PNL-6580
UC-261

15 F

The PNL Single-Tower Measurement Model of Rotationally Sampled Turbulent Wind, with User's Guide for STRS2PC

J. R. Connell D. C. Powell
V. R. Morris G. L. Gower

June 1988

Prepared for the U.S. Department of Energy
under Contract DE-AC06-76RLO 1830

Pacific Northwest Laboratory
Operated for the U.S. Department of Energy
by Battelle Memorial Institute

Battelle

PNL-6580

PACIFIC NORTHWEST LABORATORY
operated by
BATTELLE MEMORIAL INSTITUTE
for the
UNITED STATES DEPARTMENT OF ENERGY
under Contract DE-AC06-76RLO 1830

Printed in the United States of America
Available from
National Technical Information Service
United States Department of Commerce
5285 Port Royal Road
Springfield, Virginia 22161

NTIS Price Codes
Microfiche A01

Printed Copy

Pages	Price Codes
001-025	A02
026-050	A03
051-075	A04
076-100	A05
101-125	A06
126-150	A07
151-175	A08
176-200	A09
201-225	A010
226-250	A011
251-275	A012
276-300	A013

PNL-6580
UC-261

THE PNL SINGLE-TOWER MEASUREMENT MODEL OF ROTATIONALLY SAMPLED TURBULENT WIND, WITH USER'S GUIDE FOR STRS2PC

J. R. Connell
V. R. Morris
D. C. Powell
G. L. Gower

June 1988

Prepared for
the U.S. Department of Energy
under Contract DE-AC06-76RLO 1830

Pacific Northwest Laboratory
Richland, Washington 99352

SUMMARY

Wind turbines operating in real winds are subject to the stresses induced by wind turbulence. Assessment of the effects of turbulence is complicated by the fact that the turbine rotor is moving through a plane perpendicular to the wind flow, and thus the rotor experiences more variation in wind speed than a stationary point would. A means of evaluating the effects of turbulence on a rotating turbine rotor is available as a single-tower rotationally sampled wind model, STRS-2, which provides estimates of unmeasured turbulence characteristics in the crosswind plane that spans the disk of the rotor based on measurements made economically from conventional single-tower arrays of anemometers.

The theory behind STRS-2 has been published in several papers; the theory is condensed here for the benefit of the STRS-2 user. Results of model application to three wind turbine sites (for which anemometer data measured in a vertical plane are also available) are compared to illustrate how the model can be applied.

In particular, this report provides a guide to using a desktop computer version of the model, STRS2PC, developed to make the capability of estimating turbulence effects available to potential users as easily as possible. (The STRS2PC program is available from the National Energy Software Center, Argonne, Illinois.)

CONTENTS

FIGURES

TABLES

1.0 INTRODUCTION

This report describes a single-tower rotationally sampled wind model, STRS-2, that approximates a set of time series of turbulent wind experienced by individual points rotating in circles in a crosswind plane using measurements from anemometers arrayed vertically along a single line. The purposes of the model are (1) to use turbulence measurements made economically from conventional single-tower arrays of anemometers, (2) to incorporate measured characteristics of the wind at specific sites under consideration for operation of wind turbines, spanning the height range of interest, and (3) to estimate the unmeasured turbulence characteristics in the crosswind plane that spans the disk of the rotor blades.

The model, originally written for execution on a mainframe computer (Connell and George 1983b), has been found to be the best measurement-based tool for use in wind turbine tests at sites that are remote, that have rough terrain, or where the wind turbines are very large. The model is useful where neither multi-tower vertical plane arrays of anemometers nor turbulence measurements actually following points on the rotating blades can be undertaken. Previous uses of STRS-2 have been reported for the MOD-OA wind turbine at Clayton, New Mexico (Connell and George 1983b) and the WTS-4 wind turbine at Medicine Bow, Wyoming (Connell, Morris, and Hinchee 1987). The mainframe computer version of the STRS-2 model has been available for about 4 years. This report makes STRS-2 available in an interactive form on a desktop computer. The program, STRS2PC, was developed on an IBM-PC™ and can be operated on PC-compatible equipment.

Use of the computer program presumes the availability of appropriate time series of measured turbulent wind from at least two levels on a single vertical line (i.e., a meteorological tower). It also assumes that the user can correctly interpret the new time series that the model generates from the measured time series and can estimate the accuracy of the results. Part of this report is intended as a guide in developing these skills, in particular in relation

™ International Business Machines, Inc., Boca Raton, Florida.

to the spectra, without presuming to provide the complete information or training required.

The report is organized in the following fashion. Section 2 describes the physical principles upon which the rotationally sampled wind is found to be useful for wind turbine analyses. It also describes the physical basis for formulating STRS-2. Section 3 shows an example of the use of the STRS-2 model to approximate time series of turbulent winds experienced concurrently by a set of points spaced along a rotating radial line in a vertical plane. Section 4 discusses the power spectral density functions corresponding to each of the modeled wind speed time series from Section 3. No spectral analysis code is included, since it is not a part of the model. The sources and magnitudes of errors in rotationally sampled wind speed spectra computed using the STRS-2 model are discussed in relation to the more exact rotationally sampled wind speeds obtained using a crosswind circular array of anemometers. An expansion of the error analysis, and suggested correction factors for the rotational spectrum that characterizes the time series, are given in Appendix B.

A flow chart of the STRS2PC computer program and the input and output from a sample run are discussed in Section 5. Particular emphasis is given to the interpretation and proper use of the prompts in the program. The complete source code for the STRS2PC time series model is included in Appendix A.

An example of the use of the STRS-2 analysis of turbulent wind in calculation of a flatwise root bending moment response function of a wind turbine rotor blade is discussed in Section 6. Concluding remarks and cautionary guidance about the use of STRS-2 and other turbulent wind models are made in Section 7.

2.0 THE PHYSICAL PRINCIPLES OF THE ROTATIONALLY SAMPLED WIND

2.1 A COMPLETE WIND FIELD

In simplest form, the physical principles that are important in determining the characteristics of the turbulent wind experienced by a rotating wind turbine blade are not complicated. First and foremost, the blade rotates in a crosswind plane with a speed that, at most radii, is significantly greater than the average speed of the wind relative to the turbine tower (Connell 1982). Second, the turbulent wind contains a complex structure of velocity gradients throughout the disk of rotation (Connell 1980, 1984, 1985). Figure 1a indicates the crosswind flow field that a turbine rotor may experience at one instant of time. The wind structure usually changes radically over periods of time, from a fraction of the blade's rotation period to many times the rotation period (Connell 1982, 1984). In order to include all of the effects of the wind on the rotor, it is necessary to know the wind velocity at all radial locations simultaneously for all of its blades, with a time resolution in the rotating frame of reference corresponding to the highest frequency of possible interest in the response of any component of the wind turbine system.

There is a coherence to the turbulent wind field variation (Panofsky and Dutton 1984). The character of the coherence may be different for different directions and for different wind conditions (Connell et al. 1985; Connell and George 1983a). An alternative way to picture the complex structure of the turbulent wind being intercepted by the rotating blade is to visualize a spatial ensemble of loosely formed complex eddy flow structures of different sizes and shapes convecting through the disk of rotation (Connell 1980, 1982, 1984, 1985). The alternative view is suggested schematically in a highly simplified form in Figure 1b. The wind velocity variations seen by the rotor as it passes through the eddies determine the spectral character of the wind at the rotor blade (Connell 1980, 1982, 1984, 1985). Ultimately, these variations affect the spectral character of the aerodynamic forces on the blade. These concepts are discussed in more detail in references cited in this and the previous paragraph. To account for the important effects of the coherence

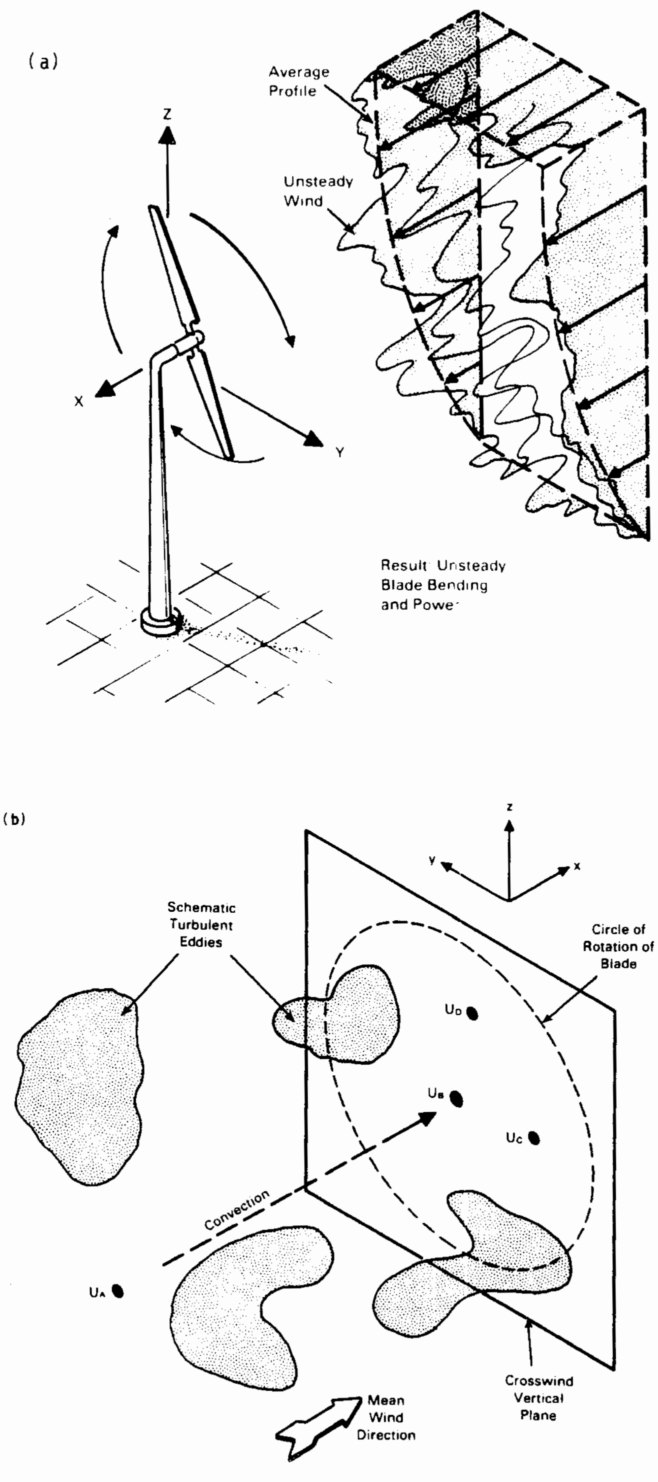

FIGURE 1. Schematic Diagrams of the Turbulent Wind Experienced by a Wind Turbine Rotor. (a) An instantaneous wind velocity gradient interpretation. (b) An interpretation of turbulence as a field of coherent turbulent eddies.

of the turbulent wind, the spatial and time variation of the wind must be accounted for with a resolution as described at the end of the preceding paragraph.

It has been noted above that an accurate wind model for a wind turbine rotor in motion must include the variations of the wind velocity in space and time across the whole disk of the rotor. Whether the rotational aspect of the character of the wind at the blade is included as part of the wind model or is obtained later during application of the aerodynamics model of the rotor is not of physical importance. Two significant effects of including the rotational aspect with the wind model, rather than later, are that (1) such a wind model can be much more efficient and economical (Connell 1982; Powell and Connell 1986a, 1987), and (2) the important features of the wind that act with special character on the rotating blade are directly observable in the modeled wind (Connell and George 1983a, 1987, 1988). There may be some situations for which the rotationally sampled version of the wind is not the best model even with its advantages. However, at this time single-tower simulation of wind throughout a crosswind disk has only been done in the rotational sense.

2.2 THE STRS-2 APPROXIMATION TO THE COMPLETE WIND FIELD

In Section 2.1, the complete wind field that affects the wind turbine rotor was discussed. A wind model that is practical to use will generally not be as complete as would be desired. For example, theoretical models exist (Powell and Connell 1986b) that produce a very detailed description of the wind field in the disk of a rotor. However, they are based upon simplifications of the assumed wind field such as homogeneous, isotropic turbulence and "frozen" turbulence. Many wind fields in which wind turbines must operate are not so simple. To correct for this defect, arrays of anemometers in a crosswind plane (vertical plane arrays or VPAs) are occasionally used to measure the wind field. It is not practical to install enough anemometers in a VPA to measure the whole wind field with the required time and spatial resolution. However, a valuable approximation of the wind field, which incorporates the actual wind characteristics of a site and of the wind throughout at least one crosswind annulus of the rotor disk, has been achieved with a VPA (Connell and George 1983a). Even this simpler version of a VPA was difficult to maintain and operate. It would be prohibitively difficult and expensive to establish a VPA for a large wind turbine, especially in complex terrain. A need for a still simpler method requiring fewer measurements of the site-specific winds to simulate wind throughout a crosswind disk led PNL to devise STRS-2 (Connell and George 1983b). A brief explanation of the physics of the wind features used in an STRS-2 simulation of a crosswind field of the wind at a rotor is given in this subsection.

The starting concept for the STRS-2 model is of a turbulent wind composed of many turbulent elements of finite size flowing through the crosswind plane of interest, as was described in Section 2.1. If crosswind measurements of this wind field were made, they would reveal crosswind correlation values in the horizontal and vertical directions of wind velocity fluctuations that suggest an average eddy width for each of those dimensions of turbulence elements. Similarly, correlation in the along-wind direction would indicate an average eddy length in that direction. The correlations represent a partial holding together of a finite entity of the flow field that will be called, loosely, an eddy. The three lengths in the three directions give us some idea of the average shape and size of turbulence eddies. These lengths also give us some idea about a relation between the coherence in the crosswind direction of wind velocity fluctuations and the coherence in the along-wind direction of the same type of fluctuations. Since for simplicity the STRS-2 model input is limited to measured wind velocity at points along a single tower, the actual wind measurements are only of time variations and vertical spatial variations. The idea extrapolated from previous measurements in all three spatial directions (Panofsky and Dutton 1984) is used to stretch the information available from the vertical line of measurements at the tower into a model of wind throughout a three-dimensional volume.

To construct the single-tower model of turbulent wind at a rotor disk, we build on several conceptual steps that do not all appear explicitly in the STRS-2 program. First, we transform the time variation of wind velocity at selected points along the tower into a model of time variation of wind velocity at selected points in the longitudinal, or along-wind, and vertical dimensions. We then make a further transformation to a model that retains a variation in the vertical dimension and further transforms the variation in the along-wind direction to represent a variation of wind velocity at points around a circular path in the crosswind horizontal and vertical dimensions. This is discussed in Section 2 of a paper by Connell and George (1983b). Figure 2a is a reproduction of Figure 2 from Connell and George's paper: It gives a schematic of three conceptual arrays in the along-wind vertical plane around which one can simulate wind at each array anemometer location, using wind values measured by anemometers in a single-tower vertical array. The two arrays in the along-wind vertical plane, an imaginary circle and an ellipse, contain symbols showing the location of imaginary anemometers whose wind measurements would be simulated in the first conceptual stages of development of the STRS-2 model.

Figure 2b shows how the simulated ellipse of measurement locations in the along-wind plane, shown in Figure 2a, represents anemometer locations along the required circle in the crosswind plane. Figure 2b is a schematic of the last conceptual stage in the formulation of the STRS-2 model. The actual process of fitting measured wind time series at the single tower to the simulated circular array involves equating wind values measured at one time to those of hypothetical measurement elsewhere at a different time. Specific measured wind values are assigned to different times than those at which they would be observed at the tower in order to correspond to simulated winds at equally spaced locations around the ring arrays.

An elliptical array of measurements in the along-wind plane is simulated by assuming that if measurements could be made at specific sequential times at equally spaced points around the circle, they would be identical to the measurements made at the actual anemometer locations on the single tower at specific earlier and later times. This lead and lag approach to building a simulated time series of wind speeds from a set of real measured time series assumes that turbulence is made up of a field of fixed spatial variations and that the time variation of wind velocity at a point is caused by convection of the "frozen" spatial gradients associated with the turbulent elements moving along with the speed of the short-term mean wind.

The actual leading and lagging done by the computer program transforms wind measurements at the single tower to a hypothetical elliptical path whose amount of ellipticity must be selected to give the best possible simulation of wind at a crosswind circle. For example, a large along-wind stretch of

(a)

(b)

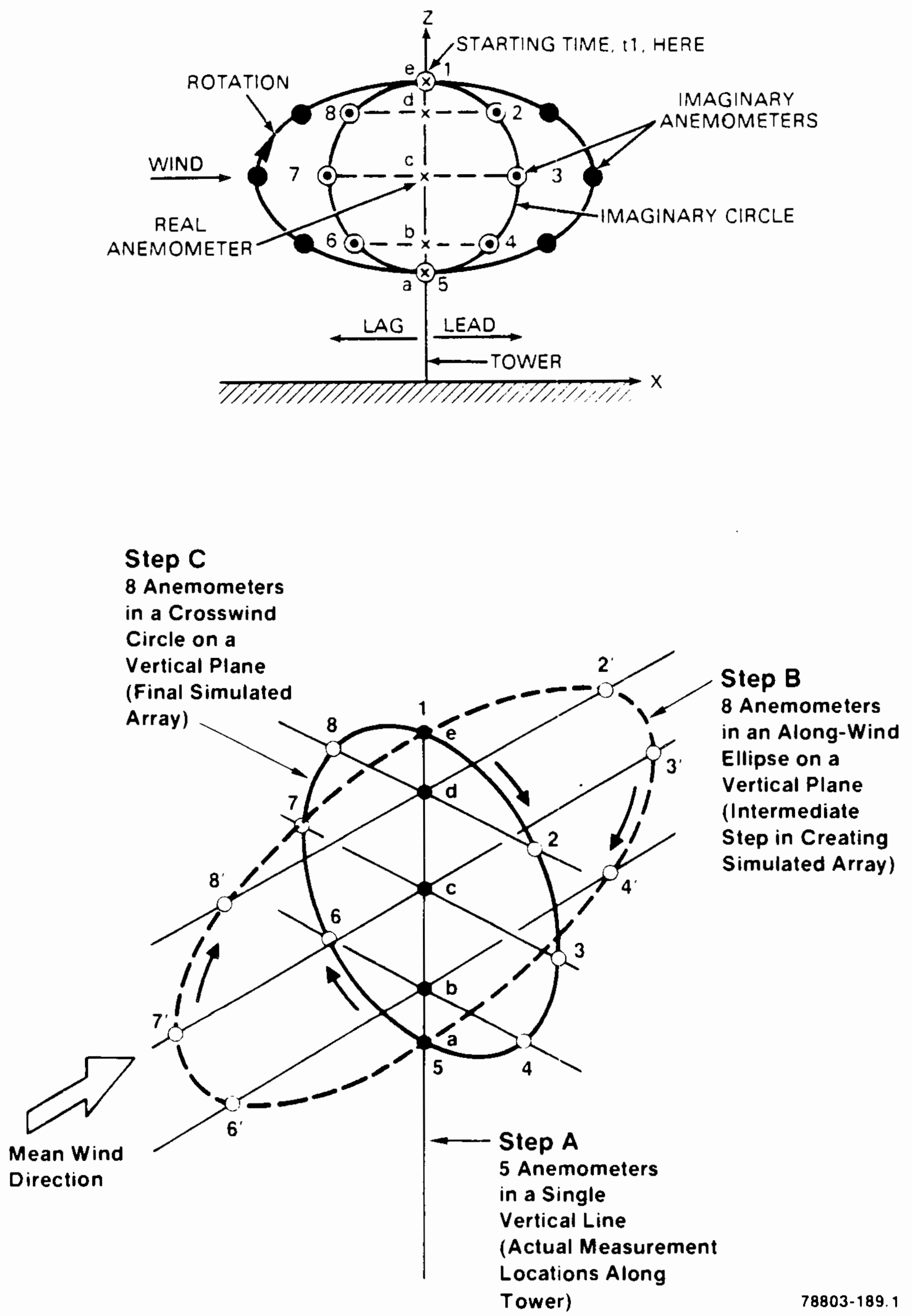

FIGURE 2. Hypothetical Arrays of Anemometers in the Vertical-alongwind Plane Used in Stage One of Development of the STRS-2 Model (Connell and George 1983b): (a) Cross-Plane View; and (b) Oblique View, Showing the Transformation of Winds at Locations in an Along-wind Elliptical Array to Locations in a Crosswind Circular Array.

the ellipse accounts for wind conditions in which the average turbulent eddies are "self-connected," or coherent, entities over a much longer spatial extent in the along-wind direction than in the crosswind directions. The model assigns to this ratio of turbulent length scales (Lu_x/Lu_y) a value of 2.5 (Counihan 1975). The limits of the model's ability to correctly simulate this feature of turbulence will be discussed in Section 7. It is this coherent relation between the crosswind and along-wind spatial distribution of the wind fluctuations that in part determines the unusual spectral characteristics of the wind that forces the rotor fluctuations.

