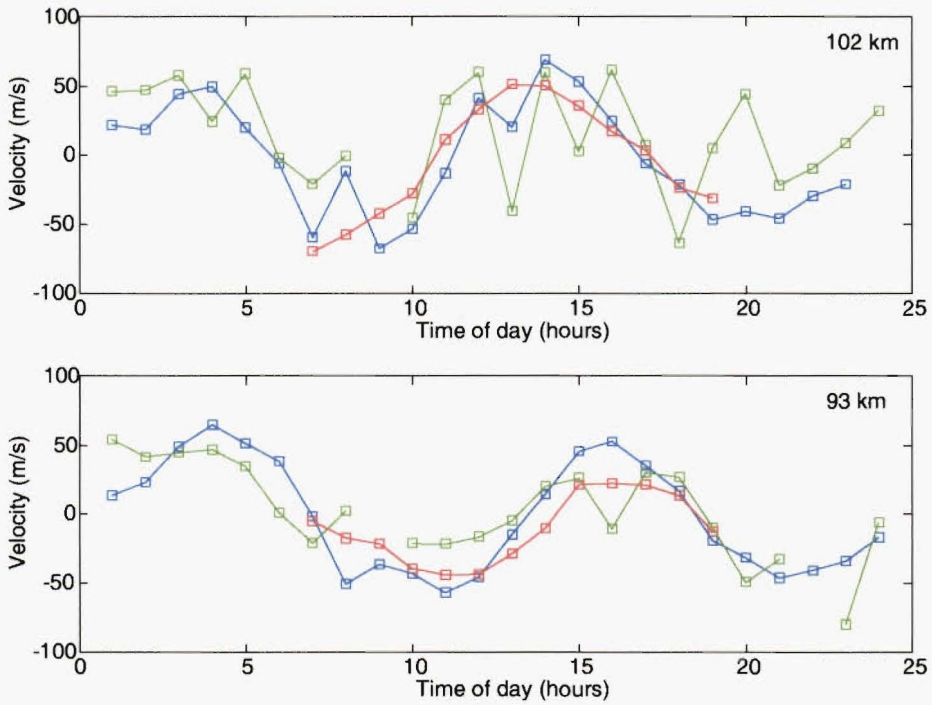

Figure 7.6: Comparison of AMOR (blue line) FPI (red line) and MF radar (green line) hourly (NZST) mean winds for 4 May 1997.

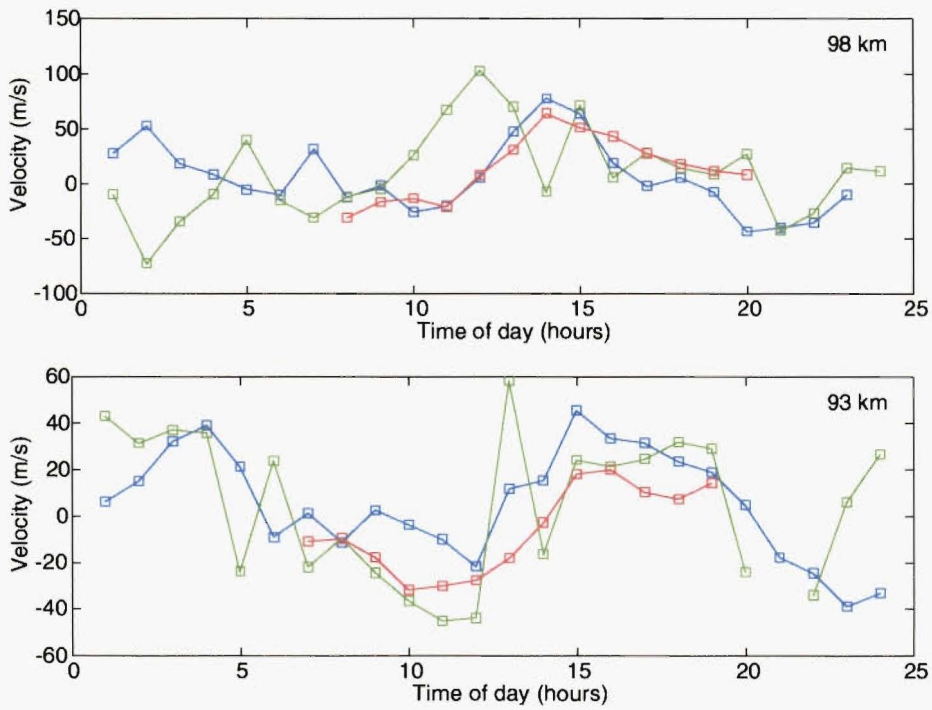

Figure 7.7: Comparison of AMOR (blue line) FPI (red line) and MF radar (green line) hourly (NZST) mean winds for 5 May 1997.

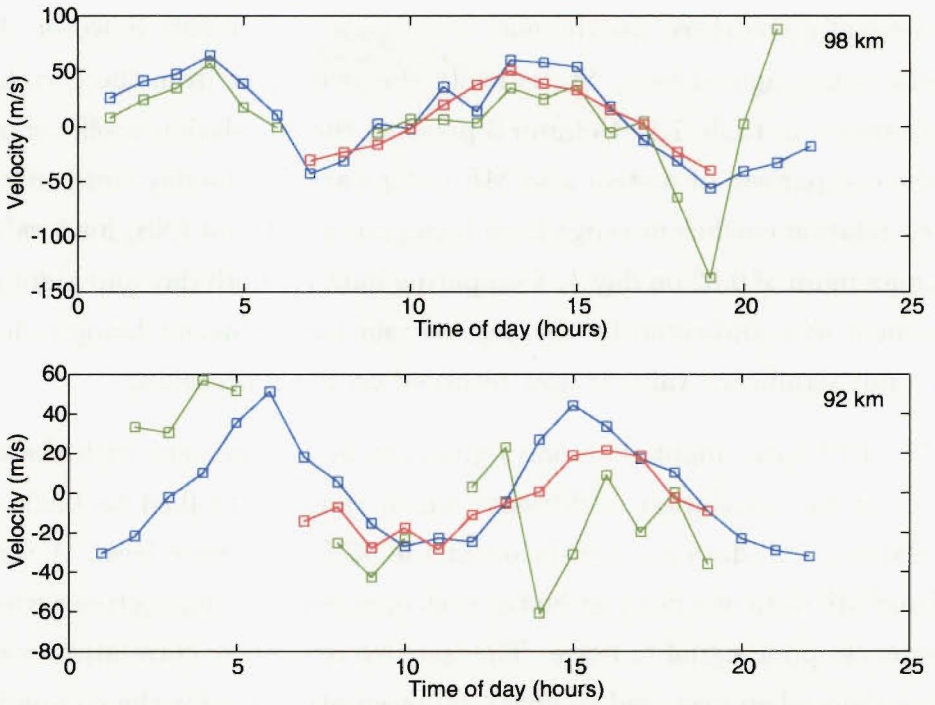

Figure 7.8: Comparison of AMOR (blue line) FPI (red line) and MF radar (green line) hourly (NZST) mean winds for 6 May 1997.

Generally the three instruments show good agreement; however the FPI and meteor radars agreed best. Statistically this is evident from the correlation coefficients shown in table 7.1. Column 3 presents the correlation coefficients calculated for the comparison of meteor and MF radar data for the day time only. Values for the correlation coefficient range from being not significant (NS) for 5 values through to a maximum of 0.82 on day 4. Comparing data for both day and night gave poorer agreement as is indicated by the best correlation coefficient being reduced to 0.71 and a not significant value is now returned on seven occasions.

The FPI data (night time only) gives the best agreement with the meteor data set returning correlation coefficients which range from 0.84 to 0.95 with no NS correlations. All data are significant at the 95% confidence level. A comparison of FPI and MF data was not justified as each operates best during times when the other experiences poor signal to noise. The last two rows show correlation coefficients for all five days when averaged to give a representative day for the comparison period. One sees again that on average the meteor radar agrees better with the FPI and is not significant with the MF radar at greater heights. Both instruments show better agreement with the meteor radar at lower heights. This is primarily due to the FPI and MF radars recording more data at these heights.

Date	Height (km)	MR-MF (day)	MR-MF all	MR-FPI
2	93	0.56	NS	0.85
	101	NS	NS	0.89
3	94	0.80	NS	0.86
	103	0.58	NS	0.90
4	93	0.82	0.71	0.95
	102	0.63	0.57	0.89
5	93	NS	NS	0.89
	98	NS	NS	0.86
6	92	NS	NS	0.84
	98	NS	0.56	0.93
all	lower	0.82	0.63	0.95
	higher	0.61	NS	0.85

Table 7.1: Correlation coefficients for wind comparison. NS refers to no significant correlation at the 95% level.

7.4 Wind Speed Variability with Ground Range

Correlation analysis of hourly mean winds has been applied to wind measurements grouped into 16 Ground Range Bins (GRBs). To reduce any effect of the semidiurnal tide's vertical structure, only meteors detected at heights between 92±0.5 km and 102±0.5 km have been included in this analysis. In addition GRB comparisons hourly mean values were only calculated if six or more echoes were used in determining its value.

The 8 GRB centres for each direction are situated 32.5 km apart with the bin nearest the Home site centred 56 km north and south and the furthest centred 284 km north and south of Birdlings Flat. Data from the northern beam are grouped as a mirror image of the south (refer figure 7.9). The closest two GRBs centres either side of the Home site are separated by the larger distance of 112 km. This is because the radar detects very few echoes at high elevation angles.

If one assumed a near isotropic atmosphere then hourly means of data in these GRBs should correlate well. Conversely poor correlation could either indicate flaws in the measurement technique or identify inhomogeneities in the measured wind field.

Tabulated values of correlation coefficient, gradient of line of best fit and number of coincident hourly means for all available data are presented in Appendix B. For perfect agreement between GRBs the correlation coefficient and the best fit gradient should both equal one.

GRBs located near either end of the sampling volume (GRBs 2 and 15) are defined as "reference bins". The remaining bins which are to be compared with the reference bins are called "comparison bins". By observing the changes in correlation between reference and comparison bins one can determine the ground range wind speed variability. The GRBs second from the ends of the sampling volumes were selected in preference to GRBs 1 and 16 simply because they provided more wind measurements. In this thesis the derived correlation values between each reference bin and the comparison bins has been termed a reference bin correlation profile. Naturally correlation analysis is only applied to hourly means for reference and comparison bins which are coincident.

Reference bin correlation profiles for AMOR winds data classified by season are

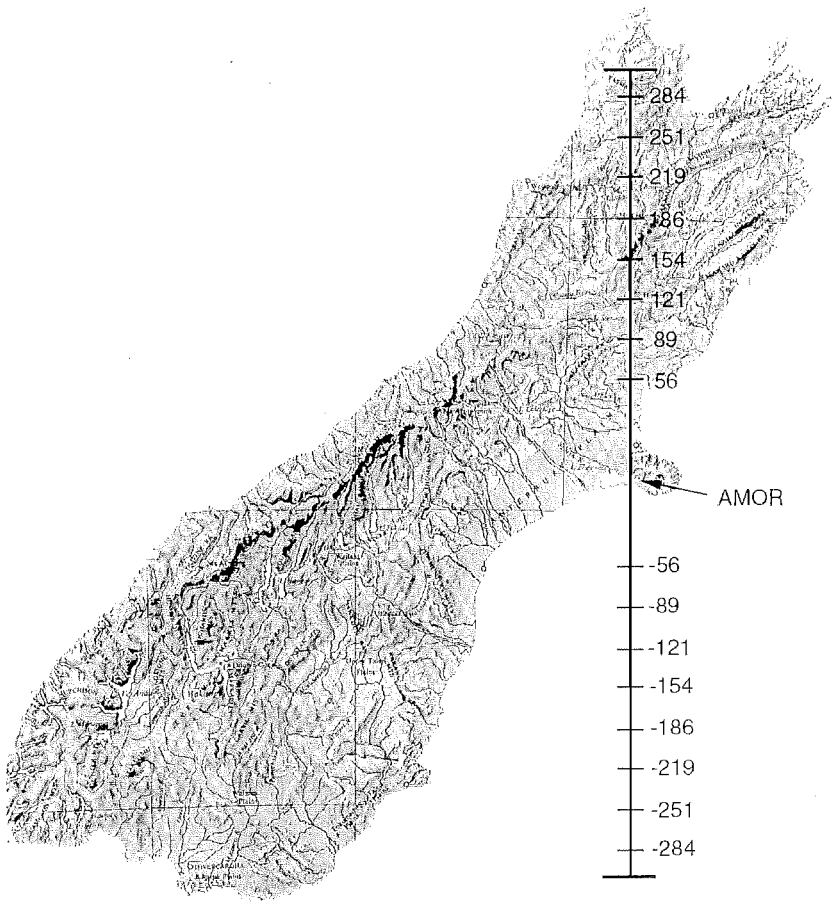

Figure 7.9: The South Island of New Zealand showing the positioning of, and distance to, ground range bin centres along the north/south line. SCALE: 1cm = 60 km.

shown in figure 7.10. In these graphs the solid circles and open circles indicate whether the reference bin was GRB 2 or 15 respectively. The uncertainties for these correlation coefficients were calculated by applying a Fisher transformation to the correlation coefficients. This produced a statistic $Z = \frac{1}{2}\ln\left(\frac{1+r}{1-r}\right)$ which has a distribution nearly gaussian and a mean given by $\mu_z = \frac{1}{2}\ln\left(\frac{1+\rho}{1-\rho}\right)$ and standard deviation $\sigma_z = \frac{1}{\sqrt{n-3}}$ where r and ρ are sample and population correlation coefficients respectively and n is the number of independent data points.

Correlation values have only been displayed in figure 7.10 if there were more than 20 hourly means available for analysis. This explains why GRBs near the Home site (ground range = 0) do not provide correlation coefficients during winter[1].

[1]There is an annual variation in meteor rate such that the number of meteors is greatest in autumn and smallest in spring (southern hemisphere).

The graphs in figure 7.10 show, as expected, that the correlation coefficient gradually reduces as the distance to the comparison bin increases. However, perhaps the most obvious feature of these graphs is the sudden reduced correlation with GRBs at the near north side of the Home site. The cause of this feature is not completely understood; however several possibilities will be discussed later in this section.

Another feature of these graphs is a seasonal effect. The southern correlation coefficients for the southern reference GRB (solid circles) remained consistent at approximately 0.8±0.1 throughout the year, yet the correlation coefficients for southern comparison GRBs with the northern reference GRB displayed a seasonal cycle. In summer months the correlation coefficients between southern comparison GRBs and the northern reference GRB had values of 0.7±0.1, this value decreased in autumn to 0.6±0.1. In winter the correlation coefficient values decreased further to 0.4±0.2 before increasing in spring back to a value of 0.6±0.1. Gravity waves are a possible explanation for this effect.

Gravity waves of horizontal phase velocity, c, cannot propagate upwards through a background wind, u, of the same velocity. Orographic gravity waves are phase locked to their source so their phase speed, $c = 0\ \mathrm{m\,s^{-1}}$ and they cannot propagate upwards through regions where the background wind $u = 0\ \mathrm{m\,s^{-1}}$. That is to say that orographically generated gravity waves should not be detected above boundaries between westerly and easterly flow.

Zonal mean winds (and temperature) under both solstice and equinox conditions as measured by the Upper Atmosphere Research Satellite (UARS) are shown in figures 7.11 and 7.12. At a latitude of 45°S the January zonal mean wind profile exhibits a westerly to easterly transition at a height of approximately 25 km and as a result theory predicts gravity waves will not be observed at meteoric heights. Similarly the October profile prevents gravity waves travelling much higher than 60 km. April and July show constant westerly behaviour to the top of the mesosphere and would permit gravity waves to travel to approximately 100 km unaffected.

This process of gravity wave filtering is therefore a possible mechanism to explain the observed seasonal correlation dependence. Seasons which coincided with allowed vertical propagation of gravity waves into the high mesosphere showed reduced agreement between southern comparison GRBs and the northern reference GRB and conversely seasons which do not facilitate vertical gravity wave propagation

showed good agreement.

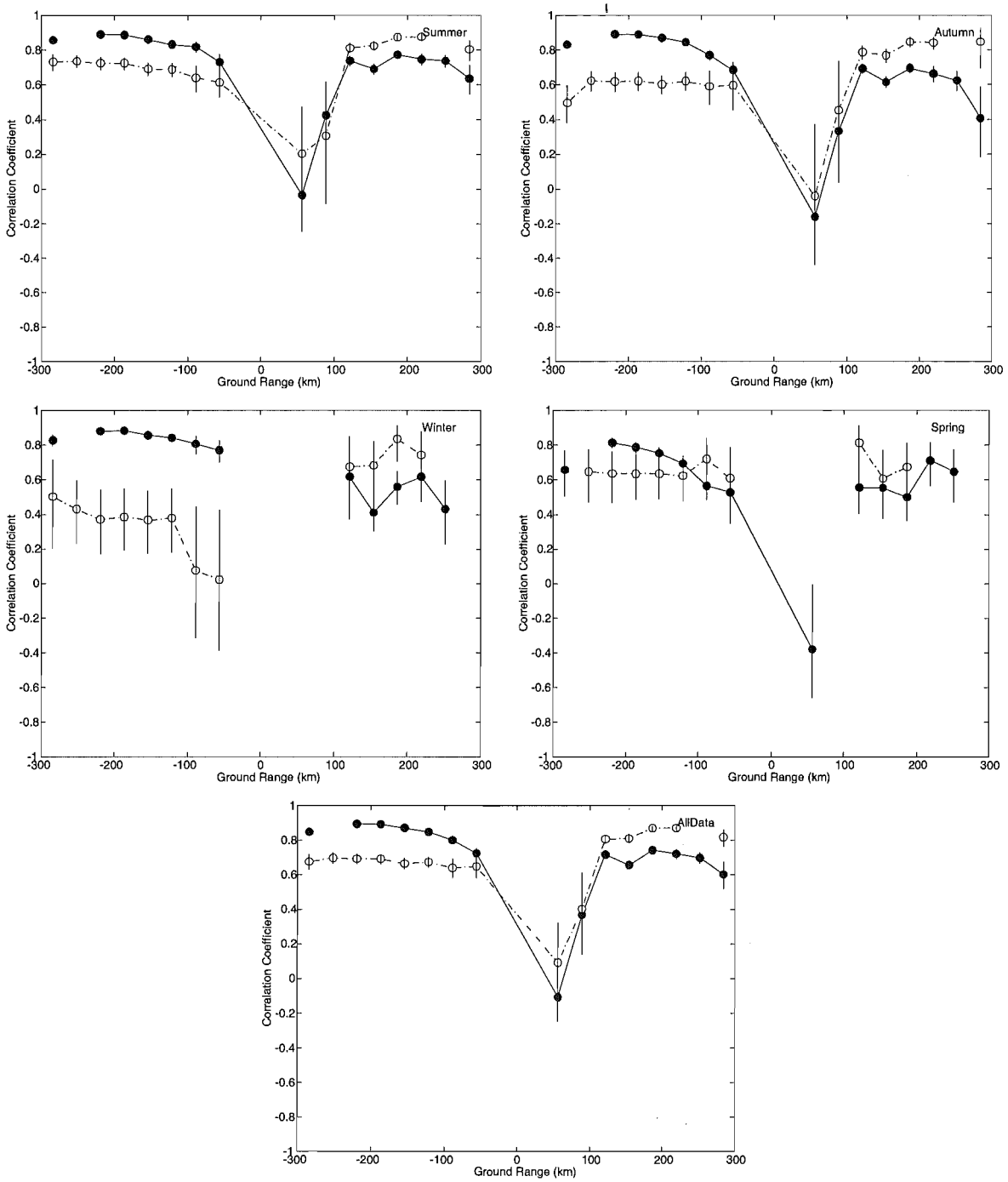

Figure 7.10: Correlation coefficients of GRB 2 (solid circles) and GRB 15 (open circles) with all GRBs at indicated ground range. A value is only displayed if 20 or more points are available. Positive ground ranges are north of the Home site and negative ground ranges are south. The Home site is at ground range =0 km. Uncertainty bars indicate 95% confidence limits.

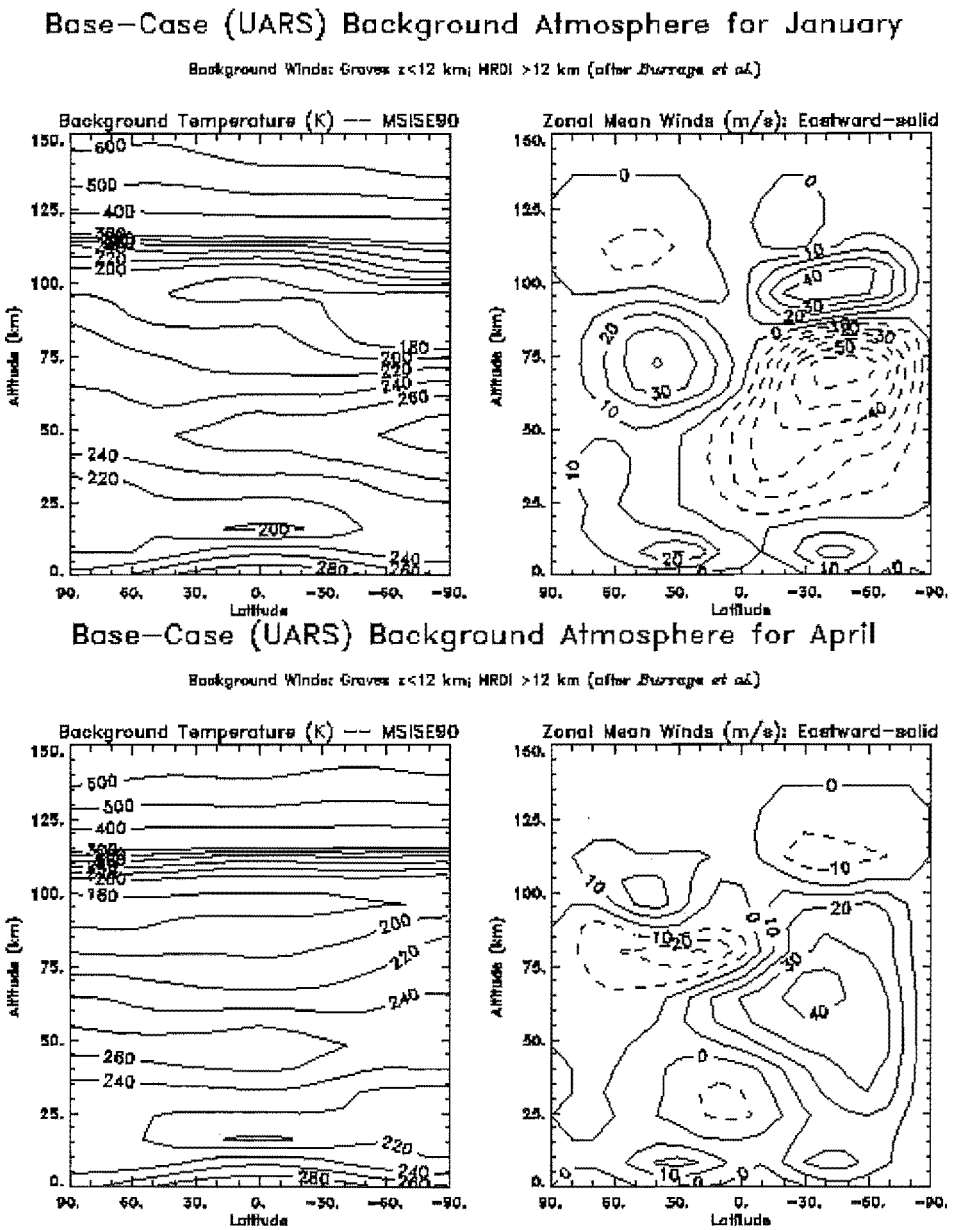

Figure 7.11: UARS temperature and zonal mean wind measurements for January and April.

