This is to certify that the

thesis entitled

THE EFFECT OF WIND AND WAVE
CHARACTERISTICS ON EVAPORATION

presented by

David William Harms

has been accepted towards fulfillment
of the requirements for

Master of Science degree in Civil Engineering

Major professor

Date 11/12/87

O-7639

THE EFFECT OF WIND AND WAVE CHARACTERISTICS ON EVAPORATION

By

David William Harms

A THESIS

Submitted to
Michigan State University
in partial fulfillment of the requirements
for the degree of

MASTER OF SCIENCE

Department of Civil and Environmental Engineering

1987

ABSTRACT

THE EFFECT OF WIND AND WAVE CHARACTERISTICS ON EVAPORATION

By

David William Harms

Evaporation from a free water surface and the effect of waves has been the subject of many past studies. The presence of waves may act to increase the surface area thereby increasing evaporation, or they may act as wind breaks to reduce surface exposure thus retarding the evaporation process. Disagreement exists regarding the onset and role of flow separation on this process. The purpose of this study was to gain further insight into the relationship between wave characteristics, the air flow over them, and their effects on evaporation rate.

Using the Environmental Wind Tunnel at Michigan State University, evaporation experiments were conducted for several combinations of fetches and wind speeds with both wind-generated and mechanically-generated waves.

As a result of the study, a linear relationship was found to exist between the mass transfer coefficient and the shear velocity. At high wind speeds the presence of mechanically-generated waves appear to significantly increase both the shear velocity and the mass transfer coefficient over the case of waves generated by wind only. The results of this study compare favorably with the field data of studies of previous researchers.

To Reinier and my wife Cindy:

Thank you for your patience and gentle persistence.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

LIST OF FIGURES - CONTINUED

LIST OF SYMBOLS

C	Wave Speed in meters per second.
C_s	Surface Concentration in grams per cubic meter.
C_{10}	Concentration at 10 cm in grams per cubic meter.
e	Vapor pressure in millimeters of Mercury.
e_s	Saturation vapor pressure in millimeters of Mercury.
H	Significant wave height in centimeters.
K	Von Karman constant.
k_g	Mass transfer coefficient in meters per hour.
k_{10}	Mass transfer coefficient at 10 cm in meters/hour.
L	Wavelength in meters.
N	Mass flux (evaporation rate) in grams per square meter per hour.
P_{atm}	Atmospheric pressure in millimeters of Mercury.
R	Gas constant for water.
T_w	Wave period in seconds.
T	Temperature in degrees Kelvin.
T_{db}	Dry bulb temperature in degrees Celsius.
T_{wb}	Wet bulb temperature in degrees Celsius.
$u*$	Shear velocity in meters per second.
U	Local velocity in meters per second.
U_{10}	Velocity at 10 cm above surface in meters/second.
z	Distance from wall/still-water-surface in meters.
z_o	Surface roughness length in meters.
α	Power law exponent.
Γ	Gamma function.
ρ	Density in grams per cubic meter.
ζ	Wave RMS in centimeters.

Chapter I

INTRODUCTION

The problem of evaporation from a free water surface has been of interest for well over the past 100 years. In that time many methods, theories, and equations have been produced for understanding and estimating evaporation. Yet today, despite the importance for managing our water resources, a full understanding of the transfer processes involved has not been achieved.

Evaporation takes place when the vapor pressure above a free surface is less than the saturated vapor pressure at the surface. When a vapor pressure difference exists, the net flux of molecules must be directed away from the water surface. With no wind then the problem is quite simple but, when a wind is introduced, the water vapor near the surface is carried up quickly because of the shear-induced turbulence, thus increasing the evaporation rate. The problem is further complicated by the fact that the dynamic interaction of the air and water produces waves which in turn affect evaporation.

There are several ways in which waves could enhance evaporation. Most simply, the presence of waves could be thought to increase the surface area and, thus, evaporation. Waves may also constitute a surface roughness and thus augment the turbulent transport of water vapor. Further, the spray blown off breaking waves would certainly be an additional vapor source.

On the other hand, in a similar way as wind breaks reduce surface exposure, waves on a water surface could give rise to organized flow patterns near the surface, specifically, flow separation which might act as a barrier to mass transport (Stewart, 1967). Should flow separation exist, this would create pockets between the wave crest where turbulence is less intense than in the flow above these pockets.

Therefore, a description of the evaporation mechanism requires knowledge of how the air flow regime changes over different types of waves. Most researchers agree that flow separation occurs at one time or another over wind generated waves. Some insist that it coincides with the onset of wave breaking (Banner and Phillips, 1974; Banner and Melville, 1976) while others have stated that it coincides with other events, such as when the wave's phase velocity becomes less than the winds shear velocity (Wu, 1969). More work is needed to determine when flow separation occurs and its effect on evaporation.

The purpose of this study is to provide experimental data which can be used to gain insights about the relationship between wave characteristics, air flow characteristics and evaporation rate.

More specifically, the objectives of this study are:

- to obtain evaporation rates for a range of wind/wave conditions;
- to correlate wind and wave parameters to the evaporation coefficient;
- to infer the onset of flow separation, and how this influences evaporation.

Chapter II

REVIEW OF LITERATURE

When wind blows over a water surface, both the velocity of the wind and waves on the surface, have a significant effect on the evaporation from that surface. This chapter will review the available literature concerning air flow and air flow separation over waves and their effects upon evaporation rates.

2.1 Wind Profiles over Waves

Many laboratory studies of air flow over water waves, Sirovica (1982), Lai and Plate (1969), and Chang (1968), have shown that the law of the wall applies to the data in their experiments. Therefore, the velocity profiles in these experiments have all been considered to be logarithmic, and capable of being described as follows:

$$U = u^*/K \ln(z/z_o) \qquad (1)$$

where: U is velocity ;

u* is shear velocity;

z is distance from a wall;

K is the Von Karman constant; and

z_o is the surface roughness length.

Although a logarithmic profile is reasonable for the main flow, Chang (1968) has shown that it fails to be

valid for the region up to one wave height above the mean water surface. Others, Kondo, Fujinawa, and Naito (1973), have found that this is not always the case but is rather Reynolds number and roughness dependent; i.e. the water surface becomes aerodynamically rough above a certain critical Reynolds number. (Aerodynamically rough meaning that the roughness protrusions extend beyond the viscous sublayer.)

Being more quantitative, Brutsaert (1975), in a review of the findings of several studies, states that the log profile fails below the level of $u/u^* < 5$, which is just above the top of the roughness elements.

2.2 Humidity Profiles over Waves

Over a homogeneous surface the mean wind profile is logarithmic. If water can be considered simply a passive admixture (Brutsaert, 1975) then its profile is also logarithmic, and Reynolds analogy is reasonable. Supporting this, the experimental data obtained by Lai and Plate (1969) showed humidity trends were similar to those of u^*.

On the other hand, as with velocity, Street (1979) has shown that a region of molecularly dominated flow exists at the interface. In this region water vapor flux is controlled by molecular diffusion only, and Reynolds analogy fails.

2.3 Flow Separation above Waves

Over 60 years ago Jeffreys (1925) stated that air flowing over a wave separates somewhere on the downstream side of the wave crest, and reattaches on the upwind face of the next crest. (This would cause an asymmetry with respect to pressure for wave crests resulting in energy transfer and wave growth.) Stewart (1967) hypothesized a streamline pattern in the form of a cats-eye in the wave trough (similar to that of flow separation over a cavity).

He stated that this must occur since, even at rather low heights above the surface, there appears to be little wave-like motion in the air stream.

The validity of this picture painted by Jeffreys and Stewart could only be checked with measurements in the wave troughs themselves, not a simple task with an undulating water surface. To bypass this problem, Chamberlain (1968), Owen and Thomson (1962), and others have studied flow over wave-like solid, flexible, or moving boundaries. But the value of these studies has been questioned by Lai and Plate (1969) who argue that the coupling of the flow in the two fluids should not be neglected; a fluid boundary can induce velocity fluctuation and turbulence which are not considered in flow over a solid boundary.

Chang (1968) examined the structure of the turbulent wind immediately above and between the crests of water waves. Using a hot-wire anemometer and a probe support system that was capable of following the fluctuating water surface, he found air flow separation between the crests supporting the separation mechanism of energy transfer outlined first by Jeffreys and, later, by Stewart. He also noticed that the waves formed sharp crests, shallow troughs and were skewed with larger, smoother upwind sides. This asymmetry suggested a separation of the air flow on the leeward face of the crests.

In lab experiments with a wave/wind tunnel, similar to the one used in this study, Easterbrook (1968) determined evaporation rates and did flow visualization for different wind/wave combinations. From numerous photographs Easterbrook postulated the existence of a twin-vortex system on the downwind slope of the wave. He concluded that as the wave steepness increased, unstable twin vortices begin to form which breakdown to cause increased turbulence. With further increase in steepness, vortices become stable and possibly limit mass transfer. However, once wave breaking occurs the twin vortices are again shed more easily.

More recently, in flow visualization experiments over a standing wave created by a submerged cylinder, Banner and Melville (1976) concluded that the occurrence of separation corresponds to the onset of wave breaking (if and only if breaking occurs). They found flow separation could be obtained at very low wind speeds, but they could not cause the air flow to separate over unbroken waves even at the highest speeds in their study. (They define the onset of breaking as the event in which certain fluid elements are moving forward with greater speed than the propagation speed of the wave as a whole.)

Although the above studies indicate that flow separation can occur, the available data does not permit the definition of the wind and wave characteristics that yield flow separation.

2.4 Effects of Flow Separation on Evaporation

Only a limited number of studies have been conducted to relate wave parameters to evaporation. Easterbrook established steady state wind and wave conditions, sealed off his tunnel and recorded the change of humidity with time. He found that at certain wind conditions, the dead air spaces in the lee of the wave crests and vortices in the wave troughs became an effective barrier to vertical transport. In this way for certain wind/wave combinations, lower evaporations rates were encountered

than if no waves were present. In his laboratory study however, some of Easterbrooks methods were quite crude. He used a hot-wire anemometer centered in his tunnel to yield the mean velocity during a run. Additionally, a wet bulb thermistor assembly with a time constant of approximately 10 seconds, (quite slow for an experiment of this type,) was centered in his tunnel to give the change of humidity with time. (To do this he had to assume complete mixing in his entire 40 x 4 x 3 foot tunnel.)

Chamberlain (1968), in his study using a wave-shaped surface covered with a wet cloth, found a bulk evaporation coefficient which seemed to have a minimum at middle wind speeds, being higher at both high and low speeds.

Unfortunately, all the studies do not agree. Lai and Plate (1969) suggest that if evaporation rate depends upon turbulent diffusion away from the surface, and molecular near the surface, then separation causing more turbulence increases evaporation. Their data have no suggestion of a decreased evaporation rate for increased wind speed. But since they had no wave maker, their data falls outside of that of Easterbrooks and approximates only short fetch conditions, while fetches of up to one mile were modeled by Easterbrook.

The information on the air flow and evaporation over waves is still limited. The results of studies that have been made are contradictory. Therefore, it seems appropriate to further investigate these phenomena in order to determine the effects of wave characteristics on evaporation.

Chapter III

EXPERIMENTAL PROCEDURES AND DATA ANALYSIS

The Environmental Wind Tunnel at Michigan State University was recently modified to lengthen the test section and install a mechanical wave generator. In this new tunnel, instruments were installed to make measurements of humidity, air speed, water surface displacement and temperature. Details of the tunnel and experiments are given below.

3.1 Environmental Wind Tunnel

The Environmental Wind Tunnel at Michigan State University has been used for this experiment. This tunnel has a test section of 17 meters, and is of the closed-circuit type in which air is recirculated. The test section is 1 meter wide with a maximum allowable water depth of 0.3 meters. This leaves a 1-m high section for the air flow. Figure 3.1.1 shows a schematic of the tunnel.

Air speeds between 0 and 15 m/s can be generated by a 15 hp variable frequency speed control system which drives a 44" diameter fan. The air flow conditions are made uniform by an aluminum honeycomb flow straightener, 1/4" inner diameter by 3" long, at the test section entrance. Turning vanes are used to turn the air flow at each corner.

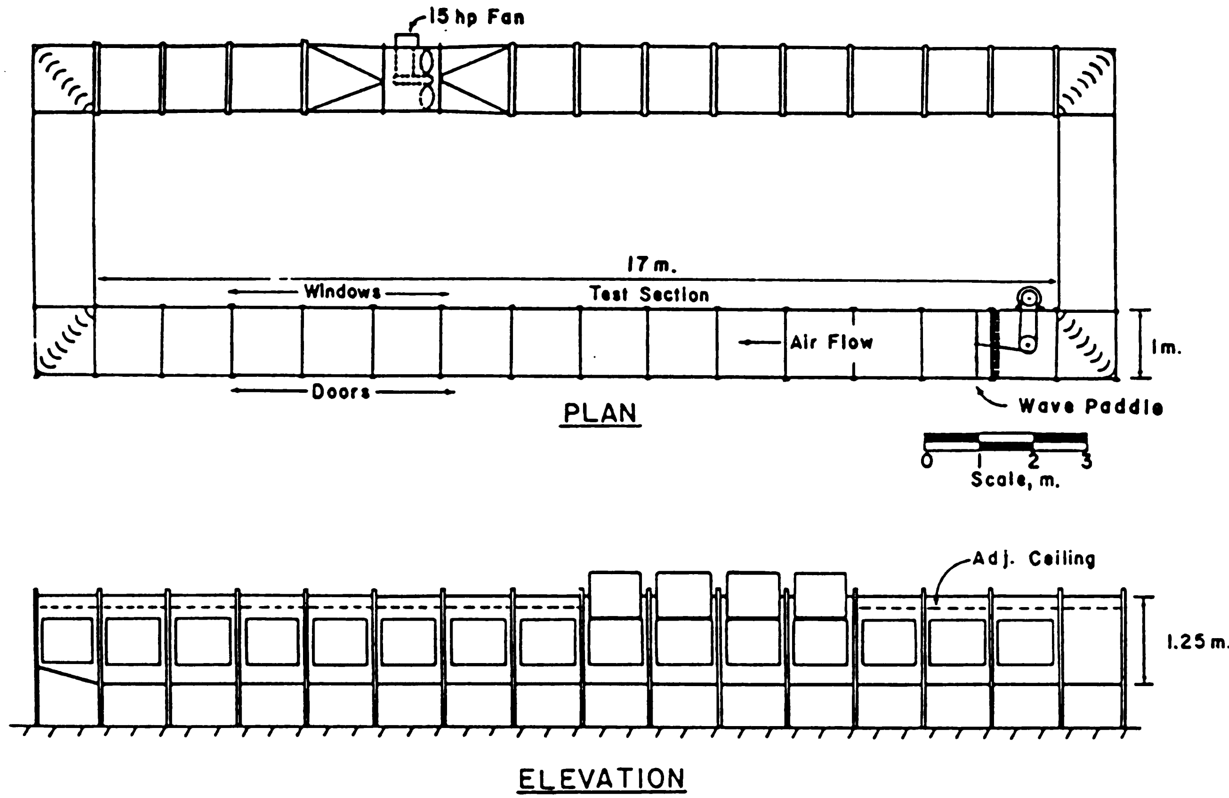

Figure 3.1.1 ENVIRONMENTAL WIND TUNNEL
MICHIGAN STATE UNIVERSITY

The wind tunnel, especially designed to study mass transfer processes at the air water interface, incorporates many features. Those pertinent to the experiment conducted are listed below:

1. A mechanical wave generator at the upwind end of the tunnel which can generate waves with varying amplitude and frequency;
2. The air temperature in the wind tunnel can be controlled between 15 and 30°C;
3. There is great flexibility to conduct visualization studies as the bottom and sides of the tunnel are constructed of lucite material;
4. The 1-m tunnel width and the adjustable ceiling height minimize the effects of the side walls on the air flow;
5. The wind tunnel air is isolated from the laboratory and can be vented to the outside through the roof at sufficiently high rates so that low background water vapor concentrations can be maintained.

The wind tunnel is equipped with an instrument support carriage, whose vertical position can be set remotely by a stepping motor control system. Vertical positions of probes can be read with a precision of 10^{-3} m.

3.2 Experimental Program

Prior to all experiments the tunnel was cleaned. It was then filled with tap water to a depth of 21.5 cm, sealed, and allowed to run for at least three hours. During this time, the ventilation system was exhausting humid air, and the water, wet bulb, and background dry bulb temperatures were monitored. Experiments then began sometime after equilibrium conditions had been reached.

The five experimental stations chosen for the experiment were positioned at fetches of 5.0, 7.0, 9.0, 12.0 and 14.75 meters, sufficiently far downwind to yield approximately uniform wind and wave conditions. Data was taken at two wind speeds, 3.5 and 7.0 m/s (at z=10 cm); and over both 2 Hz mechanically-generated waves and naturally wind-generated waves. The following measurements were performed:

1. Velocity profile measurements to determine $u*, U_{10}, z_o$;
2. Wave measurements to determine wave height, length, and speed;
3. Humidity profile measurements to evaluate mass transfer coefficient.

The water temperature in the tunnel was monitored continuously during experiments and barometric pressure was recorded daily.

3.3 Instrumentation and Measuring Procedures

This section presents a detailed discussion of the procedures and data reduction techniques applied in the experiment. A comprehensive description of the equipment employed is provided in Appendix A.

3.3.1 Velocity Measurements

The mean air velocity was measured by a Pitot-static tube of 3.25 mm outer diameter, connected to a pressure transducer. The pitot tube was mounted on a probe support carriage whose height was controlled by a stepping motor and on-line recording of the probe position. The following procedure was used to obtain velocities at discrete points in a vertical profile:

1. The pitot tube was transversed downwards from 45 cm. above the water surface.
2. The pressure difference between the stagnation and static pressure holes of the pitot tube was converted to an electrical signal by the pressure transducer.
3. At distinct points, the instrument carriage carrying the pitot tube was stopped, and the computer was signalled to sample utilizing an A/D converter. Using the program PROFILE (shown in Appendix E,) the computer recorded both the height of the pitot tube and calculated the average velocity at that height. When sampling, the computer took 5 samples per second

over a period of 30 seconds, and computed one average velocity.

(This covered approximately 60 wave passages).

4. Each velocity profile comprised approximately 30 points at different heights.

The pressure transducer was calibrated once per day as described in Appendix B.

The velocity parameters u* and z_o were determined by plotting the velocity profiles on semi-logarithmic graph paper and fitting the logarithmic velocity profile:

$$U = 5.75\ u^* \log(z/z_o) \qquad (2)$$

Velocity profiles were then replotted on a log-log scale and the power law exponent was found by fitting the power law equation:

$$U = U_{10}\ (z/z_{10})^{\alpha} \qquad (3)$$

where α is the power law exponent and z_{10} is 0.1 m.

3.3.2 Wave Measurements

A capacitance wave gage was installed in the tunnel to continuously measure the displacement of the water surface. The gage, shown in Figure 3.3.1 was constructed of 32 gauge teflon-insulated copper wire stretched vertically between stainless steel support arms. The wire of the gage and the water acted as the two plates of a capacitor with the wire insulation being the dielectric. Changes in capacitance due to changes in water depth were then converted to voltage signals by a capacitance bridge which could be monitored by an A/D converter.

Using this wave gage and the Fortran program WAVE, shown in Appendix F, 100 samples per second were recorded over a period of 200 seconds of the instantaneous wave heights passing the gage (approximately 400 waves). These samples were then processed to yield the rms (ζ) of the sampled wave train from which significant wave height was calculated as follows:

$$H = 2\zeta/0.7071 \qquad (4)$$

Wave speed was found by observation. Using a stop watch, the time that an individual wave took to pass between two markers in the wind tunnel was recorded. This process was repeated several times with the individual

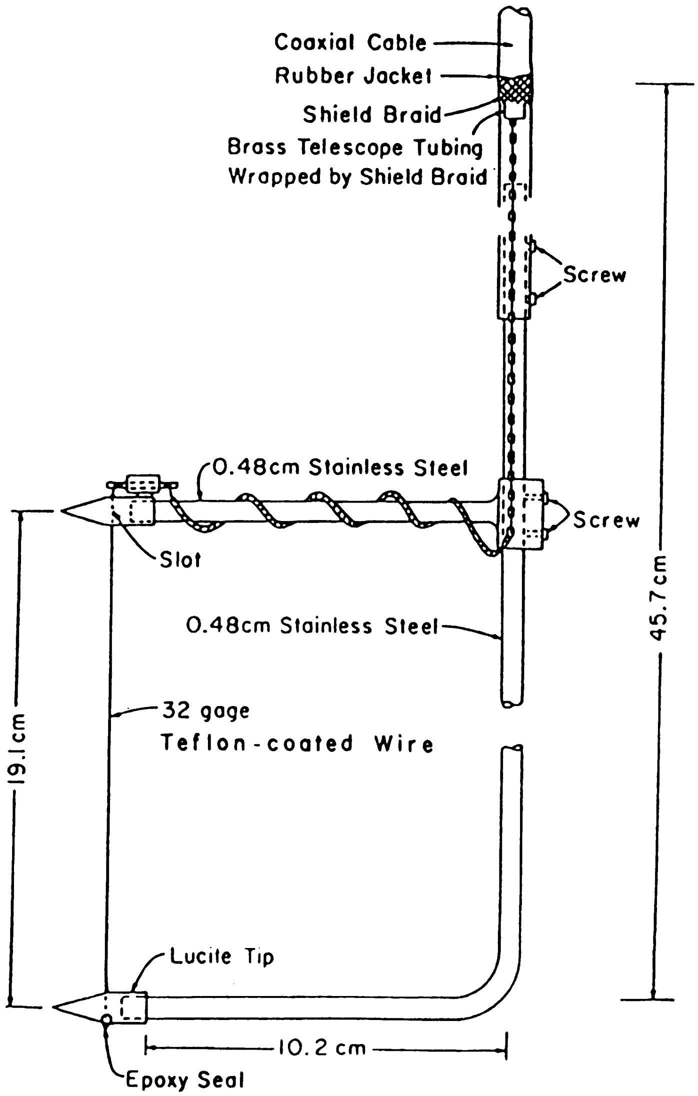

Figure 3.3.1 WAVE GAGE

times averaged. Very little variation was observed in this wave parameter suggesting that this procedure was adequate.

Wave period was obtained using a storage oscilloscope to record the signal from the wave gage. From the stored trace, the average period over 50 waves was calculated. The oscilloscope was calibrated once each day using a trace of known period. As with the wave speed measurements, little variation in wave period occurred suggesting an adequate measuring procedure.
Wavelength was found from wave speed and period by:

$$L = CT_w \tag{5}$$

where: L is the wavelength in meters;
C is the wave speed in m/s; and
T_w is the wave period in seconds.

3.3.3 Humidity measurements

The absolute humidity of the air passing over the water waves was measured using the dry and wet bulb technique. Previous studies had indicated that simply mounting a wet bulb thermometer or thermistor in the tunnel would not be sufficient, because some of the fan speeds used in the tunnel would not provide adequate or

constant wet bulb ventilation and a wick could dry out before an experiment was finished. Therefore, the humidity probe pictured in Figure 3.3.2 was designed.

The main reason for constructing the humidity probe was to provide a constant air flow rate past the wet wick at any tunnel wind speed. This is because low velocity air flow past the wet bulb is known to cause erroneously high temperature readings (Tanner, 1971; Bindon, 1963). So the design incorporated an exhaust port to which a vacuum pump could be connected to provide a constant flow rate. The proper ventilation rate was determined experimentally. Details are given in Appendix D. Another feature of the design was a self contained reservoir from which four strands of clean thread were passed through a hole to the wick of the wet bulb thermometer inside of the probe. It was found that this arrangement yielded a constant feeding rate that prevented the wet bulb from drying out for a period of approximately one hour. Finally, the probe was designed to employ a mercury-in-glass thermometer that met National Bureau of Standards specifications, rather than a thermistor. This was done because previous experiments with an early humidity probe prototype showed that the thermistors were slightly self-heating when enclosed in a wick.

Although the new humidity probe design was based upon

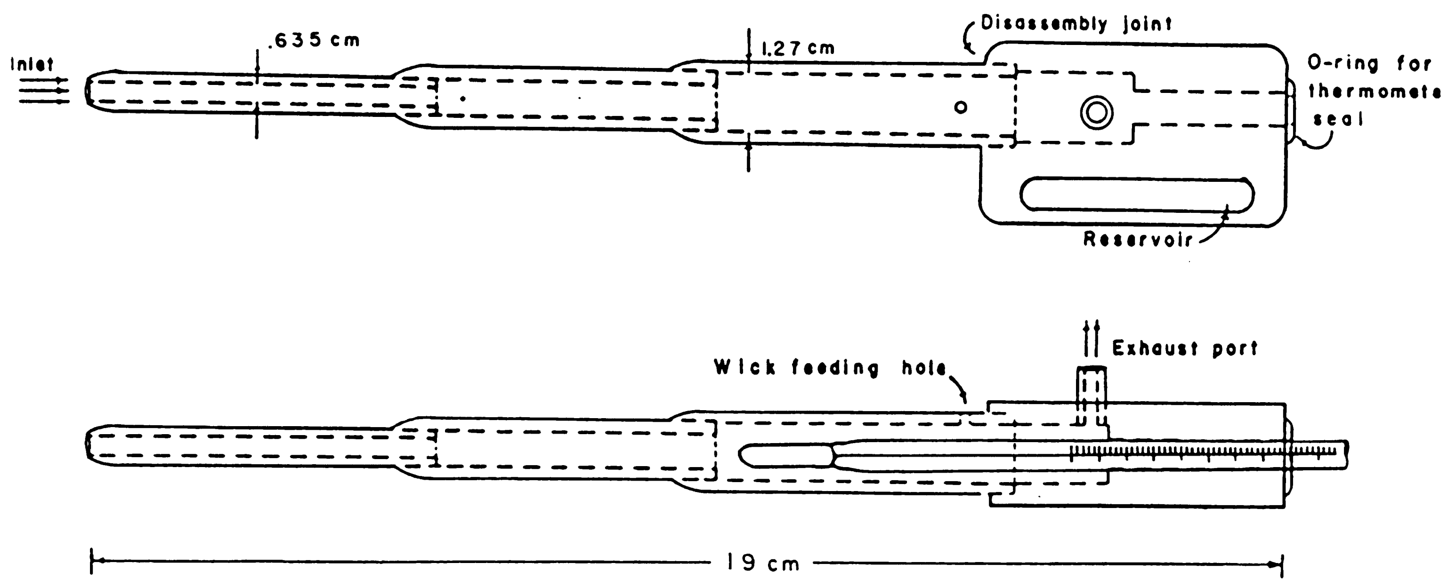

Figure 3.3.2 HUMIDITY PROBE

firm psychrometric foundations (Tanner, 1971) it was calibrated as a check against the gravimetric 'cold coil' method described in Appendix D. The correlation between the two methods was found to be excellent.

In preparation for the experiment, the humidity probe was mounted on the probe support carriage, along with a thermistor, which was attached so as to be positioned just ahead of the probes inlet hole. Then the probes reservoir was filled with distilled water, and the probes exhaust port was attached to a vacuum pump by a length of tubing. Once equilibrium conditions were obtained, as discussed earlier, the instrument carriage was then moved step-by-step to give a profile of dry bulb and wet bulb air temperatures.

To find the absolute humidity from the dry and wet bulb readings, first, the saturation vapor pressure corresponding to the latter was determined using the following equation:

$$e_s = 33.8639\left[(.00738\, T_{wb} + .8072)^8 - .000019(1.8\, T_{wb} + 48) + .001316\right] \quad (6)$$

The actual vapor pressure was then determined using:

$$e = e_s - .0006606\, p_{atm}(T_{db} - T_{wb})(1 + 0.001146\, T_{wb}) \quad (7)$$

Then the absolute humidity was found by applying the ideal gas law and solving for density in grams of water per cubic meter of moist air:

$$\rho = e / (R\ T) \tag{8}$$

where: R is the gas constant for water; and

T is temperature (^{o}K).

From the profiles of absolute humidity for different fetches, values for k_g were calculated as described in the following section.

3.4 Mass Transfer Coefficient Calculations

The gas phase mass transfer coefficient, k_g, is defined by

$$k_g = N\ (\ C_s - C_{10}) \tag{9}$$

where: N is the flux (evaporation rate);

C_s is the surface concentration; and

C_{10} is the concentration at 0.1 meters.

k_g was determined by the horizontal flux method. A detailed description of this method is given below.

3.4.1 Horizontal Flux Method

Evaluation of the gas phase mass transfer coefficient, k_g required the measurement of the mass flux across the air-water interface. In the horizontal flux method this flux is determined by applying a mass balance, i.e., the increase in horizontal mass flux between the upwind edge of the test section and a downwind location is equal to the mass leaving the air-water interface between the upwind edge and that location.

By measuring concentration and velocity profiles at the upwind edge and at the station, the increase in horizontal flux enables the determination of the average mass flux across the air-water interface.