The conceptual steps in modeling the time variation of the wind velocity at a set of points uniformly spaced around a circle in a crosswind vertical plane start with a set of time series measured at fixed points along a single tower. The STRS-2 program actually selects, from the full set of time series from all of the actual anemometers, only a small subset of the data for use in the output time series. It does this by rotationally sampling through the tower data in a specific way to generate a single time series for a point moving around a circle having a specified radius and frequency of revolution around the circle.

If time series are required for several different radii, the STRS-2 program will generate each additional time series required, deriving each one from the same original time series measured by the tower anemometers. The resultant set of simultaneous rotational velocity time series representing the wind at different radii along a rotor blade have an element of reality in their radial coherence and phase relations. As long as the original wind data used by STRS-2 are measured, real-wind velocity time series (it would be possible to use synthetic data as input to the model), the phase relation between wind velocity at any two radii selected will have the maximum element of reality that the STRS-2 approach can provide.

An example of a set of STRS-2 time series for six equally spaced radii from 0.0 R to 1.0 R, where R is the tip radius of the rotor, for a specific wind turbine configuration is shown and discussed in Section 3.

3.0 AN EXAMPLE OF STRS-2 WIND SPEED TIME SERIES

The most complete set of single-tower turbulent wind time series measurements in simple terrain that we have analyzed come from the PNL tower at Medicine Bow, Wyoming (Connell, Morris, and Hinchee 1987). This tower had five equally spaced, cup and vane anemometers spanning the height of the rotor disk of a large wind turbine (the WTS-4). The anemometer array on the tower is shown in Figure 3a. A short segment from simultaneous wind speed time series from each of the anemometers is shown in Figure 3b. The data presented are for a statically unstable case collected on April 24, 1986, at 11:30 MST. Time series, of which these are small portions, were transformed using STRS-2 into rotational time series modeling the wind experienced by points rotating around crosswind rings at the rotation rate of 0.5 Hz.

Segments of the rotational time series for six equally spaced radial locations are shown in Figure 4. Obtaining time series for each of five concentric circular paths having high time resolution out of the original time series from fixed anemometers involves several additional features of the STRS-2 model not previously discussed. First, the vertical spacing of the anemometers on the tower is not arranged so that their lateral projections intersect a circle at a set of uniformly spaced points around the circle. Second, a large number of equally spaced points on the circle (20 samples per rotation) are required to provide the time resolution required for the time series of each rotating point (10 samples per second); however, there are only five anemometers on the tower. This means that data must be simulated corresponding to heights of all points around the simulated ring. This simulation is achieved for the correct nonlinear vertical spacing and large number of simulated points using an interpolation scheme. Extending the vertically interpolated data to the points on the simulated ring is done by a lead or lag process and time interpolation. The details of the interpolation are discussed briefly in Section 7 in relation to the accuracy and limitations of the STRS-2 model.

A further analysis of time series is not intended in this report. However, it is recommended that the reader study the time series presented in Figure 4 with the objective of noting similarities and differences among the

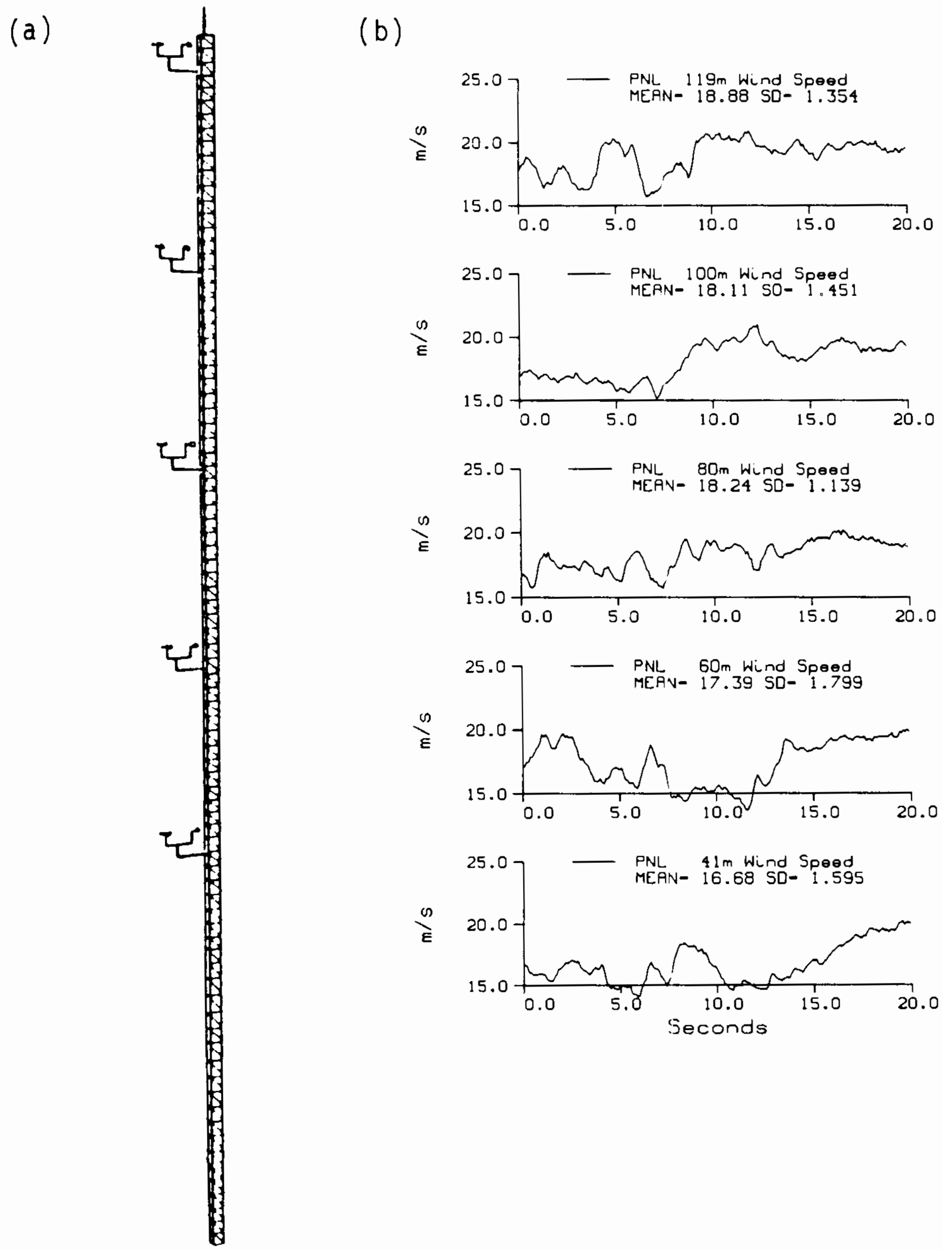

FIGURE 3. Illustration of Wind Data from Medicine Bow: (a) Configuration of the Anemometer Array on the PNL Turbulence-measuring Tower; and (b) Sample Wind Speed Time Series Corresponding to Each of the Anemometer Heights in Figure 3a. The speeds are in meters per second and the times are in seconds.

time series at different radial distances from the axis of rotation. These differences will be explored from a spectral point of view in Section 4. One obvious difference between the time series is the decrease of the low frequency amplitude and the greater loss of high frequency content of the time series with decreasing radius. Note that the sixth time series is at a radius value of zero, at which the rotation speed is zero and the rotational time series is the same as the fixed-point or Eulerian time series at the hub height, 80 m, of the wind turbine (see Figure 3b).

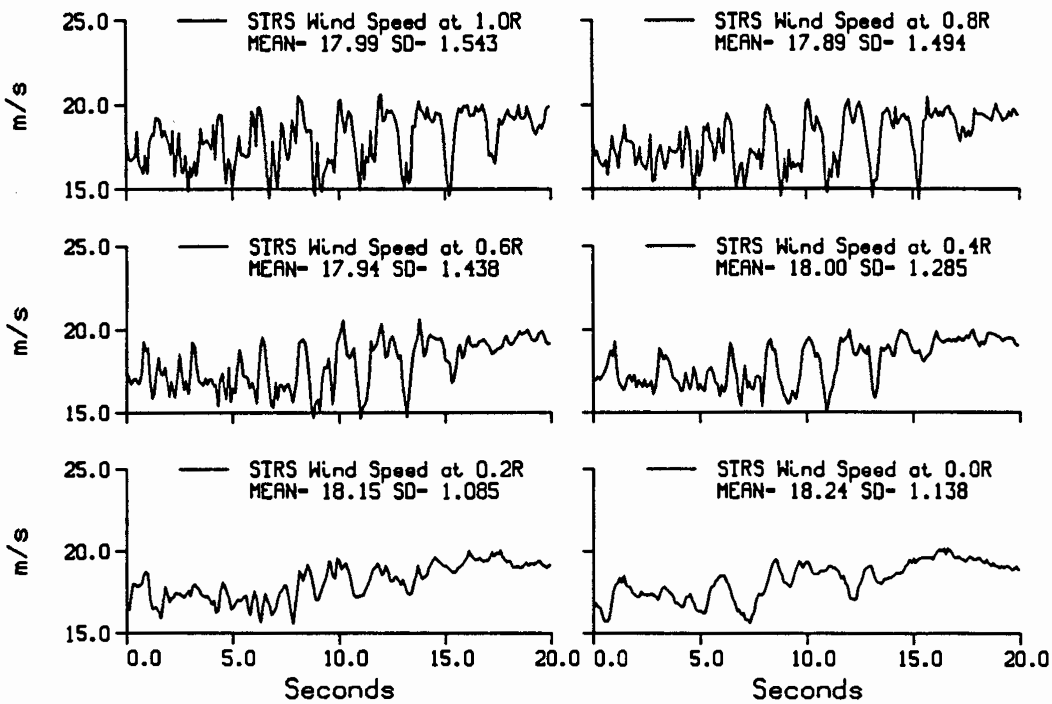

FIGURE 4. Rotationally Sampled Wind Speeds at Six Equally Spaced Radial Locations Derived for 10 rotations of the WTS-4 Turbine Using the STRS-2 Model. The radii are in units of R, the tip radius of the rotor.

4.0 A SPECTRAL CHARACTERIZATION OF THE STRS-2 WIND SPEED

The power spectral density curves corresponding to the time series at six radial locations generated for the previous section by the STRS-2 model shown in Figure 4 are shown in Figure 5. The reader is encouraged to study the differences and similarities among these six spectra. It does appear that the STRS-2 model, at least qualitatively, gives a believable representation of the wind seen by points revolving around the circles.

To assess the accuracy and completeness of STRS-2, we have used it with wind measurements from the center tower of VPAs of anemometers. The resulting spectra have been compared with the spectra for the concurrent rotationally sampled wind from the full crosswind circle array of anemometers. The most comprehensive set of comparisons comes from the Clayton, New Mexico, 7-tower VPA measurements (Connell and George 1983a; George and Connell 1983). A set of spectra for three distinct cases, shown in Figure 6, may be compared visually. A more quantitative and objective comparison is shown in Figure 7.

The graph (Figure 7) shows curves for 12 cases at Clayton, New Mexico, the ratio of the turbulence variance, σ_i^2, of the full rotationally sampled wind to the STRS-2 rotationally sampled wind in each harmonic band[a] of the spectra. The curves are a type of transfer function. If the function has a value of 1 then the STRS-2 variance is identical to that for the full-circle rotationally sampled wind spectrum. The values plotted may be used as correction factor values, f, for variance of each harmonic band computed by the STRS-2 method, such that if only the STRS-2 band variances were known, the better full VPA values would be estimated.

The 8 digits in each test name on Figure 7 and in Table 1 indicate test date and time. For example, 01011225 means the test period began on January 1 at 12:25 p.m.

Not enough examples have been observed to give confidence in the correction factors that we have obtained to correct STRS-2 data obtained for any

---

(a) A harmonic band is the band of frequencies from $n_i - 0.5\ n_1$ to $n_i + 0.5\ n_1$, where n_1 is the rotation frequency of a rotor blade and n_i is one of the multiples for harmonics of n_0 and i is the number of the harmonics 2,3,4,5....

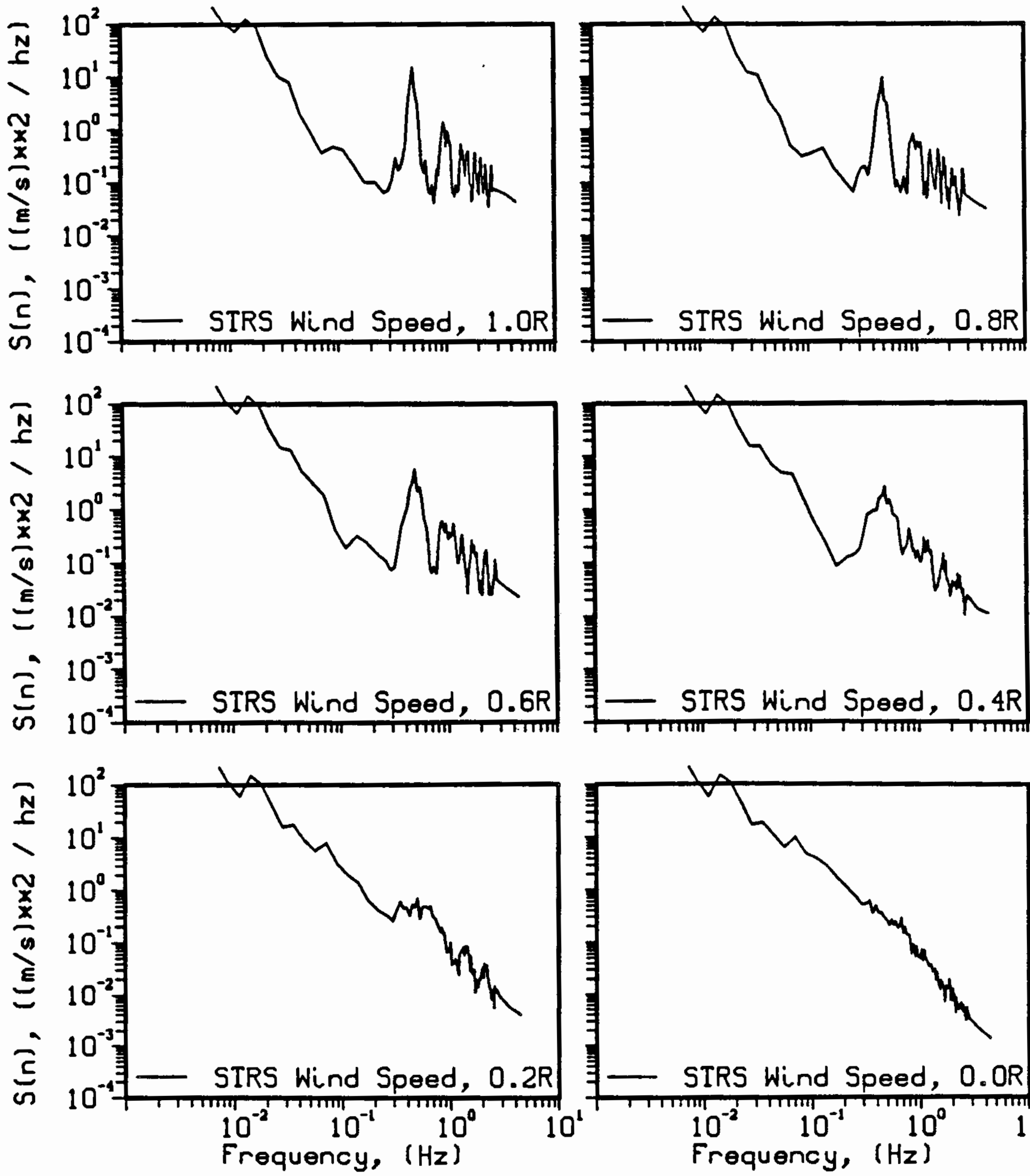

FIGURE 5. Rotationally Sampled Wind Spectra Derived from 14 min of the Time Series Sampled in Figure 4. The spectral density on the ordinate and the spectral frequency on the abscissa are plotted logarithmically, in the units meters squared per second Hertz and Hertz, respectively.

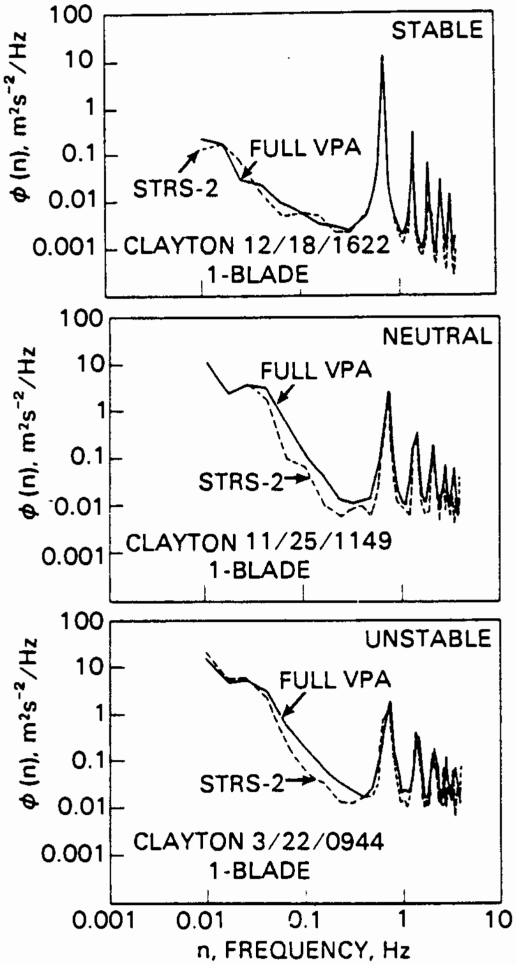

FIGURE 6. A Comparison of Rotational Spectra from the STRS-2 Method and the Full Vertical Plane Array Method for Three Cases at Clayton, New Mexico. The spectra are plotted with scales and units as in Figure 5.

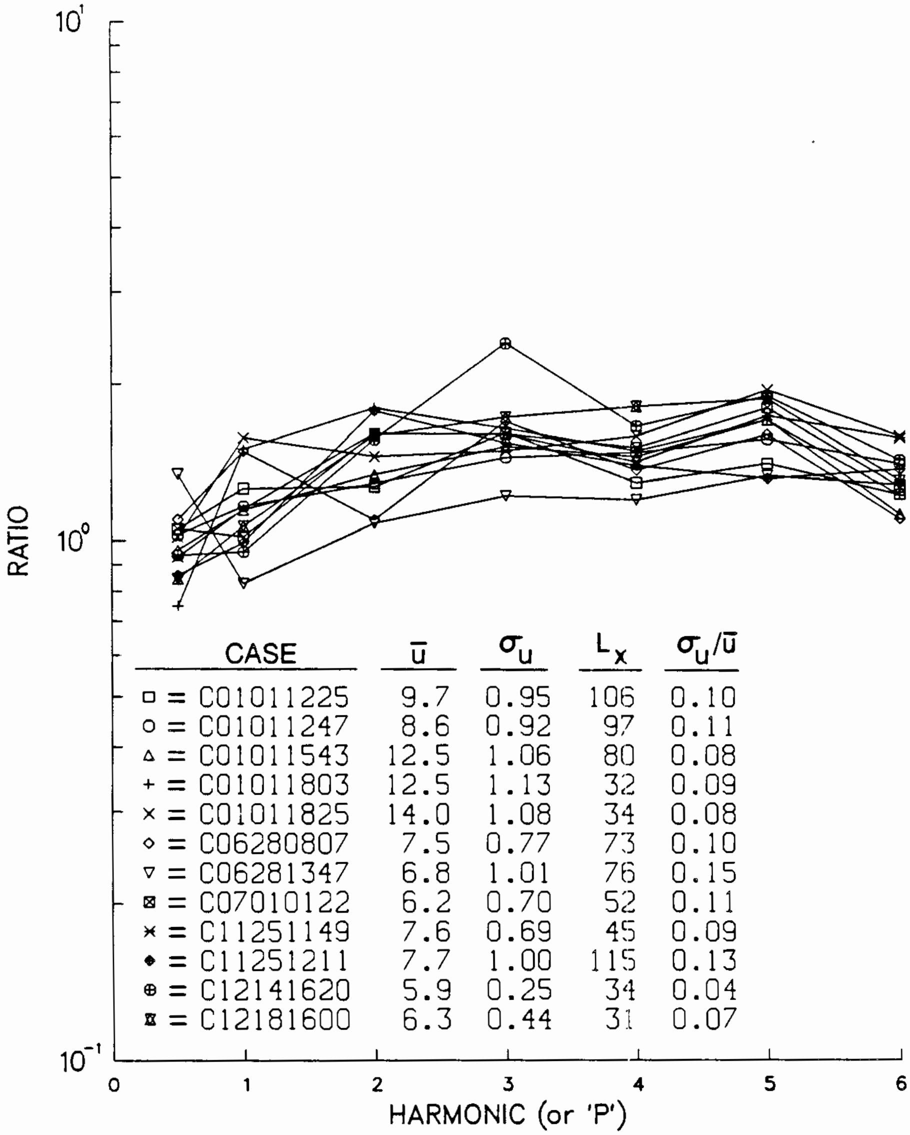

FIGURE 7. Correction Factors for STRS-2 Harmonic Band Variance for 12 Cases at Clayton, New Mexico. The reference for the correction is the band variance values from the full-circle vertical plane array (see Section 4.0 and Appendix B).