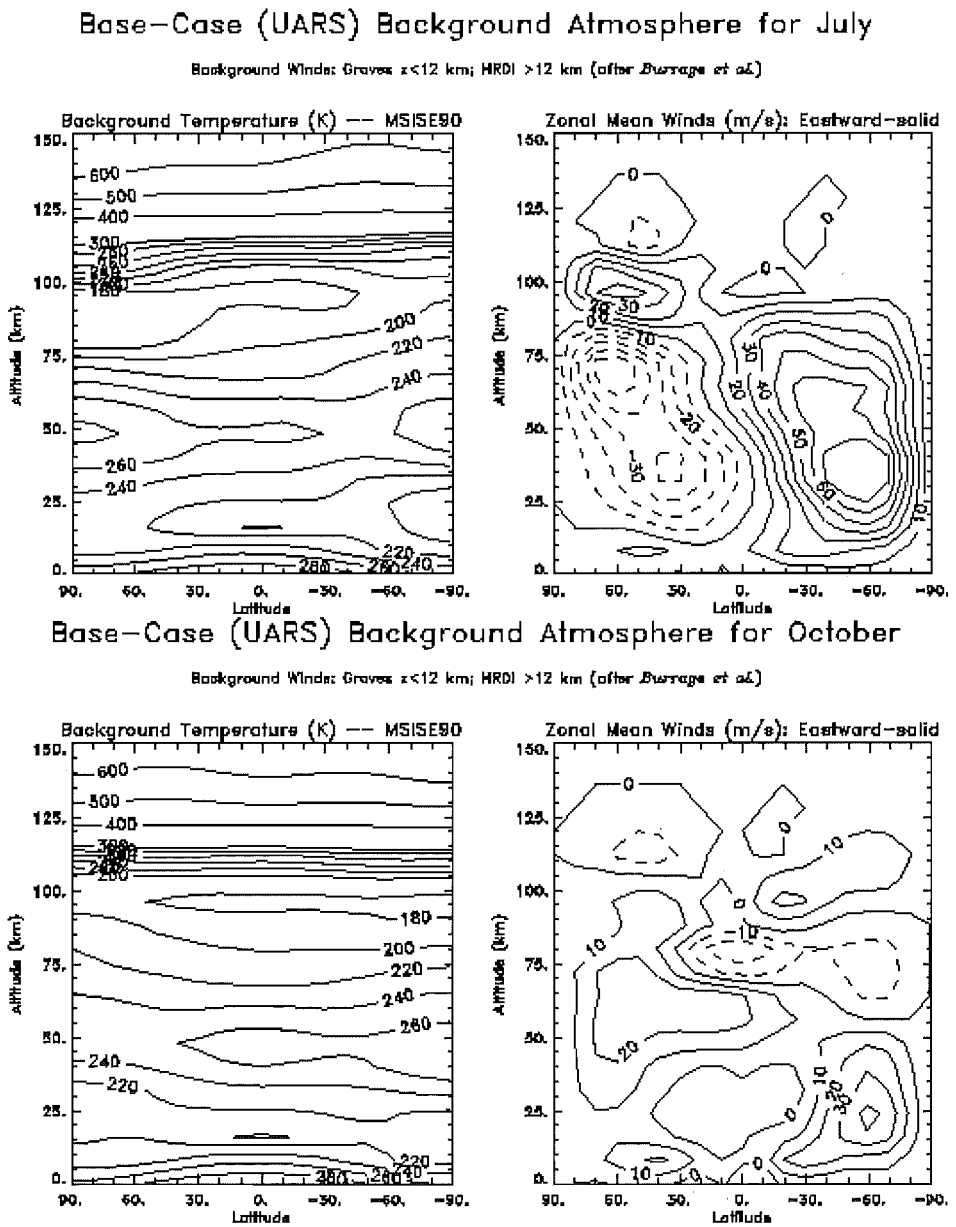

Figure 7.12: UARS temperature and zonal mean wind measurements for July and October.

As was briefly mentioned earlier in this section wind measurements from echoes in the northern near Home site GRBs exhibit poor correlation with measurements in all other GRBs. Three possible explanations for this poor agreement are:

- that the meteor radar contains a systematic error which becomes apparent for echoes measured in this region.
- that the values simply reflect the poorer statistics in this region.
- that this region is experiencing extreme, local turbulence due to gravity wave breaking.

The first of these explanations is unlikely mainly as if it were true then similar behaviour would be expected in the southern near Home site GRBs. Nor can this effect be explained by $\mathbf{u} \times \mathbf{B}$ meteor train plasma drift as it would be expected that any effect of train drift orthogonal to wind direction would become steadily more apparent as the sampling volume tended further south.

The second possible explanation is considered by refering to figure 7.13 which shows that the region of interest does consistently collect the least number of echoes. This is due to a combination of fewer echoes being detected in the northern beam and these echoes being required to be at a high elevation angle; where the transmitted power is small in comparison with lower elevation angles. However there are occurrences of low measurement count outside of the region of interest which have nevertheless produced high correlation coefficients. Examples are the four northern beam winter correlations and the three northern beam spring correlations with the northern reference GRB. In these cases, despite the number of measurements being comparable to the number of measurements generally obtained for the near Home site northern GRBs, the correlation coefficients were of the order of 0.8 ± 0.2.

To explore the theory that the low correlation coefficients could have been produced by the low data rates, 30 mean values were randomly sampled from each GRB and the correlation analysis was repeated. Although the uncertainties on the values increased, as a reflection of there being fewer points, the results (figure 7.14) still indicated good correlation for all data except those in near Home site northern GRBs. These results were repeatable and suggested that the low data rate in the northern near Home site GRBs is not the cause of the poor correlations.

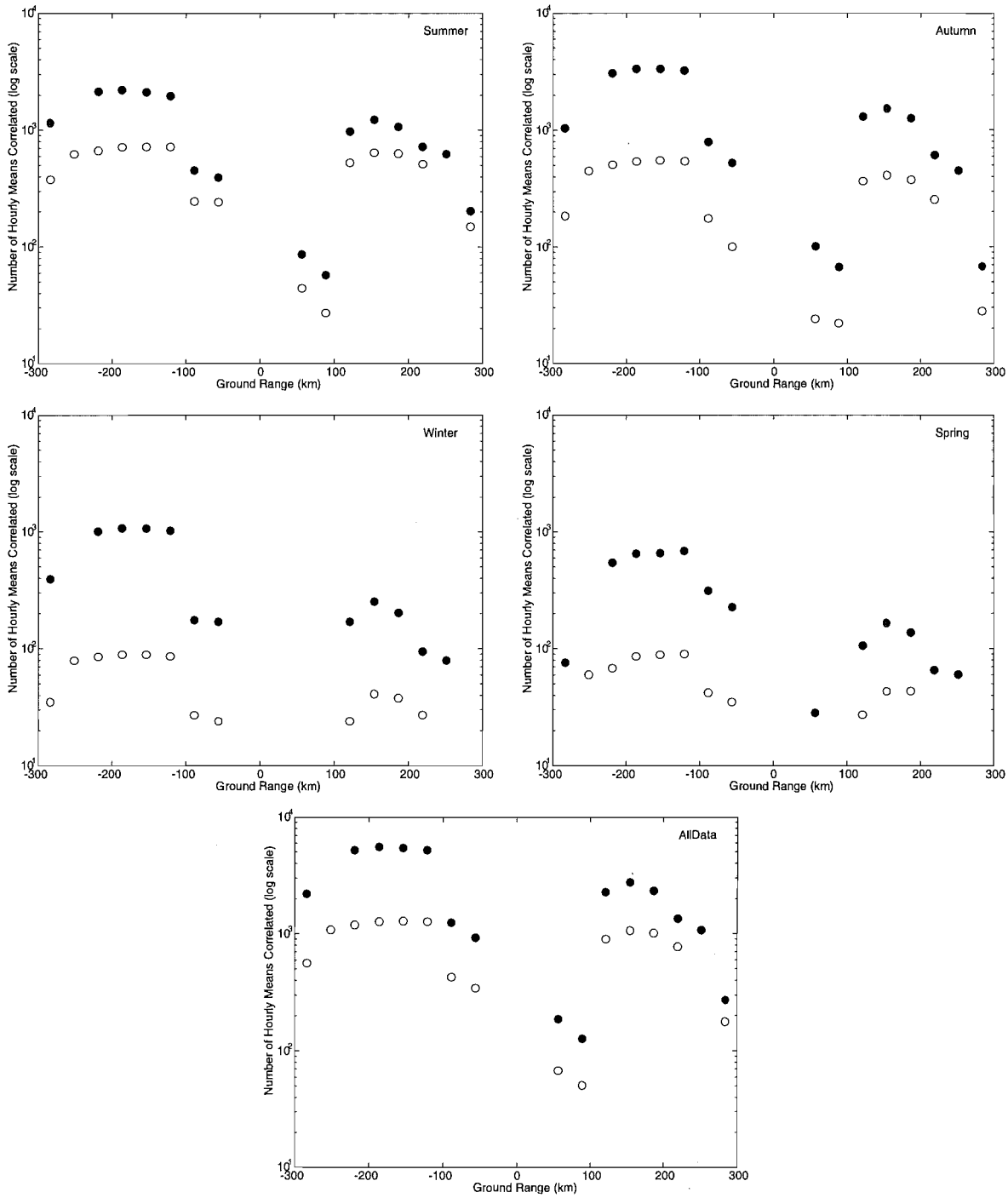

Figure 7.13: Seasonal dependence of number of hourly means available for correlalation analysis. Solid circles give the number of means for GRB 2 correlation and open circles are for GRB 15 correlations.

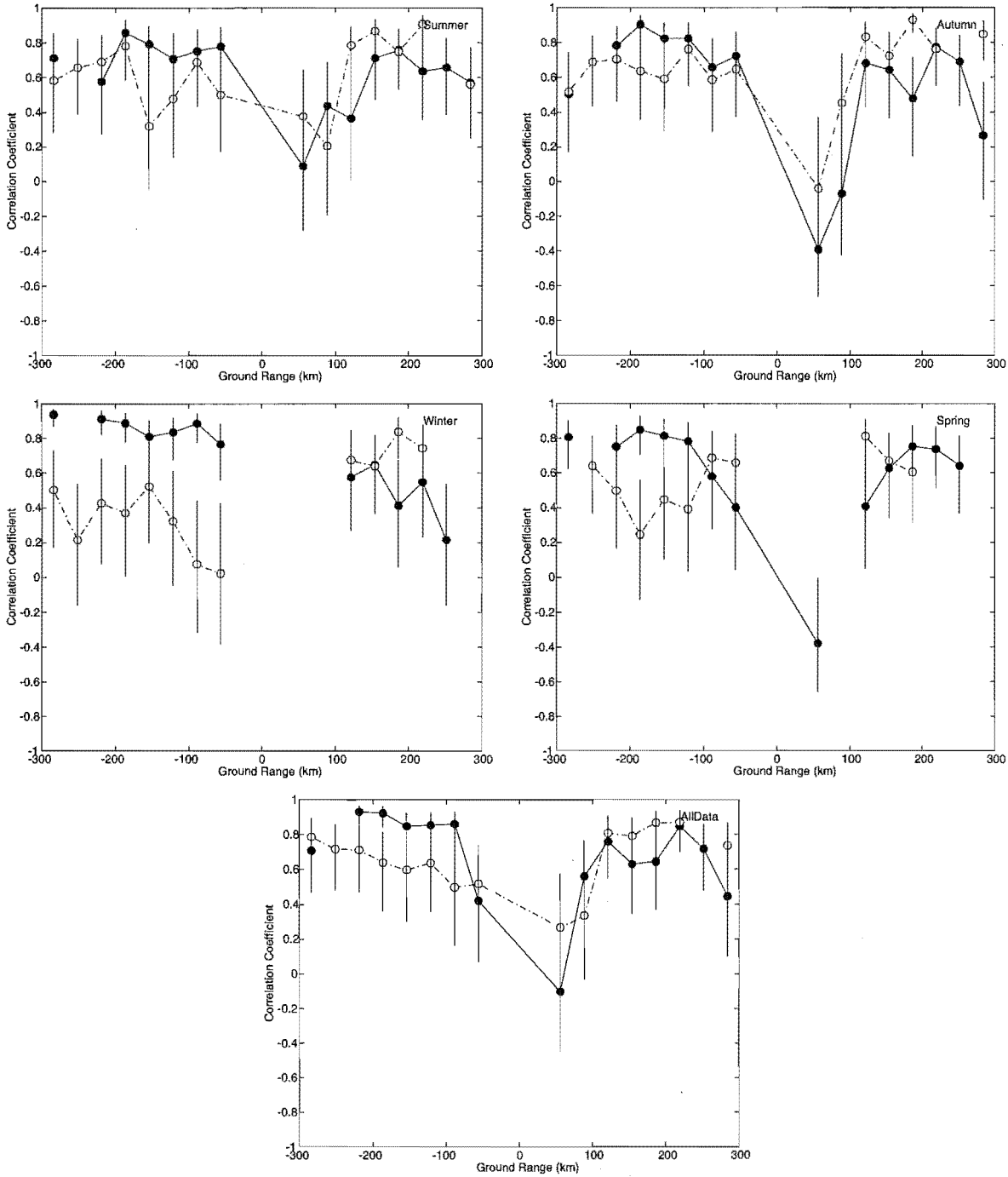

Figure 7.14: Correlation coefficients of GRB 2 (solid circles) and GRB 15 (open circles) with all bins at indicated ground range. A value is only displayed if 20 or more points are available. Data were randomly sampled to permit a maximum of 30 points. Uncertainty bars indicate 95% confidence limits.

The third possible explanation for the reduced correlation at northern ground ranges less than 100 km is gravity waves. This variation can be explained by the presence of gravity waves which have been produced by the Southern Alps. Airflow over mountains can generate gravity waves which are capable of propagating high into the Earth's atmosphere and it is believed that they would interact with the atmosphere at meteoric heights many kilometres downstream of the mountains [Schoeberl 1985].

A two-dimensional model simulation based on approximations to Navier-Stokes equations [Garcia & Prusa 1997] showed short horizontal wavelength gravity wave activity appearing at meteoric heights downstream of the forcing region under conditions of isotropic wind speed[2] (refer figure 7.15). These waves are seen to generate vigorous overturning of the atmosphere over a range of many kilometres centred at approximately 60 km downstream of the forcing region.

For the AMOR data set the likely orographic source for gravity waves is the Southern Alps which lie as a ridge in a south-west/north-east direction (refer figure 7.9) west of the Home site. The distance from the Southern Alps to the radar illuminated regions is largest at the southern extreme of the radar beam. This distance decreases as one moves northward and at $\sim$ 154 km north of the Home site the radar beam and Southern Alps cross and wind measurements are taken from directly above the mountain range. Moving further north the radar samples westward of the direction of the Southern Alps. Westward of the radar beam, at ground ranges between $\sim$ 70 and 154 km north of the Home site, the Southern Alps have reduced in height and gravity wave production is likely to be less in this region. The Southern Alps are at their largest west of the beam at ground ranges between $\sim$ 56 km north and $\sim$ -89 km south of the Home site.

The region of poorest correlation in figure 7.10 is the GRB centred 56 km north of the Home site. This sampling volume lies approximately 110 km east of the Southern Alps and Garcia & Prusa [1997] would suggest that highly turbulent gravity wave activity in this region is plausible. Their results would also suggest large gravity wave induced turbulence in neighboring GRBs further north (but not south as these sampling regions are too far downstream). There is tentative observational evidence in support of this from the next northern bin (centred at

[2]It should be realised that this assumption is unlikely to be true. However this model still provides general gravity wave behaviour and can be used as a comparison in this work.

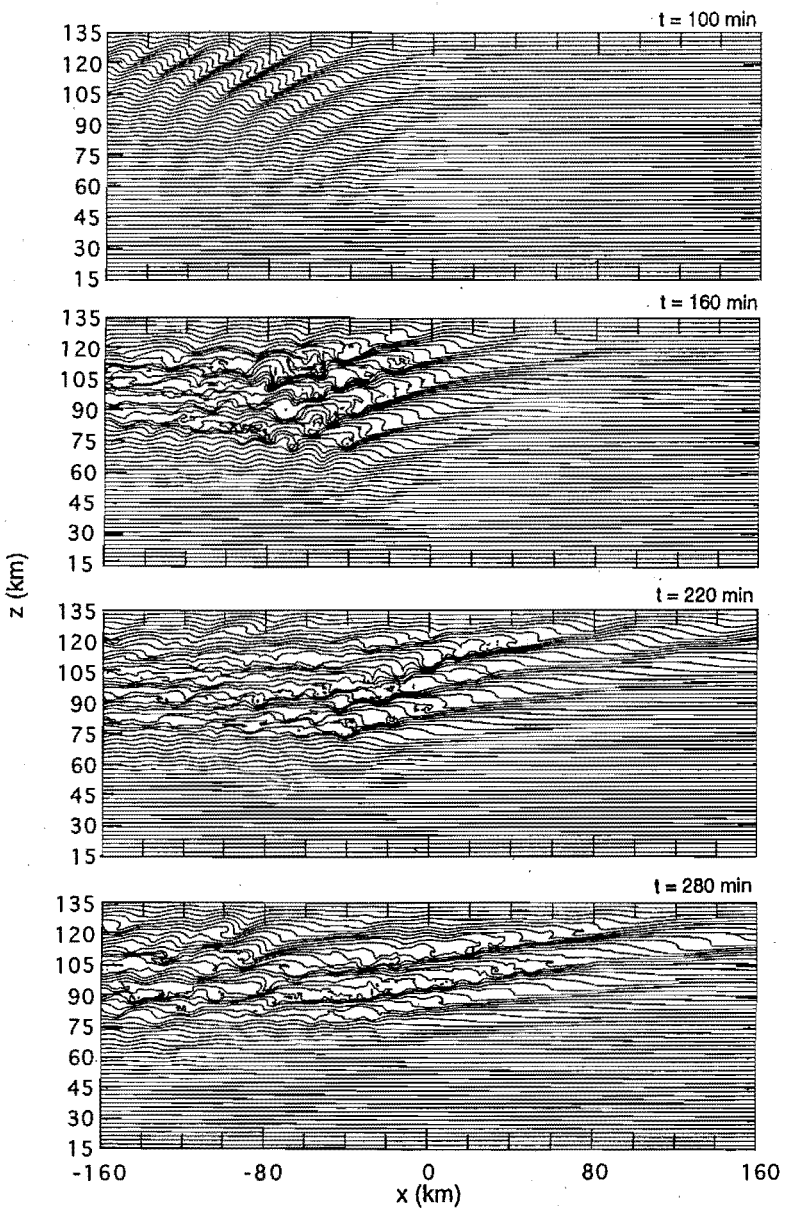

Figure 7.15: Potential temperature field at various times during simulation of Garcia & Prusa [1997] in which lower boundary forcing is located at x=0 km and the (westward) background wind is -50 ms^{-1} (After Garcia & Prusa [1997]).

89 km) but no others.

It is possible that the effects are not observed in more northern bins because the height of the Southern Alps is less west of these more northern GRBs and hence less gravity wave activity is expected. Good correlation coefficients are obtained from comparisons with GRBs north of 121 km as the Southern Alps no longer feature and hence there is no clear ridge for gravity wave production. All GRBs south of the Home site display consistently good correlation coefficients; this is to be expected as the southern sampling volume is well downstream of the gravity wave source and would not contain regions of large turbulence.

Increased confidence that orographically generated gravity waves are indeed the driving mechanism for reducing correlations with the first northern GRB could be obtained by filtering the data as a function of ground level wind velocity. It is proposed that correlations of GRBs with the first northern GRB should increase during ground level easterlies and decline during ground level westerlies.

Figure 7.16 displays the seasonal effects on the gradient of the least-squares regression line of best fit for ground range binned data from reference bins GRB 2 and 15 with all other comparison bins; again the solid circles and open circles are for reference bins GRB 2 and 15 respectively.

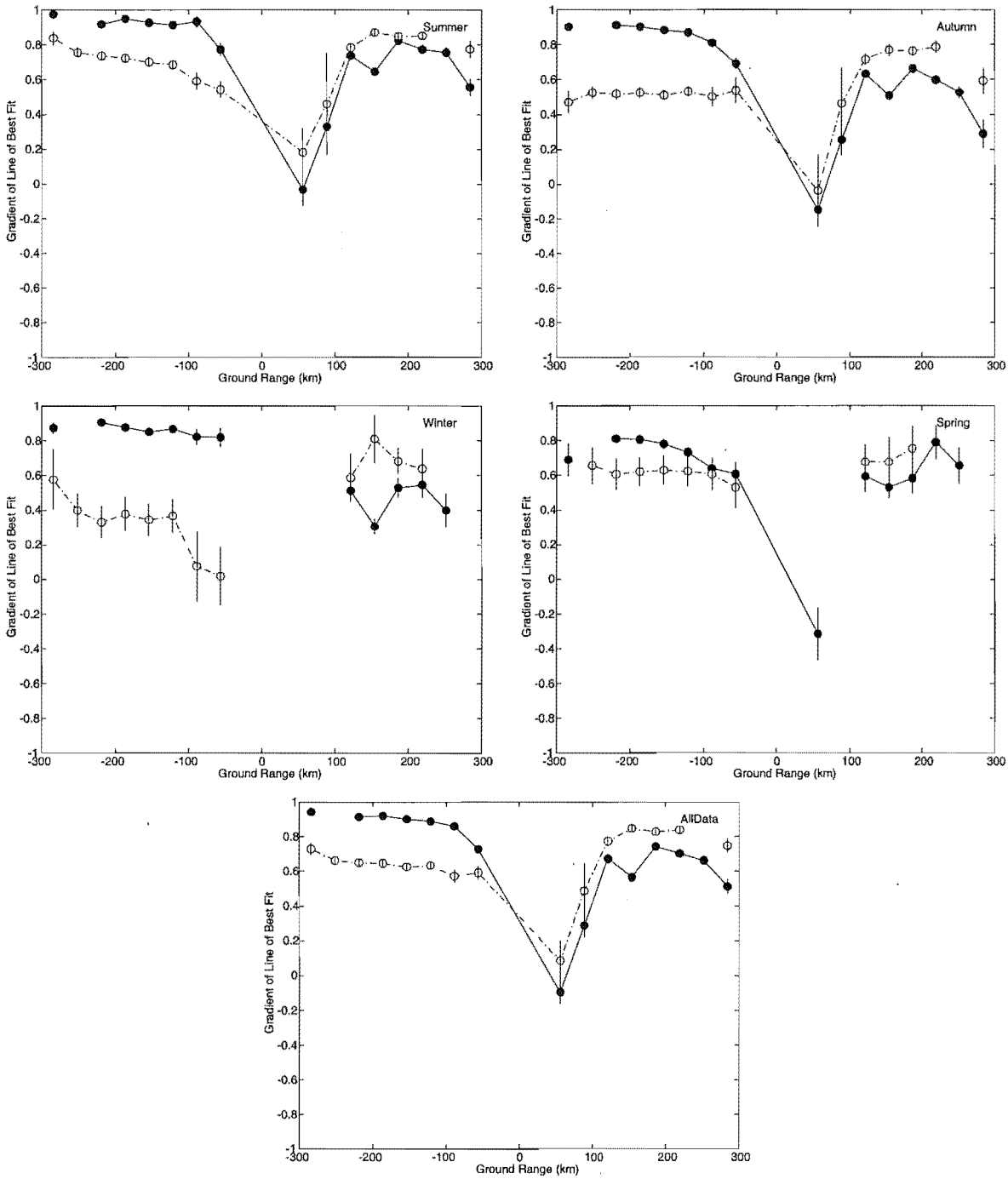

Figure 7.16: Gradient of the least-squares regression line of best fit for reference bins GRB 2 (solid circles) and GRB 15 (open circles) with all comparison GRBs at indicated ground range. A value is only displayed if 20 or more points are available. Positive ground ranges are north of the Home site and negative ground ranges are south. The Home site is at ground range =0 km. Uncertainty bars indicate 95% confidence limits.

