July 1999 • NREL/TP-500-26371

Technical Report

Conversion of Phase II Unsteady Aerodynamics Experiment Data to Common Format

M.M. Hand

National Renewable Energy Laboratory

1617 Cole Boulevard
Golden, Colorado 80401-3393

NREL is a U.S. Department of Energy Laboratory
Operated by Midwest Research Institute • Battelle • Bechtel

Contract No. DE-AC36-98-GO10337

July 1999 • NREL/TP-500-26371

Conversion of Phase II Unsteady Aerodynamics Experiment Data to Common Format

M.M. Hand

Prepared under Task No. WE901110

National Renewable Energy Laboratory
1617 Cole Boulevard
Golden, Colorado 80401-3393
NREL is a U.S. Department of Energy Laboratory
Operated by Midwest Research Institute • Battelle • Bechtel
Contract No. DE-AC36-98-GO10337

Available to DOE and DOE contractors from:
Office of Scientific and Technical Information (OSTI)
P.O. Box 62
Oak Ridge, TN 37831
Prices available by calling 423-576-8401

Available to the public from:
National Technical Information Service (NTIS)
U.S. Department of Commerce
5285 Port Royal Road
Springfield, VA 22161
703-605-6000 or 800-553-6847
or
DOE Information Bridge
http://www.doe.gov/bridge/home.html

Printed on paper containing at least 50% wastepaper, including 20% postconsumer waste

CONVERSION OF PHASE II UNSTEADY AERODYNAMICS EXPERIMENT DATA TO COMMON FORMAT

M. Maureen Hand
National Wind Technology Center
National Renewable Energy Laboratory
1617 Cole Blvd., Golden, CO 80401

Abstract

A vast amount of aerodynamic, structural, and turbine performance data were collected during three phases of the National Renewable Energy Laboratory's Unsteady Aerodynamics Experiment (UAE). To compare data from the three phases, a similar format of engineering unit data is required. The process of converting Phase II data from a previous engineering unit format to raw integer counts is discussed. The integer count files can then be input to the new post-processing software, MUNCH. The resulting Phase II engineering unit files are in a common format with current and future UAE engineering unit files. An additional objective for changing the file format was to convert the Phase II data from English units to SI units of measurement.

Nomenclature

q	Estimated dynamic pressure, from psi to Pa
ρ	Air density, from slug/ft^3 to Pa/m^3
r	Radial distance from center of hub, from in. to m
ω	Rotational speed, from rad/s to rpm
V	Wind speed, m/s
cf	Centrifugal force, from psi to Pa
C_p	Pressure coefficient
Counts	Integer value of raw data sample
slope	Calibration coefficient
offset	Calibration coefficient
α	Angle of attack corrected for upwash, deg
lfa	Local flow angle, deg
new	Converted integer count value
old	Original engineering unit value

Introduction

The National Renewable Energy Laboratory has completed Phase IV of the Unsteady Aerodynamics Experiment. A highly instrumented downwind Horizontal Axis Wind Turbine has provided a wealth of aerodynamic, structural, and turbine performance data to researchers studying issues such as the effect of unsteady aerodynamic events on wind turbine performance. Large quantities of data were collected for two rotor configurations: untwisted blades (Phase II) and twisted blades (Phase III and Phase IV). These three phases of testing produced two different engineering unit formats. The Phase II, untwisted-blade data were stored on optical disks in engineering unit form. A total of 239 channels were stored, 190 of which were measured data, 5 were time channels, and the rest were derived channels added during the post-processing process. The post-processing procedure converted integer counts produced by the data acquisition system to engineering units. In addition, the procedure corrected pressure measurements for centrifugal force effects, normalized blade surface pressures and upwash corrected angle of attack flag measurements. The low-speed shaft azimuth angle data were smoothed to account for sensor limitations, and a number of errors were introduced during this process (Miller et al., 1994).

When the Phase III data were collected, a new method of post-processing the data was implemented. This program, MUNCH, is easily adaptable and has been used in subsequent processing of data collected from the unsteady aerodynamics experiment. To provide consistency among all the unsteady aerodynamics experiment data, the Phase II engineering unit files were converted to integer count data similar to that originally produced with the instrumentation. This was accomplished by reversing the operations performed in producing the engineering unit files. These integer count data files were then input to MUNCH. This produced output engineering unit files that are now in a consistent format with the Phase III and Phase IV engineering unit files.

Procedure

Originally, the pressure tap measurements were stored as normalized pressure coefficients. The raw integer count value was converted to engineering units using the slope and offset calibration coefficients. A centrifugal force correction based on the radial location of the tap was added to the engineering unit value. This value was then normalized with the estimated dynamic pressure. These formulae follow:

$$q = \frac{1}{2} \rho \left[(r \omega)^2 + V^2 \right] \tag{1}$$

$$cf = \frac{1}{2} \rho (r \omega)^2 \tag{2}$$

$$C_p = \frac{Counts * slope + offset + cf}{q} \tag{3}$$

The wind speed and air density values represent averages over the 5-min. data campaign. The rotational frequency was assumed to be constant at 72 rpm or 7.54 rad/s. The conversion process simply reversed this procedure by solving for *Counts*. This was done in two steps. First, the dynamic pressure and centrifugal force corrections were removed. Next, slope and offset effects were eliminated in a loop for all channels at once.

The total pressure probe channels also had undergone the centrifugal force correction. This centrifugal force amount was subtracted from the dynamic pressures stored in the original data files prior to the elimination of slope and offset effects.

The upwash correction applied to the local flow angle measured by the flags follows:

$$\alpha = (-5.427E-5) * lfa^3 + (6.713E-3) * lfa^2 + 0.617 * lfa - 0.8293 \tag{4}$$

This equation resulted from curve-fitting data measured in a wind tunnel (Butterfield et al., 1992). The known angle of attack measured in the wind tunnel was plotted versus the local flow angle measured by the flag. The third order curve fit is shown above. When this data was plotted on opposite axes, a curve fit provided the following equation that was used to remove the upwash correction from the angle of attack flag data. The local flow angles that resulted from this equation were converted to integer count values through reversal of the slope and offset operations.

$$lfa = (2.5542E-4)*\alpha^3 - .021454*\alpha^2 + 1.6692*\alpha + .87154 \tag{5}$$

The time channels were copied directly from one format to the other using a slope of 1 and an offset of 0 because they were already in integer format (except for the milliseconds channel). The milliseconds channel was in decimal format and required a slope of 1000 ms/4095 counts to be converted to an integer. The low-speed shaft azimuth angle channel was treated similarly, with a slope of 360°/4095 counts and an offset of 0. The errors introduced in the azimuth angle data during the original processing remained after the conversion.

The conversion program, included in Appendix A, first reads the header file associated with the original data. This file contains the slope and offset values used to convert counts to engineering units. It also contains the average values for barometric pressure and air temperature used to determine air density and the average wind speed necessary to calculate the estimated dynamic pressure. Next, the program reads the first record containing all 239 channels into memory. The program loops through each channel individually to determine if a correction is to be removed. If the channel represents a pressure tap, the centrifugal force and dynamic pressure corrections are removed. If the channel represents a total pressure probe, the centrifugal force correction is removed. If the channel represents an angle of attack flag, the upwash correction is removed. Once all corrections are removed, each channel is then converted to a 2-byte integer as follows:

$$new = \text{int}\left(\frac{old - offset}{slope}\right) \tag{6}$$

The new record is written to output. This is done for each record recorded during the 5-min. campaign, resulting in more than 150,000 records. This new data file is stored as raw data on compact disk using the original tape filename with a '.dat' extension.