TABLE 1. Correction Factor (f) Data for STRS-2 Harmonic Band Variances Compared to Vertical Plane Array Corresponding Variances of the Rotationally Sampled Wind

Clayton, New Mexico, MOA-OA Turbine, 12-Anemometer Circle

Case	$f_{1/2P}$	f_{1P}	f_{2P}	f_{3P}	f_{4P}	f_{5P}	f_{6P}
C01011225	1.05	1.26	1.27	1.62	1.29	1.41	1.23
C01011247	1.03	1.16	1.29	1.45	1.48	1.57	1.41
C01011543	0.96	1.15	1.34	1.52	1.45	1.72	1.13
C01011803	0.75	1.50	1.80	1.65	1.49	1.72	1.23
C01011825	1.02	1.58	1.45	1.49	1.60	1.96	1.60
C06280807	1.10	1.49	1.10	1.70	1.37	1.61	1.11
C06281347	1.34	0.83	1.08	1.22	1.20	1.34	1.29
C07010122	1.05	1.02	1.61	1.61	1.51	1.81	1.27
C11251149	0.93	1.15	1.61	1.60	1.42	1.75	1.59
C11251211	0.86	0.99	1.78	1.55	1.40	1.32	1.38
C12141620	0.94	0.95	1.57	2.40	1.67	1.91	1.44
C12181600	0.84	1.07	1.59	1.73	1.82	1.89	1.31

Altamont, California, NPS Turbine, 8-Anemometer Circle

Case	$f_{1/2P}$	f_{1P}	f_{2P}	f_{3P}	f_{4P}
A08202246	1.09	1.00	0.73	0.56	0.46
A08202304	0.92	0.97	0.70	0.61	0.46
A08202322	0.95	0.90	0.73	0.64	0.45
A08211750	1.12	0.82	0.70	0.60	0.51
A08211805	1.18	0.77	0.78	0.69	0.52
A08211820	1.17	0.85	0.73	0.56	0.40

San Gorgonio, California, Howden Turbine, 6-Anemometer Circle

Case	$f_{1/2P}$	f_{1P}	f_{2P}	f_{3P}
H09111750	1.01	1.06	1.24	1.00
H09112100	1.05	1.06	1.28	0.77
H09141145	0.98	1.01	1.15	0.67
H09160725	0.84	0.89	1.17	0.73
H09161105	0.97	1.06	1.33	0.71
H09161600	0.98	1.20	1.34	0.84
H09232235	0.99	1.17	1.41	0.70
H09271310	0.99	1.12	1.28	0.60
H10011610	1.05	1.12	1.53	0.87
H10111338	1.05	1.11	1.09	0.70
H10170140	1.00	1.21	1.17	0.76
H10192010	0.97	0.94	1.25	0.85

other site or wind condition. However, the correction factor data that we have measured at available sites are included in Table 1, to be used at the discretion of users of the STRS-2 model. An extended discussion of STRS-2 errors and correction factors computed for all available test cases done by PNL is given in Appendix B.

This completes the background and analytical discussion of the STRS-2 model of the turbulent wind experienced by a rotating wind turbine rotor. The model code as it has been reconfigured for use on a desktop computer is described and discussed in Section 5. An example of its use in explaining the flatwise bending response of a wind turbine rotor blade is presented in Section 6.

5.0 A USER'S GUIDE AND FLOW CHART OF THE STRS2PC PROGRAM

The purpose of this section is to introduce the reader to the desktop computer program that performs the STRS-2 model's simulation of the rotationally sampled wind. To this end, a flow chart providing an overview of the program is presented in this section. The complete source code is printed in Appendix A. A sample of the input parameter values used in a test run and the corresponding output is shown. Illustrative graphs of output rotational time series of wind speed (and the corresponding spectra not computed by the program) are also shown. Some discussion of technique in running the program and of the time required to complete a simulation on a desktop computer conclude this section.

5.1 FLOW CHARTS FOR THREE SELECTED MODULES: STRS2PC, WSDATA, AND STRS

The STRS-2 computer model is most concisely represented by three program modules: STRS2PC (the main program that initializes variables and manages files), WSDATA (the subroutine that assigns and writes the output data), and STRS (the function that computes the STRS-2 wind speed). These modules have been made into the flowcharts shown in Figure 8. Names of subroutines called are given in parentheses.

5.2 USER'S GUIDE FOR PROGRAM STRS2PC

Program STRS2PC models the wind encountered by a point on a rotating wind turbine blade using actual wind speeds measured by anemometers on a single tower. The user is prompted for several parameters concerning the wind turbine to model and the type of output desired. The measured wind is read from a computer file.

5.2.1 Program Operation

STRS2PC.EXE is the executable image of the program and the only file (other than the wind speed data file) needed for program operation.

To execute the program, type STRS2PC.

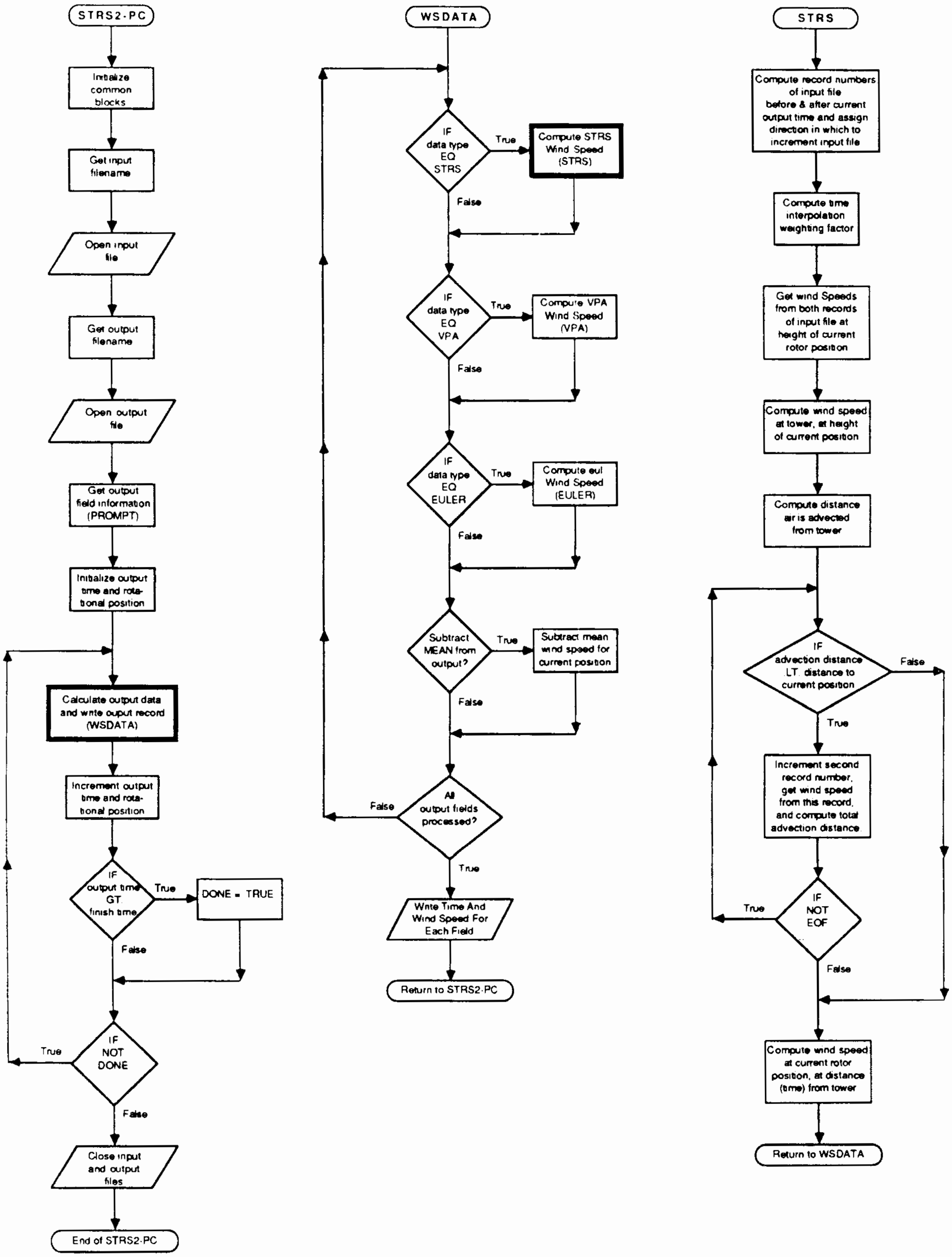

FIGURE 8. A Flow Chart of the Computer Program for the STRS-2 Model

5.2.1.1 Input

The input file must contain wind speed data measured at levels that span the rotor disk of the wind turbine to be modeled. The file must contain at least two levels of wind data. The accuracy of the modeled wind improves as the number of anemometers within the dimensions of the rotor disk increases. The wind speed at each level must be either the total, horizontal wind speed at that height or the horizontal speed in a particular component of interest (e.g., normal to the plane of the rotor disk). The wind speed must always be positive; that is, in the direction of the mean wind speed. This is a restriction that will rarely be important at wind speeds above the cut-in level of the wind turbine. The program expects two ASCII, formatted input files: a sequential header file and a direct access data file. The first record of the header file contains month, day, hour, minute, second (the starting time of the file), data acquisition system identifier (used with subsequent programs at PNL), total number of channels (including wind and other measurements to a maximum of 100), and digitization interval in seconds and is read using FORMAT(1X,2I2,1X,3I2,1X,I1,1X,I3,1X,F9.4). The next number-of-channels records of the header file contain channel number, channel identification, and units of each input channel using FORMAT(1X,I3,A30,1X,A20). The number-of-scans records in the data file contain elapsed time in seconds (after the starting time) and data for each channel using FORMAT(1X,F9.2,100(F8.3,3X)). All records of the data file must be of the same length. This length must be 11*(number-of-channels + 1).

Following is the list of questions, in the order in which they are prompted, that the user must answer when the program is run:

Enter the input file name:

Respond with the name of the input file containing the wind speed data. This must be a string of up to 48 characters.

Is this the desired file (y or n) ?

Respond whether the correct input file was selected based on the displayed file information.

Enter the output file name :

Respond with the name that the user wants the output file to be called. This must be a string of up to 48 characters.

Enter the number of output fields desired :

Respond with the number of fields or columns that will be written to the output file up to a maximum of 50. This number depends on the different types of output desired. There are three optional outputs available: the modeled rotational wind speed (STRS-2), the measured rotational wind speed (VPA or HPA), and the measured Eulerian wind speed. Each of these can have several different variations depending on radial location, anemometers used, mean wind removal, and filtering. This must be a positive integer less than 50.

What type of data do you want in output field n ?
1) Single Tower Rotational Sampling
2) Circular array rotational sampling
3) Eulerian wind speed
Select one of the above :

Respond with the type of output desired in the nth output field. Select '1' if the modeled rotationally sampled wind from anemometers along a single tower is desired. Select '2' if the rotationally sampled wind from anemometers in a circular array is desired. Select '3' if the Eulerian wind from a single anemometer or averaged over several anemometers is desired. This must be a positive integer.

The following prompts and descriptions assume that the user selected '1' for the first output field.

Enter hub height of wind turbine in meters :

Respond with the hub height of the wind turbine to model.

Enter rotor radius of wind turbine in meters :

Respond with the rotor radius of the wind turbine to model.

Enter radial location of analysis
(i.e. 1.0 = 1.0*radius, 0.5 = 0.5*radius, etc.) :

Respond with the distance along the rotor, in multiples of its radius, at which the wind will be modeled for the nth output field. This must be a positive number less than or equal to 1.0.

Enter rotation rate of wind turbine in rpm :

Respond with the blade rotation rate of the wind turbine to model.

Enter number of positions on circle to rotationally sample :

Respond with either the number of points around the circle of blade rotation for which to model the wind speed or a value equal to (60 / blade rpm) / (desired digitization interval in seconds of output). This must be a positive integer less than 100.

Enter number of anemometers to use from single tower :

Respond with the number of anemometers to use in the model for the nth output field.

Enter their sequence numbers (see table above) in ascending order :

Respond with the sequence numbers (from the displayed table) of input channels of each anemometer to be used in the model for the nth output field, listed in ascending order and separated by commas. These must be integers.

Enter their respective heights in meters :

Respond with the height of each anemometer to be used in the model for the nth output field, listed in ascending order and separated by commas.

Do you want to subtract the mean from each measurement (y or n) ?

Respond whether to subtract the mean wind speed at the appropriate height from each modeled, rotationally sampled wind speed for the nth output field.

Enter number of data points for low-pass filter
(use odd number, 1 = no filter) :

Respond with the number of data to use in filtering (enter '1' if no filtering is desired) for the nth output field. This must be an odd positive integer.

Enter description of output field :

Respond with the description of the nth output field. This must be a string of up to 30 characters.

Enter the start time and finish time in seconds into the file :

Respond with the start and finish times into the input file for which the wind data will be rotationally sampled, separated by a comma. If correlation with an actual wind turbine blade is desired, use a start time corresponding to a time when the blade was in the "12-o'clock" position.

5.2.1.2 Output

The output file created by the program is an ASCII, formatted file. The first record contains month, day, hour, minute, second (the starting time of the input file), data acquisition system identifier, number of output fields, and digitization interval in seconds and is written using FORMAT(1X,2I2,1X,3I2,1X,I1,1X,I3,1X,F9.4). The next number-of-output-fields records contain sequence number, identification, and units of each output field using FORMAT(1X,I3,A30,1X,A20). The next number-of-output-scans records contain elapsed time in seconds (after the starting time) and data for each field using FORMAT(1X,F9.2,50(F8.3,3X)).

5.2.2 Example Execution

Here is an example of how program STRS2PC appears on the terminal screen when executed. The user-inputs are underlined. The input file is named M3241129.ASC and contains 1.8 min of wind speed and direction measurements collected on 3/24/86 at 11:29:55 MST at a rate of 10 Hz from five levels along a meteorological tower immediately upwind of the WTS-4 wind turbine at Medicine Bow, Wyoming. The output file is named M3241130.WTS and contains 102 s of STRS wind speed at the radius of the rotor, STRS wind speed at 0.7 R, STRS wind speed with the mean shear removed, and hub-height wind speed for the turbine starting at 11:30:00 (5 s after the starting time of the input file). The sample input and output files for this example are included on the 5.25-in. floppy disk. Table 2 and Table 3 contain a small sample of the values found in the input and output file, respectively.

```
A> STRS2PC

   Enter the input file name :
      M3241129.ASC

   The following file was specified :

             File Name =>  M3241129.ASC
         Date Recorded =>   3/24
         Time Recorded =>  11:29:55
    Number of Channels =>   10
    Time between Scans =>  0.1000
```

Is this the desired file (y or n) ?

Y

Enter the output file name :

M3241130.WTS

Enter the number of output fields desired :

4

What type of data do you want in output field 1 ?

1) Single Tower Rotational Sampling
2) Circular array rotational sampling
3) Eulerian wind speed

Select one of the above :

1

Enter hub height of wind turbine in meters :

80.0

Enter rotor radius of wind turbine in meters :

39.0

Enter radial location of analysis
(i.e. 1.0 = 1.0*radius, 0.5 = 0.5*radius, etc.) :

1.0

Enter rotation rate of wind turbine in rpm :

30.0

Enter number of positions on circle to rotationally sample :

20

Sequence Number	Channel Number	Channel Description	Units
1	2	PNL 41m Wind Speed	m/s
2	3	PNL 41m Wind Direction	deg
3	4	PNL 60m Wind Speed	m/s
4	5	PNL 60m Wind Direction	deg
5	6	PNL 80m Wind Speed	m/s
6	7	PNL 80m Wind Direction	deg
7	8	PNL 100m Wind Speed	m/s
8	9	PNL 100m Wind Direction	deg
9	12	PNL 119m Wind Speed	m/s
10	13	PNL 119m Wind Direction	deg

Enter number of anemometers to use from single tower :

5

Enter their sequence numbers (see table above) in ascending order :

1, 3, 5, 7, 9

Enter their respective heights in meters :

41.0, 60.5, 80.0, 99.5, 119.0

Do you want to subtract the mean from each measurement (y or n) ?

N

Enter number of data points for low-pass filter
(use odd number, 1 = no filter) :

1

Enter description of output field :

STRS Wind Speed at 1.0R

What type of data do you want in output field 2 ?

1) Single Tower Rotational Sampling
2) Circular array rotational sampling
3) Eulerian wind speed

Select one of the above :

1

Enter radial location of analysis
(i.e. 1.0 = 1.0*radius, 0.5 = 0.5*radius, etc.) :

0.7

Do you want to use the same anemometers as output field 1 (y or n) ?

Y

Do you want to subtract the mean from each measurement (y or n) ?

N

Enter number of data points for low-pass filter
(use odd number, 1 = no filter) :

1

Enter description of output field :

STRS Wind Speed at 0.7R

What type of data do you want in output field 3 ?

1) Single Tower Rotational Sampling
2) Circular array rotational sampling
3) Eulerian wind speed

Select one of the above :

1

Enter radial location of analysis
(i.e. 1.0 = 1.0*radius, 0.5 = 0.5*radius, etc.) :

1.0

Do you want to use the same anemometers as output field 2 (y or n) ?

Y

Do you want to subtract the mean from each measurement (y or n) ?

Y

Enter number of data points for low-pass filter
(use odd number, 1 = no filter) :

1

Enter description of output field :

STRS Turbulence at 1.0R

What type of data do you want in output field 4 ?

1) Single Tower Rotational Sampling
2) Circular array rotational sampling
3) Eulerian wind speed

Select one of the above :

3

Do you want to use the same anemometers as output field 3 (y or n) ?

N

Sequence Number	Channel Number	Channel Description	Units
1	2	PNL 41m Wind Speed	m/s
2	3	PNL 41m Wind Direction	deg
3	4	PNL 60m Wind Speed	m/s
4	5	PNL 60m Wind Direction	deg
5	6	PNL 80m Wind Speed	m/s
6	7	PNL 80m Wind Direction	deg
7	8	PNL 100m Wind Speed	m/s
8	9	PNL 100m Wind Direction	deg
9	12	PNL 119m Wind Speed	m/s
10	13	PNL 119m Wind Direction	deg

Enter number of anemometers to use :

1

Enter their sequence numbers (see table above) :

5

Do you want to subtract the mean from each measurement (y or n) ?

N

Enter number of data points for low-pass filter
(use odd number, 1 = no filter) :

1

Enter description of output field :

Hub Height Wind Speed

Enter the start time and finish time in seconds into the file :

5.3, 107.7

WRITING DATA TO OUTPUT FILE MB03241130.WTS

Output time = 5.30 seconds
Output time = 5.40 seconds
Output time = 5.50 seconds

.
.
.

Output time = 107.50 seconds
Output time = 107.60 seconds
Output time = 107.70 seconds

TABLE 2. The First 3 s of Example Input File for STRS2PC. The file contains the elapsed time in seconds in the first column and then wind speed in m/s and direction in degrees for five different levels.