7.5 GSWM Comparison

The amplitude and phase of the meridional component of the semidiurnal tide from the Earth's surface to an altitude of 120 km are shown in figure 7.17. These data were generated by the Global Scale Wave Model (GSWM) [Hagan et al. 1999] at a latitude of 45°S during equinoctial conditions. Semidiurnal amplitudes calculated

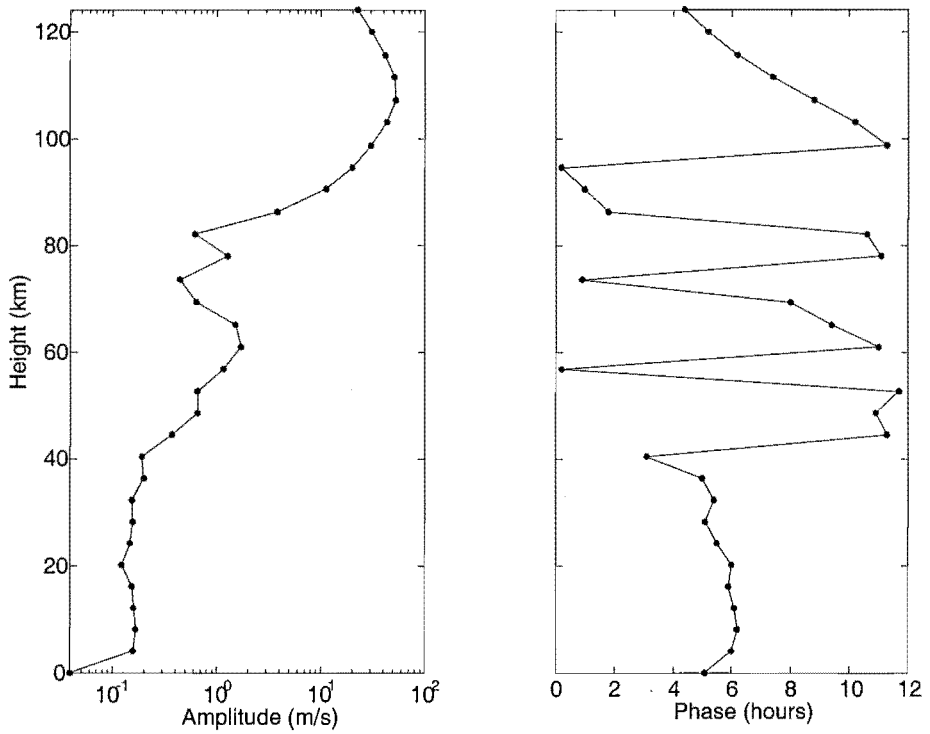

Figure 7.17: Amplitude and phase of the semidiurnal tide's meridional component as generated by the Global Scale Wave Model (GSWM) [Hagan et al. 1999] at a latitude of 45° south.

from the AMOR data set between heights of 90 and 110 km at equinox (top graph) and one month after equinox (bottom graph) are shown in figure 7.18 for comparison where the solid lines are the GSWM data at meteoric heights. The semidiurnal tide's amplitudes were obtained with the Lomb-Scargle Fourier Transform (LSFT2) using a 10 day window and 1 km height layer thickness. At equinox the meteor winds and model data show good agreement between 90 and 107 km however AMOR data at 108 and 109 km do not agree with the model and suggest the semidiurnal tide amplitude is smaller than predicted by the GSWM at these heights. One month after equinox a vertical height profile of similar shape to that predicted by the model is measured but at a height approximately 10 km lower down in the atmosphere. Comparing AMOR semidiurnal tide phase data with the GSWM at equinox (upper graph of

figure 7.19) shows phase change with height to be consistent with that produced by the model. Comparing the same GSWM data with AMOR data obtained one month after equinox indicates phase differences of approximately one hour throughout the meteor region.

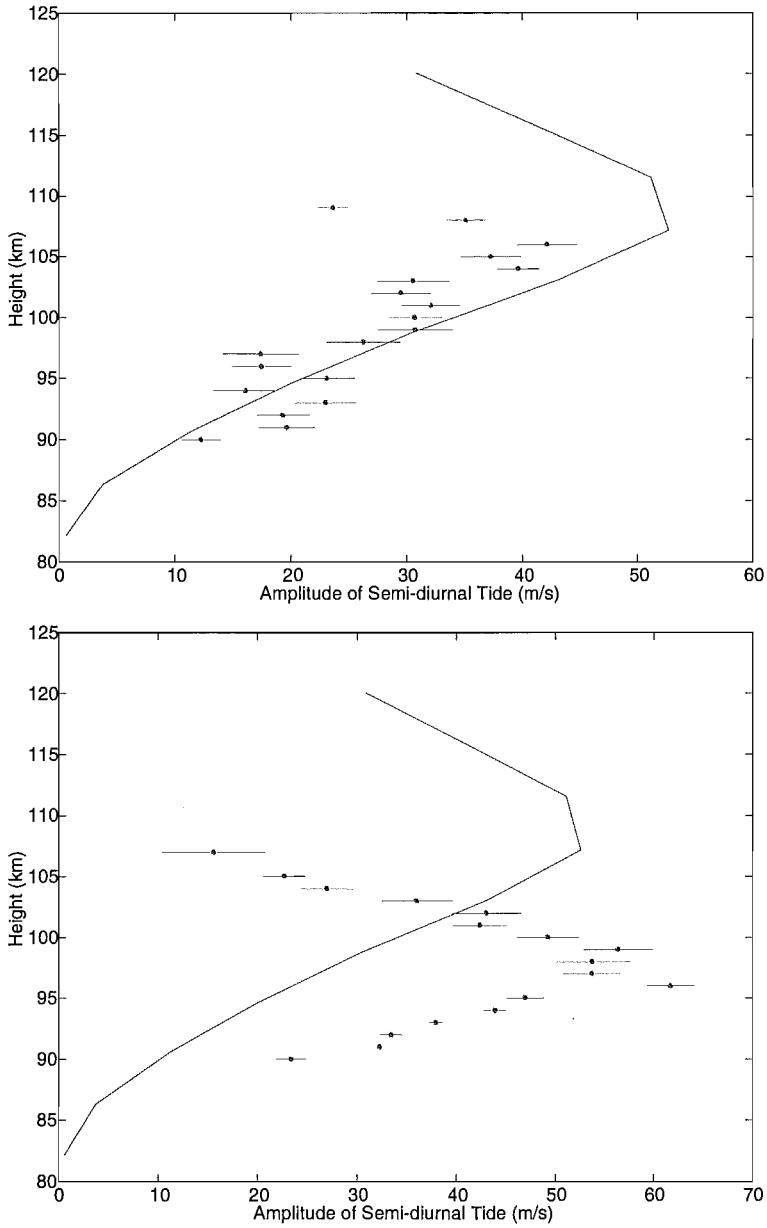

Figure 7.18: Comparison of semidiurnal tide amplitude measured by AMOR with that modelled by Hagan [1999] (solid line). The top graph contains AMOR data from 21 March 1998 and the bottom graph is from 21 April 1998.

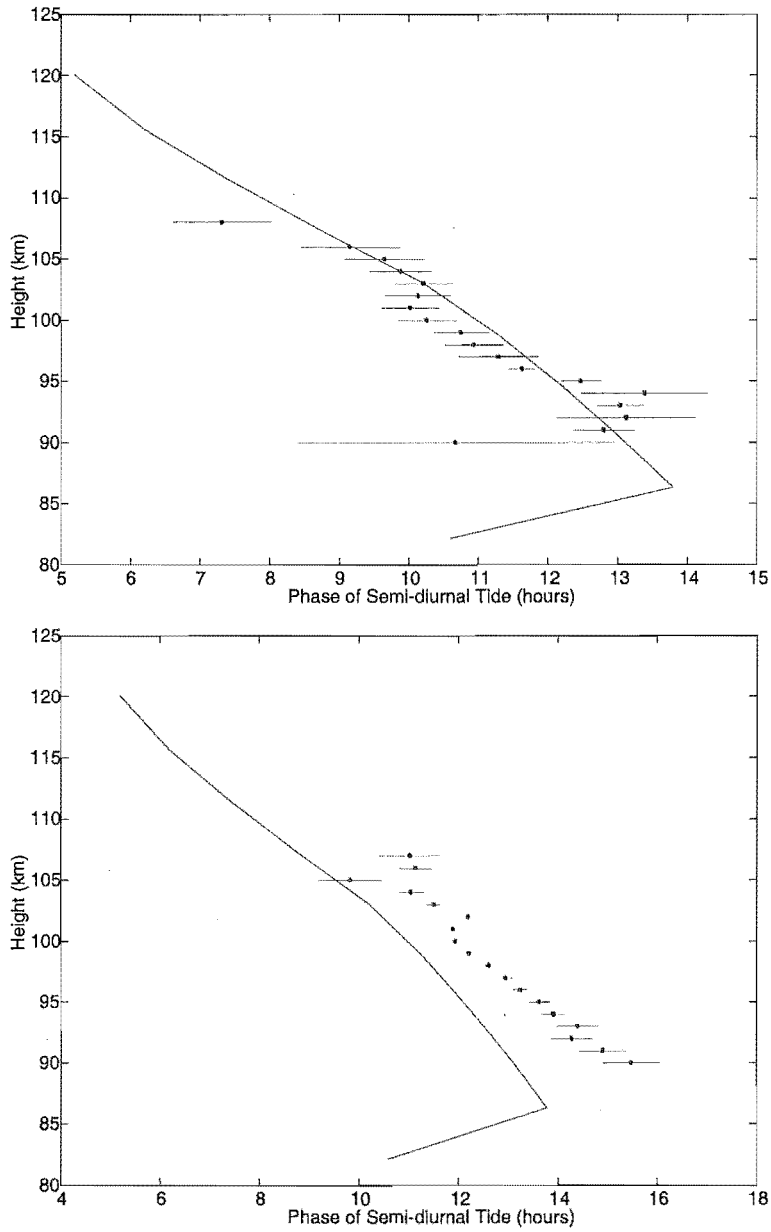

Figure 7.19: Comparison of semidiurnal tide phase measured by AMOR with that modelled by Hagan [1999] (solid line). The top graph contains AMOR data from 21 March 1998 and the bottom graph is from 21 April 1998.

Chapter 8

Further Work

8.1 Long Term Variation of Airglow Layer Heights

Wind speeds are measured with a Fabry-Perot Interferometer (FPI), at the University of Canterbury's Mount John observatory, by observing the Doppler shift of light emitted from two airglow layers at heights of approximately 87 and 97 km. Work presented in this thesis, and published in Plagmann et al. [1998], enabled airglow heights to be established by phase matching the semidiurnal tide's phase measured by the FPI, at the two airglow heights, with the range of phase values measured at the heights probed by AMOR. The monitoring of these heights over a limited time period suggested that there was a seasonal height variation. A more extensive survey over a time period of several years would provide an improved understanding of these airglow layer heights.

This analysis could have been done by combining the data output from the programme `GroundRangeAnalysis.c` into one large (height $\times$ time) hourly mean wind matrix (HMWM) with the MatLAB programme `VertProfile.m`. Next a vertical averaging window could be applied to the HMWM to emulate the vertical averaging inherent in the FPI instrument as it samples through the airglow layer. This would produce a new matrix called FPIVM. The shape of this window should indicate the airglow layer profile and for the work published in Plagmann et al. [1998]a Pascalian layer was used; this should suffice as an initial estimate [O'Brien et al. 1965].

The Lomb Scargle Fourier Transform (LSFT2) technique applied to the data matrix FPIVM would result in 3 tensors: time indices, amplitude and phase values. Comparison of the values in the phase values tensor (at times in tensor time indices) with the calculated FPI semidiurnal tide phases to find the height of best agreement would yield estimates of airglow layer height.

8.2 Detection of Orographically Generated Gravity Waves

In this thesis it has been proposed that orographically generated gravity waves are a possible explanation for the lack of correlation between hourly mean winds detected in near Home site northern ground range bins and all other ground range bins.

At the time of the writing of this thesis work is in progress to construct an east-west looking array. From a wind-delineating point of view this will enable the zonal component of the wind to be measured. Additionally data sampled from this orthogonal observation will allow a more conclusive analysis of orographically generated gravity waves as the western beam will sample the atmosphere along a column close to the Southern Alps gravity wave source. Any gravity wave contamination along this sampling volume should be identifiable as a region, approximately 100 km from the Southern Alps displaying poor correlation with other ground range bins.

Selecting AMOR wind measurements based on ground level wind speed and direction over the Southern Alps for ground range correlation analysis would provide valuable information for confirming whether the radar is detecting gravity wave induced turbulence. Should more turbulence be detected by AMOR during ground level westerlies than there is during ground level easterlies then this would be good evidence to support the detection of gravity wave induced turbulence.

8.3 Application of Correction Algorithm

Chapter 3 discussed the effect of the Earth's magnetic field on the drift of the meteor train. In this chapter it was concluded that the influence of the magnetic field was minor for wind measurements below 100 km, but above approximately 105 km the introduced error becomes progressively larger. A method of correcting for this error was proposed which, however requires that both zonal and meridional components were measured. Once the orthogonal array is constructed it could be beneficial to apply the correction and compare the results.

Chapter 9

Conclusion

The existing meteor orbit radar, AMOR, has been modified to enable measurements of the meridional component of atmospheric winds in the meteor zone of 80–110 km. This has been achieved by the addition of new electronic hardware which provides both phase and phase quadrature components of the received echo to be stored on a dedicated winds control computer. The AMOR orbit control software has been modified to inform the winds computer when an echo event occurs and developed software collects the echo profile from the two winds dedicated analogue to digital converters and writes the data to memory in the winds control computer through the process of Direct Memory Access (DMA). Data are regularly (weekly) collected from the radar site for analysis.

The line of sight (radial) wind velocity has been inferred from the Doppler shift of the radar pulse as it returns from the meteor train blown at the speed of the wind. Analysis techniques have enabled wind speed measurements to be made from echoes which are as short as 0.03 s. Approximately 2000 wind measurements are obtained each day from a possible 3000 meteor echoes. Rejection of echoes is mainly due to an inability to fix the measurement's location (only 10% of the echoes are rejected due to a non-linear phase behaviour). Assuming no vertical wind, radial velocities have been calculated with uncertainties which are less than 1 $\mathrm{m\,s^{-1}}$ (for heights less than 95 km). To account for the possibility of vertical wind conditions and magnetic field effects the uncertainty in individual horizontal wind measurements is estimated as 3 $\mathrm{m\,s^{-1}}$ (for heights less than 100 km).

A method of reducing wind speed errors due to the effect of the Earth's magnetic field, requiring measurements of both zonal and meridional components, has been derived.

Harmonic analysis of hourly averaged meteor wind measurements revealed a dominant semidiurnal tide. The amplitude and phase of this tide clearly exhibited

a seasonal behaviour. The vertical wavelength of the semidiurnal tide was also measured and enabled the identification of various tidal modes throughout the year.

Occasionally the dominant behaviour of the semidiurnal tide was suppressed by the presence of oscillations, known as planetary waves, with periods of a few days. Analysis revealed 2, 4, 5 and possibly 10-day planetary wave events within the observation period. The mean wind was also shown to display behaviour consistent with general circulation theory.

Hourly mean winds from the AMOR winds radar were compared with those from two other instruments measuring winds at similar heights over the South Island of New Zealand. Comparisons with the Fabry-Perot Interferometer at Mount John revealed an excellent agreement with correlation coefficient values up to 0.95. During the comparison process, a method of estimating the airglow layer heights by matching the measured values of the phase of the semidiurnal tide was developed. Comparisons with the second instrument, a Medium Frequency (MF) radar at Birdlings Flat consistently provided a poorer agreement.

Grouping the AMOR data by radar ground range allowed an intra-comparison of the data to be performed. Correlating data in the two extreme ground range bins (GRBs) with all other GRBs showed a gradual reduction in correlation as the distance between the compared GRBs increased. Two results were identified which may be attributable to gravity wave interaction.

The first result was a seasonal effect giving low correlations between southern GRBs and a northern GRB in winter and high correlations in summer. It was proposed that the poorer correlations were consistent with times of year when gravity waves could propagate to heights of the meteor region and induce turbulence.

The second result was a sudden decline in correlation between all GRBs and the near Home site northern GRBs. The location of these near Home site northern GRBs, with respect to the Southern Alps mountain range, puts them in the place likely to be influenced the most by orographically generated gravity waves.

Comparisons of amplitude and phase for the semidiurnal tide under equinox conditions were compared with the Global Scale Wave Model (GSWM) and showed good agreement.

Chapter 10

Acknowledgments

Many members of the Department of Physics and Astronomy have been instrumental in providing both assistance and encouragement during the period of this work.

First I would like to thank my supervisor Professor W. Jack Baggaley for his support and advice. I would also like to thank Dr Grahame Fraser for initially proposing the topic and always being available to have a quick chat. Other academic members of the department who have provided valuable advice include Dr Bryan Lawrence, Dr Don Grainger, Dr Roger Reeves and Associate Professor Peter Cottrell.

As a post-graduate student I have been privileged to share a room with David Galligan. His enthusiasm, willingness to discuss the subject and friendship have been most appreciated. Of the technical staff, my sincere thanks to Graeme Plank, Ross Ritchie and Geoff Graeme for their expertise and assistance.

Fellow students who have had an influence in this work include Dr Steve Smith, Dr Adam Dunford, Dr Manfred Plagmann, David Galligan (again) John Grant and Scott Osprey.

My thanks also go to my parents for their emotional as well as financial support throughout my, near ten, years of being a student. I also wish to thank Pauline for her encouragement and caring during the stressful completion of this work.

I would especially like to thank Dr R.G.T (Bob) Bennett for his considerable effort in assisting me with this project. His willingness to share his extensive knowledge and take an active interest in all parts of this work has been paramount in making this thesis possible.

Finally I would like to thank the Department of Physics and Astronomy for

providing financial assistance in the form of a departmental scholarship.

Appendix A

Data Collection and Reduction Source Code

A.1 Program Dop128.pas

The Borland Turbo Pascal programme, Dop128.pas, controls the collection of wind data at the radar site. Monitoring of a line from the AMOR meteor orbits pc indicates the presence of meteor echoes and initiates the data collection process. Phase data for both echo and ground pulse are extracted from the DMA memory storage and written to disk. A meteor identification number, determined by the orbits pc, is sent to the winds pc via the serial port.

```
Program Dop128;

{                                                                    }
{                                                                    }
{                                                                    }
{                                                                    }
{                                                                    }
{  for Use at Birdling's Flat to Control Doppler Radar               }
{                                                                    }
{                                                                    }
{                                                                    }
{          Programmed By S H Marsh                                   }
{                                                                    }
{                                  Mar/97                            }
{                                                                    }
{              Data saved to c:\ drive                               }
{   To allow ComPort usage have mode com2: 98,n,8,1                  }
{                   in autoexec.bat file                             }

uses dos,crt,graph,genutil,setclock;
const
```

```
        ComPort:Word=$2f8;          {Communications port #2}
        Nocursor=$2020;
        FullCursor=$010A;
        NoOfSweeps=128;
        sixteen:longint = 16;
        P300 : byte = $22;    { RxGain = hi }
        P301 : byte = 100;     { Total no of Heights}
        P302 : byte = $80;    { Enables the DMA req F/F (DMA3)}
type
    byteRec= record
        Year,Month,Day,Hour,Min,Sec,TxBin,Bin,Number       : byte;
        EchCos,EchSin,TxCos,TxSin:array[1..NoOfSweeps] of byte;
    end;
    byteRecFile=file of byteRec;

    sweepAddr = record
                    Page    : byte;
                    LSB     : byte;
                    MSB     : byte;
                end;
    AtoDLocations=record
                      Segm : word;
                      Offsm: word;
                    end;

var
    P                                                  : pointer;
    ch,c                                               : char;
    forEver                                            : boolean;
    n,code                                             : integer;
    counter, metcnt                                    : integer;
    time                                               : text;
    bin,TxBin,hc                                       : longint;
    BFlatFile                                          : byteRecFile;
    BFlatRecData                                       : byteRec;
    filename,subdir,dir,storedir,starttime             : string;
    start,hourcheck                                    : integer;
    cntr,check,timeadj, hourstr,fileext,date           : string;
    L                                                  : longint;
    Pseg,offset,y,a                                    : longint;
    AtoDmem              : array[1..NoOfSweeps] of pointer;
```

```
    Sweepaddresses      : array[1..NoOfSweeps] of sweepaddr;
    sweepnumber                                : integer;
    meteor, count                              : integer;
    AtoDMemIRQ          : array[1..NoOfSweeps] of AtoDLocations;
    Dummyread                                  : byte;

{*************************************************************}

 {$M 64000,0,300000}
 {$R-}
{ Allows the cursor to be made invisible}
procedure setcursortype(numver:word);
          inline($59/$B4/$01/$CD/$10);

{Sets up the A/D memory space}
procedure SetAtoDMemory;
begin   for sweepnumber := 1 to NoOfSweeps do
 begin
        Getmem(P,200);    {NoOfSweeps * 200 byte memoryblocks}
        Pseg:=seg(P^);
        Offset :=ofs(P^);
        AtoDMem[SweepNumber]:=ptr(Pseg,Offset);
        with AtoDMemIRQ[SweepNumber] do begin
             Segm:= Pseg;
             Offsm:=Offset;
             end;
        L := seg(P^)*sixteen+ofs(P^);
        with Sweepaddresses[sweepnumber] do
        begin
             Page:= L shr 16;
             LSB := L and $FF;
             MSB := (L shr 8) and $FF
        end;
 end;
end;

{Initialises DMA transfers}
procedure DmaSetup;
begin
with Sweepaddresses[sweepnumber] do  begin
        port[$82] :=Page;  { Set Page Register}
```

```
        port[$0c] :=0;      { reset the byte ptr F/F }
        port[$06] :=LSB;    { Set Base address LSB}
        port[$06] :=MSB;    { Set Base address MSB}
        port[$07] :=199;    { Set Word Count LSB to 199 }
        port[$07] :=0;      { Set Word Count MSB to $0 }
        port[$0b] :=$47;    { Set Mode Reg to $47 Byte Transfer}
        port[$0a] :=3;      { Unmask Channel 3 }
end;
end;

{Sends pulse to initiate the DMA transfer process}
procedure  StartDmaTransfers;              { Starts each run }
var  delay : longint;
begin
        port[$300] := P300 or $01;  { reset F/F }
        {for delay := 1 to 10 do;}  { only for faster machine}
        port[$300] := P300 and $fe;
        port[$300] := P300 or $80;  { Send start pulse }
        {for delay := 1 to 10 do;}
        port[$300] := P300 and $7f;
end;

{Coordinates timing for initiation of DMA transfers}
procedure RecordEcho;
begin
        port[$300] := P300;       {is $22}
        port[$301] := 100;        {Sets the number of range bins}
        port[$302] := 0;          {was $80}
      repeat
            inc(SweepNumber);
            DMAsetup;                  {Sets up DMA process}
            repeat until port[$300] and 2 =0;{Wait during pulse}
            repeat until port[$300] and 2 =2;{Wait for pulse}
            StartDmaTransfers;
            repeat until port[$07] =255; {all range bins counted}
      until SweepNumber = NoOfSweeps;
end;

{Defines the time stamped file name for echoes}
procedure FindName(var time:string);
     var
```

```
    hours, mins, seconds, overtime : longint;
    h,m,s                          : byte;
    secs,ta,temp                   : longint;
    timestr                        : string;
function LeadingZero(w : byte) : string;
        var
          s : string;
        begin
            Str(w:0,s);
            if Length(s) = 1 then
                s := '0' + s;
            LeadingZero := s;
        end;
        begin
            timestr:=cmosTime(h,m,s);
            val(copy(timestr,1,2),hours,code);
            val(copy(timestr,3,2),mins,code);
            val(copy(timestr,6,2),seconds,code);
            overtime:= 0;
            if hours >12 then
                    begin
                        hours := hours-12;
                        overtime := 12;
                    end;
            secs:= (hours*3600)+(mins*60)+seconds;
            val(timeadj,ta,code);
            if abs(ta) < secs then secs:= secs +ta;
            temp:=secs;
            hours := overtime + secs div  3600;
            temp:=temp-((hours-overtime)*3600);
            secs:=temp;
            mins:= secs div 60;
            seconds:= temp-(mins*60);
            time := (leadingzero(hours)+leadingzero(mins)
                    +leadingzero(seconds));
        end;

{Defines the time stamped directory into which files are written}
procedure findDir(var date:string);
const
  months : array [1..12] of string[9] =
```

```
    ('Jan','Feb','Mar',
     'Apr','May','Jun',
     'Jul','Aug','Sep',
     'Oct','Nov','Dec');
var
  y, m, d, dow                : byte;
  day,month,year,datestr      : string;
  begin
       datestr:=cmosDate(y,m,d);
       y:= y mod 100;     {prog needs to be modified in 2095 :-) }
       str(d:0,day);
       if d<10 then day := '0'+day;
       str(y:0,year);
       date := (day+months[m]+year);
  end;

{Locates from within the DMA memory space the
                ground pulse and echo signals}
procedure FindBins;
var
    hc                        : longint;
    binamps :array[1..2500] of real;
    total   :array[1..100] of real;
    max                       : real;
begin
for hc:= 1 to 100 do total[hc]:=0;
for SweepNumber := 1 to 25 do
begin
    with AtoDMemIRQ[SweepNumber] do begin
            for hc:= 1 to 100 do
               begin
               BinAmps[hc]:=
                 sqrt(sqr(mem[segm:offsm+(2*hc)-1]-124)
                   +sqr(mem[segm:offsm+(2*hc)]-124));
               total[hc]:= total[hc]+Binamps[hc]
               end;
end;
      max:= 0;
      bin:=0;
      for n := 1 to 14 do
      begin
```

```
        if total[n]>max then
           begin
             max:=total[n];
             TxBin:=n;
           end;
        end;
        max:= 0;
        bin:=0;
        for n := 15 to 100 do
        begin
        if total[n]>max then
           begin
             max:=total[n];
             bin:=n;
           end;
        end;
end;
end;

{Creates the data file for subsequent echoes to be written to}
procedure OpenNewDataFile;
begin
 findDir(SubDir);
 StoreDir:='c:\data';
 {$I-}
 MkDir(StoreDir);
 StoreDir:= StoreDir+'\'+SubDir;
 MkDir(StoreDir);
 if IOResult = 0 then  begin end;
 FindName(fileext);
 {$I+}
 fileext:= copy(fileext,1,2);
 val(fileext,hourcheck,code);
 assign(BFlatFile,Storedir + '\WIND_'+fileext+'.');
 {$I-}
 reset(BFlatFile);
 if IOResult = 0 then Seek(BFlatFile,FileSize(BFlatFile))
                      {Find the end of the file if it exists}
                 else rewrite(BFlatFile);
 {$I+}
 end;
```

```
{Checks to see if meteor present signal exists}
function Dataready : boolean;
var a:byte;
begin
     a:=Port[$2fd];
     if a and 1 = 1 then Dataready := True
                    else Dataready := False;
end;

{writes data to hard disk}
procedure SaveData;
          var tempstr:string;
begin
   findBins;
   finddir(date);
   findname(fileext);
   with BFlatRecData do begin
        val(copy(date,6,7),Year,code);
             tempstr:=copy(date,3,3);
             if tempstr='Jan' then Month:=01;
             if tempstr='Feb' then Month:=02;
             if tempstr='Mar' then Month:=03;
             if tempstr='Apr' then Month:=04;
             if tempstr='May' then Month:=05;
             if tempstr='Jun' then Month:=06;
             if tempstr='Jul' then Month:=07;
             if tempstr='Aug' then Month:=08;
             if tempstr='Sep' then Month:=09;
             if tempstr='Oct' then Month:=10;
             if tempstr='Nov' then Month:=11;
             if tempstr='Dec' then Month:=12;
        val(copy(date,1,2),Day,code);
        val(copy(fileext,1,2),hour,code);
        val(copy(fileext,3,2),min,code);
        val(copy(fileext,5,2),sec,code);
        end;
        BFlatRecData.bin:=bin;
        BFlatRecData.TxBin:=TxBin;
   for SweepNumber:= 1 to NoOfSweeps do
        begin
```

```
            with AtoDMemIRQ[SweepNumber] do begin
               with BFlatRecData do begin
                  EchCos[SweepNumber]:=mem[segm:offsm+(2*bin-1)];
                  EchSin[SweepNumber]:=mem[segm:offsm+(2*bin )];
                  TxCos [SweepNumber]:=mem[segm:offsm+(2*TxBin-1)];
                  TxSin [SweepNumber]:=mem[segm:offsm+(2*TxBin )];
               end;
            end;
  end;
  repeat until Dataready;
 {Waiting for Flag to indicate byte sent from AMOR}
  BFlatRecData.Number:=Port[ComPort];
  if Hourcheck = BFlatRecData.hour then
                                        write(BFlatFile,BFlatRecData)
                                       else
                                        begin
                                             close(BFlatFile);
                                             OpenNewDataFile;
                                             DoClockSet;
                                             MetCnt:=0;
                                        end;
end;

{Finds the time for start of observation run}
procedure sttime(var st:integer);
var stt, ststr : string;
begin
     findname(stt);
     ststr:= copy(stt,1,2);
     val(ststr,st,code);
end;

{Tasks that need to be performed when stopping the programme}
procedure ShutDown;
begin
    close(BFlatFile);
    ClrScr;
    writeln('Complete');
    SetCursortype(FullCursor);
    halt
end;
```

```
{The user interface}
procedure Display;
var hr,mn,sc:string[3];
begin
clrscr;
for counter := 1 to 6 do writeln;
with BFlatRecData do begin
str(hour,hr);
if hour<10 then hr:= '0'+hr;
str(min,mn);
if min<10 then mn:= '0'+mn;
str(sec,sc);
if sec<10 then sc:= '0'+sc;
end;
writeln('          Last meteor found at:   ',hr+':'+mn+':'+sc );
writeln;
writeln;
writeln('             ',metcnt,' meteors saved this hour');
writeln;
writeln('                 Press ''S'' to Stop');
writeln;
writeln;{('             Press ''C'' for Clock Adjustment');}
writeln;
writeln;
writeln('  Transmitter Ground Pulse Found In Range Bin: ',TxBin);
writeln;
writeln('  Last Echo Found in Range Bin                : ',Bin);
end;
begin                          {Of Main Program}
Dummyread:=Port[ComPort];
SetAtoDMemory;
SetCursortype(NoCursor);
DoClockSet;
OpenNewDataFile;
forever := False;
filename:=' ';
TxBin:=0;
Bin:=0;
sttime(start);
clrscr;
```

```
metcnt:=0;
BFlatRecdata.hour:=0;
BFlatRecData.min:=0;
BFlatRecData.sec:=0;
repeat
      begin
      Display;
     while port[$300] and 2 = 0 do;{Ensures Tx pulse exists}
     while port[$300] and 1 = 0 do {Look for meteor present_|-|_}
     begin
       if keypressed then
            begin
                 c:=readkey;
                 if upcase(c)='S' then ShutDown;
            end;
    end;
         SweepNumber:=0;
         RecordEcho;
         SaveData;
         inc(metcnt);
     end;
until forEver;
end.
```