The calculation of the mass flux involved the use of a simplified form of the convection-diffusion equation,

$$U \frac{\partial C}{\partial x} = \frac{\partial}{\partial z}\left(k \frac{\partial C}{\partial z}\right) \qquad (10)$$

where: x is the distance to the upwind edge;

z is the height above the water surface;

U is the local wind velocity, U=U(z);

C is local concentration, C=C(x,z); and

k is the mass transfer coefficient, k=k(z).

The boundary conditions are

$$C \rightarrow 0 \text{ for } z \rightarrow \infty, \quad N=0 \text{ for } x \lesssim 0$$

$$N = \left[k \frac{\partial C}{\partial t} \right]_{z=0} \qquad \text{for } x \gtrsim 0 \tag{11}$$

where N is the average mass flux across the air-water interface.

The solution for the concentrations is obtained following a procedure given by Pasquill (1974). The area source [represented by Equation (11)], is considered as the superposition of an infinite series of line sources. The first step is then to determine the analytical solution for a line source. This is done as follows:

1) The velocity profile is approximated by the power law formula

$$U = U_{10}(z/0.1)^{\alpha} \tag{12}$$

where U_{10} is the velocity at a height of 0.1 m above the water and α is the power law exponent.

2) The profile of the mass transfer coefficient above the water is approximated by

$$k_g = k_{10}(z/0.1)^{1-\alpha} \tag{13}$$

where k_{10} is the mass transfer coefficient at a height of 0.1 m. k_{10} is evaluated using the Reynolds analogy assumption of equating the mass transfer coefficient to the momentum transfer coefficient. Thus

$$k_{10} = Ku*(0.1) \qquad (14)$$

where K is Von Karman constant (=0.4) and u* is the shear velocity.

Equations (10), (11) and the above expressions for k_{10} and U_{10} yield the following closed form solution:

$$C(x,z) = \frac{Nr}{U_{10}\Gamma(s)}\left[\frac{U_{10}}{r^2 k_{10} x}\right]^s \exp\left[\frac{-U_{10} z^r}{r^2 k_{10} x}\right] \qquad (15)$$

where: $r = 1 + 2\alpha$;

$s = (\alpha + 1)/r$; and

Γ is the gamma function.

The solution for an area source is now obtained by integrating along the x-direction from the upwind edge to the point of consideration; i.e.

$$C(x,z) = \int_0^x \frac{Nr}{U_{10}\Gamma(s)}\left[\frac{U_{10}}{r^2 k_{10} x}\right]^s \exp\left[\frac{-U_{10} z^r}{r^2 k_{10} x}\right] dx \qquad (16)$$

The calculation of the mass flux N was carried out as follows. First the solution for C(x,z) was obtained by setting N equal to unity. Denoting this solution by $C_1(x,z)$ and denoting the actual concentration measured experimentally by $C_2(x,z)$, the unknown flux N was determined by relating C_2 to C_1 using linear regression.

The calculation of C_1 required the following parameters: U_{10}, u* and x. U_{10} and u* were obtained graphically by plotting the measured velocity profiles on semi-logarithmic paper. The coordinates x and z were those of the various samplers. The calculation of C_1 using Equation (16), was carried out using Fortran program DIFFUSE presented in Appendix G. The calculation procedure includes the numerical integration of line source solutions. The line sources were equally distributed over the area, except for the section directly in front of the point of consideration which had a denser distribution of line sources. This was done to avoid numerical errors due to discretization.

With a known mass flux N, k_g was determined using Equation (9).

Chapter IV

RESULTS AND DISCUSSION

The objective of this study is to gain information about the relationships between wave characteristics, the air flow over them, and the evaporation rates for various wind/wave conditions.

General experimental conditions set for each individual run are summarized in Table 4-1, regarding fetch, wind speed, and mechanical wave presence. All experiments were conducted with a mean water depth of approximately 21.5 cm ranging from 21.0 to 21.7 cm.

4.1 Velocity Data

Individual velocity profiles are shown in Figures 4.1.1 through 4.1.20. Each profile shows the mean air velocity with distance above the mean water level, (taken as the still-water surface). Fitting a straight line through the logarithmic portion of each profile, the velocity parameters U_{10}, $u*$ and z_o were found. The power law exponent was found by a similar procedure as explained in Section 3.3.1. A summary of reduced velocity data for all runs is presented in Table 4-2.

Table 4-1 Experimental Run Conditions

RUN	FETCH (m)	FAN SETTING	APPROX. WIND SPEED @10cm. (m/s)	MECH. WAVES
1A		300	3.5	yes
2A	14.75	300	3.4	no
3A		600	7.1	no
4A		600	7.0	yes
1B		300	3.5	yes
2B	12.0	300	3.6	no
3B		600	7.1	no
4B		600	7.5	yes
1C		300	3.5	yes
2C	9.0	300	3.7	no
3C		600	7.1	no
4C		600	7.6	yes
1D		300	3.5	yes
2D	7.0	300	3.6	no
3D		600	7.3	no
4D		600	7.6	yes
1E		300	3.7	yes
2E	5.0	300	3.7	no
3E		600	7.4	no
4E		600	7.6	yes

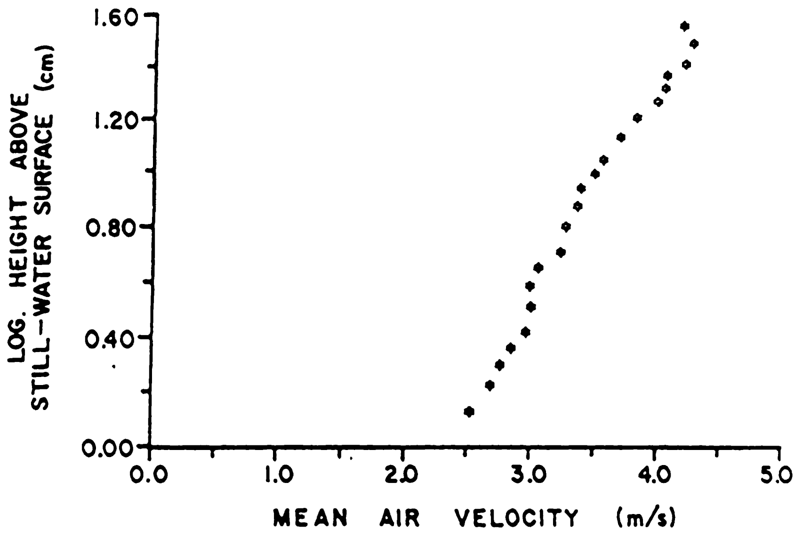

Figure 4.1.1 Velocity profile 1A.

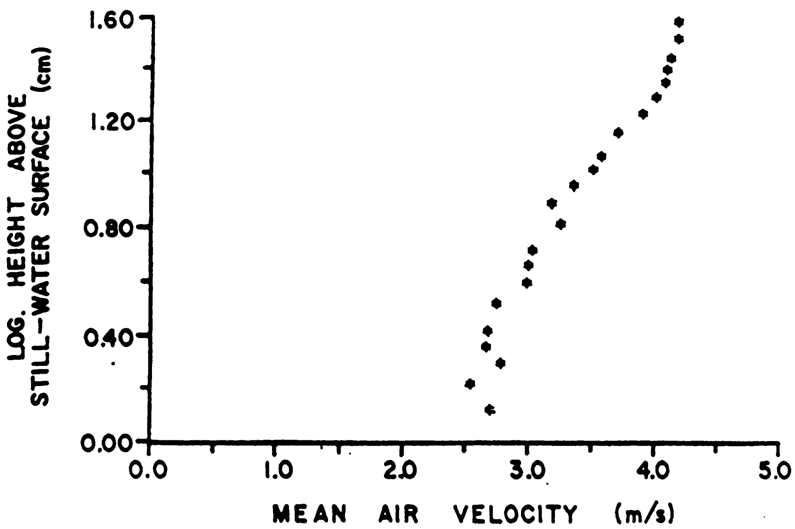

Figure 4.1.2 Velocity profile 2A.

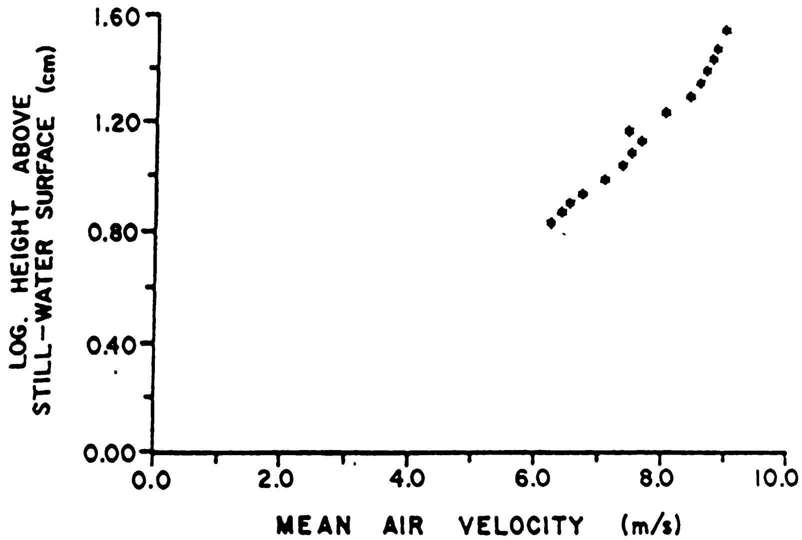

Figure 4.1.4 Velocity profile 4A.

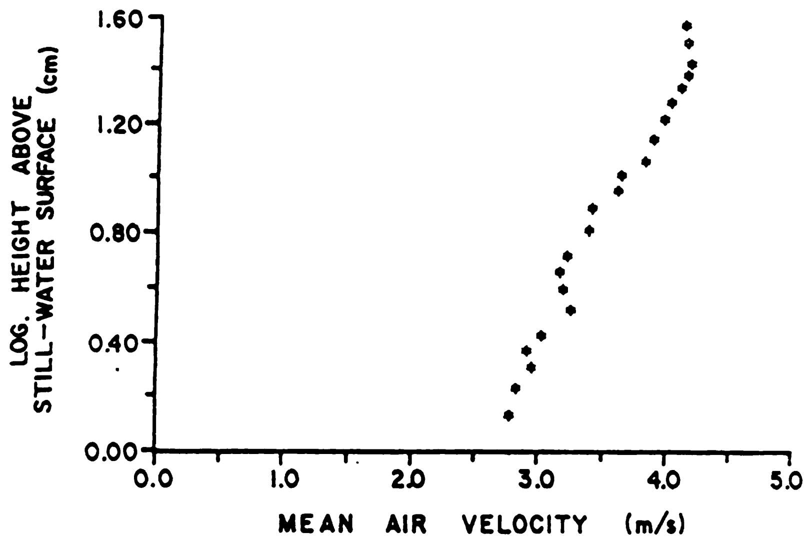

Figure 4.1.5 Velocity profile 1B.

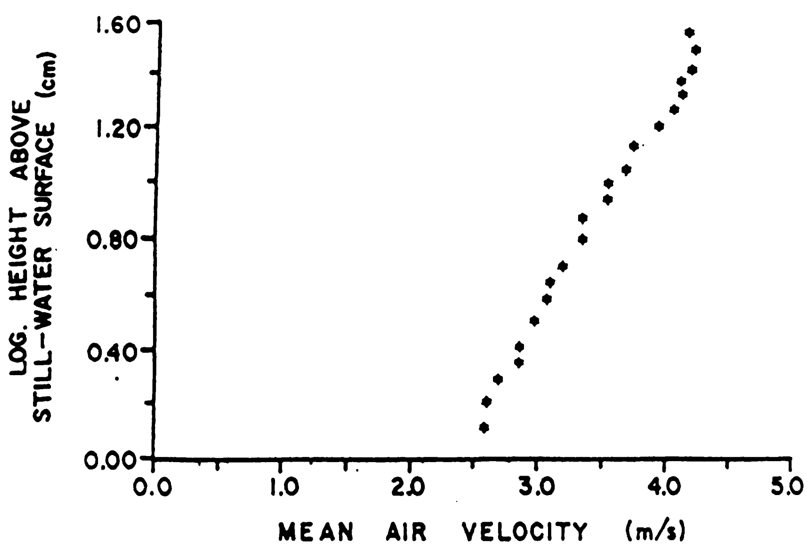

Figure 4.1.6 Velocity profile 2B.

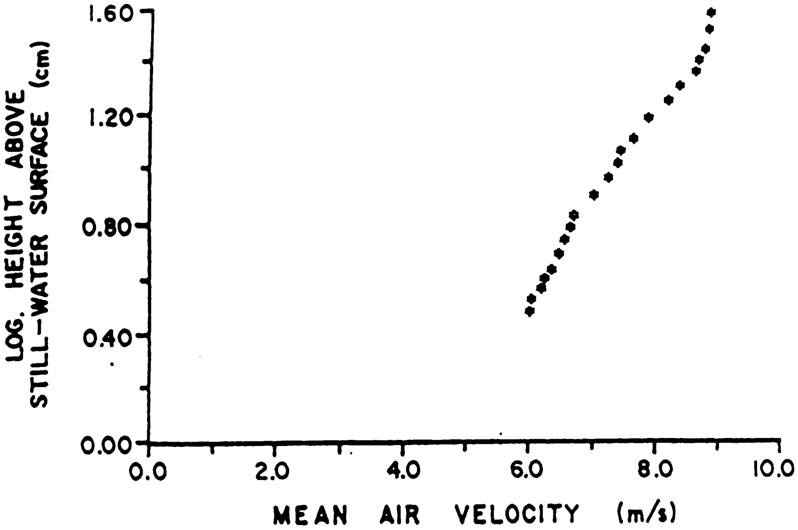

Figure 4.1.7 Velocity profile 3B.

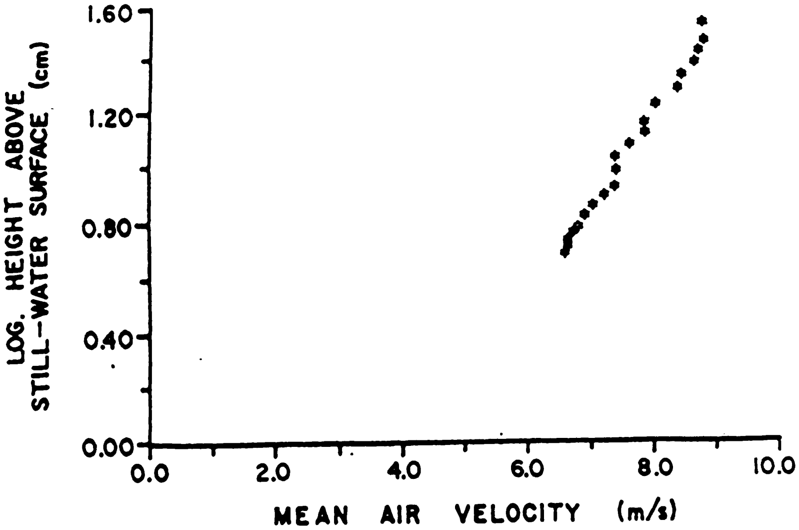

Figure 4.1.8 Velocity profile 4B.

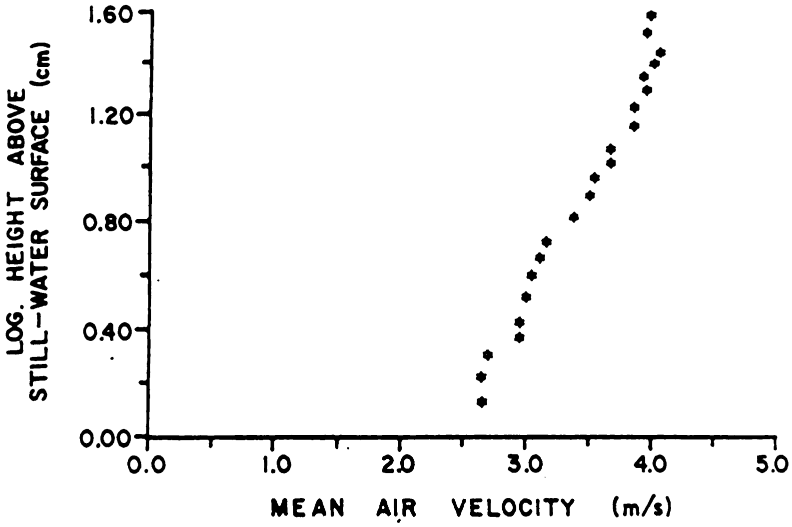

Figure 4.1.9 Velocity profile 1C.

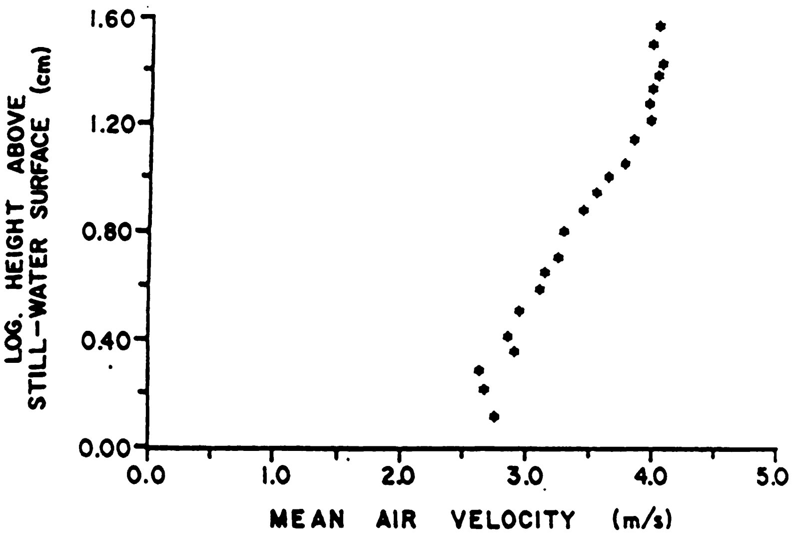

Figure 4.1.10 Velocity profile 2C.

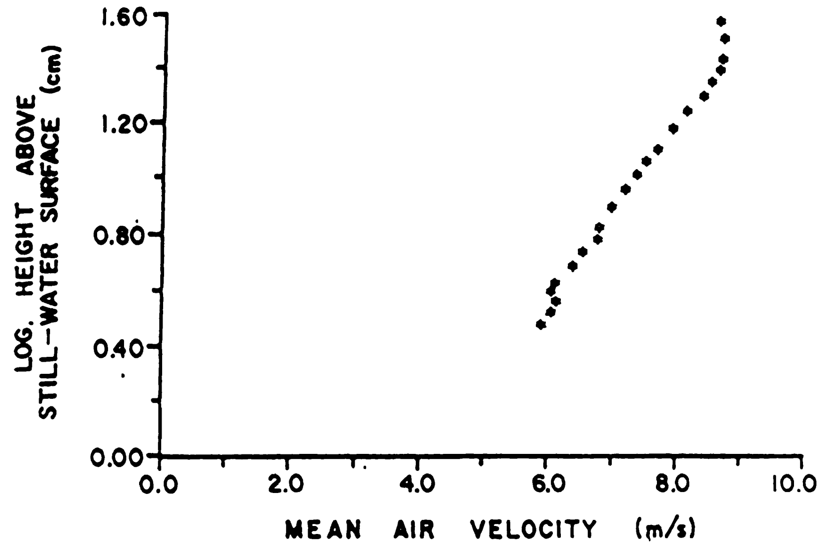

Figure 4.1.11 Velocity profile 3C.

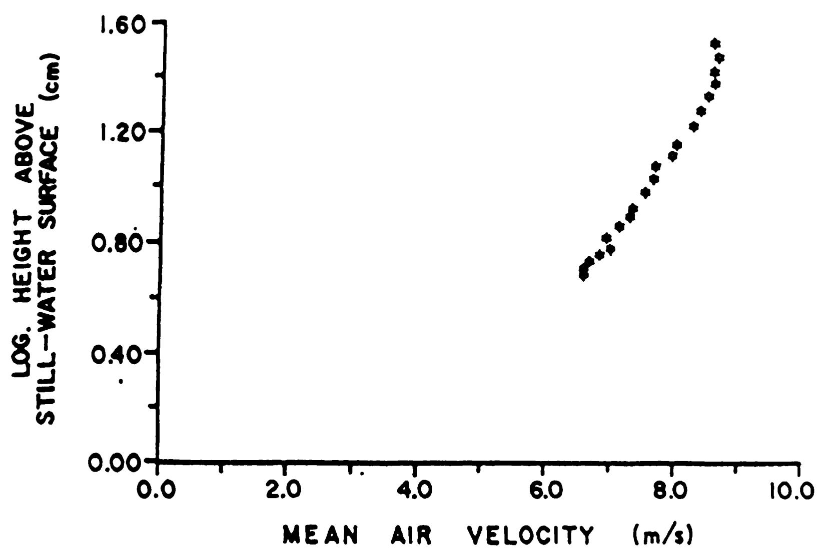

Figure 4.1.12 Velocity profile 4C.

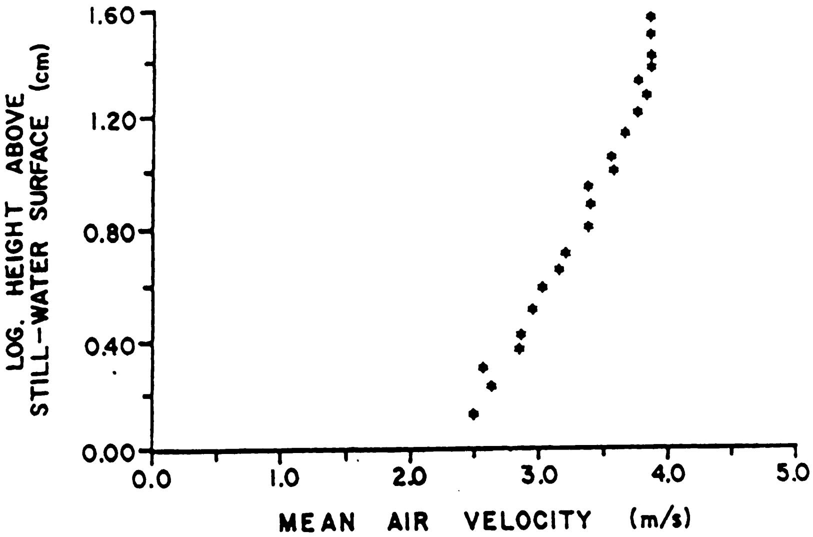

Figure 4.1.13 Velocity profile 1D.

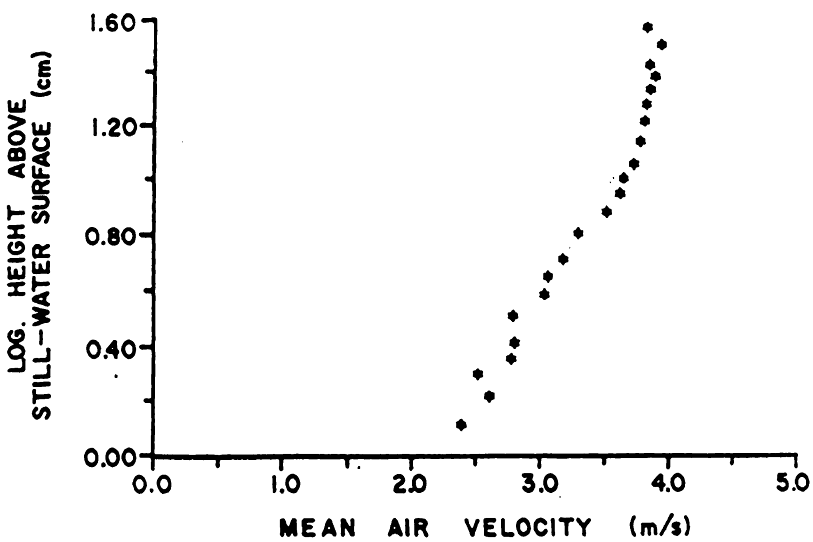

Figure 4.1.14 Velocity profile 2D.

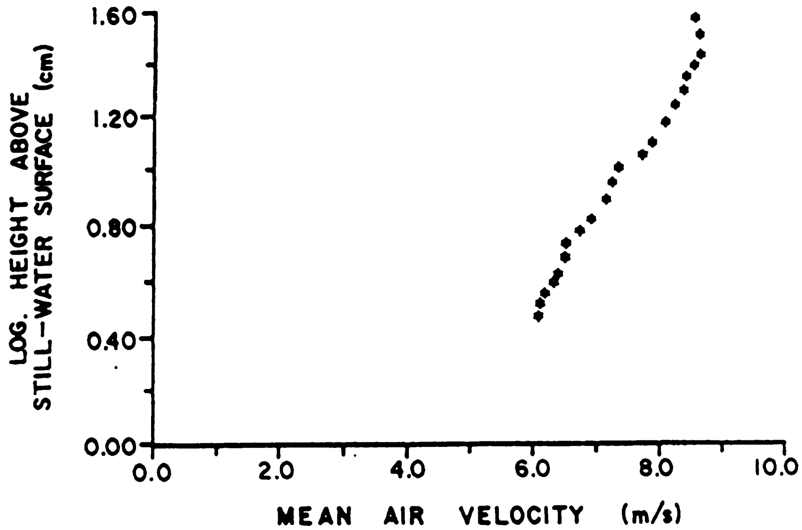

Figure 4.1.15 Velocity profile 3D.

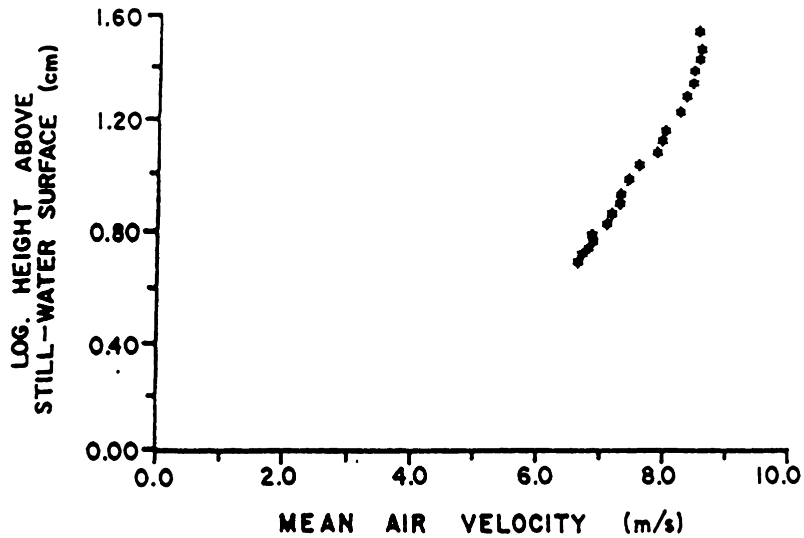

Figure 4.1.16 Velocity profile 4D.

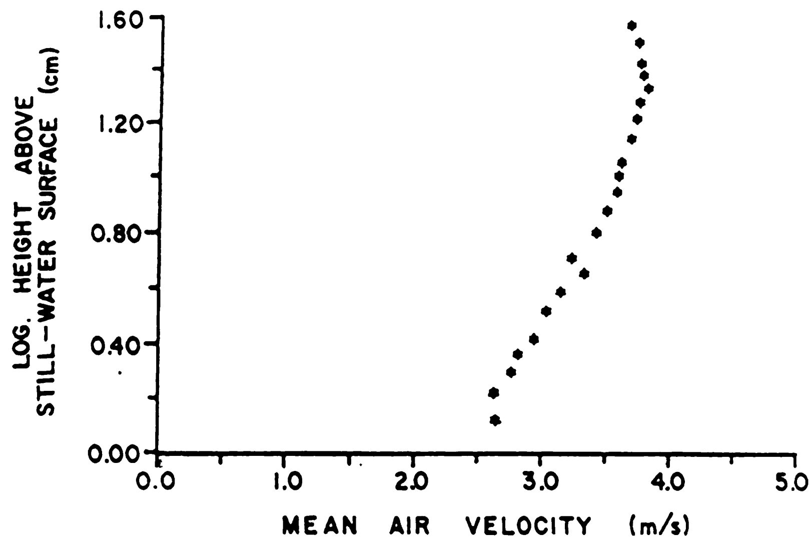

Figure 4.1.17 Velocity profile 1E.

