Wake survey analysis and positioning software
Hoskins, P.

Publisher's version / Version de l'éditeur:
https://doi.org/10.4224/8895059
Laboratory Memorandum (National Research Council of Canada. Institute for Ocean Technology); no. LM-2007-02, 2007

National Research Council Canada
Conseil national de recherches Canada

Institute for Ocean Technology
Institut des technologies océaniques

Laboratory Memorandum

LM-2007-02

Wake Survey Analysis and Positioning Software

P. Hoskins

April 2007 (orig. pub. 1995)

Canada

DOCUMENTATION PAGE

REPORT NUMBER LM-2007-02	NRC REPORT NUMBER		DATE April 16, 2007		
REPORT SECURITY CLASSIFICATION Unclassified			DISTRIBUTION Unlimited		
TITLE Wake Survey Analysis and Positioning Software Manual					
AUTHOR(S) P. Hoskins					
CORPORATE AUTHOR(S)/PERFORMING AGENCY(S) IOT					
PUBLICATION N/A					
SPONSORING AGENCY(S) IOT					
IOT PROJECT NUMBER 421009			NRC FILE NUMBER		
KEY WORDS Wake survey, software			PAGES ii, 124, App. A	FIGS. 19	TABLES nil

SUMMARY

This report is a description of the IOT wake survey analysis and positioning software as of 1995. No electronic copy of this document could be found so the paper copy was scanned into PDF format in 2007 and given an LM number so it could be tracked by CISTI.

ADDRESS

National Research Council
Institute for Ocean Technology
Arctic Avenue, P. O. Box 12093
St. John's, NL A1B 3T5
Tel.: (709) 772-5185, Fax: (709) 772-2462

Wake Survey Analysis and Positioning Software Manual

By Paul Hoskins
March 31, 1995

Institute for Marine Dynamics
National Research Council of Canada

TABLE OF CONTENTS

Document Conventions

Topics:

- ❑ Text font conventions used in this manual for VMS and MS-DOS systems

This instruction manual uses several conventions to distinguish between computer output and input. Any command typed at the command line (either DOS or VAX VMS) will presented with a plain courier font, in bold and in lower case letters (neither VMS or MS-DOS is case sensitive):

VAX VMS Commands:

```
$ dir /exclude=*.dir
$ @wake_run 23 45
```

MS-DOS Commands:

```
c:\>wake
c:\>edit wake.cfg
```

The `$` symbol represents the VMS command prompt and the `c:\>` is the MS-DOS prompt. Do not type them as part of the command. For VMS, the command prompt will have the node name attached, such as:

```
MINNIE $
```

However, the node name is not shown in the manual for simplicity. Similarly, the MS-DOS prompt is always shown as `C:\>`, the MS-DOS root directory.

When comments about commands are included, they will be in the standard font for this manual and will be preceded by an exclamation mark. DO NOT TYPE THE COMMENTS AS PART OF THE COMMAND. Example:

VAX VMS Commands:

```
$ dir /exclude=*.dir        ! List all files, but exclude directories
$ @wake_run 23 45           ! Run WAKE_RUN for runs 23 to 45
```

MS-DOS Commands:

```
c:\>wake                    ! Start the wake survey positioning software
c:\>edit wake.cfg           ! Edit the wake survey configuration file
```

Output from a command or the contents of any ASCII data files will displayed in the same font as input commands but without bold. For example:

VAX VMS Output or File Contents:

```
Contents of WHITE$DKA300:[PJ94617.WAKE_TESTS]

WAKE_RUN.COM;3        WAKE.COM;5      WAKE_RESET.COM;5
WAKE_SET.COM;3        WAKE_TEST_STATS.DAT;89
```

MS-DOS Output or File Contents:

```
****  Wake Survey Positioning System - Configuration File
****  October 16/94
Default Units                               ,mm
Number of Different Unit Systems(max 10) ,2
mm
in
```

Generally, when a program is referred to within the text it will be represented by capital letters in the plain courier font, such as `WAKE_SET.COM`.

1.0

Introduction

Topics:

- IMD's new wake survey system and its advantages
- 5-hole probe analysis theory
- Coordinate systems used in wake surveys and transforming between them

The National Research Council Institute for Marine Dynamics uses a 5 hole pitot tube probe head for measuring three dimensional flows in model wake surveys. This system was designed and implemented throughout 1994. Control and analysis software and techniques were developed and tested during 1994-95.

This wake survey system uses a five hole pitot tube head similar to those used in wind tunnels. The 5 pressures developed at the probe head can be used to determine the flow direction and magnitude. Pressures are measured by piezo-resistive pressure transducers manufactured by Endevco Corporation. The probe is traversed to locations within the flow by an x/y positioning stage mounted to the back of the model, as shown in figure 1.1. A pair of DC servo motors and an Aerotech Unidex U-14 controller set the position of the stage to a high accuracy and repeatability. A Microsoft QuickBASIC program was written to provide automated point selection and simple control over most positioning operations. An extensive series of VMS software routines was developed for handling data analysis, manipulation and storage. These programs use the standard NRC GEDAP file format and routines. A further set of routines was developed in the VMS version of Matlab for data curve fairing and data presentation.

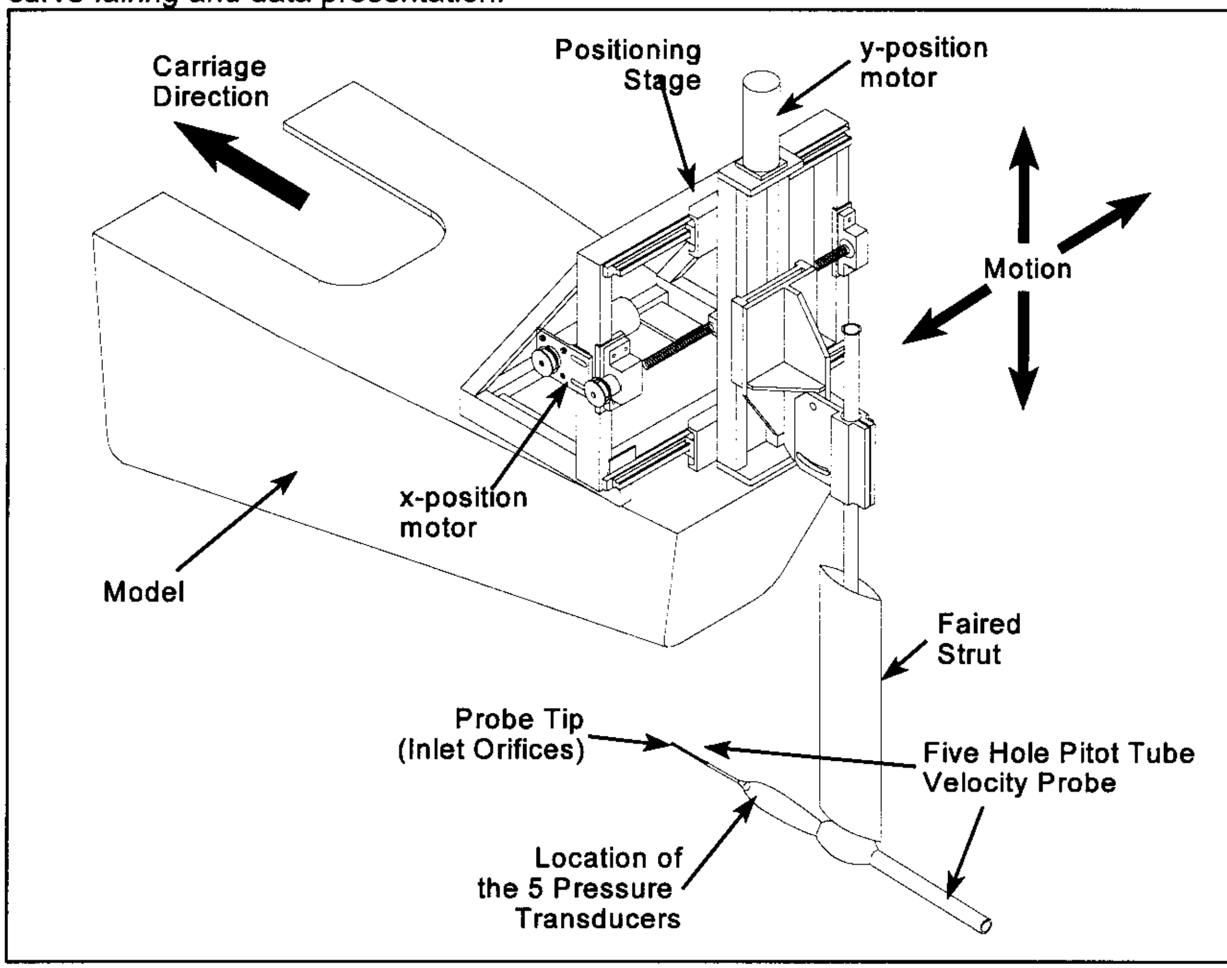


Figure 1.1 - Wake survey velocity probe and positioning stage.

The large weight of the positioning stages necessitates the need for a sinkage and pitch clamp to hold the model at any sinkage and trim combination. The standard method is to perform a series of resistance tests on the model and calculate the sinkage and trim levels that will occur at the different wake survey velocities. The model will be set at this sinkage and trim using the pitch clamp. The pitch clamp mechanism is shown in figure 1.2.

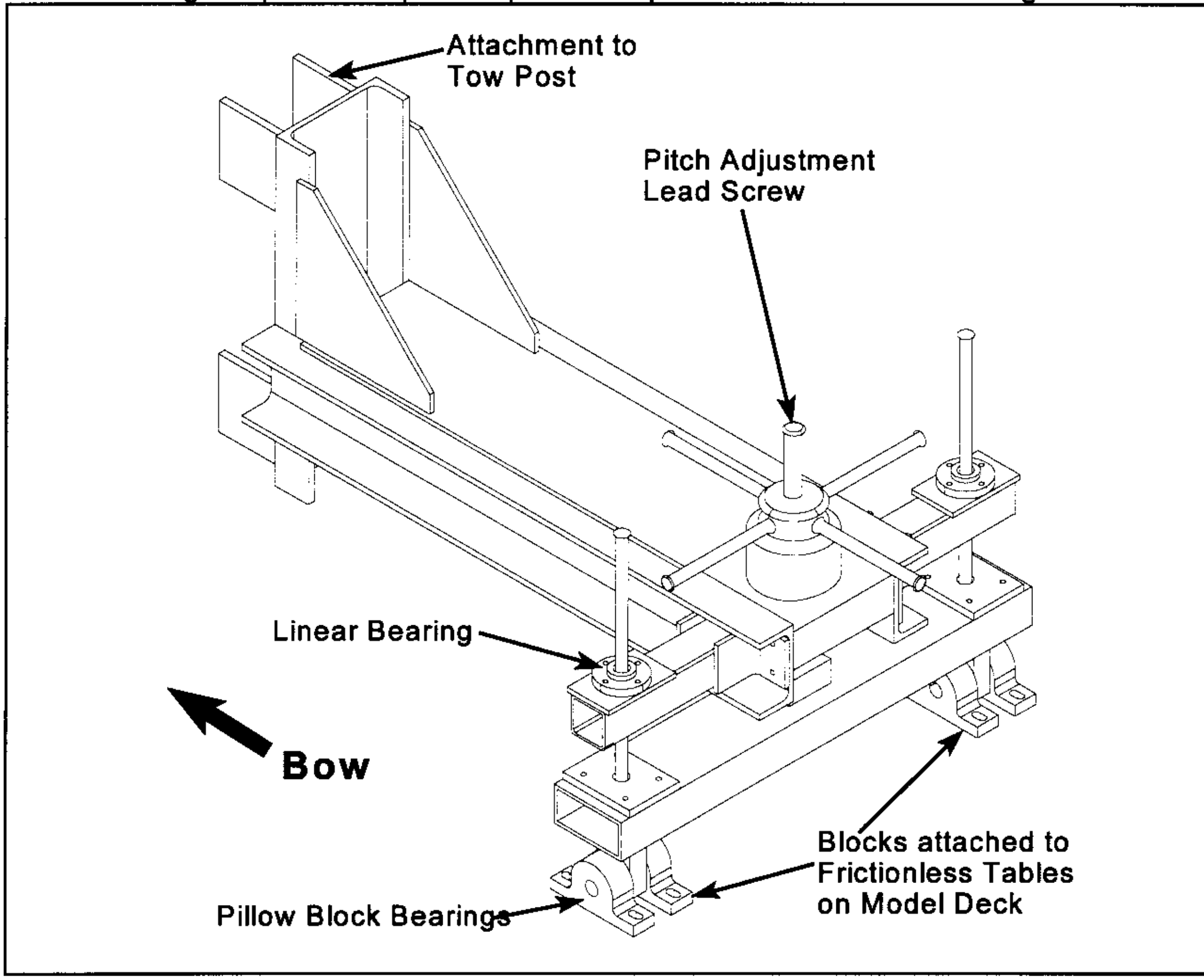


Figure 1.2 - Pitch clamp mechanism.

A new calibration system has also been developed. A previous IMD apparatus, the DOLPHIN system, has been modified to provide pitch control over the probe head. A high accuracy servo controlled motor (20,000 counts / revolution) was installed on the pitching platform to provide automated control over the yaw angle. Using this system, various combinations of yaw and pitch can be applied to the probe head. A set of VMS and Matlab software programs are used to manipulate the calibration data and prepare a set of calibration tables.

The piezo-electric pressure transducers provide near instantaneous response to pressure change. Further, temperature compensation is extremely quick. Thus, little time is required for the transducer to get an accurate reading of the pressure. This allows multiple positions in the wake to be sampled for one run down the tank, which reduces the total amount of test time required for a complete wake survey. The transducers are small enough to be placed very near the pressure inlet holes. This reduces the length of the fluid column between the start of the hole and pressure transducer, thus reducing problems with air pockets and further reducing the response time.

The pressure transducers are vented to atmosphere, so all measured pressures are gauge pressures. To check the operation of the probes, they can be moved to various water depths. The pressures measured by the transducers should be the same as the pressure due to the depth of water. This also allows the transducers to be easily calibrated and checked.

The x/y positioning stage allows for an infinite number of sample positions in a propeller wake. The probe is not limited to a fixed number of positions. Consequenctly, it can be used to investigate the flow around appendages such as struts, brackets and rudders.

The new wake survey system replaces all of the hardware and software of the previous, 5 probe differential pressure system, as described by Miles (1979). This system was replaced for many reasons. The hardware had proved difficult to use and set up. Long pressure lines had to be continually bled to remove air pockets. Further, the probe heads were fixed to rotate at a given set of radii, and in increments of 5 degrees only. This limited the selection of available sampling positions. There were also problems with very slow response times. This limited the system to sampling only one set of data points for every test run. The analysis software had also become outdated and difficult to use. Finally, the data fairing and presentation software was very difficult and tedious to use. All of the old wake survey software software was replaced by more modern routines, especially the curve fairing software and plotting programs, which were written in Matlab on a DEC Alpha VMS machine running X Windows.

1.1 Wake Survey 5 Hole Probe Analysis Theory

The wake survey numerical analysis method used at IMD is unique among 5 hole pitot tube velocity probes. This method eliminates the need to fit polynomials to the calibration data. It is based on the intersection of contour lines within the calibration tables.

The five hole probe is oriented as shown in figure 1.1. The center hole is ground perpindicular to the axis of the probe. The four outside holes are ground at 45° to the axis. The center hole is sensitive to total pressure and the four outside holes sensitive to the pitch and yaw angle of the flow with respect to the probe. The pressure difference between the top and bottom pressures is sensitive only to pitched in-flows. For example, assume a flow yawed to the positive side and pitched in the positive direction. The flow coming in from the top will increase the pressure on the top hole. It will also act to decrease the pressure on the bottom hole which will be in the shadow of the probe. Thus the pressure differenc between the two will grow larger. The two side holes will both see the same affect from the pitched inflow. Their pressure difference will essentially remain the same.

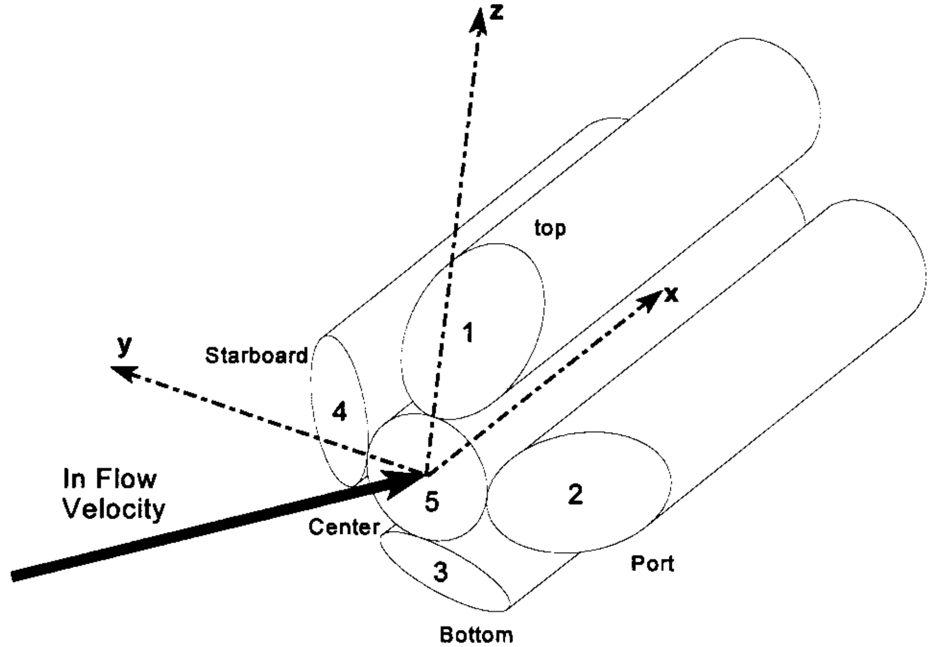


Figure 1.3 - Probe tip details and inflow orientation.

When the pressures are measured from the pitot tube they need to be non-dimensionalized by the carriage speed (or free stream velocity). This is done using the common formula:

$$C_p = \frac{P}{P_T}$$

Where C_P is a dimensionless coefficient for each probe oriface, P is the pressure in *kpa* for the orifice and P_T is the total free stream pressure away from the oriface. The Endevco pressure transducers used in IMD's probes are calibrated in *cm* of water, commonly called the water head. Therefore, they must be converted to *kpa* by multiplying the head by the specific weight of water. P then becomes $Pg\rho/100$ where g is acceleration due to gravity (*9.82 m/s*) and ρ is the density of water. The total free stream pressure P_T is a result of the carriage velocity and is found as $P_T = 1/2\rho V^2$. Therefore, the pressure coefficient C_P for each probe orifice is:

$$C_P = \frac{P}{P_T} = \frac{\rho g P/100}{1/2 \rho V^2}$$

$$C_P = \frac{gP}{50 V^2}$$

This coeffecient is computed for all five probe orifice pressures. To determine the yaw and pitch angles of the in-flow, differences in the pressure coeffiecient accross orifices 1 and 3 and across 2 and 4 can be used. These differences are designated *Q* and *R* and are defined as follows:

$$Q = C_{P2} - C_{P4}$$

$$R = C_{P3} - C_{P1}$$

Q will be sensitive to the yaw angle of the inflow and *R* will be sensitive to pitch. For calibration of the actual probe head, the probe is yawed and pitch at 5° intervals through a $\pm 40^\circ$ range of angles. By calculating *Q* and *R* for each combination of yaw and pitch, 3 dimensional calibration tables are built up. Surface plots of *Q* and *R* are shown at the end of this section. The x-axis is the yaw angle, the y axis is the pitch angle and the z axis is the *Q* or *R* parameter (which dimensionless). As can be seen from the graphs *Q* changes with yaw angle, but changes only slightly with pitch angle. Likewise, *R* changes quickly with pitch angle but only slightly with yaw angle. *Q* and *R* are also independent of depth changes because all of the probes will change by the same depth and this depth change will be cancelled in the *Q* and *R* differences.

If Q_{flow} and R_{flow} are calculated using the pressures from the probe in an unknown flow, the yaw and pitch angles of that flow can be determined using these two tables. A simple way to find the proper yaw and pitch angles for an unknown flow is to find the intersection of two contour lines. In the *Q* table there will be a contour line representing the constant value of Q_{flow}. Similarly, there will be a contour line of the constant value R_{flow} in the *R* table. Since the unknown flow must have a yaw and pitch angle that gives both Q_{flow} and R_{flow} then the two contour lines must meet at the correct yaw and pitch angle for the flow. Therefore, if each contour line can be represented as a piecewise linear line, the correct yaw and pitch angle can be found by finding the intersection of these two lines. Contour plots of Q and R separate and together are shown at the end of this chapter.

The numerical approach used in IMD's new wake survey analysis procedure is to build up the tables of *Q*, *R* and P_{Cal} during calibration. These tables are built in five degree increments over the calibration range. The tables are matrices indexed by the nominal yaw and pitch angles. Two extra tables record the actual measured pitch and yaw angle for each set of nominal angles. The completed data tables are input into a Matlab data smoothing and interpolation routine. This routine corrects for the differences between the nominal table locations and the actual values of the pitch and yaw angles. Further, it interpolates the data to a finer 1 degree grid. This produces new tables of *Q*, *R* and P_{Cal}

which contain data for yaw and pitch angles corrected to 1 degree increments. These are the three calibration tables used in wake surveys.

A different set of tables will be required for every probe head to account for head differences. Also, the data is not averaged over the four quadrants. This accounts for assymetry in the probe head.

The flow calculation program plots linear contours through this 1 degree spacing grid for both Q and R. It then calculates the intersection between these two contours lines. This intersection is the yaw and pitch angle of the flow. Once the yaw and pitch angles are known the magnitude of the flow is determined from the Pcal calibration surface.

1.2 Wake Survey Coordinate Systems

Two coordinate systems are used in wake survey analysis: the positioning stage coordinate system and the flow oriented coordinate system. The positioning stage coordinate system uses x as horizontal to starboard and y as positive upwards when facing the direction of the model. This coordinate system is used because it is the standard coordinate system used by the Unidex control system. This allows the positioning stage and its software to be easily used with other types of coordinate systems. The positioning stage coordinate system is only used to determine the physical position of the probe head. The position of the probe would be stored in this coordinate system. However, as soon as any calculations are performed, the flow oriented coordinate system is used and all data is transformed to this system.

The flow oriented coordinate system is x along the propeller axis positive from bow to stern. The y axis would be positive to starboard and the z axis would be positive upwards when facing forward.

In wake surveys, the typical output flow velocity components are the axial, tangential and radial velocity components. Also, the position of the sample point is usually given in polar coordinates. The axial velocity V_X is positive in the x-axis direction, the tangential velocity V_T is positive in the counter-clockwise direction and the radial velocity V_R is positive towards the propeller centreline. The position angle θ is measured counter-clockwise positive from the top centre position and the radius R is measured positive from the centreline. The coordinate systems are shown in figure 1.4.