A.1.1 Unit SetClock

The Borland Turbo Pascal Unit, `SetClock.pas`, is called from programme `Dop128.pas` and reads the time from the serial output of a local clock and writes it to the CMOS clock of an IBM_AT.

```
Unit   SetClock;
interface
procedure DoClockSet;
implementation
{This program is designed to read and set the CMOS clock on an IBM_AT
 using either the keyboard or an external time pip source attached to
 COMM2 (bell ring line) }

uses    Dos, Crt;

const   CmosPort = $70;
```

```
        On = $709;
        Off = $2020;

var  TimeOut, Hths, Minutes, Seconds, Hours                    : longint;
     CmosPackedTime, PackedTime , Drift, DriftTime             : longint;
     DriftSeconds                                              : real;
     ComPort, Year, Month, Day, DayOfWeek                      : word;
     Error                                                     : boolean;
     Time                                                      : string[16];

{.....................................................................}
function Dataready: boolean;
var a : byte;
begin
    Dataready := false;
    a:= port[ComPort+5];
    if a and $01 = 1 then Dataready := true;
end;

procedure    SetCom1;
var   Regs : Registers; a : byte;
begin
      ComPort := $3f8;
      with regs do
      begin
       ah := 00;    { Select Initialise Serial port }
       al :=$E3;    { E3 = 9600 baud,no parity,1 stop bit,8 data bits }
       dx := 00;    { Select Com1 }
       intr($14,regs);    { Use Bios int14 to send the byte }
      end;
      a := port[ComPort];          {Do dummy read}
end;
function CmosRam(adr : byte ): byte;
var i:word;
begin
if CmosPort=$70
 then begin {AT or PS/2 CMOS}
        inline($FA);              { cli   DisableInterrupts }
        Port[CmosPort]:=adr;      { out   70h,adr           }
        CmosRam:=Port[$71];       { in    CmosRam,71h       }
        inline($FB);              { sti   EnableInterrupts  }
```

```
      end
  else begin
        inline($FA);              { cli  DisableInterrupts }
        i:=Port[CmosPort+adr]; {read twice in case port strange}
        if i<>Port[CmosPort+adr] then {worry};
        inline($FB);              { sti  EnableInterrupts  }
        CmosRam:=i;
      end;
end;
procedure SetCmosRam(adr,NewContents : byte );
var i : byte; CheckSum : word;
begin
if CmosPort=$70
   then begin {AT or PS/2 CMOS}
        inline($FA);          { cli         ;DisableInterrupts }
        Port[$70]:=adr;           { out 70h,adr  }
        Port[$71]:=NewContents; { out 71h,NewContents }
        inline($FB);          { sti         ;EnableInterrupts  }
        end
   else begin
        Port[CmosPort+adr]:=NewContents;
        if NewContents<>Port[CmosPort+adr] then {worry};
        end;
if adr in [$10..$2D]
   then begin
        CheckSum:=0;
        for i:=$10 to $2D do inc(CheckSum,CmosRam(i));
        SetCmosRam($2E,hi(CheckSum));
        SetCmosRam($2F,lo(CheckSum));
        end;
end;
type string2 = string[2];
function   TwoDigits(n : byte) : string2;
begin
TwoDigits:=char(48+n div 10)+char(48+ n mod 10);
end;
type  string16 = string[16];
function   CmosTime : string16;
begin
     Hours    :=CmosRam(4);
     Minutes :=CmosRam(2);
```

```
      Seconds :=CmosRam(0);
      if (CmosRam($0B) and 4)=0
        then begin
               Hours   :=(Hours and $0F) + 10*(Hours div 16);
               Minutes :=(Minutes and $0F) + 10*(Minutes div 16);
               Seconds :=(Seconds and $0F) + 10*(Seconds div 16);
             end;
CmosTime := TwoDigits(Hours)+':'+TwoDigits(Minutes)
              +':'+TwoDigits(Seconds);
end;
procedure  SetCmosClock (Hrs,Mins,Secs : integer);
begin
    SetCmosRam($0,Secs); {already converted to BCD/hex}
    SetCmosRam($02,Mins);
    SetCmosRam($04,Hrs);
    write(#7);
end;
procedure  GetSerialTime;
var  a,b,c, Secs,Mins,Hrs : integer;
begin
  TimeOut := 0;
  repeat
   repeat inc(TimeOut); if TimeOut=1000000 then exit;
   until Dataready;
    a :=port[ComPort];
  until a=$FF;
   repeat inc(TimeOut); if TimeOut=500000 then exit;
   until Dataready;
    Hrs  := port[ComPort];
   repeat inc(TimeOut); if TimeOut=500000 then exit;
   until Dataready;
    Mins := port[ComPort];
   repeat inc(TimeOut); if TimeOut=500000 then exit;
   until Dataready;
    Secs := port[ComPort];
   Error := false;
   if Hrs> $23 then begin Error := true; exit; end;
   if Mins> $59 then begin Error := true; exit; end;
   if Secs> $59 then begin Error := true; exit; end;
   SetCmosClock(Hrs,Mins,Secs);
end;
```

```
{*************************************************************}
procedure DoClockSet;
begin
        clrscr;
        SetCom1;
        GetSerialTime;
        if Error then GetSerialTime;
end;

begin  {no initialisation}
end.
```

A.2 Program System.pas

The Borland Turbo Pascal programme, `System.pas`, coordinates the data reduction process. There are three stages to this process. The first stage determines line of sight velocities from the `WIND_<hh>` files. The second stage locates the position of the measurement from `NZST_<hh>` files and finally the third stage combines the line of sight velocity with its location, generating a wind measurement.

```
uses Dos, crt, speeds, heights, Match, Sort, Defns ;

const MaximumNumberOfFiles=100;
type
 ra = array[1..MaximumNumberOfFiles] of string[13];
var
 S, chk, month          : string;
 DirInfo                : SearchRec;
 FileDir, DatesFile     : text;
 Overwriting            : boolean;
 Date, DoRFDate         : string7;

{Checks for the presence of file Graph.dat for a particular date}
function NoGraphFile(Date:string7):boolean;
var ChkFile:text;
    Temp:string[10];
begin
NoGraphFile:=False;
assign(ChkFile, 'E:\ANALYSED\'+DATE+'\GRAPH.DAT');
{$I-}
```

```
reset(ChkFile);
{$I+}
if IOResult = 0 then begin
                        {$I-}
                        read(ChkFile,Temp);
                        if Temp='' then NoGraphFile:= True;
                        {Checking that file isn't empty}
                        close(ChkFile);
                        {$I+}
                        end
                  else NoGraphFile:=True;
end;

{Checks to see if there are orbit data available for
 analysis on a particular date}
function AMORDataExists(Date:string7):boolean;
var search,month:string;
DirInfo:SearchRec;
SearchFile:File;
Found:boolean;
begin
Found:=False;
search:=Drive+':\AMORDATA\'+Date+'\NZST_*.';
FindFirst(search,Archive,Dirinfo);
while (doSERROR=0) and (NOT Found) do begin
      assign(SearchFile,Drive+':\AMORDATA\'+DATE+'\'+DirInfo.Name);
      FileMode:=0;
      reset(SearchFile);
      if FileSize(SearchFile) > 0 then Found:=TRUE;
      close(SearchFile);
      FindNext(DirInfo);
      end;
AMORDataExists:=Found;
end;

{Checks to see if there are winds data available
                for analysis on a particular date}
function DoRFDataExists(Date:string7):boolean;
var search:string;
SearchFile:File;
DirInfo:SearchRec;
```

```
Found:boolean;
begin
Found:=False;
search:=Drive +':\DorfDATA\'+Date+'\WIND_*.';
FindFirst(search,Archive,Dirinfo);
while (doSERROR=0) and (NOT Found) do begin
        assign(SearchFile,Drive +':\DorfDATA\'+Date+'\'+DirInfo.Name);
        reset(SearchFile);
        if FileSize(SearchFile) > 0 then Found:=TRUE;
        close(SearchFile);
        FindNext(DirInfo);
        end;
DoRFDataExists:=Found;
end;

{Generates a Graph.dat file}
procedure MakeGraphFile(Date:string7);
var DirInfo:SearchRec;
begin
FindFirst(Drive+':\Analysed\'+Date+'\PhVels.Dat',Archive,DirInfo);
    if (doSERROR=0) and (Overwriting) then FindWindSpeeds(Date,Date);
    if (doSERROR<>0) then FindWindSpeeds(Date,Date);

FindFirst(Drive +':\Analysed\'+Date+'\AMOR.Dat',Archive,DirInfo);
    if (doSERROR=0) and Overwriting then FindHeightsEtc(Date);
    if (doSERROR<>0) then FindHeightsEtc(Date);

ComBineVelsAndHeights(Date);
end;
begin
clrscr;
OverWriting:=True;
SortDates;
assign(DatesFile,'c:\Dop128\Dates.dat');
reset(DatesFile);
    while not eof (DatesFile) do
    begin
        readln(DatesFile,Date);
        if NoGraphFile(Date) and AMORDataExists(Date) and
              DoRFDataExists(Date) then MakeGraphFile(Date)
        else if AMORDataExists(Date) and DoRFDataExists(Date)
```

```
            and Overwriting then MakeGraphFile(Date);
    end;
writeln;
writeln('All Available Data Have Been Analysed');
repeat until keypressed;
end.
```