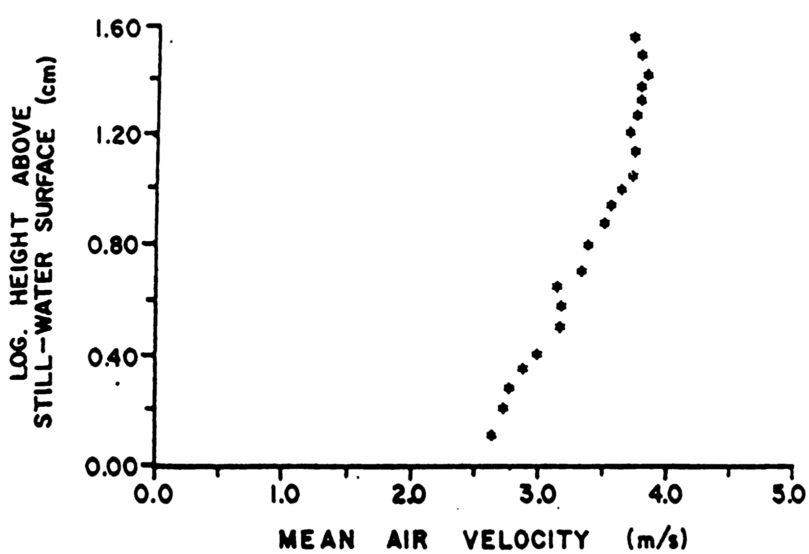

Figure 4.1.18 Velocity profile 2E.

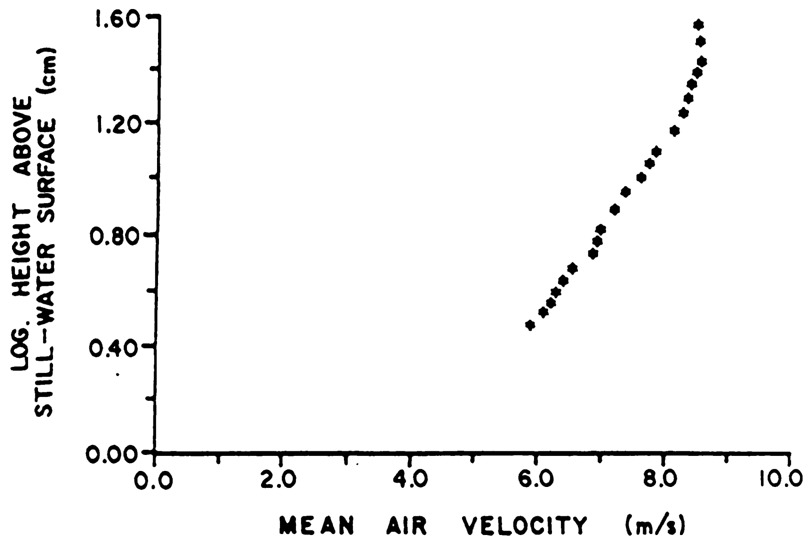

Figure 4.1.19 Velocity profile 3E.

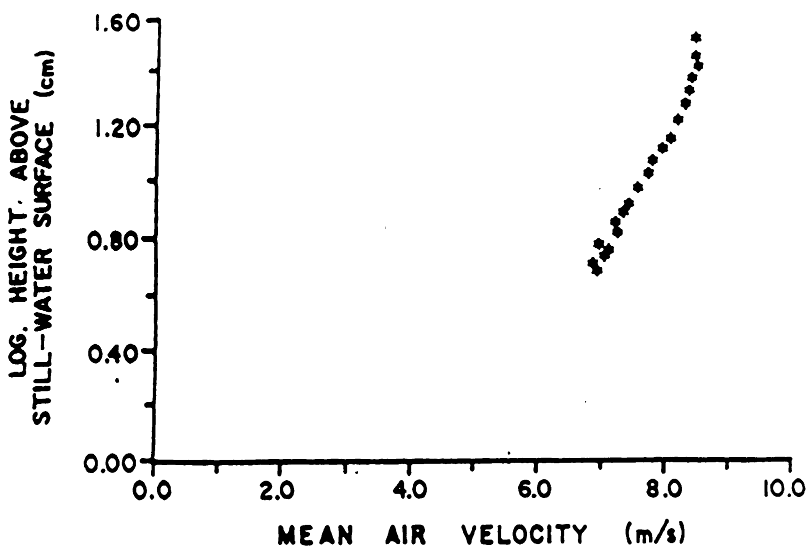

Figure 4.1.20 Velocity profile 4E.

Table 4-2 Velocity Data

Run	Temp. Water	Temp. Air	u^* (m/s)	U_{10} (m/s)	z_o (m)	Alpha	# Points
1A	22.5	24.5	.178	3.49	.00396	0.145	9/14
2A	23.1	24.5	.204	3.41	.01243	0.183	9/13
3A	22.1	24.5	.375	7.05	-	0.17	est.
4A	21.8	24.7	.795	6.95	.03018	0.313	4/8
1B	21.4	23.4	.161	3.54	.00153	0.153	8/13
2B	21.1	24.1	.205	3.55	.00971	0.167	8/14
3B	20.8	23.8	.372	7.09	.00482	0.170	9/15
4B	20.5	24.2	.564	7.49	.04866	0.175	7/13
1C	24.5	24.8	.194	3.52	.00715	0.177	7/13
2C	23.9	24.8	.253	3.68	.0293	0.189	6/12
3C	23.4	25.4	.401	7.12	.00822	0.179	5/13
4C	22.9	25.3	.617	7.59	.0723	0.186	7/12
1D	21.9	24.4	.193	3.50	.0071	0.156	5/14
2D	21.5	24.4	.247	3.60	.0294	0.205	7/15
3D	21.2	24.7	.446	7.31	.0141	0.186	5/14
4D	20.9	24.7	.552	7.61	.0399	0.175	6/13
1E	22.4	24.1	.243	3.73	.0216	0.165	6/14
2E	22.0	24.2	.214	3.68	.0102	0.160	6/14
3E	21.6	24.7	.463	7.38	.0169	0.189	5/13
4E	21.2	24.7	.427	7.63	.0078	0.147	9/14

Note: The # points is in the form x/y, where,

y is the total number of points in the profile
x is the number of points in the log portion of the profile used to estimate the velocity parameters

4.2 Wave Data

Wave data reduction was explained in Section 3.3.2. In the experimental runs in which waves were created mechanically, these waves were generated at a 2 - Hertz frequency and allowed to further develop by the air flow over them. Results with regard to wave rms, significant wave height, wave period, celerity, and the dimensionless parameter of wave height/wavelength are summarized for each run in Table 4-3.

Table 4-3 Wave Data

Run	Fan Setting	Fetch (m)	Mech. Waves	ζ (cm)	H (cm)	T_w (s)	C (m/s)	H/L
1A	300		yes	0.47	1.32	0.50	0.77	0.034
2A	300	14.75	no	0.28	0.79	0.28	0.49	0.058
3A	600		no	0.93	2.63	0.43	0.77	0.079
4A	600		yes	1.24	3.51	0.48	1.01	0.072
1B	300		yes	0.27	0.76	0.50	0.85	0.018
2B	300	12.0	no	0.19	0.53	0.26	0.49	0.042
3B	600		no	0.77	2.18	0.40	0.75	0.073
4B	600		yes	1.09	3.10	0.50	1.00	0.062
1C	300		yes	0.30	0.85	0.50	0.84	0.020
2C	300	9.0	no	0.12	0.34	0.21	0.44	0.037
3C	600		no	0.64	1.82	0.37	0.68	0.072
4C	600		yes	0.69	1.95	0.50	1.00	0.039
1D	300		yes	0.31	0.88	0.50	0.80	0.022
2D	300	7.0	no	0.08	0.23	--	0.43	---
3D	600		no	0.56	1.59	0.31	0.61	0.084
4D	600		yes	0.61	1.73	0.50	0.88	0.039
1E	300		yes	0.33	0.93	0.50	0.77	0.024
2E	300	5.0	no	0.06	0.16	--	0.39	---
3E	600		no	0.44	1.25	0.17	0.55	0.013
4E	600		yes	0.58	1.64	0.50	0.78	0.042

4.3 Evaporation Data

The change in relative humidity with height above the still-water surface was obtained using the humidity probe method described earlier in Section 3.3.3. Profiles are presented on the following pages in Figures 4.3.1 through 4.3.10. The two runs shown per graph represent identical fetch and wind conditions, but with and without mechanically-generated waves.

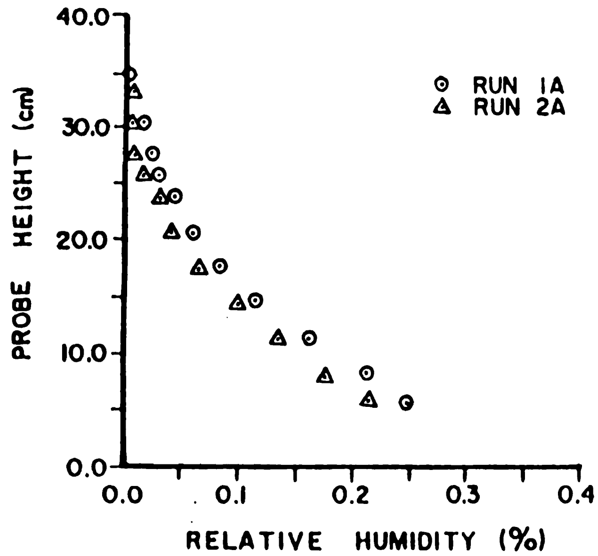

Figure 4.3.1 Humidity profiles 1A and 2A.

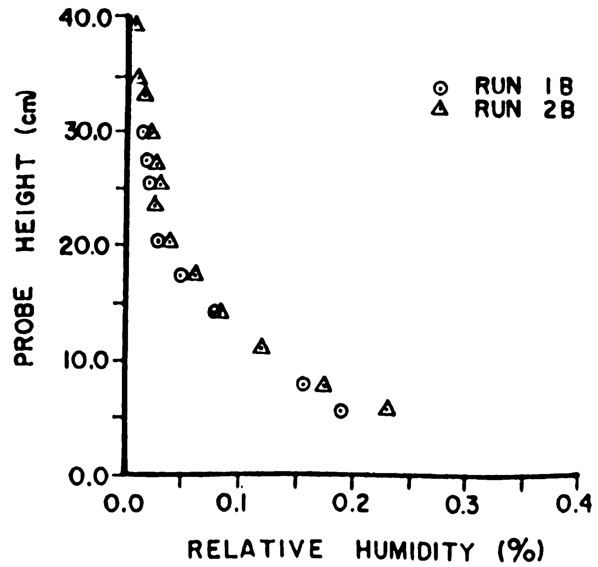

Figure 4.3.3 Humidity profiles 1B and 2B.

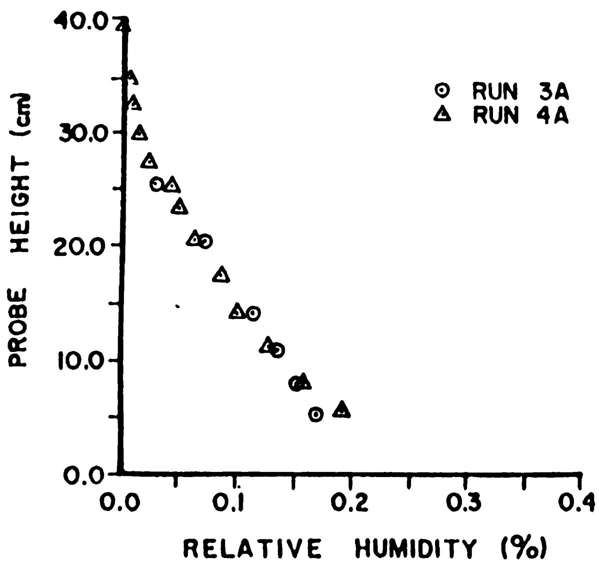

Figure 4.3.2 Humidity profiles 3A and 4A.

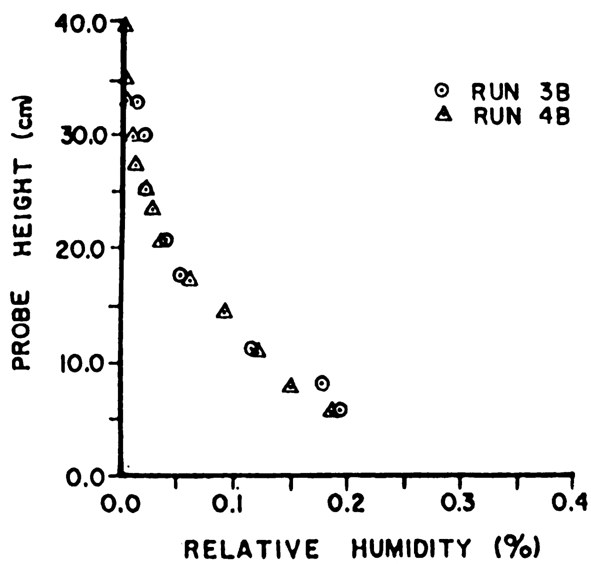

Figure 4.3.4 Humidity profiles 3B and 4B.

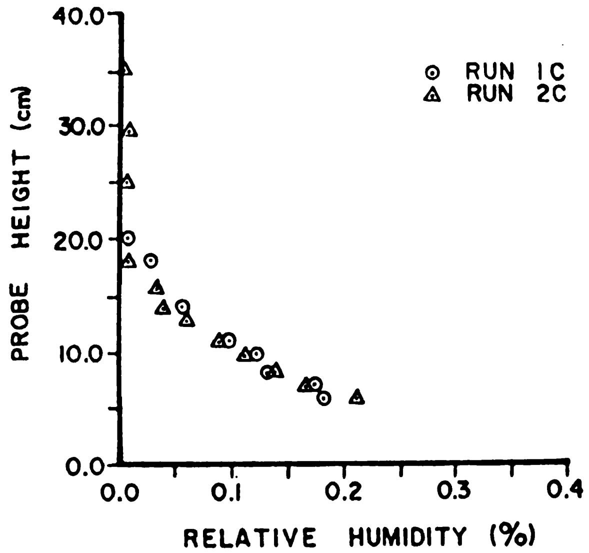

Figure 4.3.5 Humidity profiles 1C and 2C.

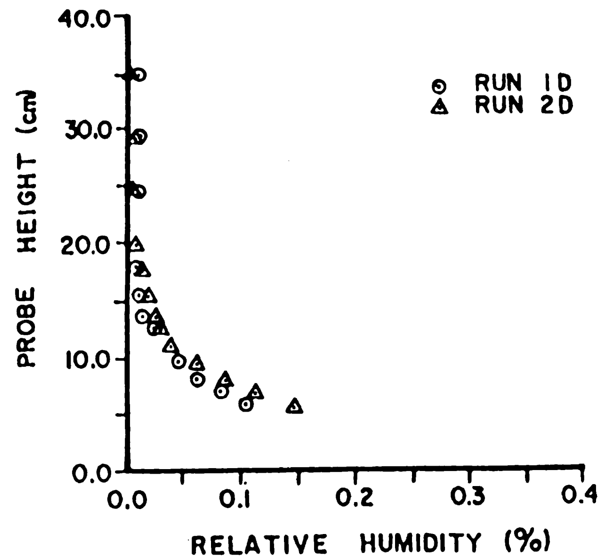

Figure 4.3.7 Humidity profiles 1D and 2D.

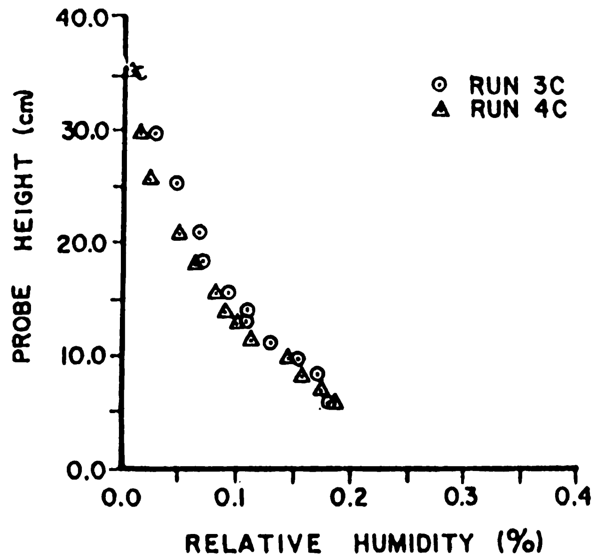

Figure 4.3.6 Humidity profiles 3C and 4C.

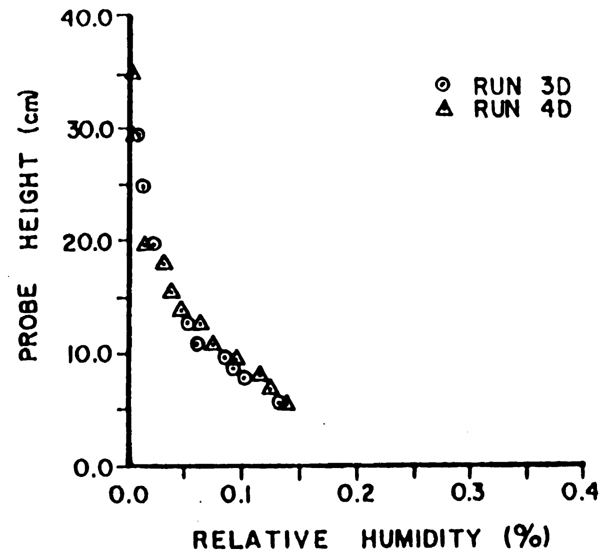

Figure 4.3.8 Humidity profiles 3D and 4D.

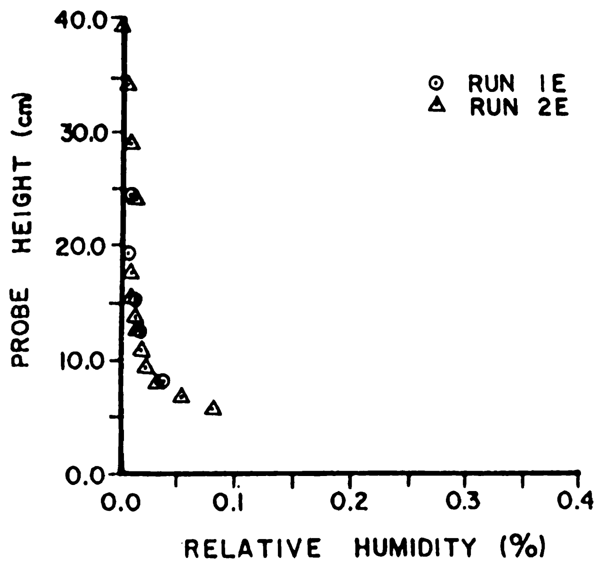

Figure 4.3.9 Humidity profiles 1E and 2E.

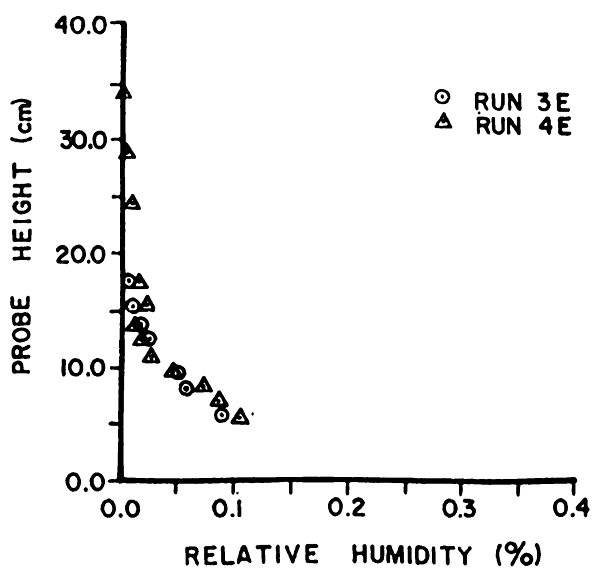

Figure 4.3.10 Humidity profiles 3E and 4E.

4.4 Determination of Evaporation Coefficient

The gas phase mass-transfer (evaporation) coefficient, k_g, was determined as explained earlier in Section 3.4. The values of measured water vapor concentration obtained from the humidity profiles previously presented, were plotted versus an expected water vapor concentration for an emission rate of 1 $g/m^2 \cdot hr$ for the aerodynamic parameters of α , U_{10}, and $u*$ particular to each run. Those plots of measured versus calculated concentrations are shown in Figures 4.4.1 through 4.4.20. Mass flux (evaporation rate) was determined by the slope of the best-fit line. The mass transfer coefficient was the found using Equation (9) presented in Section 3.4. A summary of these values is presented in Table 4-4.

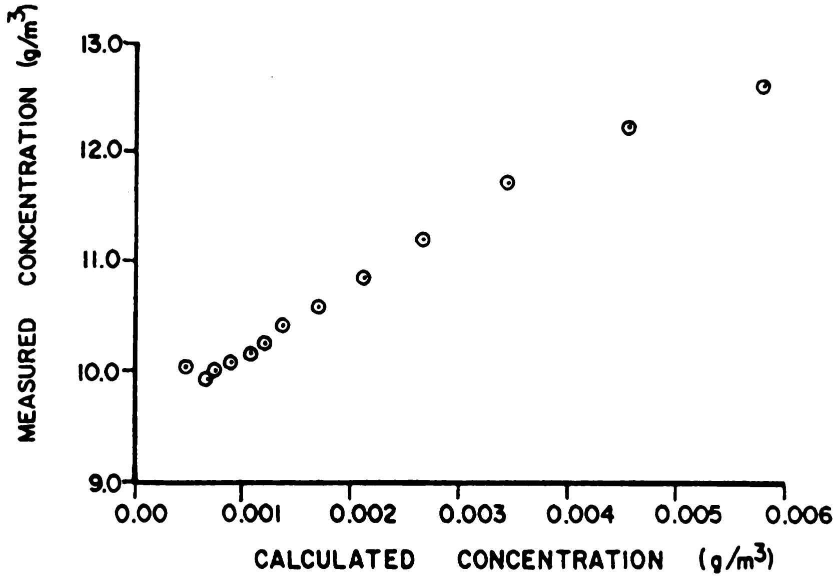

Figure 4.4.1 Measured versus calculated water vapor concentration - 1A.

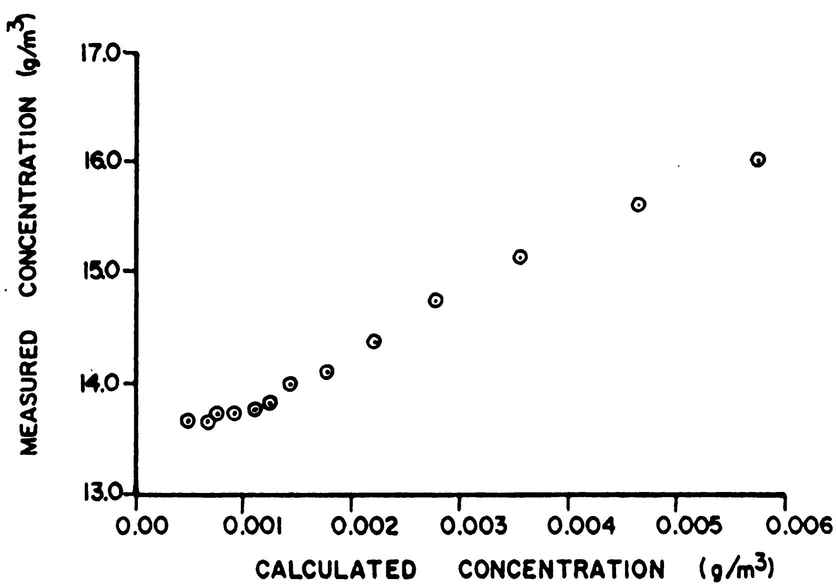

Figure 4.4.2 Measured versus calculated water vapor concentration - 2A.

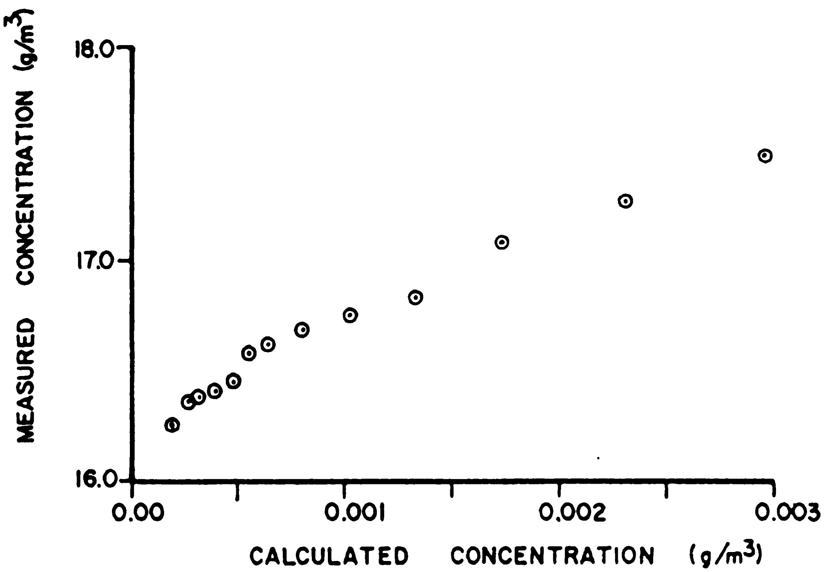

Figure 4.4.3 Measured versus calculated water vapor concentration - 3A.

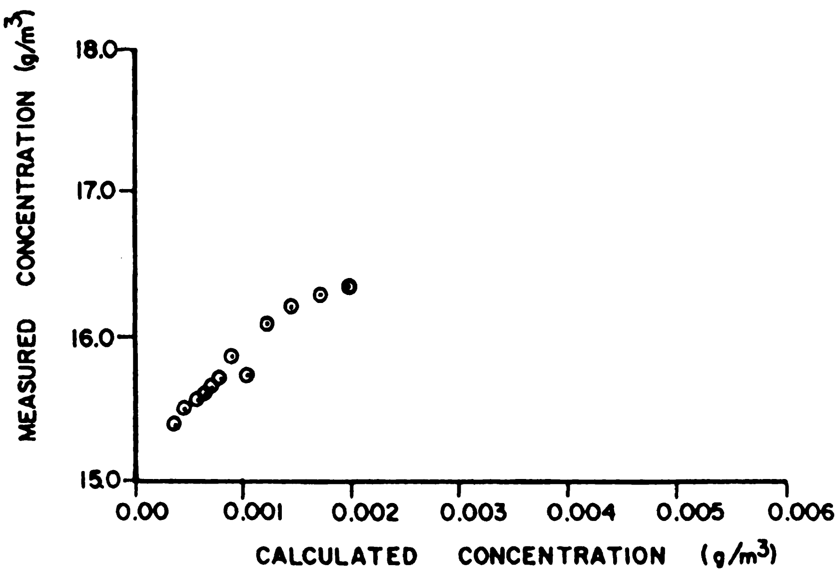

Figure 4.4.4 Measured versus calculated water vapor concentration - 4A.

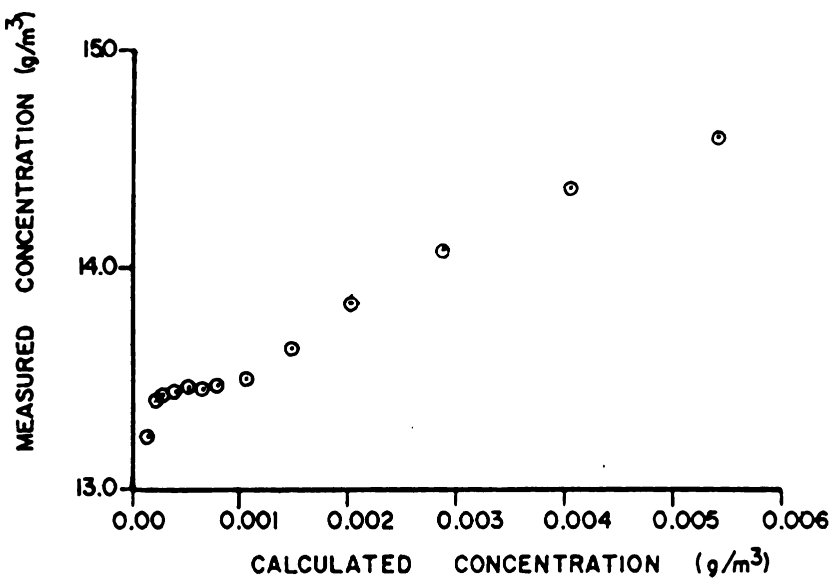

Figure 4.4.5 Measured versus calculated water vapor concentration - 1B.