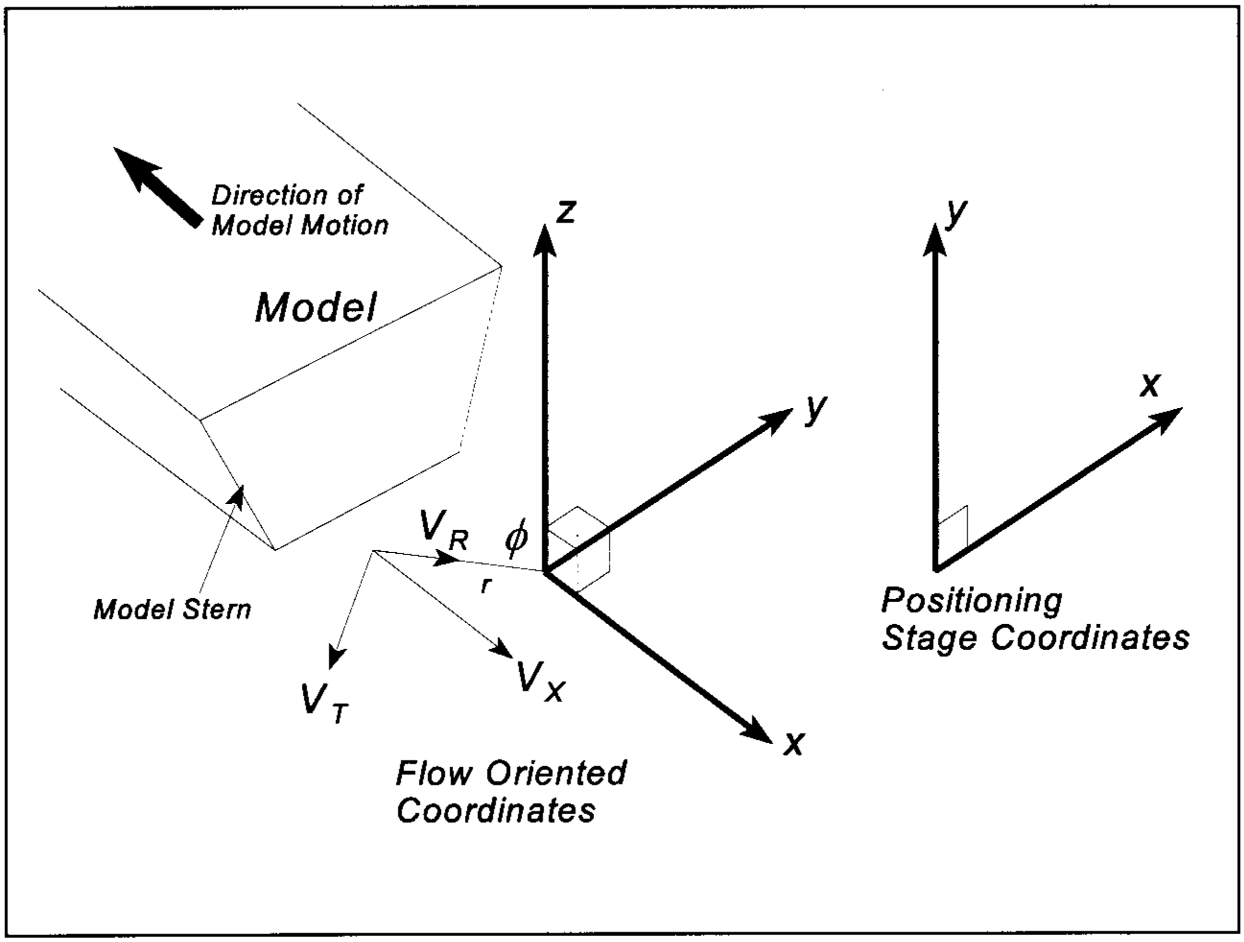


Figure 1.4 - Flow oriented and positioning stage coordinate systems.

A third set of coordinates is used which is based on the velocity magnitude and the yaw and pitch angles of the flow. This coordinate system is merely a subset of the flow oriented coordinate system. The yaw angle, α, is measured from the x axis, positive towards starboard. The pitch angle, β, is also measured from the x axis, positive in the down direction. The yaw and pitch angle orientation is shown in figure 1.5

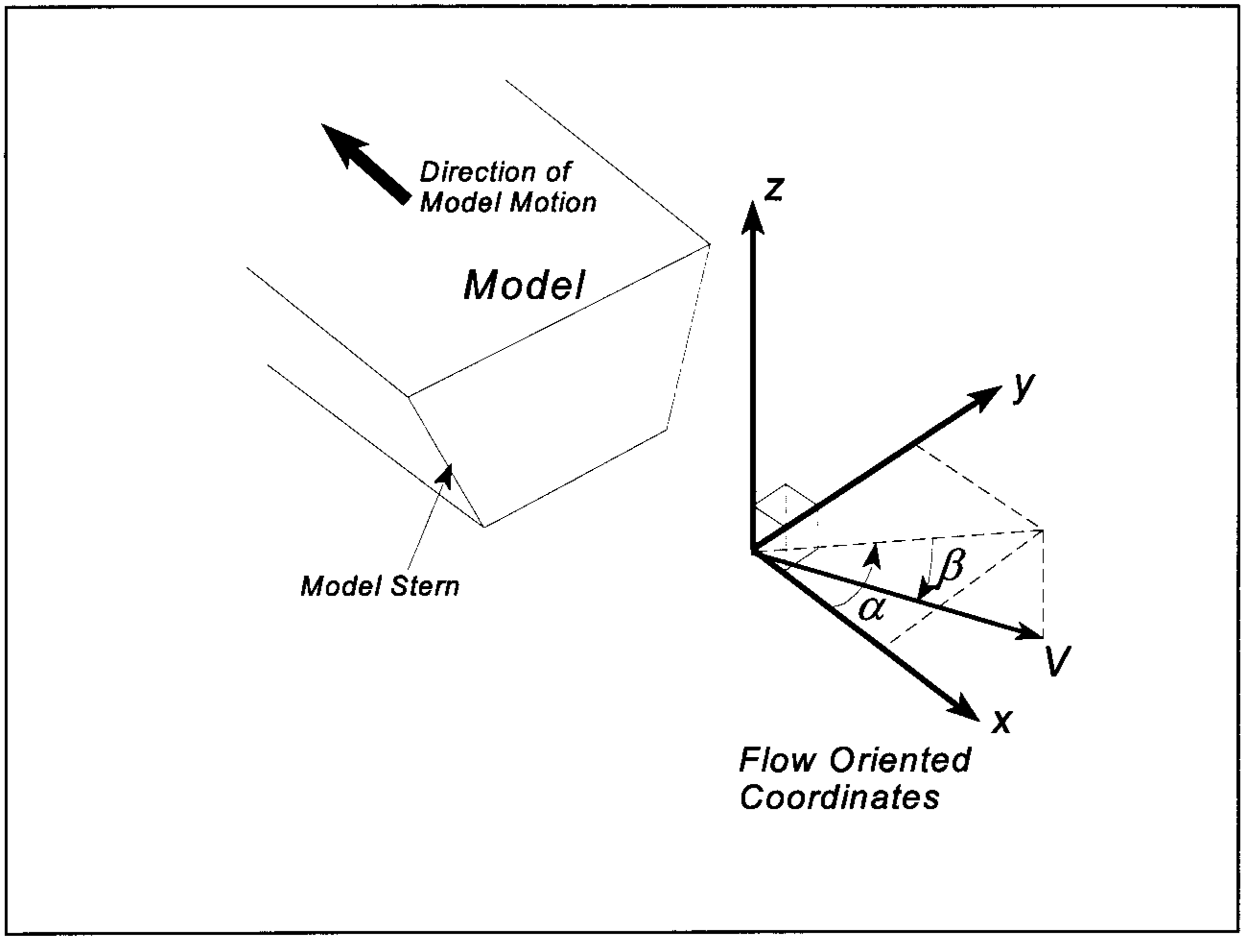


Figure 1.5 - Orientation of yaw and pitch angles for the flow oriented coordinate system.

The probes are calibrated in terms of velocity magnitude and yaw and pitch angles. The axial, tangential and radial velocity components can be found by first resolving the velocity vector *V* into its rectangular *x*, *y* and *z* components. From figure 1.5, it can be shown that:

$$V_x = V\cos(\beta)\cos(\alpha)$$

$$V_y = V\cos(\beta)\sin(\alpha)$$

$$V_z = V\sin(\beta)$$

The axial component is V_x is the same. The tangential and radial velocity components are found using the geometry of figure 1.6. They are:

$$V_T = -V_y\cos(\theta) - V_z\sin(\theta)$$

$$V_R = V_y\sin(\theta) - V_z\cos(\theta)$$

Using this coordinate system, the relationship between the yaw and pitch angle of the flow and the velocity components is found.

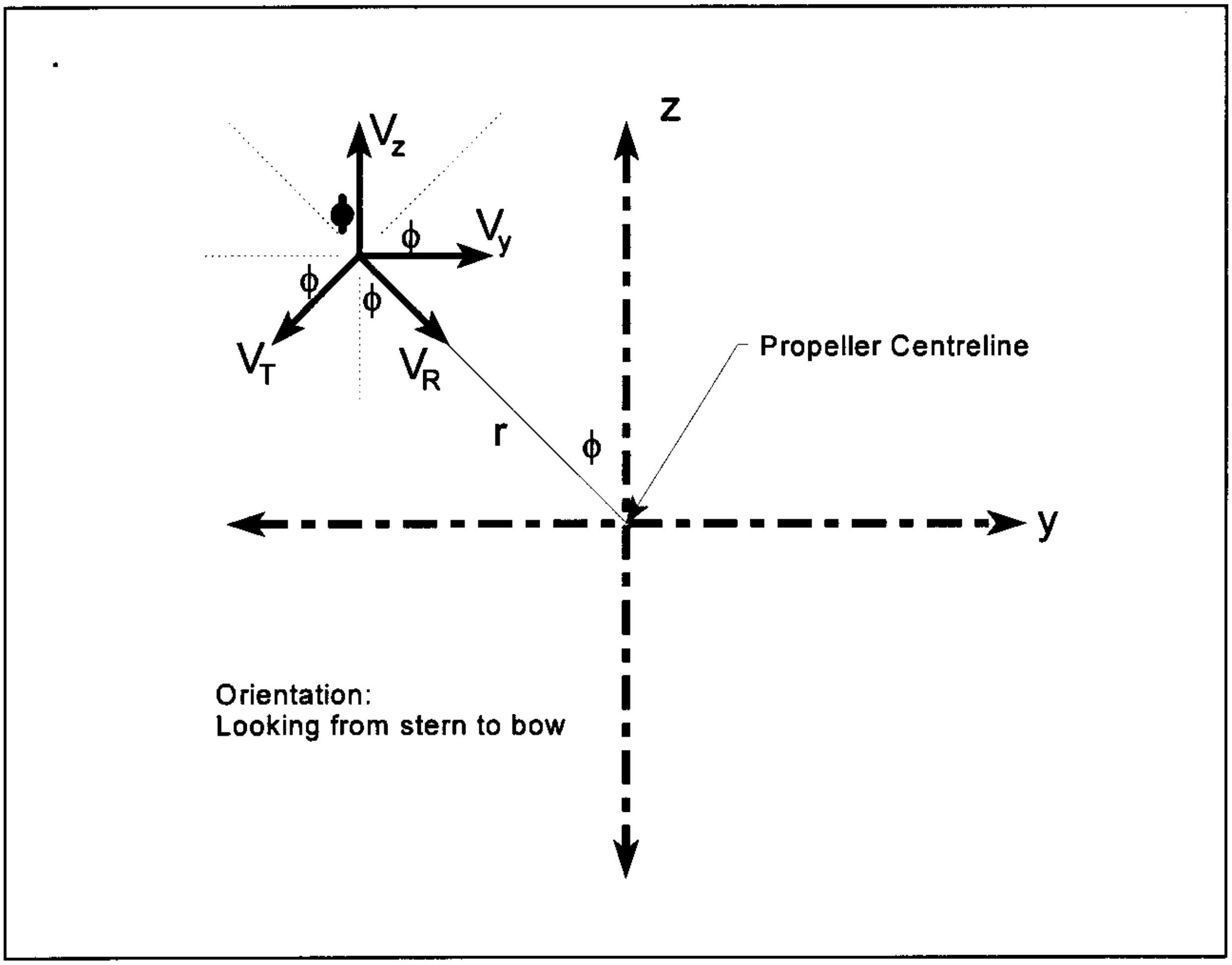


Figure 1.6 - Resolving *y* and *z* components into tangential and radial components.

2.0

Wake Survey Analysis Procedure

Topics:

- ❑ Overview of the standard wake survey analysis procedure
- ❑ Detailed description of entire procedure
- ❑ Summary of probe positioning software
- ❑ Summary of VMS analysis software
- ❑ Summary of Matlab wake survey curve fairing and plotting

Wake surveys should be performed in a consistent manner from test to test and model to model. This helps to ensure the quality and reliability of the results.

2.1 Wake Survey System and Model Setup and Checks

The model setup should be performed in the standard way. However, the pitch clamping mechanism should be installed and levelled at the time of model preparation. Once the model is ready to be installed on the carriage, the pitch clamping mechanism should be removed from the model and installed on the tow post. Then the model can be attached to the pitch clamp mechanism. This is done because of the bulk of the pitch clamp mechanism and the difficulty with installing it while the model is attached.

Before the model leaves the shop, the positioning stage should be attached to the stern. First, a plywood base is added to the model and levelled. Then the positioning stage is added to the base and lined up to ensure that it is perpendicular to the ship level and perpendicular to the centre line of the model. Once it is secured to the base, the positioning stage controller and computer should set up and started.

After the stage has been attached, the vertical position of the probe with respect to the stage should be manually adjusted while the stage is at its maximum height. At the same time, the bow to stern location of the probe should be determined and set. This ensures that the probe will be able to reach all points on the propeller disk being surveyed. When the height has been set, the controller should be homed and initialized. The rotation of the probe head should then be levelled to ensure that the top transducer is truly the top. At the same time, the probe arm should also be levelled.

Once the system has been installed and levelled, the probe can be moved to location of the propeller centre line. This can be done by either determining the actual location of the centre line physically or by determining a known point somewhere else and moving the stage a set amount to the position of the centre line. When the probe is at the centre line the stage positioning software should be calibrated and the calibration file saved. Once it is calibrated, the system should be homed using the new calibration file. Then the stage and probe should be checked to ensure that the probe can safely reach all require sampling positions without hitting the model.

Before the probe is attached to the positioning stage, it should be purged with silicon oil to remove any trapped air in the pitot tubes. Further, the connections and seals should be tightened and checked for leaks. When handling the probe outside of the water, the protective covering should be kept on the probe head at all times. This keeps the pressure transducers immersed in fluid. Once the entire system is installed on the model and on the carriage, the protective cover can be removed.

Before any tests are performed, a depth check should be performed on the probes. This involves re-zeroing the probe pressures and then moving the probes (via the positioning system and control software) to various depths. Then the supplied analysis program *WAKE_DEPTH_CHECK* can be used to check the results. The probe pressures should be nearly the same as the probe depth. If any serious descrepancies arise, the erroneous probe transducers should be re-calibrated. This can be done using the positioning stage to set the required depths. Further, probe depth checks should be performed at the beginning of every day and after every model change.

2.2 Wake Survey Test Descriptions

Wake surveys require finding the pressures at set locations in a propeller wake. Usually, these sample points are taken on several concentric circles centred on the propeller centre line. Their radii are percentages of the propeller diameter. A typical set of sample positions is shown in figure 2.1. More samples are typically taken near the hull, where the flow changes the most. Different sample points may also be selected in the presence of brackets or struts and other appendages.

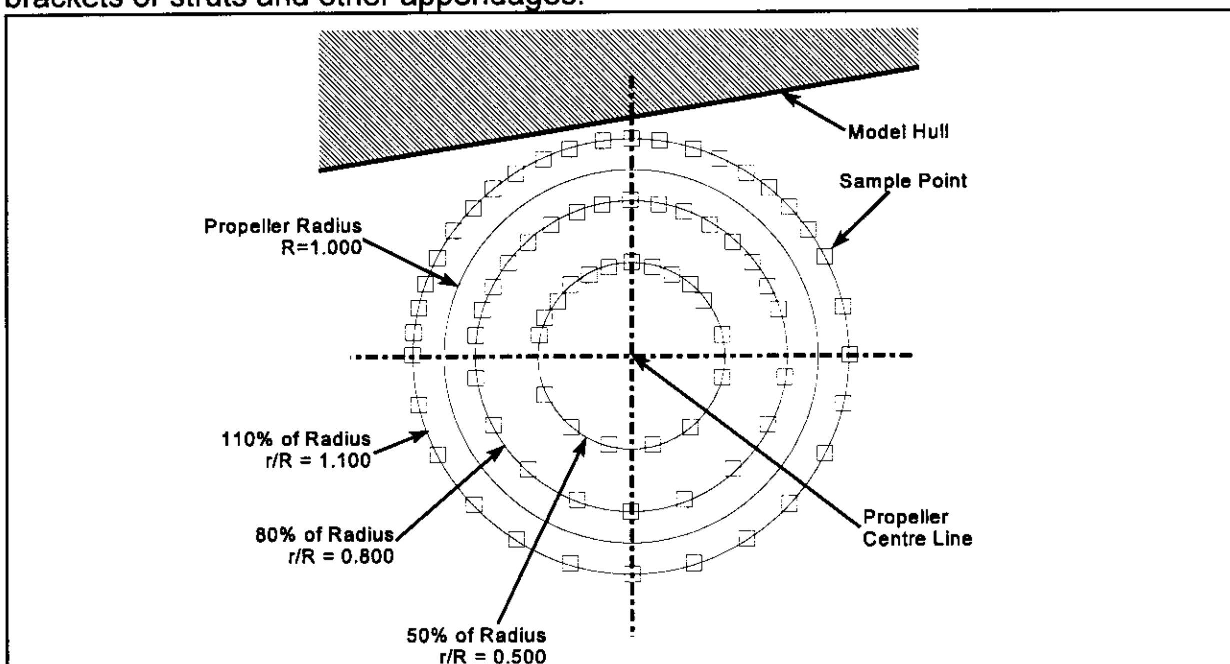


Figure 2.1 - Typical sample points for a wake survey.

For a wake survey test, the probe is manoeuvred to each of these sample points while the ship is in motion, and the pressures are measured. Then using the algorithm described in section 1.2, the axial, tangential and radial velocities are calculated.

The actual sample positions are determined ahead of the actual wake survey test. The points can be generated using the positioning stage software, called *WAKE.BAS*, which has a built-in sample position generating routine. For a wake survey test, as many as 200 sample points may be used. To reduce the amount of tank time required to sample all of these points, many points can be sampled in one carriage run down the tank. The positioning stage can move the probe to the first position, stay there for the sample time, move the probe to the next position, stay there and so on. The probe positioning software has a built in mode that takes the generated sample points and then groups them into *runs*. Each *run* contains the sample points for one run down the tank. For example, run 1 might contain the first 12 sample points, run 2 the next 12 and so on until the sample points are completed. In this manner, if 120 points are to be sampled, and 12 points can be sampled in one carriage run, then the total number of runs would be 120 / 12 = 10 carriage runs. As the ship speed increases, less points can be sampled in one carriage run (assuming the same sample time) and the number of carriage runs will increase.

These grouped runs and their points are stored in a *.RUN* file for every wake survey test. The stage positioning program can then be used to automatically execute a *.RUN* file. The user only has to select which run and then tell the program to start the positioning sequence as the carriage run is started.

During each wake survey carriage run, the pressures, probe position and carriage speed are sampled by the carriage data acquisition system. As the runs are acquired and stored, the on-line wake survey analysis software can be used to start analyzing the results. First, the test settings must be set. If the DAS data acquisition system is being used, then the raw DAS data files need to be converted to the standard GEDAP file format using a supplied utility, *WAKE_CONVERT*. Otherwise, the wake survey software can operate directly on the raw data files. The first operation is to select the time segments. The first time segment is the zero tare segment, taken at zero carriage speed just before the test. The remaining segments are taken at the various sample positions. Then the velocities at each sample position will be computed and output to a statistics storage file. Also, a check plot of the velocities versus the angle theta will be made to show the operator how the test is progressing. This cycle is repeated until all of the sample points (and hence carriage runs) have been analyzed.

During run analysis, a statistics file is built, containing all of the computed velocities at each sample point for the test. Another file, called the log file, keeps a record of all the wake survey analysis commands used for the test. When the test has been completed, another wake survey program is used to sort the output results by radius from the centre line. For example, if 5 radii were used for the sample points, then there would be five output files, one for each radii. These files contain columns of radius, angle from the top, vertical position, axial velocity, tangential velocity and radial velocity. The entire run statistics can also be converted into an ASCII file suitable for import into a spreadsheet program or database such as Excel or DBase IV.

Typical test results are shown in figure 2.2 for the axial velocity, V_x. The results are shown for three different radii. The radii are represented as some ratio to the propeller radius, R. Thus, r/R = 1.10 refers to a radius that is 1.1 times as large as the propeller diameter, and r/R = 0.50 refers to a radius that is 0.5 times as large as the propeller diameter.