0.10	18.7310	258.0000	18.8365	249.5000	18.3201	249.0000	20.1573	253.0000	20.2366	251.0000
0.20	18.5323	256.0000	18.6379	246.5000	18.7172	251.0000	20.3559	254.0000	20.4352	251.0000
0.30	18.4330	257.0000	18.7372	247.5000	19.2137	252.0000	20.3559	255.0000	20.4352	251.0000
0.40	18.7310	258.0000	18.7372	249.5000	19.2137	251.0000	19.9587	255.0000	20.6337	251.0000
0.50	18.8303	255.0000	19.2337	249.5000	19.1144	253.0000	19.7601	251.0000	20.4352	251.0000
0.60	18.5323	257.0000	19.3330	248.5000	18.8165	251.0000	19.9587	253.0000	20.5345	254.0000
0.70	18.1351	258.0000	19.6308	248.5000	18.9158	251.0000	19.9587	254.0000	20.9316	252.0000
0.80	17.8372	256.0000	19.5315	248.5000	18.8165	254.0000	19.9587	253.0000	20.5345	247.0000
0.90	17.5392	256.0000	19.2337	247.5000	18.9158	254.0000	19.8594	253.0000	20.3359	248.0000
1.00	17.2413	254.0000	19.0351	246.5000	19.2137	252.0000	19.8594	253.0000	20.9316	252.0000
1.10	16.7448	255.0000	19.1344	247.5000	19.4123	249.0000	19.3630	254.0000	20.9316	253.0000
1.20	16.6455	257.0000	19.0351	247.5000	19.5115	248.0000	18.9658	254.0000	21.1302	253.0000
1.30	16.6455	258.0000	19.0351	248.5000	19.8094	251.0000	18.6679	252.0000	21.3288	253.0000
1.40	16.4468	262.0000	18.9358	251.5000	19.9087	250.0000	18.3701	252.0000	21.5273	255.0000
1.50	15.8510	261.0000	18.6379	250.5000	19.7101	249.0000	17.9729	253.0000	21.3288	255.0000
1.60	15.9503	259.0000	18.5386	251.5000	19.8094	248.0000	17.7743	254.0000	21.0309	256.0000
1.70	16.3475	259.0000	18.6379	252.5000	19.4123	247.0000	17.2779	254.0000	21.2295	256.0000
1.80	16.6455	257.0000	18.2408	250.5000	19.0151	246.0000	17.0793	254.0000	21.0309	255.0000
1.90	16.8441	258.0000	17.9429	251.5000	18.9158	247.0000	17.0793	255.0000	21.0309	256.0000
2.00	16.9434	258.0000	17.7443	253.5000	18.8165	249.0000	17.4765	257.0000	21.0309	256.0000
2.10	17.1420	258.0000	17.7443	253.5000	18.3201	249.0000	17.6750	257.0000	20.9316	256.0000
2.20	17.4399	258.0000	17.4465	254.5000	17.9229	248.0000	17.6750	256.0000	20.5345	255.0000
2.30	17.8372	259.0000	17.3472	253.5000	17.7243	251.0000	17.7743	256.0000	20.0380	257.0000
2.40	18.0358	261.0000	17.3472	252.5000	18.2208	251.0000	17.7743	257.0000	20.1373	257.0000
2.50	18.3337	262.0000	17.4465	252.5000	18.3201	251.0000	17.8736	255.0000	19.8394	256.0000
2.60	18.7310	262.0000	17.4465	253.5000	18.4194	249.0000	17.8736	252.0000	19.9387	255.0000
2.70	19.1282	260.0000	17.3472	253.5000	18.5186	249.0000	17.4765	254.0000	19.9387	255.0000
2.80	19.1282	259.0000	16.9500	252.5000	18.6179	250.0000	17.4765	257.0000	19.8394	255.0000
2.90	19.0289	260.0000	16.9500	250.5000	18.6179	251.0000	17.6750	254.0000	18.9458	254.0000
3.00	18.4330	257.0000	16.6521	247.5000	18.6179	253.0000	17.5758	252.0000	18.3501	253.0000

TABLE 3. The First 3 s of Example Output File for STRS2PC. The file contains the elapsed time in seconds, STRS wind speed at 1.0 R, STRS wind speed at 0.7 R, STRS turbulence at 1.0 R, and hub-height wind speed in m/s, respectively.

5.30	18.350	17.707	-.689	16.533
5.40	17.864	16.700	-1.135	16.136
5.50	16.938	17.009	-1.945	15.738
5.60	16.572	16.763	-2.130	15.738
5.70	17.149	16.970	-1.596	15.838
5.80	17.579	16.765	-1.351	16.434
5.90	17.450	16.955	-1.160	17.526
6.00	15.607	16.136	-2.752	17.923
6.10	15.369	17.152	-2.853	18.121
6.20	16.465	18.578	-1.668	18.419
6.30	15.354	17.742	-2.749	18.122
6.40	17.603	17.132	-.531	18.519
6.50	17.346	16.086	-.876	18.022
6.60	17.608	16.346	-.751	17.824
6.70	19.310	17.872	.699	17.824
6.80	19.732	18.618	.802	17.526
6.90	18.555	17.952	-.190	17.327
7.00	18.219	17.117	-.483	17.228
7.10	18.126	17.881	-.757	17.228
7.20	18.637	17.182	-.362	17.526
7.30	18.151	17.329	-.887	17.427
7.40	18.736	16.431	-.263	17.327
7.50	18.143	16.656	-.740	17.327
7.60	16.026	16.511	-2.677	17.327
7.70	17.922	16.952	-.823	17.426
7.80	19.430	16.379	.499	17.327
7.90	16.321	17.267	-2.290	17.228
8.00	15.200	17.499	-3.159	17.029
8.10	17.217	15.656	-1.005	17.426
8.20	15.145	15.826	-2.988	17.724
8.30	16.050	16.769	-2.053	17.824

Table 4 presents wind speed time series data computed by the PC version of STRS-2 compared to the corresponding time series computed by the mainframe computer. It shows that the STRS2PC model gives exactly the same numbers as the mainframe version to 4 digits of precision. This is far better than the accuracy of measurement or modeling of the wind.

5.2.3 Graphical Example of Results

Graphs of time series of the STRS-2-modeled rotational wind speed and the corresponding rotational spectra for the above example are shown in Figure 9. The spectral and the plotting routines are not part of the STRS2PC model. The graphs are included only to provide another basis for the user to check the success in execution of the model.

5.2.4 Execution Time

Using data files on floppy, or better, hard disk permits the longest input time series. The example given in this report used only 102 s of data sampled at an interval of 0.1 s. Twenty points around the ring were simulated using five levels of anemometers on the tower. The time required by the IBM-PC, reading data from a floppy disk, to compute the STRS-2 rotational time series was 95 times the modeled time. Four time series were generated in this example: (1) the STRS-2 wind speed at R, (2) the STRS-2 wind speed at 0.7 R, (3) the STRS-2 turbulent wind speed, and (4) the Eulerian wind speed at hub height. Approximately 30% of the time was used for each of time series (1), (2), and (3), so a reduction of computation time is substantial if computation of one or two of these three series is not performed. The speed of simulation of the same time series also can be substantially increased if the input measured wind speeds can be read from "RAM disk" rather than from floppy or hard disk.

TABLE 4. Comparison of Wind Speed Computation Precision of Desktop and Mainframe Versions of STRS-2 Model

Elapsed Time from Beginning of File, s	Wind Speed, m/s	
	Mainframe	Desktop
5.30	18.350	18.350
5.40	17.864	17.864
5.50	16.938	16.938
5.60	16.572	16.572
5.70	17.149	17.149
5.80	17.579	17.579
5.90	17.450	17.450
6.00	15.607	15.607
6.10	15.369	15.369
6.20	16.465	16.465
6.30	15.354	15.354
6.40	17.602	17.603
6.50	17.346	17.346
6.60	17.608	17.608
6.70	19.310	19.310
6.80	19.733	19.732
6.90	18.555	18.555
7.00	18.219	18.219
7.10	18.126	18.126
7.20	18.637	18.637
7.30	18.151	18.151
7.40	18.737	18.736
7.50	18.144	18.143
7.60	16.026	16.026
7.70	17.922	17.922
7.80	19.430	19.430
7.90	16.321	16.321
8.00	15.200	15.200
8.10	17.216	17.217
8.20	15.145	15.145
8.30	16.049	16.050

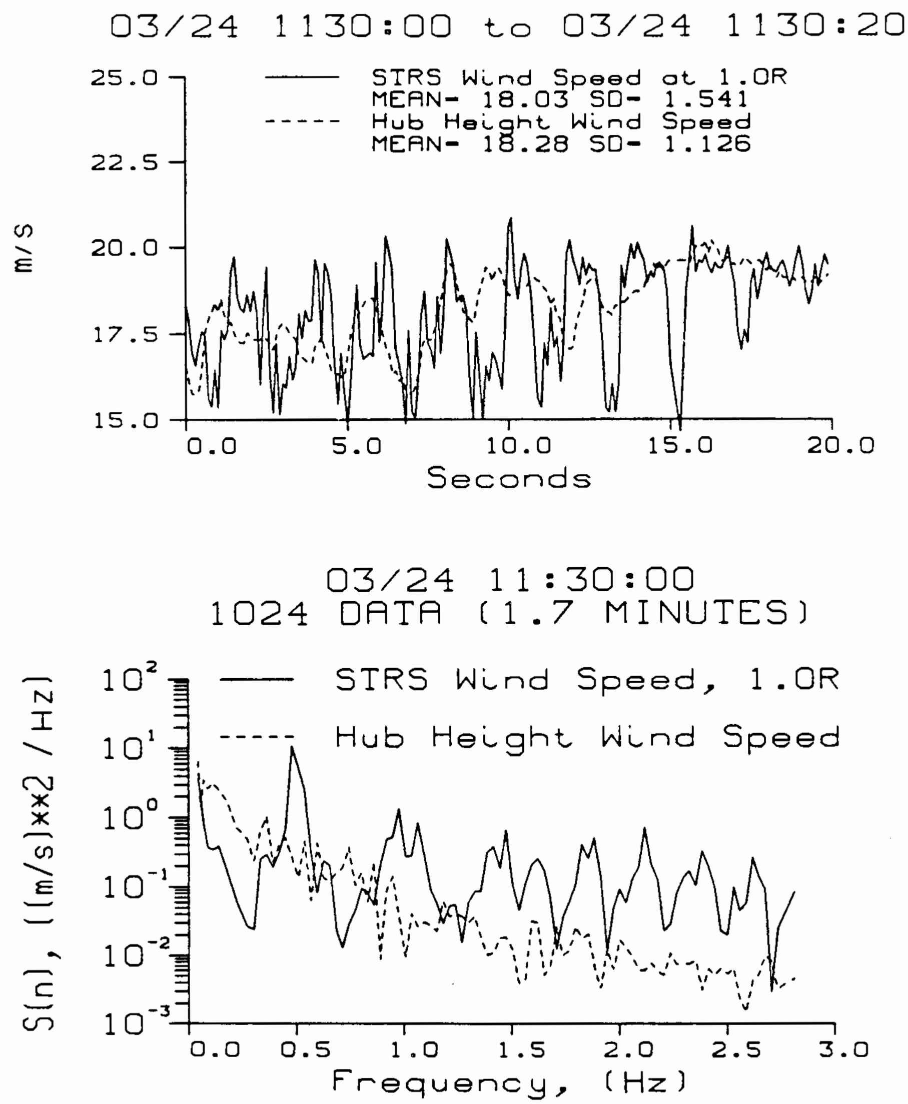

FIGURE 9. Test Case Output Time Series and Corresponding Spectra

6.0 A ROTOR BLADE RESPONSE FUNCTION BASED ON STRS-2 WIND FLUCTUATIONS

One brief, practical example of the use of the STRS-2 rotational wind is given here. The example is a result of research, conducted with other DOE laboratories and their contractors, at Medicine Bow, Wyoming. PNL provided wind measurements from a single turbulence-measurement tower and subsequent analysis of the character of the wind passing through the rotor disk of the WTS-4 wind turbine (Connell, Morris, and Hinchee 1987).

The referenced paper addresses the "Comparison of Turbine Response Characteristics with Turbulence Characteristics." A spectral response function of the root blade bending moment is computed, for three wind cases, by dividing the spectral density function of the blade bending moment by the spectral density function of the STRS-2 wind speed.

An example of those two spectra for one of the cases taken from the paper is shown in Figure 10. No correction of the STRS-2 spectrum was made to account for the lack of wind measurements made in positions lateral from the single tower. The response functions for the example in Figure 10 and for two other examples are shown in Figure 11. The discussion of the result is left to the original paper, which concluded that there was a substantial improvement in the response functions compared to others that have been computed by previous single-tower methods. The paper also noted that the STRS-2 model, though a great improvement over previous models that have only single-tower data to work with, has serious limitations for some intended uses.

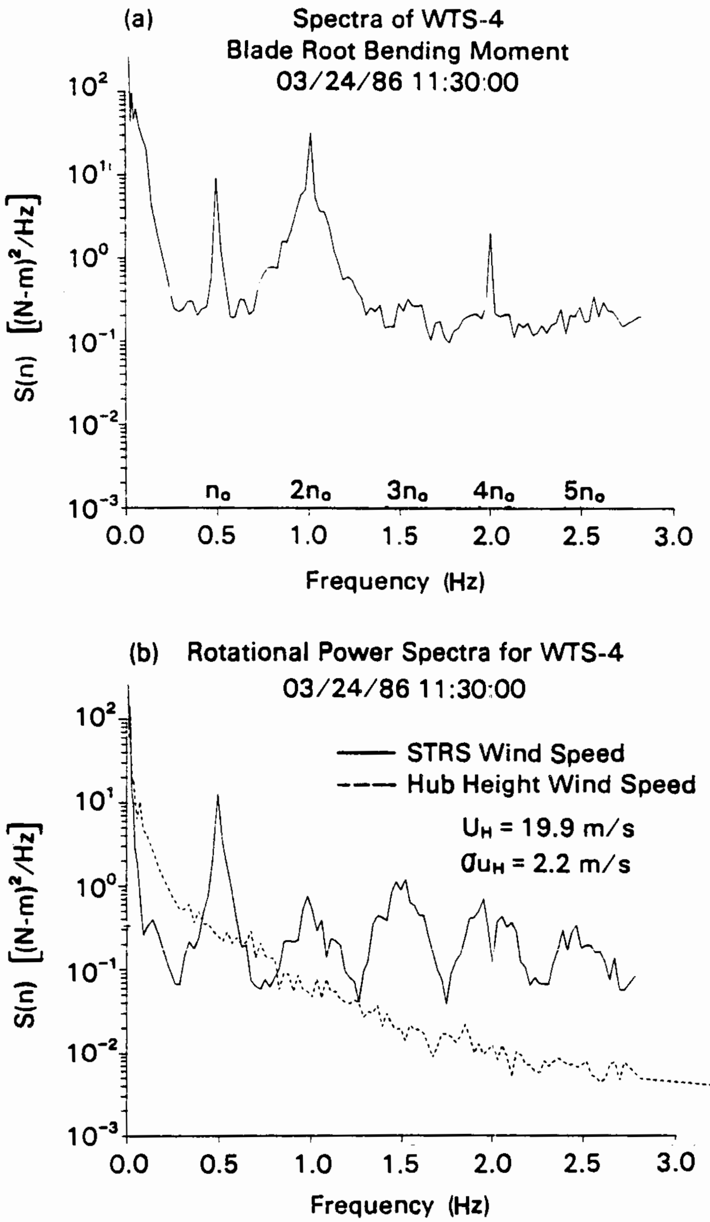

FIGURE 10. An Example of (a) the WTS-4 Root Blade Bending Moment Spectrum and (b) the Corresponding STRS-2 Wind Spectrum Used to Compute a Turbine Response Function (Connell, Morris, and Hinchee 1987)

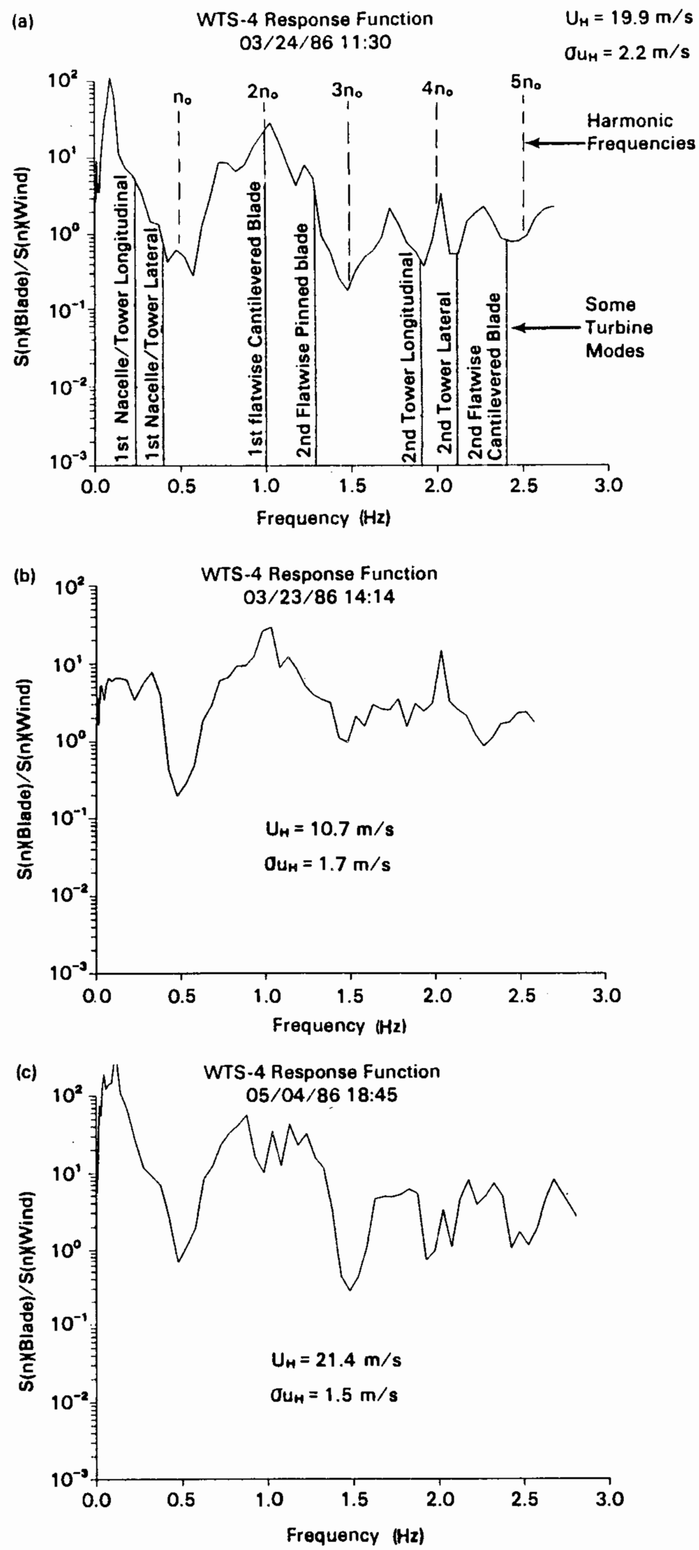

FIGURE 11. Three Examples of the WTS-4 Blade Root Bending Moment Response Function Computed Using the STRS-2 Rotationally Sampled Wind (Connell, Morris, and Hinchee 1987)

7.0 CONCLUDING REMARKS AND CAUTIONARY GUIDANCE ON THE USE OF STRS-2

The main purposes of this report have been twofold. The first was to document a desktop computer program, STRS2PC, for easy, practical modeling of the rotationally sampled wind using measurements of wind speed made from several levels on a single meteorological tower. The second was to bring together and update in one document the information about the STRS-2 wind model scattered throughout a number of references.

The positive aspect of the STRS-2 wind model is that the resultant rotationally sampled wind time series is substantially more complete and accurate than any other current model that is based upon measurements from a single tower or less. The use of wind measurements provides information to represent a significant part of the effect of specific terrain and atmospheric flow conditions, which cannot yet be modeled theoretically. The negative aspect of the STRS-2 model is that it does not utilize measured wind characteristics in the crosswind plane away from the single tower. Its simulation of the properties of turbulent flow in the crosswind region is useful, but incomplete. Often no crosswind measurements can be obtained anyway; thus the need for STRS-2. The use of correction factors that we have computed and presented in this report should be made with caution. In any case, correction factors are not included as part of the STRS-2 model. There is very little experience from which to know how generally applicable the correction factors are.

Some control of the exact nature of the STRS-2 model may be exercised by the user of the program. The wind at a series of radial locations along a blade may be simulated simultaneously from the same input measured wind velocities. The wind along another blade may be simulated by running the program again and advancing the entered start time by the time it takes the other blade to reach the same starting position as the previous blade. Also, the number and location of anemometers on the single tower to be used for measurements can be specified. When installing the anemometers along the tower, spacing can be made such that each one is at the height of a horizontal projection of one of the simulated anemometers on the circle. The expectation might be that this ideal location of tower anemometers would eliminate spatial interpolation, and sampling frequently enough would eliminate time interpolation. It will certainly reduce the errors caused by interpolation. However, some interpolation would still be done by the program because the wind speed and shape of the simulated along-wind ellipse determine the lead and lag times. The program can simulate winds around a ring of many anemometers from only two anemometers on the tower, but three or more will produce much better results.