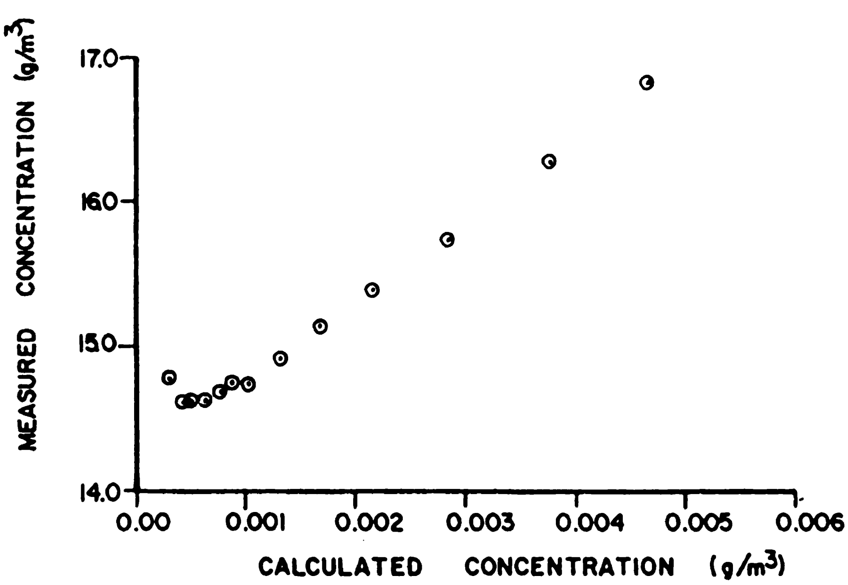

Figure 4.4.6 Measured versus calculated water vapor concentration - 2B.

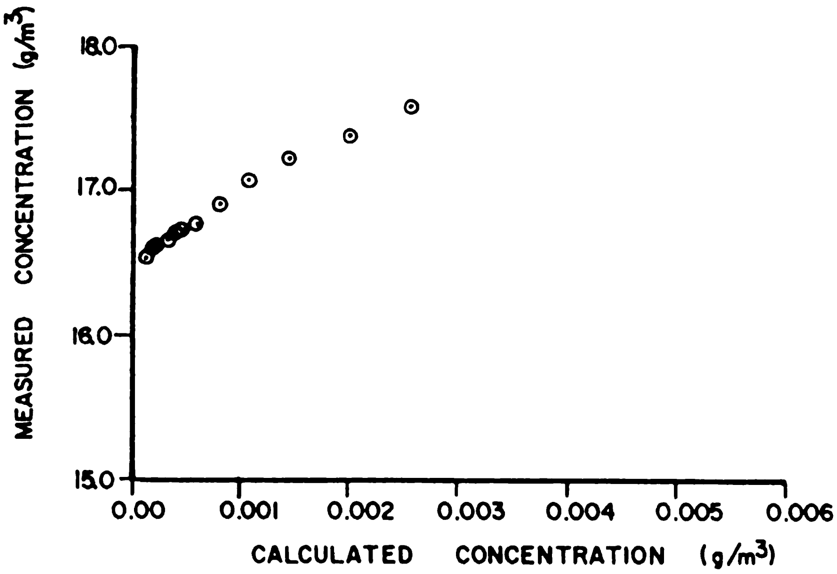

Figure 4.4.7 Measured versus calculated water vapor concentration - 3B.

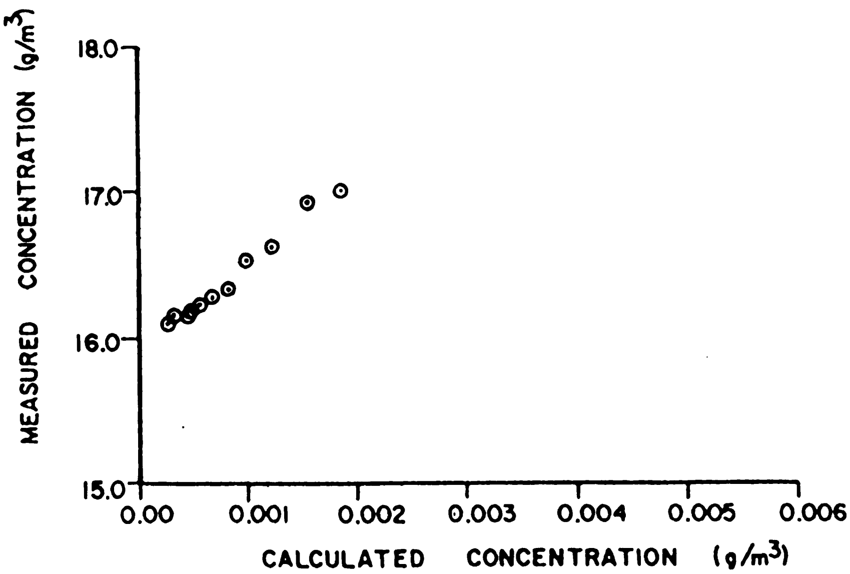

Figure 4.4.8 Measured versus calculated water vapor concentration - 4B.

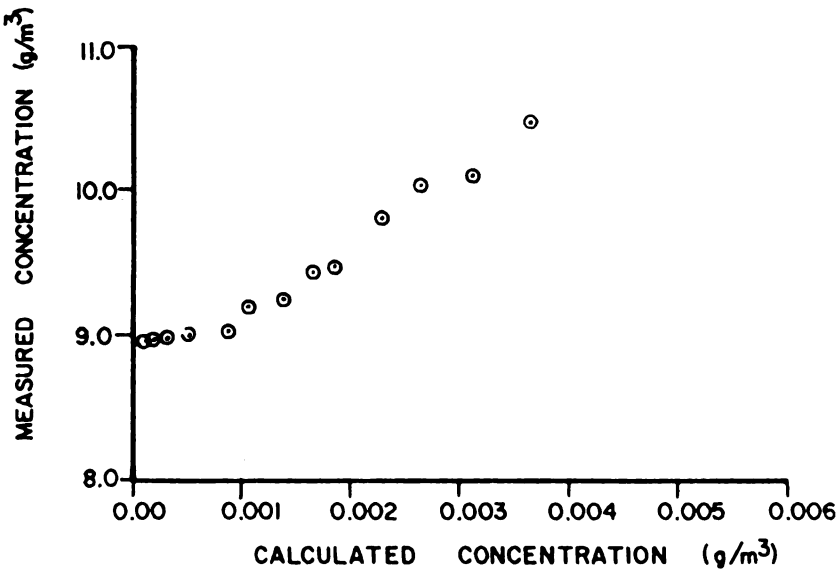

Figure 4.4.9 Measured versus calculated water vapor concentration - 1C.

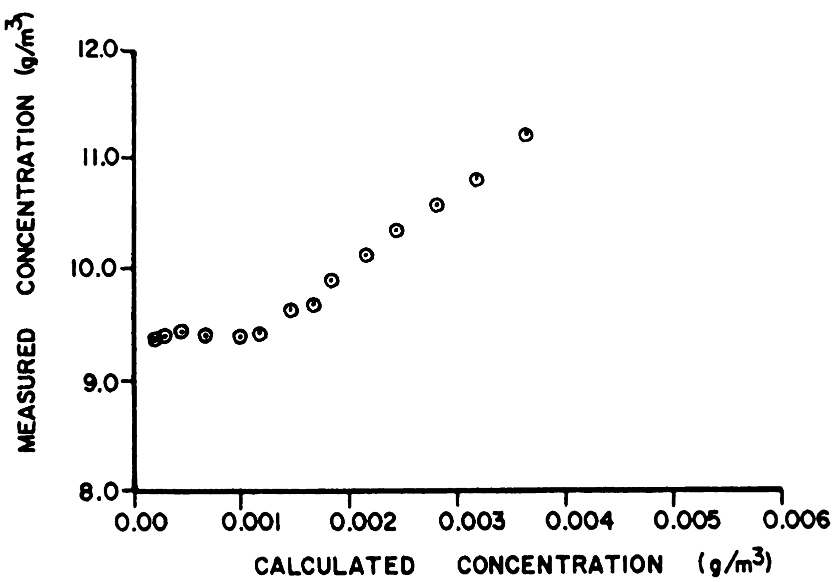

Figure 4.4.10 Measured versus calculated water vapor concentration - 2C.

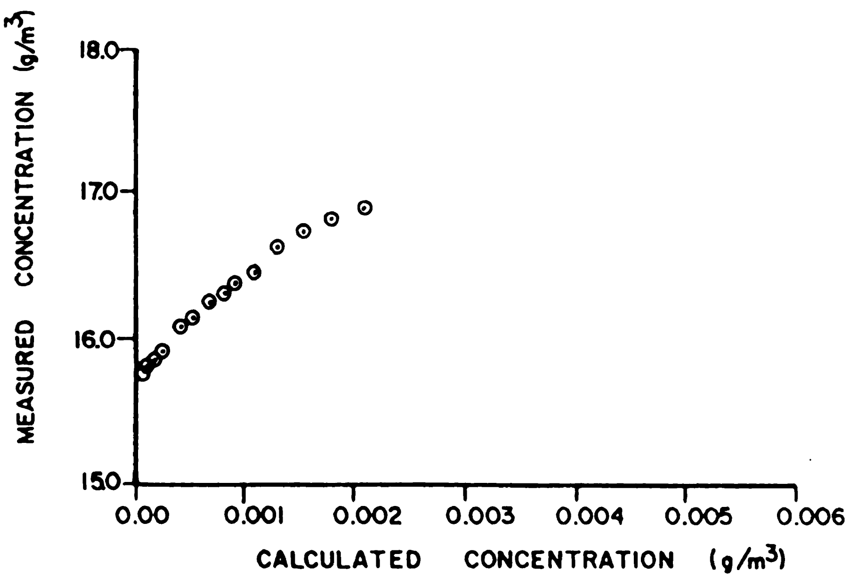

Figure 4.4.11 Measured versus calculated water vapor concentration - 3C.

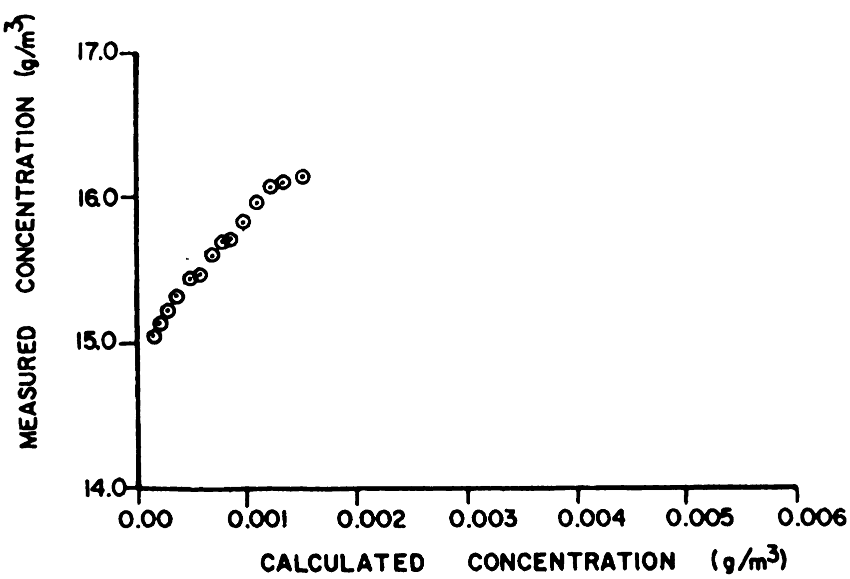

Figure 4.4.12 Measured versus calculated water vapor concentration - 4C.

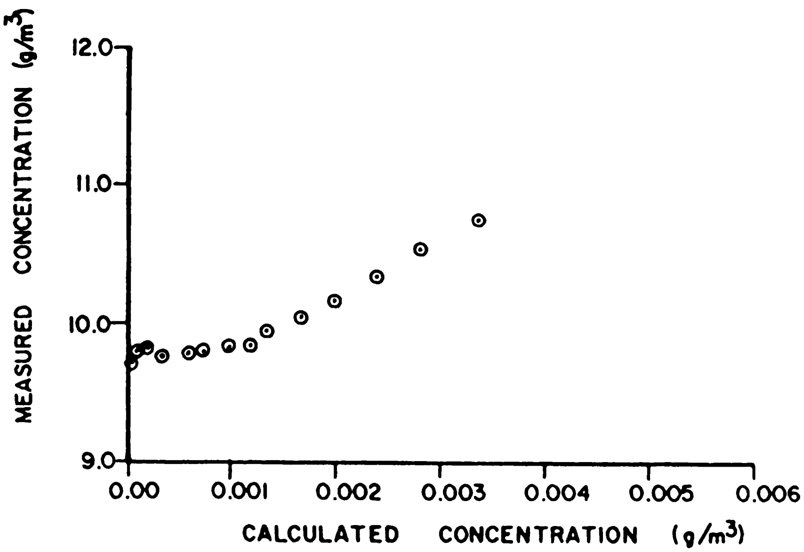

Figure 4.4.13 Measured versus calculated water vapor concentration - 1D.

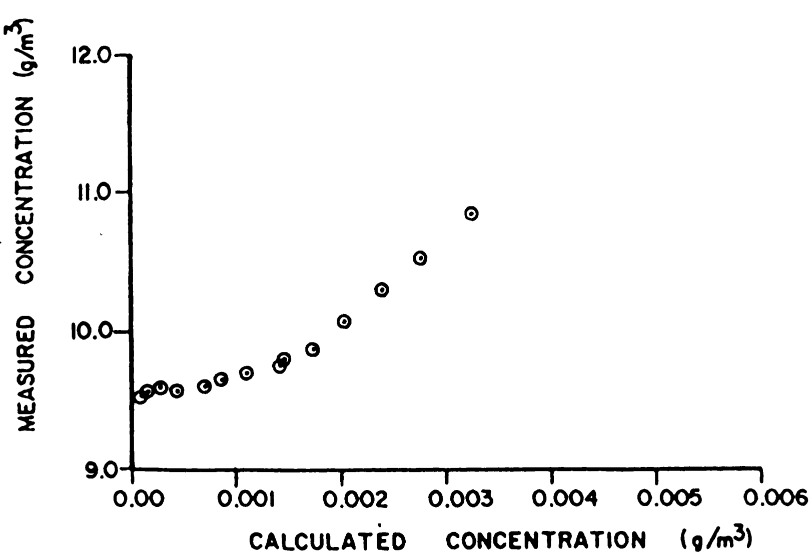

Figure 4.4.14 Measured versus calculated water vapor concentration - 2D.

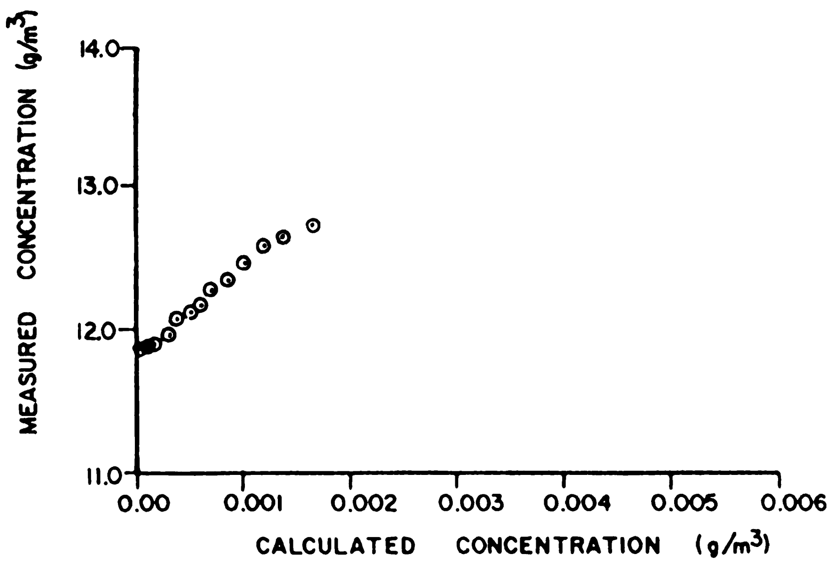

Figure 4.4.15 Measured versus calculated water vapor concentration - 3D.

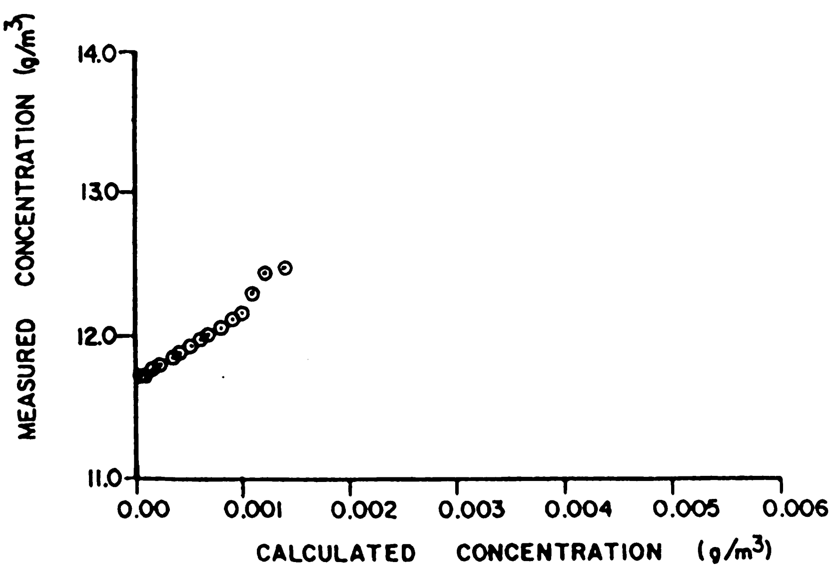

Figure 4.4.16 Measured versus calculated water vapor concentration - 4D.

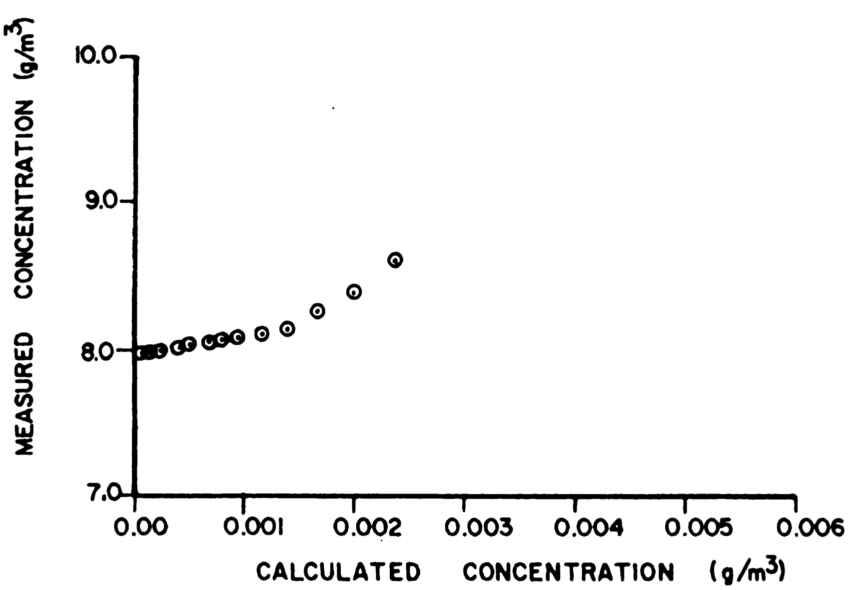

Figure 4.4.17 Measured versus calculated water vapor concentration - 1E.

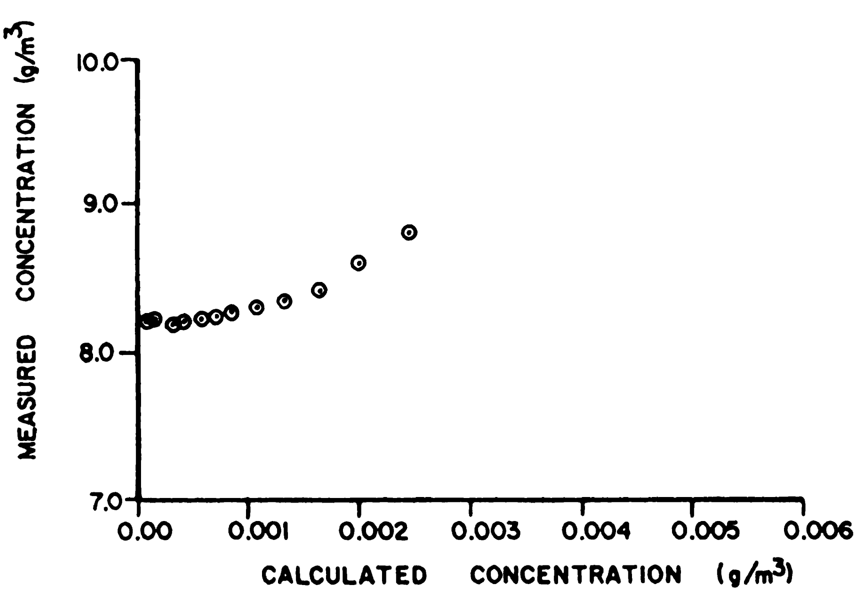

Figure 4.4.18 Measured versus calculated water vapor concentration - 2E.

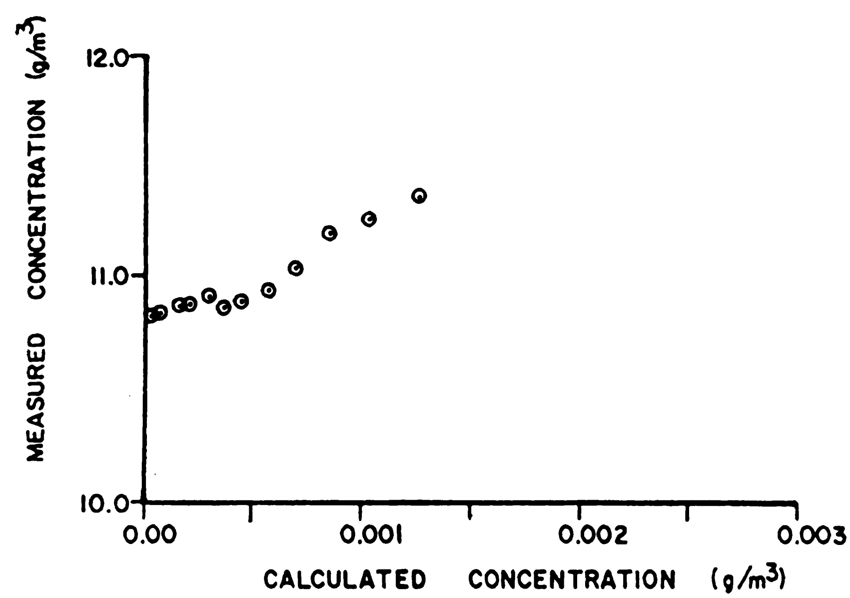

Figure 4.4.19 Measured versus calculated water vapor concentration - 3E.

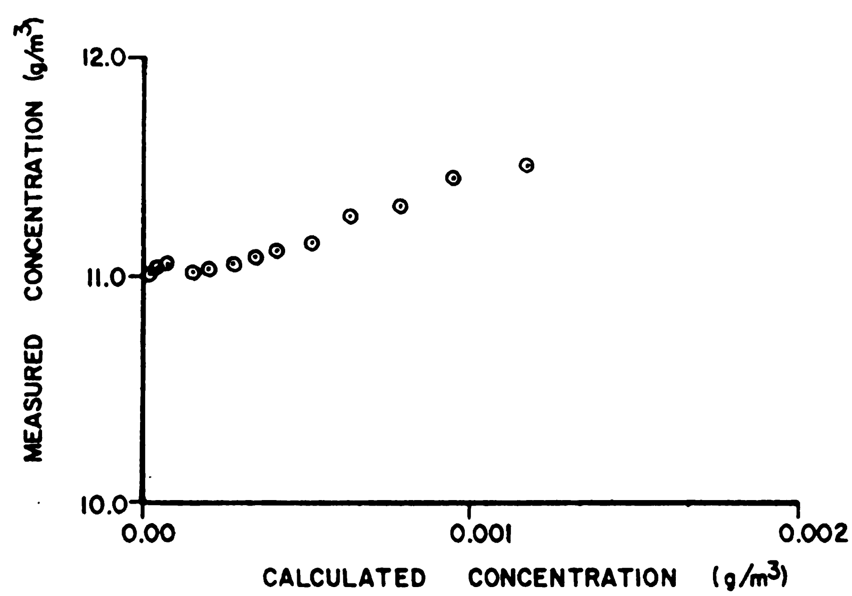

Figure 4.4.20 Measured versus calculated water vapor concentration - 4E.

Table 4-4 Summary of Humidity Data

RUN	Water Vapor Conc. @10 cm. (g/m^3)	Water Surface Conc. (g/m^3)	Measured Background Conc. (g/m^3)	Calculated Background Conc. (g/m^3)	Evap. Rate (g/m^2·hr)	k_g (m/hr)
1A	11.97	20.20	10.06	9.56	548.2	66.6
2A	15.37	24.50	13.65	13.12	458.9	51.4
3A	17.19	22.46	16.27	16.25	490.8	93.1
4A	16.25	20.84	15.40	15.19	681.8	148.4
1B	14.19	20.03	13.25	13.17	323.7	55.5
2B	15.96	23.55	14.78	14.19	411.4	54.2
3B	17.30	22.01	16.55	16.55	428.3	90.9
4B	16.74	20.55	16.10	15.93	565.3	148.4
1C	9.98	17.70	8.95	8.60	419.6	54.3
2C	10.33	18.11	9.39	8.61	515.8	66.0
3C	16.61	21.38	15.78	15.77	662.6	138.8
4C	15.91	20.20	15.05	14.97	882.9	205.8
1D	10.15	19.28	9.73	9.39	295.2	32.2
2D	10.03	18.40	9.52	8.76	415.3	49.6
3D	12.45	17.70	11.91	11.80	666.6	127.0
4D	12.12	17.25	11.72	11.69	474.6	92.5
1E	8.13	15.66	7.95	7.50	302.0	40.1
2E	8.34	15.72	8.16	7.61	322.5	43.8
3E	11.02	15.93	10.81	10.41	470.	95.
4E	11.24	16.46	11.00	10.93	515.0	98.6

Note: Water surface concentration was calculated using water temperature.

Table 4-5 Summary of Experimental Data

RUN	FAN SETTING	FETCH (m)	MECH. WAVES	H/L	u^*/C	k_g/u^*
1A	300		yes	0.034	0.23	0.104
2A	300	14.75	no	0.058	0.42	0.070
3A	600		no	0.079	0.49	0.069
4A	600		yes	0.072	0.79	0.052
1B	300		yes	0.018	0.19	0.096
2B	300	12.0	no	0.042	0.42	0.073
3B	600		no	0.073	0.50	0.068
4B	600		yes	0.062	0.56	0.073
1C	300		yes	0.020	0.23	0.078
2C	300	9.0	no	0.037	0.58	0.073
3C	600		no	0.072	0.59	0.096
4C	600		yes	0.039	0.62	0.093
1D	300		yes	0.022	0.24	0.046
2D	300	7.0	no	---	0.57	0.056
3D	600		no	0.084	0.73	0.079
4D	600		yes	0.039	0.63	0.047
1E	300		yes	0.024	0.32	0.046
2E	300	5.0	no	---	0.55	0.057
3E	600		no	0.013	0.84	0.057
4E	600		yes	0.042	0.55	0.064

4.5 Discussion of Experimental Data

The mass transfer coefficient k_g and the shear velocity u* for each run is plotted in Figure 4.5.1. From this figure it appears that the mass transfer coefficient generally increases linearly with increasing shear velocity. A linear regression yielded the following equation:

$$k_g = 7 + 226\, u^* \qquad (17)$$

where k_g is in m/hr and u* is in m/s. This equation and the 95% confidence belts are shown in Figure 4.5.1. Little variation is observed in either quantity between the low-velocity runs (Run 1 and Run 2, respectively). This is not the case, however, between Runs 3 and 4 (high wind speeds with and without mechanical waves). For these two runs, shear velocity is greater when mechanically-generated waves are present and, correspondingly, the mass transfer coefficient is larger also. It might be expected that the shear velocity is higher for larger waves due to larger exposed areas and greater wave heights or, possibly, due to flow separation.

Figure 4.5.1 reveals no obvious trend due to fetch for each chosen wind/wave condition. Looking more closely at the effect of fetch, Figure 4.5.2 shows the variation of shear velocity with fetch for all data.