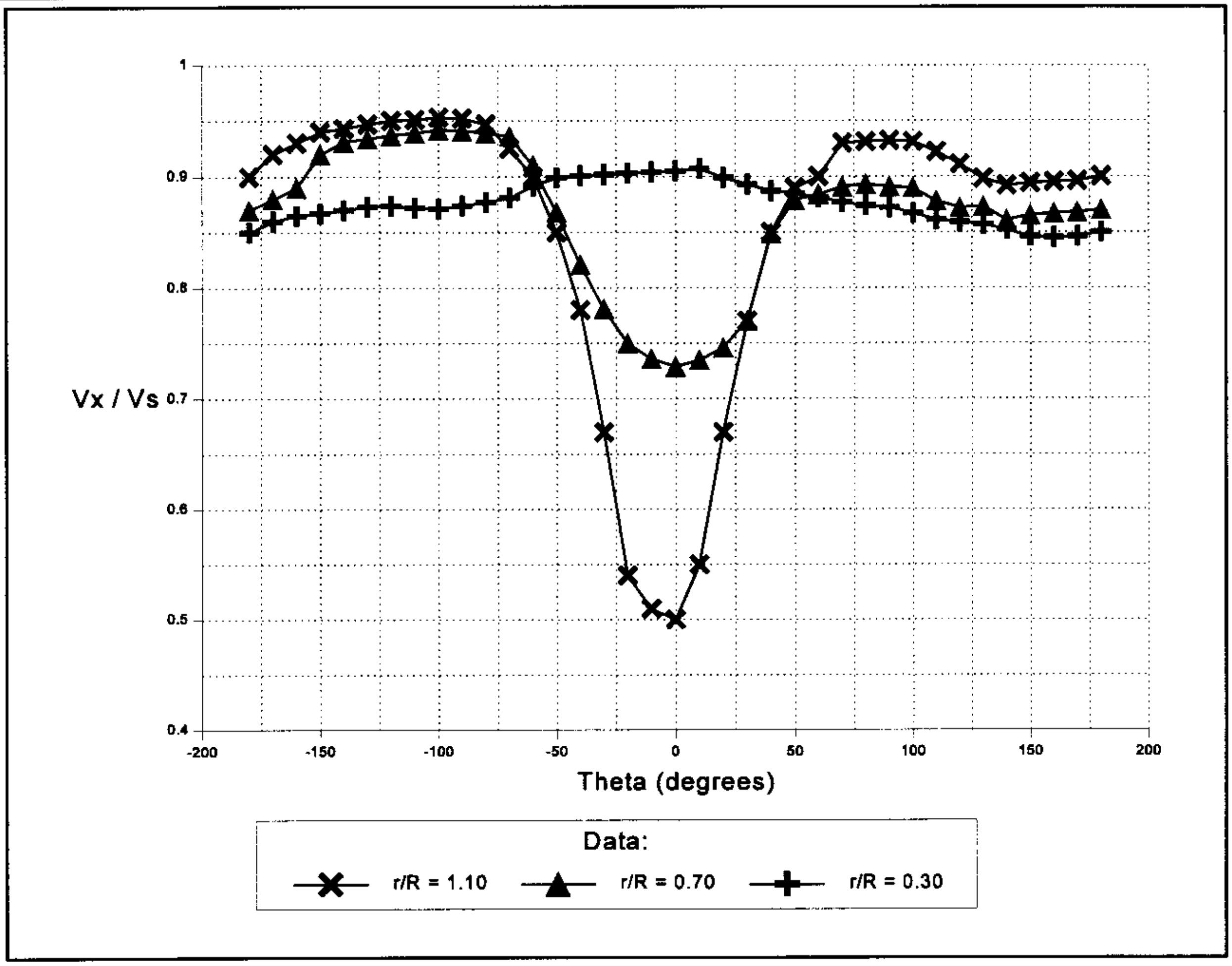


Figure 2.2 - Plot of V_x/V_s versus Theta for a typical wake survey test.

Once the wake survey test has been completed, the raw data for each set of velocities versus theta's for all of the radii in the test have to be faired. A custom Matlab application has been written to help in interactively fitting spline curves to the raw data. This program allows the user to change the control points of a spline curve fitted through the raw data. Once these curves have been fit to all of the velocities and radii, this program has built in plotting routines to produce the required output. It can also determine the Fourier coefficients for the data and output them to a file. The faired data curves are also saved for future use and editing.

3.0

VMS Analysis Software

Topics:

- VMS on-line software operation
- Wake survey data file structure and naming conventions
- Wake survey analysis procedure
- Detailed descriptions of all wake survey analysis programs

The main, on-line wake survey analysis software was written for the VMS operating system. It utilizes NRC's GEDAP data files and many GEDAP data handling routines. Several new GEDAP programs were also written in Fortran for custom applications.

Wake surveys are typically performed at several speeds. A wake survey test consists of all of the runs necessary to perform a wake survey at one speed. For a particular model, several speeds and hence several tests will usually be performed. Each of these tests will produce different output files which should be named according to the test speed and test type, for example, free to sink and trim or fixed. For example, if the project number was 617, the full scale speed was 14 knots and test type was fixed, the main output file might be name `P617_VS14_FIXED`.

The wake survey analysis software works in four stages. The first stage is the collection of data using DAS (or some future acquisition system). This data is then transferred to a permanent storage area on the main computer system in a logical drive called TOW (for tow tank data). This data will be stored as either DAS files or GEDAP files.

The appropriate run files are then copied to a different working data directory for every test. If the raw data files are DAS files then they must be converted to GEDAP as they are being copied to the working directory. Otherwise, the files can be copied directly from the permanent storage location to the temporary location. This temporary storage of the run data allows the wake survey analysis software to safely operate on the data without affecting any of the permanent storage data.

The second stage involves selecting the appropriate time segments from the data and calculating the velocity components. This produces an ASCII output file storing all calculated and raw data statistics. A different output file should be kept for each test (or speed). Stages 1 and 2 are completed once all of the runs for the test are acquired, converted to GEDAP, the segments selected and the velocity parameters calculated. Only when this is completed can stages 3 and 4 be started.

The third stage involves processing this data into several output files. One of these files will be a listing of all collected data organized in rows. This data is suitable for immediate importing into a spreadsheet program such as Excel or a database program. Another set of output files will be created to store the calculated velocity data for each radii around the shaft centre line. There will be one of these files for each radii. These files are the main output from the wake survey VMS software. The temporary raw data files stored in the data directory can be deleted once the data appears to be satisfactory.

The fourth and final stage takes the output files for each radius and uses a Matlab routine for final data smoothing and output. This routine allows the user to control a spline fit to the data by manipulating the spline control points. The program then takes the faired data for all of the radius files for that test and produces the output velocity, vector and contour plots. Finally, the program finds the Fourier series expansion for the faired data and saves this data to an ASCII file.

3.0.1 Data File Naming:

The VMS wake survey analysis software creates and uses several different data files. The first data files created are temporary copies of the raw data files which are collected on the tow carriage's data acquisition computer and stored there or on a logical storage device, such as `TOW`. If these raw data files are DAS files, then they must be converted to GEDAP data files as they are being copied to the data storage directory using the command `WAKE_CONVERT`. For the GEDAP acquisition system, the files will already be in the GEDAP format and can be copied from the permanent storage directory to the temporary directory. Either of these methods will create a separate, temporary storage directory in the current directory for every run number and store the new GEDAP files for that run in the sub-directory. The name of this directory is determined by the test name for that particular test and run number. For example, if the test name was `WAKE_TEST1` and the run number was `234`, then the storage directory would be:

```
[.WAKE_TEST1_R234]          ! R234 =  Run 234
```

Within this directory will be the temporary copies of the raw GEDAP files for every channel. They will have the same name as above, such as `WAKE_TEST1_R234.001` and `WAKE_TEST1_R234.005`.

The GEDAP data files used in wake survey analysis are standard GEDAP V1 files, which use an implicitly defined x variable. That is, they store the raw data for the y values of the data stream and a set of index variables to determine the x variable (ie. the time variable) for each y value. There will be one GEDAP file created for each data stream (or channel) in the test. The extension for the files will contain the data stream number. For example, if the data stream number was 10 then the extension for that data file would be `.010`. If the data file root name was `WAKE_TEST1`, and the DAS run number was `234` and there were `11` channels, the following GEDAP files will be created and stored in the directory `[.WAKE_TEST1_R234]`:

```
WAKE_TEST1_R234.001        ! File for data stream 1
WAKE_TEST1_R234.002        ! File for data stream 2
WAKE_TEST1_R234.003        ! File for data stream 3
      . . . . . .
WAKE_TEST1_R234.011        ! File for data stream 11
```

These files are VAX binary files. The only way to read their contents is to use the GEDAP command `LIST`. For example, to read the contents of the file `WAKE_TEST1_R234.005`, type:

```
$ list wake_test1_r234.005        ! List contents of GEDAP V1 file
```

This will display both the header parameters and raw data for the file. Generally, however, there is no need to list the contents of the GEDAP files themselves.

As the wake survey analysis is being performed for a test (ie one speed), three output files will be continuously updated. The first file called the LOG file. This file keeps a record of all of the wake survey programs and settings used during the analysis of the current test. The second file is called the STATS file. This file stores a record for each run segment. The records contain the run number, segment number, segment end and start times, raw data stream statistics and the calculated velocity parameters. The last file is called the zero tare depth check file. This files stores a the pressure differences between the five transducer readings and the depth for the static zero taken before each test. This is used to track the performance of each transducer.

These three output files will be named according to a VMS global variable called `WAKE_OUTPUT_FILE`[1]. This variable will be different for each test (ie each speed). The wake survey analysis programs will add the ending `_LOG.DAT`, `_STATS.DAT` and `_DEPTH_CHECKS.DAT` to the output file name variable. For example:

If `WAKE_OUTPUT_FILE = WKS_VS14_FIXED`

Then the output files from the wake survey analysis programs would be:

```
WKS_VS14_FIXED_LOG.DAT              ! LOG file
WKS_VS14_FIXED_STATS.DAT            ! STATS file
WKS_VS14_FIXED_DEPTH_CHECKS.DAT     ! Zero tare depth check file
```

As well, all output files use `WAKE_OUTPUT_FILE` as the root name with another description added on. For example, `WAKE_SPREAD` (described later) will produce a file with `_SPREAD.DAT` added on:

```
WKS_VS14_FIXED_SPREAD.DAT     ! Spreadsheet ASCII file
```

The `WAKE_OUTPUT_FILE` symbol name is often called the root output name because it is the first part of every output file. This symbol is set using the `WAKE_SET.COM` file.

3.1 Wake Survey VMS Analysis Procedure

The following section outlines the wake survey analysis procedure for using the VMS software. Detailed descriptions of each command procedure are contained in section 3.2.

The wake survey analysis programs are contained in an applications directory on the DEC Alpha systems (Gnome, Grynch, Ghost, Golem and Troll). These programs can only be run on the Alphas and not on any other nodes.

[1]Setting VMS global variables for use with the wake survey analysis software is explained under section 3.1 for the VMS program `WAKE_SET.COM`.

The wake survey analysis programs were written using many GEDAP routines and GEDAP file types. Therefore, the GEDAP system must be started. This is done using two commands:

```
$ gedap        ! Start the GEDAP system
$ motif        ! Set GEDAP to the MOTIF windows system
```

These lines should usually be added to your `LOGIN.COM` file in your `HOME` directory so that they are automatically executed every time you log in to the VMS system. The wake survey analysis software start-up symbol should also be added to your `LOGIN.COM` file so that it executes whenever you log on:

```
WAKE_START :== @WKS:[TEST]WAKE_START
```

If you are using a remote node you must also set the display and security options. For example, if you are on the local node MAROON, (ie. sitting in front of it) and are logging into GNOME, you must set the display in the GNOME window to MAROON:

```
GNOME $ set display/create/node=MAROON    ! Set display node to MAROON
```

This sets the display to MAROON so that any graphics windows from GNOME will be output to your screen on MAROON. **NOTE:** The security option under your session manager on MAROON should also be adjusted to allow output from your user name on GNOME to appear on MAROON.

The next step is to create a data storage directory. This should be done in a project directory or some other directory allocated to the specific wake survey project. A new directory should be created for every different wake survey test speed. This will keep all of the data files separate and all of the set-up files separate. For example, for the speed V_S = 12 knots, a directory called `[.WKS_VS12]` or some similar name (the name may have project number or whatever) should be created in the root project directory. To create the directory, us the VMS `CREATE` command:

```
$ create /directory [.directory_name]        ! Create new directory
```

This will create a sub directory in the current directory. Move to this directory using `CD` or `SET DEF`:

```
$ cd [.directory_name]        ! Change to the new directory
```

Once in this directory, the wake survey analysis programs can be activated:

```
$ wake_start        ! Activate the wake survey analysis programs.
```

This command will search the current directory for the file `WAKE_SET.COM`. This is the main setup file for wake survey tests. It determines input filenames and data storage location, the printer name and data stream settings. It also sets a root output file name symbol, called WAKE_OUTPUT_FILE. The name entered for this symbol will be the root name for all subsequent output files.

If WAKE_SET.COM is found in the directory, WAKE_START will leave it there. If not, it will copy a new version of WAKE_SET.COM to the directory. Now the wake survey software analysis commands have been activated. It should be noted that these commands are really VMS DCL command procedures. However WAKE_START sets the command names as symbols so that the commands can executed by simply typing the command name without the usual @ symbol required for running DCL command procedures. The symbols also point to the applications directory where the wake survey command procedures are located.

The WAKE_SET.COM file must now be edited to change the settings for the analysis to be performed. This file can be edited using any ASCII file editor, such as EVE:

```
$ eve wake_set.com    ! Edit the set-up program
```

Within this file all of the test settings can be adjusted. For further details see WAKE_SET.COM in section **3.2**. Once the changes have been made the file should be saved. Then, the change must be made active by running the set-up program. This can be done by simply typing its name (do not include the .COM):

```
$ wake_set            ! Make the new settings active by running the program
```

This will display the new settings to the screen.

By keeping separate directories for every wake survey test (ie. every different speed) which will contain the WAKE_SET.COM file and the output data files, changing test parameters can be made simple. For example, if every test has its own directory and hence its own WAKE_SET.COM file, then to analyze any test data simply change to that directory and activate the WAKE_SET command. If you had two test for speeds of 18 knots and 26 knots you would make two directories, say:

```
[.WKS_TEST_VS18_FIXED]
[.WKS_TEST_VS26_FIXED]
```

Each of these directories would contain it own data files for the test and a WAKE_SET.COM file for set up for that test. If you wanted to do some analysis on the 18 knot data simply change to that directory, execute the WAKE_SET command and the wake survey analysis software will set up to analyze data in that directory.

```
$ cd [.wks_test_vs18_fixed]    ! Change to the test directory
$ wake_set                     ! Activate the settings for the test
```

Every time WAKE_SET is used, it will put a listing of its settings in the LOG file. The log file would be named WKS_TEST_LOG.DAT. To list this file use the command:

```
$ wake_log                ! List the log file to the screen
```

The log file will be displayed on the screen. Note that all lines preceded with a @ are command procedures and all line preceded with a ! are settings or other comments. After the wake survey settings have been adjusted, test data can be analyzed.

First, to analyze any run, its raw data files must be copied to a sub-directory in your current directory, named according to section **3.0.1**. This is done using either `WAKE_CONVERT` or some other program. These files will be temporary and are only used during the wake survey analysis. They can be erased after the analysis is successfully completed because the raw data files are permanently stored at another location by the data acquisition system. If the raw data files from the data acquisition system are already GEDAP files, then they only need to be copied to the proper location[2]. If the wake survey run data was acquired by DAS, then the raw data needs to be converted from the DAS format to the GEDAP format. This is done with the command:

```
$ wake_convert P1 P2          ! Convert a series of runs from P1 to P2
```

This command will create the proper directories for you. Once these temporary raw data files have been copied to the current data directory, analysis of runs can begin.

Typically, the first run will be static depth check. This is done by acquiring data while the y position of the probe is changed to several different levels using the motor control system. This depth check is then analyzed using a program called `WAKE_DEPTH_CHECK.COM`:

```
$ wake_depth_check          ! Analyze the depth check run
```

This program will prompt for two variables:

```
Run Number to Process:              ! Run number of the depth check
Depth Range of the Test (cm):       ! Range of y values in the check
```

The user has to pick out the time segments for each position. The program will then find the differences and errors between the probe pressures and the probe depth (ie. the y-position). If the results appear suitable, the test can continue. If they are not suitable, the probes may need to be re-calibrated.

Standard wake survey runs are analyzed using `WAKE_RUN`. This command performs analysis on multiple runs.

[2]Note: A command procedure for copying raw data files acquired by programs other than DAS has not been developed yet. Such a program would need to copy the files to temporary storage directories according to the naming convention in section **3.0.1**.

```
$ wake_run P1 P2                    ! Analyze runs P1 to P2
```

This command starts with the segment selection routine. After the segments have been chosen, the program will calculate the flow velocity components. Finally, it will produce a plot of the data collected up that point. This is useful for monitoring the progress of the test while the actual test runs are still proceeding.

As the analysis procedure operates it builds up three files. The first file is the statistics file which stores all of the computed velocities and other parameters. This file is named using the root output name and `_STATS.DAT` added to the end. For example, if the root output name was `WKS_TEST` the statistics file would be named `WKS_TEST_STATS.DAT`. At any time during the wake survey test the statistics file can be viewed using `WAKE_STATS`. This procedure will display the stats for any user selected run or for all of the runs and also allows the choice as to which statistics are displayed. The wake survey analysis routine also computes a depth check on the zero tare taken at the beginning of each test. This data is stored in the file with the `_DEPTH_CHECKS.DAT` added to it, as in `WKS_TEST_DEPTH_CHECKS.DAT`. This file can be viewed by using the command `WAKE_DEPTH_STATS`.

The plots of the data can also be made at the prompt using the command `WAKE_PLOT_STATS`. This is the same plotting procedure used by `WAKE_RUN` at the end of every run analysis.

Once the test is completed and all of the velocities calculated, the data needs to be sorted by radius. This is done by the command `WAKE_SORT_RADII`. This will create a new file for every radius in the test. These new files will be named with the `_RADn.DAT` extension, as in `WKS_TEST_RAD3.DAT` for the third radius. Once these files have been produced, they should be imported into the Matlab program `WKS_FAIR` (see section **5.0**).

There are several other utility commands available for wake survey analysis. The command `WAKE_RESET` will erase the log file and the stats file for the current test. This can be useful when the log file becomes corrupted with erroneous data points and all of the analysis needs to be restarted. `WAKE_DELETE_RUNS` will delete the raw GEDAP data files and their storage directories.

`WAKE_SPREAD` converts the statistics file to column ASCII file delimited by spaces that is suitable for import into a spreadsheet program such as Excel. `WAKE_RAD_TO_DOS` converts long radius file names into shorter, 8 character names suitable for import into the MS-DOS operating system.

There is also a command for reanalyzing the wake survey data called `WAKE_REAN_RUN`. This command will take previously selected time segments and recompute the velocity values. This command is very useful when combined with `WAKE_RESET`. For example, if an erroneous data point was entered into the statistics file the file can only be regenerated by using `WAKE_RESET` to erase the statistics file and reanalyzing all of the data. The only difference is that `WAKE_REAN_RUN` uses all previously entered information so no user input is required. It could also be submitted as a batch job.

3.2 Wake Survey VMS Analysis Commands

The wake survey VMS analysis commands are described in this section in the general order they are used in. Each description has the following format:

Usage:	How the command is entered and its parameters.
Description:	What the command is used for.
Operation:	How the command is used.
Warnings:	Any special warnings
Log File:	How the command appears in the LOG file.
Input Files:	Input file names and types.
Output Files:	Output file names and types.

General Description of Wake Survey VMS Analysis Programs:

General Use:

These commands are used in wake survey analysis. Many times they may call other procedures that are not generally used on their own.

`WAKE_START`	Starts the wake survey software and sets all global symbols to the wake survey analysis programs.
`WAKE_SET`	Used to set all of the parameters required by the wake survey analysis programs, such as raw data name and location, storage location, output file names, calibration file name, printer names and data stream numbers.
`WAKE_CONVERT`	Converts multiple runs from the DAS format to the GEDAP format.
`WAKE_DEPTH_CHECK`	Analyzes depth checks for pressure differences.
`WAKE_RUN`	Performs analysis of multiple wake survey runs. Allows selection of time segments, tares the data, calculates the velocities and produces check plots for the operator.
`WAKE_LOG`	Displays the LOG file contents.
`WAKE_STATS`	Displays the velocity statistics collected for all of the runs in the test.
`WAKE_DEPTH_STATS`	Displays the zero tare depth check results for each run.
`WAKE_PLOT_STATS`	Plots the velocity data for the different velocities and radii.
`WAKE_RESET`	Erases the LOG, STATS and zero tare depth check output files. Used to start the analysis over again. Usually used with `WAKE_REAN_RUN`.
`WAKE_DELETE_RUNS`	Deletes the raw GEDAP data files and their storage directories for multiple runs. Only used once the data analysis is completely finished.
`WAKE_SORT_RADII`	Sorts the velocity data by radii. Creates files that can be used by WKS_FAIR in Matlab for VMS.
`WAKE_SPREAD`	Produces an ASCII column file, delimited by spaces, of the velocity statistics suitable for import into a spreadsheet

program, such as Excel.

`WAKE_REAN_RUN` Reanalyzes multiple runs using the previously entered time segments.

`WAKE_RAD_TO_DOS` Converts the long file names of the sorted radii files into shorter names compatible with MS-DOS. Only necessary when bringing files to MS-DOS.

Commands Called by the Main Commands:

These commands are not generally accessed by the user. They are usually only called by the above, main commands.

`WAKE` Selects the time segments and tares the data for a single run. Calculates the velocity and outputs the statistics to the stats file. This command is usually only accessed by `WAKE_RUN`.

`WAKE_ANALYZE` A command called by `WAKE` which does the actual velocity calculations. It is also used by `WAKE_REAN_RUN` and `WAKE_REAN`.

`WAKE_DASGEDAP` Performs a DAS to GEDAP conversion for one run. Called from `WAKE_CONVERT`.

`WAKE_PLOT_DATA` A sub component of `WAKE_PLOT_STATS` that converts the stats file to sorted GEDAP radii files and prepares them for plotting with `GPLOT`.

`WAKE_REAN` Re-analyzes one run at a time. Is only called from `WAKE_REAN_RUN`.

`WAKE_CHECK_ZERO_DEPTH` Performs depth check analysis on the zero tare segment.

WAKE_START

Usage:

```
$ wake_start                    ! Set up the wake survey programs
```

Description:

WAKE_START sets up the required symbols that point to the various wake survey analysis software programs. It also copies the WAKE_SET.COM file to the current directory for use in setting the wake survey variables.

Operation:

WAKE_START uses global DCL symbols to define the names and locations of the required analysis programs. It neeeds to be run only once for every session (ie. when you log in to a node) or any time a new directory is created (because it will copy the WAKE_SET.COM file to the new directory). The wake start global variable should be added to your LOGIN.COM file in your VMS HOME directory. Add the following line:

```
$ WAKE_START :== @WKS:[TEST]WAKE_START
```

This will enable you to simply type WAKE_START to start the wake survey analysis procedures.

Warnings:

None.

Log File:

This command does not get recorded in the log file.

Input Files:

None.

Output Files:

Will copy WAKE_SET.COM to the current directory if it does not exist there.

WAKE_SET

Usage:

```
$ eve wake_set.com        ! Edit the file (use any ASCII editor)
$ wake_set                ! To make the settings active
```

Description:

WAKE_SET.COM sets all of the variables used by the wake survey analysis programs.

Operation:

WAKE_SET.COM uses global VMS symbols to define the wake survey test parameters. They are set by editing the actual WAKE_SET.COM file and changing the global variable definitions contained in the file. The global symbols are then set by using the WAKE_SET command at the VMS prompt:

```
$ wake_set                ! Make the settings active
```

Global Symbols:

For a symbol to be global it must be specified with a double equals sign. Further, all of the symbols are defined as string symbols. For example, to set the symbol WAKE_NUMBER_OF_CHANNELS, which tells the wake survey programs how many data channels (or streams) there are, you would type:

```
$ WAKE_NUMBER_OF_CHANNELS :== 13
```

The double equals sign means it is a global variable. This simply means that it will stay active until you erase it or end your current session. The colon tells VMS that WAKE_NUMER_OF_CHANNELS is a string symbol and not an integer symbol. This command could be typed at the command line but it is better to group them all in one file, WAKE_SET.COM, which sets all of the required variables. This file contains basically lines similar to the above which are all of the wake survey program settings. Running the WAKE_SET essentially sets all of these variables using global commands.