If the sample rate of measurement of the wind from the tower is large, then little interpolation in time will be required to match the time at which the simulated point rotates through the positions of the simulated ring of anemometers. However, the program advects the wind from the tower to the simulated ring locations with a number of steps that is determined by the measurement sampling rate. Thus, the advection computation time will increase when the interpolation of time becomes less extensive on the basis of smaller time steps in the input data.

If applied within these guidelines and if the suggestion to not over-interpret the STRS-2 time series that is simulated is adhered to, the STRS2PC model should provide a simple, effective method of improving the estimation of the wind experienced by points on a rotating blade. Again a caution about overinterpretation is justified. Little research and few application experiences exist to generate confidence that a set of multi-point rotational time series produced by STRS-2 accurately incorporates the correlation, between radial points, of the actual wind experienced by a rotating blade. In the absence of further basic research on correlations along a rotating radial line (e.g., a rotor blade axis), no estimates of error will be offered by the authors.

8.0 REFERENCES

Connell, J. R. 1980. Turbulence Spectrum Observed by a Fast-Rotating Wind Turbine Blade. PNL-3426, Pacific Northwest Laboratory, Richland, Washington.

Connell, J. R. 1982. "The Spectrum of Wind Speed Fluctuations Encountered by a Rotating Blade of a Wind Energy Conversion System." Solar Energy 29(5):363-375.

Connell, J. R. 1984. "Basic Principles and Recent Observations of Rotationally Samples Wind." Paper presented at the NASA Workshop on Large Wind Turbines, May 8-10, 1984, Cleveland, Ohio.

Connell, J. R. 1985. "A Primer of Turbulence at the Wind Turbine Rotor." In Proceedings of the AWEA/DOE Wind Energy Workshop (Windpower '85), pp. 57-66. American Wind Energy Association, Alexandria Virginia; also SERI/CP-217-2902, Solar Energy Research Institute, Golden, Colorado.

Connell, J. R., and R. L. George. 1983a. "A New Look at Turbulence Experienced by a Rotating Wind Turbine." In Proceedings of the Second ASME Wind Energy Symposium, pp. 455-480. American Society of Mechanical Engineers, New York, New York.

Connell, J. R., and R. L. George. 1983b. "Scaling Wind Characteristics for Designing Small and Large Wind Turbines." In Proceedings of the American Solar Energy Society Annual Meeting (Wind Workshop VI), pp. 513-524, American Solar Energy Society, Newark, Delaware.

Connell, J. R., and R. L. George. 1987. Using a New Characterization of Turbulent Wind for Accurate Correlation of Wind Turbine Response with Wind Speed. PNL-6283, Pacific Northwest Laboratory, Richland, Washington.

Connell, J. R., and R. L. George. 1988. "Accurate Correlation of Wind Turbine Response with Wind Speed." Solar Energy 109:321-329.

Connell, J. R., R. L. George, V. R. Morris, and V. A. Sandborn. 1985. Rotationally Sampled Wind and MOD-2 Wind Turbine Response. EPRI AP-4335, Electric Power Research Institute, Palo Alto, California.

Connell, J. R., V. R. Morris, and M. E. Hinchee. 1987. "Input Turbulence Features at a Megawatt-Size Wind Turbine-Medicine Bow, Wyoming." In Proceedings of the Sixth ASME Wind Energy Symposium, pp. 201-213, American Society of Mechanical Engineers, New York, New York.

Counihan, J. 1975. "Adiabatic Atmospheric Boundary Layers." Atmospheric Environment 9:871-905.

George, R. L., and J. R. Connell. 1983. Rotationally Sampled Wind Characteristics and Correlations with MOD-OA Wind Turbine Response. PNL-5238, Pacific Northwest Laboratory, Richland, Washington.

Panofsky, H. A., and J. A. Dutton. 1984. Atmospheric Turbulence. John Wiley, New York, New York.

Powell, D. C., and J. R. Connell. 1986a. A Model for Simulating Rotational Data for Wind Turbine Applications. PNL-5857, Pacific Northwest Laboratory, Richland, Washington.

Powell, D. C., and J. R. Connell. 1986b. Review of Wind Simulation Methods for Horizontal-Axis Wind Turbine Analysis. PNL-5903, Pacific Northwest Laboratory, Richland, Washington.

Powell, D. C., and J. R. Connell. 1987. "Verification of Theoretically Computed Spectra for a Point Rotating in a Vertical Plane." Solar Energy 39(1)53-63.