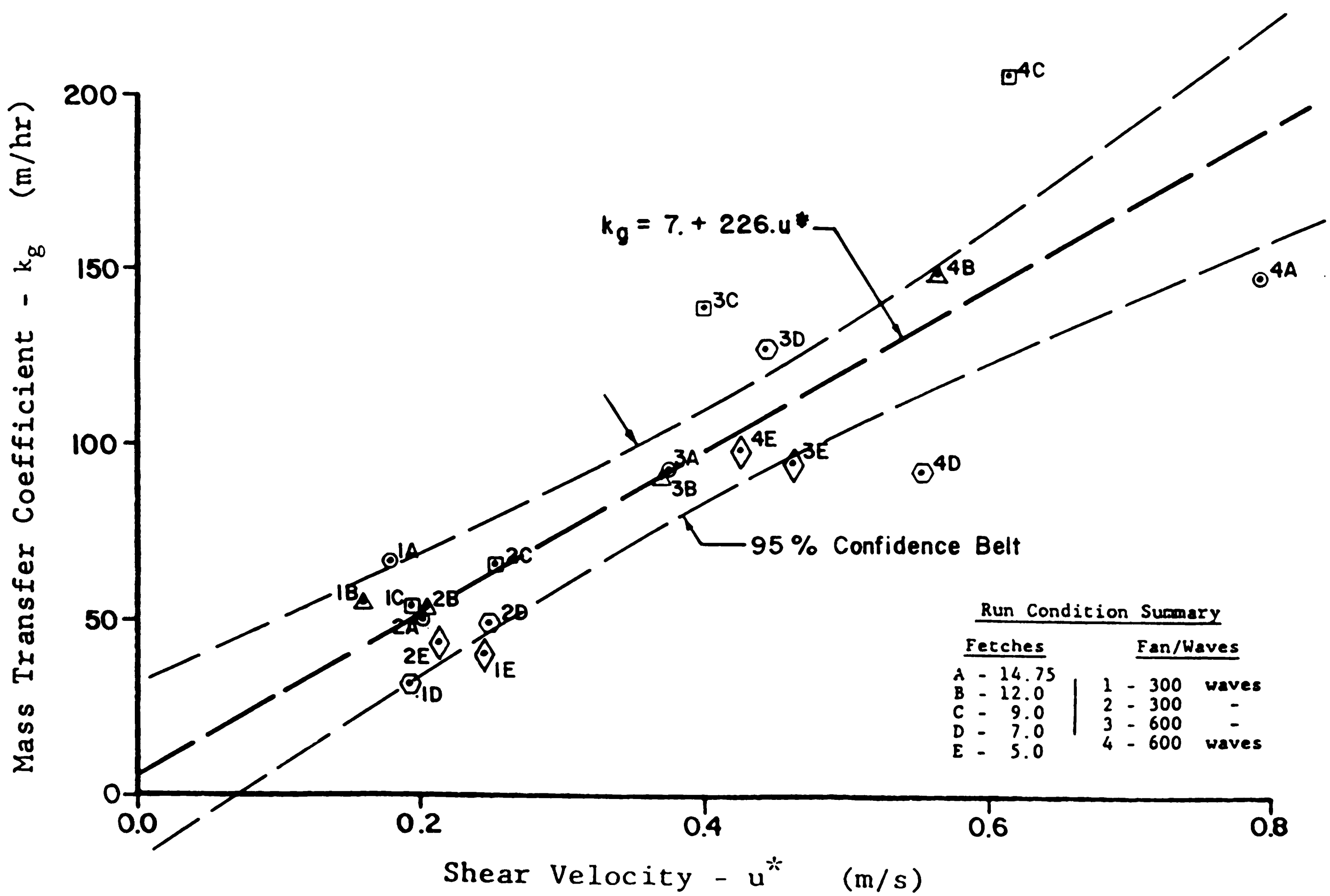

Figure 4.5.1 Mass Transfer Coefficient -vs- Shear Velocity - All Data

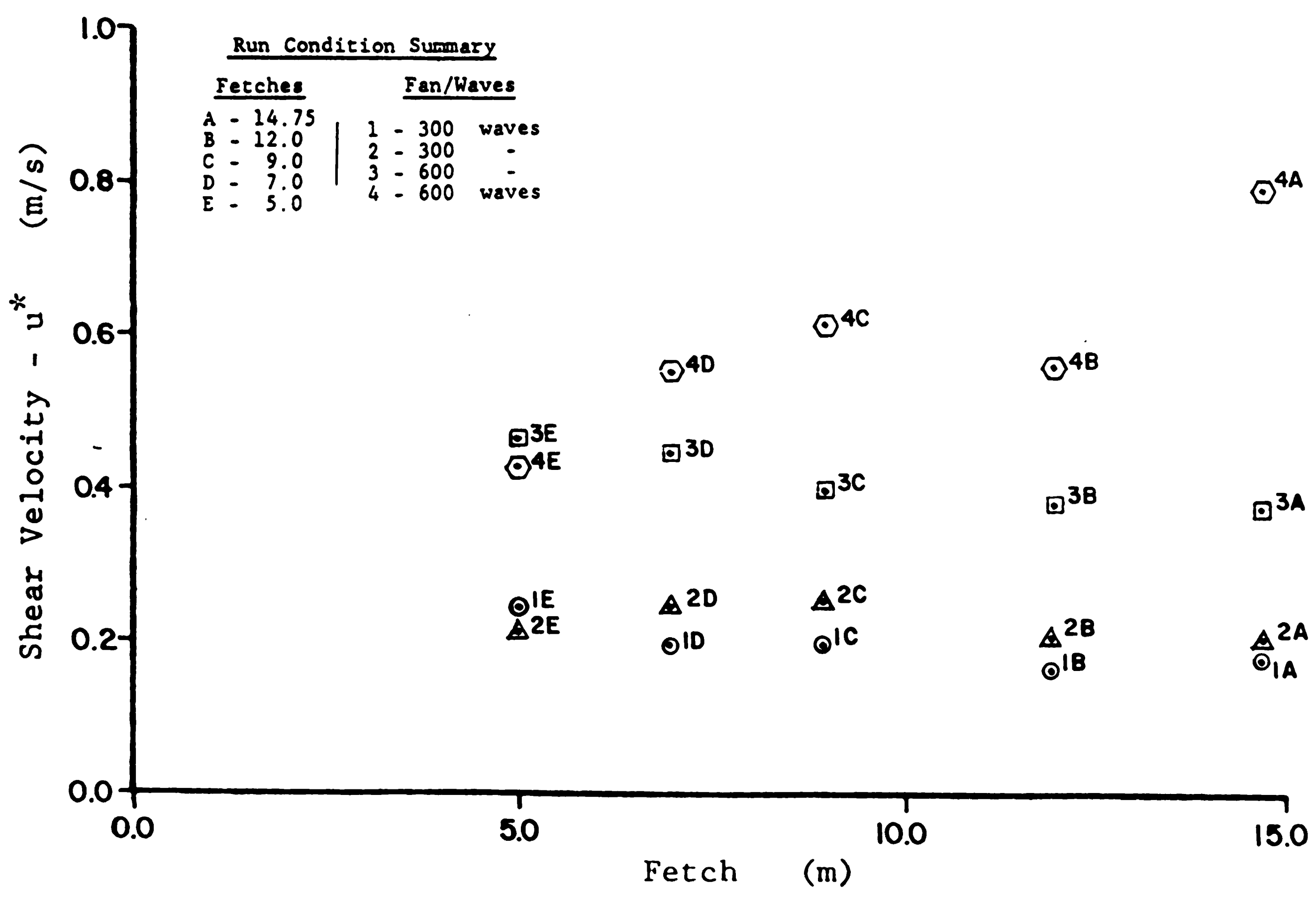

Figure 4.5.2 Variation of u^* with Fetch - All Data

For Runs 1,2, and 3 it appears that fetch has little effect on the shear velocity; perhaps shear velocity decreases very slightly with longer fetches. Run 4, (high wind speed with mechanical waves,) indicates an opposite trend with regard to shear velocity.

A plot of mass transfer coefficient versus fetch is given in Figure 4.5.3. There is considerable variation in k_g with fetch. However, there is no significant correlation with fetch which is similar to what was observed for u*. A comparison between Figures 4.5.2 and 4.5.3 shows the same trends for the different wind and wave conditions indicating a strong correlation between k_g and u*. Figure 4.5.4 shows the fetch- averaged values of k_g versus u*. More clearly than in Figure 4.5.1 we see a linear dependency between u* and k_g. A regression analysis yielded the following relation:

$$k_g = 4 + 234\ u^* \qquad (18)$$

where u* is in m/s and k_g is in m/hr.

The lack of correlation between both the shear velocity and fetch and the mass transfer coefficient and fetch implies that changes of wave steepness in the channel do not significantly affect the mass transfer process. The effect of the mechanically-generated waves for higher wind speed appears significant at larger

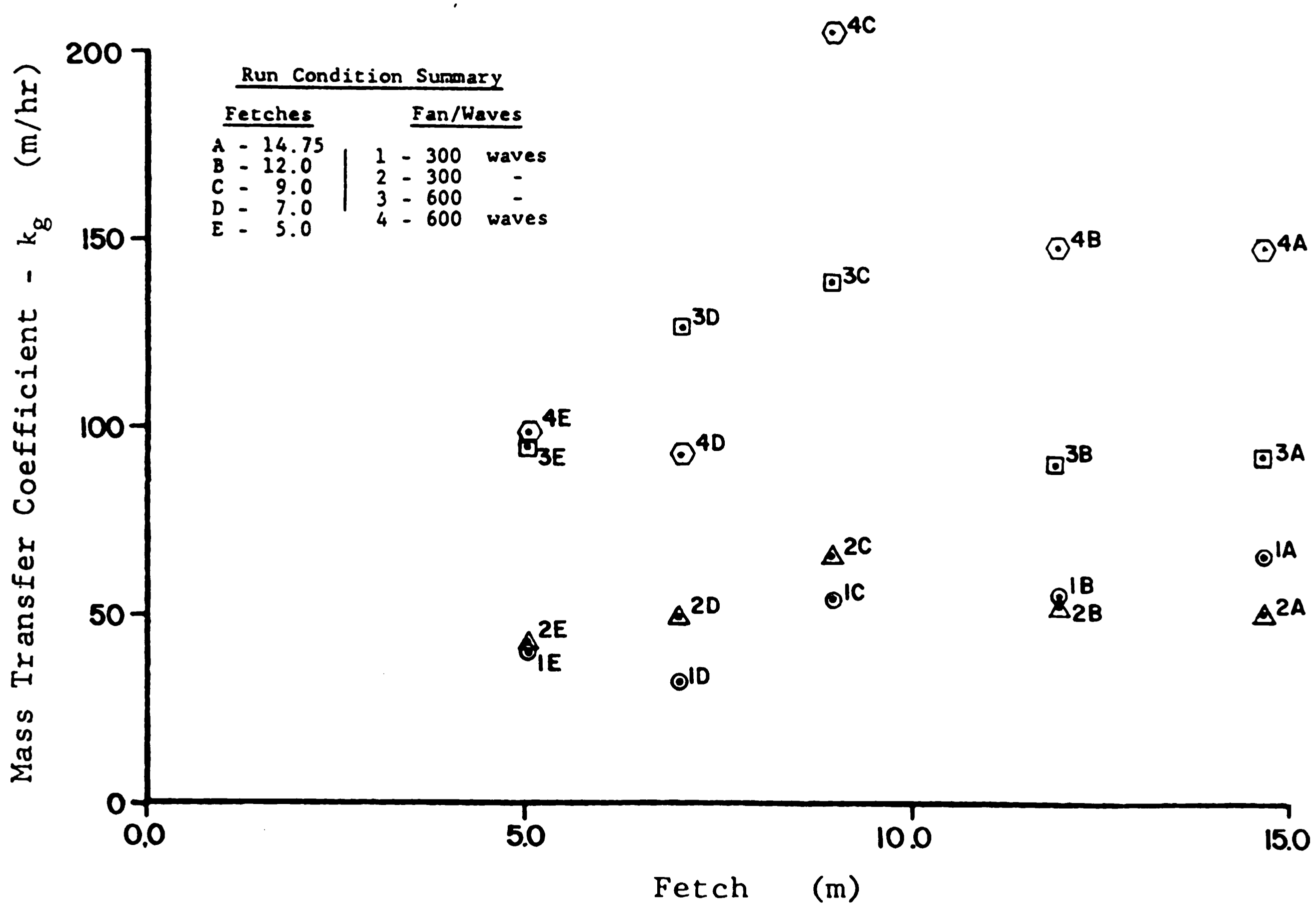

Figure 4.5.3 Variation of Mass Transfer with Fetch - All Data

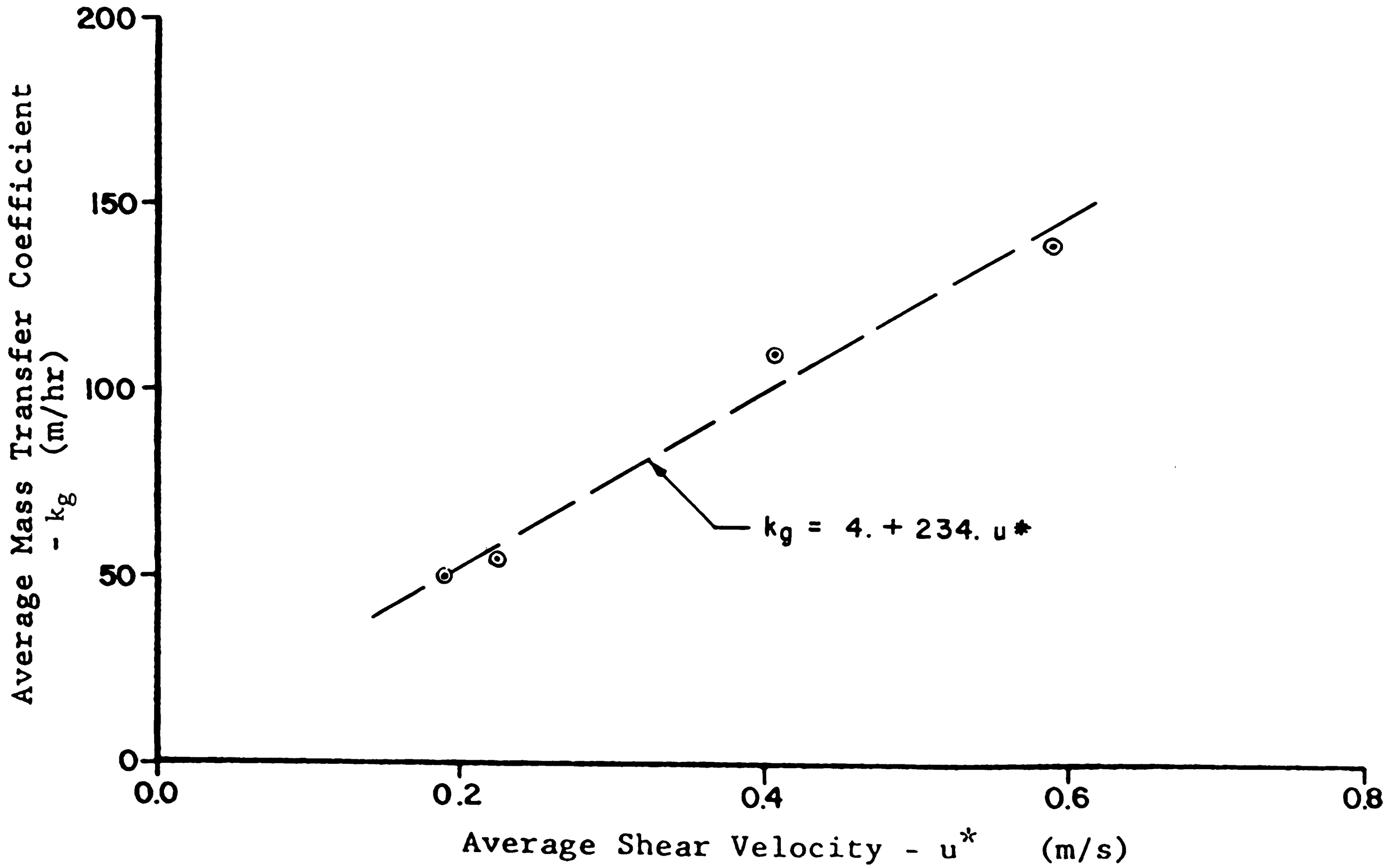

Figure 4.5.4 Fetch-Averaged Shear Velocity versus Fetch-Averaged Mass Transfer Coefficient

fetches (see Figures 4.5.2 and 4.5.3); both k_g and u* increase by about 25 percent when mechanically-generated waves are superimposed. It is possible that this is a result of wave breaking and the flow separation associated with it. Unfortunately no direct evidence exists because no flow visualization studies could be carried out with the available equipment.

4.6 Comparison with Field Data

Marciano et. al. (1954) developed the following equation to predict evaporation rates from lakes:

$$E = 6.25x10^{-4}\ U_8(P_s - P_8) \qquad (19)$$

where: E is the evaporation rate in cm/3hrs,

U_8 is wind speed in knots at 8m,

P_s is the saturation vapor pressure in mb, and

P_8 is the water vapor pressure at 8 m. in mb.

To relate Equation (28) to the wind tunnel data, we follow Sirovica (1982) by making the following assumptions:

1. The surface shear stress u* is used to relate the wind conditions in the wind tunnel to those in the field;
2. The humidity and velocity profiles are logarithmic;
3. The relation between u* and the surface roughness length z_o is given by the dimensionless quantity proposed by Charnock (1955):

$$z_o/(u*^2/g) = 0.0156 \qquad (20)$$

Using these assumptions the following equation for k_g is found:

$$k_g = 5.207\ U_8\ \ln(8/z_o)/\ln(1/z_o) \qquad (21)$$

where z_o is related to u* by equation (20) and U_8 is related to u* by

$$U_8 = 2.5\ u^* \ln(5031/u^{*2}) \tag{22}$$

Equation (21) is plotted in Figure 4.6.1. The fetch-averaged data presented earlier in Figure 4.5.4 show that the wind-tunnel data compare very well with the field equation.

Others, Sirovica (1982) and Easterbrook (1968), observed a consistently lower mass transfer coefficient in their laboratory tests than those found under equivalent field conditions (using the shear velocity as the criterion).

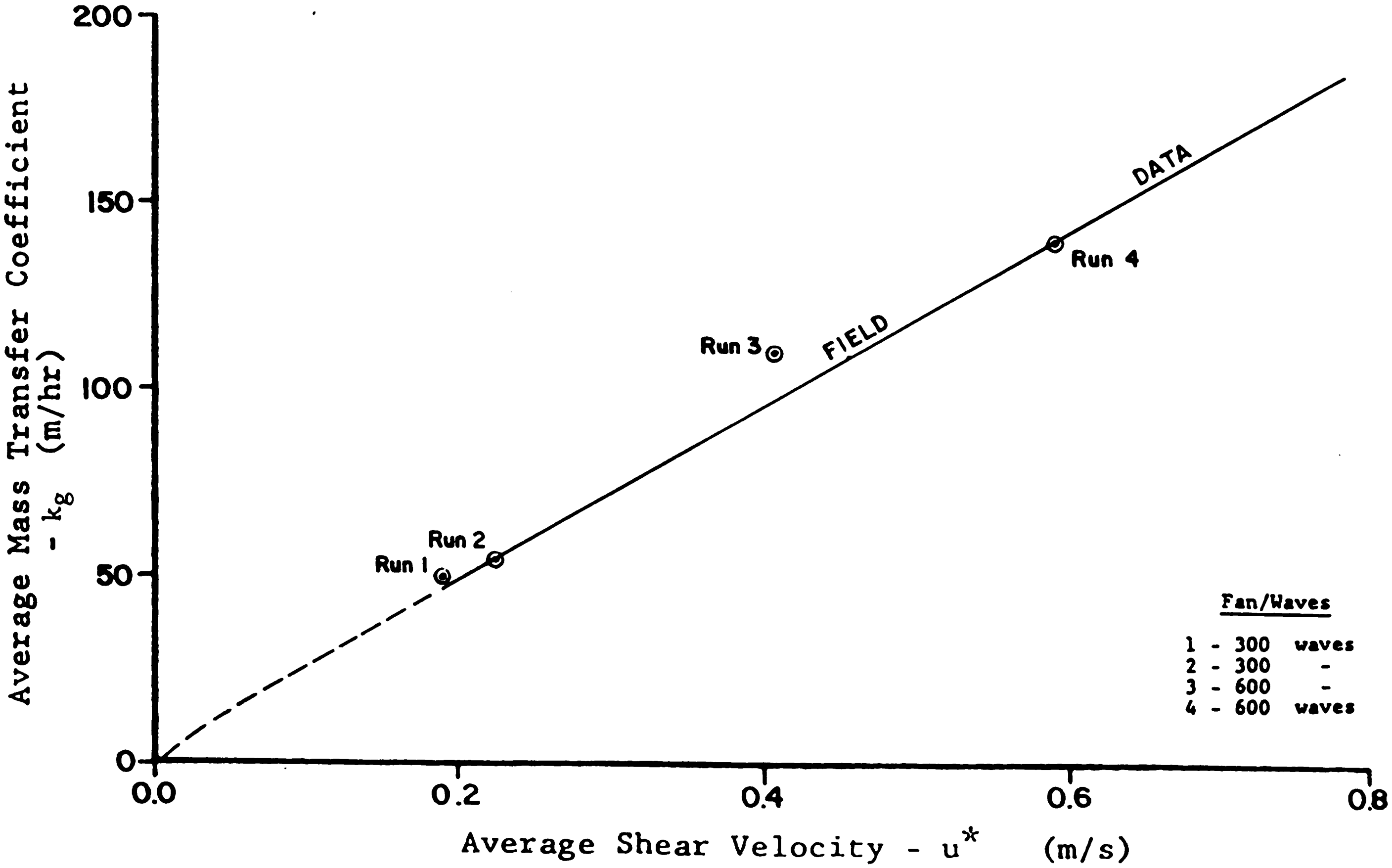

Figure 4.6.1 Average k_g -vs- Average u* for each Run - Comparison to Field Data

Chapter V

CONCLUSIONS AND RECOMMENDATIONS

The experimental results presented and discussed support the following conclusions:

1. The relation between the mass transfer coefficient (in m/hr) and the shear velocity (in m/s) under wind tunnel conditions is found to be:

$$k_g = 7 + 226\ u* \tag{17}$$

2. Both the mass transfer coefficient and shear velocity increased at the higher wind speed due to mechanically-generated waves. The increase was about 25 percent at the higher wind speed. At the lower wind speed there was no significant increase. Although no direct evidence exists that flow separation occurred, the data suggest that flow separation occurred at the higher wind speed and that this phenomena led to higher mass transfer coefficients and shear velocities.
3. No significant variation of shear velocity and mass transfer coefficient with fetch was observed.
4. Fetch-averaged values of mass transfer coefficient compared favorably with an equation developed for lake conditions (Marciano 1959).

The experimental errors combined with relatively few data points causes difficulty in drawing clear conclusions about evaporation rate and mass transfer. Therefore, additional similar experiments should be carried out for a range of wind conditions.

To further examine the possibility of flow separation, or changing air flow patterns, more data at higher wind speeds should be obtained. This combined with flow visualization studies may lead to a better understanding and assessment of the mass transfer process directly above the water surface. In such experiments care should be taken to generate waves that depict conditions that occur naturally with regard to both frequency and wave height.

APPENDICES

Appendix A

INSTRUMENTATION

Velocity Measurements:

- United Sensor Pitot Static Tube
- Datametrics Pressure Transducer, Model 590D
- Dwyer Differential Micromanometer
- Superior Electric Co. Slo-syn translator module, Model STM101
- Superior Electric Co. Slo-syn stepping motor, Model MO61-FD02

Humidity and Temperature Measurements:

- YSI thermistor, Model 702A
- YSI thermivol signal conditioner, Model 740A
- Sartorious top loading electronic balance Model 602
- Orifice, non-commercial, diameter 1/64"
- Aluminum coils, non-commercial, 1/8" I.D., 12' uncoiled length, coiled to 2 1/4" diameter.

Wave Measurements:

- Capacitance wave gage, non-commercial, original design by Fluid Dynamics and Diffusion Laboratory at Colorado State University
- Capacitance bridge, non-commercial
- Tektronix dual trace storage oscilloscope, Model D13, with two dual trace amplifiers, Models 5A18N, and one time base amplifier, Model 5B10N

Many of the signals measured with the instruments described previously were transmitted to a microcomputer which stored and processed the data. The system consists of a Digital Equipment Corporation LSI-11/2 microcomputer with 64-Kb core memory, an ADV11-A A/D converter, and a KWV11-C Real-Time Clock. Mass storage was provided by a dual-drive, double density, floppy disk subsystem. The software described in Appendices E and F were used to sample data, calculate mean values, RMS, probability distributions, etc.

Appendix B

VELOCITY CALIBRATION

The pressure transducer which was used to measure the pressure difference between the stagnation and static pressure holes of the Pitot tube, was calibrated prior to velocity measurements. The calibration setup consisted of a micromanometer, a large syringe, and the transducer arranged as shown in Figure B-1. Using the arrangement shown:

a) Initially, the micromanometer is zeroed by opening the valve connecting the two sides of the pressure transducer and the voltage reading for zero pressure difference is determined;

b) The valve is then closed fully;

c) Using the syringe, one side of the transducer is pressurized slightly;

d) The voltage reading from the transducer, and the water height in the micromanometer is then found;

e) Steps c and d are repeated until 10 to 15 calibration points are obtained;

f) The pressure transducer is now removed from the calibration setup and connected to the Pitot tube.

At this point manometer readings of water height were converted to velocities using a reduced Energy Equation:

$$V = 2g(h_2 - h_1) \qquad (31)$$

where: V is velocity in meters per second;
g is the Gravitational Constant; and
h_1 and h_2 are water heights in meters.

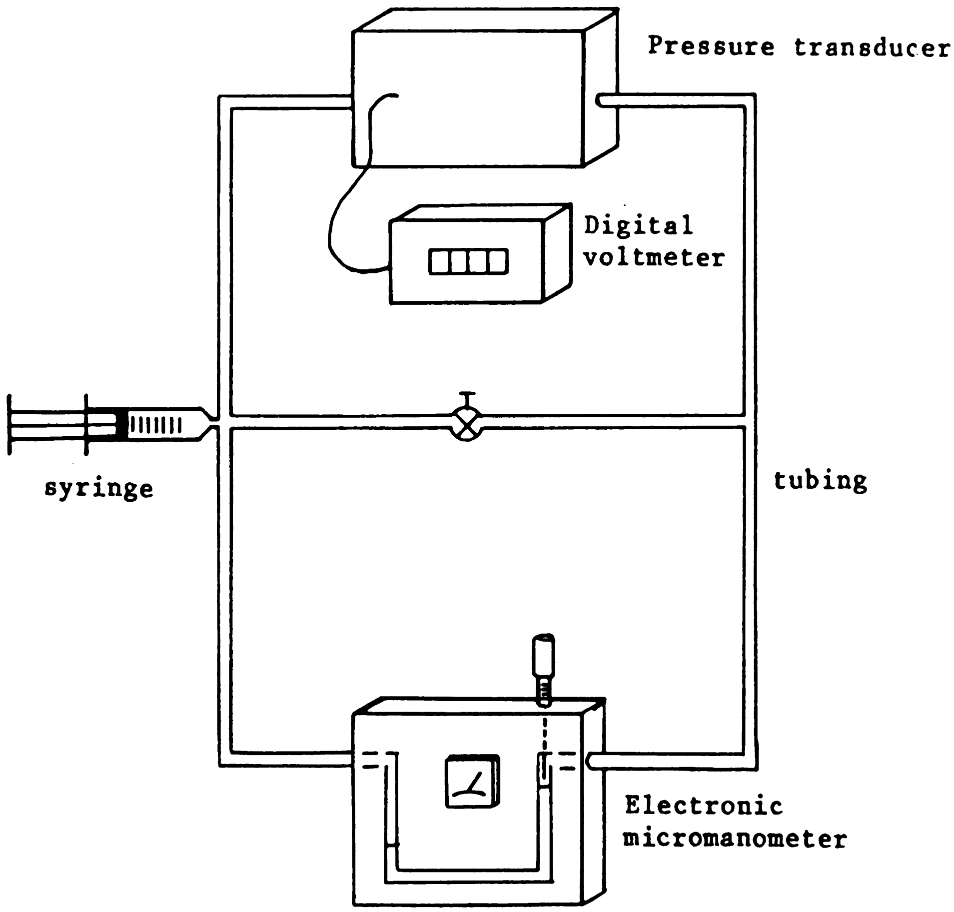

Figure B-1. Pressure Transducer Calibration Setup

The voltage readings of the transducer were next plotted against velocity, and the best-fit line determined. This line was then used to calculate air speeds directly from voltages with Fortran program PROFILE presented in Appendix E.

A typical velocity calibration is shown in Figure B-2.