To change any of the variables, you must edit the file using whatever editor you prefer. Then, change the required settings. For instance, to change the number of channels, find the following lines in WAKE_SET.COM:

```
$!
$!--------------------------------------------------------------------
$! WAKE_NUMBER_OF_CHANNELS :==
$!    Tells the program how many channels there are in total.
$!    Example: WAKE_NUMBER_OF_CHANNELS :== 7   <--There are 7 channels
$!--------------------------------------------------------------------
$ WAKE_NUMBER_OF_CHANNELS :== 11
$!
```

Note that the lines which start with `$!` are comment lines. They are there only to help the user understand the nature of each wake survey variable. Only lines starting with just `$` are actual command lines. To change the number of wake survey data channels, simply change the number 11 to the new number, say 9.

```
$!
$!--------------------------------------------------------------------
$! WAKE_NUMBER_OF_CHANNELS :==
$!    Tells the program how many channels there are in total.
$!    Example: WAKE_NUMBER_OF_CHANNELS :== 7   <--There are 7 channels
$!--------------------------------------------------------------------
$ WAKE_NUMBER_OF_CHANNELS :== 9
$!
```

When the settings have been set properly, write the file back to the disk (ie. save it!). Then run the `WAKE_SET` command by typing:

`$ wake_set` ! Make the new setting(s) active

at the VMS `$` prompt. This will set all of the wake survey global symbols to the settings you just specified in the file.

Hint: It is often helpful to use the comment statement (!) to allow the selection of several options. For instance, the global symbol `WAKE_SOURCE_LOCATION` might typically have only three settings. Therefore, put all three in `WAKE_SET.COM`, and comment out the two not being used. In the code shown below, the current setting is TOW because it does not have an exclamation point (which means a comment in VMS) while other two do.

```
$!
$!--------------------------------------------------------------------
$! WAKE_SOURCE_LOCATION :==
$!    Sets the location of the raw source data time series files.
$!    Possible locations include TOW and ICE
$!    Example:     WAKE_SOURCE_LOCATION :== TOW
$!--------------------------------------------------------------------
$ WAKE_SOURCE_LOCATION :== TOW
$! WAKE_SOURCE_LOCATION :== ICE
$! WAKE_SOURCE_LOCATION :== DISK$PROJECT1:[PJ94617]
$!
```

To see what a current symbol is while you are in VMS, use `SHOW SYMBOL` (or `SH SYM` for short). For example, to see `WAKE_NUMBER_OF_CHANNELS` type:

```
$ sh sym wake_number_of_channels
```

VMS displays the message:

```
WAKE_NUMBER_OF_CHANNELS == "11"
```

If the symbol is not defined, VMS will give the error message:

```
%DCL-W-UNDSYM,undefined symbol - check validity and spelling
```

All of the wake survey symbols start with the word `WAKE`. Therefore, to see what all of the wake surveys symbols are set as, type:

```
$ sh sym wake*
```

VMS Displays: (VMS sorts them alphabetically when displaying)

```
WAKE_CALIBRATION_FILE == "WKS"
WAKE_CARRIAGE_SPEED_CHAN == "1"
WAKE_HEAVE_CHAN == "2"
WAKE_NUMBER_OF_CHANNELS == "11"
WAKE_OUTPUT_FILE == "WKS_VS14_FIXED"
WAKE_PITCH_CHAN == "3"
WAKE_PRINTER == "LASER5"
WAKE_PROBE1_PRES_CHAN == "7"
WAKE_PROBE2_PRES_CHAN == "8"
WAKE_PROBE3_PRES_CHAN == "9"
WAKE_PROBE4_PRES_CHAN == "10"
WAKE_PROBE5_PRES_CHAN == "11"
WAKE_ROLL_CHAN == "4"
WAKE_SOURCE_LOCATION == "TOW"
WAKE_STORAGE_DIRECTORY == "[PJ94617.WAKE_TESTS]"
WAKE_TEST_NAME == "617_WAKE2"
WAKE_TOTAL_CHANNELS == "1-11"
WAKE_XPOS_CHAN == "5"
WAKE_YPOS_CHAN == "6"
```

Wake Survey Settings:

The global variables to be set are defined below. Similar descriptions are contained as comments before each global setting in the actual `WAKE_SET.COM` file.

```
$ WAKE_SOURCE_LOCATION :==
```

This variable points to the permanent source location of data files for the tests. This location can either be one of the logical names for the standard storage devices (`TOW`, `ICE` etc) or the data could be directly taken from the data acquisition machine, currently `VIOLET` on the tow carriage.

```
$!
$!------------------------------------------------------------------------
$! WAKE_SOURCE_LOCATION :==
$!    Sets the location of the raw source data time series files.
$!    Possible locations include TOW and ICE
$!    Example:     WAKE_SOURCE_LOCATION :== TOW
$!------------------------------------------------------------------------
$ WAKE_SOURCE_LOCATION :== TOW
$!
```

$ WAKE_TEST_NAME :==

This is the test name of the permanent raw data files for the current test you wish to analyze.

```
$!
$!------------------------------------------------------------------------
$! WAKE_TEST_NAME :==
$!    Sets the test name of the raw data time series files.
$!    Example:     WAKE_TEST_NAME :== WAKE_TEST1
$!------------------------------------------------------------------------
$ WAKE_TEST_NAME :== 617_WAKE2
$!
```

$ WAKE_STORAGE_DIRECTORY :==

This is the name of the VMS directory where the temporary raw data files will be stored. Usually, it should be in the `[.RAW]` subdirectory of the directory you are working in:

```
$!
$!------------------------------------------------------------------------
$! WAKE_STORAGE_DIRECTORY :==
$!Sets the storage directory where new GEDAP raw data files will be stored.
$!Include the logical disk name if the storage Directory is on a disk
$!other than the one you are running these programs from.
$!Example:     WAKE_STORAGE_DIRECTORY :== [PJ94617.WAKE_TESTS]
$!------------------------------------------------------------------------
$ WAKE_STORAGE_DIRECTORY :== [PJ94617.WAKE_TESTS.RAW]
$!
```

$ WAKE_PRINTER :==

Determines the printer where any plots will be sent. If the printer is specified as `NONE` then no prints will be made. Usually, the raw channel data plots are produced with either `WAKE_RUN` and `WAKE_DEPTH_CHECK`.

```
$!
$!------------------------------------------------------------------------
$! WAKE_PRINTER :==
$!    Sets the printer name that any plots will be plotted on.
```

```
$!
$!    *** NOTE:  Set WAKE_PRINTER :== NONE  for no printing ***
$!
$!    Example:      WAKE_PRINTER :== LASER5
$!------------------------------------------------------------------------
$ WAKE_PRINTER :== NONE
$!
```

$ WAKE_OUTPUT_FILE :==

Determines the root file name for all output file from the wake survey programs. For example, if `WAKE_OUTPUT_FILE :== WKS_VS14_FIXED` was the output file name, then the stats and log files would be named:

```
WKS_VS14_FIXED_STATS.DAT        ! Run statistics file
WKS_VS14_FIXED_LOG.DAT          ! Log file
```

It is different from the `WAKE_TEST_NAME`, the name of the raw data files only.

```
$!
$!------------------------------------------------------------------------
$! WAKE_OUTPUT_FILE :==
$!    Sets the main file name for the output data files.
$!
$!    Example: WAKE_OUTPUT_FILE :== WAKE_VS12_FIXED
$!
$!    Will produce output files: WAKE_VS12_FIXED_STATS.DAT  <-- Stats file
$!                               WAKE_VS12_FIXED_LOG.DAT  <-- Log file
$!                               WAKE_VS12_FIXED_DEPTH_STATS.DAT
$!                                            ^ Zero tare depth check stats
$!                               etc...
$!
$!    *** Warning! ***
$!
$!    The filename cannot start with a number. ie. :== 617_WKS is invalid
$!    but :== P617_WKS is valid.
$!
$!------------------------------------------------------------------------
$ WAKE_OUTPUT_FILE :== WAKE_TEST
$!
```

▼ **IMPORTANT:** All of the wake survey programs operate using this output file name as the input and output root name.

$ WAKE_CALIBRATION_FILE

Sets the root name and location of the calibration files. A different set of calibration files will be produced for every probe head. The calibration files are always stored in the `WKS:[TABLES]` directory. For example, if the C2 probe was being used, then the `WAKE_CALIBRATION_FILE` would be set to `WKS:[TABLES]WKS_C2`. This would correspond to using the calibration files:

```
WKS:[TABLES]WKS_C2_Q.001        ! Q parameter calibration file
WKS:[TABLES]WKS_C2_R.001        ! R parameter calibration file
WKS:[TABLES]WKS_C2_PCAL.001     ! Pcal parameter calibration file
```

```
$!
$!-----------------------------------------------------------------------
$! WAKE_CALIBRATION_FILE :==
$! Sets the ROOT name of the three GEDAP wake calibration files containing
$!    the parameters Q, R and PCAL.  The three files will be:
$!                 'WAKE_CALIBRATION_FILE'_Q.001
$!                 'WAKE_CALIBRATION_FILE'_R.001
$!                 'WAKE_CALIBRATION_FILE'_PCAL.001
$!
$!   **** NOTE! ****
$!
$!   The wake survey calibration files are contained in the [TABLES]
$!   directory of the logical device WKS.  To access these tables the
$!   the calibration files must be proceeded by WKS:[TABLES]. For example,
$!   if the ROOT file name for the calibration tables is WKS_C2 then the
$!   ROOT file name WAKE_CALIBRATION_FILE should be set as:
$!
$!              WAKE_CALIBRATION_FILE :== WKS:[TABLES]WKS_C2
$!
$!   Naming of Calibration Files:
$!
$!   The calibration files are named according to the probe serial number
$!   For example, WKS_C2 indicates the calibration files for the probe C2
$!
$!   Example:  WAKE_CALIBRATION_FILE :== WKS:[TABLES]WKS_C2
$!
$!   Uses cal files:     WKS:[TABLES]WKS_C2_Q.001
$!                       WKS:[TABLES]WKS_C2_R.001
$!                       WKS:[TABLES]WKS_C2_PCAL.001
$!
$!-----------------------------------------------------------------------
$ WAKE_CALIBRATION_FILE :== WKS:[TABLES]WKS_C3
$!
```

WAKE_*XXX*_CHAN :==

Sets the data stream numbers for each data channel. The settings must not have any leading zeros. For example, `5` should be used and NOT `005`. These are data stream numbers and NOT neff channel numbers.

```
$!
$!-----------------------------------------------------------------------
$! WAKE_CARRIAGE_SPEED_CHAN :==
$! WAKE_PROBE1_PRES_CHAN :==
$! WAKE_PROBE2_PRES_CHAN :==
$! WAKE_PROBE3_PRES_CHAN :==
$! WAKE_PROBE4_PRES_CHAN :==
$! WAKE_PROBE5_PRES_CHAN :==
$! WAKE_XPOS_CHAN :==
```

```
$! WAKE_YPOS_CHAN :==
$! WAKE_PITCH_CHAN :==
$! WAKE_HEAVE_CHAN :==
$! WAKE_ROLL_CHAN :==
$!    Tells the program what data stream correspond to the various
$!    measured parameters.  The parameter must not have any leading
$!    zeros.  For example, channel 5 would be 5 and not 005
$!
$!    **** WARNING! ****
$!
$!    These are DAS or GEDAP data stream numbers not NEFF Channel numbers!
$!
$!    Example:    WAKE_CARRIAGE_SPEED_CHAN :== 1    <--Set to Channel 1
$!                WAKE_PROBE1_PRES_CHAN :== 3       <--Set to Channel 3
$!                WAKE_PROBE2_PRES_CHAN :== 4       <--Set to Channel 4
$!                WAKE_PROBE3_PRES_CHAN :== 5       <--Set to Channel 5
$!                WAKE_PROBE4_PRES_CHAN :== 6       <--Set to Channel 6
$!                WAKE_PROBE5_PRES_CHAN :== 7       <--Set to Channel 7
$!                WAKE_XPOS_CHAN :== 8              <--Set to Channel 8
$!                WAKE_YPOS_CHAN :== 9              <--Set to Channel 9
$!                WAKE_PITCH_CHAN :== 10            <--Set to Channel 10
$!                WAKE_HEAVE_CHAN :== 11            <--Set to Channel 11
$!                WAKE_ROLL_CHAN :== 12             <--Set to Channel 12
$!------------------------------------------------------------------------
$ WAKE_CARRIAGE_SPEED_CHAN :== 1
$ WAKE_PROBE1_PRES_CHAN :== 7
$ WAKE_PROBE2_PRES_CHAN :== 8
$ WAKE_PROBE3_PRES_CHAN :== 9
$ WAKE_PROBE4_PRES_CHAN :== 10
$ WAKE_PROBE5_PRES_CHAN :== 11
$ WAKE_XPOS_CHAN :== 5
$ WAKE_YPOS_CHAN :== 6
$ WAKE_PITCH_CHAN :== 3
$ WAKE_HEAVE_CHAN :== 2
$ WAKE_ROLL_CHAN :== 4
$!
```

WAKE_TOTAL_CHANNELS :==

Specifies exactly which data streams are being used. For example, certain streams can be skipped. The data streams can defined using the following notation:

`1,3-7, 11, 13 ,15-17` means channels `1,3,4,5,6,7,11,13,15,16,17`

```
$!
$!------------------------------------------------------------------------
$! WAKE_TOTAL_CHANNELS :==
$!    Tells the program which data streams are being used.
$!
$!    Define the data streams as follows:
$!    If using data streams 1,2,4,5,6,8 then use 1-2,4-6,7
$!    If using data streams 1,2,3,4,5,6,7 then use 1-7
$!    Example: WAKE_TOTAL_CHANNELS :== 1,3-7   <--Process 1,3,4,5,6,7
$!------------------------------------------------------------------------
```

```
$ WAKE_TOTAL_CHANNELS :== 1-11
$!
```

WAKE_NUMBER_OF_CHANNELS :==

This is the total number of channels being used.

```
$!
$!-----------------------------------------------------------------------
$! WAKE_NUMBER_OF_CHANNELS :==
$!    Tells the program how many data streams there are in total.
$!    Example: WAKE_NUMBER_OF_CHANNELS :== 7   <--There are 7 channels
$!-----------------------------------------------------------------------
$ WAKE_NUMBER_OF_CHANNELS :== 11
$!
```

Warnings:

Any changes in the test settings, such as the output file name, have to made be made in `WAKE_SET.COM`. Further, every time you are starting a new VMS session or DECTerm Window, you must run `WAKE_SET.COM` after running `WAKE_START` to make the wake survey settings active. When you log off, the symbol settings will be erased requiring you to run (but not edit, if there are no changes required) `WAKE_SET` the next time you log on.

Log File:

`WAKE_SET.COM` updates the log file with the name `$ @WAKE_SET` and also adds comments indicating what the settings were when it was run. An example log entry would be:

```
$ @WAKE_SET
$!****************SETTINGS*********************
$!  WAKE_SOURCE_LOCATION:== TOW
$!  WAKE_TEST_NAME :== 617_WAKE2
$!  WAKE_STORAGE_DIRECTORY :== [PJ94617.WAKE_TESTS]
$!  WAKE_PRINTER :== LASER5
$!  WAKE_OUTPUT_FILE :== WKS_VS14_FIXED
$!  WAKE_CALIBRATION_FILE :== WKS
$!  WAKE_CARRIAGE_SPEED_CHAN :== 1
$!  WAKE_PROBE1_PRES_CHAN :== 7
$!  WAKE_PROBE2_PRES_CHAN :== 8
$!  WAKE_PROBE3_PRES_CHAN :== 9
$!  WAKE_PROBE4_PRES_CHAN :== 10
$!  WAKE_PROBE5_PRES_CHAN :== 11
$!  WAKE_XPOS_CHAN :== 5
$!  WAKE_YPOS_CHAN :== 6
$!  WAKE_PITCH_CHAN :== 3
$!  WAKE_HEAVE_CHAN :== 2
$!  WAKE_ROLL_CHAN :== 4
$!  WAKE_TOTAL_CHANNELS :== 1-11
$!  WAKE_NUMBER_OF_CHANNELS :== 11
$!********************************************
```

Input Files:

None

Output Files:

None

WAKE_CONVERT

Usage:

```
$ wake_convert P1 P2
```

Where: `P1` = the first run to convert
`P2` = the last run to convert

Description:

This command procedure converts raw data DAS files to GEDAP files and stores them in sub-directories in the temporary storage file, defined by `WAKE_STORAGE_DIRECTORY` in `WAKE_SET.COM`.

Operation:

This command convert multiple runs. It uses the GEDAP program `DAS_GEDAP` to perform all of the conversions. The output file names and directory structure is defined in section **3.0.1**.

Warnings:

When taking data directly from the data acquisition computer, such as VIOLET the number of network buffers must be set as follows:

```
$ set rms_default/network_buffer_count = 30    ! Set the buffer count
```

This command must be typed at the beginning of every session. It is not necessary when taking data from the TOW disk.

Log File:

This program will create a log entry similar to the one below:

```
$ WAKE_CONVERT 675 677
$ WAKE_DASGEDAP 675
$ WAKE_DASGEDAP 676
$ WAKE_DASGEDAP 677
```

`WAKE_CONVERT 675 677` is the command issued. The `WAKE_DASGEDAP` commands signify that the command `WAKE_DASGEDAP` was used for each of the runs in the specified series. `WAKE_CONVERT` uses `WAKE_DASGEDAP` for every run to be converted.

Input Files:

The input files would be the DAS raw data storage files located on either the TOW disk or on the data acquisition computer such as VIOLET.

Output Files:

Creates the following output for every run that was converted:

If the raw data test name was WAKE_TEST2 and the run number was 234 then the files created would be:

Directory: [.WAKE_TEST2_R234]

Files in the Directory:

```
WAKE_TEST2_R234.001
WAKE_TEST2_R234.002
 .  .  .  .  .  .
WAKE_TEST2_R234.011
```

These output files would be raw data GEDAP files.

WAKE_DEPTH_CHECK

Usage:

```
$ wake_depth_check P1 P2
```

Where: `P1` = Run number of the depth check run
`P2` = Overall depth range of the depth check in cm

Description:

Analyzes a probe depth check test and computes the probe pressure errors for each probe. The overall depth is the range of the depth check from its highest position to its lowest. This is used in calculating the overall percent error in the pressure readings compared to the probe depth.

Usage:

This command will start by making the user select the proper time segments for each depth. This data is analyzed and the results saved to an output file and printed to the screen. The first segment selected is used as the zero tare segment. The results appear in the following form:

```
Range used for calculating % errors (spec by user)=250 cm
DEPTH CHECK STATISTICS FOR EACH SEGMENT:

Chan:              Mean:  Press Dif:       %Err:    Std Dev:
-------------------------------------------------------------
Depth            15.0151       * NA *     * NA *      0.0028
P1  (Top)        15.0898       0.0747        0.0      0.0170
P2  (Stbd)       15.1363       0.1212        0.0      0.0149
P3  (Bot)        15.2134       0.1983        1.0      0.0170
P4  (Port)       15.5214       0.5063        2.0      0.0137
P5  (Cent)       15.2169       0.2017        1.0      0.0158
```

The first column is the channel description. The second column is the mean reading of that channel. The third column is the pressure difference between each probe and the depth (in centimetres). The fourth column is the percent error in the pressure difference (column 3) over the pressure range, specified by the user. Column five is the standard deviation for each channel. There will one of these above entries for each segment in the depth check. If the errors were too high, a re-calibration of the probes would be required.

Warnings:

This command tares all data from the first segment selected so all depths and pressures are relative to this first segment.

Log File:

A line similar to the one that follows will be added to the LOG file:

```
$ @WAKE_DEPTH_CHECK 901
$! Output File        :WKS_TEST_DEPTH_CHECK_R901.DAT
$! Depth Check Range :250
```

Input Files:

The raw data GEDAP files for the depth check run.

Output File:

An ASCII text file will be created that records the depth check statistics for each segment. It will be named according to the root output file name with the addition of `_DEPTH_CHECK_R`*runnumber*`.DAT`. For example, if the output file name, `WAKE_OUTPUT_FILE` was set to `WKS_TEST` and the run number of the depth check was 267, then the ASCII output file would be:

```
WKS_TEST_DEPTH_CHECK_R267.DAT
```

WAKE_RUN

Usage:

```
$ wake_run P1 P2        ! Perform wake survey analysis on runs P1 to P2
```

Where: P1 = First GEDAP /DAS run number to analyze
Where: P2 = Last GEDAP/DAS run number to analyze

Description:

Performs analysis on the wake survey data for the runs specified by the user. The user selects the appropriate time segments, the data is tared, the velocity statistics are computed and check plots are produced at the end of every run.

Operation:

When `WAKE_RUN` first starts, it will copy the GEDAP raw data files from the storage directory for the run being analyzed to a temporary processing directory named `[.PROCESS]` which will be created in the directory where you are running your program. Then it will start `INTERACTIVE_SEGMENT_SELECT.COM` which allows the user to select the time segments of interest. When `ISS.COM` is started, it will create another window and plot up the data for carriage speed, x position, the top wake probe pressure and the port wave probe pressure. In the original window, the GEDAP program `GPLOT` will be running which displays the prompt `GPLOT>`. Click on this window to make it active. Then type `SEL` at the prompt to select a time segment.

```
GPLOT> sel              ! Select a time segment
```

After you press enter, move the mouse to the graphics window displaying the plotted data. A set of cross hairs should now follow the mouse. Clicking the mouse button on a position will select the beginning of the time segment. This marker will be drawn in on the bottom plot. Clicking the button on another position selects the end time for the time segment. Again, this marker is displayed on the bottom plot. The first time segment is now selected. *The first time segment you select has to be the zero tare segment. That is, select the time segment for the model in still water before the run (ie. carriage speed = 0). The command will flag an error if the carriage speed of the first segment is not approximately equal to zero.* Once the zero tare segment is selected, you may select all of the other time segments by again typing `SEL` at the prompt (or pressing up-arrow) and selecting the segment ends. When the time segments have been selected, type `EXIT` at the `GPLOT>` prompt.

```
GPLOT> exit             ! Finished selecting segments
```

Once the time segments have been selected, `WAKE_RUN` will split the original GEDAP data files into separate data files for each segment. Thus, if 5 segments were selected (including the first segment, which is the zero tare segment), then 5 times the number of data streams (or channels) new files will be created. If there are 11 data streams, then 5*11 = 55 new data file will be created. If the raw GEDAP files were called:

```
WAKE_TEST_R123.001          ! Original GEDAP files for channels 1 to 11
WAKE_TEST_R123.002
      . . . . . . .
WAKE_TEST_R123.011
```

Then WAKE.COM will create new files:

```
WAKE_TEST_R123_S1.001       ! New GEDAP files for segment 1 (S1 = segment 1)
WAKE_TEST_R123_S1.002
      . . . . . . .
WAKE_TEST_R123_S1.011
WAKE_TEST_R123_S2.001       ! New GEDAP files for segment 2 (S2 = segment 2)
WAKE_TEST_R123_S2.002
      . . . . . . .
WAKE_TEST_R123_S2.011
      . . . . . . .
      . . . . . . .
WAKE_TEST_R123_S5.001       ! New GEDAP files for segment 5 (S5 = segment 5)
WAKE_TEST_R123_S5.002
      . . . . . . .
WAKE_TEST_R123_S5.011
```

WAKE_RUN will then calculate the mean value of the first segment (the zero tare segment) for each data stream and subtract these values from all of the other segments. Thus, the segment data files will contain the time series data for each segment and they will also be tared. These tared segment data files are then used to calculate the velocity parameters. The results will be automatically stored in the STATS file.