APPENDIX A

THE STRS2PC SOURCE CODE

APPENDIX A

THE STRS2PC SOURCE CODE

```
$LARGE
      PROGRAM RSWIND

C
C PROGRAM DESCRIPTION:
C
C        This program models the turbulent wind encountered by a point on a
C     rotating wind turbine blade by transforming a time series of the wind
C     speed measured by anemometers on either a single tower or a
C     multi-tower array from eulerian to rotational coordinates.
C
C AUTHOR:
C
C     Victor R. Morris
C
C CREATION DATE:
C
C     July 1987
C
C
C         C H A N G E   L O G
C
C       Date       | Name  | Description
C-----------------+-------+------------------------------------------------
C   12-AUG-1987   |  VRM  | Included SUBROUTINE zfact and FUNCTION euler,
C                 |       | added nanemo to COMMON output
C-----------------+-------+------------------------------------------------
C   17-AUG-1987   |  VRM  | Corrected arithmetic error in FUNCTION strs, line 74
C-----------------+-------+------------------------------------------------
C   19-AUG-1987   |  VRM  | Included SUBROUTINE rtfact and FUNCTION vpa
C-----------------+-------+------------------------------------------------
C   19-AUG-1987   |  VRM  | Modified FUNCTION ws, eliminating need of
C                 |       | SUBROUTINE zfact, added r to COMMON output
C-----------------+-------+------------------------------------------------
C   28-AUG-1987   |  VRM  | Deleted all references to common variable nblade
C-----------------+-------+------------------------------------------------
C   25-SEP-1987   |  VRM  | Included FUNCTION INDEX2
C-----------------+-------+------------------------------------------------
C   28-SEP-1987   |  VRM  | Included SUBROUTINE average
C-----------------+-------+------------------------------------------------
C   29-SEP-1987   |  VRM  | Added avwind to COMMON output, included avwind
C                 |       | computation to SUBROUTINE yzfact and SUBROUTINE
C                 |       | rtfact, and included SUBROUTINE zfact
C-----------------+-------+------------------------------------------------
C   25-NOV-1987   |  GLG  | Added the OPEN statement for the new header file.
C-----------------+-------+------------------------------------------------
C   09-DEC-1987   |  VRM  | Corrected error in end-of-file checking routine in
```

```
C                |       | SUBROUTINE AVERAG and FUNCTION WS, from an IOSTAT
C                |       | value of -1 to an IOSTAT of 36 (FOR$_ATTACCNON)
C----------------+-------+---------------------------------------------------
C   10-DEC-1987  |  GLG  | Modified the STRS subroutine to correctly handle
C                |       | the EOF or Non-existant record condition.
C----------------+-------+---------------------------------------------------
C   10-DEC-1987  |  GLG  | Modified the record number calculation to correct
C                |       | for an error in Microsoft FORTRAN Compiler.
C                |       | (Note: and error occurs when the formula
C                |       |    I = INT(A/B)    is used, replace with the lines
C                |       |
C                |       |    C = A/B
C                |       |    I = INT(C)      this will correct the error.)
C----------------+-------+---------------------------------------------------
C                |       |
C----------------+-------+---------------------------------------------------
C                |       |
C----------------+-------+---------------------------------------------------
C [change_entry]
C
      INTEGER   pos, npos
      REAL      dtout, tout, sTime, fTime
      CHARACTER filin*48, filout*48, filhdr*48

C
C     < Initialize COMMON blocks
C
      INTEGER   maxin, maxout, maxpos
      PARAMETER (maxin=100, maxout=50, maxpos=100)
      INTEGER*2 chin(maxin)
      INTEGER   month, day, hour, minute, second, system
      INTEGER   nchin, nchout
      INTEGER   dattyp(maxout), nanemo(maxout)
      INTEGER   sqnc(maxin,maxout), nfilt(maxout)
      INTEGER   below(maxpos,maxout), above(maxpos,maxout)
      REAL      y(maxpos,maxout), weight(maxpos,maxout)
      REAL      dtin, r(maxout), avwind(maxpos,maxout)
      LOGICAL   done, shadow(maxout), mean(maxout)
      CHARACTER desin(maxin)*30, units(maxin)*20

      COMMON /header/ month, day, hour, minute, second, system
      COMMON /chinfo/ nchin, chin, dtin, desin, units
      COMMON /output/ nchout, dattyp, nanemo, nfilt,
     1                sqnc, r, avwind, shadow, mean
      COMMON /interp/ above, below, weight, y
      COMMON /finish/ done

C
C     < Prompt user for input file name
C
      CALL INFIL (filin)
C
```

```
C      < Open input file
C
       OPEN (
      1  UNIT=1,
      1  FILE = filin,
      1  STATUS = 'OLD',
      1  ACCESS = 'DIRECT',
      1  RECL = (nchin+1)*11,
      1  FORM = 'FORMATTED')
C
C      < Prompt user for output file name
C
       CALL OUTFIL (filout)
       filhdr = filout(1:index(filout,'.'))//'hdr'
C
C      < Open output file and header file
C
       OPEN (
      1  UNIT=2,
      1  FILE = filout,
      1  STATUS = 'NEW',
      1  FORM = 'FORMATTED')

       OPEN (
      1  UNIT=4,
      1  FILE = filhdr,
      1  STATUS = 'NEW',
      1  FORM = 'FORMATTED')
C
C      < Prompt user for output field information
C
       CALL PROMPT (npos, dtout, sTime, fTime)
C
C      < Initialize output time and rotational position
C
       tout = sTime
       pos = 1
       done = .FALSE.

       WRITE (*,*)
       WRITE (*,*) 'WRITING DATA TO OUTPUT FILE ', filout
       WRITE (*,*)

100    CONTINUE
C
C      < Calculate output data and write output record
C
         CALL WSDATA (pos, tout)
C
C      < Increment output time and rotational position
C
         tout = tout + dtout
```

```
          pos = pos + 1
          IF (tout .GT. fTime) done = .TRUE.
          IF (pos .GT. npos) pos = 1
        IF (.NOT. done) GO TO 100
C
C       < Close input and output file
C
        CLOSE (UNIT=1)
        CLOSE (UNIT=2)
        STOP
        END
```

```
$LARGE
      SUBROUTINE anemo(chnl,height,azimth,same)

C
C FUNCTIONAL DESCRIPTION:
C
C       This subroutine displays the descriptions of each channel of the
C     input file and prompts the user for the anemometers to use for
C     the current output field data type.
C
C DUMMY ARGUMENTS:
C    INPUTS:
C    chnl                     sequence number of output data channel
C    same                     indicates whether to use same as last anemometers
C    OUTPUTS:
C    height(1..nanem,chnl)    height of each anemometer of output data channel
C    azimth(1..nanem+1,chnl)  azimuth of each anemometer of output data channel
C
C COMMON INPUTS:
C
C    nchin                    number of channels of data in input file
C    chin(1..nchin)           DAS channel number of each input data channel
C    desin(1..nchin)          description of each input data channel
C    units(1..nchin)          units of measurement of each input data channel
C    dattyp(1..nchout)        data type of each output data channel
C
C COMMON OUTPUTS:
C
C    nanemo(1..nchout)        number of anemometers of each output data channel
C    sqnc(1..nanemo+1,1..nchout) input data channel sequence number
C
C SIDE EFFECTS:
C
C    none
C
C

      INTEGER   maxin, maxout, maxpos
      PARAMETER (maxin=100, maxout=50, maxpos=100)
      INTEGER*2 chin(maxin)
      INTEGER   ch, chnl, anem, nchin
      INTEGER   dattyp(maxout), nanemo(maxout), sqnc(maxin,maxout)
      INTEGER   dummy2, dummy3(maxout)
      REAL      height(maxin,maxout), azimth(maxin,maxout)
      REAL      dummy1, dummy4(maxout), dummy5(maxpos,maxout)
      LOGICAL   same, dummy6(maxout), dummy7(maxout)
      CHARACTER desin(maxin)*30, units(maxin)*20

      COMMON /chinfo/ nchin, chin, dummy1, desin, units
      COMMON /output/ dummy2, dattyp, nanemo, dummy3,
     1                sqnc, dummy4, dummy5, dummy6, dummy7
```

```
10    FORMAT(//)
20    FORMAT(1X,'Sequence',2X,'Channel',6X,'Channel',25X,'Units')
30    FORMAT(1X,'Number',4X'Number',7X,'Description')
40    FORMAT(4X,I2,7X,I2,9X,A30,2X,A20)
50    FORMAT(/,' Enter number of anemometers to use ', $)
51    FORMAT('from single tower : ', $)
52    FORMAT('from circular array : ', $)
53    FORMAT(': ', $)
60    FORMAT(' Enter their sequence numbers ', $)
61    FORMAT('(see table above) in accending order : ')
62    FORMAT('(see table above) in order of blade rotation : ')
63    FORMAT('(see table above) : ')
90    FORMAT(' Enter their respective heights in meters: ')
100   FORMAT(' Enter sequence number of hub height anemometer : ', $)

C
C     < Determine anemometers to use for each output channel
C
      IF (.NOT.same) THEN
        WRITE(*,10)
        WRITE(*,20)
        WRITE(*,30)
        DO 200, ch = 1, nchin
          WRITE(*,40) ch, chin(ch), desin(ch), units(ch)
200     CONTINUE
        WRITE(*,50)
        IF (dattyp(chnl).EQ.1) WRITE(*,51)
        IF (dattyp(chnl).EQ.2) WRITE(*,52)
        IF (dattyp(chnl).EQ.3) WRITE(*,53)
        READ(*,*) nanemo(chnl)
        WRITE(*,60)
        IF (dattyp(chnl).EQ.1) WRITE(*,61)
        IF (dattyp(chnl).EQ.2) WRITE(*,62)
        IF (dattyp(chnl).EQ.3) WRITE(*,63)
        READ(*,*) (sqnc(anem,chnl), anem=1,nanemo(chnl))
        IF (dattyp(chnl).EQ.1) THEN
          WRITE(*,90)
          READ(*,*) (height(anem,chnl), anem=1,nanemo(chnl))
        END IF
        IF (dattyp(chnl).EQ.2) THEN
          WRITE(*,100)
          READ(*,*) sqnc(nanemo(chnl)+1,chnl)
          DO 300, anem = 1,nanemo(chnl)+1
            azimth(anem,chnl) = FLOAT(anem-1)
     1                        / FLOAT(nanemo(chnl)) * 360.
300       CONTINUE
        END IF
      ELSE
        nanemo(chnl) = nanemo(chnl-1)
        DO 400, anem=1,nanemo(chnl)
          sqnc(anem,chnl) = sqnc(anem,chnl-1)
400     CONTINUE
```

```
        IF (dattyp(chnl).EQ.1) THEN
          IF (height(nanemo(chnl),chnl-1).EQ.0.0) THEN
            WRITE(*,90)
            READ(*,*) (height(anem,chnl), anem=1,nanemo(chnl))
          ELSE
            DO 500, anem=1,nanemo(chnl)
              height(anem,chnl) = height(anem,chnl-1)
500         CONTINUE
          END IF
        END IF
        IF (dattyp(chnl).EQ.2) THEN
          IF (azimth(nanemo(chnl),chnl-1) .EQ. 0.0) THEN
            WRITE(*,100)
            READ(*,*) sqnc(nanemo(chnl)+1,chnl)
            DO 600, anem = 1,nanemo(chnl)+1
              azimth(anem,chnl) = FLOAT(anem-1)
     1                          / FLOAT(nanemo(chnl)) * 360.
600         CONTINUE
          ELSE
            sqnc(nanemo(chnl)+1,chnl) = sqnc(nanemo(chnl)+1,chnl-1)
            DO 700, anem=1,nanemo(chnl)+1
              azimth(anem,chnl) = azimth(anem,chnl-1)
700         CONTINUE
          END IF
        END IF
      END IF
      RETURN
      END
```

```
$LARGE
      SUBROUTINE averag(sTime,fTime,avdata)

C
C FUNCTIONAL DESCRIPTION:
C
C      This subroutine reads the required records of the input file and computes
C    the average of the data on each channel.
C
C DUMMY ARGUMENTS:
C     INPUTS:
C     sTime                       start time of output data
C     fTime                       finish time of output data
C     OUTPUT:
C     avdata(1..nchin)            average value of data for each input channel
C
C COMMON INPUTS:
C
C     nchin                     number of channels of data in input file
C     dtin                        time interval between samples of input data
C
C COMMON OUTPUTS:
C
C     none
C
C SIDE EFFECTS:
C
C     none
C
C


      INTEGER   maxin, maxout, maxpos
      PARAMETER (maxin=100, maxout=50, maxpos=100)
      INTEGER*2 dummy1(maxin)
      INTEGER   ch, i, nchin
      INTEGER   irec1, irec2, count, ios
      INTEGER   dummy4, dummy5(maxout), dummy6(maxout)
      INTEGER   dummy7(maxout), dummy8(maxin,maxout)
      REAL      sTime, fTime, eTime, dtin, rec1, rec2
      REAL      data(maxin), avdata(maxin)
      REAL      dummy9(maxout)
      LOGICAL   dumy10(maxout), dumy11(maxout)
      CHARACTER dummy2(maxin)*30, dummy3(maxin)*20

      COMMON /chinfo/ nchin, dummy1, dtin, dummy2, dummy3

10    FORMAT(1X,F9.2,100(F8.3,3X))

C
C     < Compute record number of start and finish of average
C
      rec1 = stime/dtin
```

```
      rec2 = ftime/dtin
      irec1 =  int(rec1)
      irec2 = nint(rec2)

C
C     < Initialize average data array
C
      DO 100, ch = 1, nchin
        avdata(ch) = 0.0
100   CONTINUE
C
C     < Compute sum of data for each input channel
C
      count = 0
      DO 300, i = irec1,irec2
        READ (UNIT=1, REC=i, IOSTAT=ios, FMT=10)
     1        eTime, (data(ch), ch=1,nchin)
        IF (ios.EQ.-1) GO TO 400
        DO 200, ch = 1, nchin
          avdata(ch) = avdata(ch) + data(ch)
200     CONTINUE
        count = count + 1
300   CONTINUE
C
C     < Compute average of data for each input channel
C
400   CONTINUE
      IF (count.GT.0) then
        DO 500, ch = 1, nchin
          avdata(ch) = avdata(ch)/FLOAT(count)
500     CONTINUE
      END IF
      RETURN
      END
```

```
$LARGE
      REAL FUNCTION euler(chnl,tout)

C
C FUNCTIONAL DESCRIPTION:
C
C        This function computes the average wind speed for the current
C      output time and current output field.
C
C DUMMY ARGUMENTS:
C
C     chnl                    sequence number of output data channel
C     tout                    output time
C
C COMMON INPUTS:
C
C     dtin                    time interval between samples of input data
C     nanemo(1..nchout)       number of anemometers of each output data channel
C
C COMMON OUTPUTS:
C
C     none
C
C FUNCTION VALUE:
C
C     euler                   eulerian wind speed
C
C SIDE EFFECTS:
C
C     none
C
C

      INTEGER   maxin, maxout,maxpos
      PARAMETER (maxin=100, maxout=50, maxpos=100)
      INTEGER*2 dummy2(maxin)
      INTEGER   anem, chnl, index1, index2, nanemo(maxout)
      INTEGER   dummy1, dummy5, dummy6(maxout)
      INTEGER   dummy7(maxout), dummy8(maxin,maxout)
      REAL      dtin, tout, twght, ws, ws1, ws2, rindex
      REAL      dummy9(maxout), dumy10(maxpos,maxout)
      LOGICAL   dumy11(maxout), dumy12(maxout)
      CHARACTER dummy3(maxin)*30, dummy4(maxin)*20

      COMMON /chinfo/ dummy1, dummy2, dtin, dummy3, dummy4
      COMMON /output/ dummy5, dummy6, nanemo, dummy7,
     1                dummy8, dummy9, dumy10, dumy11, dumy12

C
C     < Compute record numbers of input file
C     < surrounding current output time
C
```

```
      rindex = tout/dtin
      index1 = INT(rindex)
      index2 = index1 + 1
C
C     < Compute time interpolation weighting factor
C
      twght = ABS(tout - FLOAT(index1)*dtin) / dtin
C
C     < Initialize sum of wind speeds
C
      ws1 = 0.0
      ws2 = 0.0
C
C     < Get wind speeds from input file
C     < surrounding current output time
C     < for each anemometer in average and sum them
C
      DO 100, anem = 1,nanemo(chnl)
        ws1 = ws1 + ws(anem,chnl,index1)
        ws2 = ws2 + ws(anem,chnl,index2)
100   CONTINUE
C
C     < Compute average wind speeds
C     < surrounding current output time
C
      ws1 = ws1/nanemo(chnl)
      ws2 = ws2/nanemo(chnl)
C
C     < Compute wind speed at current output time
C
      euler = ws1 + (ws2-ws1)*twght
      END
```

```
$LARGE
      INTEGER FUNCTION INDEX2(c1,c2)

C
C FUNCTIONAL DESCRIPTION:
C
C       This function searches for a substring (c2) in a specified character
C     string (c1), and, if it finds the substring, returns the starting position
C     of the substring's last (rightmost) occurrence.  If c2 does not occur in
C     c1, the value zero is returned.  This function is similar to the FORTRAN
C     character function INDEX which returns the starting position of the
C     substring's first (leftmost) occurrence.
C
C DUMMY ARGUMENTS:
C
C    c1                         character string to be searched
C    c2                         character substring
C
C COMMON INPUTS:
C
C    none
C
C COMMON OUTPUTS:
C
C    none
C
C FUNCTION VALUE:
C
C    INDEX2                     starting position of last substring
C
C SIDE EFFECTS:
C
C    none
C
C
      INTEGER       len1, len2, i
      CHARACTER*(*) c1, c2

      len1 = LEN(c1)
      len2 = LEN(c2)
      INDEX2 = 0
      DO 100 i = len1-len2+1, 1, -1
        IF (c1(i:i+len2-1) .EQ. c2) THEN
          INDEX2 = i
          GO TO 200
        END IF
100   CONTINUE
200   END
```

```
$LARGE
      SUBROUTINE INFIL (filin)

C
C FUNCTIONAL DESCRIPTION:
C
C       This subroutine prompts the user for the input file name and
C     opens and reads the input header file.
C
C DUMMY ARGUMENTS:
C    OUTPUTS:
C    filin                        input file name
C
C COMMON INPUTS:
C
C    none
C
C COMMON OUTPUTS:
C
C    month, day                   date of input file
C    hour, minute, second         time of input file
C    system                       number of data acquisition system used
C    nchin                        number of channels of data in input file
C    dtin                         time interval between samples of input data
C    chin(1..nchin)               DAS channel number of each input data channel
C    desin(1..nchin)              description of each input data channel
C    units(1..nchin)              units of measurement of each input data channel
C
C SIDE EFFECTS:
C
C    none
C
C

      INTEGER   maxin
      PARAMETER (maxin=100)
      INTEGER*2 chin(maxin)
      INTEGER   month, day, hour, minute, second, system
      INTEGER   ch, nchin, INDEX2
      REAL      dtin
      CHARACTER filin*48, filhdr*48, answer*1
      CHARACTER desin(maxin)*30, units(maxin)*20

      COMMON /header/ month, day, hour, minute, second, system
      COMMON /chinfo/ nchin, chin, dtin, desin, units

10    FORMAT('1')
20    FORMAT(' Enter the  input file name : ',$)
30    FORMAT(A)
31    FORMAT(1X, 2I2, 1X, 3I2, 1X, I1, 1X, I3, 1X, F9.4)
40    FORMAT('1',' The following file was specified : ',//)
41    FORMAT('                         File Name =>  ',25A,/)
```

```
42     FORMAT(/,'           Date Recorded =>  ',I2,'/',I2,/)
43     FORMAT('             Time Recorded =>  ',I2,':',I2,':',I2,/)
44     FORMAT('        Number of Channels =>  ',I3,/)
45     FORMAT('        Time between Scans =>  ',F6.4,////)
50     FORMAT(' Is this the desired file (y or n) ? ',$)
60     FORMAT(1x, i3, 1x, A30, 1X, A20)
C
C      < Prompt user for input file name
C
       WRITE(*,10)
100    CONTINUE
         WRITE(*,20)
         READ(*,30) filin
C
C      < Open header file
C
         filhdr = filin(1:INDEX2(filin,'.')) // 'HDR'
         OPEN (
     1     UNIT=3,
     1     FILE = filhdr,
     1     STATUS = 'OLD',
     1     FORM = 'FORMATTED')
C
C      < Read file header
C
         READ (UNIT=3, FMT=31) month, day, hour, minute, second,
     1                         system, nchin, dtin
C
C      < Type header information
C
         WRITE(*,40)
         WRITE(*,41) filin
         WRITE(*,42) month, day
         WRITE(*,43) hour, minute, second
         WRITE(*,44) nchin
         WRITE(*,45) dtin
C
C      < Prompt user for correct file selection
C
         WRITE(*,50)
         READ(*,30) answer
       IF (answer.NE.'y' .AND. answer.NE.'Y') GO TO 100
C
C               < Read additional header information
C               < which includes actual channel number,
C               < description of channel, and units of measure.
C
       DO 200, ch = 1, nchin
         READ (UNIT=3, FMT=60) chin(ch), desin(ch), units(ch)
200    CONTINUE
       RETURN
       END
```

```
$LARGE
      SUBROUTINE OUTFIL (filout)

C
C FUNCTIONAL DESCRIPTION:
C
C       This subroutine prompts the user for the output file name.
C
C DUMMY ARGUMENTS:
C    OUTPUTS:
C    filout                    name of output file
C
C COMMON INPUTS:
C
C    none
C
C COMMON OUTPUTS:
C
C    none
C
C SIDE EFFECTS:
C
C    none
C
C

      CHARACTER filout*48

10    FORMAT(/,' Enter the output file name : ',$)
20    FORMAT(A)

C
C     < Prompt user for output file name
C
      WRITE(*,10)
      READ(*,20) filout
      RETURN
      END
```

```
$LARGE
      SUBROUTINE PROMPT (npos,dtout,sTime,fTime)

C
C FUNCTIONAL DESCRIPTION:
C
C       This subroutine prompts the user for information about the type of
C     data to write to the output file and for parameters that each type
C     of data require (e.g. size of wind turbine to model).
C
C DUMMY ARGUMENTS:
C    OUTPUTS:
C    npos                      number of positions on 'circle'
C    dtout                     time interval between samples of output data
C    sTime                     start time of output data
C    fTime                     finish time of output data
C
C COMMON INPUTS:
C
C    month, day                date of input file
C    hour, minute, second      time of input file
C    system                    number of data acquisition system used
C    dtin                      time interval between samples of input data
C    units(1..nchin)           units of measurement of each input data channel
C
C COMMON OUTPUTS:
C
C    nchout                    number of channels of data in output file
C    dattyp(1..nchout)         data type of each output data channel
C    r(1..nchout)              radial location of each output data channel
C    shadow(1..nchout)         indicates whether to include tower shadow
C    mean(1..nchout)           indicates whether to subtract mean wind
C    nfilt(1..nchout)          number of data to use in low pass filter
C
C WRITTEN OUTPUTS:
C
C    month, day                date of input file
C    hour, minute, second      time of input file
C    system                    number of data acquisition system used
C    chout                     number of output data channel
C    desout(1..nchout)         description of each output data channel
C    untout                    units of measurement of output data channel
C    m                         conversion factor of output data channel
C    b                         offset of output data channel
C
C SIDE EFFECTS:
C
C    none
C
C
      INTEGER   maxin, maxout, maxpos
      PARAMETER (maxin=100, maxout=50, maxpos=100)
```

```
      INTEGER*2 chout, dummy2(maxin)
      INTEGER   month, day, hour, minute, second, system
      INTEGER   chnl, nchout, npos
      INTEGER   dattyp(maxout), sqnc(maxin,maxout), nfilt(maxout)
      INTEGER   dummy1, dummy4(maxout)
      REAL      height(maxin,maxout), azimth(maxin,maxout)
      REAL      dtin, dtout, sTime, fTime, avdata(maxin)
      REAL      h, radius, r(maxout), rpm, m, b
      REAL      dummy5(maxpos,maxout)
      LOGICAL   shadow(maxout), mean(maxout), nomean, same
      CHARACTER dummy3(maxin)*30, units(maxin)*20
      CHARACTER desout(maxin)*30, untout*20, answer*1

      COMMON /header/ month, day, hour, minute, second, system
      COMMON /chinfo/ dummy1, dummy2, dtin, dummy3, units
      COMMON /output/ nchout, dattyp, dummy4, nfilt,
     1                sqnc, r, dummy5, shadow, mean

10    FORMAT(/, ' Enter the number of output fields desired : ', $)
20    FORMAT(//, ' What type of data do you want in output field ',
     1         I2,' ?', /)
21    FORMAT('       1) Single Tower Rotational Sampling', /)
22    FORMAT('       2) Circular array rotational sampling', /)
23    FORMAT('       3) Eulerian wind speed', /)
30    FORMAT(' Select one of the above : ', $)
50    FORMAT(/, ' Enter hub height of wind turbine in meters : ', $)
60    FORMAT(' Enter rotor radius of wind turbine in meters : ', $)
70    FORMAT(' Enter radial location of analysis')
71    FORMAT(' (i.e. 1.0 = 1.0*radius, 0.5 = 0.5*radius, etc.) : ', $)
90    FORMAT(' Enter rotation rate of wind turbine in rpm: ', $)
100   FORMAT(' Enter number of positions on circle ', $)
101   FORMAT('to rotationally sample : ', $)
110   FORMAT(' Do you want to include tower shadow (y or n) ? ', $)
111   FORMAT(A)
120   FORMAT(' Enter new time interval between samples : ', $)
130   FORMAT(' Do you want to use the same anemometers ', $)
131   FORMAT('as output field ', I2, ' (y or n) ? ', $)
140   FORMAT(' Do you want to subtract the mean ', $)
141   FORMAT('from each measurement (y or n) ? ', $)
150   FORMAT(' Enter number of data points for low-pass filter')
151   FORMAT(' (use odd number, 1 = no filter) : ', $)
160   FORMAT(' Enter description of output field : ', $)
170   FORMAT(//, ' Enter the start time and finish time ',$)
171   FORMAT('in seconds into the file : ')
180   FORMAT(1X, 2I2, 1X, 3I2, 1X, I1, 1X, I3, 1X, F9.4)
181   FORMAT(1X, I3, 1X, A30, 1X, A20, 2F9.4)

C
C     < Prompt user for number of output channels
C
      WRITE(*,10)
      READ(*,*) nchout
```

```
C
C     < Initialize output file parameters
C
      h = -1.0
      radius = -1.0
      rpm = -1.0
      dtout = -1.0
      npos = 1
C
C     < Prompt user for parameters needed for each output channel
C
      DO 400, chnl = 1, nchout
200     CONTINUE
          WRITE(*,20) chnl
          WRITE(*,21)
          WRITE(*,22)
          WRITE(*,23)
          WRITE(*,30)
          READ(*,*) dattyp(chnl)
        IF (dattyp(chnl).GT.3) GO TO 200
        IF (dattyp(chnl).EQ.1) THEN
          IF (h.LT.0.0) THEN
            WRITE(*,50)
            READ(*,*) h
          END IF
          IF (radius.LT.0.0) THEN
            WRITE(*,60)
            READ(*,*) radius
          END IF
        END IF
        IF (dattyp(chnl).EQ.1 .OR. dattyp(chnl).EQ.2) THEN
          WRITE(*,70)
          WRITE(*,71)
          READ(*,*) r(chnl)
          IF (rpm.LT.0.0) THEN
            WRITE(*,90)
            READ(*,*) rpm
          END IF
          IF (dtout.LT.0.0) THEN
            WRITE(*,100)
            WRITE(*,101)
            READ(*,*) npos
            dtout = (60./rpm) / FLOAT(npos)
          ELSE IF (npos .EQ. 1) THEN
            npos = NINT((60./rpm) / dtout)
          END IF
        END IF
        IF (dattyp(chnl).EQ.3) THEN
          IF (dtout.LT.0.0) THEN
            WRITE(*,120)
            READ(*,*) dtout
          END IF
```

```
        END IF
        same = .FALSE.
        IF (chnl.GT.1) THEN
          WRITE(*,130)
          WRITE(*,131) chnl-1
          READ(*,111) answer
          IF (answer.EQ.'y' .OR. answer.EQ.'Y')
     1      same = .TRUE.
        END IF
        CALL anemo(chnl,height,azimth,same)
        WRITE(*,140)
        WRITE(*,141)
        READ(*,111) answer
        IF (answer.EQ.'y' .OR. answer.EQ.'Y')
     1    mean(chnl) = .TRUE.
        WRITE(*,150)
        WRITE(*,151)
        READ(*,*) nfilt(chnl)
        WRITE(*,160)
        READ(*,111) desout(chnl)
400   CONTINUE
C
C     < Prompt user for start and finish time of data to analyze
C
      WRITE(*,170)
      WRITE(*,171)
      READ(*,*) sTime, fTime
      IF (sTime .LT. dtin) sTime = dtin
C
C     < Compute interpolation weighting factors and average wind
C     < speeds for each rotational position for each output channel
C
      nomean = .TRUE.
      DO 500, chnl = 1, nchout
        IF (mean(chnl).AND.nomean) THEN
          CALL averag(sTime, fTime, avdata)
          nomean = .FALSE.
        END IF
        IF (dattyp(chnl).EQ.1)
     1    CALL yzfact(npos,chnl,h,radius,height,avdata)
        IF (dattyp(chnl).EQ.2)
     1    CALL rtfact(npos,chnl,azimth,avdata)
        IF (dattyp(chnl).EQ.3)
     1    CALL zfact(npos,chnl,avdata)
 500  CONTINUE
C
C     < Write output file header
C
      WRITE (UNIT=4,FMT=180) month, day, hour, minute,
     1                   second, system, nchout, dtout
      m = 1.0
      b = 0.0
```

```
      DO 600, chnl = 1,nchout
        chout = chnl
        untout = units(sqnc(1,chnl))
        WRITE (UNIT=4,FMT=181) chout, desout(chnl), untout, m, b
600   CONTINUE
      RETURN
      END
```

```
$LARGE
     SUBROUTINE rtfact(npos,chnl,azimth,avdata)

C
C FUNCTIONAL DESCRIPTION:
C
C       This subroutine computes the interpolation weighting factors in
C     r-theta coordinates and assigns to an array indices of the anemometers
C     to use for the circular array of the current output field.  It also
C     computes the mean wind speed at each of the interpolated positions.
C
C DUMMY ARGUMENTS:
C    INPUTS:
C    npos                        number of positions on 'circle'
C    chnl                        sequence number of output data channel
C    azimth(1..nanem+1,chnl)     azimuth of each anemometer of output data channel
C    avdata(1..nchin)            average value of data for each input channel
C
C COMMON INPUTS:
C
C    nanemo(1..nchout+1)          number of anemometers of each output data channel
C    sqnc(1..nanemo+1,1..nchout)  input data channel sequence number
C    r(1..nchout)                 radial location of each output data channel
C    mean(1..nchout)              indicates whether to subtract mean wind
C
C COMMON OUTPUTS:
C
C    above(1..npos,1..nchout)   indice of anemometer above position on 'circle'
C    below(1..npos,1..nchout)   indice of anemometer below position on 'circle'
C    weight(1..npos,1..nchout)  weighting factor of position on 'circle'
C    avwind(1..npos,1..nchout)  average wind speed at position on 'circle'
C
C SIDE EFFECTS:
C
C    none
C
C

      INTEGER   maxin, maxout, maxpos
      PARAMETER (maxin=100, maxout=50, maxpos=100)
      INTEGER   chnl, pos, anem, npos, nanemo(maxout)
      INTEGER   above(maxpos,maxout), below(maxpos,maxout)
      INTEGER   sqnc(maxin,maxout)
      INTEGER   dummy1, dummy2(maxout), dummy3(maxout)
      REAL      weight(maxpos,maxout), azimth(maxin,maxout), theta
      REAL      r(maxout), avdata(maxin), wind
      REAL      avwind(maxpos,maxout), dummy9(maxpos,maxout)
      LOGICAL   mean(maxout), dummy7(maxout)

      COMMON /output/ dummy1, dummy2, nanemo, dummy3,
     1                sqnc, r, avwind, dummy7, mean
      COMMON /interp/ above, below, weight, dummy9
```

```
C
C     < Compute angle, anemometer indices, weighting factor,
C     < and average wind speed for each rotational position
C
      DO 300, pos = 1,npos
        theta = FLOAT(pos-1)/FLOAT(npos) * 360.
        DO 100, anem = 2,nanemo(chnl)+1
          IF (theta.LE.azimth(anem,chnl)) THEN
            above(pos,chnl) = anem
            IF (anem.GT.nanemo(chnl)) above(pos,chnl) = 1
            below(pos,chnl) = anem-1
            weight(pos,chnl) = (theta-azimth(below(pos,chnl),chnl))
     1                        / (azimth(above(pos,chnl),chnl)
     1                          -azimth(below(pos,chnl),chnl))
            GO TO 200
          END IF
100     CONTINUE
200     IF (mean(chnl)) THEN
          wind = avdata(sqnc(below(pos,chnl),chnl))
     1         + (avdata(sqnc(above(pos,chnl),chnl))
     1          - avdata(sqnc(below(pos,chnl),chnl)))
     1          * weight(pos,chnl)
          avwind(pos,chnl) = wind
     1                     + (avdata(sqnc(nanemo(chnl)+1,chnl))
     1                     - wind)*(1.0-r(chnl))
        END IF
300   CONTINUE
      RETURN
      END
```

```
$LARGE
      REAL FUNCTION strs(pos,chnl,tout)

C
C FUNCTIONAL DESCRIPTION:
C
C        This function computes the STRS-2 modeled wind speed for the current
C     blade position and current output field.
C
C DUMMY ARGUMENTS:
C
C    pos                     position on 'circle'
C    chnl                    sequence number of output data channel
C    tout                    output time
C
C COMMON INPUTS:
C
C    y(1..npos,1..nchout)    horizontal distance from tower to 'circle'
C    dtin                    time interval between samples of input data
C
C COMMON OUTPUTS:
C
C    none
C
C FUNCTION VALUE:
C
C    strs                    single tower rotationally sampled wind speed
C
C SIDE EFFECTS:
C
C    none
C
C
      INTEGER   maxin, maxout, maxpos
      PARAMETER (maxin=100, maxout=50, maxpos=100)
      INTEGER*2 dummy2(maxin)
      LOGICAL   done
      INTEGER   pos, chnl, index1, index2, increm, dummy1
      INTEGER   dummy5(maxpos,maxout), dummy6(maxpos,maxout)
      REAL      dtin, tout, y(maxpos,maxout), rindex
      REAL      twght, ws, ws1, ws2, yadv1, yadv2
      REAL      dummy7(maxpos,maxout)
      CHARACTER dummy3(maxin)*30, dummy4(maxin)*20

      COMMON /chinfo/ dummy1, dummy2, dtin, dummy3, dummy4
      COMMON /interp/ dummy5, dummy6, dummy7, y
      COMMON /finish/ done
C
C     < Compute record numbers of input file
C     < surrounding current output time
C     < and assign direction in which to increment input file
C
```

```
      rindex = tout/dtin
      IF (y(pos,chnl).GE.0.0) THEN
        index1 = INT(rindex)
        index2 = index1 + 1
        increm = 1
      ELSE
        index2 = INT(rindex)
        index1 = index2 + 1
        increm = -1
      END IF
C
C     < Compute time interpolation weighting factor
C
      twght = ABS(tout - FLOAT(index1)*dtin) / dtin
C
C     < Get wind speeds from input file
C     < surrounding current output time
C     < for current rotational position
C
      ws1 = ws(pos,chnl,index1)
      ws2 = ws(pos,chnl,index2)
C
C     < Compute wind speed at current output time
C
      ws1 = ws1 + (ws2-ws1)*twght
C
C     < Compute distance air is advected from tower
C
      yadv1 = 0.0
      yadv2 = (ws1+ws2)/2. * (1.-twght)*dtin
C
C     < Continue above process until advection distance from tower
C     < is greater than or equal to horizontal distance from tower
C     < of current rotational position.
C
c     < Check to see if the data read exceeded the end of the file
c     < if so, then exit the STRS subroutine.
c
      IF (done) GO TO 200
100   CONTINUE
        IF (yadv2 .LT. ABS(y(pos,chnl))) THEN
          index2 = index2 + increm
          ws1 = ws2
          ws2 = ws(pos,chnl,index2)
          yadv1 = yadv2
          yadv2 = yadv2 +(ws1+ws2)/2. * dtin
          GO TO 100
        END IF
200   CONTINUE
C
C     < Compute wind speed at horizontal distance from tower
C     < of current rotational position
```

```
C
      strs = ws1 + (ws2-ws1) * (ABS(y(pos,chnl))-yadv1)/(yadv2-yadv1)
      END
```

```
      real function sind(theta)
c
c       This subroutine was written to be the equivalent of the
c     VAX function of the same name.
c
c     INPUT:
c          theta             degrees that the sine function is to
c                              be calculated for.
c
c     FUNCTIONAL VALUE:
c          sind              the sine of the degrees passed to the routine.
c
      REAL FRADS, THETA, PI, DEGRAD
      PI = ATAN(1.) * 4
      DEGRAD = 180./PI
      frads = theta / degrad
      sind = sin(frads)
      end

      real function cosd(theta)
c
c       This subroutine was written to be the equivalent of the
c     VAX function of the same name.
c
c     INPUT:
c          theta             degrees that the cosine function is to
c                              be calculated for.
c
c     FUNCTIONAL VALUE:
c          cosd              the cosine of the degrees passed to the routine.
c

      real frads, theta, DEGRAD, PI

      PI = ATAN(1.0) * 4
      DEGRAD = 180./PI
      frads = theta / degrad
      cosd = cos(frads)
      end
```

```
$LARGE
      REAL FUNCTION vpa(pos,chnl,tout)

C
C FUNCTIONAL DESCRIPTION:
C
C       This function computes the circular array wind speed for the current
C     blade position and current output field.
C
C DUMMY ARGUMENTS:
C
C    pos                       position on 'circle'
C    chnl                      sequence number of output data channel
C    tout                      output time
C
C COMMON INPUTS:
C
C    dtin                      time interval between samples of input data
C
C COMMON OUTPUTS:
C
C    none
C
C FUNCTION VALUE:
C
C    vpa                       vertical plane array wind speed
C
C SIDE EFFECTS:
C
C    none
C
C

      INTEGER   maxin
      PARAMETER (maxin=100)
      INTEGER*2 dummy2(maxin)
      INTEGER   pos, chnl, index1, index2, dummy1
      REAL      dtin, tout, rindex
      REAL      twght, ws, ws1, ws2
      CHARACTER dummy3(maxin)*30, dummy4(maxin)*20

      COMMON /chinfo/ dummy1, dummy2, dtin, dummy3, dummy4

C
C     < Compute record numbers of input file
C     < surrounding current output time
C
      rindex = tout/dtin
      index1 = INT(rindex)
      index2 = index1 + 1
C
C     < Compute time interpolation weighting factor
```

```
C
      twght = ABS(tout - FLOAT(index1)*dtin) / dtin
C
C     < Get wind speeds from input file
C     < surrounding current output time
C     < for current rotational position
C
      ws1 = ws(pos,chnl,index1)
      ws2 = ws(pos,chnl,index2)
C
C     < Compute wind speed at current output time
C
      vpa = ws1 + (ws2-ws1)*twght
      END
```

```
$LARGE
      REAL FUNCTION ws(pos,chnl,indx)

C
C FUNCTIONAL DESCRIPTION:
C
C        This function reads the required records of the input file and computes
C      the filtered wind speed value for the current output data type.
C
C DUMMY ARGUMENTS:
C
C     pos                           position on 'circle'
C     chnl                          sequence number of output data channel
C     indx                          record index of input file
C
C COMMON INPUTS:
C
C     nchin                         number of channels of data in input file
C     dattyp(1..nchout)             data type of each output data channel
C     nanemo(1..nchout)             number of anemometers of each output data channel
C     sqnc(1..nanemo+1,1..nchout)   input data channel sequence number
C     nfilt(1..nchout)              number of data to use in low pass filter
C     above(1..npos,1..nchout)      indice of anemometer above position on 'circle'
C     below(1..npos,1..nchout)      indice of anemometer below position on 'circle'
C     weight(1..npos,1..nchout)     weighting factor of position on 'circle'
C     r(1..nchout)                  radial location of each output data channel
C
C COMMON OUTPUTS:
C
C     done                          indicates whether analysis is done
C
C FUNCTION VALUE:
C
C     ws                            wind speed from input file
C
C SIDE EFFECTS:
C
C     none
C
C

      INTEGER    maxin, maxout, maxpos
      PARAMETER  (maxin=100, maxout=50, maxpos=100)
      INTEGER*2  dummy1(maxin)
      INTEGER    pos, chnl, ch, i, nchin
      INTEGER    rec1, rec2, indx, count, ios
      INTEGER    sqnc(maxin,maxout), nfilt(maxout)
      INTEGER    above(maxpos,maxout), below(maxpos,maxout)
      INTEGER    dattyp(maxout), nanemo(maxout)
      INTEGER    dummy5
      REAL       eTime, data(maxin), r(maxout), weight(maxpos,maxout)
      REAL       wspos
```

```
      REAL      dummy2, dummy6(maxpos,maxout), dumy11(maxpos,maxout)
      LOGICAL   done, dummy9(maxout), dumy10(maxout)
      CHARACTER dummy3(maxin)*30, dummy4(maxin)*20

      COMMON /chinfo/ nchin, dummy1, dummy2, dummy3, dummy4
      COMMON /output/ dummy5, dattyp, nanemo, nfilt,
     1                sqnc, r, dummy6, dummy9, dumy10
      COMMON /interp/ above, below, weight, dumy11
      COMMON /finish/ done

10    FORMAT(1X,F9.2,100(F8.3,3X))

C
C     < Compute record numbers of input file
C     < bounding data to be filtered
C
      rec1 = indx - (nfilt(chnl)-1)/2
      IF (rec1 .LT. 1) rec1 = 1
      rec2 = indx + (nfilt(chnl)-1)/2
      IF (rec2 .LT. 1) rec2 = 1
C
C     < Initialize wind speed and number of records read
C
      ws = 0.0
      count = 0
C
C     < Read time and data from input file
C     < and compute space-weighted wind speed
C     < for current output channel and rotational position
C
      DO 100, i = rec1,rec2
        READ (UNIT=1, REC=i, IOSTAT=ios, FMT=10)
     1  eTime, (data(ch), ch=1,nchin)

        IF (ios .EQ. -1) THEN
          WRITE(*,*) 'END OF FILE ENCOUNTERED'
          done = .TRUE.
          GO TO 200
        END IF

        IF (dattyp(chnl) .EQ. 1) THEN
          wspos = data(sqnc(below(pos,chnl),chnl))
     1         + (data(sqnc(above(pos,chnl),chnl))
     1         -  data(sqnc(below(pos,chnl),chnl)))
     1         * weight(pos,chnl)
          ws = ws + wspos
        END IF

        IF (dattyp(chnl) .EQ. 2) THEN
          wspos = data(sqnc(below(pos,chnl),chnl))
     1         + (data(sqnc(above(pos,chnl),chnl))
     1         -  data(sqnc(below(pos,chnl),chnl)))
```

```
     1          * weight(pos,chnl)
          ws = ws + wspos + (data(sqnc(nanemo(chnl)+1,chnl))
     1         - wspos)*(1.0-r(chnl))
        END IF

        IF (dattyp(chnl) .EQ. 3) THEN
          wspos = data(sqnc(pos,chnl))
          ws = ws + wspos
        END IF
        count = count + 1
100   CONTINUE
C
C     < Compute filtered wind speed
C
200   CONTINUE
      If (count.GT.0) ws = ws/FLOAT(count)

      END
```

```
$LARGE
      SUBROUTINE WSDATA (pos, tout)

C
C FUNCTIONAL DESCRIPTION:
C
C       This subroutine determines and writes to the output file the wind speed
C     data for each output field for the current output time.
C
C DUMMY ARGUMENTS:
C    INPUTS:
C    pos                          position on 'circle'
C    tout                         output time
C
C COMMON INPUTS:
C
C    nchout                       number of channels of data in output file
C    dattyp(1..nchout)            data type of each output data channel
C    avwind(1..npos,1..nchout) average wind speed at position on 'circle'
C    shadow(1..nchout)            indicates whether to include tower shadow
C    mean(1..nchout)              indicates whether to subtract mean wind
C
C WRITTEN OUTPUTS:
C
C    tout                         output time
C    RSwind(1..nchout)            computed wind speed of each output data channel
C
C SIDE EFFECTS:
C
C    none
C
C

      INTEGER   maxin, maxout, maxpos
      PARAMETER (maxin=100, maxout=50, maxpos=100)
      INTEGER   pos, chnl, nchout
      INTEGER   dattyp(maxout)
      INTEGER   dummy1(maxout), dummy2(maxout), dummy3(maxin,maxout)
      REAL      RSwind(maxout), strs, vpa, euler
      REAL      avwind(maxpos,maxout), tout, dummy4(maxout)
      LOGICAL   shadow(maxout), mean(maxout)

      COMMON /output/ nchout, dattyp, dummy1, dummy2,
     1                dummy3, dummy4, avwind, shadow, mean

 10   FORMAT ('+Output time = ', F8.2, ' seconds')
 20   FORMAT (1X,F9.2,50(F8.3,3X))

C
C     < Display current time of output data
C
      WRITE (*,10) tout
```

```
C
C     < Compute rotationally-sampled wind speed
C     < for each output channel
C
      DO 100, chnl=1,nchout
        IF (dattyp(chnl).EQ.1) RSwind(chnl) = strs(pos,chnl,tout)
        IF (dattyp(chnl).EQ.2) RSwind(chnl) = vpa(pos,chnl,tout)
        IF (dattyp(chnl).EQ.3) RSwind(chnl) = euler(chnl,tout)
        IF (mean(chnl)) RSwind(chnl) = RSwind(chnl)
     1                                  - AVwind(pos,chnl)
100   CONTINUE
C
C     < Write time and wind speed data
C     < for each output channel
C
      WRITE (UNIT=2, FMT=20) tout, (RSwind(chnl), chnl=1,nchout)
      RETURN
      END
```

```
$LARGE
      SUBROUTINE yzfact(npos,chnl,h,radius,height,avdata)
C
C FUNCTIONAL DESCRIPTION:
C
C       This subroutine computes the interpolation weighting factors in
C     yz coordinates and assigns to an array indices of the anemometers
C     to use for the STRS-2 model of the current output field.  It also
C     computes the mean wind speed at each of the interpolated positions.
C
C DUMMY ARGUMENTS:
C    INPUTS:
C    npos                      number of positions on 'circle'
C    chnl                      sequence number of output data channel
C    radius                    radius of wind turbine rotor
C    h                         hub height of wind turbine
C    height(1..nanem,chnl)     height of each anemometer of output data channel
C    avdata(1..nchin)          average value of data for each input channel
C
C COMMON INPUTS:
C
C    nanemo(1..nchout)         number of anemometers of each output data channel
C    sqnc(1..nanemo,1..nchout) input data channel sequence number
C    r(1..nchout)              radial location of each output data channel
C    mean(1..nchout)           indicates whether to subtract mean wind
C
C COMMON OUTPUTS:
C
C    y(1..npos,1..nchout)      horizontal distance from tower to 'circle'
C    above(1..npos,1..nchout)  indice of anemometer above position on 'circle'
C    below(1..npos,1..nchout)  indice of anemometer below position on 'circle'
C    weight(1..npos,1..nchout) weighting factor of position on 'circle'
C    avwind(1..npos,1..nchout) average wind speed at position on 'circle'
C
C SIDE EFFECTS:
C
C    none
C
C

      INTEGER   maxin, maxout, maxpos
      PARAMETER (maxin=100, maxout=50, maxpos=100)
      INTEGER   chnl, pos, anem, npos, nanemo(maxout)
      INTEGER   above(maxpos,maxout), below(maxpos,maxout)
      INTEGER   sqnc(maxin,maxout)
      INTEGER   dummy1, dummy2(maxout), dummy3(maxout)
      REAL      height(maxin,maxout), avdata(maxin)
      REAL      r(maxout), y(maxpos,maxout), weight(maxpos,maxout)
      REAL      radius, rad, theta, LxLy, z, h, avwind(maxpos,maxout)
      LOGICAL   mean(maxout), dummy6(maxout)
      PARAMETER (LxLy=2.5)
```

```
      COMMON /output/ dummy1, dummy2, nanemo, dummy3,
     1                sqnc, r, avwind, dummy6, mean
      COMMON /interp/ above, below, weight, y
C
C     < Compute radius of 'circle' for current output channel
C
      rad = r(chnl)*radius
C
C     < Compute distance, height, anemometer indices, weighting
C     < factor, and average wind speed for each rotational position
C
      DO 300, pos = 1,npos
        theta = FLOAT(pos-1)/FLOAT(npos) * 360.
        y(pos,chnl) = rad*SIND(theta) * LxLy
        z = h + rad*COSD(theta)
        DO 100, anem = 1,nanemo(chnl)
          IF (z.LE.height(anem,chnl)) THEN
            IF (anem.EQ.1) THEN
              above(pos,chnl) = anem+1
              below(pos,chnl) = anem
            ELSE
              above(pos,chnl) = anem
              below(pos,chnl) = anem-1
            END IF
            GO TO 200
          ELSE IF (anem.EQ.nanemo(chnl)) THEN
            above(pos,chnl) = anem
            below(pos,chnl) = anem-1
          END IF
100     CONTINUE
200     CONTINUE
        weight(pos,chnl) = (z-height(below(pos,chnl),chnl))
     1                    / (height(above(pos,chnl),chnl)
     1                      -height(below(pos,chnl),chnl))
        IF (mean(chnl))
     1    avwind(pos,chnl) = avdata(sqnc(below(pos,chnl),chnl))
     1                   + (avdata(sqnc(above(pos,chnl),chnl))
     1                    - avdata(sqnc(below(pos,chnl),chnl)))
     1                    * weight(pos,chnl)

300   CONTINUE
      RETURN
      END
```

```
$LARGE
      SUBROUTINE zfact(npos,chnl,avdata)

C    .
C FUNCTIONAL DESCRIPTION:
C
C       This subroutine computes the average of the mean wind speed of each of
C     the specified anemometers.
C
C DUMMY ARGUMENTS:
C    INPUTS:
C    npos                       number of positions on 'circle'
C    chnl                       sequence number of output data channel
C    avdata(1..nchin)           average value of data for each input channel
C
C COMMON INPUTS:
C
C    nanemo(1..nchout)          number of anemometers of each output data channel
C    sqnc(1..nanemo,1..nchout) input data channel sequence number
C    mean(1..nchout)            indicates whether to subtract mean wind
C
C COMMON OUTPUTS:
C
C    avwind(1..npos,1..nchout) average wind speed
C
C SIDE EFFECTS:
C
C    none
C
C

      INTEGER   maxin, maxout, maxpos
      PARAMETER (maxin=100, maxout=50, maxpos=100)
      INTEGER   pos, chnl, anem, npos, nanemo(maxout)
      INTEGER   sqnc(maxin,maxout)
      INTEGER   dummy1, dummy2(maxout), dummy3(maxout)
      REAL      avdata(maxin), wind
      REAL      avwind(maxpos,maxout), dummy5(maxout)
      LOGICAL   mean(maxout), dummy6(maxout)

      COMMON /output/ dummy1, dummy2, nanemo, dummy3,
     1                sqnc, dummy5, avwind, dummy6, mean

C
C     < Initialize wind speed
C
      wind = 0.0
C
C     < Compute average wind speed
C
      IF (mean(chnl)) THEN
        DO 100, anem = 1,nanemo(chnl)
```

```
          wind = wind + avdata(sqnc(anem,chnl))
100     CONTINUE
        wind = wind/FLOAT(nanemo(chnl))
      END IF
C
C     < Assign average wind speed to each rotational position
C
      DO 200, pos = 1, npos
        avwind(pos,chnl) = wind
200   CONTINUE
      RETURN
      END
```