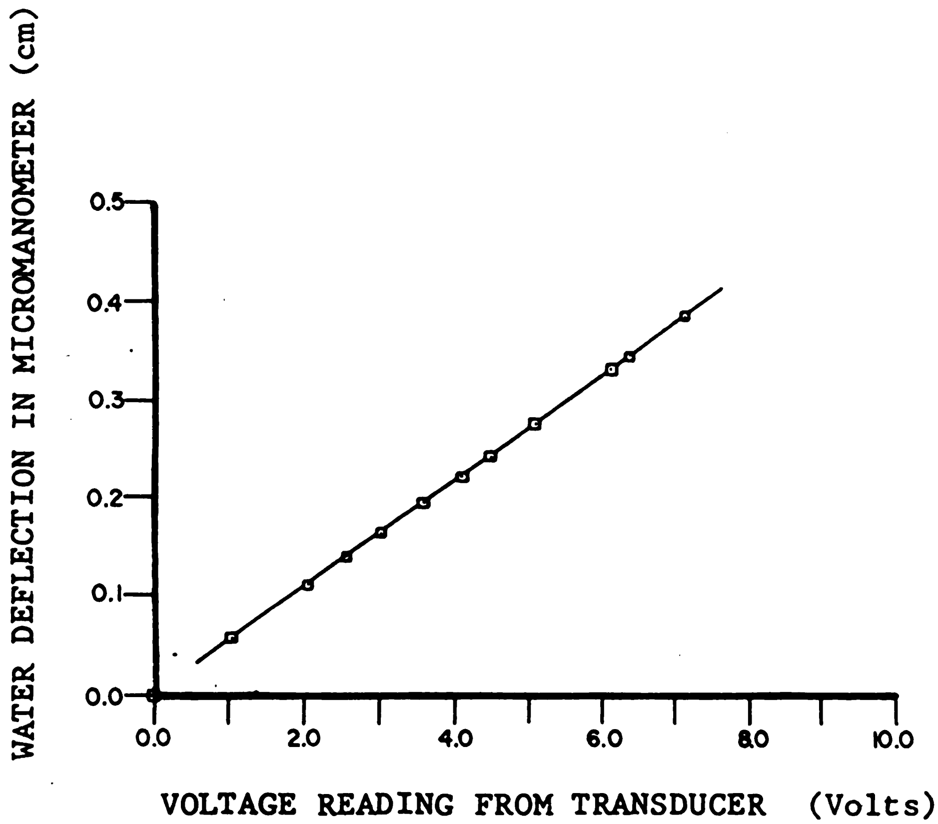

Figure B-2. Typical velocity calibration

Appendix C

WAVE GAGE CALIBRATION

The capacitance wave gage was cleaned with methanol and distilled water, then calibrated at the start of each experiment day. For the calibration, the gage was mounted on the instrument support carriage, which was then moved and stopped at various locations after which the voltage from the gage and the carriage displacement readings were recorded. After 10 to 15 points were obtained in this manner, the points were plotted, a least-squares analysis performed, and a best-fit line was drawn. A typical calibration is presented in Figure C-1. Very little change from day to day was noted in the slope of the calibration line. The y-intercept however, did show some variation, but since the gage was used to obtain the RMS of the waves, a slow DC drift was of small concern.

Following calibration, the wave gage was mounted on a fixed support in the tunnel, with the still water level at its approximate center, and the equation for the line wave input into the Fortran program WAVE, presented in Appendix F.

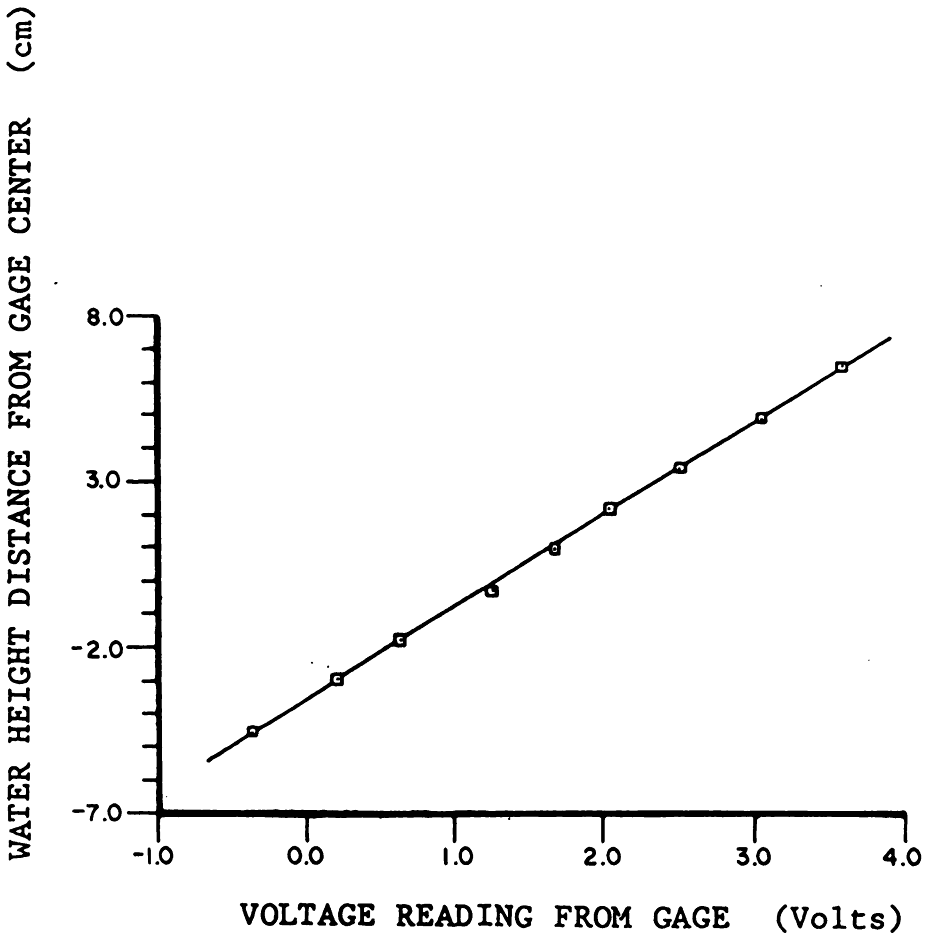

Figure C-1. Typical wave gage calibration

Appendix D

HUMIDITY CALIBRATIONS

1. Acceptable Velocity Determination

The vast volumes of psychrometric literature are quite inconsistent as to what is an adequate ventilation velocity in order not to change the psychrometric constant of the wet bulb by improper cooling. Therefore, the appropriate first step was to determine this acceptable velocity. The following procedure was used:

a) The humidity probe described in Section 3.3.3 was located centrally in the wind tunnel;

b) The tunnel was sealed and a steady state humidity condition was developed;

c) Wet bulb temperature readings were taken at several different flow rates through the humidity probe. These readings were compared against a duplicate wet bulb with a constant high flow rate, (assuring adequate ventilation).

An orifice/ manometer system was used to determine the air stream velocity through the humidity probe. The results are shown in Figure D-1. From this curve it is seen that the acceptable velocity past the wet bulb must be greater than approximately 0.50 m/s, in order that the psychrometric constant be a true constant for the humidity probe.

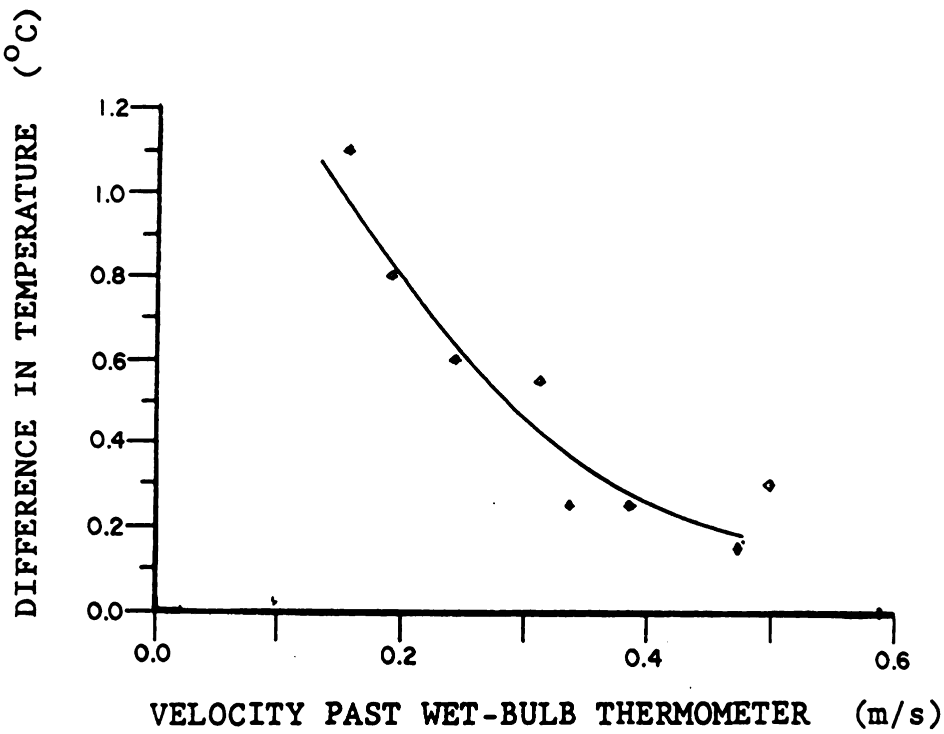

Figure D-1. Change in wet bulb temperature with flow rate

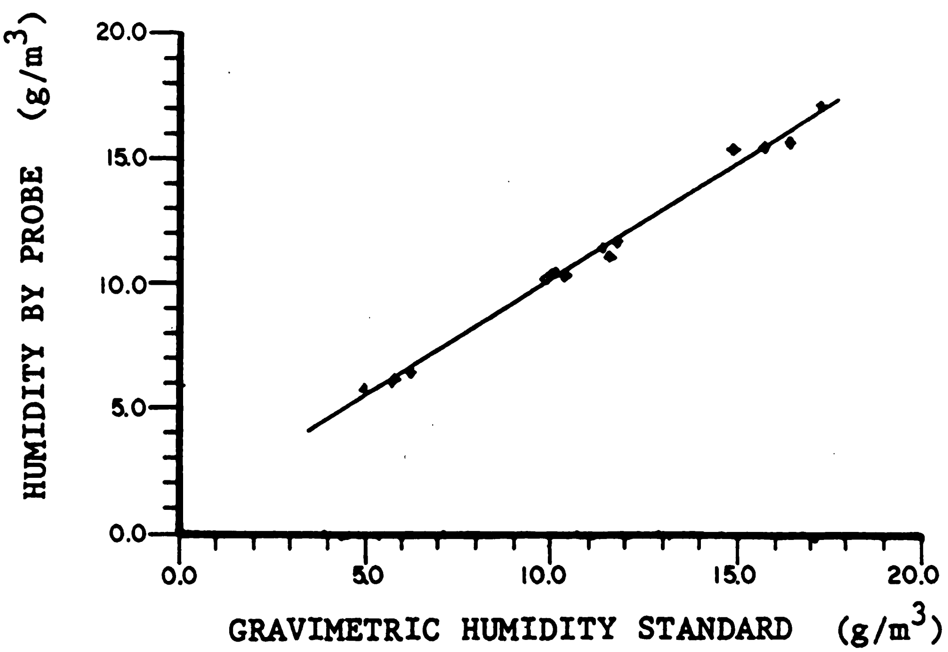

Figure D-2. Variation of humidity by probe from a standard

2. Calibration for Humidity

The calibration of the humidity probe was accomplished by a method developed using aluminum coils that work using the same principle as the gravimetric hygrometer used by the National Bureau of Standards. In this method, a sampling tube was set up in the wind tunnel at the same location as the humidity probe, and the wind tunnel was run a sufficient amount of time to assure a steady state. The aluminum coil was kept in a bath of 1-propanol alcohol cooled to about -60°C by dry ice. As the air passed through the cold coil at a constant flow rate of 1 m_3/s, the water vapor in it condensed and froze to the inside wall of the coil. The humidity of the sampled air was then determined by simply determining the change in weight of the coil over the sampling time period of a constant flow rate. During the procedure, the following precautions were exercised:

a) the coil was kept clean and dry on the outside during critical periods;

b) the coil was at room temperature during its weighing;

c) the coil openings were sealed at all times except for sampling and weighing;

d) the coil had reached bath temperature before sampling was begun;

e) flow rate through the coil was constant;

f) After final weighing, the condensed moisture was blown out of the coil with clean compressed air, and the coil was heated then cooled before the dry weight was determined for the next run.

Results from these tests are shown in Figure D-2. It can be seen that the correlation is quite good.

It should be noted that at first two coils were put in series to determine if any water vapor escaped past the first coil to the second. From these tests it was found that less than 1% of the vapor escaped. Therefore, the use of one coil was sufficient for calibration.

3. Calibration of the Thermistor

For the humidity experiments described in Section 3.3.3, a thermistor attached to the humidity probe was used for dry bulb temperatures. This thermistor was calibrated with the thermometer used for wet bulb temperatures. The results of heating and cooling tests over a range of temperatures are plotted in Figure D-3. Equation (32), was used to correct for the difference of the thermistor from thermometer during the calculation of the actual vapor pressure.

$$T.C. = -0.25227 - 6.8225x10^{-4}T + 8.2335x10^{-5}T^2 - 1.4x10^{-5}T^3 \quad (32)$$

where: T.C. is the thermistor temperature correction; and
T is the temperature of the thermistor in degrees Celsius.