The command will also check the probe pressure against depth for every zero tare segment. This is a way to check the pressure transducer performance and calibration before each run. This information will be contained in a file with the name _DEPTH_CHECKS.DAT as in WKS_TEST_DEPTH_CHECKS.DAT.

Finally, the velocity versus angle theta for the various radii will plotted to the screen. This allows the user to monitor the progress of the test.

WAKE_RUN is called at the VMS command prompt as explained above. If the parameters P1 and P2 are not entered, the program will prompt for them. Entering a zero or a blank line for P2 will cause the program to prompt the user with 'Do you want to analyze another run?' after the completion of every run.

To analyze one run with WAKE_RUN enter the same number for both P1 and P2:

```
$ wake_run 234 234    ! Analyze run 234
```

Warnings:

When selecting time segments be sure to always select the zero tare segment (for zero carriage speed) first. Failing to do this will cause severe errors that may make the STATS file invalid. Once the STATS file becomes invalid, all analysis of the ENTIRE TEST must be started again by using `WAKE_RESET.COM` to erase the STATS file. (`WAKE_REAN_RUN` can be used to analyze all runs up to the erroneous run without having to re-select the time segments) The reason for this is that any data computed using segments with the wrong zero tare segment will be erroneous but it will not crash the program. Therefore, the erroneous values will be stored in the STATS file. The only way to get rid of these values is to reset the wake survey analysis (ie. erase the STATS file and start over) or to edit the STATS file and erase the erroneous data.

Log File:

`WAKE_RUN` will produce the following line in the LOG file:

```
$ @WAKE_RUN P1 P2          ! Runs P1 to P2
```

For example, if `P1` = 123 and `P2` = 131 then the log file would read:

```
$ @WAKE_RUN 123 131
```

Further, because `WAKE_RUN` uses two sub-commands called `WAKE` and `WAKE_CHECK_ZERO_DEPTH`. These lines will appear in the LOG file for every run. The `WAKE_CHECK_ZERO_DEPTH` appears first because it is the first once checked. The WAKE command appears after and generally indicates that the run analysis completed successfully.

A typical log file would appear as:

```
$ @WAKE_RUN 241 242                              ! Analyze runs 241 to 242
$ @WAKE_CHECK_ZERO_DEPTH 241                     ! Zero depth check on run 241
$! Output File       :WKS_TEST_DEPTH_CHECKS.DAT
$! Depth Check Range :30
$! Segment to Check  :1
$ @WAKE 241                                      ! Finished analysis of run 241
$ @WAKE_CHECK_ZERO_DEPTH 242                     ! Zero depth check on run 242
$! Output File       :WKS_TEST_DEPTH_CHECKS.DAT
$! Depth Check Range :30
$! Segment to Check  :1
$ @WAKE 242                                      ! Finished analysis of run 242
```

Input File:

The GEDAP raw data file

Output File:

Raw GEDAP files for each selected segments (see above).
The _STATS.DAT file will be updated with the velocity statistics for the analyzed run segments.

WAKE_LOG, WAKE_STATS and WAKE_DEPTH_STATS

Usage:

```
$ wake_log                ! Display the LOG file
$ wake_stats              ! Display the STATS file
$ wake_depth_stats        ! Display the zero tare depth check file
```

Description:

These three commands are used to list the LOG file, the STATS file and the zero tare depth check file.

Operations:

WAKE_LOG: Displays the entire LOG file to the screen.

WAKE_STATS: Will display the statistics for any run or all of the runs in the STATS file. The user has the choice to display the velocity stats only or combinations of the computed parameters, velocity stats and raw channel statistics.

WAKE_DEPTH_STATS:
Displays the entire zero tare depth check file to the screen.

Warnings:

If any of the above files do not exist, the programs cannot display the results and will inform the user.

Log File:

These procedures are not recorded in the LOG file.

Input Files:

The LOG file, the STATS file and the zero tare depth check file.

Output Files:

None.

WAKE_PLOT_STATS

Usage:

`$ wake_plot_stats P1 P2 P3 P4` ! Plot the current velocity statistics

Where:
`P1` = Allowable absolute deviation in the radii during sorting
`P2` = Theta range: '1' for -180 to +180 or '2' for 0 to +360 degrees
`P3` = Make end points the same value, 'Y' or 'N'
`P4` = Is the data symmetrical, 'Y' or 'N'

Description:

This command sorts the current velocity statistics in the STATS file by radius and plots the V_x, V_t and V_r velocities versus the corresponding angle theta.

Procedure:

`WAKE_PLOT_STATS` will automatically sort the runs and plot them in `GPLOT`. The parameters are explained below:

P1 Allowable absolute deviation. When the program sorts the data into different groups by radii there can often be small differences in the radii. The deviation determines how big these discrepancies can be before another group is formed. For example if one radius was 87.5 mm and the deviation was 5 (as in +/- 5 mm) then any radii between 87.5 + 5 = 92.5 and 87.5 - 5 = 82.5 would be included in this group. Any radii outside this range will be moved to another group.

P2 Plot range. `WAKE_PLOT_STATS` can plot the data over two ranges of theta: -180 to +180 degrees and 0 to 360 degrees. These tow ranges allow flexibility in the displaying the information. For example, data that changes rapidly near 0 degrees and stays steady near 180 and -180 degrees the range should be -180 to +180 degrees. This will put the interesting part of the plot in the middle of the graph rather than splitting it at the ends.

P3 Make end points the same. The wake survey data wraps around because it is radially oriented data. Therefore, the end points at -180 and +180 degrees have to have the same value because they are the same physical point. This option makes the two end point values equal to each other.

P4 Is the data symmetrical. For single screw models wake surveys may be performed on only one side and then mirror to the other. This option tells the plotting routine wether the data is symmetrical. If the data is symmetrical, the program will mirror it about the theta = 0 axis.

Warnings:

If only a few runs have been analyzed, the data may appear distorted because only a few points have been analyzed. The data will be automatically scaled as it is plotted. Therefore, what may seem like large changes my actually appear quite small after all of the data has been collected.

Log File:

This command does not appear in the LOG file.

Input File:

Data from the STATS file is displayed by this command.

Output File:

None. Only some temporary GEDAP files are created which are later deleted by this command.

WAKE_RESET

Usage:

```
$ wake_reset            ! Reset the wake survey analysis of the current test
```

Description:

This procedure erases the LOG, STATS and zero tare depth check files. It is useful for starting a wake survey over again. It is also useful for preparing to use WAKE_REAN_RUN which should really only be started with blank STATS file.

Operation:

Starting this command will display a warning to the user:

```
This command procedure will copy the _LOG.DAT, _STATS.DAT and
_DEPTH_CHECKS.DAT file for the current output file name to a new set of
files with the _LOG_OLD.DAT, _STATS_OLD.DAT and _DEPTH_CHECKS_OLD.DAT
extensions added.  It will then delete the existing files, if requested.

**** Warning! ****

Use this command program ONLY if you wish to start over in the analysis of
the current test name.

Do you wish to continue with the operation? [N]:
```

Answering 'N' to this question will exit the command and change nothing. Answering 'Y' will make the command copy the LOG, STATS and zero depth check files to back-up files called _LOG_OLD.DAT, _STATS_OLD.DAT and _DEPTH_CHECKS_OLD.DAT. The computer will then ask if you want to delete the current files one by one (so you can select which files you want to delete).

Warnings:

Care should be taken when erasing these files, but generally the back-up copies make this command fairly safe.

Log File:

The following is an example LOG file entry:

```
$ @WAKE_RESET
$! WKS_TEST_LOG.DAT copied to WKS_TEST_LOG_OLD.DAT
$! WKS_TEST_STATS.DAT copied to WKS_TEST_STATS_OLD.DAT
$! WKS_TEST_DEPTH_CHECKS.DAT copied to WKS_TEST_DEPTH_CHECKS_OLD.DAT
$! WKS_TEST_LOG.DAT was deleted
$! WKS_TEST_STATS.DAT was deleted
$! WKS_TEST_DEPTH_CHECK.DAT was deleted
```

Input and Output Files: None.

WAKE_DELETE_RUNS

Usage:

`$ wake_delete_runs` ! Delete a series of stored GEDAP files

Description:

This command will delete the raw data GEDAP files from the temporary storage directory and then delete the directory. This command should only be used when the analysis is complete and all of the data manipulation has been performed including curve faring and plotting. Once the runs have been deleted they would need to be re-copied to the storage directories and the time segments would have to be selected again.

Operation:

When this command is first used it will ask you to confirm that you want to continue with the operation:

```
This command procedure will delete any series of runs from the current
DAS test name.  This command should ONLY be invoked AFTER the data from
runs has been analyzed and faired.  Once the runs have been deleted, they
would need to be converted from storage in the DAS form to GEDAP form and
all time segments would need to be re-selected. ie. You start over again.

Current Output File Name :== WKS_TEST

Do you wish to continue with the operation? [N]
```

The current output file name is the root file name defined by `WAKE_OUTPUT_FILE` in the `WAKE_SET.COM` file. Answering 'Y' to this question will begin the run deletion utility. Any other answer will exit the command and no changes will be made.

The run deletion utility will then ask for the first run to delete and the last run to delete. Note that it will delete all of the runs in between the end runs as well.

Warnings:

This file will delete the temporary GEDAP raw data files. Once they are deleted the time segments will have to be picked again if any re-analysis is required. `WAKE_DELETE_RUNS` should only be used after the data has been analyzed, faired and plotted.

Log File:

Example Log File:

```
$!**********************************************
$ @WAKE_DELETE_RUNS 245 246
$!**********************************************
```

```
$! All files in [PJ94617.TEST.617_WAKE2_R245] were deleted
$! Directory [PJ94617.TEST.617_WAKE2_R245] was deleted
$! All files in [PJ94617.TEST.617_WAKE2_R246] were deleted
$! Directory [PJ94617.TEST.617_WAKE2_R246] was deleted
$!*******************************************
$! All Files Deleted Successfully
$!*******************************************
```

There will be two entry lines for every run that was deleted.

Input and Output Files:

None

WAKE_SORT_RADII

Usage:

`$ wake_sort_radii` ! Sort the velocities by radius from centre line

Description:

This command takes the velocity statistics from the STATS file for the current analysis test name and sorts the data into groups of radii from the propeller centre line. These sorted runs are then output to a single file or a separate file for each radii. The separate file option will create data files that can be imported into `WKS_FAIR` in Matlab for curve fairing.

Operation:

When it is started, the following prompts will be displayed, in order:

```
Would you like to output the sorted radii to separate files? [Y]:
```

The sorted radii can either be grouped in one ASCII file or grouped into separate files for each radius. Separate files are used when the data is to be output for use with `WKS_FAIR` in Matlab for curve fairing and plotting. If the output is to one file, the program will prompt you for an output file name.

```
Allowable absolute deviation in radii [5]:
```

When the program sorts the data into different groups by radii there can often be small differences in the radii. The deviation determines how big these discrepancies can be before another group is formed. For example if one radius was 87.5 mm and the deviation was 5 (as in +/- 5 mm) then any radii between 87.5 + 5 = 92.5 and 87.5 - 5 = 82.5 would be included in this group. Any radii outside this range will be moved to another group.

```
Theta Range: '1' for -180 to +180 degrees '2' for 0 to +360 degrees [1]:
```

WAKE_SORT_RADII can plot the data over two ranges of theta: -180 to +180 degrees and 0 to 360 degrees. These two ranges allow flexibility in the displaying the information. For example, data that changes rapidly near 0 degrees and stays steady near 180 and -180 degrees the range should be -180 to +180 degrees. This will put the interesting part of the plot in the middle of the graph rather than splitting it at the ends.

```
Make end points the same value (Y/N) [YES]:
```

Make end points the same. The wake survey data wraps around because it is radially oriented data. Therefore, the end points at -180 and +180 degrees have to have the same value because they are the same physical point. This option makes the two end point values equal to each other.

```
Is the input data symmetrical (Y/N) [NO]:
```

For single screw models wake surveys may be performed on only one side and then mirror to the other. This option tells the sorting routine wether the data is symmetrical. If the data is symmetrical, the program will mirror it about the theta = 0 axis.

Warnings:

Do not set the allowable deviation too low because the program may create more groups than there actually were. The number of output files should also b the same as the number of radii that were used for the test.

Log File:

Sample Log File Listing:

```
$ @WAKE_SORT_RADII
$! Input STATS File              : WKS_TEST_STATS.DAT
$! Allowable Absolute Deviation       5.000
$! Make Separate Files (Y/N)     : YES
$! Range of Theta : -180 to +180 degrees
$! Make end points the same (Y/N) : YES
$! Is the data symmetrical (Y/N) : NO
$! Output File: WKS_TEST_RAD1.DAT
$! Output File: WKS_TEST_RAD2.DAT
$! Output File: WKS_TEST_RAD3.DAT
$! Output File: WKS_TEST_RAD4.DAT
$! Output File: WKS_TEST_RAD5.DAT
$! Output File: WKS_TEST_RAD6.DAT
```

Input Files:

The STATS file for the current test.

Output Files:

For the separate files option the output files will have the extension `_RADn.DAT` added to the root output name. For example, if the root name was `WAKE_OUTPUT_FILE :== WKS_TEST` then the output files would be:

```
WKS_TEST_RAD1.DAT        ! Sorted RADII file for Radius 1, the largest radius
WKS_TEST_RAD1.DAT        ! Sorted RADII file for Radius 2
WKS_TEST_RAD1.DAT        ! Sorted RADII file for Radius 3
   .  .  .  .  .
WKS_TEST_RADn.DAT        ! Sorted RADII file for Radius n, smallest radius
```

The files are named such that radius 1 is the largest radius and radius *n* is the smallest radius. These output files can then be used with Matlab and `WKS_FAIR`. Each of these files is a simple ASCII file with the data in columns according to the following format:

```
Radius        Theta (Degrees)         Vx/Vs        Vt/Vs        Vr/Vs
```

For the case of single file output, all of the data will be output into one file which can be named by the user. It will have the same format as the separate files, but all of the radii will be grouped in this one file separate by spaces between each group.

WAKE_SPREAD

Usage:

```
$ wake_spread          ! Output the statistics to an ASCII file for spreadsheets
```

Description:

WAKE_SPREAD will output all of the computed statistics to an ASCII file that can be imported into a computer spreadsheet program such as Excel.

Operation:

WAKE_SPREAD will ask for an output filename for the spread sheet file. The default is the root output name, the WAKE_OUTPUT_FILE symbol, plus _STATS_SPREAD.DAT. For example, WKS_TEST_STATS_SPREAD.DAT. To import this file into Excel simply open the file using Excel's *Open* command and then answer the on-screen prompts. The data is delimited by spaces.

Warnings:

None

Log File:

The log file will contain the command name @WAKE_SPREAD and then the name of the output spreadsheet file.

Sample log file listing:

```
$ @WAKE_SPREAD WKS_TEST_STATS_SPREAD.DAT   ! Make spreadsheet file called
                                             WKS_TEST_STATS_SPREAD
```

Input File:

The STATS file.

Output File:

ASCII text file for import into spreadsheet. Usually with the extension _STATS_SPREAD.DAT.

WAKE_REAN_RUN

Usage:

```
$ wake_rean_run P1 P2          ! Re-analyze runs P1 to P2
```

Where: `P1` = First run to re-analyze
`P2` = Last run to be re-analyzed

Description:

Will re-calculate the velocities for the tests using the time segments previously selected. This procedure is particularly useful when the STATS file becomes corrupted or when the flow calculation program and/or calibration files are changed. It can also be useful when testing changes to the calculation program by automatically analyzing a given set of data repeatedly using the exact same time segments.

Operation:

Before using `WAKE_REAN_RUN`, the STATS file should be erased using `WAKE_RESET`. This will make a backup copy of the old STATS file. `WAKE_REAN_RUN` can then be started. It uses the previously selected time segments, does not perform plots of the original data to either the printer or the screen. It also does not produce check plots.

Warnings:

If any segments were selected incorrectly, `WAKE_REAN_RUN` will not be able to fix the results. Such segments should be re-analyzed using the standard `WAKE_RUN` command and re-selecting the segments. Then the STATS file should be erased and `WAKE_REAN_RUN` can be used to re-analyze of the chosen runs using the select time segments.

Log File:

Will add the WAKE_REAN_RUN P1 P2 line to the file. It will also add a WAKE_REAN line for each run indicating that each run was reanalyzed successfully.

WAKE_RAD_TO_DOS

Usage:

$ wake_rad_to_dos ! Convert sorted radii file names from WAKE_SORT_RADII to shorter names for use with MS-DOS.

Description:

Takes the separate output files from WAKE_SORT_GEDAP and copies them to new file with shorter, 8 character names suitable for use with MS-DOS.

Operation:

When first run, the program will prompt for the root name of the files from WAKE_SORT_RADII. For example, the files WKS_TEST_RAD1.DAT to WKS_TEST_RAD*n*.DAT have the root name WKS_TEST. Generally, this name is the root output name defined by WAKE_OUTPUT_FILE in WAKE_SET.COM. This name appears as the default entry.

The program will then ask for the output root name of the MS-DOS files. This name should be 6 or less characters for 9 radii files or 5 or less for 10 or more radii files. The output files will be named according to the following example:

MS-DOS Root Name = VS12FX (***VS=12*** knots, ***F****i****X***ed)

Output Files:	VS12FXR1.DAT	! Radius 1 file
	VS12FXR2.DAT	! Radius 2 file
	. . . . .	
	VS12FXR*n*.DAT	! Radius *n* file

Warnings:

None

Log File:

Example Log File Entry:

```
$ @WAKE_RAD_TO_DOS
$! WAKE_SORT_RADII Output Root Name   :  WKS_TEST
$! MS-Dos Root Name                   :  VS14FX
$! Number of Sorted Radii Files       :  5
$! WKS_TEST_RAD1 Copied to WKS14R1.DAT
$! WKS_TEST_RAD2 Copied to WKS14R2.DAT
$! WKS_TEST_RAD3 Copied to WKS14R3.DAT
$! WKS_TEST_RAD4 Copied to WKS14R4.DAT
```

```
$! WKS_TEST_RAD5 Copied to WKS14R5.DAT
```

Input Files:

Sorted radii files from WAKE_SORT_RADII.DAT

Output Files:

Copies of the sorted radii files with shorter names suitable for use with MS-DOS.

3.3 Wake Sweep Analysis

Wake sweep data is analyzed using the following commands:

`$ wake_sweep p1`	! Will analyse wake sweep run number P1

Select the zero tare segment and then the sweep segement. Then use:

`$ wake_sweep_smooth p1`	! Smooth the wake sweep file. The output file can be imported into `WKS_FAIR` in Matlab

4.0

Probe Positioning Software

Topics:

- Unidex-14 operation
- Using WAKE.BAS and QuickBASIC 4.5
- WAKE.BAS configuration file set-up
- .RUN file description and use
- Generating and executing .RUN files
- Controlling the position stage
- Calibrating and homing the position stage

The wake survey positioning stage is based on an Aerotech Unidex-14 motor control system. The Unidex-14 is capable of controlling up to 4 motors simultaneously. It has an on-board Motorola 68000 CPU to handle complex motion instructions. It communicates with a PC using a PC bus card. Commands are sent to the controller using a simple ASCII command language. Responses from the controller are also in ASCII. A complete description of the Unidex-14 control system is contained in the Unidex-14 operating manual in appendix A. The controller also has built in handling of limit switches (which cut motor power in cases of over-travel) and home switches (which calibrate the position of the stage).

The Unidex-14 control language uses simple 2 letter commands. For example, the command `VL300` sets the motor velocity to 300 units/second. There are also commands for controlling which axis is controlled by the commands. For instance, to set the xaxis as the current axis, the `AX` command would be used. After this command is issued all subsequent commands will apply only to the x axis, unless another axis command is issued. To control all of the axis at once, the `AA` (all axes) command is used. Further information on the use of the Unidex-14 command language are contained in the Unidex-14 manual.

To automate the positioning task and free the user from having to remember commands, a QuickBASIC program call `WAKE.BAS` was written to control the motor using the Unidex-14 command language. Ths program issues the approriate commands to the controller as required by the use of the program.

`WAKE.BAS` is menu driven and provides graphical feedback the user. There are 5 main components to the program. The main component is automatic positioning during a carriage test run. The second component is the selection of the points to be sampled during an entire wake survey test. The third component is home point initialization and origin calibration. The fourth is general motor control. The final component is interactive low level communication with the controller using the Unidex's own language.

4.1 Starting WAKE.BAS and Using QUICKBASIC

`WAKE.BAS` was written in QuickBASIC 4.5 for MS-DOS. Because of its large size in terms of code and memory requirements, it is not able to compile to run as an MS-DOS executable (.EXE) file. (This is mainly due to the memory restrictions imposed by MS-DOS) Therefore, it must be run in the QuickBASIC environment. A small batch file called `WAKE.BAT` should be installed in the `\BATCH` or `\BAT` directory (or any directory in the MS-DOS search path) to run QuickBASIC and `WAKE.BAS` automatically. This file should contain the following lines:

```
cd \wake                      ! Switch to \WAKE directory
\qb45\qb.exe /run wake        ! Start QB45, load wake.bas and run it
```

Once the QuickBASIC environement is loaded, it will automatically load `WAKE.BAS` and run it. Another way to start the program is to start QuickBASIC itself using `\qb45\qb` in the directory where `WAKE.BAS` is located. `WAKE.BAS` must always be run from the same directory as where it is stored. This is because `WAKE.BAS` uses a configuration file from the `WAKE.BAS` directory.

Once QuickBASIC is running it can be stopped at any time by using *CTRL-Break*. This will cause the program to stop running and will display the QuickBASIC editor and environment. Selecting *Quit* from the main menu of `WAKE.BAS` will also return the program back to the QuickBASIC program. To start the program again, use *SHIFT-F5*. This will reset the program and start it from the beginning. To exit the QuickBASIC environment choose *Exit* from the *Files* menu (use the *ALT* key to display the QuickBASIC menus).

Note: Do not make any changes to the QuickBASIC software accidently. When you are exiting QuickBASIC it will ask you if you want to save your changes (if you have accidently changed anything). Always answer No to this question.