A new header file was also produced as output. This file contains the slope and offset values for each channel. All of the slopes and offsets for the pressure channels, total pressures, barometric pressure, and absolute pressure were converted to Pascals from psi or mb for consistency with the Phase III data format. The following conversions were used: 6894.8 Pa/psi, 100 Pa/mb. The blade number order was changed to be consistent with subsequent phases of the experiment. Looking upwind, the instrumented blade is blade 3, followed by blade 1 and blade 2 in the clockwise direction.

These two output files were then used as input to the MUNCH program which produces engineering unit files. A comparison was made between the MUNCH-processed engineering unit file and the original engineering unit file on a sample-by-sample basis for each channel. If, for a given sample, the difference between the new and old values was outside the range ± 0.00024**old*, a counting variable was incremented. (The 12-bit resolution of the PCM streams indicates that a difference of 1 bit in 4095 will introduce an error of ± 0.00024. This range is applicable for all channels except pressure-related measurements, angle of attack flags [±0.1 degrees--Butterfield et al., 1992], azimuth angle [±0.088 degrees or 360/4095], and milliseconds [±0.244 or 1000/4095]). The maximum difference was also recorded. This procedure was performed on several campaigns, and an example is included in Appendix B.

The channels compare favorably, with little or no difference between the engineering units processed from the converted files and the engineering units created with the original processing procedures. The pressure measurements cannot be compared because they were normalized differently. The angle of attack measurements are within 0.028 degrees of each other, which is within the resolution of the flag at 0.1 degrees. The local meteorological measurements were converted exactly from one format to the other except for the wind elevation measurements. Because the maximum elevation angle difference introduced is less than 0.0008 degrees, this difference was considered negligible. The azimuth angle channel has a resolution of 0.87, which is greater than the maximum difference recorded. The strain gage measurement differences exceed the limit at values very close to zero. This does not happen frequently, and is negligible when comparing fractions of Newton·meters. The maximum difference between the millisecond measurements is less than the resolution of .244 ms.

Another comparison was made between the 5-min. mean, maximum, and minimum values that were stored in the original header files and those stored in the newly created header files. An example of this comparison is included in Appendix C. The total pressure channels are very close on average, so the differences observed on a sample by sample basis were attributed to fluctuations in rotor speed and air density. The original centrifugal force correction originally was based on a constant rpm and 5-min. average air density. The current processing uses an instantaneous rpm and air density, which results in different centrifugal force corrections.

The normalized pressures were used in MUNCH to determine normal and tangential force coefficients at each span location. The baseline data were established by using methods discussed in Simms et al. 1996, and were plotted against angle of attack. This resulted in a plot consistent with that produced before the conversion process and provides confidence in the conversion of the normalized pressure coefficients.

Summary

The vast amount of data collected from Phases II, III, and IV of NREL's Unsteady Aerodynamics experiment is stored on compact discs in a common engineering unit format. The Phase II data were successfully converted to integer counts and reprocessed with the current software, MUNCH. Thus, data comparisons can easily be made using software common to all phases of the experiment.

References

Butterfield, C.P., W.P. Musial, and D.A. Simms (1992): "Combined Experiment Phase I Final Report", NREL/TP-257-4655, National Renewable Energy Laboratory, Golden, CO.

Miller, M.S., D.E. Shipley, M.C. Robinson, M.W. Luttges (1996): "Determination of Data Reliability for Phase II of the Combined Experiment", NREL/TP-442-6914, National Renewable Energy Laboratory, Golden, CO.

Simms, D.A., M.C. Robinson, M.M. Hand, L.J. Fingersh (1996): "A Comparison of Baseline Aerodynamic Performance of Optimally-Twisted Versus Non-Twisted HAWT Blades", Proceedings of 15th ASME Wind Energy Symposium, Houston, TX.