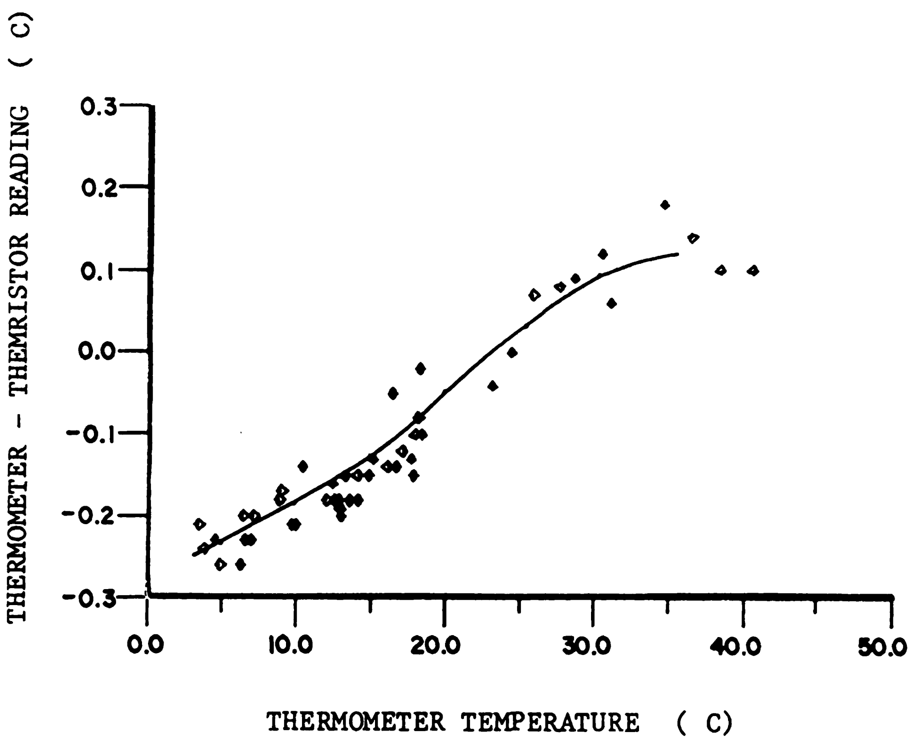

Figure D-3. Variation of thermistor from thermometer temperature

APPENDIX E

```
FORTRAN IV      V02.5-2                                    PAGE 001

0001          PROGRAM PROFILE
        C
        C     CONTROL PROGRAM FOR MULTICHANNEL A/D SAMPLING AT USER DESIGNATED
        C     POINTS ALONG THE TRAVERSE OF A PROBE.  THE USER CONTROLS THE
        C     STEPPING MOTOR THAT MOVES THE PROBE, AND THE COMPUTER FOLLOWS
        C     THE PROBE POSITION AND MOVEMENT.
        C
        C     ASSEMBLY LANGUAGE SUBROUTINE MODULE:
        C     TRAVSE(N(1),NSMPL,NTICK,NRATE,ICHAN,NCHAN,IERR)
        C         SUBROUTINE TRAVSE.MAC KEEPS TRACK OF PROBE POSITION
        C         AND SAMPLES AT UP TO 30 CHOSEN LOCATIONS
        C
        C         -FURTHER DESCRIPTION IN TRAVSE.MAC
        C                   N(1)= FIRST VALUE OF SAMPLE BUFFER
        C                 NSMPL = NUMBER OF SAMPLES TO TAKE
        C                 NTICK = NUMBER OF  TICKS BETWEEN SAMPLES
        C                 NRATE = CLOCK RATE (0-7)
        C                         0=STOP
        C                         1=1 MHZ
        C                         2=100 KHZ
        C                         3=10 KHZ
        C                         4=1 KHZ
        C                         5=100 HZ
        C                         6=ST1
        C                         7=LINE FREQ(60 HZ)
        C                 ICHAN = A/D CHANNEL TO SAMPLE (0-15)
        C                 NCHAN = NUMBER OF CHANNELS TO SAMPLE
        C                  IERR = NUMBER OF SAMPLING ERRORS
        C
        C
        C
        C
        C     LINKING INSTRUCTIONS FOR MODULES:
        C     LINK PROFIL,TRAVSE
        C
        C
0002          REAL MEAN(30,3)
0003          COMMON /FLTR/ V(30,1,150),NCHAN,NSMPL
0004          COMMON /BLOCK1/SAMPHT(30),HEIGHT,N(4500),NPOINT
0005          DIMENSION TITLE (20), TITLE2 (20)
0006          DATA SUM/0./
        C
        C     SPECIFY SAMPLING CONDITIONS
        C
0007          WRITE (7,1406)
0008    1406  FORMAT ('0PROF3-  30 SAMPLINGS OF 150 EACH ON 1 CHANNEL.')
0009    129   WRITE(7,105)
0010    105   FORMAT('0NUMBER OF SAMPLES/CHANNEL ? ',$)
0011          READ(5,110) NSMPL
0012    110   FORMAT(I5)
0013          WRITE(7,120)
0014    120   FORMAT('0CLOCK RATE (0-7) ? ',$)
0015          READ(5,110) NRATE
0016          IF(NRATE.EQ.1) DELT=.001
0018          IF(NRATE.EQ.2) DELT=.01
```

```
      IF(NRATE.EQ.3) DELT=.1
      IF(NRATE.EQ.4) DELT=1.0
      IF(NRATE.EQ.5) DELT=10.0
      WRITE(7,130)
130   FORMAT('0NUMBER OF CLOCK TICKS/SAMPLE ? ',$)
      READ(5,110) NTICK
      DELT=DELT*NTICK
      WRITE(7,140)
140   FORMAT('0FIRST CHANNEL TO SAMPLE ? ',$)
      READ(5,110) ICHAN
      WRITE(7,145)
145   FORMAT('0NUMBER OF CHANNELS TO SAMPLE ? ',$)
      READ(5,110) NCHAN
      NSMPT=NSMPL*NCHAN
      WRITE (7,1)
1     FORMAT('0DATA FILE NAME ?  ',$)
      CALL ASSIGN (2,'TT:',-1,'NEW')
      WRITE (7,151)
151   FORMAT ('0DO YOU WANT THE RAW DATA IN THE DATA FILE (Y/N) ?',$)
      READ (5,220) LIST
      WRITE (7,1201)
1201  FORMAT ('0PLOTTING FILE FOR VELOCITY PROFILES ?  ',$)
      CALL ASSIGN (3,'TT:',-1,'NEW')
C
C     CHECK IF USER WANTS TO CONVERT FROM VOLTS TO OTHER UNITS
C
      WRITE(7,231)
231   FORMAT('0DO YOU WANT TO CONVERT DATA FROM VOLTS TO OTHER UNITS (Y/
     1N)?',$)
      READ(5,220) ICONVT
      IF(ICONVT.NE.1HY) GOTO 148
      CALL CONVRT (NPOINT)
148   WRITE (7,3)
3     FORMAT('0COMMENTS FOR DATA FILE ?',$)
      READ (5,4) TITLE
      READ (5,4) TITLE2
4     FORMAT (20A4)
4000  FORMAT (';',20A4)
      WRITE (2,4000) TITLE
      WRITE (2,4000) TITLE2
C
C          ECHOING SAMPLING PARAMETERS
C
      WRITE(7,150) NSMPL,NTICK,NRATE
      WRITE(2,1510) NSMPL,NTICK,NRATE
150   FORMAT(//,'SAMPLING CONDITIONS:',///,
     1' NSMPL =',I5,//,' NTICK=',I5,//,' NRATE=',I5,/)
1510  FORMAT(';SAMPLING CONDITIONS:',/,
     1';NSMPL =',I5,/,';NTICK=',I5,/,';NRATE=',I5,)
      WRITE(7,160) NCHAN,ICHAN
      WRITE(2,160) NCHAN,ICHAN
160   FORMAT(';SAMPLING ',I2,' CHANNELS STARTING ON CHANNEL ',I2,)
C
C          GETTING PARAMETERS FOR THE PROFILE
```

```
      C
                WRITE (7,301)
        301     FORMAT('0SAMPLING PROBE STARTING HEIGHT (cm) ? ',$)
                READ (5,302) HEIGHT
        302     FORMAT (F8.4)
      C
                WRITE (7,391)
        391     FORMAT (' PRESS:',/,10X,' "S" KEY TO TAKE A SAMPLE SET.',/,
               110X,' "P" KEY TO DISPLAY CURRENT PROBE HEIGHT.',/,
               19X,' "^C" KEY TO PROCESS SAMPLES.',//)
      C
      C             HANDING CONTROL OVER TO TRAVSE.MAC
      C
                CALL TRAVSE(N(1),NSMPL,NTICK,NRATE,ICHAN,NCHAN,IERR)
      C
      C
                WRITE(7,170) IERR
        170     FORMAT('0**********SAMPLING FINISHED**********',//,
               1I5,' - A/D ERRORS ENCOUNTERED',/)
        220     FORMAT(A1)
      C
      C         PROCESS BUFFER TO YIELD VOLTAGES AS
      C         V ( SAMPLING#, CHANNEL#, POINT# IN SAMPLING )
      C
                DO 230 K=1,NPOINT
                DO 230 J=1,NCHAN
                DO 230 I=1,NSMPL
        230     V(K,J,I)=(N((NCHAN*I+J-NCHAN)+NCHAN*NSMPL*(K-1))-2048)/400.
      C
      C         CONVERT FROM VOLTS TO OTHER UNITS USING CALIBRATION DATA
      C
        153     IF(ICONVT.NE.1HY) GOTO 126
                CALL CONVRT (NPOINT)
        126     IF (LIST.NE.1HY) GO TO 259
                DO 1104 K=1,NPOINT
                WRITE (2,1100) K, SAMPHT(K)
        1100    FORMAT (';SAMPLING NUMBER',I4,'   OCCURING AT',F8.3,'  cm.')
                DO 1104 I=1,NSMPL
        1104    WRITE (2,*) (V(K,J,I),J=1,NCHAN)
      C
      C         DETERMINING THE SAMPLE MEANS
      C
        259     DO 1105 K=1,NPOINT
                DO 2300 J=1,NCHAN
                DO 232 I=1,NSMPL
        232     SUM=SUM+V(K,J,I)
                MEAN (K,J)= SUM/NSMPL
        2300    SUM=0.0
                SLOGHT=ALOG10(SAMPHT(K))
                MEAN (K,1)=SQRT(MEAN(K,1))
                WRITE (3,1106) (MEAN (K,J), J=1,NCHAN),SLOGHT
        1105    WRITE (2,1106) (MEAN (K,J), J=1,NCHAN),SAMPHT(K)
        1106    FORMAT ('RD',2G15.7)
                WRITE (2,1017)
```

```
0103          WRITE (3,1017)
0104   1017   FORMAT ('ED')
0105          CALL CLOSE (2)
0106          CALL CLOSE (3)
0107   127    CALL EXIT
0108          END
```

```
FORTRAN IV      V02.5-2                                          PAGE 001

      C
      C
0001        SUBROUTINE PRNT (ICOUNT,SFLAG,DIRCTN)
      C
      C     SUBROUTINE FOR PRINTING THE CURRENT HEIGHT OF THE
      C     SAMPLING PROBE (IN CENTIMETERS)
      C
      C     WHERE:
      C         ICOUNT=THE # OF COUNTS BEWTEEN PERTURBATIONS OF SYSTEM
      C         SFLAG=FLAG SIGNALLING THAT WE JUST SAMPLED SO WE WISH
      C                 TO PASS THIS HEIGHT BACK TO MAIN PROGRAM
      C         DIRCTN=INTEGER SHOWING WHICH DIRECTION THE PROBE IS MOVING
      C                 ITS VALUES CAN BE: 1 -UP,  -1 -DOWN.
      C
0002        DATA M/0/
0003        INTEGER ICOUNT,DIRCTN
0004        LOGICAL*1 SFLAG
0005        COMMON /BLOCK1/SAMPHT(30),HEIGHT,N(4500),M
      C
      C
      C         CONVERTING COUNTS TO CENTIMETERS, AND PRINTING
      C
0006        HEIGHT=HEIGHT+DIRCTN*(ICOUNT/1574.8)
0007        WRITE (7,100) HEIGHT
0008  100   FORMAT ('+PROBE HEIGHT=',F8.3,' cm.',//)
      C
      C         COMING FROM SAMPLING- SAVE FOR USE IN MAIN PROGRAM
      C
0009        IF (.NOT.SFLAG) GO TO 200
0011        M=M+1
0012        SAMPHT(M)=HEIGHT
0013        SFLAG=.FALSE.
0014  200   RETURN
0015        END
```

```
FORTRAN IV      V02.5-2                                           PAGE 001

C
C
C
C
C
            SUBROUTINE CONVRT (NPOINT)
            COMMON /FLTR/ V(30,1,150),NCHAN,NSMPL
            DIMENSION A(2),B(2)
            LOGICAL CCALL
            DATA CCALL/0/
            IF (CCALL) GO TO 6
            DO 4 J=1,NCHAN
            WRITE(7,1) J
      1     FORMAT(' ENTER CALIBRATION DATA FOR CHANNEL',I3/5X,'Y1(NEW UNITS),
           1X1(VOLTS),Y2(NEW UNITS),X2(VOLTS):',$)
            READ(5,*) Y1,X1,Y2,X2
            A(J)=(Y2-Y1)/(X2-X1)
            B(J)=Y1-A(J)*X1
            WRITE (2,11) J
      11    FORMAT (';CALIBRATION FOR CHANNEL',I4)
            WRITE (2,12) A(J), B(J)
      12    FORMAT (';NEW UNITS=VOLTAGE    *',F10.5,'    +',F10.5)
      4     CCALL=.TRUE.
            RETURN
      6     DO 5 M=1,NPOINT
            DO 5 J=1,NCHAN
            DO 5 I=1,NSMPL
      5     V(M,J,I)=A(J)*V(M,J,I)+B(J)
      10    CONTINUE
            RETURN
            END
```

```
FORTRAN IV      V02.5-2                                              PAGE 001

      C
      C
      C
      C
0001        SUBROUTINE NEWDLY (DELAY)
      C
      C         TEMPORARY SUBROUTINE TO ADJUST MACRO
      C         PROGRAM TO THE PULSING HARDWARE
      C
0002        INTEGER DELAY
0003        WRITE (7,10)
0004  10    FORMAT ('ONEW MICROSECOND DELAY TIME ? ',$)
0005        READ (5,20) DELAY
0006  20    FORMAT (I6)
0007        RETURN
0008        END
```

```
.TITLE TRAVSE
.CSECT  TRAVSE
;
;PROGRAM WRITTEN BY D. HARMS DURING MOST OF AUGUST 1982
;
;TRAVSE.MAC IS A SUBROUTINE TO BE USED WITH A
;STEPPING MOTOR CONTROLLER.  CONTROLLER PULSES
;ARE SENT TO BOTH THE STEPPING MOTOR AND TO SCHMIDTT
;TRIGGER 1.
;
;THE CLOCK IS USED IN RATE 6: COUNT ST1 FIRINGS SO
;ST1 FIRINGS CORRESPOND TO A FIXED DISTANCE THAT THE
;STEPPING MOTOR HAS TRAVELLED.
;
;SCHMIDTT TRIGGER 2 IS USED TO SIGNAL THE DIRECTION OF
;MOTOR ROTATION, (UP OR DOWN TRAVERSE IN OUR CASE.)
;NOTE THAT ADDITIONAL HARDWARE IS USED HERE.
;ST2 WILL RECEIVE ONE PULSE WHEN THE CONTROLLER IS
;SWITCHED TO TURN THE MOTOR COUNTER-CLOCKWISE.
;ST2 WILL RECEIVE TWO PULSES WITHIN A CERTAIN TIME
;INTERVAL WHEN THE CONTROLLER IS SWITCHED TO TURN THE
;MOTOR CLOCKWISE. (UP IN OUR CASE.)
;
;SUBROUTINE CALLED FROM FORTRAN BY:
;CALL TRAVSE (IBUF(1),NSAMPL,NTICK,NRATE,ICHAN,NCHAN,ERROR)
;       WHERE:
;       IBUF(1)=FIRST VALUE IN SAMPLE ARRAY
;       NSMPL=NUMBER OF SAMPLES
;       NTICK=NUMBER OF CLOCK TICKS/SAMPLE
;       NRATE=CLOCK TICK RATE
;       ICHAN=FIRST CHANNEL NUMBER
;       NCHAN=NUMBER OF CHANNELS TO BE SAMPLED
;       ERROR=NUMBER OF ERRORS WHILE SAMPLING
;
;PROGRAM SHOULD BE LINKED WITH A MAIN CONTROL PROGRAM AND
;THE PRINTING SUBROUTINE, PRNT.FOR.  A SAMPLING ROUTINE IS
;INCLUDED IN TRAVSE.MAC FOR SAMPLING AT DESIRED LOCATIONS
;ALONG THE MOTOR'S TRAVERSE.  SHOULD ANOTHER SAMPLING ROUTINE
;BE DESIRED MAJOR CHANGES ARE NECESSARY.
;
        .GLOBL TRAVSE,PRNT,NEWDLY
        .MCALL .TTYOUT,.PRINT
                                ;
                                ;DEFINING AND LOCATING VARIOUS VARIABLES
BLK1:   .WORD   3               ;FIRST BLOCK FOR PASSING TO FORTRAN SUBROUTINE
        .WORD   TEMPCK          ;PRNT- FOR PRINTING THE PROBE HEIGHT
        .WORD   SFLAG           ;
        .WORD   UFLAG           ;
        TEMPCK: 0               ;TEMPORARY CLOCK BUFFER FOR PROBE HEIGHT
        SFLAG:  0               ;FLAG FOR SIGNALLING THAT WE CAME FROM SAMPLE
        UFLAG:  1               ;FLAG SIGNALLING MOTOR DIRECTION 1-UP, -1-DOWN
                                ;
BLK2:   .WORD   1               ;NEXT BLOCK FOR SUBROUTINE PASSING
        .WORD   DELAY           ;TO FORTRAN SUBROUTINE NEWDLY
        DELAY:  177324          ;HARDWARE DEPENDANT VALUE FOR 2'(MICRO-
                                ;SECOND DELAY) BETWEEN THE TWO PULSES
                                ;SIGNALLING UPWARD MOTOR MOVEMENT.
```

```
                              ;
        TEMPR5: 0             ;TEMPORARY STORAGE LOCATION FOR R5 CONTENTS
                              ;DURING SUBROUTINE CALLS, TO BE LATER RESTORED
        NSAMP:  0             ;TOTAL NUMBER OF SAMPLES PER CHANNEL
        COUNT:  0             ;LOOP COUNTER FOR NSAMP
        ERROR:  0             ;A/D ERROR COUNT
        CLOCK:  0             ;TEMPORARY CLOCK STATUS REGISTER FOR A/D
        TICKS:  0             ;2's COMPLEMENT FOR CLOCK TICKS BETWEEN
                              ;A/D CONVERSIONS
        TEMPAD: 0             ;TEMPORARY A/D CONTROL/STATUS REGISTER
        NCHAN:  0             ;NUMBER OF CHANNELS TO SAMPLE FROM
        COCHAN: 0             ;LOOP COUNTER FOR NCHAN
        DFLG:   0             ;DONE FLAG SIGNALLING ALL SAMPLES TAKEN
        ADDR:   0             ;STARTING ADDRESS OF SAMPLE BUFFER
        ADVEC1=400            ;A/D DONE INTERRUPT VECTOR
        ADVEC2=402            ;-AND PRIORITY
        ERVEC1=404            ;A/D ERROR INTERRUPT VECTOR
        ERVEC2=406            ;-AND PRIORITY
        OVFL1=440             ;CLOCK OVERFLOW INTERRUPT VECTOR
        OVFL2=442             ;-AND PRIORITY
        ST2=444               ;ST2 INTERRUPT VECTOR
        ADSR=177000           ;A/D CONTROL/STATUS REGISTER
        ADSR1=177001          ;HIGH BYTE OF ADSR (FOR INCREMENTING CHANNELS)
        ADBR=177002           ;A/D DATA BUFFER
        CLKSR=170420          ;REAL TIME CLOCK CONTROL/STATUS REGISTER
        CLKBR=170422          ;CLOCK BUFFER/PRESET REGISTER
        TKS=177560            ;KEYBOARD CSR
        TKB=TKS+2             ;KEYBOARD BUFFER
        TPS=TKS+4             ;TERMINAL CSR
        TPB=TKS+6             ;TERMINAL BUFFER
                              ;
TRAVSE:                       ;
        CLR @#ADSR            ;INITIALLIZING A/D CSR
        CLR ERROR             ;INITIALLIZING THE ERROR COUNT
                              ;
                              ;FIRST PASSING ARGUMENTS FROM FORTRAN
        MOV 2(R5),ADDR        ;BEGINNING ADDRESS OF SAMPLE OUTPUT BUFFER
        MOV @4(R5),NSAMP      ;NUMBER OF SAMPLES
        MOV @6(R5),R1         ;T=# OF CLOCK TICKS
        NEG R1                ;
        MOV R1,TICKS          ;PUT -T INTO TEMPORARY CLOCK BUFFER
        MOV @10(R5),CLOCK     ;CLOCK RATE FOR SAMPLING
                              ;
                              ;SETTING UP A TEMPORARY CLOCK STATUS
                              ;REGISTER THAT WILL BE LOADED FOR SAMPLING
        ASL CLOCK             ;PUTTING CLOCK RATE IN
        ASL CLOCK             ;     BITS 3-5
        ASL CLOCK             ;
        BIC #177707,CLOCK     ;ZERO OTHER BITS
        BIS #3,CLOCK          ;CLOCK STATUS:
                              ;REPEATED INTERVAL
                              ;START WHEN LOADED INTO CLKSR
                              ;
                              ;SETTING UP A TEMPORARY A/D STATUS REGISTER
        MOV @12(R5),TEMPAD    ;GET FIRST CHANNEL NUMBER
        BIC #177600,TEMPAD    ;ZERO OTHER BITS
        SWAB TEMPAD           ;SWAP BYTES
```

```
        BIS #040140,TEMPAD      ;SET UP A/D STATUS
                                ;WHEN LOADED IT WILL:
                                ;ENABLE REAL TIME CLOCK
                                ;INTERRUPT WHEN A/D IS DONE
                                ;INTERRUPT FOR AN A/D CONVERSION ERROR
                                ;NOTE: A/D SAMPLING IS HARD-WIRED TO THE
                                ;CLOCK OVERFLOW (THIS BIT NEED NOT BE SET)
        MOV @14(R5),NCHAN       ;THE NUMBER OF CHANNELS TO SAMPLE
                                ;
                                ;
                                ;SETTING UP ALL INTERRUPT SERVICE ROUTINE
                                ;VECTORS AND PRIORITIES
        MOV #FIRE1,@#ST2        ;INTERRUPT SERVICE ROUTINE FOR FIRST
                                ;ST2 FIRING
        MOV #340,ST2+2          ;PRIORITY 7
        MOV #IPRINT,@#OVFL1     ;PRINT THE PROBE HEIGHT ON A CLOCK OVERFLOW
        MOV #340,@#OVFL2        ;PRIORITY 7
        MOV #ISR1,@#ADVEC1      ;SET UP A/D DONE ISR VECTOR
        MOV #340,@#ADVEC2       ;PRIORITY 7
        MOV #ERR,@#ERVEC1       ;SET UP A/D ERROR ISR VECTOR
        MOV #340,@#ERVEC2       ;PRIORITY 7
        BIC #100,@#TKS          ;CLEAR KEYBOARD INTERRUPT ENABLE
                                ;
                                ;READY, SET, AND GO !!
        MOV #40165,@#CLKSR      ;START CLOCK COUNTING ON ST1 FIRINGS,
                                ;FROM ZERO, INTERRUPT ON OVERFLOW
                                ;OR ST2 FIRING
TT:                             ;
        INC @#TKS               ;SETTING THE ENABLE BIT FOR THE KEYBOARD
TTY:    TSTB @#TKS              ;DO WE HAVE A CHARACTER FROM THE KEYBOARD ?
        BPL TTY                 ;LOOP IF NONE IS READY
        CMPB #123,@#TKB         ;IS IT A SAMPLE SIGNAL ?  "S"
        BEQ S                   ;TAKE AN A/D SAMPLE IF IT IS
        CMPB #003,@#TKB         ;^C TO SIGNAL AN EXIT ?
        BEQ BYE                 ;YES- THEN GO BYE BYE
        CMPB #104,@#TKB         ;DO WE WANT TO CHANGE THE DELAY FOR ST2 ?
        BEQ D                   ;YES
        CMPB #160,@#TKB         ;IS IT A SMALL "P"
        BEQ P                   ;TELL USER WHATS WRONG
        CMPB #120,@#TKB         ;IS IT A PRINT SIGNAL ?   "P"
        BNE TT                  ;CONTINUE IF NOT
        JSR PC,MPRINT           ;OTHERWISE PRINT OUT THE HEIGHT OF THE PROBE
        BR TT                   ;WAIT FOR MANUAL TRIGGERING FOR SAMPLING
                                ;OR FOR ONE OF SEVERAL INTERRUPTS
S:      JSR PC,SAMPLE           ;
        CLR @#TKB
        BR TT                   ;RETURN FOR MORE ACTION
D:      BIS #100,@#TKS
        MOV R5,TEMPR5           ;SAVE THE ARGUEMENT POINTER DURING CALL-
        JSR PC,NEWDLY           ;OF FORTRAN SUBROUTINE NEWDLY (NEW DELAY)
        MOV @2(R5),DELAY        ;RETRIEVE THE NEW DELAY
        MOV TEMPR5,R5           ;RESTORE REGISTER 5's CONTENTS
        COM DELAY               ;USE THE 2'S COMPLEMENT FOR COUNTING DOWN
        BIC #100,@#TKS
        BR TT
P:      .PRINT #MESSG4          ;
        BR TT                   ;
```

```
BYE:                                    ;
        CLR OVFL2                       ;MAKE THE INTERRUPT PRIORTIES-
        CLR ST2+2                       ;       ZERO
        CLR @#CLKSR                     ;STOP THE CLOCK TO PREVENT UNEXPLAINABLE
                                        ;OCCURANCES AT A LATER TIME
        MOV ERROR,@16(R5)               ;PASSING THE NUMBER OF ERRORS TO FORTRAN
        BIS #100,@#TKS                  ;ENABLING THE KEYBOARD ON THE WAY OUT
        RTS PC                          ;EXIT TO FORTRAN MAIN PROGRAM
                                        ;
                                        ;
FIRE1:                                  ;SERVICE ROUTINE FOR ST2 FIRING
        MOV #UP,@#ST2                   ;SET UP FOR THE SECOND ST2 FIRING
                                        ;SIGNALLING UPWARD MOVEMENT
        MOV #64,@#CLKSR                 ;CHANGE CLOCK STATUS REGISTER TO ENABLE
                                        ;THE READING OF THE CURRENT HEIGHT
        CLR @#CLKBR                     ;
        MOV @#CLKBR,TEMPCK              ;SAVING THE CLOCK'S VALUE-(PROBE HEIGHT)
        MOV #002,@#CLKSR                ;ENABLE US TO MOVE THE DELAY INTO CLKBR
        MOV DELAY,@#CLKBR               ;START COUNTDOWN FOR NEXT PULSE
        MOV #40113,@#CLKSR              ;USE 1 MHZ RATE,INTERRUPT ON OVERFLOW
                                        ;OR FOR ANOTHER ST2 PULSE
        MOV #DOWN,@#OVFL1               ;INTERRUPT SERVICE ROUTINE DOWN
                                        ;THE MOTOR IS MOVING DOWN IF AN OVERFLOW
                                        ;OCCURS BEFORE A SECOND ST2 PULSE.
        RTI                             ;RETURN FOR MORE ACTION
                                        ;
                                        ;
                                        ;
UP:                                     ;
        CLR @#CLKSR                     ;STOP THE TIMER
        MOV #FIRE1,@#ST2                ;INITIALIZE ST2 INTERRUPT VECTOR AGAIN
        MOV #IPRINT,@#OVFL1             ;SAMPLE ON A CLOCK OVERFLOW AGAIN
        MOV #1,UFLAG                    ;SIGNAL THAT WE ARE GOING UP
        .PRINT #MESSG2                  ;AND TELL THE USER
        JSR PC, SPRINT                  ;PRINT OUT THE CURRENT PROBE HEIGHT
        RTI                             ;
                                        ;
                                        ;
DOWN:                                   ;
        MOV #FIRE1,@#ST2                ;INITIALIZE ST2 INTERRUPT VECTOR AGAIN
        CLR @#CLKSR                     ;RESUME COUNTING ON ST1 PULSES FROM ZERO
        MOV #IPRINT,@#OVFL1             ;SAMPLE ON A CLOCK OVERFLOW AGAIN
        MOV #-1,UFLAG                   ;SIGNAL THAT WE ARE GOING DOWN
        .PRINT #MESSG3                  ;AND TELL THE USER
        JSR PC, SPRINT                  ;PRINT OUT THE CURRENT PROBE HEIGHT
        RTI                             ;GO BACK, WAIT FOR ANOTHER SWITCH MOVEMENT
                                        ;AND COUNT ST1 PULSES
                                        ;
                                        ;
                                        ;
SAMPLE:                                 ;SAMPLING SUBROUTINE
                                        ;SAMPL IS AN INTERUPT-DRIVEN, CLOCKED
                                        ;SAMPLING SUBROUTINE.
                                        ;SAMPLING IS INITIATED ON CLOCK
                                        ;OVERFLOWS OR WHEN A SPECIAL CHARACTER
                                        ;IS SENSED.
                                        ;
```

```
        CLR DFLG                ;INITIALLIZING THE DONE FLAG
        MOV NSAMP,COUNT         ;MAXIMUM NUMBER OF SAMPLES
        MOV NCHAN,COCHAN        ;SET UP CHANNEL COUNTER
        MOV #64,@#CLKSR         ;CHANGE CLOCK STATUS REGISTER TO ENABLE
                                ;THE READING OF THE CURRENT HEIGHT
        CLR @#CLKBR             ;
        MOV @#CLKBR,TEMPCK      ;SAVING THE CLOCK'S VALUE-(PROBE HEIGHT)
        MOV #002,@#CLKSR        ;ENABLE LOADING OF CLOCK BUFFER WITH TICKS
        MOV TICKS,@#CLKBR       ;TICKS: TIME BETWEEN A/D SAMPLES
        MOV TEMPAD,@#ADSR       ;LOAD A/D STATUS REGISTER
        MOV CLOCK,@#CLKSR       ;LOADING CLOCK FOR A/D SAMPLING PARAMETERS
        MOV #007,@#TPB          ;BEEP WHEN SAMPLING BEGINS
AGAIN:                          ;WAITING FOR AN INTERRUPT
        BIS #1,@#TKS            ;SETTING THE ENABLE BIT FOR THE KEYBOARD
        TSTB @#TKS              ;DO WE HAVE A CHARACTER FROM THE KEYBOARD ?
        CMPB #003,@#TKB         ;^C TO SIGNAL AN EXIT ?
        BEQ STOP                ;YES- THEN GO BYE BYE
        TST DFLG                ;ARE WE FINISHED ?
        BEQ AGAIN               ;BACK FOR MORE WAITING
        MOV #1,SFLAG            ;SIGNAL A SAMPLE HAS BEEN TAKEN
        MOV #33,@#TPB           ;TELL TERMINAL TO ENTER
        MOV #160,@#TPB          ;REVERSE VIDEO MODE
        .PRINT #MESSG1          ;"SAMPLING HEIGHT (cm) = "
        JSR PC,SPRINT           ;PRINT OUT THE SAMPLING HEIGHT
        MOV #33,@#TPB           ;EXIT FROM
        MOV #33,@#TPB           ;EXIT FROM
        MOV #161,@#TPB          ;REVERSE VIDEO MODE
        RTS PC                  ;RETURN TO TRAVSE AGAIN
                                ;
ISR1:
SERV21:                         ;A/D DONE SERVICE ROUTINE
        MOV @#ADBR,@ADDR        ;MOVE A/D SAMPLE TO THE BUFFER
        ADD #2,ADDR             ;POINT TO THE NEXT BUFFER ADDRESS
        DEC COCHAN
        BEQ SERV29              ;ALL CHANNELS SAMPLED
        INCB @#ADSR1            ;NO,INCREMENT CHANNEL
        BIS #1,@#ADSR           ;START NEXT SAMPLE
SERV22: TSTB @#ADSR             ;SAMPLE DONE?
        BMI SERV21              ;YES, GO GET IT
        JMP SERV22              ;NO WAIT SOME MORE
SERV29:
        DEC COUNT               ;DECREMENT SAMPLE COUNT
        BEQ STOP                ;ENOUGH SAMPLES TAKEN ?
        MOV TEMPAD,@#ADSR       ;NO, SET UP A/D AGAIN
        MOV NCHAN,COCHAN        ;RESET CHANNEL COUNTER
        RTI                     ;RETURN FOR MORE A/D SAMPLES ON CLKOV
                                ;
                                ;
ERR:                            ;A/D ERROR SERVICE ROUTINE
        INC ERROR               ;COUNTING THE NUMBER OF A/D ERRORS
        BIC #100200,ADSR        ;CLEAR ERROR CONDITION
        BIC #200,@#CLKSR        ;CLEAR THE OVERFLOW FLAG
        RTI                     ;
                                ;
STOP:   CLR @#CLKSR             ;STOP THE CLOCK
        CLR @#ADSR              ;STOP ADDITIONAL A/D STARTS ON CKLOVFL
        MOV #1,DFLG             ;SIGNAL THAT ALL SAMPLES ARE TAKEN
```

```
        RTI                         ;CLEANING UP REMAINING INTERRUPT
                                    ;
MESSG1: .ASCIZ  /     DURING SAMPLING     /
MESSG2: .ASCIZ  /                                 PROBE MOVING UP /
MESSG3: .ASCIZ  /                                 PROBE MOVING DOWN /
MESSG4: .ASCIZ  / TRY CAPITOL LETTERS /
                                    ;
                                    ;
IPRINT: JSR PC,MPRINT               ;INTRODUCTION ROUTINE FOR GETTING TO MPRINT BY
        RTI                         ;INTERRUPT RATHER THAN SUBROUTINE PASSAGE.
MPRINT:                             ;PRINTING SUBROUTINE
                                    ;PASSES TO THE TERMINAL THE HEIGHT OF
                                    ;THE PROBE
        MOV #64,@#CLKSR             ;CHANGE CLOCK STATUS REGISTER TO ENABLE
                                    ;THE READING OF THE CURRENT HEIGHT
        CLR @#CLKBR                 ;
        MOV @#CLKBR,TEMPCK          ;SAVING THE CLOCK'S VALUE-(PROBE HEIGHT)
SPRINT: MOV #40165,@#CLKSR          ;LETTING THE CLOCK RESUME ON ST1 PULSES
OUT:    MOV R5,TEMPR5               ;SAVE REGISTER 5's CONTENTS
        MOV #BLK1,R5                ;LOAD THE BLOCK INTO A REGISTER FOR PASSING
                                    ;TO FORTRAN SUBROUTINE - PRNT.FOR
                                    ;INFORMATION PASSED INCLUDES THE NUMBER OF
                                    ;COUNTS SINCE LAST PERTURBATION, AND THE
                                    ;STATE OF DIRECTION AND SAMPLING FLAGS
        JSR PC,PRNT                 ;CALL PRNT
        MOV TEMPR5,R5               ;POINTING R5 BACK TO THE CORRECT LOCATION
        RTS PC
        .END TRAVSE
```