4.2 The WAKE.CFG Configuration File

`WAKE.BAS` uses an ASCII configuration file located in the `\WAKE`. This file contains all of the configuration information used by `WAKE.BAS`. A sample listing of the `WAKE.CFG` file is found in figure 3.1.

Generally, the configuration file would usually be changed once for every different model. The only common parameter to change is the home calibration file, set in **line 10** or **line 31** depending on the unit system being used. These results from the changing propeller location for each model and hence the need for different calibration files defining the motor home point.

```
****  Wake Survey Positioning System - Configuration File
****  October 16/94
Default Units                                ,mm
Number of Different Unit Systems(max 10) ,2
mm
in
**************Defaults for mm Unit system
Unit Name                                   ,mm
Unit Conversion 393.701 Pulse=1mm           ,393.701
Calibration Home File                       ,617_stbd.cal
Xvel (mm/s):,     50
Yvel (mm/s):,     50
Xacc (mm/s^2):,     50
Yacc (mm/s^2):,     50
Number of Jog Velocities,        7
jogvelocityslow (mm/s),          1
JogVelocityLow,                  5
JogVelocityMedium,               10
JogVelocityHigh,                 20
JogVelREALLY HIGH,               30
JvelUltraHigh,                   40
JvelSUpper,                      50
Number of Jog Accelerations,     4
JogAccelerationLow (mm/s^2),     1
JogAccelerationMed,              5
JogAccelerationHigh,             15
JogAccUltraHi    ,               100
****************Defaults for Inch unit system
Unit Name                                   ,in
Unit Conversion 10000 Pulse=1 inch          ,10000
Calibration Home File                       ,wake_in.cal
Xvel (in/s):,     1
Yvel (in/s):,     1
Xacc (in/s^2):,     1
Yacc (in/s^2):,     1
Number of Jog Velocities,        4
jogvelocityslow (in/s),          .1
JogVelocityLow,                  .3
JogVelocityMedium,               .6
JogVelocityHigh,                 1
Number of Jog Accelerations,     4
JogAccelerationLow (in/s^2),     .1
JogAccelerationMed,              .4
JogAccelerationHigh,             .6
JogAccUltraHi    ,               1
```

Figure 3.1 - Contents of the `WAKE.CFG` configuration file for `WAKE.BAS`. Note: Line numbering is not part of the file.

This file can be modified as necessary using any ASCII text editor. When making changes, the commas that separate the text and numbers must be maintained, otherwise, QuickBASIC will not be able to read the configuration file.

The `WAKE.CFG` file is broken into 3 parts. The first parts contains the basic set-up information. The second and third parts contain information which determines the settings for each system of units. In the above file, two unit systems are supported, millimetres and inches, but other unit systems can be added if necessary.

Description of Configuration File Settings:

Lines 1-2: *Configuration File headers.* These lines are simply headers which indicate what the file is and the date on which it was created.

Lines 3-6: *Default units and unit names.* These lines determine the types of unit systems used by `WAKE.BAS`. **Line 3:** *Default Units* sets the default units. In this example, the default units are millimetres. This is the units used by the program at startup. **Line 4:** *Number of Unit Systems* contains the number of unit system to be used. In this example there are 2 unit systems. Finally, **lines 5-6:** *Unit Names* contain the names of the unit systems, `mm` for millimetres and `in` for inches. Any extra unit system would also have their name added to this list. Note that the order of the unit system names MUST be the same as the order of the unit system default settings for each unit system, described below.

Lines 7-27: *Millimetre Default Settings.* These lines determine the set up information for the millimetre unit system. A similar set of lines as described below is required for each unit system.

Ln 7: *Header* for the millimetre unit system

Ln 8: *Unit Name*: It must match the name given in **line 5**.

Ln 9: *Unit Conversion Factor:* This is the unit calibration factor, which is the number of pulses required for every ONE unit of travel. For example, the Unidex-14 uses units of pulses. However, other unit systems can be used by using the relationship between motor pulses and the unit of measure. For the wake survey apparatus, the Unidex motors have *2000* pulses for every revolution. The lead screws on the positioning frame have a ratio of *1* inch of movement for every five revolutions. Therefore, for every *5*2000=10000* pulses, the positioning frame will move *1* inch. Since *1* inch equals *25.4* mm, then for every *10000* revolutions, the frame will move *25.4* mm. This means that the motor has to turn *10000 / 25.4 = 393.701* pulses for every *1* mm of travel. This is the conversion rate used by WAKE.BAS.

Ln 10: *Calibration Home File:* The name of the default calibration home file when millimetre units are being used. In this example, the file name is `617_STBD.CAL` which would contain the home point numbers for this particular set of tests.

Lns 11-14: *Xvel, Yvel, Xacc, and Yacc:* These are the default settings for the x and y velocities and accelerations used during point to point moves.

Ln 15: *Number of Jog Velocities:* Determines how many different jog velocities there are.

Lns 16-22: *Jog Velocity Names and Values:* These are the different jog velocities available during motor jogging operations. Note that the actual names of the velocities are for reference only. They can be anything at all. WAKE.BAS uses only the numbers to determine the velocities.

Ln 23: *Number of Jog Accelerations:* Determines how many different jog accelerations there are.

Lns 24-27: *Jog Acceleration Names and Values:* These are the different jog accelerations available during motor jogging operations. Note that the actual names of the accelerations are for reference only. They can be anything at all. WAKE.BAS uses only the numbers to determine the accelerations.

Lns 28-45: These lines determine the set up information for the inch unit system. These lines contain the same sort of information as the lines for the millimetre unit system but the units would be in inches not millimetres.

For general applications, two calibrations files, `WAKE_MM.CAL` and `WAKE_IN.CAL` are available. Each of these files contains a calibration for homing the positioning stage to the approximate center of the stage travel. `WAKE_MM.CAL` is for the mm units system and `WAKE_IN.CAL` is for the inch unit system.

4.3 Using .RUN Files

To simplify the task of moving the probe to multiple points during one tank run, simple ASCII *.RUN* files are used. These files contain all of the sample positions for one wake survey test grouped by run number. For example, assume a given speed and that *160* points will be sampled for one test. Further, assume that at that speed, it is possible to sample *10* points per every tank run. Therefore, there would be *16* groups representing groups of ten points to be sampled for each run. `WAKE.BAS` uses a routine to automatically position the probe to the run points for each run. There is also a routine to select the points that will be sampled and to group the points for each tank run.

The *.RUN* files are plain text ASCII files of the form shown in figure 3.2. These files can be generated by any program but generally, the WAKE.BAS file generation and point selection mode is used.

```
1    ** Test RUN File
2    **
3    ** File Generated on Date: 03-20-1995     Time: 18:18:52
4    ** Diameters used:  45, 75, 105, 135, 165 mm
5    ** File used and Modified on Date:03-20-1995   Time: 19:37:07
6    ** File used and Modified on Date:03-21-1995   Time: 11:05:17
7    Basic Diameter (mm)      , 150
8    Total Number of Points in Test       , 120
9    Total Number of Runs in this test     , 12
10   Units,mm
11   Calibration File Name,quest.cal
12   Xvel(mm/s), 50
13   Yvel(mm/s), 50
14   Xacc(mm/s^2), 50
15   Yacc(mm/s^2), 50
16   FORMAT: comment,run no.,times run, point in run,coordinate type
     (rect. or polar), point1 x (or r), point1 y (or theta), .....
17   RUN#,1,0,10,R,21.73333,5.823429,5000,R,19.48557,11.25,5000,R,15.9099,
     15.9099,5000,R,11.25,19.48557,5000,R,5.823429,21.73333,5000,R,0,22.5,
     5000,R,-5.823429,21.73333,5000,R,-11.25,19.48557,5000,R,-15.9099,15.9
     099,5000,R,-19.48557,11.25,5000
18   RUN#,2,0,10,R,-21.73333,5.823429,5000,R,-22.5,0,5000,R,-21.73333,-5.8
     23429,5000,R,-19.48557,-11.25,5000,R,-15.9099,-15.9099,5000,R,-11.25,
     -19.48557,5000,R,-5.823429,-21.73333,5000,R,0,-22.5,5000,R,5.823429,-
     21.73333,5000,R,11.25,-19.48557,5000
19   RUN#,3,0,10,R,15.9099,-15.9099,5000,R,19.48557,-11.25,5000,R,21.73333
     ,-5.823429,5000,R,22.5,0,5000,R,37.5,0,5000,R,36.22222,-9.705714,5000
     ,R,32.47595,-18.75,5000,R,26.5165,-26.5165,5000,R,18.75,-32.47595,500
     0,R,9.705714,-36.22222,5000
20   RUN#,4,0,10,R,0,-37.5,5000,R,-9.705714,-36.22222,5000,R,-18.75,-32.47
     595,5000,R,-26.5165,-26.5165,5000,R,-32.47595,-18.75,5000,R,-36.22222
     ,-9.705714,5000,R,-37.5,0,5000,R,-36.22222,9.705714,5000,R,-32.47595,
     18.75,5000,R,-26.5165,26.5165,5000
     . . . . .
     . . . . .
```

Figure 3.2 - Contents of .RUN file.

Line 1-2: The first lines in the *.RUN* are user entered comments. There can be as many of these lines as the user enters. Tyipically the comments would include the scale and model speeds and whether the model was fixed or free.

Line 3: File generation date

Line 4: Offset diameters that were used, given in the unit type.

Line 5-6: Various times and dates that the file was loaded, used and updated by `WAKE.BAS`.

Line 7: Basic diameter of the propeller. The offset diameters in **Line 4** are calculated as percentages of the basic propeller diameter.

Line 8: Total number of positions to be sampled in the test.

Line 9: Total number of grouped runs in the test.

Line 10: Type of units the coordinates are given in.

Line 11: Calibration file name for the model and test.

Line 12-15: Velocity and acceleration data for the x and y axes.

Line 16: Comment line describing the format used for each run group.

Line 17-20: First 4 run groups.

The run groups each have the following format:

Comment	Run Number	Number of times its been run	Number of points in the

Run Data

run	Coordinate Type: Rectangular or Polar	x or r	y or theta	Delay Time

Point 1 in the Run

(milliseconds)	Coordinate Type (Rectangular or Polar)	x or r	....

Point 2 in the Run

The first 4 values in the run group are applicable to the whole run group. After the run data comes the data for each point in the run group. There are four values for each point in the run.

4.4 WAKE.BAS Main Screen

The main screen for `WAKE.BAS` contains 4 main sections. The upper left corner of the screen is *Probe Position* window devoted to plotting the position of the controller and the position of the sample points for each test. The upper right side of the screen is the *menu* window. All menus are contained in this area. Just below the menu area is the *settings* window, which displays the current settings for the set velocities and accelerations. The current position of the probe is also displayed in the settings area. Finnaly, the bottom half of the screen is the *input/output* window. This window is used to enter any required keyboard entries as well as to display any information to the user. The main `WAKE.BAS` screen is shown in figure 3.2.

The Main Menu is used to select all of the main components and operations of `WAKE.BAS`. It is the first menu displayed in the *Menu* window when `WAKE.BAS` is first started.

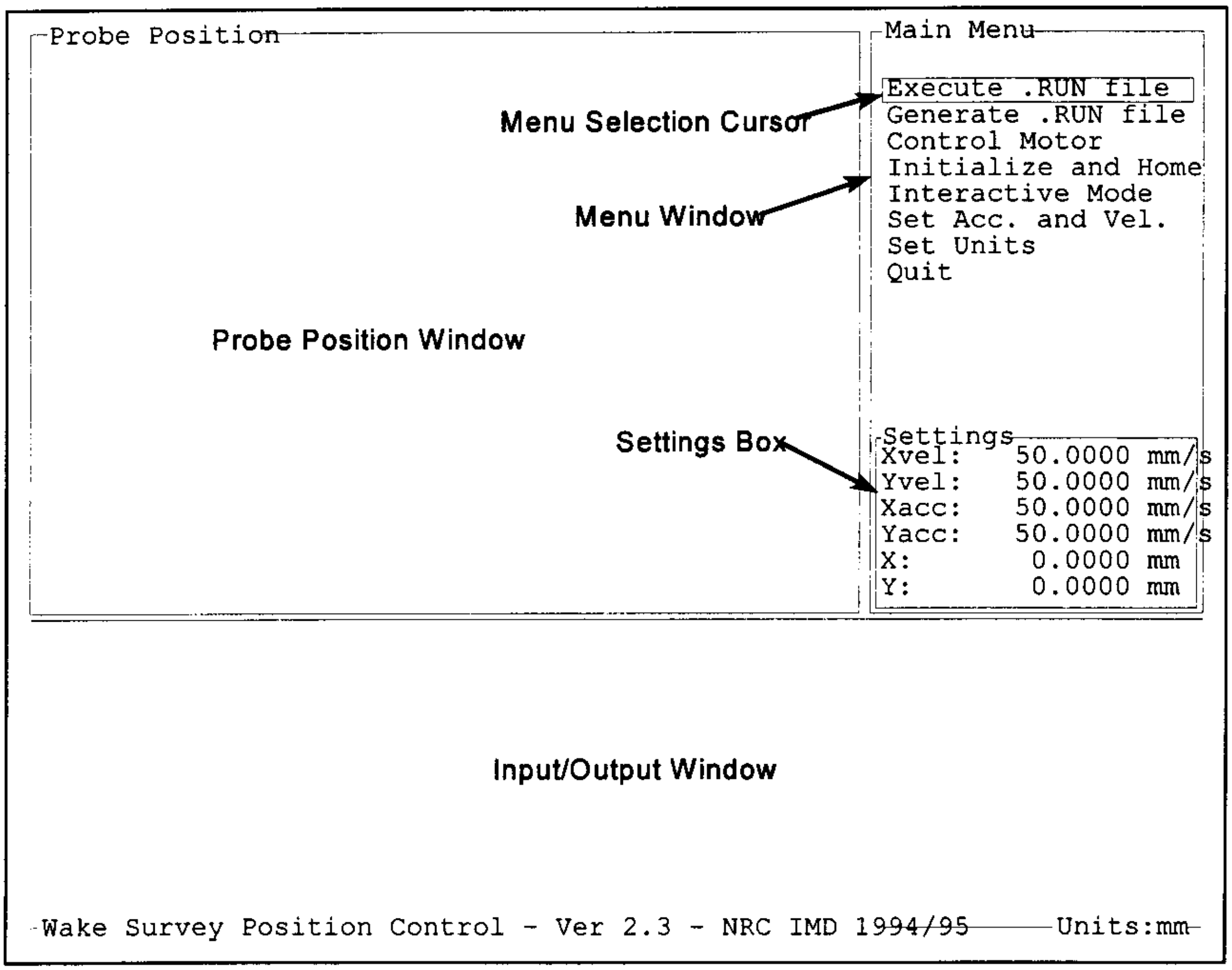


Figure 4.1 - `WAKE.BAS` main menu screen layout.

4.5 Execute a .RUN File

Execute a .RUN file is the main function of `WAKE.BAS`. It provides an automated method for positioning the probe at multiple sample positions for each carriage run. It uses *.RUN* files generated either manually (not recommended) or by `WAKE.BAS`'s *Generate .RUN file* option. When *Execute a .RUN file* is selected, the *menu* and *input/output* windows will be cleared. The program will then display all of the `.RUN` files in the directory and prompt for the name of `.RUN` file to load. The most previous `.RUN` file name used since `WAKE.BAS` started running will appear as the default filename. Entering the file name loads the file and displays the main *Execute a .RUN file* control screen, shown in figure 3.3.

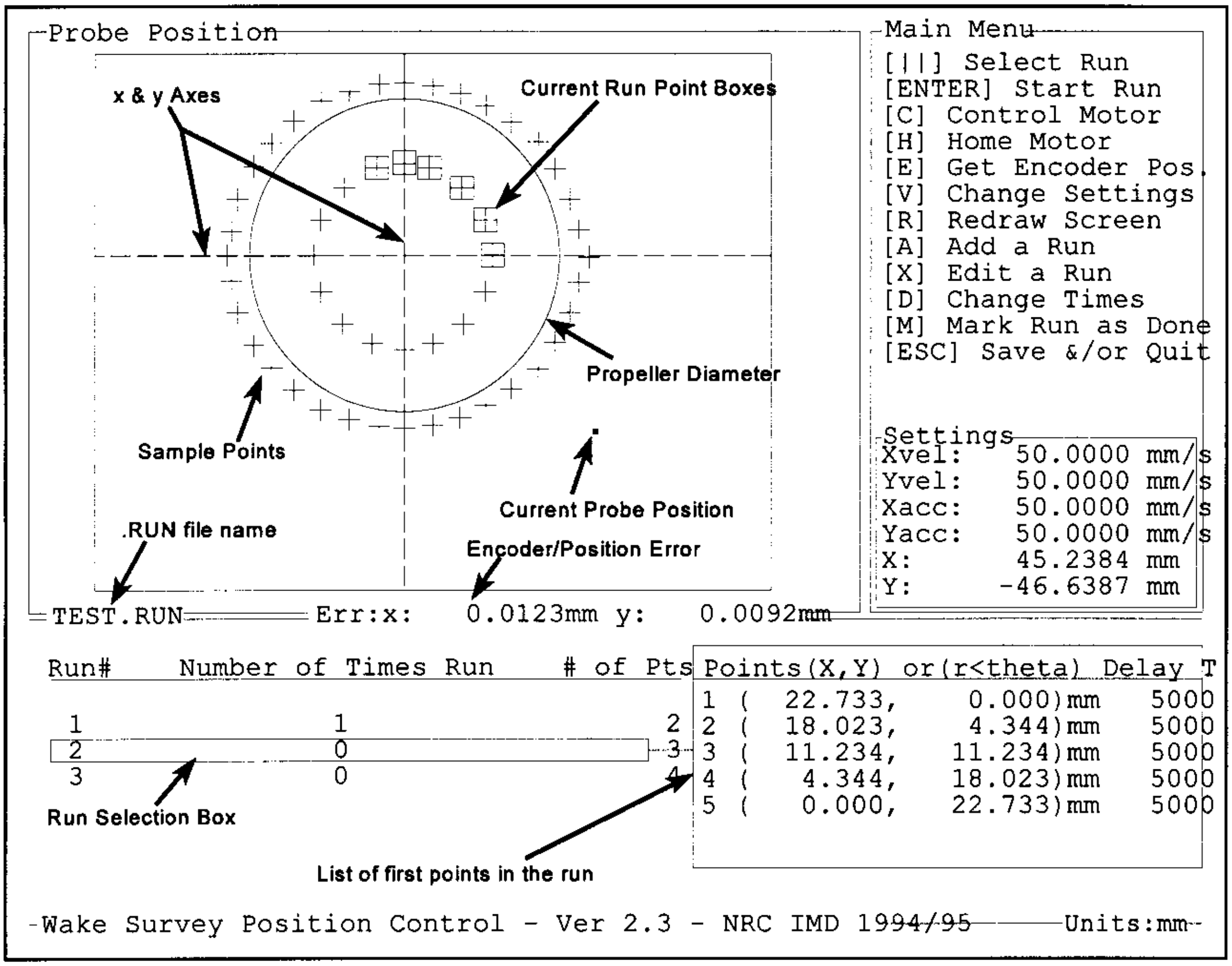


Figure 4.2 - *Execute .RUN File* main screen layout.

All of the sample points for the test are displayed in the probe position window as '+'s. Grey '+'s belong to runs that have not been executed (ie. the points have not been sampled). White '+'s belong to runs that have already been completed at least once. Yellow boxes around a set of points indicate the points that belong to the currently selected run. The propeller diameter and x/y axes are also displayed. The current position of the probe is indicated by the small green dot. At the top of the probe position window is the current test completion statistics. The first section, `Pt:  0 of 120= 0%` indicates how many of the overall test points have been completed. The second part, `Run  0 of 12`, indicates the number of test runs that have been complete out of the total number of runs in the test. At the bottom of the probe position window is displayed the name of *.RUN* file being executed and the encoder/position error. This errror is the absolute difference between the controller position and the actual position of the motor encoders. See section 4.15 for more details on the encoder error.

The menu window lists all of the available options. The input/output window displays three of the runs. The run inside the run selection box is the current run. A previous run and the next run are shown above and below this box. To the right of this list is another box that displays up to the first five points of the currently selected run. It displays their coordinates and their delay or sample times.

When the program is started, it will be in the *Run Selection* mode. The cursor keys can be used to select the run to be executed. Other keys are used to access motor control and other positioning functions.

[↑↓] Select Run:

When in the *Run Selection mode* the cursor keys change the current run (shown in the *run selection box*). The run list goes from low run numbers at the top to high numbers at the bottom. Therefore, the down cursor key goes to next higher run number and the down cursor key decreases to the most previous lower run number. Using **[Page up]** and **[Page Down]** skips five runs at a time. The **[Home]** key will go to the first run and **[End]** key will go to the last run.

[ENTER] Start Run:

Pressing **[ENTER]** while in the *Run Selection* mode causes the current run to be executed. Immediately upon starting the run, the probe will move to the first point in the run. The message:

```
Please wait...Positioning probe to start run..
```

will be displayed in a small box at the bottom of the input/output window. The green position dot will move to the new position leaving a trail of green dots. Also, the points in the run will change to blue '+'s and will be connected by grey lines. The first point in the run will have a green triangle drawn over it. The last point in the run will have a red octagon drawn around it.

If the probe was already at the new position when **[ENTER]** was pressed to start a new run, the program will not be able to tell. It will display the above message and appear to lock up. However, pressing the **[ESC]** key will break out of this loop and the program will continue as normal.

Once the probe is at the first position, a new message will be displayed:

```
Probe is in position for Run 3. [ENTER] to start run.
```

The data acquisition can now start sampling (this will be the zero tare segment). When a suitable amount of sample time (10 to 20 seconds) has elapsed, the carriage can start its run. When the carriage has reached its constant speed, press **[ENTER]** to start the run sequence. Another message will be displayed:

```
Run  3 has begun....Run Point  1 of 10
```

The probe will stay at the first sample point for the specified delay time. Then it will move to the next point in the run. The message will then change to `...POINT 2 of 10.` Again, it will stay at the point for the delay time and then move to the next point. This process continues until the probe reaches the last the point in the run. The probe will stay there for the delay time. When the time is finished the probe will stay at that point and the following message is displayed:

```
Run   3 has been completed...Press any key...
```

This indicates that the run has completed succesfully. Pressing any key returns the program to the *Run Selection* mode. The *number of times run* of the run will be increased by one. Also, the *run selection box* will move to the next run.

To escape from the *Start Run* mode, press **[ESC]** at the prompt:

```
Probe is in position for Run 3. [ENTER] to start run.
```

This will exit from the run without executing the rest of the run. It will return the program to the *Run Selection* mode. The number of times run will not be increased and the run selection will stay at the same run number.

[C] Control Motor:

This calls the *Control Motor* main screen, while still displaying the list of sample points. The *Control Motor* screen controls manual jogging and positioning of the probe. Its operation is explained in section 4.7. Quitting from *Control Motor* returns the program back to the *Run Selection* mode.