Appendix A

FORTRAN Code for Conversion Program

```
program convert

***** THIS PROGRAM CONVERTS THE ENGINEERING UNIT FILES PRODUCED          *****
***** DURING PHASE II TESTING BACK TO COUNTS IN ORDER TO PROCESS         *****
***** THE DATA IN THE SAME FASHION AS PHASE III TESTING.                 *****
***** WRITTEN BY: MAUREEN HAND                                           *****
***** LIST OF VARIABLES                                                  *****
*****                                                                    *****
***** adens       air density average over 5 min. data set              *****
***** aoa(4)      PCM numbers of angle of attack flags                   *****
***** atemp       air temperature average over 5 min. data set          *****
***** AZ          index of azimuth angle channel                         *****
***** baro        Barometric pressure average over 5 min. data set       *****
***** cardmask()  sets high end bits to indicate card change             *****
***** chan()      list of channel descriptions required for new header   *****
***** chanid()    list of 8 character channel id codes required in new header**
***** cforce      centrifugal force                                      *****
***** count
***** disk        current disk being processed                           *****
***** disks       names of disks to be processed                         *****
***** diskstr     name of file containing list of disks to be processed  *****
***** header      name of header file                                    *****
***** IB          counter used to read binary data stored in unix format *****
***** infil       name of input file                                     *****
***** instr       required when reading records stored in unix format    *****
***** ITB
***** newrec()    record of integer count values to be saved as output   *****
***** NUMCHAN     Number of channels to convert-190 measured,5 time      *****
***** NUMPRESS    Number of channels representing pressure taps          *****
***** numdisks    number of disks to be processed                        *****
***** NPCM()      New number saved to header file to identify card change *****
***** ncount
***** nskip       Number of original channels to skip in conversion process****
***** offset()    Array containing offset listed in header file          *****
***** OMEGA       rotational speed (72 r.p.m.)                           *****
***** outfil      name of output file                                    *****
***** pchan()     list of PCM numbers associated with pressure taps      *****
***** PCM()       list of PCM numbers in order                           *****
***** q           dynamic pressure used for normalization                *****
***** radius()    Array containing radii (in inches) associated with each *****
*****             pressure tap by PCM number                             *****
***** record()    One time step containing all channels from old E.U. file*****
***** skip()      PCM numbers of channels to be skipped in conversion process**
***** slope()     Array containing slope listed in header file           *****
***** tempstr     temporary string for user input                        *****
***** tprad(4)    radius in inches to each total pressure probe          *****
***** tpress(4)   PCM numbers of total pressure probes                   *****
***** windsp      wind speed average over 5 min. data set                *****
***** ZERO        used to initialize data file with zeros                *****
********************************************************************************


      parameter (NUMCHAN=195)
      parameter(NUMPRESS=121)
      parameter(nskip=11)

      real slope(NUMCHAN),offset(NUMCHAN),record(239),radius(NUMPRESS)
      real tprad(4)
      double precision OMEGA,cforce,baro,atemp,windsp,adens,q
      integer PCM(NUMCHAN),pchan(NUMPRESS),aoa(4),tpress(4)
      integer NPCM(NUMCHAN),numdisks,disk,AZ
      integer IB,test(20),h,i,j,k,m,n,SKIP(25),ncount,ITB,count
      integer*2 newrec(NUMCHAN)
      integer*2 cardmask(NUMCHAN) /62*16#0000,
     2                             58*16#4000,
     2                             59*16#8000,
     2                             16*16#C000/
      integer*2 tbit(0:1) /16#0000,16#2000/
      integer*4 ZERO /0/
      character*50 infil,outfil,testfil,header,diskstr,tempstr
      character chanid(NUMCHAN)*10,disks(1000)*20
      character*100 instr(239)*4,chan(NUMCHAN)

      equivalence (instr,record)

      data (aoa(j),j=1,4)/935,936,937,938/
```

```
      data (tpress(j),j=1,4)/934,458,832,862/
      data (pchan(j),j=1,NUMPRESS)/901,902,903,904,905,906,907,908,909,
     2    910,911,912,913,914,915,916,917,918,919,920,921,922,923,924,
     2    925,926,927,928,929,930,931,932,933,427,428,429,430,431,432,
     2    433,434,435,436,437,438,439,440,441,442,443,444,445,446,447,
     2    448,449,450,451,452,453,454,455,456,457,804,805,806,807,808,
     2    809,810,811,812,813,814,815,816,817,818,819,820,821,822,823,
     2    824,825,826,827,828,829,830,831,833,834,835,836,837,838,839,
     2    840,841,842,843,844,845,846,847,848,849,850,851,852,853,854,
     2    855,856,857,858,859,860,861/
      data (radius(j),j=1,NUMPRESS)/103.4,103.4,125.3,125.3,125.3,125.3,
     2    125.3,125.3,125.3,125.3,125.3,125.3,125.3,125.3,125.3,125.3,
     2    125.3,125.3,125.3,125.3,125.3,125.3,125.3,125.3,125.3,125.3,
     2    125.3,125.3,125.3,125.3,114.4,136.4,114.4,158.4,158.4,158.4,
     2    158.4,158.4,158.4,158.4,158.4,158.4,158.4,158.4,158.4,158.4,
     2    158.4,158.4,158.4,158.4,158.4,158.4,158.4,158.4,158.4,158.4,
     2    158.4,158.4,158.4,158.4,158.4,147.3,147.3,136.4,59.4,59.4,
     2    59.4,59.4,59.4,59.4,59.4,59.4,59.4,59.4,59.4,59.4,59.4,59.4,
     2    59.4,59.4,59.4,59.4,59.4,59.4,59.4,59.4,59.4,59.4,59.4,70.5,
     2    70.5,81.4,92.3,92.3,92.3,92.3,92.3,92.3,92.3,92.3,92.3,92.3,
     2    92.3,92.3,92.3,92.3,92.3,92.3,92.3,92.3,92.3,92.3,92.3,92.3,
     2    92.3,92.3,92.3,92.3,92.3,92.3,81.4/
      data (tprad(j),j=1,4)/133.3,170.9,67.4,100.3/

***** USE THIS STATEMENT WHEN PROCESSING TAPES 651-703    *****
      data(SKIP(j),j=1,nskip)/310,311,409,108,314,315,316,415,417,
     2    107,937/

***** USE THIS STATEMENT WHEN PROCESSING TAPES 704-731    *****
c      data(SKIP(j),j=1,nskip)/310,311,409,108,415,937,314,315,316/

***** USE THIS STATEMENT WHEN PROCESSING TAPES 732-752    *****
c      data(SKIP(j),j=1,nskip)/310,311,409,108,937/

      AZ=57

      chan(1)="Pressure#11, 52% span, 36% upper"",""Cp"
      chan(2)="Pressure#18, 52% span, 4% upper"",""Cp"
      chan(3)="Pressure#1, 63% span, trailing"",""Cp"
      chan(4)="Pressure#2, 63% span, 92% upper"",""Cp"
      chan(5)="Pressure#4, 63% span, 80% upper"",""Cp"
      chan(6)="Pressure#6, 63% span, 68% upper"",""Cp"
      chan(7)="Pressure#8, 63% span, 56% upper"",""Cp"
      chan(8)="Pressure#10, 63% span, 44% upper"",""Cp"
      chan(9)="Pressure#11, 63% span, 36% upper"",""Cp"
      chan(10)="Pressure#12, 63% span, 28% upper"",""Cp"
      chan(11)="Pressure#13, 63% span, 20% upper"",""Cp"
      chan(12)="Pressure#14, 63% span, 14% upper"",""Cp"
      chan(13)="Pressure#15, 63% span, 10% upper"",""Cp"
      chan(14)="Pressure#16, 63% span, 8% upper"",""Cp"
      chan(15)="Pressure#17, 63% span, 6% upper"",""Cp"
      chan(16)="Pressure#18, 63% span, 4% upper"",""Cp"
      chan(17)="Pressure#19, 63% span, 2% upper"",""Cp"
      chan(18)="Pressure#20, 63% span, 1% upper"",""Cp"
      chan(19)="Pressure#21, 63% span, 0.5% upper"",""Cp"
      chan(20)="Pressure#22, 63% span, 0% leading"",""Cp"
      chan(21)="Pressure#23, 63% span, 0.5% lower"",""Cp"
      chan(22)="Pressure#24, 63% span, 1% lower"",""Cp"
      chan(23)="Pressure#25, 63% span, 2% lower"",""Cp"
      chan(24)="Pressure#26, 63% span, 4% lower"",""Cp"