APPENDIX F

```
FORTRAN IV      V02.5-2                                              PAGE 001

0001          PROGRAM WAVE
      C       CONTROL PROGRAM FOR MULTICHANNEL A/D SAMPLING
      C       AND PROBABILITY ANALYSIS SPECIFICALLY ALTERED FOR THE
      C       SAMPLING OF LONG WAVE TRAINS
      C
      C       PROGRAM AVERAGES THE PROBABILITY RESULTS FOR NUMEROUS RUNS
      C       AND CALCULATES THE STANDARD DEVIATION OF RMS VALUES OF RUNS
      C
      C       LAST ALTERATIONS -  8/1/83
      C
      C       A/D SAMPLING MODULE CALL:
      C       SAMPL(N(1),NSMPL,NTICK,NRATE,ICHAN,NCHAN,IERR)
      C           SAMPLES DATA ON CLOCK INTERUPT AFTER INITIAL TRIGGER
      C                   N(1)= SAMPLE BUFFER
      C                   NSMPL = NUMBER OF SAMPLES TO TAKE
      C                   NTICK = NUMBER OF  TICKS BETWEEN SAMPLES
      C                   NRATE = CLOCK RATE (0-7)
      C                           0=STOP
      C                           1=1 MHZ
      C                           2=100 KHZ
      C                           3=10 KHZ
      C                           4=1 KHZ
      C                           5=100 HZ
      C                           6=ST1
      C                           7=LINE FREQ(60 HZ)
      C                   ICHAN = A/D CHANNEL TO SAMPLE (0-15)
      C                   NCHAN = NUMBER OF CHANNELS TO SAMPLE
      C                   IERR =  NUMBER OF SAMPLING ERRORS
      C
      C
      C
      C       LINKING INSTRUCTIONS FOR MODULES:
      C       LINK WAVE,SAMPL
      C
      C
0002          DIMENSION N(2000),TITLE(20),TITLE2(20)
0003          REAL*8 VARRMS(2)
0004          COMMON /FLTR/ V(2000,1),NCHAN,NSMPL,VH(2000,1)
0005          COMMON /PROB/ IPROB,IFILTR,ICORR,SRM(2),SRMS(2),SSKEW(2),
             1SRKURT(2),SCOR,NCLASS,RMS(2,50),VARRMS,IPASS
0006          NTPASS=0
0007          WRITE(7,100)
0008  100     FORMAT(' BEGIN EXECUTION OF PROGRAM WAVE:',/)
      C
      C       SPECIFY SAMPLING CONDITIONS
      C
0009          WRITE(7,140)
0010  140     FORMAT('0ENTER FIRST CHANNEL TO SAMPLE:',$)
0011          READ(5,110) ICHAN
0012          WRITE(7,145)
0013  145     FORMAT('0ENTER NUMBER OF CHANNELS TO SAMPLE:',$)
0014          READ(5,110) NCHAN
0015  2       WRITE(7,105)
0016  105     FORMAT('0ENTER NUMBER OF SAMPLES/CHANNEL:',$)
```

```
FORTRAN IV      V02.5-2                                      PAGE 002

          READ(5,110) NSMPL
    110   FORMAT(I5)
          WRITE(7,120)
    120   FORMAT('0ENTER CLOCK RATE (0-7):',$)
          READ(5,110) NRATE
          WRITE(7,130)
    130   FORMAT('0ENTER NUMBER OF CLOCK TICKS/SAMPLE:',$)
          READ(5,110) NTICK
          SRATE=(10.**(7-NRATE))/NTICK
          DO 7  J=1,NCHAN
          SRM(J)=0.0
          SRMS(J)=0.0
          SSKEW(J)=0.0
          VARRMS(J)=0.0
    7     SRKURT(J)=0.0
          SCOR=0.0
          IPASS=0
          WRITE (7,131)
    131   FORMAT ('0HOW MANY SAMPLING PASSES ARE TO BE AVERAGED ?',$)
          READ (5,110) NPASS
          STIME=NPASS*(NSMPL/SRATE)
          WRITE (7,1)
    1     FORMAT('0DATA FILE NAME ?  ',$)
          CALL ASSIGN (2,'TT:',-1,'NEW')
          WRITE (7,151)
    151   FORMAT ('0DO YOU WANT THE RAW DATA IN THE DATA FILE (Y/N) ?',$)
          READ (5,220) LIST
          IF (NTPASS .EQ. 1) GO TO 1148
          NTPASS=1
C
C         CHECK IF USER WANTS TO CONVERT FROM VOLTS TO OTHER UNITS
C
          WRITE(7,231)
    231   FORMAT('0DO YOU WANT TO CONVERT DATA FROM VOLTS TO OTHER UNITS (Y/
         1N)?',$)
          READ(5,220) ICONVT
          IF(ICONVT.NE.1HY) GOTO 148
          CALL CONVRT
    148   WRITE(7,106)
    106   FORMAT('0DO YOU WANT TO FILTER DATA (Y/N)?',$)
          READ(5,220) IFILTR
          WRITE(7,108)
    108   FORMAT('0DO YOU WANT TO CALCULATE MEAN,RMS,ETC. (Y/N)?',$)
          READ(5,220) IPROB
          IF (IPROB .NE. 1HY) GO TO 1148
          WRITE(7,3000)
    3000  FORMAT('0ENTER NUMBER OF CLASSES IN FREQUENCY TABLE (<20):',$)
          READ(5,*) NCLASS
          WRITE(7,11)
    11    FORMAT('0DO YOU WANT TO CALCULATE THE CORRELATION COEFFICIENT',
         1' (Y/N)?',$)
          READ(5,220) ICORR
    1148  WRITE(7,149)
    149   FORMAT(///,' ********************************************',//)
```

```
C
C     PRINT SAMPLING CONDITIONS
C
      WRITE (7,3)
3     FORMAT('0COMMENTS FOR DATA FILE ?',$)
      READ (5,4) TITLE
      READ (5,4) TITLE2
4     FORMAT (20A4)
      WRITE (2,4) TITLE
      WRITE (2,4) TITLE2
      WRITE(7,150) NSMPL,NTICK,NRATE
      WRITE(2,150) NSMPL,NTICK,NRATE
150   FORMAT(1X,'SAMPLING CONDITIONS:',///,
     1' NSMPL =',I5,//,' NTICK=',I5,//,' NRATE=',I5,/)
      WRITE(7,160) NCHAN,ICHAN
      WRITE(2,160) NCHAN,ICHAN
160   FORMAT(' SAMPLING ',I2,' CHANNELS STARTING ON CHANNEL ',I2,/)
169       CALL SAMPL(N(1),NSMPL,NTICK,NRATE,ICHAN,NCHAN,IERR)
      IPASS=IPASS+1
      WRITE(7,170) IPASS,IERR
170   FORMAT('0**********SAMPLING ',I4,'   FINISHED**********',/,
     1I5,' - A/D ERRORS ENCOUNTERED',/)
      IERR=0
220   FORMAT(A1)
C
C     PROCESS BUFFER
C
      DO 230 J=1,NCHAN
      DO 230 I=1,NSMPL
230   V(I,J)=(N(NCHAN*I+J-NCHAN)-2048)/400.
C
C     CONVERT FROM VOLTS TO OTHER UNITS USING CALIBRATION DATA
C
153   IF(ICONVT.NE.1HY) GOTO 126
      CALL CONVRT
126   IF (LIST.NE.1HY) GO TO 259
      DO 221 J=1,NCHAN
221   WRITE (2,*) (V(I,J),I=1,NSMPL)
259   DO 260 J=1,NCHAN
      DO 260 I=1,NSMPL
260   VH(I,J)=V(I,J)
C
C     HIGH-LOW PASS FILTERING
C
      IF(IFILTR.NE.1HY) GOTO 125
      DO 250 ICH=1,NCHAN
      WRITE(7,101) ICH
101   FORMAT(' CH.',I2,'; DO YOU WANT TO DO HIGH PASS FILTERING',
     1' (Y/N)?',$)
      READ(5,220) IC
      IF(IC.EQ.1HN) GOTO 121
      WRITE(7,103)
103   FORMAT('0ENTER FILTER VARIABLE "A"',$)
      READ(5,111) AA
```

```
111   FORMAT(F5.3)
      CALL HGHPSS (A,ICH)
121   WRITE(7,102) ICH
102   FORMAT(' CH.',I2,'; DO YOU WANT TO DO LOW PASS FILTERING',
     1' (Y/N)?',$)
      READ(5,220) IC
      IF(IC.EQ.1HN) GOTO 250
      WRITE(7,103)
      READ(5,111) AA
      CALL LOWPSS (A,ICH)
250   CONTINUE
C
C     CALCULATE MEAN, RMS, ETC.
C
125   IF(IPROB.NE.1HY) GOTO 127
      CALL PROBAN
      WRITE (7,*) RMS (1,IPASS)
127   IF (IPASS .LT. NPASS) GO TO 169
      WRITE (2,1100)
      WRITE (7,1100)
1100  FORMAT (' CHANNEL     MEAN              RMS             SKEWNESS
     1          FLATNESS')
      DO 1102  J=1,NCHAN
      SRM(J)=SRM(J)/IPASS
      SRMS(J)=SRMS(J)/IPASS
      SRKURT(J)=SRKURT(J)/IPASS
      SSKEW(J)=SSKEW(J)/IPASS
      WRITE(2,1101) J,SRM(J),SRMS(J),SSKEW(J),SRKURT(J)
      WRITE(7,1101) J,SRM(J),SRMS(J),SSKEW(J),SRKURT(J)
1101  FORMAT(' ',I3,5X,E12.5,5X,E12.5,5X,E12.5,5X,E12.5)
      DO 1111  L=1,IPASS
1111  VARRMS(J)=VARRMS(J)+(RMS(J,L)-SRMS(J))**2.
      VARRMS(J)=SQRT(1./(IPASS-1.)*VARRMS(J))
      WRITE (2,1112) VARRMS(J)
      WRITE (7,1112) VARRMS(J)
1112  FORMAT (21X,'STD DEV. OF RMS=',E12.5)
1102  CONTINUE
      SCOR=SCOR/IPASS
      WRITE(2,1103) SCOR
1103  FORMAT (' AVERAGE CORRELATION COEFFICIENT =',E12.5)
      WRITE (7,1144)SRATE,STIME
1144  FORMAT ('0SAMPLING RATE OF',F8.2,'   Hz.,  OVER',F9.2,
     1'  SECONDS',/)
      CALL CLOSE (2)
      DO 3333 I=1,100
3333  CALL IPOKE ("177566,"7)
      WRITE(7,222)
222   FORMAT(' DO YOU WANT TO SAMPLE ANOTHER EVENT (Y/N)?',$)
      READ(5,220) IC
      IF(IC.EQ.1HY) GOTO 2
      CALL EXIT
      END
```

```
FORTRAN IV      V02.5-2                                          PAGE 001

          SUBROUTINE PROBAN
          REAL*8 VARRMS(2)
          COMMON /FLTR/ V(2000,1),NCHAN,NSMPL,VH(2000,1)
          COMMON /PROB/ IPROB,IFILTR,ICORR,SRM(2),SRMS(2),SSKEW(2),
         1SRKURT(2),SCOR,NCLASS,RMS(2,50),VARRMS,IPASS
          DIMENSION VMIN(2),VMAX(2),DX(2),RM(2),RMM(4),RKURT(2)
          DIMENSION RL(20),RK(20,2),CRKL(20,20),RKP(20)
    C
    C     RK    = STANDARDIZED VARIATE
    C     RL    = RELATIVE FREQUENCY
    C     N     = NUMBER OF CLASSES IN FREQUENCY TABLE
    C     RM    = MEAN
    C     SKEW  = SKEWNESS FACTOR
    C     RKURT= FLATNESS FACTOR
    C     CRKL  = JOINT RELATIVE FREQUENCY
    C     COR   = CORRELATION COEFFICIENT
    C
          DO 100 J=1,NCHAN
    C
    C     DETERMINE MINIMUM AND MAXIMUM VALUES OF DATA
    C
          VMIN(J)=1000000.
          VMAX(J)=-1000000.
          DO 10 I=1,NSMPL
          IF(VH(I,J).GT.VMAX(J))VMAX(J)=VH(I,J)
    10    IF(VH(I,J).LE.VMIN(J))VMIN(J)=VH(I,J)
    C
    C     GENERATE FREQUENCIES
    C
          DX(J)=(VMAX(J)-VMIN(J))/NCLASS
          DO 20 K=1,NCLASS
    20    RL(K)=0
          DO 30 I=1,NSMPL
          K=(VH(I,J)-VMIN(J))/DX(J)
          K=K+1
          IF(K.LT.1)K=1
          IF(K.GT.NCLASS)K=NCLASS
    30    RL(K)=RL(K)+1
    C
    C     CALCULATE MEAN (AND RELATIVE FREQUENCIES)
    C
          RM(J)=0.
          DO 40 K=1,NCLASS
          RK(K,J)=(K-.5)*DX(J)+VMIN(J)
          RL(K)=RL(K)/NSMPL
    40    RM(J)=RM(J)+RL(K)*RK(K,J)
    C
    C     CALCULATE RMS, SKEWNESS, AND KURTOSIS
    C
          DO 50 K=1,NCLASS
    50    RK(K,J)=RK(K,J)-RM(J)
          DO 60 M=2,4
    60    RMM(M)=0.
          DO 70 M=2,4
```

```
FORTRAN IV      V02.5-2                                    PAGE 002

0036            DO 70 K=1,NCLASS
0037     70     RMM(M)=RMM(M)+RK(K,J)**M*RL(K)
0038            RMS(J,IPASS)=SQRT(RMM(2))
0039            DO 80 K=1,NCLASS
0040            RK(K,J)=RK(K,J)/RMS(J,IPASS)
0041     80     RKP(K)=RK(K,J)
0042            SKEW(J)=RMM(3)/(RMS(J,IPASS)**3)
0043            RKURT(J)=RMM(4)/(RMS(J,IPASS)**4)
         C
         C      SUMMING PROBABILITY RESULTS
         C
0044            SRM(J)=SRM(J)+RM(J)
0045            SRMS(J)=SRMS(J)+RMS(J,IPASS)
0046            SSKEW(J)=SSKEW(J)+SKEW(J)
0047            SRKURT(J)=SRKURT(J)+RKURT(J)
         C
         C      PRINT RESULTS
         C
0048            WRITE(2,1) J,RM(J),RMS(J,IPASS),SKEW(J),RKURT(J)
0049     1      FORMAT(' ',I3,5X,E12.5,5X,E12.5,5X,E12.5,5X,E12.5)
0050     100    CONTINUE
         C      CALCULATE CORRELATION COEFFICIENT
         C
0051            IF(NCHAN.EQ.1) RETURN
0053            IF(ICORR.NE.1HY) RETURN
         C
         C      GENERATE JOINT FREQUENCY TABLE
         C
0055            DO 110 I=1,NCLASS
0056            DO 110 J=1,NCLASS
0057     110    CRKL(I,J)=0.
0058            DO 120 I=1,NSMPL
0059            K=(VH(I,1)-VMIN(1))/DX(1)+1.
0060            L=(VH(I,2)-VMIN(2))/DX(2)+1.
0061            IF(K.LT.1)K=1
0063            IF(K.GT.NCLASS)K=NCLASS
0065            IF(L.LT.1)L=1
0067            IF(L.GT.NCLASS)L=NCLASS
0069     120    CRKL(K,L)=CRKL(K,L)+1
         C
         C      CALCULATE JOINT RELATIVE FREQUENCIES
         C
0070            DO 130 I=1,NCLASS
0071            DO 130 J=1,NCLASS
0072     130    CRKL(I,J)=CRKL(I,J)/NSMPL
0073            COR=0.
0074            DO 140 K=1,NCLASS
0075            DO 140 L=1,NCLASS
0076     140    COR=COR+RK(K,1)*RK(L,2)*CRKL(K,L)
0077            WRITE(2,13) COR
0078     13     FORMAT('        CORRELATION COEFFICIENT =',E12.5)
0079            SCOR=SCOR+COR
0080            RETURN
0081            END
```

```
FORTRAN IV      V02.5-2                                  PAGE 001

         SUBROUTINE HGHPSS(A,J)
         COMMON /FLTR/ V(2000,1),NCHAN,NSMPL,VH(2000,1)
         IF (A.EQ.1)RETURN
         VH(1,J)=V(1,J)
         DO 10 I=2,NSMPL
   10    VH(I,J)=(1.-A)*V(I,J)+A*VH(I,J)
         DO 20 I=1,NSMPL
   20    VH(I,J)=V(I,J)-VH(I,J)
         RETURN
         END
```

```
FORTRAN IV      V02.5-2                                  PAGE 001

      C
      C
      C
            SUBROUTINE LOWPSS(A,J)
            COMMON /FLTR/ V(2000,1),NCHAN,NSMPL,VH(2000,1)
            IF(A.EQ.0)RETURN
            VH(1,J)=V(1,J)
            DO 10 I=2,NSMPL
      10    VH(I,J)=(1.-A)*V(I,J)+A*VH(I-1,J)
            RETURN
            END
```

```
FORTRAN IV      V02.5-2                                              PAGE 001

      C
      C
      C
            SUBROUTINE CONVRT
            COMMON /FLTR/ V(2000,1),NCHAN,NSMPL,VH(2000,1)
            DIMENSION A(2),B(2)
            LOGICAL CCALL
            DATA CCALL/0/
            IF (CCALL) GO TO 6
            DO 4 J=1,NCHAN
            WRITE(7,1) J
      1     FORMAT(' ENTER CALIBRATION DATA FOR CHANNEL',I3/5X,'Y1(NEW UNITS),
           1X1(VOLTS),Y2(NEW UNITS),X2(VOLTS):',$)
            READ(5,*) Y1,X1,Y2,X2
            A(J)=(Y2-Y1)/(X2-X1)
            B(J)=Y1-A(J)*X1
            WRITE (2,11) J
      11    FORMAT (' CALIBRATION FOR CHANNEL',I4)
            WRITE (2,12) A(J), B(J)
      12    FORMAT (' NEW UNITS=VOLTAGE    *',F10.5,'    +',F10.5,/)
      4     CCALL=.TRUE.
            RETURN
      6     DO 5 J=1,NCHAN
            DO 5 I=1,NSMPL
      5     V(I,J)=A(J)*V(I,J)+B(J)
      10    CONTINUE
            RETURN
            END
```

```
        .TITLE SAMPL.MAC
        ;SAMPL IS AN INTERUPT-DRIVEN, CLOCKED SAMPLING
        ;SUBROUTINE.  SAMPLING BEGINS WITH A POSITIVE VOLTAGE
        ;CROSSING OF THE SCHMIDTT TRIGGER 2 LEVEL AND CONTINUES
        ;FOR A SPECIFIED TIME
        ;
        ;CALLED FROM FORTRAN MAIN PROGRAM WITH:
        ;  CALL SAMPLE(IBUF(1),NSAMPL,NTICK,NRATE,ICHAN,NCHAN,ERROR)
        ;       IBUF=SAMPLE ARRAY
        ;       NSMPL=NUMBER OF SAMPLES
        ;       NTICK=NUMBER OF CLOCK TICKS/SAMPLE
        ;       NRATE=CLOCK TICK RATE
        ;       ICHAN=FIRST CHANNEL NUMBER
        ;       NCHAN=NUMBER OF CHANNELS TO BE SAMPLED
        ;       ERROR=NUMBER OF ERRORS WHILE SAMPLING
        ;
        .GLOBL SAMPL
COUNT:  0
ERROR:  0
TEMPCK: 0
TEMPAD: 0
NCHAN:  0
COCHAN: 0
DFLG:   0
ADDR:   .WORD 0
        ADVEC1=400
        ADVEC2=402
        ERVEC1=404
        ERVEC2=406
        ADSR=177000
        ADSR1=177001
        ADBR=177002
        CLKSR=170420
        CLKBR=170422
        TTPDB=177566
SAMPL:
        CLR ERROR               ;INITIALLIZING THE A/D ERROR COUNT TO 0
        CLR DFLG                ;INITIALLIZING THE DONE FLAG
        MOV 2(R5),ADDR          ;BEGINNING ADDRESS OF SAMPLE OUTPUT BUFFER
        MOV @4(R5),R0           ;NUMBER OF SAMPLES
        MOV @6(R5),R1           ;T=# OF CLOCK TICKS
        NEG R1                  ;
        MOV R1,@#CLKBR          ;PUT -T INTO CLOCK BUFFER
        MOV @10(R5),TEMPCK      ;CLOCK RATE
        ASL TEMPCK              ;SET UP CLOCK RATE
        ASL TEMPCK              ; BITS 3-5
        ASL TEMPCK              ;
        BIC #177707,TEMPCK      ;ZERO OTHER BITS
        BIS #20002,TEMPCK       ;CLOCK STATUS:
                                ;REPEATED INTERVAL
                                ;START WHEN SCHMIDT TRIGGER 2 FIRES
        MOV @12(R5),TEMPAD      ;GET FIRST CHANNEL NUMBER
        BIC #177600,TEMPAD      ;ZERO OTHER BITS
        SWAB TEMPAD             ;SWAP BYTES
        BIS #040140,TEMPAD      ;SET UP A/D STATUS:
                                ;ENABLE REAL TIME CLOCK
                                ;INTERRUPT WHEN A/D IS DONE
```

```
                                ;INTERRUPT FOR AN A/D CONVERSION ERROR
        MOV #ISR1,@#ADVEC1      ;SET UP A/D DONE ISR VECTOR
        MOV #340,@#ADVEC2       ;PRIORITY 7
        MOV #ERR,@#ERVEC1       ;SET UP A/D ERROR ISR VECTOR
        MOV #340,@#ERVEC2       ;PRIORITY 7
        MOV @14(R5),NCHAN       ;GET NUMBER OF CHANNELS TO SAMPLE
        MOV NCHAN,COCHAN        ;SET UP CHANNEL COUNTER
        MOV TEMPAD,@#ADSR       ;LOADING A/D STATUS REGISTER
        MOV R0,COUNT            ;MAXIMUM NUMBER OF SAMPLES
        MOV TEMPCK,@#CLKSR      ;LOADING CLOCK STATUS REGISTER
        MOV #007,@#TTPDB        ;BEEP WHEN SAMPLING BEGINS
AGAIN:  WAIT                    ;WAITING FOR AN INTERRUPT
        TST DFLG                ;ARE WE FINISHED ?
        BEQ AGAIN               ;BACK FOR MORE WAITING
        RTS PC                  ;RETURN TO THE MAIN PROGRAM

                                ;
ISR1:                           ;A/D DONE SERVICE ROUTINE
SERV21: MOV @#ADBR,@ADDR        ;MOVE A/D SAMPLE TO THE BUFFER
        ADD #2,ADDR             ;POINT TO THE NEXT BUFFER ADDRESS
        DEC COCHAN
        BEQ SERV29              ;ALL CHANNELS SAMPLED
        INCB @#ADSR1            ;NO,INCREMENT CHANNEL
        BIS #1,@#ADSR           ;START NEXT SAMPLE
SERV22: TSTB @#ADSR             ;SAMPLE DONE?
        BMI SERV21              ;YES, GO GET IT
        JMP SERV22              ;NO WAIT SOME MORE
SERV29: DEC COUNT               ;DECREMENT SAMPLE COUNT
        BEQ STOP                ;ENOUGH SAMPLES TAKEN ?
        MOV TEMPAD,@#ADSR       ;NO, SET UP A/D AGAIN
        MOV NCHAN,COCHAN        ;RESET CHANNEL COUNTER
        RTI                     ;RETURN FOR MORE A/D SAMPLES ON CLKOV
                                ;
                                ;
ERR:                            ;A/D ERROR SERVICE ROUTINE
        INC ERROR               ;COUNTING THE NUMBER OF A/D ERRORS
        BIC #100200,ADSR        ;CLEAR ERROR CONDITION
        BIC #200,@#CLKSR        ;CLEAR THE OVERFLOW FLAG
        RTI                     ;
                                ;
STOP:   CLR @#CLKSR             ;STOP THE CLOCK
        MOV ERROR,@16(R5)       ;PASSING THE NUMBER OF ERRORS TO FORTRAN
        MOV #1,DFLG             ;SIGNAL THAT ALL SAMPLES ARE TAKEN
        RTI                     ;CLEANING UP REMAINING INTERRUPT
        .END SAMPL
```

APPENDIX G

```
FORTRAN IV      V02.5-2    Wed 14-Sep-83 12:50:42              PAGE 001

             PROGRAM DIFFUSE
      C
      C      PROGRAM FOR CALCULATING THE CONCENTRATION OF A COMPOUND AT A
      C      LOCATION DUE TO AN ASSUMED EMISSION RATE OF AN AREA SOURCE.
      C      WRITTEN BY R.BOUWMEESTER,Ph.D.
      C      ADAPTED TO RT-11 VERSION 4 BY D.HARMS.
      C
             COMMON/CHAR/R,D,B,S,DX,E,TRIP1
             DIMENSION C(20),ZC(20),CCAL(20),CMS(20)
             LOGICAL DIAGNS,TRIP1,GRAMS
             REAL K1
      C
      C      INITIALLIZING
             GRAMS=.TRUE.
      50     CONTINUE
      C      GRAMS IS A FLAG FOR THE UNITS OF THE MEASURED CONC.
      C      INPUTTED: IT IS FALSE FOR MILLIGRAMS.
      C
      C      OPENING A DATA FILE ON DISK FOR "READ ONLY" PURPOSES:
             WRITE(7,65)
      65     FORMAT('1ENTER DATA FILE NAME:  '$)
             CALL ASSIGN (3,'TT:',-1,'RDO')
             WRITE (7,38)
      38     FORMAT('1ENTER PLOTTING FILE NAME:   '$)
             CALL ASSIGN (2,'TT:',-1,'NEW')
      C
      C      READING IN AND PRINTING THE PROFILE DESCRIPTIONS:
      C
      C      READING IN THE AERODYNAMIC PARAMETERS
      C      WHERE:      A= POWER LAW EXPONENT, (typically 0.10-0.40)
      C          U1=VELOCITY AT 10 cm (m/s)
      C          US=SHEAR VELOCITY
             WRITE (7,20)
      20     FORMAT ('0ENTER ALPHA, U10, U*  :    ',$)
             READ(5,*) A,U1,US
      C
      C      SETTING A DIAGNOSTIC FLAG
             DIAGNS=.FALSE.
             WRITE(7,73) A,U1,US
      73     FORMAT('0','WIND CONDITIONS :   alpha= ',F8.4,/,
            1       '                        U-10 = ',F8.4,/,
            1       '                        U *  = ',F8.4,//)
      C
      C      SETTING THE EMMISION RATE TO 1 g/(m**2*hr), WHICH IN
      C      g/(m**2*sec) IS:
             Q=2.78E-6
      C
      C      CALCULATING EDDY DIFFUSIVITY (K1), AT 10 cm
             K1=US*US/(A*U1)*10.
             IF(DIAGNS) WRITE(7,*) K1
      C
      C      CALCULATING THE GAMMA FUNCTION (GMMA), FROM THE POWER LAW EXPONENT
             R=2.*A+1.
             S=(A+1.)/R
```

```
0026         GMMA=(1.-.575*S+.951*S**2.-.7*S**3.+.425*S**4.-.101*S**5.)/S
       C
       C     SECTION FOR CALCULATING CONCENTRATION AT A CHOSEN POINT
       C     DUE TO AN AREA SOURCE: CALCULATION OF TWO TERMS (B AND D),
       C     THAT ARE INDEPENDANT OF FETCH AND HEIGHT.  THE TERMS ARE
       C     PARTS OF EQUATION 12, ON PAGE 10 OF "WIND-TUNNEL SIMULATION
       C     AND ASSESSMENT OF AMMONIA VOLATILIZATION FROM PONDED WATER"
       C              -BOUWMEESTER,VLEK.
0027         U1=U1*10.
0028         B=Q*R/(U1*GMMA)*(U1/(R*R*K1))**S
0029         D=-U1/(R*R*K1)
0030         IF(.NOT.DIAGNS)GO TO 125
0032           WRITE(7,75)S,R,GMMA
0033   75      FORMAT('0CALCULATED S,R,GMMA ARE:  ',3F10.5,/)
0034           WRITE(7,100)B,D
0035   100     FORMAT('0CALCULATED B,D ARE:   ',E12.4,8X,E12.4,//)
       C
       C     READING IN THE DATA
       C     WHERE:
       C         FETCH=  FETCH (m)
       C         DX=     THE INTERVAL FETCH IS DIVIDED INTO (m)
       C         ZC=     HEIGHTS OF CONCENTRATION POINTS (cm)
       C         CMS=    MEASURED CONCENTRATION
       C         NP=     NUMBER OF POINTS IN THE PROFILE
0036   125   WRITE (7,25)
0037   25    FORMAT ('0ENTER FETCH, DX  :   ',$)
0038         READ(5,*) FETCH,DX
0039         WRITE(7,130)FETCH,FETCH/DX,DX
0040   130   FORMAT('0THE FETCH OF ',F4.2,'m  WAS DIVIDED INTO',F3.0,
            1' INCREMENTS, DX= ',F4.2,' m.',//)
0041         I=1
0042         READ (3,45) COMMNT
0043   45    FORMAT (A4)
0044   150   READ(3,40) IOI, CMS(I), ZC(I)
0045         ZC(I)=ZC(I)/100.
0046   40    FORMAT (A2,G15.7, G15.7)
0047         IF (IOI.NE. 2HRD) GO TO 200
0049         I=I+1
0050         GO TO 150
       C
       C     CARRYING OUT SEVERAL CONVERSIONS
0051   200   NP=I-1
0052         X=FETCH*10.
0053         DX=DX*10.
       C
       C     CARRYING OUT CONCENTRATION CALCULATIONS AND PRINTING OUTPUT
0054         WRITE(7,210)
0055         WRITE(7,220)
0056         WRITE(7,230)
0057         IF(GRAMS) WRITE(7,240)
0059         IF (.NOT. GRAMS) WRITE(7,245)
0061         WRITE(7,250)
0062   210   FORMAT('0',20X,'CALCULATED AND MEASURED CONCENTRATIONS')
0063   220   FORMAT(20X,    ' **************************************')
```

```
FORTRAN IV      V02.5-2    Wed 14-Sep-83 12:50:42            PAGE 003

      230   FORMAT('0',5X,'HEIGHT',12X,'CALCULATED',12X,'MEASURED')
      240   FORMAT(5X,    ' (cm) ',12X,'  (g/m3)  ',12X,' (g/m3) ')
      245   FORMAT(5X,    ' (cm) ',12X,'  (mg/m3)  ',11X,' (mg/m3) ')
      250   FORMAT(5X,    ' ______',12X,'__________',12X,'________ ')
              DO 300,K=1,NP
                 IF(K.EQ.1) TRIP1=.TRUE.
      C          TRIP1 IS A LOGICAL VARIABLE FLAGGING THE FIRST TRIP
      C          TO THE SUBROUTINE CALCON
                 Z=ZC(K)*10.
                 E=D*Z**R
                 CALL CALCON(X,CCAL(K))
                 TRIP1=.FALSE.
                 WRITE(7,290)ZC(K)*100.,CCAL(K),CMS(K)
      290        FORMAT('0',5X,F6.1,12X,E10.3,12X,E10.3)
                    WRITE (2,37)CCAL(K), CMS(K)
      37         FORMAT ('RD',2G15.7)
      300     CONTINUE
      C
      C     CLOSING THE DATA FILE TO OPEN A NEW ONE
            CALL CLOSE (3)
            CALL CLOSE (2)
            GO TO 50
      C
            STOP
            END
```

```
FORTRAN IV      V02.5-2    Wed 14-Sep-83 12:50:50            PAGE 001

0001            SUBROUTINE CALCON (F,C)
        C
        C       SUBROUTINE FOR CALCULATING THE CONCENTRATION AT A CERTAIN
        C       POINT.  THIS IS CARRIED OUT BY APPROXIMATING THE AREA SOURCE
        C       BY NUMEROUS LINE SOURCES.
        C
        C       WHERE:  F= FETCH (meters)
        C               C= CONCENTRATION (RESULTING FROM ALL LINE SOURCES)
        C
0002              DIMENSION  XX(20),XH(20),DXL(20)
0003            LOGICAL DIAGN,TRIP1
        C         IN LOCATING THE CLOSEST SIX LINE SOURCES TO THE POINT OF INTEREST
        C         THESE ARRAYS ARE USED:
        C         XX(N)= DISTANCE FROM THE POINT OF INTEREST TO THE LINE SOURCE
        C         XH(N)= DISTANCE FROM POINT OF INTEREST
        C         DXL(N)=DISTANCE BETWEEN SUBSEQUENT XH's, [XH(N)-XH(N-1)]
        C
0004              COMMON/CHAR/ R,D,B,S,DX,E,TRIP1
        C
        C         SETTING THE DIAGNOSTIC FLAG
0005              DIAGN=.FALSE.
0006              IF (.NOT.DIAGN)GO TO 700
0008              WRITE(7,500)
0009    500       FORMAT('0THE FOLLOWING VALUES WERE TRANSFERED FROM THE MAIN
              1PROGRAM FOR F,R,D,B,S,DX,E :'//)
0010              WRITE(7,*)F,R,D,B,S,DX,E
        C
        C         THE FETCH IS DIVIDED UP INTO SMALLER INTERVALS WHERE K IS THE TOTAL
        C         NUMBER OF INTERVALS, AND DL IS THE DISTANCE BETWEEN THE INTERVALS.
0011    700       K=F/DX*4.0
0012              XK=K
0013              DL=F/XK
        C
        C         VERY CLOSE TO THE POINT OF INTEREST,SEVERAL SMALLER LINE
        C         SOURCES PLACED APPROPRIATELY ARE NEEDED.
        C         DIVIDING OUR SMALLER INTERVAL INTO 63 PARTS, OF LENGTH DXX:
0014              DXX=DL/63.
        C
0015              IF(DIAGN.AND.TRIP1)WRITE(7,800)
0017    800         FORMAT('0LOCATIONS OF LINE SOURCES AND THEIR CORRESPONDING
              1CALCULATED CONCENTRATIONS ARE:',//)
        C
        C         LOCATING THE FIRST SIX LINE SOURCES CLOSEST TO THE POINT OF
        C         INTEREST
        C
0018              DXL(1)=DXX
0019              XH(1)=DXX
0020              XX(1)=DXX/2.0
0021              DO 1000,J=2,6
0022                 DXL(J)=DXX*2.0**(J-1)
0023                 XH(J)=XH(J-1)+DXL(J)
0024                 XX(J)=(XH(J)+XH(J-1))/2.0
0025    1000        CONTINUE
        C
```

```
FORTRAN IV      V02.5-2    Wed 14-Sep-83 12:50:50             PAGE 002

C        INITIALLIZING THE BACKGROUND CONCENTRATION TO ZERO
        C=0.0
C
C        CALCULATING THE CONCENTRATION AT THE POINT OF INTEREST DUE TO
C        THESE FIRST SIX LINE SOURCES:
         DO 2000,J=1,6
            X=XX(J)
            EX=E/X
            IF(EX.LT.-300.)EX=-300.
            CON=B*X**(-S)*EXP(EX)*DXL(J)*1000.
            IF(DIAGN.AND.TRIP1)WRITE(7,*)CON,X,B,S,EX,DXL(J)
C
C           SUMMING THESE LINE SOURCES
            C=C+CON
            IF(DIAGN.AND.TRIP1)WRITE(7,1700)XX(J),CON,C
1700          FORMAT('0',20X,F10.7,15X,E10.4,15X,E10.4)
2000    CONTINUE
C
C       LOCATING ALL FURTHER AWAY LINE SOURCES,AND CALCULATING CONCENTRATIONS
        X=DL/2.0
        DO 3000,J=2,K
          X=X+DL
          CON=B*X**(-S)*EXP(E/X)*DL*1000.
          C=C+CON
          IF(DIAGN.AND.TRIP1) WRITE(7,1700)X,CON,C
3000    CONTINUE
        RETURN
        END
```

LIST OF REFERENCES

Banner M. L., and Melville W. K.. On the Separation of Air Flow over Water Waves, J. Fluid Mechanics, 77, pp. 825-842, (1976).

Banner, M. L., and Phillips, O. M.. On the Incipient Breaking of Small Scale Wind Waves, J. Fluid Mechanics, 65, pp.647-656, (1974).

Bindon, H.H.. A Critical Review of Tables and Charts used in Psychrometry, Department of Transport, Canada, pp 3- 13, (1963).

Brutsaert, Wilfred. A Theory for Local Evaporation (or Heat Transfer) from Rough and Smooth Surfaces at Ground Level. Water Resources Research, 11,4, 543-550, (1975).

Chamberlain, A. C.. Transport of Gases to and from Surfaces with Bluff and Wave-like Roughness elements. pp. 318-332, (1968).

Chang, Po-Cheng. Laboratory Measurements of Air Flow Over Wind Waves. Dissertation, Colorado State University, (1968).

Easterbrook, C. C.. A Study of the Effects of Waves on Evaporation from Free Water Surfaces. Water Resources Technical Publication, Research report no. 18, United States Dept. of the Interior, Bureu of Reclaimation, (1968).

Jeffreys, H.. On the Formation of Water Waves by Wind. Proc. Roy. Soc. A110, 341, (1925).

Kondo, J. , Fujinawa, Y. , Naito, G.. High-Frequency Components of Ocean Waves and their Relation to the Aerodynamic Roughness. J. Physical Oceanography, 3, pp. 197-202, (1973).

Lai, J., Plate, E. J.. Evaporation from Small Wind Waves. Technical Report CER68-69JRL35, Colorado State University (1969).

Owen, P. R., and Thomson, W. R.. Heat Transfer Across Rough Surfaces Fluid Mechanics, 15, pp.321-334, (1962).

Pasquill, F.. Atmospheric Diffusion. Wiley and Sons, New York, (1974).

Sirovica, S.. Transfer of Volatile Chemicals Across the the Air-Water Interface under different Wind Conditions. M. S. Thesis, Civil Engineering Dept, Michigan State University,(1982).

Stewart, R. W.. Mechanics of the Air-Sea Interface. J. Fluid Mechanics, Physics of Fluids Supplement,pp. S47-S55, (1967.)

Street, R. L.. Turbulent Heat and Mass Transfers Across a Rough, Air-Water Interface: A Simple Theory. Int. J. Heat Mass Transfer, 22, pp. 885-899, (1979).

Tanner, C. B.. Application of Psychrometry in Micrometeorology. Precision Humidity Analysis, E.G.&G. Environmental Equipment Division, Part VI,(1973).

Wu, J.. Laboratory Studies of Wind-Wave Interactions. J. Fluid Mechanics,34, 91, (1968).