A.2.1 Unit Speeds

The Borland Pascal Unit, `Speeds.pas`, is called from the control programme `System.pas` to derive line of sight wind speeds from the raw `WIND_<hh>` files.

```
Unit Speeds;
interface
uses Dos,Crt,Defns,Diff;
procedure FindWindSpeeds(Dir,SaveDir:string7);


implementation

{This program takes as input files produced at the Birdlings' Flat
 field station. These data are analysed to produce line of sight
 velocities of meteor trail drift in the atmosphere at heights of
 80 - 120 kms.

The method used invloves determining the time rate of change of
phase for both transmitter and echo}
Programmed by:      Steven H Marsh
                University of Canterbury
                      11 Nov 96


}


const MaximumNumberofFiles=24;
    {  ndatap=128;}
      wavelength=2.998e+8 / 26.2e+6;
      Nocursor=$2020;
      FullCursor=$010A;
      prf=379;
```

```
        MinNumOfTransPulses=5;

type
 {    realarrayNDATA=ARRAY[1..ndatap] of real;}

     byteRec= record
                Year,Month,Day,Hour,Min,Sec,TxBin,Bin,Number:byte;
                EchCos,EchSin,TxCos,TxSin:array[1..128] of byte;
              end;
     byteRecFile=file of byteRec;

var
    tempfile                                              : text;
    byteRF                                                : byteRecFile;
    Bflatfile                                             : byteRecFile;
    BFlatRecData                                          : byteRec;
    FileDir,datefile, misc,velsout                        : text;
    Phase,sine,cosine,hyp,ratio,txsine,txcosine,lastphase,thisphase,
    temp, total,sineav, cosav, txsineav, txcosav,av, bv, txgrad,
    echgrad, velocity,firsthf,sechf, txuncertainty, echouncertainty,
    totuncertainty,veluncertainty                         : real;
    a,n,count,Stop, Start, NoOfDates, FileNo, NoOFFiles : integer;
    FilesExist, DayFilesLeft,HourFileFinished,fancy,
    DataAvailable                                         : boolean;
    date, Path,AccFileStr                                 : string;
    t,txph,PartialEchPh,echph,amp,EchPhW,TxPhW:realarrayNDATA;
    c                                                     : char;
    AccFile,RejFile,FilesOpened                           : longint;
    NoAnalF                                               : integer;
    LastNumber                                            : byte;
    SaveDate                                              : string7;

{Initialisation routine}
procedure Setup;
begin
      assign(tempfile,'c:\temp.dat');
      assign(misc,'c:\misc.dat');
      DayFilesLeft:=true;
      FileNo:=0;
      for a:=1 to 128 do t[a]:=a/prf;
end;
```

```
{Determining average values for phase and quad components}
procedure FindAverages;
var costotal,sintotal:real;
begin
with BFlatRecData do begin
sintotal := 0;
costotal := 0;
      for a:=1 to 128 do
      begin
           sintotal := sintotal + EchSin[a];
           costotal := costotal + EchCos[a];
      end;
      sineav:=sintotal/128;
      cosav:=costotal/128;
end;
end;

{Derive amplitude profile from phase data}
procedure MakeAmps;
var x:integer;
a,b:byte;
begin
with BFlatRecData do begin
for x:= 1 to 128 do
begin
     av:=echcos[x]-cosav;
     bv:=echsin[x]-sineav;
     amp[x]:=sqrt(av*av+bv*bv);
end;
end;
end;

{Arctan from -Pi:Pi}
function Atan2(Y,X:real):real;
var ax,ay,Phi:real;
    begin
         if(X=0.0) and (Y=0.0)
         then Atan2:=0.0
         else
         begin
```

```
            ax:= abs(X); ay:= abs(Y);
            if(ax>ay) then Phi:= Arctan(ay/ax)
                      else Phi:=(Pi/2)-Arctan(ax/ay);
            if(X<0.0) then Phi:=Pi-Phi;
            if(Y<0.0) then Phi := -Phi;
            Atan2:= Phi;
        end;
    end;

{Remove A/D induced DC and unwraps the phase data}
procedure CalcPhases;
begin
      with BflatRecData do begin
      for a:=1 to 128 do
      begin
            cosine:=EchCos[a]-cosav;
            sine:=Echsin[a]-sineav;
            txcosine:=txcos[a]-cosav;
            txsine:=txsin[a]-sineav;
           EchPhW[a]:=Atan2(sine,cosine);
           TxPhW[a]:=Atan2(txsine,txcosine);
      end;

      end;
      n:=0;
      txph[1]:=TxPhW[1];
      a:=2;
      while a<129 do
            begin
                  if TxPhW[a]-TxPhW[a-1] > Pi then n:=n-1;
                  if TxPhW[a]-TxPhW[a-1] <=-Pi then n:=n+1;
                  TxPh[a]:=TxPhW[a]+(n*2*Pi);
                  inc(a);
            end;
      n:=0;
      echph[1]:=EchPhW[1];
      a:=2;
      while a<129 do
            begin
                  if EchPhW[a]-EchPhW[a-1] > Pi then n:=n-1;
                  if EchPhW[a]-EchPhW[a-1] <=-Pi then n:=n+1;
```

```
                    EchPh[a]:=EchPhW[a]+(n*2*Pi);
                    inc(a);
                end;
end;
{Calculates sign of a number}
function Sign(Number : real) : real;
   {returns -1 if Number < 0,or +1 if Number >= 0 }
   begin
         if Number= 0.0
            then Sign:= 1
            else Sign:=Abs(Number)/Number
   end;

{Determines boundaries of good phase data from within which
 good wind measurement can be made}
procedure FindScanLimits;
const
    alittlebit=1.5;
var
    EchDeriv,Ech2Deriv                    : realarrayNData;
    i,stp,stt,range,biggestrange          : integer;
    foundstart                            : boolean;
begin
Differentiate(EchPh,EchDeriv,Ech2Deriv);
FoundStart:=false;
stt:=1;
biggestrange:=0;
   for i:= 2 to 127 do begin
       if (abs(EchDeriv[i]-EchDeriv[i+1]) < alittlebit)
           and (sign(EchDeriv[i])=sign(EchDeriv[i+1])) then
    { if abs(Ech2Deriv[i])<2 then}
       begin
            if i=127 then begin
                        stp:=i;
                        range:=stp-stt;
                        if range>biggestrange then  begin
                                start:=stt;
                                stop:=stp;
                                stt:=i;
                                BiggestRange:=Range;
                               end;
```

```
                                end;
            end
            else
            begin
            stp:=i;
            range:=stp-stt;
            if range>biggestrange then  begin
                                        start:=stt;
                                        stop:=stp;
                                        stt:=i;
                                        BiggestRange:=Range;
                                        end
                                        else stt:=i;
         end;
      end;

end;

{Refer Numrical Recipes}
function gammln(xx: real): real;
const
   stp  =  2.50662827465;
var
   x,tmp,ser: double;
begin
   x := xx-1.0;
   tmp := x+5.5;
   tmp := (x+0.5)*ln(tmp)-tmp;
   ser := 1.0+76.18009173/(x+1.0)-86.50532033/(x+2.0)+24.01409822/
          (x+3.0)-1.231739516/(x+4.0)+0.120858003e-2/(x+5.0)-
           0.536382e-5/(x+6.0);
   gammln := tmp+ln(stp*ser)
end;

{Refer Numrical Recipes}
procedure gcf(a,x: real;
   var gammcf,gln: real);
label 99;
const
   itmax = 100;
   eps = 3.0e-7;
```

```
var
   n: integer;
   gold,g,fac,b1,b0,anf,ana,an,a1,a0: real;
begin
   gln := gammln(a);
   gold := 0.0;
   a0 := 1.0;
   a1 := x;
   b0 := 0.0;
   b1 := 1.0;
   fac := 1.0;
   for n := 1 to itmax do begin
      an := 1.0*n;
      ana := an-a;
      a0 := (a1+a0*ana)*fac;
      b0 := (b1+b0*ana)*fac;
      anf := an*fac;
      a1 := x*a0+anf*a1;
      b1 := x*b0+anf*b1;
      if a1 <> 0.0 then begin
         fac := 1.0/a1;
         g := b1*fac;
         if abs((g-gold)/g) < eps then GOto 99;
         gold := g
      end
   end;
   writeln('pause in GCF - a too large, itmax too small');
   readln;
99:
   gammcf := exp(-x+a*ln(x)-gln)*g
end;

{Refer Numrical Recipes}
procedure gser(a,x: real;
    var gamser,gln: real);
label 99;
const
   itmax = 100;
   eps = 3.0e-7;
var
   n: integer;
```

```
    sum,del,ap: real;
begin
    gln := gammln(a);
    if x <= 0.0 then begin
        if x < 0.0 then begin
            writeln('pause in GSER - x less than 0');
            readln
        end;
        gamser := 0.0
    end
    else begin
        ap := a;
        sum := 1.0/a;
        del := sum;
        for n := 1 to itmax do begin
            ap := ap+1.0;
            del := del*x/ap;
            sum := sum+del;
            if abs(del) < abs(sum)*eps then GOto 99
        end;
        writeln('pause in GSER - a too large, itmax too small');
        readln;
99:     gamser := sum*exp(-x+a*ln(x)-gln)
    end
end;

{Refer Numrical Recipes}
function gammq(a,x: real): real;
var
    gamser,gammcf,gln: real;
begin
    if (x < 0.0) OR (a <= 0.0) then begin
        writeln('pause in GAMMQ - invalid arguments');
        readln
    end;
    if x < a+1.0 then begin
        gser(a,x,gamser,gln);
        gammq := 1.0-gamser
    end
    else begin
        gcf(a,x,gammcf,gln);
```

```
      gammq := gammcf
   end
end;

{Least squares fit routine: refer Numrical Recipes}
procedure fit(var x,y: realarrayNDATA;
                 ndata: integer;
              var sig: realarrayNDATA;
                  mwt: integer;
          var a,b,siga: real;
       var sigb,chi2,q: real);
var
   i: integer;
   wt,t,sy,sxoss,sx,st2,ss,sigdat: real;
begin
   sx := 0.0;
   sy := 0.0;
   st2 := 0.0;
   b := 0.0;
   if mwt <> 0 then begin
      ss := 0.0;
      for i := 1 to ndata do begin
         wt := 1.0/sqr(sig[i]);
         ss := ss+wt;
         sx := sx+x[i]*wt;
         sy := sy+y[i]*wt
      end
   end
   else begin
      for i := 1 to ndata do begin
         sx := sx+x[i];
         sy := sy+y[i]
      end;
      ss := ndata
   end;
   sxoss := sx/ss;
   if mwt <> 0 then begin
      for i := 1 to ndata do begin
         t := (x[i]-sxoss)/sig[i];
         st2 := st2+t*t;
         b := b+t*y[i]/sig[i]
```

```
      end
   end
   else begin
      for i := 1 to ndata do begin
         t := x[i]-sxoss;
         st2 := st2+t*t;
         b := b+t*y[i]
      end
   end;
   b := b/st2;
   a := (sy-sx*b)/ss;
   siga := sqrt((1.0+sx*sx/(ss*st2))/ss);
   sigb := sqrt(1.0/st2);
   chi2 := 0.0;
   if mwt = 0 then begin
      for i := 1 to ndata do
         chi2 := chi2+sqr(y[i]-a-b*x[i]);
      q := 1.0;
      sigdat := sqrt(chi2/(ndata-2));
      siga := siga*sigdat;
      sigb := sigb*sigdat
   end
   else begin
      for i := 1 to ndata do
         chi2 := chi2+sqr((y[i]-a-b*x[i])/sig[i]);
      q := gammq(0.5*(ndata-2),0.5*chi2)
   end;
end;

{Determines the phase gradient}
procedure FindGradients;
var middle,i,ndata,mwt                    : integer;
    x,y,sig                               : realarrayNdata;
    a,b,siga,sigb,chi2,q                  : real;

begin
if stop-start>=MinNumOfTransPulses then
begin
     mwt:=0;
     for i:=1 to NDataP do x[i]:=i/379;
     ndata:=ndatap;
```

```
      fit(x,txph,ndata,sig,mwt,a,b,sigb,sigb,chi2,q);
      TxGrad:=b;
      TxUncertainty:=SIGb;
      for i:=start to stop do begin
                              x[i-start+1]:=i/379;
                              PartialEchPh[i-start+1]:=EchPh[i];
                              end;
      ndata:=stop-start+1;
      fit(x,PartialEchPh,ndata,sig,mwt,a,b,sigb,sigb,chi2,q);
      EchGrad:=b;
      EchoUncertainty:=SIGb;
      Totuncertainty:=Echouncertainty + txuncertainty;
end;
end;

{Coodinates wind velocity calculation}
procedure FindVelocity;
var p:real;
hourstr,minstr,secstr:string;
begin
if stop-start<MinNumOfTransPulses then
   begin
        if stop-start>=0 then
          begin
           if not fancy then write(path,' [',stop-start,']
                                   Too noisy to compute: ')
          end
          else
          begin
            if not fancy then
                write(path,' [0]  Too noisy to compute: ');
          end;
      p:=(100*((FileNo) / NoOfFiles));
      if not fancy then writeln(p:3:2,' % HourFileFinished');
      inc(FIlesOpened);
      inc(RejFile);
end

else
begin
      velocity:=(wavelength/(4*pi))*(echgrad-txgrad);
```

```
      veluncertainty:= (0.5*(wavelength/(2*pi))*totuncertainty);
      if not fancy then writeln(path,' [',stop-start,']   ',
        velocity:4:1,' ',chr(241),' ',veluncertainty:4:1 ,' m/s');
      with BFlatRecData do
      begin
           str(hour,hourstr);
           str(min,minstr);
           str(sec,secstr);
           if hour<10 then hourstr:='0'+hourstr;
           if min<10 then minstr:='0'+minstr;
           if sec<10 then secstr:='0'+secstr;
           if VelUncertainty < 4 then begin
              write(velsout,date,' ');
              write (velsout,hourstr+minstr+secstr,' ');
          end;
      end;
      if VelUncertainty < 4 then
      begin
         write(velsout,velocity:10:1);
         write(velsout,veluncertainty:10:1);
         write(velsout,BFlatRecData.Bin:6);
         write(velsout,BFlatRecData.Number:5);
         write(VelsOut,Stop-Start:5);
         write(VelsOut,Start:5);
         writeln(VelsOut,Stop:5);
         inc(AccFile);
         inc(FilesOpened);
      end;
      if veluncertainty>=4 then inc(RejFile);
      lastnumber:=BFlatRecData.Number;
end;
end;

{Determines available files}
procedure FindFiles;
type
 ra = array[1..MaximumNumberOfFiles] of string[13];
var
 S,chk :string;
 Temp  :string[13];
 Store : array[0..23] of ^ra;
```

```
 best,
 i,j,k : integer;
 N     : array[0..23] of integer;

 DirInfo:SearchRec;
begin
assign(FileDir,'c:\Dop128\Files.Dir');
rewrite (FileDir);
     chdir(Drive+':');
     chdir('\');
     chdir('DorfData');
     {$I-}
     chdir(date);
     if IOResult <> 0 then
        begin
        writeln;
        writeln('Can''t find ',Drive,':\DorfDATA\',date);
        FilesExist := False;
        end
        else
        begin
             FilesExist := true;
             FindFirst('WIND*.*', Archive, DirInfo);
             GetDir(DriveNo,S);
             fillchar(N,sizeof(N),0);
             for k:=0 to 23 do
                 getmem(Store[k],MaximumNumberOfFiles*14);
             while doSERROR=0 do
                   begin
                   val(copy(DirInfo.Name,6,2),i,j);
                   inc(N[i]);
                   Store[i]^[N[i]]:=DirInfo.Name;
                   FindNext(DirInfo);
                   end;
             NoOfFiles:=0;
             for k:=0 to 23 do
                 begin
                 for i:=1 to N[k]-1 do
                       begin
                       best:=i;
                       for j:=i+1 to N[k] do
```

```
                        if Store[k]^[j]<Store[k]^[best] then
                           best:=j;
                     if best<>i then
                        begin Temp:=Store[k]^[best];
                           Store[k]^[best]:=Store[k]^[i];
                           Store[k]^[i]:=temp; end;
                     end;
               for i:=1 to N[k] do
                     begin
                       inc(NoOfFiles);
                       writeln (FileDir,S,'\',Store[k]^[i],' ');
                       inc(NoAnalF);
                    end;
               end;
         end;
            close(FileDir);
            reset(FileDir);
             for k:=0 to 23 do
                 freemem(Store[k],MaximumNumberOfFiles*14);
end;

{Initialising files}
procedure Setvariables;
          begin
             {$I-}
             Mkdir('E:\analysed');
             MkDir('E:\analysed\'+SaveDate);
             if IOResult <> 0 then writeln('');
             assign(velsout,'E:\analysed\'+SaveDate+'\Phvels.dat');
             rewrite(velsout);
          end;

{Allows invisible cursor}
procedure setcursortype(numver:word);
            inline($59/$B4/$01/$CD/$10);

{Produces a nice gui for program - not good for testing}
procedure dofancy;
var
  mettime, metdate:string;
begin
```

```
  SetCursortype(NoCursor);
  textBackground(Black);    { Clear screen }
  ClrScr;
    metdate:=copy(path,13,6);
    writeln('DATE: ',metdate);
    mettime:=copy(path,20,7);
    writeln('Meteor File: ',mettime);
    writeln('Total No of Files: ',NoAnalF);
    writeln;
    writeln;
    writeln ('No of Files Opened: ',FilesOpened);
    writeln('No of Files Accepted: ',accfile);
    writeln{('No of Files Rejected: ',RejFile)};
    if filesOpened <>0 then writeln('Success Percentage:
                                    ',(accfile*100)div(filesOpened),'%');
    writeln;
    writeln('Computing...');
end;


{Determines next file for analysis}
procedure NextFile;
begin
  if not eof(filedir) then
    begin
     DoFancy;
     HourFileFinished:=false;
     FileNo := FileNo+1;
     {$I-}
     close(BFlatFIle);
     {$I+}
     if IOResult <>0 then begin end;
     readln(FileDir,path);
     FileMode:=0;
     assign(BFlatFile,Path);
     reset(bflatfile);
     DayFilesLeft:=true;
    end
   else
    begin
     {$I-}
```

```
      close(BFlatFIle);
      {$I+} if IOResult <>0 then begin end;
      {$I-}
      close(velsout);
      {$I+}if IOResult <>0 then begin end;
      {$I-}
      close(FileDir);
      {$I+} if IOResult <>0 then begin end;
      DayFilesLeft:= false;
      HourFileFinished:=true;
      {$I-}
      if eof(datefile) then
         close(datefile);
         {$I+} if IOResult <>0 then begin end;
    end;
end;

{Extracts meteor data from hourly files}
procedure OrganiseData;
var time,date:string;
begin
{$I-}
read(BflatFile,BFlatRecData);
{$I+}
if IOResult=0 then
              begin
              if eof(BFlatFile) then HourFileFinished:=true;
              DataAvailable:=true;
              end
              else begin
                   repeat
                   NextFile;
                   DoFancy;
                   {HourFileFinished:=true; }
                   {$I-}
                   read(BflatFile,BFlatRecData);
                   {$I+}
                   until (IOResult=0) or (DayFilesLeft=False);
                   if IOResult=0 then DataAvailable:=True;
                   if DayFilesLeft=False then DataAvailable:=False;
                   end;
```

```
if not fancy then
   begin
        writeln;
        writeln('Range Bin No: ',bflatrecdata.bin);
   end;

end;
{Coordinates wind speed calculation for a days files}
procedure DoAnalysis;
var day:integer;
    a:integer;
begin
    SetCursortype(NoCursor);
          FindFiles;
          if FilesExist then
          begin
          Setvariables;
          repeat
                NextFile;
                {organisedata;}
                if DayFilesLeft then
                   begin
                        if(fancy) and ((FileNo=1) or (FileNo=
                          NoAnalF) or ((FilesOpened) mod 100=0))
                        then dofancy;
                        repeat
                        OrganiseData;
                        if DataAvailable then
                        begin
                             if Bflatrecdata.number<>lastnumber
                             then begin
                                  FindAverages;
                                  MakeAmps;
                                  CalcPhases;
                                  FindScanLimits;
                                  FindGradients;
                                  FindVelocity;
                             end;
                        end;
                        if not (fancy) then begin
                        repeat until keypressed;
```

```
                    if keypressed then
                                    begin
                                    c:=readkey;
                                    if upcase(c)='S' then
                                    begin
                                      HourFileFinished:=true;
                                      DayFilesLeft:=false;
                                    end;
                                    end;
                                    end;
                    until HourFileFinished;
                end;
            until not DayFilesLeft;
            end;   {of filesexisting}
end;

{Routine called from outside of Unit}
procedure FindWindSpeeds(Dir,SaveDir:string7);
begin
    date:=Dir;
    SaveDate:=SaveDir;
    fancy := true;
    NoAnalF:=0;
    AccFile:=0;
    FilesOpened:=0;
    RejFile:=0;
    LastNumber:=0;
    Clrscr;
    Setup;
    DoAnalysis;
    DoFancy;
end;

begin  {No Initialisation}
end.
```

A.2.2 Unit Heights

The Borland Turbo Pascal Unit, `Heights.pas`, is called from the control programme `System.pas` and calculates range, elevation angle and height of the meteor echo's

specular reflection point.

```
Unit Heights;
interface
uses  OrbDef93, {PhaseReN,} PhaseReZ, Dos, GtlStv, Crt,Defns;

procedure FindHeightsEtc(Dir:string7);
implementation

{GtlStv is same as genutil except that format for datestr
 and timestr has been changed slightly i.e. the removal of
  unwanted spaces S. Marsh sep 95}

{label} {999;} {777;}     {Reject= True, Imax to in range 20-120}

const
        DeltaCalib = 0;
        EarthsRadius = 6468;
        AerialSpacing14 = 3.00;
        AerialSpacing15 = 11.3;
        CosElevWindow = 0.036;{Allowed window of acceptance between
         e.g. CosElev142 and CosElev1510 to confirm which phase15
         lobe the echo Elev lies in. value 0.020 means 20% of
         angular dist to next lobe limit, 0.025 31%, 0.030 40% }
        F15 = 1.0;  {to get final AltitudeTru and ElevationTru from
         the two measuremments using Phase14 & Phase15 we average
         with weighting the two values. F15 is weight for Phase15 so
         (1-F15) is weighting for Phase14 (quite apart from
         Phase14 used to narrow down correct Phase15}



type    array250 = array[0..250] of longint;
        array360 = array[0..360] of longint;
        array180 = array[0..180] of longint;
        array7_12 = array[70..120] of longint;

var     DirName,FileName : string;  {Location of input data file}
        HourName : string;          {File name less the extension}
        StartDir,endDir,SearchDir                    : string;
        Start1File,end1File,Start2File,end2File      : string;
```

```
StartHour, endHour, StartFile, endFile        : string;
InName, OutName                               : string;
InData, OutFile                               : file of Observation;
PostName                                      : string;
SFile, DFile                                  : SearchRec;
SearchName                                    : string;
Dir1, Dir2, Dir3, Dir4, Dir5, Dir6, Dir7, Dir8, Dir9   : string;
Answer                                        : char;
i, j                                          : word;
InFile                                        : file of Observation;
Buffer                                        : ReducedData;
SmoothBuff                                    : Observation;
RecNum                                        : longint;
Outtext, Outtext1, Outtext3, Outtext4, Outtext5,        : text;
Outtext6 Intext                                         : text;
Ilimit, Num14Low, Num14High, Num14Mid                   : byte;
Sum14Low, Sum14High, Sum14Mid                           : real;
Mean14Low, Mean14High, Mean14Mid                        : real;
SLow, Shigh, SumSqrLow, SumSqrHigh, StandDevLow,        : real;
StandDevHigh, StandDev14, StandDev15                    : real;
Phase14Low, Phase14High, Phase14Mid                     : real;
MeanTinLowV, MeanTinHighV, MeanTinMidV                  : real;
DiffMeans14, DiffMeans15, DiffPhase14                   : real;
NumPhase14                                              : byte;
SumPhase14, SumSqrPhase14, MeanPhase14                  : byte;
StandDevPhase14, X, SS, MeanPhase14Uncert               : real;
Phaz14, Phase14, ErrPhase14, Phase14Diff, Phase15Diff   : real;
Num15Low, Num15High, Num15Mid                           : byte;
Mean15Low, Mean15High, Mean15Mid                        : real;
Phase15Low, Phase15High, Phase15Mid, Phaz15, Phase15,   : real;
ErrPhase15, Phazz15                                     : real;
Outtext2                                                : text;
Range, Lag23                                            : real;
CosElev141, CosElev142, CosElev140                      : real;
AltFlat                                                 : real;
Elev141, Alt141, Elev142, Alt142, Elev140, Alt140,
Alt14Tru, Elev14Tru, Alt142_R                           : real;
Alt142array                                             : array250;
Alt141array                                             : array250;
Alt140array                                             : array250;
AltTruearray                                            : array250;
```

```
NumAlt141, NumAlt142, NumAlt140, NumAltTru          : longint;
Phase14array                                        : array360;
Phase15array                                        : array360;
ElevXYarray                                         : array180;
HomeAltarray                                        : array7_12;
Total                                               : longint;
AutoScale                                           : boolean;
NumPhases                                           : longint;
Reject14, Reject15, Accept                          : boolean;
ValueR                                              : real;
Value                                               : byte;
Num1, Num2, NumNot, Num14, Num15                    : longint;
DetNutt, DetSpit                                    : real;
Method14, Method15                                  : byte;
Elev1510, Elev159, Elev158, Elev157, Elev156, Elev155,
Elev154, Elev153, Elev152, Elev151, Alt1510, Alt159,
Alt158, Alt157, Alt156, Alt155, Alt154, Alt153, Alt152,
Alt151                                              : real;
ElevTru, AltTru                                     : real;
ElevTry2, ElevTry1, ElevTry0                        : real;
AltTry2, AltTry1, AltTry0                           : real;
NumBadTry2, NumBadTry1, NumBadTry0, NumBadSelect    : integer;
Count142, Count141                                  : longint;
Start, Limit                                        : byte;
ElevImax, Elevmax                                   : integer;
ElevImaxTru                                         : real;
IO                                                  : integer;
Cal14Phase1, Cal14Phase2, Cal14Phase3, Cal14Phase4,
Cal14Phase5, Cal14Phase6                            : real;
HourSet, DateLine, DateNow                          : string;
got                                                 : boolean;
code                                                : integer;
TotalCount                                          : integer;
Error                                               : word;
Difference                                          : real;
Direction, Elevdirection                            : real;
Ph14const, Ph15const, Alt142F, AltTruF, Alt142B, AveFdeg: real;
DateAnalysing                                       : longint;
PhaseExpect, PhaseExpectU, PhaseExpectL, Phase14B   : real;
CommentF, CommentB                                  : string;
RatioF, RatioB                                      : real;
```

```
        forwardOK, Circuits, Selection                                  : byte;
        Toty1, Toty2, Toty3, Losttot                                    : integer;
        LagGood, Front, Back, Ambiguous, AgreeFront, AgreeBack,
        GoodFront, GoodBack                                             : boolean;
        Tot1, Tot2, Tot3, Tot4, Not12, FrontTotal, BackTotal            : integer;
        Alt142Av1, Alt142Av2, Alt142Av3, Alt142Av4, Alt142Av5,
        Elev15, Alt142PkF, AltTruPkF, Alt142PkB, ElevTruS,
        ElevTruN, AltTruS, AltTruN                                      : real;
        Numfor, NumTru, Numbak                                          : longint;
        countAltTru3S, countAltTru3N, countAltTru4S, count
        AltTru4N                                                        : longint;
        ElevTry2S, ElevTry1S, ElevTry0S, AltTry2S, AltTry1S,
        AltTry0S                                                        : real;
        ElevTry2N, ElevTry1N, ElevTry0N, AltTry2N, AltTry1N,
        AltTry0N                                                        : real;
        Elev142S, Elev141S, Elev1510S, Elev159S, Elev158S,
        Elev157S, Elev156S, Elev155S, Elev154S, Elev153S,
        Elev152S, Elev151S, Elev150S                                    : real;
        Elev142N, Elev141N, Elev1510N, Elev159N, Elev158N,
        Elev157N, Elev156N, Elev155N, Elev154N, Elev153N,
        Elev152N, Elev151N, Elev150N                                    : real;
        Alt142ModeF, AltTruModeF, Alt142ModeB                           : real;
        loop                                                            : shortint;
        Elev150, Alt150, Phazz14, Diff2, Diff1, Diff0                   : real;
{$R+}
{$S+}
procedure WhichWay(var Range: integer;
                   var y:real;var PhaseExpectU:real;
                   var PhaseExpectL:real);
var x    : integer;
    bott,U,L    : real;
begin
  {case 1 of front echo:}
   x:=Range;
    bott:= x*( 0.54 + ( (x-168)/(137))*0.16  );
    y:= 32 - (sqr(222-x))/bott;
   y:= x + y;
   U:= 1.82/(raise((0.01*x),3));
   L:= 1.80/(raise((0.01*x),3));
 {because the elevation given by Elev14 is too,low lower than
  about 21 deg(see change in GetElevAlt1415 in PhaseRed) then
```

```
  need to increase upper limit}
  if (x > 250) then U:= U + (0.07);
   PhaseExpectU:= y +  (x*U);
   PhaseExpectL:= y -  (x*L);
end;
procedure FindHeightsEtc (Dir:string7);
label 999;
var month:string;
begin
StartDir:=Dir;
endDir:=Dir;
{$I-}
MkDir('E:\Analysed\'+Dir);
{$I+}
if IOResult <> 0 then writeln('Directory already created');
assign(Outtext2,'E:\analysed\'+Dir+'\Amor.dat');
rewrite(Outtext2);
Start1File:='NZST_00';
                        end1File:=   'NZST_x';
                        Start2File:= 'NZST_y';
end2File:='NZST_23';
AutoScale:=True;
assign(Intext,'C:\Dop128\CalNew14.cal');
ClrScr;
writeln;
writeln;
writeln('Calculating Height Information for: ',Dir);
FillChar(Alt142array,SizeOf(Alt142array),#0);
FillChar(Alt141array,SizeOf(Alt141array),#0);
FillChar(Alt140array,SizeOf(Alt140array),#0);
FillChar(AltTruearray,SizeOf(AltTruearray),#0);
FillChar(Phase14array,SizeOf(Phase14array),#0);
FillChar(ElevXYarray,SizeOf(ElevXYarray),#0);
FillChar(HomeAltarray,SizeOf(HomeAltarray),#0);
FillChar(Phase15array,SizeOf(Phase15array),#0);
TotalCount:= 0;error:=0;
StartFile:=Start1File;  endFile:=end2File;
SearchDir:= Drive +':\AMORDATA\*';
FindFirst(SearchDir,Directory,DFile);
while DosError = 0 do if (DFile.Name < StartDir)
    or (DFile.Name > endDir) then  FindNext(DFile)
```

```
else begin
DirName:= Drive+':\AMORDATA\' + DFile.Name;
reset(Intext);
got:=false;
 readln(Intext);readln(Intext);readln(Intext);readln(Intext);
 readln(Intext);readln(Intext);readln(Intext);readln(Intext);
 readln(Intext);readln(Intext);readln(Intext);
 readln(Intext); {the Date, Ph14 etc line...}
while (not eof(Intext)) and (not got) do begin
      readln(Intext,DateLine);
        DateNow:=copy(DateLine,1,6);
        if (DateNow = DFile.Name) then got:= true;
end;
val(copy(DateLine,10,3),Ph14const,code);
val(copy(DateLine,16,3),Ph15const,code);
val(copy(DateLine,26,5),Alt142F,code);
val(copy(DateLine,33,5),AltTruF,code);
val(copy(DateLine,40,5),Alt142B,code);
val(copy(DateLine,47,5),AveFdeg,code);
SearchName:= DirName + '\NZST_*.';
FindFirst(SearchName,AnyFile,SFile);
while DosError = 0 do
  if (SFile.Name + '1' >= Start1File)
       and (SFile.Name <= end2File)
  then begin
    assign(InFile,DirName +'\'+ SFile.Name);
    {$I-}
    reset(InFile);
    {$I+}
    IO:=IOResult;
    if IO <> 0 then begin
    write('Problem at line 257 of Heights');
    write('  IOResult = ',io);
    halt;
    end;
    writeln('Processing file :  ',DirName +'\'+ SFile.Name);
    while not eof(InFile) do begin
      RecNum:= FilePos(InFile);
      read(InFile,ObsvBuff);
      FillChar(Buffer,SizeOf(Buffer),#0);
      TransferObsvData(ObsvBuff,Buffer);
```

```
with ObsvBuff do
   DeSpikeProfile(Home.Profile,N1);
   TriSmooth(ObsvBuff.Home, SmoothBuff.Home,3);
   with Buffer.Home do FindMax(SmoothBuff.Home,Imax,Max,250);
        Get_Ranges(ObsvBuff, Buffer, Error);
        case Error of
        23 : Comment:='Range < 50 or > 400 kms';end;
        if Error = 23 then goto 999;
       {This section finds the Phases and so elevation angle.}
        Phase14:=99; Phase15:=77;
        ErrPhase14:=0; ErrPhase15:=0;
        with Buffer do
        GetPhase(ObsvBuff.Tin,Home.Imax,{Value,}Num14Low,
                 Num14High,Num14Mid,Phaz14,Phase14Diff,
                 ErrPhase14,Phase14Low,Phase14High,Phase14Mid,
                 Mean14Low,Mean14High,DiffMeans14,Mean14Mid,
                 StandDev14,Method14,Reject14);
        if (Reject14 = True) then begin
           Phase14 :=999;
           Elev142:=999;
           Alt142:=999;
           Elev141:=999;
           Alt141:=999;
           GoTo 999;
        end;
        Phase14:=(360-Phaz14)+ph14const+(8*AveFDeg);
        if (Phase14 < 0)   then Phase14:= 360+Phase14;
        if (Phase14 > 360) then Phase14:= Phase14 - 360;
        with Buffer do
        GetPhase(ObsvBuff.Tos,Home.Imax,{Value,}Num15Low,
                 Num15High,Num15Mid,Phaz15,Phase15Diff,
                 ErrPhase15,Phase15Low,Phase15High,Phase15Mid,
                 Mean15Low,Mean15High,DiffMeans15,Mean15Mid,
                 StandDev15,Method15,Reject15);
       if (Reject15 = True) then begin
          Phase15 :=999;
          Elev152:=999;
          Alt152:=999;
          Elev151:=999;
          Alt151:=999;
          GoTo 999;
```

```
            end;
            Phase15:= (360-Phaz15) + Ph15const;
            if (Phase15 < 0)   then Phase15:= 360+Phase15;
            if (Phase15 > 360) then Phase15:= Phase15 - 360;
           {Check that meteor is not travelling closely east-west}
            if ((Buffer.Home.Max > 5) and ( Buffer.Home.range >= 100)
                and (Phase14 >= 0) and (Phase14 < 360)
                and (Phase15 >= 0) and (Phase15 < 360))
                                    then begin
            error:=0;
            if ( (Phaz14 >=25) and (Phaz14 < 325)) and
               (StandDev14 > 8) then error:= 1;
            if ( (Phaz14 >=325) or (Phaz14 < 25)) and
               (StandDev14 > 15) then error:= 1;
            if ( (Phaz15 >=25) and (Phaz15 < 325)) and
               (StandDev15 > 12) then error:= 1;
            if ( (Phaz15 >=325) or (Phaz15 < 25)) and
                (StandDev15 > 18) then error:= 1;
            if (error > 0) then goto 999;
            inc(TotalCount);
            ElevDirection:= -Direction;
            val(StartDir,DateAnalysing,Code);
        {if the range is >= 140 we can decide the approx elevation
         by comparing the actual range and Phase14....}
            WhichWay(Buffer.Home.Range,PhaseExpect,
                     PhaseExpectU, PhaseExpectL);
            Phase14B:=360-Phase14;
            if ((Phase14 - PhaseExpect) > 0) then
            RatioF:=  (Phase14-PhaseExpect)/
                       (PhaseExpectU-PhaseExpect);
            if ((Phase14 - PhaseExpect) < 0) then
            RatioF:=  (Phase14-PhaseExpect)/
                       (PhaseExpect-PhaseExpectL);
            if ((Phase14B - PhaseExpect) > 0) then
            RatioB:=  (Phase14B-PhaseExpect)/
                       (PhaseExpectU-PhaseExpect);
            if ((Phase14B - PhaseExpect) < 0) then
            RatioB:=  (Phase14B-PhaseExpect)/
                       (PhaseExpect-PhaseExpectL);
           {Test if we have obvious front/back/both}
            Front:= False;Back:= False; Ambiguous:= False;
```

```
    CommentF:='Fbefore';CommentB:='Bbefore';
    if ( Abs(RatioF) <= 1.25) and (Abs(RatioB) > 1.25)
       then begin
            Front:=TRUE; Comment:= 'FRONT'; end
    else CommentF:='NOTfront';
    if ( Abs(RatioB) <= 1.25) and (Abs(RatioF) > 1.25)
       then begin
            Back:= TRUE;
            Comment:= 'BACK'; end
       else CommentB:='NOTback';
    if ( Abs(RatioF) <= 1.25) and (Abs(RatioB) < 1.25)
     then begin
       Ambiguous:= TRUE; Comment:= 'Ambiguous'; end;
  {Test whether Lag23 is good}
    LagGood:=FALSE;
    with Buffer do begin
     if (Home.Max   > 4*Home.Noise)
     and (Nutt.Max > 4*Spit.Noise)
     and (Spit.Max > 4*Spit.Noise)
     and (Lag23 < 200) and (abs(Lag23) > 4) then
        LagGood:=TRUE;
  end;
{case 1}
 if LagGood then begin
    Selection:=1; Circuits:=1;  inc(Tot1); end
 {case 2}
 else  if (Buffer.Home.Range >= 140) and (not LagGood)
       and (not Ambiguous) then begin
           Selection:=2; Circuits:= 1; inc(Tot2); end
{case 3}
 else if (Buffer.Home.Range >= 140) and (not LagGood)
           and (Ambiguous) then begin
      Selection:= 3; Circuits:= 2; inc(Tot3); end
{case 4}
 else if (Buffer.Home.Range < 140) and not LagGood then
       begin
           Selection:= 4; Circuits:= 2; inc(Tot4); end
{case 5}
 else  begin
           Selection:= 5; inc(LostTot);  goto 999; end;
 inc(Toty3);
```

```
ElevTruS:= 0.5; ElevTruN:= 0.5; AltTruS:= 0.5;
AltTruN:= 0.5; ElevTry2S:=0.3; Elevtry1S:=0.3;
ElevTry0S:= 0.3; ElevTry2N:= 0.3; Elevtry1N:=0.3;
ElevTry0N:=0.3;
for loop:= 1 to Circuits do begin
if (Selection = 1) then begin
  Buffer.Lag23:= Buffer.Lag23;
end;
if (Selection = 2) then begin
 if Front then Buffer.Lag23:= -100;
 if Back then Buffer.Lag23:= +100;
end;
if (Selection = 3) then begin
 if (loop = 1) then Buffer.Lag23:= -100;
 if (loop = 2) then Buffer.Lag23:= +100;
end;
if (Selection = 4) then begin
 if (loop = 1) then Buffer.Lag23:= -100;
 if (loop = 2) then Buffer.Lag23:= +100;
end;
GetElevAlt1415(Buffer.Lag23,Phase14,Phase15,Buffer.Home.
Range,Elev142,Elev141,Elev140,ElevTry2,ElevTry1,ElevTry0,
AltTry2,AltTry1,AltTry0,Elev1510,Elev159,Elev158,Elev157,
Elev156,Elev155,Elev154,Elev153,Elev152,Elev151,Elev150,
Alt142,Alt141,Alt140, Alt142_R,Alt1510,Alt159,Alt158,
Alt157,Alt156,Alt155,Alt154,Alt153,Alt152,Alt151,Alt150,
Phazz14,Phazz15,Diff2,Diff1,Diff0);
GetTruElevAlt(ElevTry2,ElevTry1,ElevTry0,AltTry2,AltTry1,
AltTry0,Alt142_R,ElevTru,AltTru,NumBadTry2,NumBadTry1,
NumBadTry0,NumBadSelect);
if (Selection <= 2) and (AltTru > 70) and (AltTru < 120) then
  begin
    ElevTru:=ElevTru; AltTru:= AltTru;Elev142:=Elev142;
    Elev141:=Elev141;ElevTry2:=ElevTry2; ElevTry1:=ElevTry1;
    ElevTry0:=ElevTry0;AltTry2:=AltTry2; AltTry1:=AltTry1;
    AltTry0:=AltTry0;Elev1510:=Elev1510; Elev159:=Elev159;
    Elev158:=Elev158;Elev157:=Elev157; Elev156:=Elev156;
    Elev155:=Elev155; Elev154:=Elev154; Elev153:=Elev153;
    Elev152:=Elev152; Elev151:=Elev151;Elev150:= Elev150;
  end;
if (Selection = 3) or (Selection = 4) then
```

```
          begin
           if (Loop = 1) then begin
          {USE SOUTH 1 AND NORTH 2... }
           ElevTruS:= ElevTru;AltTruS:= AltTru;Elev142S:=Elev142;
           Elev141S:=Elev141;ElevTry2S:=ElevTry2; ElevTry1S:=ElevTry1;
           ElevTry0S:=ElevTry0;AltTry2S:=AltTry2; AltTry1S:=AltTry1;
           AltTry0S:=AltTry0;Elev1510S:=Elev1510; Elev159S:=Elev159;
           Elev158S:=Elev158; Elev157S:=Elev157; Elev156S:=Elev156;
           Elev155S:= Elev155; Elev154S:=Elev154; Elev153S:=Elev153;
           Elev152S:=Elev152; Elev151S:=Elev151; Elev150S:= Elev150;
          end;
        if (Loop = 2) then
          begin
           ElevTruN:= ElevTru;AltTruN:= AltTru;Elev142N:=Elev142;
           Elev141N:=Elev141;ElevTry2N:=ElevTry2; ElevTry1N:=ElevTry1;
           ElevTry0N:=ElevTry0;AltTry2N:=AltTry2; AltTry1N:=AltTry1;
           AltTry0N:=AltTry0;Elev1510N:=Elev1510; Elev159N:=Elev159;
           Elev158N:=Elev158; Elev157N:=Elev157; Elev156N:=Elev156;
           Elev155N:= Elev155; Elev154N:=Elev154; Elev153N:=Elev153;
           Elev152N:=Elev152; Elev151N:=Elev151; Elev150N:= Elev150;
         end;

end;              {of the one or two loop~~~~~~~~~~~}
if (Selection = 3) or (Selection = 4) then begin
  GoodFront:=False; GoodBack:= False;
  if (AltTruS > 70) and (AltTruS < 120) and (AltTruN < 70)
     then GoodFront:=True;
  if (AltTruS > 90) and (AltTruS < 102) and (AltTruN > 110)
     then GoodFront:=True;
  if (AltTruS > 90) and (AltTruS < 102) and (AltTruN < 80)
     then GoodFront:=True;
  if (AltTruN > 70) and (AltTruN < 120) and (AltTruS < 70)
     then GoodBack:=True;
  if (AltTruN > 90) and (AltTruN < 102) and (AltTruS > 110)
     then GoodBack:=True;
  if (AltTruN > 90) and (AltTruN < 102) and (AltTruS < 80)
     then GoodBack:=True;
  if GoodFront and not GoodBack then begin
     ElevTru:=ElevTruS; AltTru:= AltTruS; Buffer.Lag23:= -100;
     Elev142:=Elev142S; Elev141:=Elev141S;ElevTry2:=ElevTry2S;
     ElevTry1:=ElevTry1S;ElevTry0:=ElevTry0S;AltTry2:=AltTry2S;
```

```
        AltTry1:=AltTry1S; AltTry0:=AltTry0S; Elev1510:=Elev1510S;
        Elev159:=Elev159S; Elev158:=Elev158S; Elev157:=Elev157S;
        Elev156:=Elev156S; Elev155:=Elev155S;  Elev154:=Elev154S;
        Elev153:=Elev153S; Elev152:=Elev152S;  Elev151:=Elev151S;
        Elev150:= Elev150s;
    end;
    if GoodBack and not GoodFront then begin
        ElevTru:=ElevTruN; AltTru:= AltTruN;  Buffer.Lag23:= +100;
        Elev142:=Elev142N; Elev141:=Elev141N; ElevTry2:=ElevTry2N;
        ElevTry1:=ElevTry1N; ElevTry0:=ElevTry0N; AltTry2:=AltTry2N;
        AltTry1:=AltTry1N; AltTry0:=AltTry0N; Elev1510:=Elev1510N;
        Elev159:=Elev159N; Elev158:=Elev158N; Elev157:=Elev157N;
        Elev156:=Elev156N; Elev155:=Elev155N;  Elev154:=Elev154N;
        Elev153:=Elev153N; Elev152:=Elev152N;  Elev151:=Elev151N;
        Elev150:= Elev150N;
    end;
    if GoodBack and GoodFront then begin
        ElevTru:=0.4; AltTru:= 0.4; end;
    if (not GoodBack) and (not GoodFront) then begin
        ElevTru:=0.4; AltTru:= 0.4; end;
end;
Difference:= rand(0,200/ElevTru);
if (ElevTru > 10) and (ElevTru < 60) then begin
    inc(Count142);
end;
inc(NumPhases);
inc(Phase14array[trunc(Phase14)]);
inc(Phase15array[trunc(Phase15)]);
if (ElevTru >= 5) and (ElevTru <= 180) then
inc(ElevXYarray[round(ElevTru)]);
with ObsvBuff do begin
  if (ElevTru>5.0) and (AltTru>5.0) then
    begin
    if Buffer.Lag23<=0 then Direction:=1;  {Looking South}
    if Buffer.Lag23>0 then Direction:=-1;{Looking North}
    if Direction=-1 then ElevTru:=180-ElevTru;
    write(Outtext2,Datestr(Year,Month,Day),' ');
    write(Outtext2,Timestr(Hour,Min,Sec),' ');
    write(Outtext2,ElevTru:8:1);
    write(Outtext2,AltTru:8:1);
    write(Outtext2,Buffer.Home.Range:8);
```

```
          write(Outtext2,Direction:8:0);
          writeln(Outtext2,Number:4,'    ');
        end;
      end;
      if (AltTru >= 70) and (AltTru <= 120) then
      inc(HomeAltarray[round(AltTru)]);
      999:
  end;
end;
close(InFile);
FindNext(SFile);
end else FindNext(SFile);
end;
close(Intext);
close(Outtext2);
{$I+}
if IOResult<>0 then begin end;
end;
begin    {No initialisation}
end.
```