[H] Home Motor:

This will call the home motor routine directly from the *Execute .RUN file* screen. *Home Motor* is decribed in section 4.9. After the motor have been successfully homed, the program will return to the *Run Selection* mode.

[E] Get Encoder Position:

Displays the absolute encoder position. Pressing **[E]** will display the current encoder readings in x and y. Pressing any key will erase the reading and redraw the screen.

[V] Change Settings:

This option calls the *change velocities and acclerations* routine, described in section 4.9. It is used to set the x and y velocities and accelerations used during the probe moves. After the *change velocities and acclerations* routine is finished, the program wil return to the *Run Selection* mode.

This option sets the x/y velocities and accelrations of the probe positioning system. The input/output window will be cleared and a new menu will be displayed:

```
Current Settings:
     Xvel 50.0000 mm/s
     Yvel 50.0000 mm/s
     Xacc 50.0000 mm/s
     Yacc 50.0000 mm/s
     Finished
```

The cursor box can be used to select a setting to change. Pressing enter on any of the four settings will display a prompt requesting the new value:

```
Current Settings:
     Xvel 50.0000 mm/s
     Yvel 50.0000 mm/s
     Xacc 50.0000 mm/s     New Value for Xacc:█
     Yacc 50.0000 mm/s
     Finished
```

Typing a new value will change the value of the setting. If

[R] Redraw Screen:

The *Execute .RUN file* screen can become cluttered after several runs. For thsi reason, a *Redraw* feature is included.

[A] Add a Run:

Manually add a run to the end of the current run list. The input/output window will be cleared and the following prompt displayed:

```
Adding RUN# 14
How many points in the run ( 20 maximum):█
```

This is the number of points that will comprise the new run. Once a number is entered, the program will prompt for the values for each point:

```
Format:[R]ectangular or [P]olar, x or r, y or theta, delay (milliseconds)
Enter Point 1  :█
```

Using the format shown, the new points can be entered. The entries must be made on one line using commas to separate each parameter. If not enough entries are made the program will display: `Redo From Start` and the entry has to entered again in the proper manner. As each point is entered, a white '+' will be added to the probe position window.

For example: to enter a rectangular point of x=10.5, y=20.3 and a delay of 5000 milliseconds, enter:

```
R, 10.5, 20.3, 5000
```

Likewise, to enter a polar point at radius = 20.5 and angle theta (from the positive x axis) of 45 and a delay time of 8500 seconds, enter:

```
P, 20.5, 45, 8500
```

Note: The run will be ordered by the order the numbers are entered. The first number entered will be the first point in the run.

When the total number of points have been entered, the program will prompt:

```
Run has been added...Do you wish to enter another?
```

Pressing 'Y' will add another run. Pressing 'N' will redraw the screen and return the program to the *Run Selection* mode.

[X] Edit a Run

Edit the points in the current selected run. The input/output window will be cleared and the following message (if editing run 5 with 12 points, for example) will be displayed:

```
Editing RUN# 5
There are currently 12 points in this run.
Do you wish to change this?
```

Entering 'Y' will display:

```
How many points in the run ( 20 max):█
```

A new number of points for the run can now be added. Answering 'N' to the question leaves the number of points in the run as the same.

▲ ***WARNING:*** Entering a number of points LOWER than the original number of points will remove the last points in the run. For example, entering 7 for a run with 12 points will remove the last 5 points in the run. However, these points can be brought back by using **[X] Edit** again and entering the original number of points in the run.

After the number of points has been determined, the program will display the coordinate and delay time information from the first point and check to see if it should be changed:

```
Point  1 is: R  x:    15.9099mm y:   -25.7243mm  Delay:  5000ms
Do you wish to change this point?
```

The `R` indicates the coordinate type is rectangular. The `x` and `y` are the rectangular coordinates and `Delay` is the delay time in milliseconds. The coordinate type was polar, the program would have read:

```
Point  1 is: P  r:    24.3442mm th:   45.2484mm  Delay:  5000ms
Do you wish to change this point?
```

The `P` indicates the coordinate type is polar, `r` is the radius, `th` is the angle measured counterclockwise from the positive x-axis, and `Delay` is the delay time in milliseconds.

Entering 'Y' allows the user to enter a new coordinate and delay time in the same manner as **[A] Add a Run**:

```
Format:[R]ectangular or [P]olar, x or r, y or theta, delay (milliseconds)
Enter Point 1  :█
```

The entries must be made on one line using commas to separate each parameter. If not enough entries are made the program will display: `Redo From Start` and the entry has to entered again in the proper manner. See **[A] Add a run**, above, for more information.

For cases where the new specified number of points is greater than the original number of points, zeros will appear as the current points when the editing point is higher than the original number of points.

[D] Change Times:

Changes the delay times for every point in the *.RUN* file. To change single points, use the **[X] Edit a Run**.

[M] Mark run as Done:

This command will increment the *Number of Times Run* variable for the current selected run. It will then move to the next run in the sequence. It is useful for cases where the program prematurely ends before the *.RUN* file can be saved.

[ESC] Save &/or Quit:

This command is used to both save the *.RUN* file and to exit from the *Execute .Run File* screen. Saving the *.RUN* file records any added or edited runs and also records how many times each run has been completed (ie. the *Number of Times Run* variable).

When **[ESC]** is first pressed, the input/output window will be cleared and the following prompt will be displayed:

```
Do you wish to save this file?
```

Answering 'Y' will bring up the prompt for the filename. The default name is original name of the *.RUN* file. Pressing enter will save the file with the default name, erasing over the previous *.RUN* file. Typing another name will save the file under the new name.

After the save is finished, or if 'N' was answered to the `Save file?` question, the program will display another prompt:

```
Do you wish to return to the main menu?
```

Answering 'Y' will return the program to the main menu. Answering 'N' will return the program back to the *Run Selection* mode of the *Execute .RUN file* screen.

4.6 Generate .RUN File

`WAKE.BAS` has a built in *.RUN* file generation routine. This system is chosen from the *main menu* screen. Selecting *Generate .RUN file* will clear the input/output window and the menu window. A new menu will displayed in the menu window:

```
Generate .RUN File

Create New Points
Load Points File
Return to Main Menu
```

The menu has two basic options: *Create New Points* and *Load Points File*. The third option, *Return to Main Menu* will exit the *Generate* routine and return the program to the *main menu.*

Load Points File:

WAKE.BAS allows the user to save their selected sample points in a *Points* file. This file can now be recalled using the *Load Points File* selection. A *Points* contains only the positions of the points. It does not contain any groupings according to runs. For this reason, it is often convenient to create one *Points* file containing all of the sample points for one particular model. This file can then be used to create a *.RUN* file for each velocity being tested.

Choosing *Load Points File* lists all of the points files in the current directory and then displays a prompt for the *Points* file name. *Points* files will have the extension `.PTS`. Once the name has been entered the WAKE.BAS will load the *points* file and then display the **Select Points** menu, discussed in section **4.6.1**.

Create New Points:

Using the *Create New Points* file starts with no points or other selections. After this option has been selected, the following menu will be displayed in the menu window:

```
Generate New Points

Units in mm
Units in Inches
Return to Main Menu
```

This menu sets the type of units, either *inches* or *millimetres (mm)*. The third option, returns to the *Main Menu*. Selecting the unit type brings up another menu:

```
Calibration File Nm.
Change Vel. & Accel.
Done - Select Points
```

The first option, *Calibration File Name*, selects which home file will be used for this *.RUN* file. The default home file will be the home home file specified in the WAKE.CFG configuration file. The second option, *Change Vel. and Accel.* sets the velocities and accelerations of the x and y axis of the positioning stage. This option calls the *Change Vel and Accel* menu, described in section 4.8. When the calibration file and positioning stage settings have been selected, *Done - Select Points* is used to start selecting the points.

Select Points:

Selecting points starts by clearing the menu window and the input/output window. The x and y axes will be displayed in the probe position window along with a box outlining the maximum travel of the positioning stage. The program will prompt for the basic propeller diameter:

```
Enter the propeller diameter (diam=2*R): █
```

This is the main diameter of the propeller. Entering the diameter brings up another prompt:

```
How many offset ratios (r/R) will be used [Default=5, Max=10]: █
```

This is the number of offset ratios from the propeller diameter. They are represented by r/R, where r is the radius of the offset circle and R is the radius of the propeller. If the default of 5 is used the program will display:

```
Accept defaults of r/R = 0.30, 0.50, 0.70, 0.90 and 1.10 ? [Default = Y]
```

If the answer is 'N' or a number of diameters other than five was entered, then the program will begin to prompt for the offset ratios to use:

```
Enter r/R ratio for each diameter:
Enter r/R ratio for diameter   1 of   5 : █
```

Enter the ratio r/R. This will continue until all of the offsets have been entered. The ratios can be entered in any order.

Once all of the ratios have been entered, the program will display the actual offset diameters in units on the bottom of the probe position window:

```
Diam:45.72|76.2|106.68|137.16|167.64
```

At the bottom of the input/output window another prompt will be displayed:

```
Do you wish to change these diameters? [Default = N]
```

Answering 'Y' to this questions returns the program to the `number of offsets` prompt, were the offsets can be entered again. Answering 'N' to question brings up the *Select Points* screen, discussed in the next section.

4.6.1 The Select Points Screen

The *select points* screen plots the offset diameters in the probe position window. It also displays the *Select Points* menu in the menu window. If a *points* file was loaded, the points will be displayed as yellow boxes on the probe position screen. When no *points* file has been loaded, the screen look like the one shown in figure 4.6.

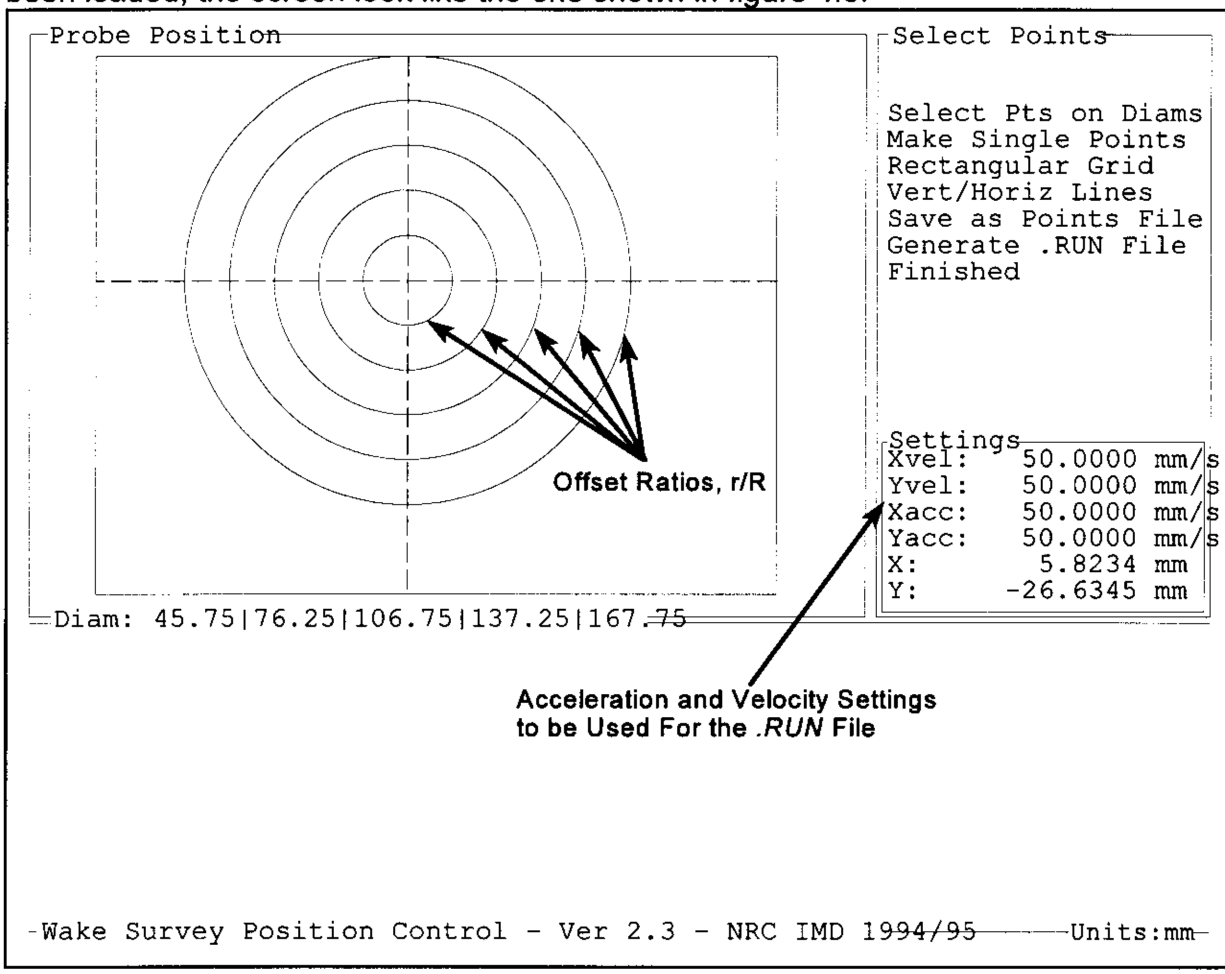


Figure 4.2 - *Execute .RUN File* main screen layout.

The *Select Points* menu options are described below.

Select Pts on Diams (Select Points on Diameters):

This is the main point generation command. The individual offset circles can be selected and points generated on them both manually and automatically. Selecting this option displays the screen shown in figure 4.3.

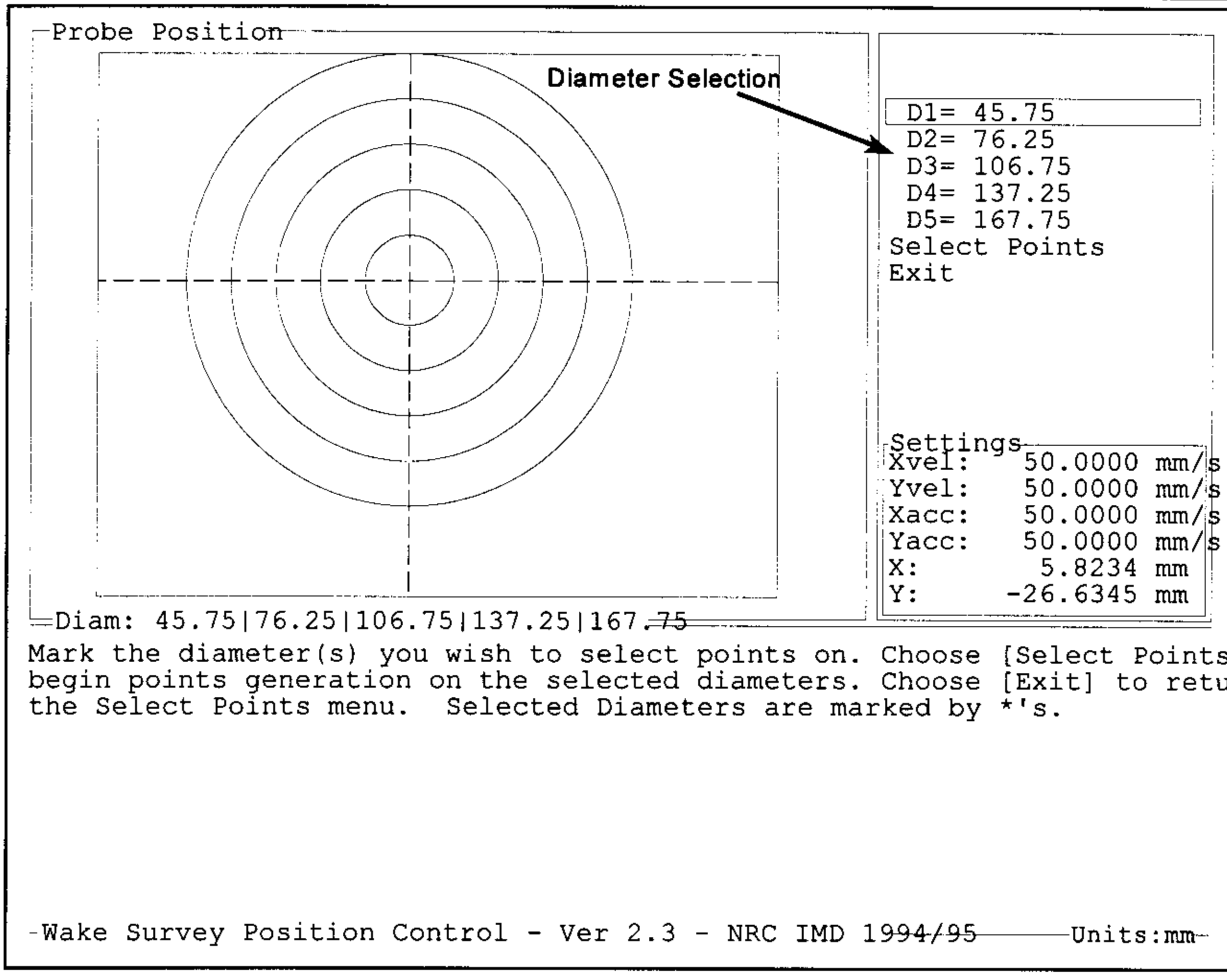


Figure 4.2 - *Generate .RUN File*, *Select Points* menu screen.

A simple menu displaying the offset diameters being used is displayed in the menu window. A short set of instructions appears in the input/output window. The menu is used to select the diameter(s) which will have points generated on. Selecting a diameter will put a '*' in front of it to indicate it has been selected. Selecting a diameter which already has a '*' in front of it will remove the '*' and unmark the diameter.

When the diameters have been selected, choosing *Select Points* from the menu will display the available point generation options:

```
Choose Point Generation Option:
[S]ingle points   [G]enerate equally spaced points   [ESC] Finished
```

[S]ingle Points Generation:

Generates single points on the marked diameter(s) at the angle specified by the user. The program will prompt for the angle, measured counter-clockwise from the positive x axis:

```
Single Points:enter degree offset from the x-axis, counter-clockwise
is positive.
Degree offset [ENTER to end]: █
```

For example, typing 45 will produce points on the marked diameter(s) at 45 degrees from the positive x axis. Yellow boxes will mark these points on the diameter(s). After entering the angle, the prompt will be repeated. More angles can continue to be entered. When finished selecting the angles, pressing enter on a prompt will display the following message and then return the program to the

```
End of angle entry
Total Points = 35

Press any Key to continue...
```

The total points lists the total number of points created so far. Pressing any key returns the program to the `Choose Point Generation Option` prompt. At this point, point generation can continue on the current marked diameters or pressing **[ESC]** will return the program to diameter selection menu.

[G]enerate Equally Spaced Points:

Generates a series of eqaully spaced points over the entire circumference of each marked diameter. The program will prompt for the degree spacing to use:

```
Degree spacing (Xaxis is 0 Deg) [Default = 15]: █
```

For example, typing 10 degrees would select points every 10 degrees along the circumference of each marked diameter (0, 10, 20, 30...). Yellow boxes will mark the selected points. After the points have been generated, the following prompt will appear:

```
Total number of angles is: 36
Finished generation of equally spaced points...
Total points = 180

Press any key to continue...
```

Pressing any key returns the program to the `Choose Point Generation Option` prompt.

[ESC] Finished:

Returns back to the diameter selection menu.

The *Select Points on Diameter* command can be accessed many times during the point selection process. WAKE.BAS will continually update the points list and keep the points sorted by angle and diameter.

▲ ***WARNING!*** There are limits to the number of points that may be generated on the diameters. Currently, the software is set to handle 73 points on one diameter. This would correspond to 5 degree equally spaced points. Trying to add more points will cause a `subscript out of range` error in QuickBASIC.

▲ ***WARNING!*** Once the a *.RUN* file has been generated, the *Select Points on Diams* function will not work properly. This is because the run generation routine has to convert all coordinates to rectangular coordinate system. This will cause *Select Points on Diams* to behave erroneously.

Make Single Points:

This option adds single, user-defined points. The input/output window will display the following prompt:

```
Generate groups of single points
Point #1
Format: [R]ectangular or [P]olar , x or r , y or theta(deg)
Enter Point 1 :█
```

Using the format shown, the new points can be entered. Theta is measure positive counter-clockwise form the positive direction of the x axis. The entries must be made on one line using commas to separate each parameter. If not enough entries are made the program will display: `Redo From Start` and the entry has to entered again in the proper manner. As each point is entered, a yellow box will be added to the probe position window.

For example: to enter a rectangular point of x=10.5, y=20.3 and a delay of 5000 milliseconds, enter:

```
R, 10.5, 20.3, 5000
```

Likewise, to enter a polar point at radius = 20.5 and angle theta (from the positive x axis) of 45 and a delay time of 8500 seconds, enter:

```
P, 20.5, 45, 8500
```

After each point is entered, the program will prompt:

```
Enter another? Press [N] to stop...
```

Pressing 'N' will stop the single point generation routine. Pressing any other key will begin another point entry.

Rectangular Grid:

This option will produce a mesh-grid of rectangular points centred around the x and y axes. The program will display the prompt:

```
Generate Rectangular Grid

Width of grid (X size): █
```

This is the width of the grid (in the z direction) in real units. The program will then prompt for the hieght (Y size). After the size of the grid has been determined, the number of points in each direction is required:

```
Enter number of x  points.  Should be an odd number to get the y axis.
Number of x points: █

Enter number of y  points.  Should be an odd number to get the x axis.
Number of y points: █
```

Making these number odd will put points at the x=0 and/or y=0 positions. A typical grid is shown in figure 4.7. Once these parameters have been entered, the grid will be produced. Horizontal and vertical lines will be plotted showing the x and y positions of the grid points. Yellow boxes, indicating the new points will also be added. The program will then display the total number of points and pause:

```
Grid created.  Total Points  155
Finished....Press any key....
```

The total number of points is all of the points created so far. Pressing any key returns to the *Select Points* menu.