      chan(25)="Pressure#28, 63% span, 8% lower"",""Cp"
      chan(26)="Pressure#30, 63% span, 14% lower"",""Cp"
      chan(27)="Pressure#32, 63% span, 28% lower"",""Cp"
      chan(28)="Pressure#34, 63% span, 44% lower"",""Cp"
      chan(29)="Pressure#36, 63% span, 68% lower"",""Cp"
      chan(30)="Pressure#38, 63% span, 92% lower"",""Cp"
      chan(31)="Pressure#18, 58% span, 4% upper"",""Cp"
      chan(32)="Pressure#18, 69% span, 4% upper"",""Cp"
      chan(33)="Pressure#11, 58% span, 36% upper"",""Cp"
      chan(34)="Total Pressure Probe, 67% span"",""Pascal"
      chan(35)="Digital 80% LFA"",""deg"
      chan(36)="Digital 63% LFA"",""deg"
      chan(37)="Digital 47% LFA"",""deg"
      chan(38)="Digital 30% LFA"",""deg"
      chan(39)="VPA Prop Vane Speed WS-1 (12:00)"",""mps"
      chan(40)="VPA Prop Vane Speed WS-2 ( 1:30)"",""mps"
```

```
chan(41)="VPA Prop Vane Speed WS-3 ( 3:00)"","mps"
chan(42)="VPA Prop Vane Speed WS-4 ( 4:30)"","mps"
chan(43)="VPA Prop Vane Speed WS-5 ( 6:00)"","mps"
chan(44)="VPA Prop Vane Speed WS-6 ( 7:30)"","mps"
chan(45)="VPA Prop Vane Speed WS-7 ( 9:00)"","mps"
chan(46)="VPA Prop Vane Speed WS-8 (10:30)"","mps"
chan(47)="VPA Prop Vane Speed WS-9 Hub Height"","mps"
chan(48)="VPA Prop Vane Direction WD-9 Hub Height"","deg"
chan(49)="VPA Bi-Vane Speed WS-12 (3:00 @100%)"","mps"
chan(50)="VPA Bi-Vane Direction WD-12 (3:00 @100%)"","deg"
chan(51)="VPA Bi-Vane Elevation WE-12 (3:00 @100%)"","deg"
chan(52)="VPA Bi-Vane Speed WS-13 (9:00 @100%)"","mps"
chan(53)="VPA Bi-Vane Direction WD-13 (9:00 @100%)"","deg"
chan(54)="VPA Bi-Vane Elevation WE-13 (9:00 @100%)"","deg"
chan(55)="VPA Prop Vane Speed WS-10 (12:00 @40%)"","mps"
chan(56)="VPA Prop Vane Speed WS-11 (6:00 @40%)"","mps"
chan(57)="Low Speed Shaft Azimuth Angle"","deg"
chan(58)="Yaw Moment"","N-m"
chan(59)="Tower Bending about East-West Axis (X)"","N-m"
chan(60)="Tower Bending about North-South Axis (Y)"","N-m"
chan(61)="Yaw Angle"","deg"
chan(62)="Generator Power"","kW"
chan(63)="TSI (South) X-Film X (Not Working)"","Volt"
chan(64)="TSI (South) X-Film Y (Not Working)"","Volt"
chan(65)="Sonic Anemometer Channel A "","mps"
chan(66)="Sonic Anemometer Channel B "","mps"
chan(67)="Sonic Anemometer Channel C "","mps"
chan(68)="Strain Blade 3A, root flap bending"","N-m"
chan(69)="Strain Blade 3B, root flap bending"","N-m"
chan(70)="Strain Blade 1, root flap bending"","N-m"
chan(71)="Strain Blade 2, root flap bending"","N-m"
chan(72)="Strain Blade 3, 20% flap bending"","N-m"
chan(73)="Strain Blade 3, 40% flap bending"","N-m"
chan(74)="Strain Blade 3, 50% flap bending"","N-m"
chan(75)="Strain Blade 3, 70% flap bending"","N-m"
chan(76)="Strain Blade 3, 90% flap bending"","N-m"
chan(77)="Strain Blade 3, root edge bending"","N-m"
chan(78)="Strain Blade 3, 20% edge bending"","N-m"
chan(79)="Strain Blade 3, 50% edge bending"","N-m"
chan(80)="70% blade torque"","N-m"
chan(81)="Root torque (link)"","N-m"
chan(82)="Blade 3 Torsion at 50% Span"","N-m"
chan(83)="Strain X-X LSS bending"","N-m"
chan(84)="Strain Y-Y LSS bending"","N-m"
chan(85)="Strain LSS torque A"","N-m"
chan(86)="Strain LSS torque B"","N-m"
chan(87)="Pressure#1, 80% span, trailing"","Cp"
chan(88)="Pressure#2, 80% span, 92% upper"","Cp"
chan(89)="Pressure#4, 80% span, 80% upper"","Cp"
chan(90)="Pressure#6, 80% span, 68% upper"","Cp"
chan(91)="Pressure#8, 80% span, 56% upper"","Cp"
chan(92)="Pressure#10, 80% span, 44% upper"","Cp"
chan(93)="Pressure#11, 80% span, 36% upper"","Cp"
chan(94)="Pressure#12, 80% span, 28% upper"","Cp"
chan(95)="Pressure#13, 80% span, 20% upper"","Cp"
chan(96)="Pressure#14, 80% span, 14% upper"","Cp"
chan(97)="Pressure#15, 80% span, 10% upper"","Cp"
chan(98)="Pressure#16, 80% span, 8% upper"","Cp"
chan(99)="Pressure#17, 80% span, 6% upper"","Cp"
chan(100)="Pressure#18, 80% span, 4% upper"","Cp"
chan(101)="Pressure#19, 80% span, 2% upper"","Cp"
chan(102)="Pressure#20, 80% span, 1% upper"","Cp"
chan(103)="Pressure#21, 80% span, 0.5% upper"","Cp"
chan(104)="Pressure#22, 80% span, 0% leading"","Cp"
chan(105)="Pressure#23, 80% span, 0.5% lower"","Cp"
chan(106)="Pressure#24, 80% span, 1% lower"","Cp"
chan(107)="Pressure#25, 80% span, 2% lower"","Cp"
chan(108)="Pressure#26, 80% span, 4% lower"","Cp"
chan(109)="Pressure#28, 80% span, 8% lower"","Cp"
chan(110)="Pressure#30, 80% span, 14% lower"","Cp"
chan(111)="Pressure#32, 80% span, 28% lower"","Cp"
chan(112)="Pressure#34, 80% span, 44% lower"","Cp"
chan(113)="Pressure#36, 80% span, 68% lower"","Cp"
chan(114)="Pressure#38, 80% span, 92% lower"","Cp"
chan(115)="Pressure#18, 74% span, 4% upper"","Cp"
chan(116)="Pressure#11, 74% span, 36% upper"","Cp"
```

```
chan(117)="Pressure#11, 69% span, 36% upper"",""Cp"
chan(118)="Total pressure probe, 86% span"",""Pascal"
chan(119)="Absolute reference pressure"",""Pa"
chan(120)="Blade 3 Pitch angle"",""deg"
chan(121)="Pressure#1, 30% span, trailing"",""Cp"
chan(122)="Pressure#4, 30% span, 80% upper"",""Cp"
chan(123)="Pressure#6, 30% span, 68% upper"",""Cp"
chan(124)="Pressure#8, 30% span, 56% upper"",""Cp"
chan(125)="Pressure#10, 30% span, 44% upper"",""Cp"
chan(126)="Pressure#11, 30% span, 36% upper"",""Cp"
chan(127)="Pressure#13, 30% span, 20% upper"",""Cp"
chan(128)="Pressure#14, 30% span, 14% upper"",""Cp"
chan(129)="Pressure#15, 30% span, 10% upper"",""Cp"
chan(130)="Pressure#16, 30% span, 8% upper"",""Cp"
chan(131)="Pressure#17, 30% span, 6% upper"",""Cp"
chan(132)="Pressure#18, 30% span, 4% upper"",""Cp"
chan(133)="Pressure#19, 30% span, 2% upper"",""Cp"
chan(134)="Pressure#21, 30% span, 0.5% upper"",""Cp"
chan(135)="Pressure#22, 30% span, 0% leading"",""Cp"
chan(136)="Pressure#23, 30% span, 0.5% lower"",""Cp"
chan(137)="Pressure#24, 30% span, 1% lower"",""Cp"
chan(138)="Pressure#25, 30% span, 2% lower"",""Cp"
chan(139)="Pressure#26, 30% span, 4% lower"",""Cp"
chan(140)="Pressure#28, 30% span, 8% lower"",""Cp"
chan(141)="Pressure#30, 30% span, 14% lower"",""Cp"
chan(142)="Pressure#31, 30% span, 20% lower"",""Cp"
chan(143)="Pressure#34, 30% span, 44% lower"",""Cp"
chan(144)="Pressure#36, 30% span, 68% lower"",""Cp"
chan(145)="Pressure#38, 30% span, 92% lower"",""Cp"
chan(146)="Pressure#18, 36% span, 4% upper"",""Cp"
chan(147)="Pressure#11, 36% span, 36% upper"",""Cp"
chan(148)="Pressure#11, 41% span, 36% upper"",""Cp"
chan(149)="Total pressure probe, 34% span"",""Pascal"
chan(150)="Pressure#1, 47% span, trailing"",""Cp"
chan(151)="Pressure#2, 47% span, 92% upper"",""Cp"
chan(152)="Pressure#4, 47% span, 80% upper"",""Cp"
chan(153)="Pressure#6, 47% span, 68% upper"",""Cp"
chan(154)="Pressure#8, 47% span, 56% upper"",""Cp"
chan(155)="Pressure#10, 47% span, 44% upper"",""Cp"
chan(156)="Pressure#11, 47% span, 36% upper"",""Cp"
chan(157)="Pressure#12, 47% span, 28% upper"",""Cp"
chan(158)="Pressure#13, 47% span, 20% upper"",""Cp"
chan(159)="Pressure#14, 47% span, 14% upper"",""Cp"
chan(160)="Pressure#15, 47% span, 10% upper"",""Cp"
chan(161)="Pressure#16, 47% span, 8% upper"",""Cp"
chan(162)="Pressure#17, 47% span, 6% upper"",""Cp"
chan(163)="Pressure#18, 47% span, 4% upper"",""Cp"
chan(164)="Pressure#19, 47% span, 2% upper"",""Cp"
chan(165)="Pressure#20, 47% span, 1% upper"",""Cp"
chan(166)="Pressure#21, 47% span, 0.5% upper"",""Cp"
chan(167)="Pressure#22, 47% span, 0% leading"",""Cp"
chan(168)="Pressure#23, 47% span, 0.5% lower"",""Cp"
chan(169)="Pressure#25, 47% span, 2% lower"",""Cp"
chan(170)="Pressure#26, 47% span, 4% lower"",""Cp"
chan(171)="Pressure#27, 47% span, 6% lower"",""Cp"
chan(172)="Pressure#28, 47% span, 8% lower"",""Cp"
chan(173)="Pressure#30, 47% span, 14% lower"",""Cp"
chan(174)="Pressure#32, 47% span, 28% lower"",""Cp"
chan(175)="Pressure#34, 47% span, 44% lower"",""Cp"
chan(176)="Pressure#36, 47% span, 68% lower"",""Cp"
chan(177)="Pressure#38, 47% span, 92% lower"",""Cp"
chan(178)="Pressure#18, 41% span, 4% upper"",""Cp"