A.2.3 Unit Match

The Borland Turbo Pascal Unit, `Match.pas`, is called from the control programme `System.pas` and combines meteor echo range, elevation angle and height information from Unit `Heights.pas` with the wind speed measurements from Unit `Speeds.pas`.

```
Unit Match;
interface
uses Dos,Crt,defns;
procedure CombineVelsandHeights(Dir:string7);
 implementation
{                                                    }
{                                                    }
{   This program is designed to combine data         }
{   from Orbit and Winds files.                      }
{                                                    }
{   A matching Number is required to be present      }
{   in both files.                                   }
```

```
const MaxPosRange:integer = 300{km};
      MinDuration=10{Sweeps containing echo info};

var   OutPutFile, AMORFile, DoRFFile : text;
      Date                           : string[7];
      Time                           : longint;
      Height, Angle, Range, Velocity : real;
      Uncertainty                    : real;
      AMORTime,DoRFTime              : longint;
      Bin,Duration,Direction         : integer;
      DorfEchoStart,DorfEchoend      : integer;
      OutDate                        : string[6];

{Returns true if meteor time and number match}
function MatchFound(AH,DH,AN,DN:integer):boolean;
begin
 if (AH = DH) and (AN = DN) then MatchFound := True
                            else MatchFound := False;
end;

{writes data to output file}
procedure writeOutput;
var
   RelVelUnc,RelCosAngleUnc,TotalRelUnc,VelocityHoriz    :real;
begin
     write(OutPutFile,OutDate,' ');
     write(OutPutFile,AMORTime:6,' ');
     write(OutPutFile,Height:6:1,' ');
     VelocityHoriz:=velocity/(cos(0.01745*Angle));{Pi/180 =0.01745}
     write(OutPutFile,VelocityHoriz:6:0,' ');
     write(OutPutFile,Angle:6:1,' ');
     if Velocity <> 0 then RelVelUnc:=Uncertainty/Velocity
        else RelVelUnc:= 0.0;
     if cos(0.01745*Angle) <> 0 then
        RelCosAngleUnc:=(sin(0.01745*Angle)*(0.01745*0.5))
             /(cos(0.01745*Angle)) {0.5deg is delta theta}
        else RelCosAngleUnc:=0.0;
     TotalRelUnc:=abs(RelVelUnc)+abs(RelCosAngleUnc);
     Uncertainty:=abs(TotalRelUnc*Velocity);
     write(OutPutFile,Uncertainty:6:1);
```

```
      write(OutPutFile,Direction:4);
      write(OutPutFile,Duration:5,' ');
      write(OutPutFile,Range:6:1);
      writeln(OutPutFile);
end;

{reads a meteor record from orbit data file}
procedure readAMORData(var AMORHour,AMORNumber:integer);
begin
      read(AMORFile,Date);
      readead(AMORFile,AMORTime);
      AMORHour := AMORTime div 10000;
      read(AMORFile,Angle);
      read(AMORFile,Height);
      read(AMORFile,Range);
      read(AMORFile,Direction);
      readln(AMORFile,AMORNumber);
end;

{reads a meteor record from winds data file}
procedure readDoRFData(var DoRFHour,DoRFNumber:integer);
var code:integer;
    Date:string[7];
begin
      read(DoRFFile,Date);
      read(DoRFFile,DoRFTime);
      DoRFHour:=DoRFTime div 10000;
      read(DoRFFile,Velocity);
      read(DoRFFile,Uncertainty);
      read(DoRFFile,Bin);
      read(DoRFFile,DoRFNumber);
      read(DoRFFile,Duration);
      read(DoRFFile,DorfEchoStart);
      readln(DorFFile,DorfEchoend);
end;

{Returns true if meteor duration is long enough}
function LongEnough(Duration:integer):boolean;
begin
      if Duration>=MinDuration then LongEnough:=True
                               else LongEnough:=False;
```

```
end;
procedure CombineDataFiles;
var AH, AN, DH, DN : integer;
    Option         : integer;

begin
     readAMORData(AH,AN);
     readDoRFData(DH,DN);
     repeat
      if MatchFound (AH,DH,AN,DN) then begin
        if (Range < MaxPosRange) and (LongEnough(Duration)) then
              writeOutput;
        readDoRFData(DH,DN);
      end
      else begin
        if AH > DH then option:=1;
        if DH > AH then option:=2;
        if DH = AH then option:=3;
        case option of
          1: readDoRFData(DH,DN);
          2: readAMORData(AH,AN);
          3: begin
             if (AMORTime >= DoRFTime) then
                    readDoRFData(DH,DN)
               else if (DoRFTime > AMORTime) then
                    readAMORData(AH,AN);
             end;
        end;
      end;
     until Eof(AMORFile) or Eof(DoRFFile);
end;

{Initialises files}
procedure Setup;
begin
     assign(OutPutFile,'E:\Analysed\'+date+'\Graph.dat');
     assign(AMORFile,'E:\Analysed\'+date+'\Amor.dat');
     assign(DoRFFile,'E:\Analysed\'+date+'\PhVels.dat');
     reset(DoRFFile);
     reset(AMORFile);
     rewrite(OutPutFile);
```

```
end;

{closes open files}
procedure ConClude;
begin
      close(OutPutFile);
      close(DoRFFile);
      erase(DoRFFile);
      close(AMORFile);
      erase(AMORFile);
end;

{Routine called from outside of Unit}
procedure CombineVelsandHeights(Dir:string7);
begin
      date:=dir;
      OutDate:=Dir;
      writeln;
      writeln('Combining File for ',date);
      writeln;
      Setup;
      CombineDataFiles;
      ConClude;
end;
begin {No Initialisation}
end.
```

A.2.4 Unit Sort

The Borland Turbo pascal Unit, `Sort.pas`, is called from the control programme `System.pas` and sorts the available data into chronological order.

```
Unit Sort;
interface

uses Dos,Defns;
const NoOfDates=365;
type Dates=string[7];
var
   Datesarray     : array[1..NoOfDates] of Dates;
```

```
    DirInfo          : SearchRec;
    i                : integer;
    outfile          : text;
procedure SortDates;
Implementation
procedure GetDates;
begin
       FindFirst(Drive+':\DorfData\*.*', Directory, DirInfo);
       i:=0;
       while (DosError = 0) and (i<=NoOfDates) do
       begin
              if (dirinfo.name<>'.')and (dirinfo.name<>'..')then
              begin
                inc(i);
                Datesarray[i]:=DirInfo.name;
              end;
              FindNext(DirInfo);

       end;
end;
procedure DateSort;
var Month                    : string[3];
    ValidFile                : boolean;
    MonthVal                 : longint;
    j,k                      : integer;
    DateStr,temp             : string[7];
    code                     : integer;
    year,day,MinIndex        : longint;
    MinDateValue,DateValue: longint;


begin
rewrite(outfile);
    for j:=1 to i do begin
     MinDateValue:=999999;
     for k:= j to i do begin
        ValidFile:=True;
        datestr:=Datesarray[k];
        Val(copy(datestr,1,2),Year,code);
        Val(copy(datestr,5,2),Day,code);
        Val(copy(DateStr,3,2),MonthVal,code);
```

```
            datevalue:=year*10000+monthval*100+day;
            if DateValue<MinDateValue then begin
                                                MinDateValue:= DateValue;
                                                Minindex:=k;
                                                end;
        end;
        temp:=Datesarray[j];
        Datesarray[j]:=Datesarray[MinIndex];
        Datesarray[MinIndex]:=temp;
        writeln(outfile,datesarray[j]);

      end;
      close(OutFile);
end;
procedure SortDates;
begin
        assign(outfile,'c:\dop128\dates.dat');
        GetDates;
        DateSort;
end;
begin {No Initialisation}
end.
```

A.2.5 Unit Defns

The Borland Turbo Pascal Unit, `Defns.pas`, is called by the control programme `System.pas` to define constants and variable types which are shared between Units.

```
Unit Defns;
interface
const
        Drive='F'; {======> Must be in CAPITALS !!!!!!!! }
        DriveNo=Ord(Drive)-64;
        MaxDataPoints=800;
        NDataP=128;

type
      realarray       = array[0..MaxDataPoints] of real;
      Intarray        = array[0..MaxDataPoints] of integer;
      string7         = string[7];
      Parray          = array[1..MaxDataPoints] of real;
```

```
    Frarray          = array[1..MaxDataPoints] of integer;
    realarrayNData = array[1..NDataP] of real;
implementation
begin
end.
```