APPENDIX B

STRS-2 ERROR AND USE OF CORRECTION FACTORS

APPENDIX B

STRS-2 ERROR AND USE OF CORRECTION FACTORS

B.1 COMPARISON OF STRS-2 AND FULL VERTICAL PLANE ARRAY ROTATIONAL WINDS

The basic data that we have at our disposal for this discussion are three sets of spectral analyses of STRS-2 time series for three different sites. Each set was derived from either three or five uniformly spaced anemometers on the center tower of a large VPA that supported anemometers in a crosswind circle. Thus, winds rotationally sampled around the crosswind circle were also available against which to compare the STRS-2 model estimate of those winds. The arrays were not specifically designed to optimize development of the model as a function of site differences, and they were not identical in diameter, hub height, or in numbers of anemometers around the circular ring. Nevertheless, they are all the measurements that are useful to study the accuracy of the STRS-2 method. We discuss the results in the light of those array differences and the physical phenomena that determine the characteristics of the rotationally sampled wind.

The error factor, or inversely, the correction factor that we calculated for each example is derived from a simplification of the spectra of the rotationally sampled wind. The variance associated with each harmonic of the rotation frequency of the modeled wind turbine is computed to provide a single number representing the spectral energy in that band. Then the variance for each spectral band derived from the "exact" rotational time series using the complete vertical plane array circle is divided by the corresponding variance for the corresponding spectral band from the STRS-2 spectrum. Twelve examples of curves of these ratios as a function of the harmonic number are plotted in Figure B.1. All of the low-frequency energy in the spectrum was collected into the subharmonic number 1/2.

The curves in Figure B.1 represent the site at Clayton, New Mexico, a relatively smooth and uniform site. The legend code identifies the Clayton site by the initial "C". This is followed with two sets of four numerals that specify the date (month and day) and the time of day (hour and minute in local standard time) of the start of the measurement of the time series. Four columns of numbers giving the mean wind speed in m/s, the variance of the

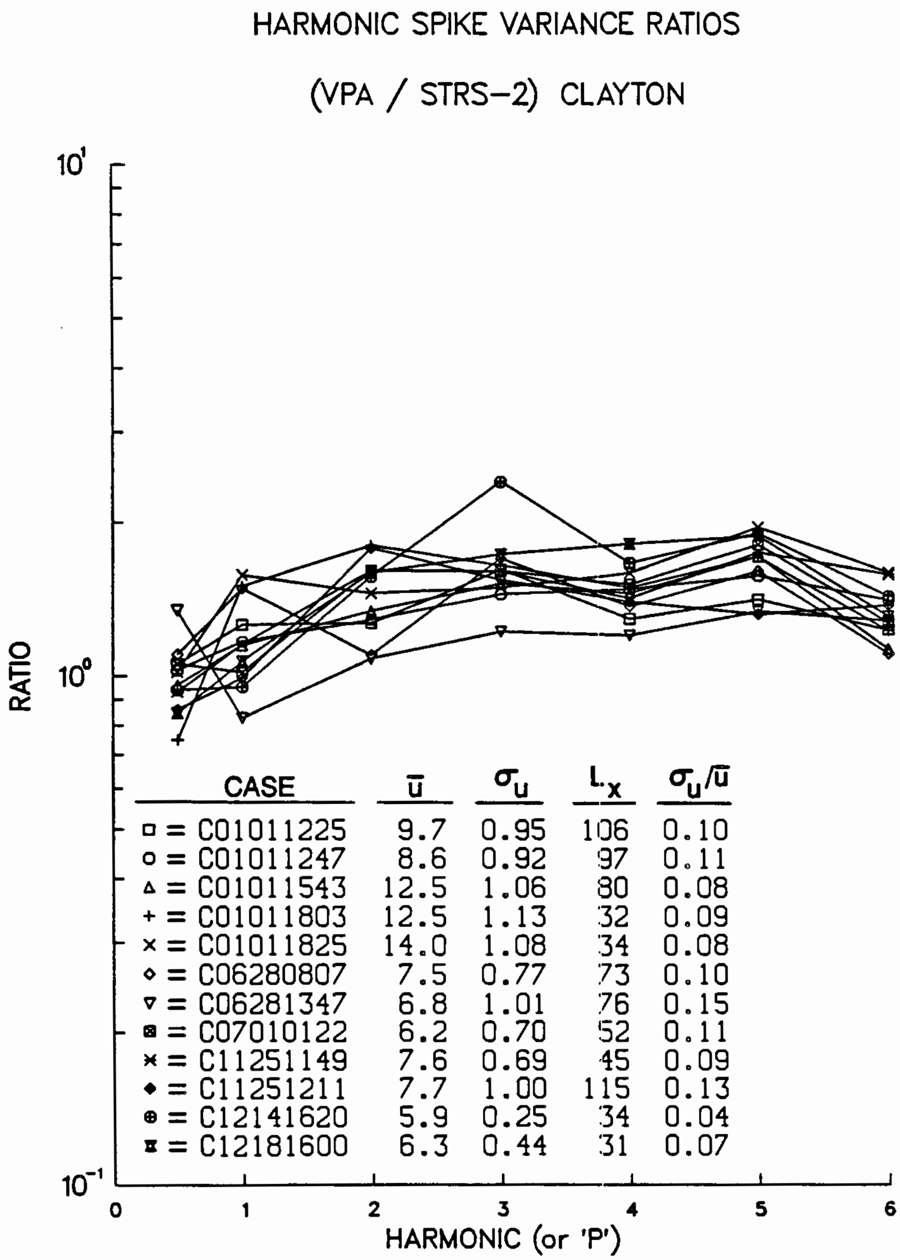

FIGURE B.1. Correction Factors for STRS-2 Harmonic Band Variance for 12 Cases at Clayton, New Mexico. The mean and standard deviation of the wind speed in m/s, integral length scale in m, and turbulence intensity for each case is given.

wind speed in m/s, the integral length scale in m (computed by integrating the autocorrelation function of hub-height wind speed to the first maximum of the integral), and the turbulence intensity follow the case identification numbers.