chan(179)="Total pressure probe, 51% span"",""Pascal"
chan(180)="North met WD 5 m"",""deg"
chan(181)="North met WS 5 m"",""mps"
chan(182)="North met WD 10 m"",""deg"
chan(183)="North met WS 10 m (Not working)"",""mps"
chan(184)="North met WD 20 m"",""deg"
chan(185)="North met WS 20 m"",""mps"
chan(186)="North met WD 50 m"",""deg"
chan(187)="North met WS 50 m"",""mps"
chan(188)="North met Temperature 5 m"",""degC"
chan(189)="North met Delta Temperature T50-T05"",""degC"
chan(190)="Barometric Pressure"",""Pascal"
chan(191)="Time-day"",""DAY"
chan(192)="Time-hour"",""HOUR"
```

```
      chan(193)="Time-minute"",""MINUTE"
      chan(194)="Time-second"",""SECOND"
      chan(195)="Time-millisecond"",""MSEC"


*****   PROMPT THE USER FOR THE NAME OF THE INPUT FILE NAME   *****
***** IF THE FILE EXTENSION IS '.lst', A LIST OF DISKS IS ASSUMED *****
       print*
60     print*,'Enter the input file name (no extension) [',
     2 infil(1:((len_trim(infil))-4)),']'
       write(6,4)
4     format(' Hit ENTER to accept the default:  ',$)
       read'(a50)',tempstr
       if (len_trim(tempstr) .gt. 0)then
          if (tempstr((len_trim(tempstr)-3):len_trim(tempstr))
     2     .eq.'.lst')then
             diskstr=tempstr(1:len_trim(tempstr))
             open (unit=13, file=diskstr, iostat=inerr, status='old')
             if (inerr .ne. 0) then
               print*,'File ',tempstr(1:len_trim(tempstr)),
     2           ' does not exist.  Try again.'
               goto 60
             endif
             m=1
76           read(13,'(a)',end=77)disks(m)
       print*,disks(m)
             m=m+1
             goto 76
77           numdisks=m-1
       print*,numdisks
             close(13)
             infil = disks(1)(1:len_trim(disks(1)))//'.dat'
             header = infil(1:((len_trim(infil))-4))//'.hdr'

*****       SET THE NEXT DISK TO BE PROCESSED AFTER THE CURRENT ONE IS FINISHED *****
             disk=2
          else
*****       PROCESS ONLY ONE DISK *****
             infil = tempstr(1:len_trim(tempstr))//'.dat'
             header = infil(1:((len_trim(infil))-4))//'.hdr'
             tempstr = ' '

*****       PREVENT LOOPING THROUGH DISKS *****
             disk=0
          endif
       endif
       continue


***** USE THIS BRANCH WHEN PROCESSING TAPES 651-703  *****
79    if (infil(9:11).eq.'d66' .or. infil(9:11).eq.'d68')then
         numskip=nskip+1
         SKIP(nskip+1)=105
         print*,'Converting',NUMCHAN-numskip,'channels.'
         print*,'Skipping 10mWD, 5mWD'

***** USE THIS BRANCH WHEN PROCESSING TAPES 704-714  *****
      elseif (infil(9:12).eq.'d704')then
         numskip=nskip+3
         SKIP(numskip-1)=107
         SKIP(numskip)=105
         SKIP(numskip-2)=417
         print*,'Converting',NUMCHAN-numskip,'channels.'
         print*,'Skipping 5mWD, 10mWD'
      elseif (infil(9:12).eq.'d711')then
         numskip=nskip+1
         SKIP(numskip)=105
         print*,'Converting',NUMCHAN-numskip,'channels.  Skipping 5mWD.'

***** USE THIS BRANCH WHEN PROCESSING TAPES 721-731  *****
      elseif (infil(9:12).eq.'d722' .or. infil(9:12).eq.'d724')then
         numskip=nskip+1
         SKIP(numskip)=105
         print*,'Converting',NUMCHAN-numskip,'channels.  Skipping 5mWD.'
      elseif (infil(9:12).eq.'d731') then
         numskip=nskip-3
```

```
         SKIP(7)=0
         SKIP(8)=0
         SKIP(9)=0
         print*,'Converting',NUMCHAN-numskip,'channels.  Adding Sonic.'


***** USE THIS BRANCH WHEN PROCESSING TAPES 732-752  *****
      elseif (infil(9:11).eq.'d73')then
         numskip=nskip+2
         SKIP(numskip-1)=415
         SKIP(numskip)=105
         print*,'Converting ',NUMCHAN-numskip,'channels. Skipping 5mWD'
         print*,'and 20% Edge'

      else
         numskip=nskip
         print*,'Converting ',NUMCHAN-numskip,'channels.'
      endif

      call readheader(slope,offset,baro,atemp,PCM,header)

*****    THE HEADER FILE FOR TAPE 75 CONTAINED AN ERROR IN THE PCM NUMBERS.   *****
*****    THE ANGLE OF ATTACK NUMBERS WERE INCORRECT, SO THIS PROGRAM ASSIGNS THE  *****
*****    CORRECT PCM NUMBERS. *****
      if (infil(9:11).eq.'d75') then
         PCM(35)=935
         PCM(36)=936
         PCM(37)=937
         PCM(38)=938
      endif

      OMEGA = 7.54
      adens = baro/(822.0*(atemp+459.6))

*****   OPEN THE INPUT FILE    *****
        open (unit=15,recl=239*4,form='unformatted',access='direct',
     2  file=infil,iostat=inerr,status='old')
        if (inerr .ne. 0) then
          print*,infil(1:len_trim(infil)),' does not exist.  Try again.'
          goto 60
        endif

***** PROMPT THE USER FOR THE NAME OF THE OUTPUT FILE NAME *****
c80    print*
c       print*,'Enter the output file name (no extension).'
c       read'(a50)',outfil
      outfil='d:\header\'//infil(9:len_trim(infil))
      open(unit=16,access='sequential',form='binary',
     2    file=outfil,iostat=inerr,status='unknown')
        if (inerr .ne. 0) then
          print*,infil(1:len_trim(infil)),' does not exist.  Try again.'
c         goto 80
        endif
***** WRITE THE HEADER OF 2048 BYTES EQUAL ZERO TO THE FILE. *****
      do 100,i=1,512
         write(16)ZERO
100   continue

      print*,'Reading file:  ',infil
      print*,'Writing to file:  ',outfil

        numrec=1
        count=0

***** THE ORIGINAL DATA FILES WERE STORED IN UNIX FORMAT ON OPTICAL   *****
***** DISK.  THE FOLLOWING READ STATEMENT ASSUMES IT IS READING UNIX  *****
***** FORMAT INTO DOS FORMAT                                          *****
      ITB = 0
200     read(15,rec=numrec,end=700)
     2   ((instr(m)(IB:IB),IB=4,1,-1),m=1,239)

      count=count+1
      windsp=DBLE(record(47)*3.2808)

         h=1
         j=1
```

```
         k=1
         i=1
         ncount=1
      do 400, n=1,NUMCHAN

***** CHECK TO SEE IF THE CHANNEL SHOULD BE SKIPPED       *****
          m=1
          do while (PCM(n).ne.SKIP(m) .and. m .le. numskip)
             m=m+1
          end do
          if (PCM(n).eq.SKIP(m)) then
             if (PCM(n).eq.aoa(j)) then
                j=j+1
             endif
             goto 400
          endif

***** REMOVE DYNAMIC PRESSURE NORMALIZATION AND CENTRIFUGAL FORCE     *****
***** CORRECTION FROM PRESSURE STATIONS.*****
         if (PCM(n).eq.pchan(h).and.h.le.NUMPRESS) then
            q=.5*adens*(radius(h)*omega*radius(h)*omega/144.0+
     2         windsp*windsp)/144.0
            cforce=.5*adens*radius(h)*omega*radius(h)*omega/144.0/144.0
            record(n)=record(n)*q-cforce
            h=h+1
         endif

***** REMOVE UPWASH CORRECTION FROM ANGLE OF ATTACK FLAGS *****
         if (PCM(n).eq.aoa(j).and.j.le.4) then
            record(n)=(((((4.0666E-11*record(n)+5.1949E-8)*record(n)-
     2         9.0752E-6)*record(n)+6.5012E-4)*record(n)-0.024315)*
     2         record(n)+1.5946)*record(n)+1.2825
            j=j+1
         endif

***** REMOVE CENTRIFUGAL FORCE CORRECTION FROM TOTAL PRESSURE PROBES  *****
         if (PCM(n).eq.tpress(k).and.k.le.4)then
            cforce=.5*adens*tprad(k)*omega*tprad(k)*omega/144.0/144.0
            record(n)=record(n)-cforce
            k=k+1
         endif


***** ASSIGN THE INTEGER VALUE TO THE NEW RECORD TO BE WRITTEN.  NEXT, *****
***** ADD THE HIGH END CARD AND TOGGLE BITS TO THE COUNT VALUE. *****
***** CHANGE THE TOGGLE BIT TO ITS OTHER VALUE FOR THE NEXT TIME *****
***** THROUGH *****
         newrec(ncount)=INT2(ANINT((record(n)-offset(n))/slope(n)))
         if (newrec(ncount).lt.0)then
            print*,'out of bounds',newrec(ncount),PCM(n),ncount,count
            newrec(ncount)=0
         elseif (newrec(ncount).gt.4095)then
            print*,'out of bounds',newrec(ncount),PCM(n),ncount,count
            newrec(ncount)=4095
        endif
         newrec(ncount)=IOR(newrec(ncount),cardmask(n))
         newrec(ncount)=IOR(newrec(ncount),tbit(ITB))
         ncount=ncount+1
400   continue