A.3 Program GroundRangeAnalysis

The C programme, `GroundRangeAnalysis.c`, reads the files generated by `System.pas` and produces hourly averaged wind speeds from individual measurements. Acceptance layer thickness, number of ground range bins and height are required as input.

```
/*

Program by Steven Marsh July 1998.

Generates a 3D matrix of velocity values in heights from LoHeight
to HiHeight km.

*/

#include <stdlib.h>
#include <stdio.h>
#include <string.h>
#include <math.h>
#include ''nrutil.h''
#define TRUE 1
#define FALSE 0
#define Pi 4*atan(1)
#define MaxRange 300    //kms
#define MinRange 40     //kms
#define AcceptanceLevel 0.5  // ratio data filled hrs to hrs
#define significance_level 0.05 //for Fisher's test
#define lambda 0.6      // for Siegel's test
#define window_width 4  //days
#define shortest_period window_width*24
#define Dest_Drive 'H'
#define IGREG (15+31L*(10+12L*1582))
// begin Prototypes
// Obtains values from the user
```

```
void GetInputParameters(long *ThisDate, long *endDate,
                        short *NoOfDays,short *NoOfGroundBins,
                        short *LoHeight, short *HiHeight);

// All the real work is done here
short GroundRange (float height,float elevangle,short direction,
                   short NoOfGroundBins);

// Finds the next data with data available
long GetNextDate(long ThisDate);

// Calculates Julian Day
long julday(short mm, short id, short iyyy);

// Applies a cosine tapered window to data
void win_cost(float *tt, float *yy, short counter, float alpha);

// end Prototypes
void main()
{
char Filestring[80]='''', LsftFilestring[80]='''';
long ThisDate, endDate;
FILE *InputFile, *OutputFile;
long date, time;
float height, elevangle, velunc,Range,junk2;
short noofpulses,direction, velocity,HalfLayerThickness;
float ***SumVelsTensor, ***AvVelsTensor,***SumSqrsVelsTensor,
      ***varianceTensor,***AvHoursTensor,***varHoursTensor,
      ***SumHoursTensor,***SumSqrsHoursTensor;
short ***CountVelsTensor;
short SameHour=1,FirstTime=1,FirstDate;
short **IndexMatrix,**DataWasFoundMatrix,index=0;
short DayCount=0,Hour=0,LastHour=0,GroundRangeInd=0,HtIndex,junk1;
short i=0,j=0,k=0,NoOfDays,NoOfGroundBins,LoHeight,HiHeight;
short windowwidth,shortestperiod, temptime, minutes, seconds;
short Hindex;
long randomseed=-3;
short offs = 0;


 printf(''Welcome to GroundRangeAnalysis created Feb 1999\n
```

```
        for generating Metxx.Dat files. \nHourly averaging.\n\n'');
printf(''Enter Half Layer Thickness (4 = 9 km)\n'');
scanf(''%hd'',&HalfLayerThickness);
printf(''Layer Thickness = %hd km\n'',2*HalfLayerThickness+1);
GetInputParameters(&ThisDate, &endDate, &NoOfDays, &NoOfGroundBins,
                   &LoHeight,&HiHeight);
printf(''\nNo of days = %hd\n'',NoOfDays);
AvHoursTensor=
  f3tensor(1,2*NoOfGroundBins,LoHeight,HiHeight,0,NoOfDays*24);
SumHoursTensor=
  f3tensor(1,2*NoOfGroundBins,LoHeight,HiHeight,0,NoOfDays*24);
varHoursTensor=
  f3tensor(1,2*NoOfGroundBins,LoHeight,HiHeight,0,NoOfDays*24);
SumVelsTensor=
  f3tensor(1,2*NoOfGroundBins,LoHeight,HiHeight,0,NoOfDays*24);
SumSqrsVelsTensor=
  f3tensor(1,2*NoOfGroundBins,LoHeight,HiHeight,0,NoOfDays*24);
SumSqrsHoursTensor=
  f3tensor(1,2*NoOfGroundBins,LoHeight,HiHeight,0,NoOfDays*24);
varianceTensor=
  f3tensor(1,2*NoOfGroundBins,LoHeight,HiHeight,0,NoOfDays*24);
CountVelsTensor=
  i3tensor(1,2*NoOfGroundBins,LoHeight,HiHeight,0,NoOfDays*24);
AvVelsTensor=
  f3tensor(1,2*NoOfGroundBins,LoHeight,HiHeight,0,NoOfDays*24);
IndexMatrix=imatrix(1,2*NoOfGroundBins,LoHeight,HiHeight);
DataWasFoundMatrix=imatrix(1,2*NoOfGroundBins,LoHeight,HiHeight);
FirstDate = TRUE;

//initialise variables

for(i=1;i<=2*NoOfGroundBins;i++)
{
 for(j=LoHeight;j<=HiHeight;j++)
 {
  IndexMatrix[i][j]=1;
  DataWasFoundMatrix[i][j]=FALSE;

  for(k=0;k<=NoOfDays*24;k++)
  {
   SumVelsTensor[i][j][k]=(float)0;
```

```
    CountVelsTensor[i][j][k]=(short)0;
    SumSqrsVelsTensor[i][j][k]=(short)0;
    AvVelsTensor[i][j][k]=(float)999;
    AvHoursTensor[i][j][k]=(short)0;
    SumHoursTensor[i][j][k]=(short)0;
    SumSqrsHoursTensor[i][j][k]=(short)0;

   }
  }
 }
// end of initialise
while(ThisDate<endDate) //still analyses endDate data
{
    do
    {
     if (!FirstDate)
     {
      ThisDate=GetNextDate(ThisDate);
      DayCount++;
     }
     Filestring[0]='\0';
   sprintf(Filestring,''E:\\Analysed\\%lu\\Graph.Dat'',ThisDate);
   printf(''Opening File: %s\n\n'',Filestring);
   FirstDate=FALSE;
  }
  while ((InputFile=fopen(Filestring,''r''))==NULL);
FirstTime=TRUE;
SameHour=TRUE;
  while(!feof(InputFile))
  {
   fscanf(InputFile,''%lu%lu%f%hd%f%f%hd%hd%f'',&date,&time,&height,
        &velocity,&elevangle,&velunc,&direction,&noofpulses,&Range);
   Hour=(short)((DayCount*24)+(time/10000));
   temptime = time % 10000;
   seconds = temptime % 100;
   minutes = (temptime - seconds)/100;
   if (FirstTime)
   {
    LastHour=Hour;
    FirstTime=FALSE;
   }
```

```
if (Hour!=LastHour)
{
 for(j=LoHeight;j<=HiHeight;j++)
 {
  for (i=1;i<=2*NoOfGroundBins;i++)
  {
   if (DataWasFoundMatrix[i][j])
    {
     IndexMatrix[i][j]++;
     DataWasFoundMatrix[i][j]=FALSE;
    }
  }
 }//for(j=LoHeight;j<=HiHeight;j++)
 LastHour=Hour;
}//if (Hour!=LastHour)
for (offs=-HalfLayerThickness;offs<=HalfLayerThickness; offs++)
{
 HtIndex=(short) floor(height+offs+0.5);
 if((HtIndex>=LoHeight) && (HtIndex<=HiHeight))
 {
  GroundRangeInd=
      GroundRange(height,elevangle,direction,NoOfGroundBins);
  CountVelsTensor[GroundRangeInd][HtIndex][IndexMatrix
      [GroundRangeInd][HtIndex]]++;
  junk1=CountVelsTensor[GroundRangeInd][HtIndex][IndexMatrix
      [GroundRangeInd][HtIndex]];
  SumHoursTensor[GroundRangeInd][HtIndex][IndexMatrix[GroundRangeInd]
      [HtIndex]]+=((float)minutes/60)+((float)seconds/3600);
  junk2=SumHoursTensor[GroundRangeInd][HtIndex][IndexMatrix
      [GroundRangeInd][HtIndex]];
  SumSqrsHoursTensor[GroundRangeInd][HtIndex][IndexMatrix
  [GroundRangeInd][HtIndex]]+=(((float)minutes/60)+
      ((float)seconds/3600))*((float)minutes/60)+
      ((float)seconds/3600));
  AvHoursTensor[GroundRangeInd][HtIndex][IndexMatrix
      [GroundRangeInd][HtIndex]]= Hour +
      SumHoursTensor[GroundRangeInd][HtIndex][IndexMatrix
      [GroundRangeInd][HtIndex]]/CountVelsTensor[GroundRangeInd]
      [HtIndex][IndexMatrix [GroundRangeInd][HtIndex]];
  if (junk1>1){
   varHoursTensor[GroundRangeInd][HtIndex][IndexMatrix
```

```
        [GroundRangeInd][HtIndex]]=(SumSqrsHoursTensor
        [GroundRangeInd][HtIndex][IndexMatrix[GroundRangeInd]
        [HtIndex]]/CountVelsTensor[GroundRangeInd][HtIndex]
        [IndexMatrix [GroundRangeInd][HtIndex]])-pow(
        (SumHoursTensor[GroundRangeInd][HtIndex][IndexMatrix
        [GroundRangeInd][HtIndex]]/CountVelsTensor[GroundRangeInd]
        [HtIndex][IndexMatrix [GroundRangeInd][HtIndex]]),2);
      }
      else varHoursTensor[GroundRangeInd][HtIndex][IndexMatrix
          [GroundRangeInd][HtIndex]]=0;
      SumVelsTensor[GroundRangeInd][HtIndex][IndexMatrix
          [GroundRangeInd][HtIndex]]+=(float)velocity;
      SumSqrsVelsTensor[GroundRangeInd][HtIndex][IndexMatrix
          [GroundRangeInd][HtIndex]]+=((float)velocity*
          (float)velocity);
      AvVelsTensor[GroundRangeInd][HtIndex][IndexMatrix
            [GroundRangeInd][HtIndex]]=SumVelsTensor[GroundRangeInd]
            [HtIndex][IndexMatrix[GroundRangeInd][HtIndex]]/
            CountVelsTensor[GroundRangeInd][HtIndex][IndexMatrix
            [GroundRangeInd][HtIndex]];
      if (junk1>1){
       varianceTensor[GroundRangeInd][HtIndex][IndexMatrix
            [GroundRangeInd][HtIndex]]=(SumSqrsVelsTensor
            [GroundRangeInd][HtIndex][IndexMatrix[GroundRangeInd]
            [HtIndex]]/CountVelsTensor[GroundRangeInd][HtIndex]
            [IndexMatrix [GroundRangeInd][HtIndex]])-pow(
            (SumVelsTensor[GroundRangeInd][HtIndex][IndexMatrix
            [GroundRangeInd][HtIndex]]/CountVelsTensor[GroundRangeInd]
            [HtIndex][IndexMatrix [GroundRangeInd][HtIndex]]),2);
      }
      else varianceTensor[GroundRangeInd][HtIndex][IndexMatrix
            [GroundRangeInd][HtIndex]]=0;
      DataWasFoundMatrix [GroundRangeInd][HtIndex]=TRUE;
    }
    }// for (offs=-HalfLayerThickness;offs<=HalfLayerThickness;
     // offs++)
   }//while(!feof(InputFile))
   fclose(InputFile);
  }//while(ThisDate<=endDate)
  // beginning of output section
printf(''LoHeight=%hd\tHiHeight = %hd\n'',LoHeight,HiHeight);
```

```
shortestperiod=shortest_period;
for(Hindex=LoHeight;Hindex<=HiHeight;Hindex++)
{
height=(float)Hindex;
// while (1==1)
for (GroundRangeInd=1;GroundRangeInd<=2*NoOfGroundBins;
      GroundRangeInd++)
 {
  windowwidth=window_width;
  printf(''Analysing Height = %5.2f\tGround Range Bin = %hd\n'',
           height,GroundRangeInd);
  if(height==999) exit(1);
  Filestring[0]='\0';
  sprintf(Filestring,''%c:\\DataDisk\\Met%hd%hd.Dat'', Dest_Drive,
                                    (short)height,GroundRangeInd);
  if ((OutputFile = fopen(Filestring,''w''))==NULL)
  {
   printf(''OutputFile error\n'');
  }
  for(i=1;i<IndexMatrix [GroundRangeInd][(short)height];i++)
  {
   if (fabs(AvVelsTensor[GroundRangeInd][(short)height][i]) < 300)
   {
    fprintf(OutputFile,''%5.2f\t%5.2f\t%5.2f\t%5.2f\t%2hd\n'',
      AvHoursTensor[GroundRangeInd][(short)height][i],
      AvVelsTensor[GroundRangeInd][(short)height][i],
      sqrt(varianceTensor[GroundRangeInd][(short)height][i]),
      sqrt(varHoursTensor[GroundRangeInd][(short)height][i]),
      CountVelsTensor[GroundRangeInd][(short)height][i]);
    }
  }
  fclose(OutputFile);
 } //for GroundRangeInd = 1 to 2*NoOfGroundRanges
 }  //for height=LoHeight to HiHeight
 //  end of output section
 printf(''Done.\n'');
} //main
void GetInputParameters(long *ThisDate, long *endDate, short
                        *NoOfDays,  short *NoOfGroundBins,
                         short *LoHeight, short *HiHeight )
{
```

```
 char DThisDate[10],DendDate[10];
 short TYear,TMonth,TDay,EYear,EMonth,EDay;
// All initial reading from the keyboard is done here!
 printf(''Enter StartDate: '');
 scanf(''%lu'',ThisDate);
 printf(''\n\nEnter endDate: '');
 scanf(''%lu'',endDate);
 printf(''\n\nEnter Lowest Height of Interest: '');
 scanf(''%hd'',LoHeight);
 printf(''\n\nEnter Geatest Height of Interest: '');
 scanf(''%hd'',HiHeight);
 printf(''\n\nEnter No of ground bins per direction: '');
 scanf(''%hd'',NoOfGroundBins);
 sprintf(DThisDate,''%lu'',*ThisDate);
 sscanf(DThisDate ,''%2hd%2hd%2hd'',&TYear,&TMonth,&TDay);
 if (TYear<70) TYear+=2000;
  else TYear+=1900;
 sprintf(DendDate,''%lu'',*endDate);
 sscanf(DendDate ,''%2hd%2hd%2hd'',&EYear,&EMonth,&EDay);
 if (EYear<70) EYear+=2000;
  else EYear+=1900;
 *NoOfDays=(short)julday(EMonth,EDay,EYear)-
                    julday(TMonth,TDay,TYear);
}
short GroundRange (float height,float elevangle,short direction,
                                                  short NoOfBins)
{
 float GrndRange;
 short GroundRangeInd;
//  An Approximation that does NOT consider earth's curvature
 GrndRange=fabs(height/((sin(elevangle*(Pi/180)))/
                   (cos(elevangle*(Pi/180)))));
 switch(direction)
 {
  case 1:  //SouthWard
  {
   if (GrndRange>MaxRange) GroundRangeInd=1;
   else if (GrndRange<MinRange) GroundRangeInd=NoOfBins;
   else
       {
    GroundRangeInd=floor(NoOfBins*(GrndRange-MinRange)/
```

```
                        (MaxRange-MinRange));
    GroundRangeInd=NoOfBins-GroundRangeInd;
    }
  }
  break;
  case -1:    //NorthWard
  {
   if (GrndRange>MaxRange) GroundRangeInd=2*NoOfBins;
   else if (GrndRange<MinRange) GroundRangeInd=NoOfBins+1;
   else
    {
     GroundRangeInd=floor(NoOfBins*(GrndRange-MinRange)/
                                    (MaxRange-MinRange))+1;
      GroundRangeInd+=NoOfBins;
    }// else
  }
 }
    if ((GroundRangeInd <1) || (GroundRangeInd > 2* NoOfBins))
          printf("\n WARNING Ground Range Indicater -> %d\n",
                  GroundRangeInd);
 return (GroundRangeInd);
}
long GetNextDate(long ThisDate)
{ short Year,Month,Day;
 short MaxDays[13]={0,31,28,31,30,31,30,31,31,30,31,30,31};
 char Dummy[10];
 long NewDay;
 sprintf(Dummy,"%lu",ThisDate);
 sscanf(Dummy ,"%2hd%2hd%2hd",&Year,&Month,&Day);
 if (((Year % 4==0) && (Year % 100 != 0))|| (Year % 400 ==0))
                  MaxDays[2]=29;//leap year test
 Day++;
 if(Day>MaxDays[Month])
 {
  Day=1;
  Month++;
  if (Month>12)
  {
   Year++;
   Month=1;
  }
```

```
 }
 NewDay=Year*10000+Month*100+Day;
 return (NewDay);
}
long julday(short mm, short id, short iyyy)
{
 void nrerror(char error_text[]);
 long jul;
 short ja,jy=iyyy,jm;
 if (jy == 0) nrerror(''julday: there is no year zero.'');
 if (jy < 0) ++jy;
 if (mm > 2) {
  jm=mm+1;
 } else {
  --jy;
  jm=mm+13;
 }
 jul = (long) (floor(365.25*jy)+floor(30.6001*jm)+id+1720995);
 if (id+31L*(mm+12L*iyyy) >= IGREG) {
  ja=(int)(0.01*jy);
  jul += 2-ja+(int) (0.25*ja);
 }
 return jul;
}
/* (C) Copr. 1986-92 Numerical Recipes Software 51#. */
#undef IGREG
```

A.4 Program LSFT2

The C programme `LSFT2.c` performs a Lomb-Scargle Fourier Transform analysis only at specified frequencies. This programme is designed to be compiled by, and then accessed from within, `MatLAB`.

```
#include <math.h>
#include ''mex.h''
#include ''matrix.h''

#define Pi 3.1415926
#define tol1 1e-4
#define tol2 1e-8
```

```
#define E 2.7182818
#define const1 1/sqrt(2)

//input arguments

#define tt_in prhs[0]
#define yy_in prhs[1]
#define pers_in prhs[2]
#define nn_in prhs[3]
#define np_in prhs[4]

//output arguements

#define ftrp_pr plhs[0]
#define ftip_pr plhs[1]

typedef struct Complex
        {
         double rp, ip;
        } Complex;

Complex work, tmp;

// begin Prototypes

// Main procedure
void lsft2(double tsamp[],     // sampled times
     double xx[],      // sampling data
     double freqs[],    // PERIODS to be analysed
     short nn,      // length of time and data vectors
     short nfreq,     // length of freqs vector
     short si,      // sign of transform +1 for time to freq domain
     double ftrp[],    // real part of Fourier transform
     double ftip[]);    // imag part of Fourier transform

// Complex number multiplication
Complex cmult (Complex *a, Complex *b);

// Converts value to a complex exponential
Complex cexp(Complex *a);
void win_cost(double *tt, double *yy, short counter, double alpha);
```

```
// end Prototypes

void mexfunction(
     int nlhs,mxarray *plhs[],
     int nrhs,const mxarray *prhs[]
    )
{
 double *ftrp_out,*ftip_out,*tt,*yy,*pers,alpha;
 int nn,np,si,i;


 si=1;
 alpha=0.5;

 if(nrhs!=5 || nlhs!=2)
  mexErrMsgTxt(''Input should be as follows: [a b] =
                          lsft2dll(tt,yy,pers,nn,2*np)'');

 nn=(int)mxGetScalar(nn_in);
 np=(int)mxGetScalar(np_in);
 ftrp_pr=mxCreateDoubleMatrix(np,1,mxREAL);
 ftip_pr=mxCreateDoubleMatrix(np,1,mxREAL);
 ftrp_out=(double *)mxGetPr(ftrp_pr);
 ftip_out=(double *)mxGetPr(ftip_pr);
 tt=   (double *)mxGetPr(tt_in);
 yy=   (double *)mxGetPr(yy_in);
 pers=  (double *)mxGetPr(pers_in);
 win_cost(tt,yy,nn,alpha);    // puts a cosine
                              // through data alpha=0.5
 lsft2(tt,yy,pers,nn,np,si,ftrp_out,ftip_out); //calling c function
}

void lsft2(double tsamp[],     // sampled times
    double xx[],       // sampling data
    double freqs[],    // PERIODS to be analysed
    short nn,      // length of time and data vectors
    short nfreq,     // length of freqs vector
    short si,      // sign of transform    +1 for time to freq domain
    double ftrp[],    // real part of Fourier transform
    double ftip[])    // imag part of Fourier transform
```

```
{  // declarations

 double wuse, const2, sumt, sumx,  tau0, wdel, wrun, ttt, csum,
     ssum, sumtc,sumts, arg, tcos, tsin, watan, wtau, wtnew,
     sumr, sumi, scos2, ssin2, cross,tim, xd, phase, ftid, ftrd,
     nstop, avgdt, tp, fzero, tzero, wz, fny, lfreq, sqrtnn,
     ampfactor;
 short fnn, i, istop, cnt, iput;

 // end of declarations

 avgdt=(tsamp[nn-1]-tsamp[0])/(nn-1); // average timestep
 tp=avgdt*nn;          // period of a segment
 fzero=1/tp;           // fundamental frequency
 tzero=0.0;           // set fiducial origin of time to zero
 wz = 2.0*Pi*fzero;         // Omega
 fny= 1.0/(2.0*avgdt);       // average nyquist frequency
 lfreq= 2*nfreq;
 wuse = wz;
    fnn = (double)nn;
    const2 = si*const1;
    sumt = 0;
    sumx = 0;
    sqrtnn = sqrt(nn);
    if(si==1) ampfactor = 2/sqrtnn;
     else ampfactor = sqrtnn/2;
    for(i=0;i<nn;i++)
    {
      sumt += tsamp[i];
      sumx += xx[i];
    }
    // initialise for zero frequency
    tau0 = sumt/fnn;
    ftrp[0]=sumx/fnn;
    ftip[0]=0;

    //start frequency loop
    istop = nfreq-1;
    wdel = wuse;
```

```
cnt = 1;
wrun = 2*Pi/freqs[cnt];
do
{
  csum=0;
  ssum=0;
  sumtc=0;
  sumts=0;
  for(i=0;i<nn;i++)
   {
      ttt = tsamp[i];
        arg = 2*wrun*ttt;
        tcos = cos(arg);
        tsin = sin(arg);
        csum += tcos;
        ssum += tsin;
        sumtc = sumtc + ttt * tcos;
        sumts = sumts + ttt * tsin;
    } // for(i=0;i<nn;i++)
  if ((fabs(ssum)>tol1) || (fabs(csum)>tol1))
        watan = atan2(ssum,csum);
   else watan = atan2(-sumtc,sumts);
  wtau = 0.5*watan;
  wtnew = wtau;
  sumr = 0;
  sumi = 0;
  scos2 = 0;
  ssin2 = 0;
  cross = 0;
  for (i=0;i<nn;i++)
  {
   tim = tsamp[i];
     arg = wrun * tim -wtnew;
     tcos = cos(arg);
     tsin = sin(arg);
     cross = cross + tim *tcos * tsin;
     scos2 = scos2 + tcos * tcos;
     ssin2 = ssin2 + tsin * tsin;
     xd = xx[i];
     sumr = sumr + xd*tcos;
     sumi = sumi + xd*tsin;
```

```
        } // for (i=0;i<nn;i++)
        ftrd = const1 * sumr/sqrt(scos2);
        if(ssin2<=tol1)
        {
          ftid = const2 * sumx/sqrt(fnn);
           if(fabs(cross) > tol2)
           {
            ftid = 0;
           }

        }   // if(ssin2<=tol1)

        else     // (ssin2 > tol1)
        {
         ftid = const2 * sumi/sqrt(ssin2);
        }

        phase = wtnew - wrun * tzero;
        work.rp = ftrd;
        work.ip = ftid;
        tmp.rp = 0;
        tmp.ip = phase;
        tmp = cexp(&tmp);
        work = cmult(&work,&tmp);
        ftrp[cnt] =  work.rp * ampfactor;
        ftip[cnt] = work.ip * ampfactor;
        cnt++;
        wrun = 2*Pi/freqs[cnt];
    }while (cnt <= istop);
}


Complex cmult (Complex *a, Complex *b)
{
 Complex c;
 c.rp = a->rp*b->rp-a->ip*b->ip;
 c.ip = a->ip*b->rp+a->rp*b->ip;
 return c;
}

Complex cexp(Complex *a)
```

```
{
  Complex c;
   c.rp=pow(E,a->rp) * cos(a->ip);
   c.ip=pow(E,a->rp) * sin(a->ip);
   return c;
}

void win_cost(double *tt, double *yy, short counter, double alpha)
{
 double *weights,sumweights,cycle;
 short middle,nlow,nhigh,i;
 if(counter>0)
 {
 middle = ceil((tt[counter-1]-tt[0])/2);
 if((weights = (double*)malloc(2*middle*sizeof(double)+10))==NULL)
    {
     printf(''counter = %hd  middle = %hd malloc error\n'',
              counter,middle);
     exit(1);
    }
 nhigh = middle;
 nlow = ceil(alpha*middle);
 cycle = Pi/(nhigh - nlow);
 sumweights = 0;
// calculate end taper
 for(i=middle+nlow;i<=tt[counter-1]-tt[0];i++)
 {
  weights[i]=0.5+0.5*cos((i-nlow-middle)*cycle);
  weights[(short)tt[counter-1]-(short)tt[0]-i] = weights[i];
  sumweights += weights[i];
 }
 for(i=middle-nlow+1;i<=middle+nlow-1;i++)
 {
  weights[i]=1;
 }
 sumweights *=2;
 sumweights +=(2*nlow);
 if(sumweights != 0.0)
  for (i=0;i<counter;i++)
  {
   yy[i]*=(weights[(short)(tt[i]-tt[0])]*(tt[counter-1]-tt[0])/
```

```
            sumweights);

  }
 free(weights);
 }
}
```

A.4.1 MatLAB LSFT2

The `MatLAB` m-file `LSFT2.m` performs data checking on data to be processed with Lomb-Scargle Fourier analysis via `LSFT2.c`.

```
% Computes the Lomb–Scargle Fourier Transform
% for data tt and yy at specific periods pp.
% yy are data values and tt times of yy values.
%
% A cosine taper window is also applied to the
% data with alpha=0.5
%
% Care needs to be taken as per=1 should NOT
% be included as it is the nyquist period.
% also the dc component MUST always be included
% as the first period.
% Written by Steven Marsh Sept 1998
%
% Usage [transform]=lsft2(tt,yy,pp)

function [transform]=lsft2(tt,yy,pers)
if (length(tt)==length(yy))
   finite_indices=isfinite(yy);
   if length(yy(finite_indices))>1
      [ftrp ftip]=lsft2dll(tt(finite_indices),yy(finite_indices),
      pers,length(tt(finite_indices)),length(pers));
      transform=ftrp+i*ftip;
   else
      transform(1:length(pers))=NaN;
      transform=transform';
   end
else error('Input vectors must be the same length.');
end;
```

Appendix B

Tabulated Data from Ground Range Bin Comparisons

B.1 Correlation Coefficients

The following graphs detail correlation coefficients from ground range bin comparisons for all data and data grouped by season.

GRB	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
1	0.85	0.84	0.83	0.81	0.77	0.78	0.70	-0.18	0.45	0.63	0.62	0.69	0.67	0.68	0.50
2		0.90	0.89	0.87	0.85	0.80	0.72	-0.11	0.37	0.72	0.66	0.74	0.72	0.70	0.60
3			0.91	0.90	0.88	0.81	0.77	-0.05	0.30	0.73	0.67	0.75	0.72	0.69	0.58
4				0.91	0.90	0.85	0.79	-0.01	0.39	0.74	0.69	0.76	0.73	0.69	0.60
5					0.92	0.87	0.81	-0.05	0.36	0.73	0.67	0.75	0.72	0.67	0.54
6						0.88	0.81	-0.04	0.28	0.73	0.67	0.74	0.71	0.67	0.58
7							0.80	-0.25	0.17	0.73	0.71	0.74	0.67	0.64	0.53
8								-0.16	0.17	0.67	0.65	0.68	0.69	0.65	0.49
9									0.73	0.18	0.17	0.16	0.03	0.09	NaN
10										0.67	0.55	0.47	0.45	0.40	NaN
11											0.84	0.86	0.85	0.81	0.68
12												0.85	0.83	0.81	0.70
13													0.88	0.87	0.77
14														0.87	0.79
15															0.82

Table B.1: GRB comparison correlation coefficients - All Data.

GRB	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
1	0.86	0.85	0.82	0.80	0.77	0.78	0.70	-0.23	0.50	0.66	0.68	0.74	0.72	0.73	0.54
2		0.89	0.89	0.86	0.83	0.82	0.73	-0.04	0.42	0.74	0.69	0.77	0.74	0.74	0.63
3			0.91	0.90	0.87	0.84	0.75	0.07	0.36	0.73	0.70	0.78	0.74	0.73	0.61
4				0.92	0.90	0.88	0.78	0.12	0.38	0.76	0.71	0.79	0.76	0.72	0.63
5					0.92	0.88	0.79	0.08	0.38	0.73	0.70	0.76	0.75	0.69	0.56
6						0.89	0.77	0.03	0.17	0.74	0.69	0.76	0.74	0.69	0.59
7							0.82	-0.15	NaN	0.76	0.73	0.76	0.71	0.64	0.56
8								0.24	NaN	0.62	0.66	0.66	0.69	0.61	0.49
9									NaN	0.22	0.24	0.26	0.12	0.20	NaN
10										0.71	0.59	0.36	0.46	0.30	NaN
11											0.85	0.87	0.86	0.81	0.64
12												0.87	0.85	0.82	0.66
13													0.89	0.87	0.75
14														0.88	0.78
15															0.80

Table B.2: GRB comparison correlation coefficients - Summer.