The ratio may be viewed as a correction factor in the sense that multiplication of a value of STRS-2 variance by the value of the ratio for the same frequency band gives the value of variance for the same frequency band in the full rotationally measured wind, as determined from the full VPA. Correction factor values for individual harmonics among all of the 12 cases range from 0.8 to 2.5, a factor of 3. Some sort of mean curve representing all cases would vary between about 1.0 and 1.8. Note that the spread in cases spans a factor of about 2. Much more measurement and careful analysis would have to be done to develop a correction factor relation that accurately represented the variety of atmospheric conditions at this simple site. It is not clear that the physics is well enough captured with STRS-2 to justify such an effort to determine whether the ratios are accurate correction factors.

The ratio curves for wind cases at a more complex site in the downslope flow from San Gorgonio Pass in California are shown in Figure B.2. Notice that the character of the curves is similar to those for Clayton in the lower frequencies, but dramatically reverses slope at the highest frequency. Correction factors range from 0.5 to about 1.6, a factor of 3.

Finally, curves for the most complex site in the downslope winds from Altamont Pass, California, are shown in Figure B.3. For this site, the curves have a completely reversed slope from those for Clayton. The correction factors range from 0.4 to 1.2, again a factor of 3.

The range of variation of the ratio reported in these figures may reflect the number of cases measured at each site and the restricted range of atmospheric boundary-layer conditions represented in the limited data sets. The data plotted in the figures are printed for the reader's use in Table B.1. In the rest of this appendix, we discuss some additional possible reasons for the variations in the ratios of VPA to STRS-2 band variance.

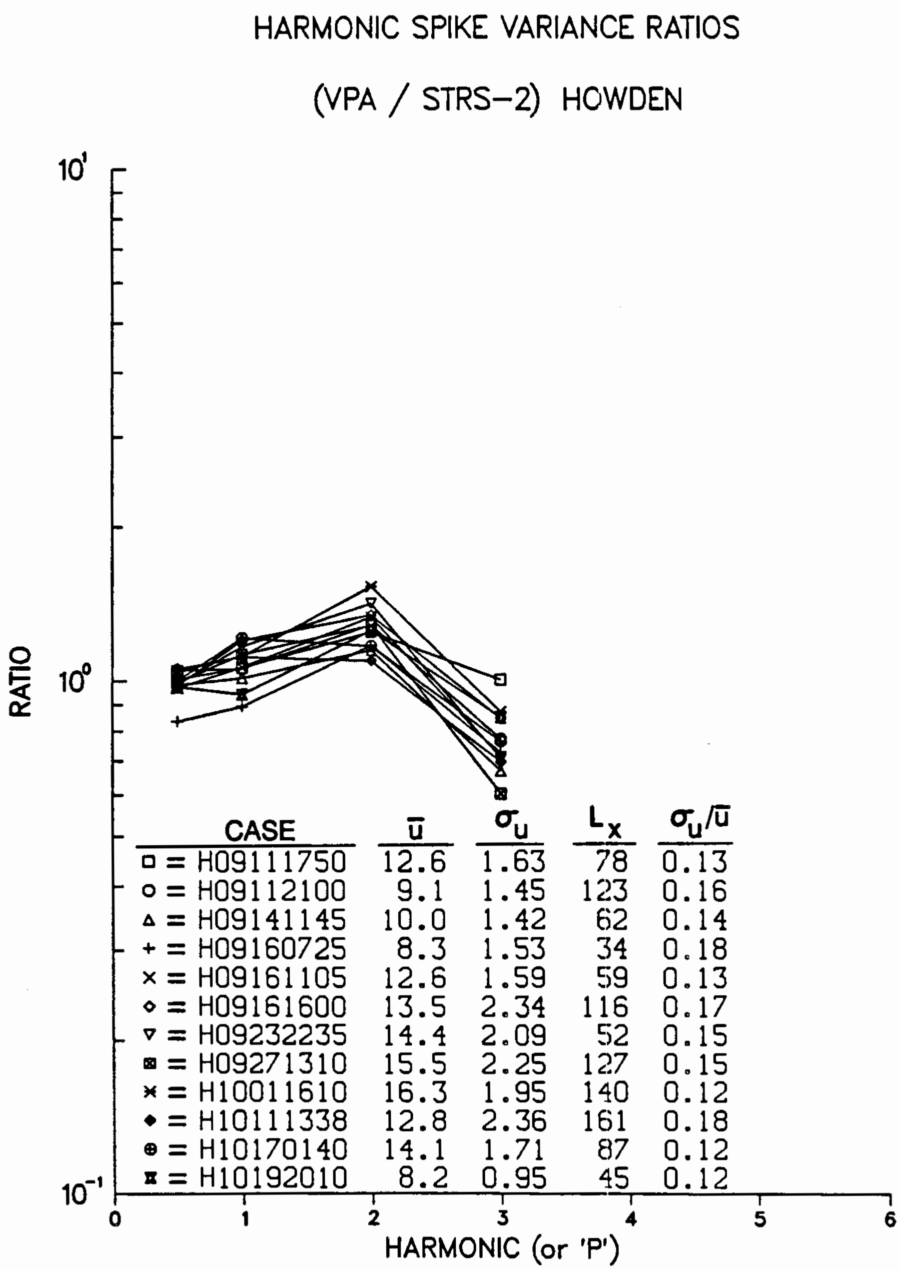

FIGURE B.2. Correction Factors for STRS-2 Harmonic Band Variance for 12 Cases at San Gorgonio Pass, California. The mean and standard deviation of the wind speed in m/s, integral length scale in m, and turbulence intensity for each case is given.

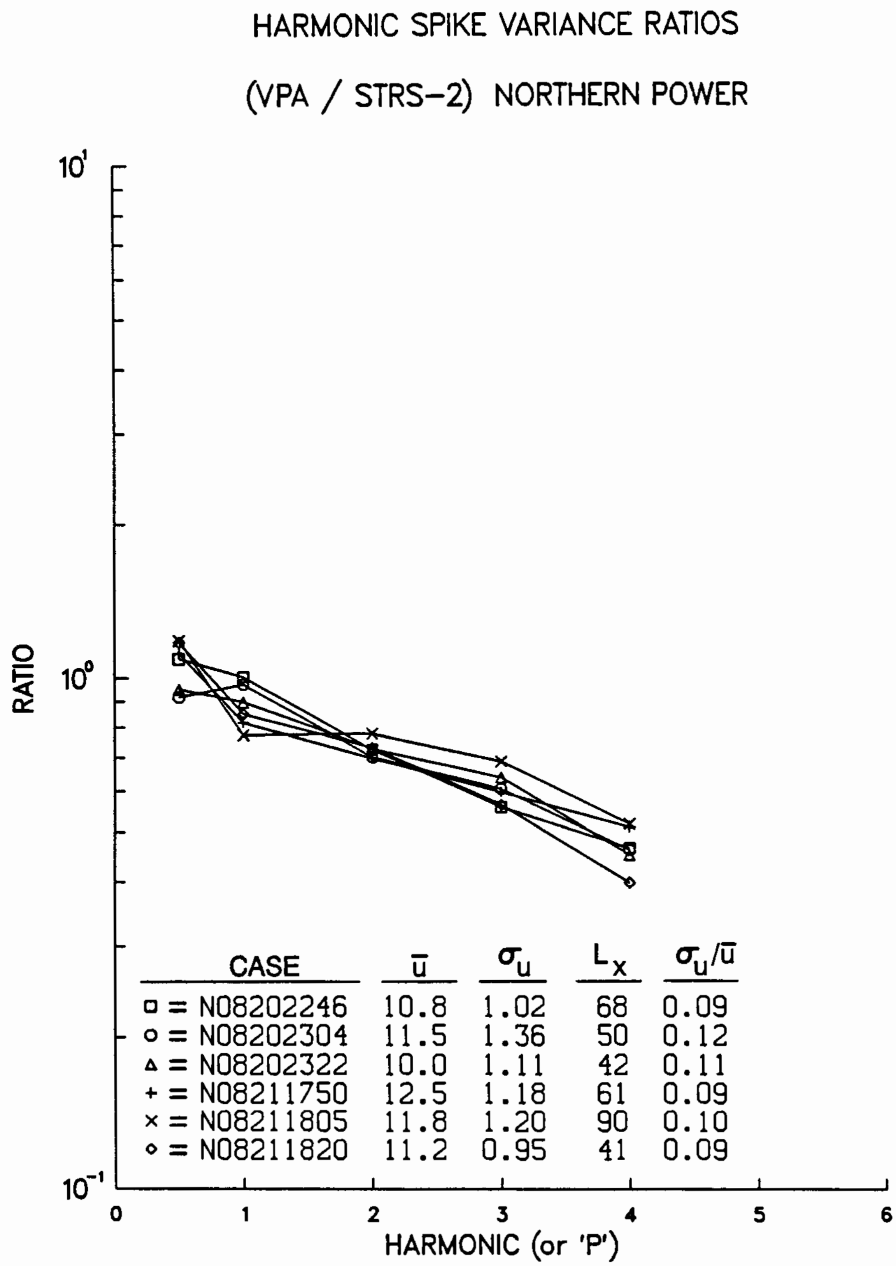

FIGURE B.3. Correction Factors for STRS-2 Harmonic Band Variance for 6 Cases at Altamont Pass, California. The mean and standard deviation of the wind speed in m/s, integral length scale in m, and turbulence intensity for each case is given.

TABLE B.1. Correction Factor (f) Data for STRS-2 Harmonic Band Variances Compared to Vertical Plane Array Harmonic Band Variances of the Rotationally Sampled Wind

Clayton, New Mexico, MOA-OA Turbine, 12-Anemometer Circle

Case	$f_{1/2P}$	f_{1P}	f_{2P}	f_{3P}	f_{4P}	f_{5P}	f_{6P}
C01011225	1.05	1.26	1.27	1.62	1.29	1.41	1.23
C01011247	1.03	1.16	1.29	1.45	1.48	1.57	1.41
C01011543	0.96	1.15	1.34	1.52	1.45	1.72	1.13
C01011803	0.75	1.50	1.80	1.65	1.49	1.72	1.23
C01011825	1.02	1.58	1.45	1.49	1.60	1.96	1.60
C06280807	1.10	1.49	1.10	1.70	1.37	1.61	1.11
C06281347	1.34	0.83	1.08	1.22	1.20	1.34	1.29
C07010122	1.05	1.02	1.61	1.61	1.51	1.81	1.27
C11251149	0.93	1.15	1.61	1.60	1.42	1.75	1.59
C11251211	0.86	0.99	1.78	1.55	1.40	1.32	1.38
C12141620	0.94	0.95	1.57	2.40	1.67	1.91	1.44
C12181600	0.84	1.07	1.59	1.73	1.82	1.89	1.31

Altamont, California, NPS Turbine, 8-Anemometer Circle

Case	$f_{1/2P}$	f_{1P}	f_{2P}	f_{3P}	f_{4P}
A08202246	1.09	1.00	0.73	0.56	0.46
A08202304	0.92	0.97	0.70	0.61	0.46
A08202322	0.95	0.90	0.73	0.64	0.45
A08211750	1.12	0.82	0.70	0.60	0.51
A08211805	1.18	0.77	0.78	0.69	0.52
A08211820	1.17	0.85	0.73	0.56	0.40

San Gorgonio, California, Howden Turbine, 6-Anemometer Circle

Case	$f_{1/2P}$	f_{1P}	f_{2P}	f_{3P}
H09111750	1.01	1.06	1.24	1.00
H09112100	1.05	1.06	1.28	0.77
H09141145	0.98	1.01	1.15	0.67
H09160725	0.84	0.89	1.17	0.73
H09161105	0.97	1.06	1.33	0.71
H09161600	0.98	1.20	1.34	0.84
H09232235	0.99	1.17	1.41	0.70
H09271310	0.99	1.12	1.28	0.60
H10011610	1.05	1.12	1.53	0.87
H10111338	1.05	1.11	1.09	0.70
H10170140	1.00	1.21	1.17	0.76
H10192010	0.97	0.94	1.25	0.85

B.2 SITE AND ARRAY CONFIGURATION FACTORS

A more detailed discussion of site effects on Eulerian and rotational framework wind characteristics at the location of vertical plane arrays is given by Connell (1987). An expansion of Figure 3 in that paper to include the anemometer array geometry at the site in the downslope of Altamont Pass, California, is given in Figure B.4. Not only is the terrain different at each site, with the consequent differences in characteristic energies, turbulence intensities and integral scales of turbulence, but the anemometer arrays are different in diameter, hub height, and numbers of anemometers spaced equally around the array crosswind circle. These factors complicate the interpretation of variance ratio curves in assessing the modeling of the effects of terrain on the accuracy of the STRS-2 model.

A very important consideration is the length scales of turbulence eddies that carry most of the variance or energy of turbulence. The ideal measurement and analysis for each different site used in the comparison would be made with identical vertical plane arrays, representing a single wind turbine rotor at the three sites. Then the effect of changes in spectral shape and shift,

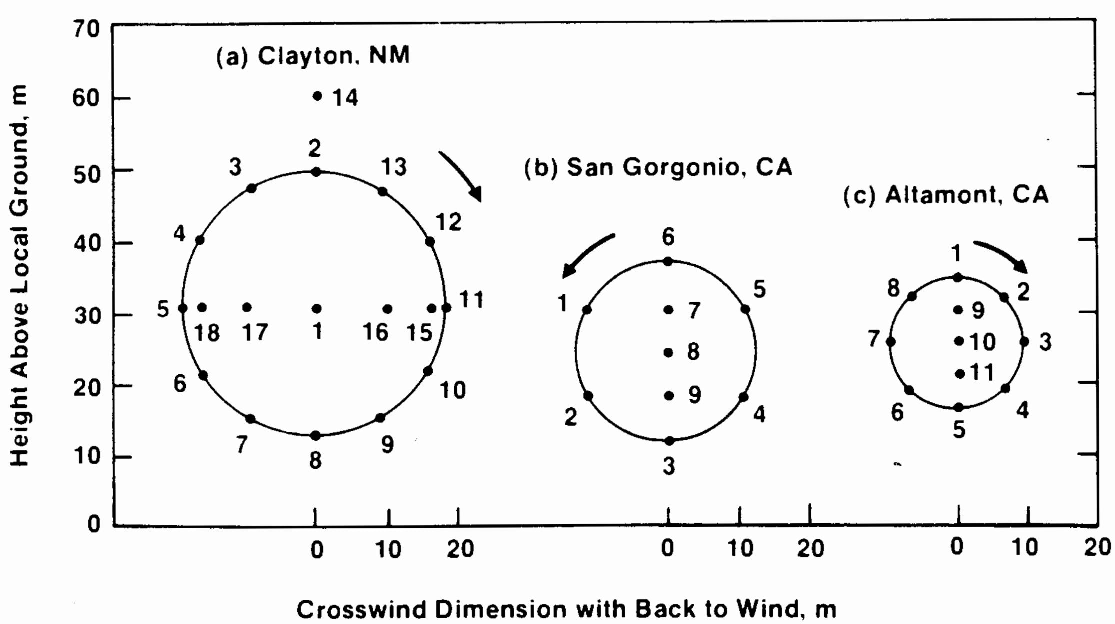

FIGURE B.4. The Geometry of Anemometer Arrays at (a) Clayton, New Mexico, (b) San Gorgonio, California, and (c) Altamont, California.

which relate to the length scales of turbulence, would be weighted on an equal basis at each site. The mean value of observed integral scales for the set of data at each site is as follows. The scale is 65 m at the more benign Clayton site, 90 m at San Gorgonio, and 59 m at Altamont, the site with the roughest terrain. In the latter case, the few measurements made were done on only two consecutive days and may not be as representative of the site as the data for the other sites.

Unfortunately, as the length scale increased with site complexity, the diameter of the array of anemometers used was smaller. One would argue that the smallest array would span less far across turbulence structures and there would be greater correlation across the array, and there would be greater correlation yet for larger-length-scale turbulence. Thus the STRS-2 model, which simulates crosswind correlation effects using exaggerated separation of along-wind correlation, needs to use a different exaggeration or ellipticity that is a function of both array diameter and turbulence length scale. No research on correlation vs. terrain has been undertaken by which to achieve a refinement of the original crude STRS-2 model. It is not known that an effort to do so is justified, until some such research has been done!

Another factor is the spatial density of wind measurements around the circles of the VPAs as well as along the center single tower. The more the anemometers, the better the rotationally sampled wind description. The ratio of the numbers of anemometers on the center tower and on the ring (shown in parentheses) varied from 4(12) at Clayton to 1.2(6) at San Gorgonio and 1.6(8) at Altamont.

At all sites, one interpretation is that the STRS-2 model is noisier and gives less accurate simulation as the frequency of the spectral function increases.

The distinct differences in the variance ratio curves for the different sites simply make it impossible at this stage of understanding to recommend correction factors for anything other than simple terrain with hydrostatically neutral or, possibly, unstable turbulent flow. Using single-tower simulation of an essentially three-dimensional space and time variation in wind at least represents part of the important turbulence characteristics that cannot be represented by theory. Each user must estimate the accuracy required for the problem to be solved. In the more demanding problems, those in complex terrain, some other measurement method must be used to obtain the important turbulence characteristics in the rotational frame of reference. Some of the potential methods are dense vertical plane arrays, rotationally scanning laser anemometers, or hotfilm anemometers attached to actual rotor blades.

B.3 REFERENCE

Connell, J. R., D. C. Powell, and V. R. Morris. 1987. "Site Effects on Wind Characteristics at a Turbine Rotor." In Proceedings of Windpower '87 Conference, American Wind Energy Association, Alexandria, Virginia.

DISTRIBUTION

No. of
Copies

OFFSITE

C. I. Aspliden
Battelle Memorial Institute
Washington Office
2030 M Street, NW
Washington, DC 20036

J. Cadogan
U.S. Department of Energy
Wind/Ocean Technologies Division
1000 Independence Avenue
Forrestal Building, Room 5F064
Washington, DC 20585

G. P. Tennyson
U.S. Department of Energy
Albuquerque Operations Office
P.O. Box 5400
Albuquerque, NM 87110

10 DOE/Office of Scientific and Technical Information

S. M. Chan
Systems Control, Inc.
1801 Page Mill Road
P.O. Box 10025
Palo Alto, CA 94303

R. L. George
TERRA Sciences, Inc.
7555 W. 10th Avenue
Lakewood, CO 80214

R. H. Kirchhoff
Department of Mechanical Engineering
University of Massachusetts
Amherst, MA 01003

P. Klimas
Sandia National Laboratories
Division 6225
Albuquerque, NM 87185

No. of
Copies

P. M. Moretti
Oklahoma State University
Mechanical and Aerospace Engineering
Engineering North 218
Stillwater, OK 74074

D. C. Reda
Sandia National Laboratories
Division 6225
Albuquerque, NM 87185

R. B. Schlueter
Department of Electrical Engineering
Michigan State University
East Lansing, MI 48824

D. Spera
NASA/Lewis Research Center
21000 Brookpark Road
Cleveland, OH 44135

J. P. Sullivan
Aeronautics & Astronautics
Grissom Hall
Purdue University
W. Lafayette, IN 47907

R. M. Sundar
Embry-Riddle Aeronautical University
3200 N. Willow Creek Road
Prescott, AZ 86301

R. W. Thresher
Solar Energy Research Institute
1617 Cole Boulevard
Golden, CO 80401

V. A. Vachon
P.O. Box 149
Manchester, MA 01944

No. of
Copies

P. Veers
2700 Del Medio Court #304
Mountain View, CA 94040

S. N. Walker
Department of Mechanical
Engineering
Oregon State University
Corvallis, OR 97331

ONSITE

DOE Richland Operations Office

J. J. Sutey/D. R. Segna

No. of
Copies

37 Pacific Northwest Laboratory

J. C. Barnard
J. R. Connell (10)
C. E. Elderkin
D. L. Elliott
J. W. Falco
M. J. Graham
J. M. Hales
P. C. Hays
R. A. Keefe (7)
V. R. Morris
E. L. Owczarski
D. C. Powell
J. A. Stottlemyre
L. L. Wendell
R. E. Wildung
Technical Information Files (5)
Publishing Coordination (2)