      write(16)(newrec(m),m=1,ncount-1)
      ITB=1-ITB


***** USE THIS STATEMENT TO LIMIT THE NUMBER OF RECORDS PROCESSED *****
c      if (numrec.eq.10)then
c         print*,'moving on to next file',ncount-1,numskip
c         goto 700
c      endif

      numrec=numrec+1

      goto 200

700   close(16)

9000  format(4f15.10)
```

```
9100  format(i10,4f15.10)
9200  format(i10,f15.10)
9300  format(i10,2f15.6)
c9300  format(i10,f15.10)
9400  format(i10,3f15.10)
9500  format(a,4i6)

***** CONVERT THE PRESSURE CHANNELS FROM PSI TO PA BY MULTIPLYING THE SLOPES AND OFFSETS
*****
***** BY THE CONVERSION FACTOR 6894.8 PA/PSI.  CONVERT THE ATMOSPHERIC CHANNELS FROM MB
*****
***** TO PA BY MULTIPLYING THE SLOPES AND OFFSETS BY THE CONVERSION FACTOR 100 PA/MB.
*****
      h=1
      j=1
      do 800, n=1,NUMCHAN
         if (PCM(n).eq.pchan(h).and.h.le.NUMPRESS) then
            slope(n)=slope(n)*6894.8
            offset(n)=offset(n)*6894.8
            h=h+1
         elseif (PCM(n).eq.tpress(j).and.j.le.4)then
            slope(n)=slope(n)*6894.8
            offset(n)=offset(n)*6894.8
            j=j+1
         elseif(PCM(n).eq.459)then
            slope(n)=slope(n)*100
            offset(n)=offset(n)*100
         elseif(PCM(n).eq.116)then
            slope(n)=slope(n)*100
            offset(n)=offset(n)*100
         endif
800   continue

***** SAVE THE NEW HEADER FILE WITH THE SAME NAME IN A DIFFERENT DIRECTORY.   *****
***** ASSUME THE OLD HEADER FILE IS ON DISK SO THAT THE FIRST 8 CHARACTERS ARE *****
***** THE DRIVE AND DIRECTORY NAME, I.E. D:\D662\XXXXXXX.XXX  *****
      header=outfil(1:(len_trim(outfil)-4))//'.hdr'
      open(unit=18,file=header)
      open(unit=19,file='d:\chanid.txt')
      do 850, k=1,NUMCHAN
         read(19,8300)NPCM(k),chanid(k)
850   continue
      close(19)

      write(18,'(i3)')NUMCHAN-numskip
      do 900, k=1,195
          j=1
          do while (PCM(k).ne.SKIP(j) .and. j .le. numskip)
             j=j+1
          end do
          if (PCM(k).eq.SKIP(j)) then
             goto 900
          endif

***** ANGULAR MEASUREMENTS ARE GIVEN 'N' (0-360) OR 'O' (-180,180)    *****
***** TYPE DECLARATIONS FOR USE IN MUNCH.  ALL OTHERS ARE 'M'.        *****
          if (chanid(k).eq.'VPAWD9'.or.chanid(k).eq.'VPAWD12'.or.
     2       chanid(k).eq.'VPAWD13'.or.chanid(k).eq.'B3AZI'.or.
     2       chanid(k).eq.'YAW') then
             write(18,8100)NPCM(k),char(34),
     2         chan(k)(1:len_trim(chan(k))),char(34),offset(k),slope(k)
         elseif(chanid(k).eq.'NMWD5M'.or.chanid(k).eq.'NMWD10M'.or.
     2          chanid(k).eq.'NMWD20M'.or.chanid(k).eq.'NMWD50M') then
             write(18,8100)NPCM(k),char(34),
     2         chan(k)(1:len_trim(chan(k))),char(34),offset(k),slope(k)
         elseif (chanid(k).eq.'B3PITCH'.or.chanid(k).eq.'30LFA' .or.
     2           chanid(k).eq.'63LFA'.or.chanid(k).eq.'80LFA') then
            write(18,8200)NPCM(k),char(34),
     2         chan(k)(1:len_trim(chan(k))),char(34),offset(k),slope(k)
         else
           write(18,8000) NPCM(k),char(34),chan(k)(1:len_trim(chan(k))),
     2        char(34),offset(k),slope(k)
         endif

900   continue
      close(18)
```

```
8000  format(i3,',',a,a,a,',2,'f15.6,',',f10.6,
     2   ',0,0,0,0,-999999.9,999999.9,0,"M",','0,0,0')
8100  format(i3,',',a,a,a,',2,'f15.6,',',f10.6,
     2   ',0,0,0,0,-999999.9,999999.9,0,"N",','0,0,0')
8200  format(i3,',',a,a,a,',2,'f15.6,',',f10.6,
     2   ',0,0,0,0,-999999.9,999999.9,0,"O",','0,0,0')
8300  format(i3,1x,a)


       if (disk .ge. 1 .and. disk .le. numdisks) then
          infil = disks(disk)(1:len_trim(disks(disk)))//'.dat'
          header = infil(1:(len_trim(infil)-4))//'.hdr'
          disk = disk + 1
          SKIP(nskip+1)=0
          SKIP(nskip+2)=0
        SKIP(nskip+3)=0
       print*,infil,header
          goto 79
       endif
      stop
      end

**********************************************************************************
**********************************************************************************
      subroutine readheader(slope,offset,baro,atemp,PCM,inname)

***** THIS SUBROUTINE READS THE HEADER FILE FROM THE OLD PHASE II DATA.*****

      parameter (NUMCHAN=195)

      real slope(NUMCHAN),offset(NUMCHAN),tempr,mean,max,min,stdev
      double precision baro,atemp
      integer PCM(NUMCHAN),PCMindex,tempi,NCHAN
      character*80 tempstr,inname

******************************************************************************
***** LIST OF VARIABLES                                                  *****
***** atemp       air temperature, C                                     *****
***** baro        barometric pressure, mb                                *****
***** inname      name of input header file                              *****
***** max         maximum value for 5 min. data set                      *****
***** mean        mean value for 5 min. data set                         *****
***** min         minimum value for 5 min. data set                      *****
***** NCHAN       number of channels in PCMindex series                  *****
***** offset()    Array containing offset listed in header file          *****
***** PCM()       list of PCM numbers in order                           *****
***** PCMindex    index of PCM number series in header file              *****
***** slope()     Array containing slope listed in header file           *****
***** stdev       standard deviation for 5 min. data set                 *****
***** tempr       temporary real variable                                *****
***** tempi       temporary integer variable                             *****
***** tempstr     temporary string                                       *****
******************************************************************************



*****   PROMPT THE USER FOR THE NAME OF THE INPUT FILE NAME   *****
c       print*
c       print*,'Enter the header file name (no extension).'
c       read'(a50)',inname
        open (unit=12, file=inname, iostat=inerr, status='old')

      do 100, n=1,30
         read(12,'(a)')tempstr
c         print*,tempstr
100   continue
      n=1
      do 150, m=1,6
         read(12,*)PCMindex
c         print*,PCMindex
         read(12,'(a)')tempstr
c         print*,tempstr
         read(12,'(a)')tempi
c         print*,tempi
         read(12,'(a)')tempstr
```

```
c         print*,tempi
         read(12,*)NCHAN
c         print*,NCHAN

         do 200, k=1,NCHAN
            read(12,*)tempi
c            print*,tempi
            read(12,*)tempi
            PCM(n)=PCMindex*100+mod(tempi,100)
c            print*,PCM(n),n
            read(12,*)tempi
c            print*,tempi
            read(12,*)slope(n)
c            print*,slope(n)
            read(12,*)offset(n)
c            print*,offset(n)
            read(12,'(a)')tempstr
c            print*,tempstr
            read(12,*)mean
c            print*,mean
            if (PCM(n).eq.116) then
               baro=mean
            elseif (PCM(n).eq.113) then
               atemp=(mean)*9.0/5.0 + 32.0
            elseif (PCM(n).eq.303)then
               offset(n)=0.0

***** THIS SLOPE =360/4095
               slope(n)=0.0879
            endif
            read(12,*)max
            read(12,*)min
            read(12,*)stdev
            do 250, j=1,13
               read(12,'(a)')tempstr
c               print*,tempstr
250         continue
            n=n+1
200      continue
150   continue
      do 300, k=191,195
         PCM(k)=700+(k-190)
         slope(k)=1.0
         offset(k)=0.0
300   continue
***** THIS SLOPE =1000/4095
      slope(195)=.24420
      print*,baro,atemp

      close(12)

8900  format(i5,4f20.10)
      return
      end
**********************************************************************************
**********************************************************************************
```