GRB	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
1	0.83	0.83	0.82	0.80	0.75	0.71	0.63	-0.08	0.44	0.56	0.52	0.59	0.55	0.50	0.21
2		0.89	0.89	0.87	0.85	0.77	0.68	-0.16	0.33	0.69	0.61	0.69	0.66	0.62	0.41
3			0.91	0.90	0.87	0.78	0.75	-0.19	0.24	0.71	0.64	0.71	0.65	0.62	0.42
4				0.91	0.89	0.83	0.78	-0.12	0.39	0.72	0.65	0.71	0.66	0.62	0.45
5					0.91	0.85	0.81	-0.17	0.32	0.72	0.64	0.72	0.66	0.60	0.42
6						0.87	0.82	-0.10	0.36	0.71	0.64	0.71	0.65	0.62	0.45
7							0.78	-0.35	0.21	0.67	0.66	0.68	0.60	0.59	0.41
8								-0.66	NaN	0.66	0.55	0.63	0.61	0.60	NaN
9									0.30	0.14	0.08	0.03	-0.13	-0.04	NaN
10										0.62	0.51	0.58	0.38	0.45	NaN
11											0.81	0.84	0.83	0.79	0.72
12												0.82	0.78	0.77	0.75
13													0.85	0.85	0.80
14														0.84	0.74
15															0.85

Table B.3: GRB comparison correlation coefficients - Autumn.

GRB	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
1	0.83	0.82	0.81	0.79	0.74	0.76	0.66	NaN	NaN	0.55	0.30	0.45	0.48	0.50	NaN
2		0.88	0.88	0.86	0.84	0.81	0.77	NaN	NaN	0.62	0.41	0.56	0.62	0.43	NaN
3			0.90	0.89	0.87	0.81	0.81	NaN	NaN	0.62	0.43	0.56	0.58	0.37	NaN
4				0.91	0.89	0.86	0.81	NaN	NaN	0.64	0.42	0.58	0.54	0.38	NaN
5					0.90	0.87	0.82	NaN	NaN	0.63	0.41	0.59	0.58	0.37	NaN
6						0.88	0.82	NaN	NaN	0.65	0.40	0.60	0.55	0.38	NaN
7							0.72	NaN	NaN	0.50	0.38	0.36	0.42	0.08	NaN
8								NaN	NaN	0.67	0.38	0.55	0.33	0.02	NaN
9									NaN	NaN	NaN	NaN	NaN	NaN	NaN
10										NaN	NaN	NaN	NaN	NaN	NaN
11											0.66	0.81	0.70	0.67	NaN
12												0.72	0.79	0.68	NaN
13													0.82	0.84	NaN
14														0.74	NaN
15															NaN

Table B.4: GRB comparison correlation coefficients - Winter.

GRB	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
1	0.66	0.70	0.59	0.61	0.45	0.37	0.39	NaN	NaN	NaN	0.10	NaN	NaN	NaN	NaN
2		0.81	0.79	0.75	0.69	0.57	0.53	-0.38	NaN	0.55	0.55	0.50	0.71	0.65	NaN
3			0.82	0.82	0.75	0.67	0.69	-0.22	NaN	0.61	0.57	0.58	0.55	0.64	NaN
4				0.85	0.83	0.75	0.75	-0.26	NaN	0.64	0.59	0.59	0.63	0.63	NaN
5					0.88	0.81	0.81	-0.11	NaN	0.68	0.59	0.64	0.68	0.63	0.28
6						0.85	0.83	-0.19	-0.02	0.70	0.61	0.53	0.65	0.62	0.13
7							0.76	-0.12	NaN	0.68	0.63	0.54	0.49	0.72	NaN
8								NaN	NaN	0.57	0.59	0.37	0.61	0.61	NaN
9									NaN	NaN	-0.56	-0.33	NaN	NaN	NaN
10										NaN	NaN	NaN	NaN	NaN	NaN
11											0.77	0.71	0.77	0.81	NaN
12												0.68	0.71	0.61	NaN
13													0.75	0.67	NaN
14														NaN	NaN
15															NaN

Table B.5: GRB comparison correlation coefficients - Spring.

B.2 Best Fit Gradients

The following graphs detail the gradient of the line of best fit from ground range bin comparisons for all data and data grouped by season.

GRB	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
1	0.94	0.96	0.96	0.94	0.90	0.94	0.77	-0.21	0.42	0.64	0.58	0.77	0.72	0.73	0.46
2		0.92	0.92	0.90	0.89	0.86	0.73	-0.09	0.29	0.67	0.57	0.74	0.70	0.66	0.51
3			0.92	0.92	0.90	0.87	0.77	-0.05	0.24	0.67	0.57	0.74	0.70	0.65	0.48
4				0.91	0.91	0.91	0.79	-0.00	0.31	0.68	0.58	0.73	0.70	0.65	0.48
5					0.93	0.91	0.81	-0.05	0.28	0.67	0.57	0.74	0.70	0.63	0.45
6						0.91	0.80	-0.04	0.27	0.67	0.57	0.72	0.69	0.63	0.46
7							0.78	-0.27	0.11	0.63	0.58	0.70	0.64	0.57	0.39
8								-0.17	0.15	0.63	0.55	0.67	0.68	0.59	0.39
9									0.71	0.18	0.16	0.16	0.03	0.08	NaN
10										0.79	0.62	0.59	0.67	0.49	NaN
11											0.79	0.90	0.87	0.77	0.57
12												0.95	0.90	0.85	0.65
13													0.85	0.83	0.66
14														0.84	0.68
15															0.75

Table B.6: GRB comparison best fit gradients - All Data.

GRB	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
1	0.97	0.99	1.00	0.98	0.94	0.95	0.84	-0.26	0.49	0.73	0.70	0.87	0.82	0.84	0.53
2		0.92	0.95	0.93	0.91	0.93	0.77	-0.03	0.33	0.74	0.64	0.82	0.77	0.76	0.56
3			0.95	0.94	0.93	0.92	0.81	0.06	0.30	0.72	0.64	0.81	0.77	0.74	0.52
4				0.92	0.93	0.93	0.81	0.11	0.28	0.74	0.64	0.80	0.77	0.72	0.51
5					0.94	0.92	0.80	0.07	0.28	0.72	0.64	0.79	0.77	0.70	0.48
6						0.92	0.78	0.03	0.23	0.73	0.62	0.77	0.74	0.69	0.48
7							0.81	-0.19	NaN	0.68	0.63	0.75	0.69	0.59	0.41
8								0.27	NaN	0.56	0.56	0.65	0.65	0.54	0.42
9									NaN	0.21	0.23	0.27	0.10	0.18	NaN
10										0.78	0.69	0.50	0.83	0.46	NaN
11											0.81	0.90	0.87	0.78	0.56
12												0.94	0.91	0.87	0.64
13													0.86	0.85	0.68
14														0.85	0.69
15															0.77

Table B.7: GRB comparison best fit gradients - Summer.

GRB	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
1	0.90	0.93	0.92	0.90	0.85	0.86	0.66	-0.10	0.39	0.54	0.46	0.62	0.54	0.47	0.15
2		0.91	0.90	0.88	0.87	0.81	0.69	-0.15	0.25	0.63	0.51	0.66	0.60	0.53	0.29
3			0.91	0.90	0.88	0.84	0.76	-0.18	0.18	0.64	0.52	0.67	0.59	0.52	0.29
4				0.91	0.90	0.90	0.79	-0.11	0.32	0.65	0.52	0.66	0.59	0.52	0.30
5					0.92	0.91	0.82	-0.15	0.26	0.65	0.52	0.68	0.60	0.51	0.31
6						0.90	0.82	-0.10	0.28	0.63	0.53	0.66	0.60	0.53	0.34
7							0.75	-0.35	0.13	0.57	0.52	0.61	0.54	0.50	0.30
8								-0.70	NaN	0.66	0.46	0.59	0.61	0.54	NaN
9									0.25	0.14	0.08	0.02	-0.10	-0.04	NaN
10										0.84	0.56	0.65	0.49	0.46	NaN
11											0.75	0.86	0.82	0.72	0.52
12												0.92	0.83	0.77	0.57
13													0.80	0.76	0.56
14														0.79	0.54
15															0.59

Table B.8: GRB comparison best fit gradients - Autumn.

GRB	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
1	0.87	0.92	0.87	0.85	0.84	0.92	0.75	NaN	NaN	0.50	0.22	0.51	0.50	0.58	NaN
2		0.91	0.88	0.85	0.87	0.82	0.82	NaN	NaN	0.51	0.31	0.53	0.54	0.40	NaN
3			0.88	0.86	0.87	0.85	0.85	NaN	NaN	0.50	0.30	0.53	0.50	0.33	NaN
4				0.90	0.91	0.94	0.83	NaN	NaN	0.53	0.31	0.58	0.50	0.38	NaN
5					0.93	0.92	0.83	NaN	NaN	0.53	0.30	0.58	0.51	0.34	NaN
6						0.89	0.83	NaN	NaN	0.54	0.30	0.57	0.52	0.37	NaN
7							0.71	NaN	NaN	0.46	0.28	0.31	0.38	0.08	NaN
8								NaN	NaN	0.58	0.27	0.44	0.32	0.02	NaN
9									NaN	NaN	NaN	NaN	NaN	NaN	NaN
10										NaN	NaN	NaN	NaN	NaN	NaN
11											0.52	0.88	0.76	0.59	NaN
12												0.87	0.78	0.81	NaN
13													0.69	0.68	NaN
14														0.64	NaN
15															NaN

Table B.9: GRB comparison best fit gradients - Winter.

GRB	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
1	0.69	0.71	0.69	0.71	0.58	0.58	0.55	NaN	NaN	NaN	0.10	NaN	NaN	NaN	NaN
2		0.81	0.81	0.78	0.73	0.64	0.61	-0.32	NaN	0.59	0.53	0.58	0.79	0.66	NaN
3			0.84	0.85	0.80	0.78	0.79	-0.28	NaN	0.76	0.55	0.65	0.58	0.61	NaN
4				0.84	0.86	0.86	0.87	-0.26	NaN	0.79	0.54	0.64	0.73	0.62	NaN
5					0.91	0.91	0.95	-0.10	NaN	0.75	0.57	0.71	0.72	0.63	0.24
6						0.92	0.90	-0.20	-0.01	0.78	0.58	0.59	0.70	0.62	0.11
7							0.78	-0.11	NaN	0.67	0.50	0.53	0.49	0.61	NaN
8								NaN	NaN	0.63	0.51	0.32	0.62	0.53	NaN
9									NaN	NaN	-0.47	-0.30	NaN	NaN	NaN
10										NaN	NaN	NaN	NaN	NaN	NaN
11											0.67	0.69	0.80	0.68	NaN
12												0.75	0.75	0.68	NaN
13													0.68	0.75	NaN
14														NaN	NaN
15															NaN

Table B.10: GRB comparison best fit gradients - Spring.

B.3 No Of Points

The following graphs detail the number of data points in ground range bin comparisons for all data and data grouped by season.

GRB	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
1	2200	2335	2373	2300	2186	527	375	101	56	1139	1351	1137	654	560	172
2		5174	5514	5428	5204	1247	924	188	126	2274	2746	2327	1334	1075	271
3			6425	6341	6091	1439	1033	213	138	2552	3108	2605	1479	1180	289
4				7176	6939	1571	1162	233	152	2781	3357	2808	1588	1264	300
5					7166	1601	1177	227	146	2825	3410	2861	1610	1279	303
6						1634	1197	244	156	2807	3410	2866	1622	1271	306
7							461	88	55	695	835	774	498	423	113
8								61	23	501	618	570	395	343	104
9									58	209	187	162	94	68	0
10										150	126	110	71	51	0
11											2465	2104	1194	896	207
12												2397	1344	1058	250
13													1292	1009	249
14														770	196
15															177

Table B.11: GRB comparison number of data points - All Data.

GRB	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
1	1157	1197	1205	1146	1054	271	208	58	34	593	713	615	399	375	135
2		2124	2190	2108	1974	451	394	87	57	973	1224	1068	722	623	203
3			2428	2342	2187	496	424	97	62	1060	1344	1153	787	671	217
4				2497	2329	514	458	100	67	1132	1429	1252	846	720	224
5					2347	511	460	96	64	1146	1429	1259	859	723	222
6						516	461	96	62	1150	1436	1267	876	725	225
7							146	37	0	319	376	367	282	246	85
8								37	0	288	344	331	260	243	88
9									0	106	96	86	60	44	0
10										70	57	51	35	27	0
11											1136	1025	663	528	163
12												1187	779	645	197
13													760	633	196
14														512	163
15															149

Table B.12: GRB comparison number of data points - Summer.

GRB	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
1	1043	1138	1168	1154	1132	256	167	43	22	546	638	522	255	185	37
2		3048	3322	3318	3228	795	530	101	67	1299	1520	1257	610	450	68
3			3995	3997	3902	942	609	116	74	1490	1762	1450	690	507	72
4				4677	4608	1056	704	133	83	1647	1926	1554	740	542	76
5					4817	1089	717	131	80	1677	1979	1600	749	554	81
6						1117	736	148	92	1655	1972	1597	744	544	81
7							315	51	34	375	458	406	215	176	28
8								24	0	213	274	239	135	100	0
9									38	103	91	76	34	24	0
10										78	67	57	34	22	0
11											1327	1077	529	366	44
12												1208	563	411	53
13													530	374	53
14														256	33
15															28

Table B.13: GRB comparison number of data points - Autumn.

GRB	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
1	393	421	424	417	410	76	70	0	0	92	139	109	47	35	0
2		1008	1077	1069	1023	175	170	0	0	170	251	201	94	79	0
3			1282	1263	1210	213	183	0	0	185	280	223	100	85	0
4				1458	1402	230	207	0	0	202	298	232	107	89	0
5					1423	241	211	0	0	198	295	234	98	89	0
6						241	214	0	0	196	290	234	102	86	0
7							72	0	0	44	55	58	30	27	0
8								0	0	44	67	55	36	24	0
9									0	0	0	0	0	0	0
10										0	0	0	0	0	0
11											81	75	26	24	0
12												103	43	41	0
13													50	38	0
14														27	0
15															0

Table B.14: GRB comparison number of data points - Winter.

GRB	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
1	76	87	94	94	97	47	35	0	0	0	24	0	0	0	0
2		544	651	658	686	311	224	28	0	105	164	137	66	60	0
3			856	873	914	399	285	34	0	125	210	179	81	68	0
4				1186	1257	480	344	44	0	177	265	215	92	86	0
5					1379	490	352	39	0	184	273	228	100	89	21
6						517	367	53	22	204	299	246	101	90	21
7							193	27	0	74	119	106	47	42	0
8								0	0	69	98	81	40	35	0
9									0	0	23	22	0	0	0
10										0	0	0	0	0	0
11											94	77	34	27	0
12												117	44	43	0
13													49	43	0
14														0	0
15															0

Table B.15: GRB comparison number of data points - Spring.

References

Andrews, D., Holton, J. & Leovy, C. [1987]. *Middle Atmosphere Dynamics.*, Academic Press.

Avery, S., Avery, J., Valentic, T., Palo, S., Leary, M. & Obert, R. [1990]. A new meteor echo detection and collection system: Christmas Island mesospheric measurements., *Radio Science.* **25**: 657–669.

Baggaley, W. & Bennett, R. [1996]. The meteor orbit radar facility AMOR: Recent developments., *Astronomical Society of the Pacific Conference Series.* **104**: 65–70.

Baker, D. & Stair, A. [1988]. Rocket measurements of the altitude distribution of the hydroxyl airglow, *Physica Scripta.* (37): 611–622.

Beard, A., Mitchell, N., Williams, P. & Kunitake, M. [1999]. Non-linear interactions between tides and planetary waves resulting in periodic tidal variability, *Journal of Atmospheric and Solar Terrestrial Physics.* **61**: 363–376.

Bevington, P. [1969]. *Data Reduction and Error Analysis for the Physical Sciences.*, McGraw-Hill.

Chapman, S. & Lindzen, R. [1970]. *Atmospheric Tides - Thermal and Gravitational*, D. Reidel, Dordrecht, Holland.

Fauliot, V., Thuillier, G. & Herse, M. [1995]. Observations of the E-region horizontal winds in the auroral zone and at mid-latitudes by a ground-based interferometer., *Ann. Geophys.* **13**: 1172–1186.

Fellous, J., Spizzichino, A., Glass, M. & Massebeuf, M. [1974]. Vertical propagation of tides at meteor heights., *Journal of Atmospheric and Terrestrial Physics.* **36**: 385–396.

Forbes, J. M. [1982]. Atmospheric tides 2. The solar and lunar semidiurnal components., *Journal of Geophysical Research* **87**(A7): 5241–5252.

Forbes, J., Portnyagin, Y., Makarov, N., Palo, S., Merzlyakov, E. & Zhang, X. [1999]. Dynamics of the lower thermosphere over South Pole from meteor radar wind measurements., *Earth Planets and Space.* **51**(7–8): 611–620.

Fraser, G. J. [1984]. Partial-reflection spaced antenna wind measurements, *MAP* (13): 233–247.

Garcia, R. & Prusa, J. [1997]. The propagation and breaking of gravity waves excited by forcing in the troposphere., *NATO ASI Series.* **50**: 169–186.

Greenhow, J. & Hall, J. [1960]. Diurnal variations of density and scale height in the upper atmosphere., *Journal of Atmospheric and Terrestrial Physics.* **18**: 203.

Hagan, M., Burrage, M., Forbes, J., Hackney, J., Randel, W. & Zhang, X. [1999]. GSWM-98: Results for migrating solar tides., *Journal of Geophysical Research.* **104**: 6813–6828.

Hall, J. & Bullough, K. [1963]. *Nature, Lond.* **200**: 642.

Harris, F. [1978]. On the use of windows for harmonic analysis with the discrete fourier transform., *Proceedings of the IEEE.* **66**(1): 51–83.

Hernandez, G. [1986]. *Fabry-Perot Interferometers*, Cambridge University Press.

Hocking, W. [1997]. System design, signal-processing procedures, and preliminary results for the Candadian (London, Ontario)VHF atmospheric radar., *Radio Science.* **32**(2): 687–706.

Holton, J. [1992]. *An Introduction To Dynamic Meteorology*, Academic Press.

Kaiser, T., Pickering, W. & Watkins, C. [1969]. Ambipolar diffusion and motion of ion clouds in the Earth's magnetic field, *Planetary Space Sci.* **17**: 519–552.

Kallman-Bijl, H., Boyd, R., Lagow, H., Poloskov, S. & Priester, W. [1961]. *Commitee on space research, international reference atmosphere. North-Holland, Amsterdam.* .

Laplace, P. [1799]. *Méchanique Céleste*, Paris.

Lawrence, B. [1997]. Some aspects of the sensitivity of stratospheric climate simulation to model lid height., *Journal of Geophysical Research - Atmospheres.* **102**(D20): 23805–23811.

Lindzen, R. & Chapman, S. [1969]. Atmospheric tides, *Space Science Reviews.* **10**: 3–.

Lindzen, R. & Hong, S. [1974]. Effects of mean winds and horizontal temperature gradients on solar and lunar semidiurnal tides in the atmosphere., *Journal Atmospheric Science.* **31**: 1421–1466.

Lomb, N. [1976]. *Applied Space Sciences.* **39**: 447.

Longuet-Higgins, M. [1968]. The eigenfunctions of Laplace's tidal equations over a sphere., *Philos. Trans. R. Soc. London Ser. A* (262): 511–607.

Manson, A., Meek, C., Teitelbaum, H., Schminder, R., Kurschner, D., Smith, M., Fraser, G. & Clark, R. [1989]. Climatologies of semidiurnal and diurnal tides in the middle atmosphere (70–110 km) at middle latitudes (40–55°)., *Planetary and Space Sciences.* **51**(7/8): 579–593.

Mitchell, N. & Howells, V. [1998]. Vertical velocities associated with gravity waves measured in the mesosphere and lower thermosphere with the EISCAT VHF radar., *Annales Geophysicae-Atmospheres Hydrosphers and Space Sciences.* **16**(10): 1367–1379.

Mitchell, N., Williams, P., Beard, A., Buesnel, G. & Muller, H. [1996]. Non-linear planetary tidal wave interactions in the lower thermosphere observed by meteor radar., *Annales Geophysicae-Atmospheres Hydrospheres and Space Sciences.* **14**(3): 364–366.

Motorola [1990]. *Motorola Linear and Interface Integrated Circuits Handbook*, Motorola inc.

Nakamura, T., Toshitaka, T. & Masaki, T. [1997]. Development of an external interferometer for meteor wind observation attached to the MU radar, *Radio Science.* **32**(3): 1203–1214.

O'Brien, B., Allum, F. & Goldwire, H. [1965]. Rocket measurements of midlatitude airglow and particle precipitation, *J. Geophys Res.* **70**.

Plagmann, M., Marsh, S., Baggaley, W., Bennett, R., Deutsch, K., Fraser, G., Hernandez, G., Lawrence, B., Plank, G. & Smith, R. [1998]. Annual variation of airglow heights derived from wind measurements, *Geophysical Research Letters.* **25**: 4457–4460.

Poole, L. [1988]. The Grahamstown all-sky meteor radar., *Journal Atmospheric and Terrestrial Physics.* **56**: 1731–1752.

Raghava Reddi, C., Rajeev, K., Ramkumar, G., Kamath, K. & Shenoy, K. [1993]. Meteor wind radar system at Trivandrum (8.5°N, 77°E)., *Indian Journal of Atmospheric and Terrestrial Physics.* **55**(9): 1219–1231.

Salah, J., Deng, W. & Clark, R. [1997]. Observed dynamical coupling through tidal wave propagation in the mesosphere and lower thermosphere at midlatitudes., *Journal of Atmospheric and Solar-Terrestrial Physics.* **59**(6): 641–654.

Salby, M. [1996]. *Fundamentals of Atmospheric Physics.*, Academic Press.

Scargle, J. [1982]. Studies in astronomical time series analysis. II. Statistical aspects of spectral analysis of unevenly spaced data., *The Astrophysical Journal.* **263**: 835–853.

Scargle, J. [1989]. Studies in astronomical time series analysis. III. fourier transforms, autocorelation functions, and cross-correlation functions of unevenly spaced data., *The Astrophysical Journal.* **343**(2): 874–887.

Schoeberl, M. [1985]. The penetration of mountain waves into the middle atmosphere, *Journal of the Atmospheric Sciences.* **42**(24): 2856–2864.

Siegel, A. [1980]. Testing for periodicity in a time series., *Journal of the American Statistical Association.* **75**: 345–348.

Smith, R. [1998]. Vertical winds: A tutorial., *Journal of Atmospheric and Solar Terrestrial Physics.* **60**(14): 1425–1434.

Smith, S. [1996]. *Optical and Radar Wind Comparisons in the Mesosphere and Lower Thermosphere.*, PhD thesis, University of Canterbury.

Taylor, A. [1991]. *A Meteor Orbit Radar.*, PhD thesis, University of Canterbury.

Tsuda, T. [1995]. A preliminary report on observations of equatorial atmosphere dynamics in Indonesia with radars and radiosondes., *J. Meteorol. Soc. Jpn* **73**: 393–406.

Vial, F. [1989]. Tides in the middle atmosphere., *Journal of Atmospheric and Terrestrial Physics.* **51**(1): 3–17.

Vincent, R., Tsuda, T. & Kato, S. [1988]. A comparitive study of mesospheric solar tides at Adelaide and Kyoto., *J. geophys. Res.* **93**(D1): 699–708.

von Biel, H. [1995]. Ground based radar investigation of the Antarctic mesosphere., *Advanced Space Research.* **16**(5): 591–598.

Walterscheid, R., DeVore, J. & Venkateswaren, S. [1980]. Influence of mean zonal motion and meridional temperature gradients on the solar semidiurnal atmospheric tide: A revised spectral study with improved heating rates., *J. Atmos. Sci* **37**: 455–470.

Walterscheid, R. & Venkateswaren, S. [1979a]. Influence of mean zonal motion and meridional temperature gradients on solar semidiurnal atmospheric tide: A spectral study, Part I, Theory, *Journal of Atmospheric Science.* **36**: 1623–1635.

Walterscheid, R. & Venkateswaren, S. [1979b]. Influence of mean zonal motion and meridional temperature gradients on solar semidiurnal atmospheric tide: A spectral study, Part II, Numerical results, *Journal of Atmospheric Science.* **36**: 1636–1635.