Appendix B

Comparison of Channels on a Sample by Sample Basis

Channel	New Value at Maximum	Old Value at Maximum	Difference at Maximum	Maximum % Difference	No. Times Difference Exceeded
TP67	269.8412	301.3969	31.5557	10	154124
80AOA	-0.0172	0.0104	0.0276	266	153219
63AOA	-0.0229	0.0101	0.0330	326	154438
30AOA	-0.0253	0.0024	0.0277	1139	143210
VPAWS1	0.0000	0.0000	0.0000	-9999999	0
VPAWS2	0.0000	0.0000	0.0000	-9999999	0
VPAWS3	0.0000	0.0000	0.0000	-9999999	0
VPAWS4	0.0000	0.0000	0.0000	-9999999	0
VPAWS5	0.0000	0.0000	0.0000	-9999999	0
VPAWS6	0.0000	0.0000	0.0000	-9999999	0
VPAWS7	0.0000	0.0000	0.0000	-9999999	0
VPAWS8	0.0000	0.0000	0.0000	-9999999	0
VPAWS9	0.0000	0.0000	0.0000	-9999999	0
VPAWD9	0.0000	0.0000	0.0000	-9999999	0
VPAWS12	0.0000	0.0000	0.0000	-9999999	0
VPAWD12	0.0000	0.0000	0.0000	-9999999	0
VPAWA12	0.0060	0.0068	0.0008	12	41039
VPAWS13	0.0000	0.0000	0.0000	-9999999	0
VPAWD13	0.0000	0.0000	0.0000	-9999999	0
VPAWA13	0.0013	0.0006	-0.0007	117	17399
VPAWS10	0.0000	0.0000	0.0000	-9999999	0
VPAWS11	0.0000	0.0000	0.0000	-9999999	0
B3AZI	0.0000	0.0377	0.0377	100	39657
YAWMOM	0.0000	0.0000	0.0000	-9999999	0
TBEWAX	0.0000	0.0000	0.0000	-9999999	0
TBNSAY	0.0000	0.0000	0.0000	-9999999	0
YAW	0.0000	0.0000	0.0000	-9999999	0
GENPOW	0.0083	0.0083	0.0000	0	55
B1ARFB	0.9479	0.9471	-0.0008	0	855
B1BRFB	0.2920	0.2919	-0.0001	0	196
B2RFB	0.1126	0.1128	0.0002	0	170
B3RFB	0.6064	0.6058	-0.0006	0	648
B120FB	0.0396	0.0392	-0.0003	1	273
B140FB	0.4484	0.4478	-0.0006	0	1300
B150FB	0.5232	0.5238	0.0006	0	2120
B170FB	0.1965	0.1961	-0.0003	0	2764
B190FB	0.0537	0.0528	-0.0009	2	66701
B1AREB	0.3656	0.3665	0.0009	0	381
B120EB	1.3417	1.3412	-0.0006	0	286
B150EB	1.1064	1.1057	-0.0006	0	1031
70TQ	0.0000	0.0000	0.0000	-9999999	0
RTTQ	0.0390	0.0389	-0.0001	0	2596
50TSN	0.1137	0.1133	-0.0004	0	4414
LSSXXB	1.3948	1.3965	0.0017	0	89
LSSYYB	0.8380	0.8404	0.0024	0	259
LSSTQA	2.1682	2.1690	0.0008	0	4
LSSTQB	1.5952	1.5944	-0.0008	0	6

Channel	New Value at Maximum	Old Value at Maximum	Difference at Maximum	Maximum % Difference	No. Times Difference Exceeded
TP86	500.2993	461.0866	-39.2127	9	154303
ABSRP	0.0000	0.0000	0.0000	-9999999	0
B1PITCH	0.0000	0.0000	0.0000	-9999999	0
TP34	0.1865	0.4621	0.2755	60	151541
TP51	-2.3864	2.5815	4.9679	192	155988
NMWD5M	0.0000	0.0000	0.0000	-9999999	0
NMWS5M	0.0000	0.0000	0.0000	-9999999	0
NMWD10M	0.0000	0.0000	0.0000	-9999999	0
NMWD20M	0.0000	0.0000	0.0000	-9999999	0
NMWS20M	0.0000	0.0000	0.0000	-9999999	0
NMWD50M	0.0000	0.0000	0.0000	-9999999	0
NMWS50M	0.0000	0.0000	0.0000	-9999999	0
NMT5M	0.0000	0.0000	0.0000	-9999999	0
NMDT	0.0000	0.0000	0.0000	-9999999	0
BARO	0.0000	0.0000	0.0000	-9999999	0
DAY	0.0000	0.0000	0.0000	-9999999	0
HOUR	0.0000	0.0000	0.0000	-9999999	0
MINUTE	0.0000	0.0000	0.0000	-9999999	0
SECOND	0.0000	0.0000	0.0000	-9999999	0
MILLISEC	0.2441	0.1600	-0.0841	53	39578

Appendix C

Comparison of Header Files

Channel	% Difference	Maximum Difference	Minimum Difference
TP67	1	8.7264	1.3572
80LFA	0	-0.0203	0.1625
63LFA	0	0.0194	2.5089
30LFA	0	-0.0342	1.1516
VPAWS1	0	0.0000	0.0000
VPAWS2	0	0.0000	0.0000
VPAWS3	0	0.0006	0.0003
VPAWS4	0	0.0000	0.0000
VPAWS5	0	0.0000	0.0000
VPAWS6	0	0.0000	0.0000
VPAWS7	0	0.0000	0.0000
VPAWS8	0	0.0000	0.0000
VPAWS9	0	0.0000	0.0000
VPAWD9	0	0.0000	0.0000
VPAWS12	0	0.0001	0.0000
VPAWD12	0	0.0003	0.0002
VPAWA12	0	0.0009	0.0005
VPAWS13	0	0.0000	0.0000
VPAWD13	0	-0.0012	-0.0010
VPAWA13	0	-0.0009	-0.0003
VPAWS10	0	0.0000	0.0000
VPAWS11	0	0.0000	0.0000
B3AZI	0	0.0487	0.0041
YAWMOM	0	0.0001	0.0001
TBEWAX	0	-0.0010	-0.0010
TBNSAY	0	0.0012	0.0009
YAW	0	0.0000	0.0000
GENPOW	0	0.0000	0.0000
B1ARFB	0	-0.0004	-0.0007
B1BRFB	0	0.0001	0.0001
B2RFB	0	-0.0001	0.0006
B3RFB	0	-0.0009	-0.0006
B120FB	0	0.0001	0.0000
B140FB	0	-0.0007	-0.0005
B150FB	0	0.0011	0.0003
B190FB	0	-0.0010	-0.0008
B1AREB	0	0.0012	0.0010
B150EB	0	-0.0009	-0.0006
RTTQ	0	-0.0001	-0.0001
50TSN	0	-0.0009	-0.0003
LSSXXB	0	0.0020	0.0014
LSSYYB	0	0.0034	0.0022
LSSTQA	0	0.0011	0.0008
LSSTQB	0	-0.0012	-0.0008

Channel	% Difference	Maximum Difference	Minimum Difference
TP86	1	5.5052	5.4022
ABSRP	0	0.0000	0.0001
B1PITCH	0	-0.0003	-0.0003
TP34	0	1.0040	0.6008
TP51	2	2.6836	6.6508
NMWD5M	0	0.0000	0.0000
NMWS5M	0	0.0000	0.0000
NMWD10M	0	0.0000	0.0000
NMWD20M	0	0.0000	0.0000
NMWS20M	0	-0.0002	-0.0001
NMWD50M	0	0.0000	0.0000
NMWS50M	0	0.0003	0.0000
NMT5M	0	-0.0001	-0.0001
NMDT	0	0.0002	0.0002
BARO	0	0.0000	0.0000

REPORT DOCUMENTATION PAGE

Form Approved
OMB NO. 0704-0188

1. AGENCY USE ONLY (Leave blank)

2. REPORT DATE
July 1999

3. REPORT TYPE AND DATES COVERED
Technical Report

4. TITLE AND SUBTITLE
Conversion of Phase II Unsteady Aerodynamics Experiment Data to Common Format

5. FUNDING NUMBERS
WE901110

6. AUTHOR(S)
M.M. Hand

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)
National Renewable Energy Laboratory
1617 Cole Blvd.
Golden, CO 80401-3393

8. PERFORMING ORGANIZATION REPORT NUMBER

9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)
National Renewable Energy Laboratory
1617 Cole Blvd.
Golden, CO 80401-3393

10. SPONSORING/MONITORING AGENCY REPORT NUMBER
TP-500-26371

11. SUPPLEMENTARY NOTES

12a. DISTRIBUTION/AVAILABILITY STATEMENT
National Technical Information Service
U.S. Department of Commerce
5285 Port Royal Road
Springfield, VA 22161

12b. DISTRIBUTION CODE

13. ABSTRACT *(Maximum 200 words)*
A vast amount of aerodynamic, structural, and turbine performance data were collected during three phases of the National Renewable Energy Laboratory's Unsteady Aerodynamics Experiment (UAE). To compare data from the three phases, a similar format of engineering unit data is required. The process of converting Phase II data from a previous engineering unit format to raw integer counts is discussed. The integer count files can then be input to the new post-processing software, MUNCH. The resulting Phase II engineering unit files are in a common format with current and future UAE engineering unit files. An additional objective for changing the file format was to convert the Phase II data from English units to SI units of measurement.

14. SUBJECT TERMS
wind energy, wind turbines, wind turbine aerodynamics

15. NUMBER OF PAGES

16. PRICE CODE

17. SECURITY CLASSIFICATION OF REPORT

18. SECURITY CLASSIFICATION OF THIS PAGE

19. SECURITY CLASSIFICATION OF ABSTRACT

20. LIMITATION OF ABSTRACT
UL