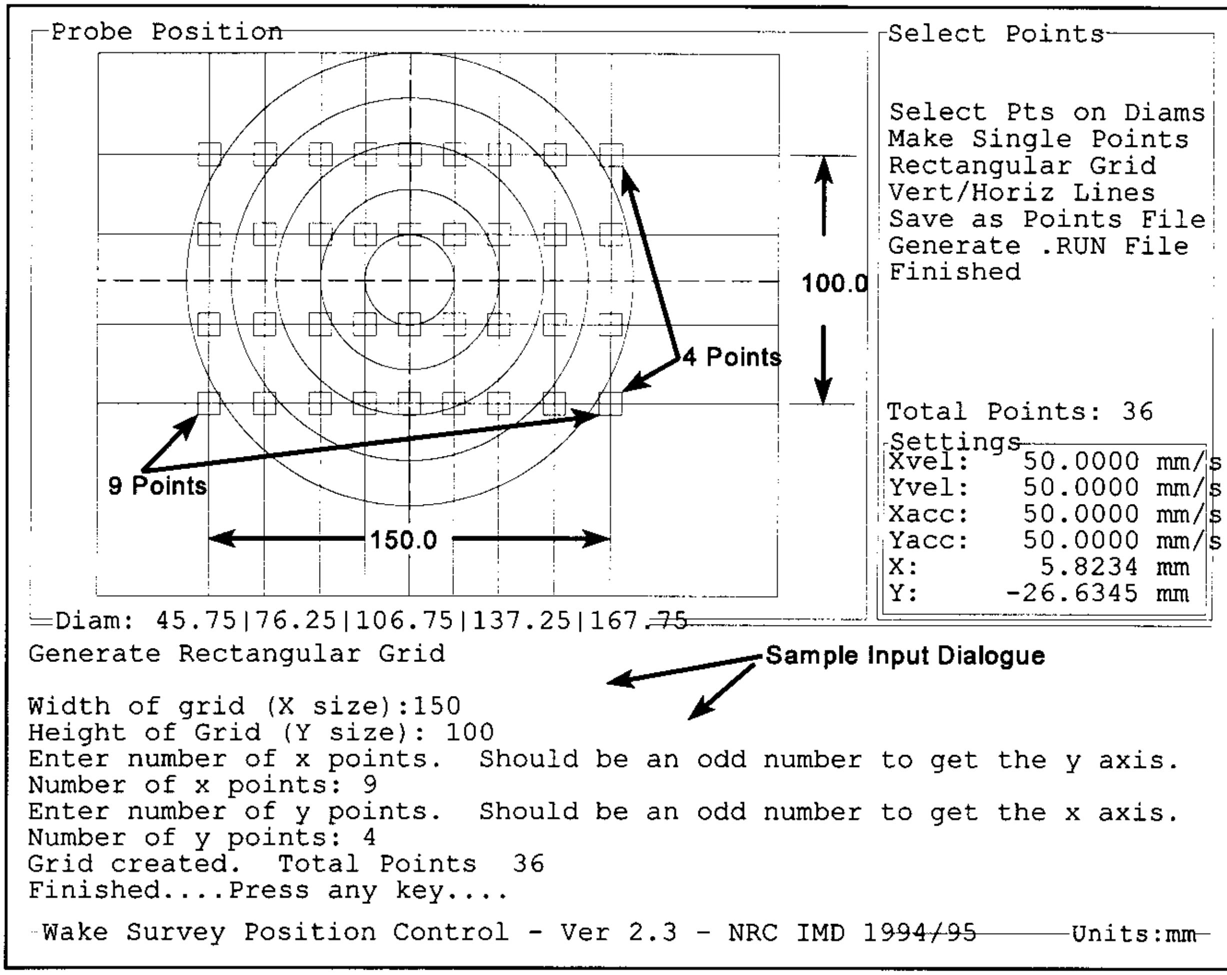


Figure 4.2 - *Generate .RUN File*, *Select Points* menu screen.

Vert / Horiz Lines (Vertical or Horizontal Lines):

Creates horizontal or vertical lines of a specified length and creates a specified number of points on that line. These lines are centred at the axis. When this option is selected the program will display:

```
Generate a Single Line...
[H]orizontal or [V]ertical?
```

[V]ertical:

Choosing Vertical will display another prompt:

```
Generate Vertical Line:
Enter x position [default = 0.00]: █
```

This is the x position of the vertical line. When entered, a solid gray line will be plotted on the graph at the x position entered. The computer will then request the number of points on the line:

```
Enter number of points: █
```

This is the total number of points in the y-direction of the line. Finally, the total length, in the y direction, of the line is entered:

```
Enter length of line: █
```

The program will then create the points and display the following message:

```
Total points:  18
Finished....Press any key....
```

Pressing any key returns to the *Select Points* menu. A sample output is shown on figure 4.9.

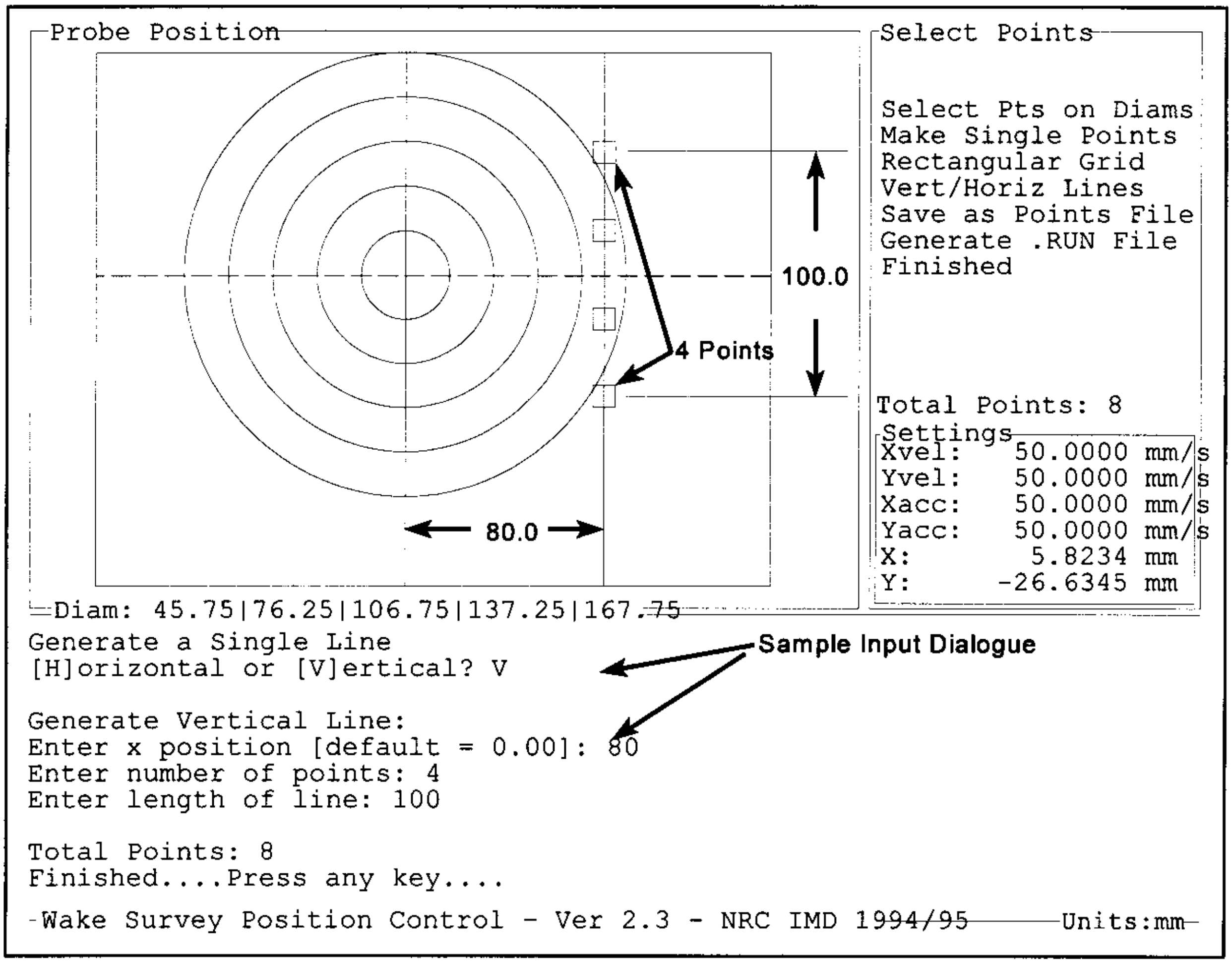


Figure 4.2 - *Generate .RUN File*, *Select Points* menu screen.

[H]orizontal:

Choosing Horizontal will display another prompt:

```
Generate Horizontal Line:
```

```
Enter y position [default = 0.00]: █
```

This is the y position of the horizontal line. When entered, a solid gray line will be plotted on the graph at the y position entered. The computer will then request the number of points on the line:

```
Enter number of points: █
```

This is the total number of points in the x-direction of the line. Finally, the total length, in the x direction, of the line is entered:

```
Enter length of line: █
```

The program will then create the points and display the following message:

```
Total points:  18
Finished....Press any key....
```

Pressing any key returns to the *Select Points* menu. The output is the same as for the vertical line shown in figure 4.9, except the line is horizontal.

Save as Points File:

To prevent having to generate the same points every time a new *.RUN* file is required, the points can be saved as a *Points* file with extension `.PTS`. This opton will save the point positions and also the calibration file, units and the accelerations and velocities of the positioning stage. This file can be loaded up at the beginning of the *Generate .RUN file* command.

When selected, the program will display:

```
Save as a Points File
This will save all of your points collected to a point storage file.
It is recommended that you name the file with a .PTS extension.
Filename: █
```

After entering a proper filename, the program will save the data and display the following message when its done:

```
Saving.....Done
Press any key to continue....
```

Pressing any key returns control back to the *Select Points* menu.

Generate .RUN File:

Generates the *.RUN* files used by *Execute .RUN file*. When selected, the following is diplayed:

```
Generate .RUN file

This will generate a .RUN file.  It is recommended that you name the file
with a .RUN extension.
Enter the name of the run file: █
```

The *.RUN* files should always be saved with the `.RUN` extension which must be explicitly typed as part of the name. Further, the name should be chosen to represent the name of the project and the carriage speed for the trun file. For example, for projext 95617 and V_s = 12 knots, the filename could be `617VS12.RUN`. Entering a valid filename will bring up another prompt:

```
Do you wish to create a .PTS , .TXT and .XY file? [Default = N]
```

The `.PTS` is the namne of the points file that is automatically saved every time `.RUN` file is generated. The `.TXT` file is an ASCII text file which contains a summarized list of every run in the `.RUN` file and every point in the run. The `.XY` file contains a simple list of all of the x/y coordinates of the sample points for import into a spreadsheet program, such as Excel. This prompt controls the selection of the name for the `.PTS`, `.TXT` and `.XY` files. Entering 'Y' will name the files using the .RUN file name. For example, if the run file was named `W617VS12.RUN` the other files will named `W617VS12.PTS`, `W617VS12.TXT` and `W617VS12.XY`. Answering 'N' to the prompt will save the files with the names `LAST.PTS`, `LAST.TXT` and `LAST.XY`. This prevents the build up of unwanted files by only keeping these files for the last time a `.RUN` file was generated.

The program will next prompt for the delay time, in milliseconds, at each point:

```
Delay time (milliseconds) at each point [Default = 5000]: █
```

After the time has been set, the program will display the following prompt:

```
[S]et number of points or let computer [O]ptimize points per run?
```

This determines how the points will be grouped into runs. Entering a **[S]et** amount will put the same number of points in each run. Entering **[O]ptimize** will make the computer calculate the optimal number of points per run based on the distance between each point and the time it takes to move the probe over this distance.

[S]et Number of Points Per Run:

Selecting this option will display the prompt:

```
Enter number of points per run: █
```

The sorting routine will make every run have this number of points. The only exception is the last run whcih can have less than this number of points.

[O]ptimize Number of Points Per Run:

A series of prompts will be displayaed:

```
Enter the constant speed distance of the carriage (m): █
```

This is the total constant speed distance of the carriage for the specified model velocity and carriage acceleration and decceleration.

```
Enter the carriage constant speed (m/s): █
```

This is the model speed (ie. the constant speed of the carriage).

```
Enter the time safety factor in percent: █
```

This is the time safety factor to account for inconsistencies in the carriage constant spped distance. It will effectively reduce the carriage constant speed distance entered above by the specified percentage. Usually, it should be left as zero.

```
Enter the lead time before run is started after set vel reached: █
```

Usually, the test is started just after the constant spped velocity is reached. This allows the flow to become fully developed. It also accounts for sometimes small overshoots in the carriage speed that occur just when the constant speed velocity is reached. Usually 3 to 5 seconds is sufficient.

After the velocity and distance parameters have been entered, the program will calculate the approximate time available for one carriage run. It will also display the reduced time due to the time safety factor and the and the subtraction of the lead time:

```
Total Time = 125 m / 1.723 m/s = 72.54788 s
Total Time after Safety Factor = 72.54788 s
Total Time after subtraction of lead time  = 68.54788 s
```

After the points per run selection method has been completed the program will beginning sorting the points into runs. The following message will be displayed:

```
Please wait....creating runs....
```

After the runs have been completed the number of runs and total number of points will be displayed:

```
Finished creating runs.
Summary:
Number of Runs : 12
Total Number of Points: 120
```

Below these summary statements is a prompt for the entry of comments to the run file. These comments are stored at the beginning of the run file.

```
Enter comments for the file.  Press [ENTER] on a blank line when finished.
>>
```

Any number of comments can be added to the run file. Pressing the enter key on a blank line will write these comments to the run file. Then the files will be saved:

```
Run File    :617VS12.RUN
Text File   :LAST.TXT
x/y File    :LAST.XY
Point File  :LAST.PTS
Creating files...please wait..................Done!
The files have been saved...Press any key to continue....
```

Pressing any key returns the program back to the *Select Points* menu.

Finished:

Returns back to the *Main Menu*

4.7 Control Motor

WAKE.BAS allows for seversal different methods of controlling the position of the motor. Most of these can be accessed using the *Control Motor* screen selected through the *Main Menu* or by selecting the *[C] Control Motor* option in the *Execute .RUN File* screen. Choosing this option will display Control Motor screen, shown in figure 4.2. If the *Control Motor* screen was called from the *Execute .RUN file* screen, the *Probe Position* window will keep the display of the sample points.

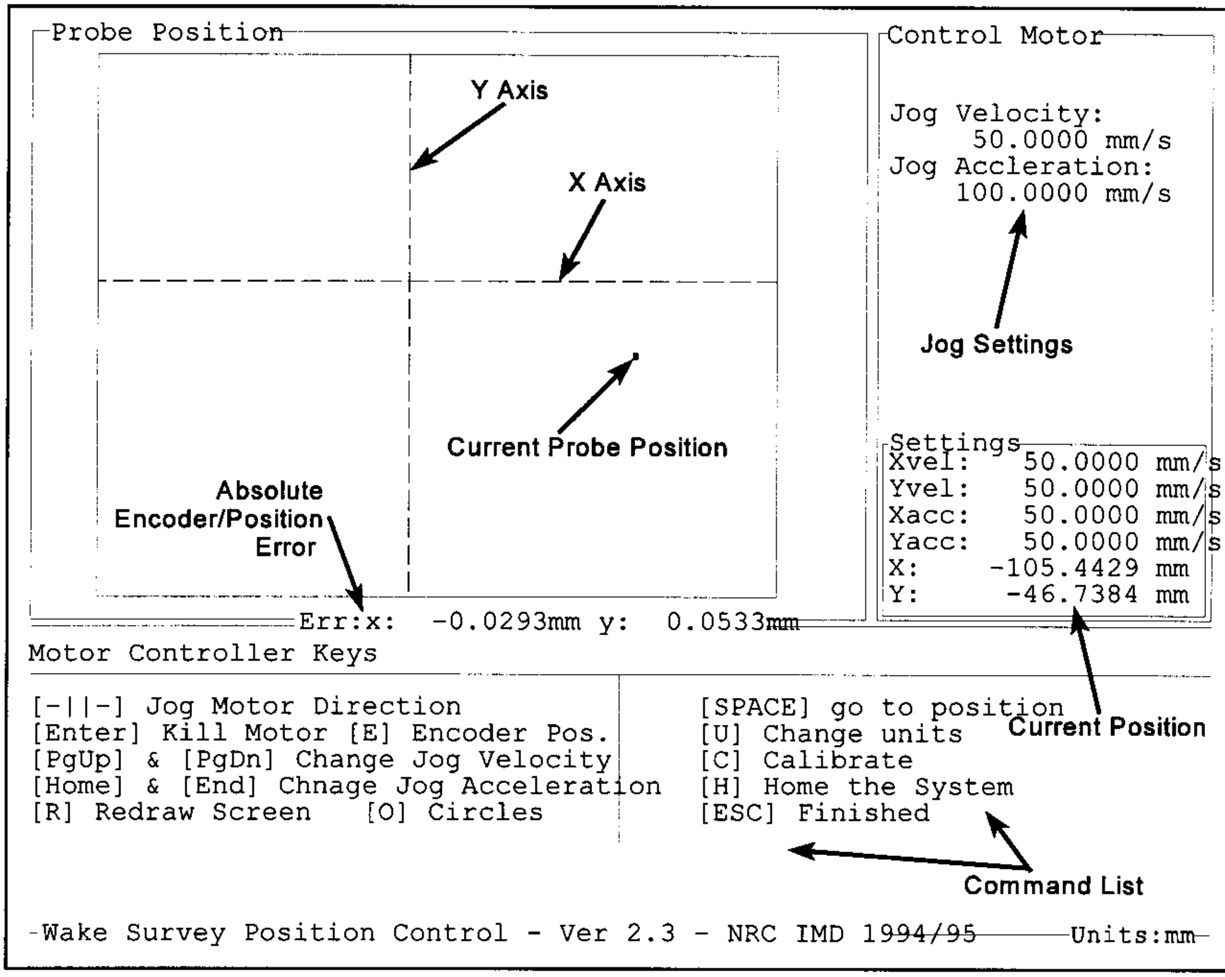


Figure 4.2 - *Generate .RUN File*, *Select Points* menu screen.

The *menu* window contains the current settings for the jog velocity and jog accelerations. The *input/output window* lists all available motor control commands. The *probe position graph* now displays the x and y axis as well as a box indicated the maximum travel of the position stage. It will also display a small green dot indicating the current position of the probe. The commands available are described below.

[←↑↓→] Jog Motor Direction:

Jogging the motor allows the user to manually start and stop the motors to position the stage. The four cursor keys control the direction in which the motor will jog. For example, pressing the up arrow key once (DO NOT HOLD IT DOWN!) will start the y-motor jogging in the up direction. The green position dot will begin to move in the up direction, leaving a green trail. The motor will continue to move until another key is pressed or until the stage hits a limit switch. Pressing the same direction key will decelerate the motor to an easy stop. When a different direction arrow key is pressed, the motor will be killed (see **[ENTER] - Kill Motor**), after which the stage will start moving in the new direction. Pressing any other command key will also kill the motor and then start the selected command. Finally, pressing any other key will have no effect.

[Enter] Kill Motor:

Killing the motor will stop the motor instantaneously. It will also flush any commands from the motor command queue which means that it really resets the motors without effecting the other command sequences. Generally, killing the motor should only be used as a last resort because it causes unecessary wear on the positioning stage. When jogging the motor, the same direction key should be used to stop the motor which will provide a smooth deceleration.

[PgUp] & [PgDn] Change Jog Velocity:

The jog velocity is displayed in the *menu* area. Using the *Page Up* and *Page Down* keys will cycle through the available jog velocities as specified in the configuration file. *Page Down* will decrease the velocity and *Page Up* will increase it. Under general operations and movements, the jog velocity can be fairly high (*50 mm/s* or *2 inches/s*). However, for precise movements, the jog velocity should be turned down (*1 mm/s* or *0.1 inches/s*). These lower velocities can provide extremely precise control over the stage position.

[Home] & [End] Change Jog Acceleration:

The jog accleration can be changed in a similar manner to the jog velocity. Generally, the acceleration does not need to be changed.

[R] Redraw Screen:

Because WAKE.BAS contuously plots the position of the probe, the screen can sometimes get messy. The *redraw* option will clean the screen up.

[O] Circles:

The *circles* option allows the user to perform smooth circular sweeps with the probe. Selecting this option will clear the *input/output window* and display the following:

```
Circular Movement

Enter center of the circle (X,Y):
```

This is the center in x/y coordinates of the circle to be swept. Generally, the center of the circle will be at 0,0. The input should be of the form x,y. For example:

```
Enter center of the circle (X,Y): 0,0
```

After entering the center of the circle, the program will request the diameter of the cicle:

```
Enter Diameter of the circle:
```

Next, the circular velocity will be requested. The circular velocity is the overall velocity of the probe as it makes its traverse.

```
Enter the circular velocity:
```

The probe will move to the starting position, which is directly below the circle. WAKE.BAS will display the message:

```
Moving to starting position....please wait
```

Once the probe is in the starting position, the window will be cleared and the message:

```
Press [ENTER] to start circle
```

will be displayed. Pressing the *Enter* key will start the circle: the probe will begin to accelerate towards the circle and then complete it. When the circle starts, another message will be displayed:

```
Starting run...Press [ENTER] to stop monitoring the position.
```

WAKE.BAS will update the position of the probe as it makes its circular traverse. However, the program has no way of knowing when the probe has finished traversing. Therefore, the user must monitor the motion until the probe has finished moving and then press return to stop monitoring the probe position. When the probe has finished moving, press the *Enter* key to return the motor control main screen.

▲ ***Warning:*** The control system will not always be able to track the motion of the position stage due to a problem with the Unidex-14 built in software. From the position graph it may appear that the stage has not moved. However, usually the stage will finish the circle, but the position will not be updated. In these situations, do NOT press the *Enter* key until the stage has finsihed the circle even though the position may not be updated any further.

[SPACE] go to position:

Pressing the *SPACE* bar brings up the go to a position screen, shown in figure 4.4. The *input/output window* is cleared and the *Set Position* menu is displayed in the *menu* area. The *go to position* screen sends the motor to a user specified location in either absolute or relative, rectangular or polar coordinates.

From the *Set Position* menu, several different options are available.

Rectangular Absolute: Sends the probe to a specific position in absolute coordinates. When selected, a prompt for the absolute rectangular position will be displayed. The coordinate should be entered as an x,y pair separated by a comma. As soon as the new coordinate is entered, the controller will move the stage in a straight line to the new position. The position in the *Settings* display will be updated as the stage moves. The position will also be plotted to the *Probe Position* graph window as a series of green dots.

Polar Absolute: Moves the probe to an absolute position in polar coordinates. The position will be in terms of r, theta where r is the radius from the origin and theta is the angle measured positive counter-clockwise from the positive x axis. Either positive or negative angles may be entered.

Rectangular Relative: Works the same as *Rectangular Absolute* but the coordinates are relative to the current probe position.

Polar Relative: Same as *Polar Absolute* but the coordinates are relative to the current probe position.

Change Vel and Acc: Allows the user to change the x and y velocities and accelerations. It is the same command as *Set Acc. and Vel.* in the *Main Menu.* Refer to section 4.5 for more information.

Change Units: Allows the user to change the units system being used. This the same command as *Set Units* in the *Main Menu.* Refer to section 4.6 for more information.

Finished: Returns control back to the *Motor Control* screen.

[U] to change units:

Changes the unit system. It is the same command as *Set Units* in the *Main Menu.* Refer to section 4.6 for more information.

[C] Calibrate:

This is the calibration for setting the relationship between the stage home point and the origin of the coordinate system. The *Control Motor* screen is used to precisely position the probe to the origin and then program will calibrate without homing and losing the origin position. Refer to section 4.6 for complete description of the position stage calibration procedure.

[H] Home the System:

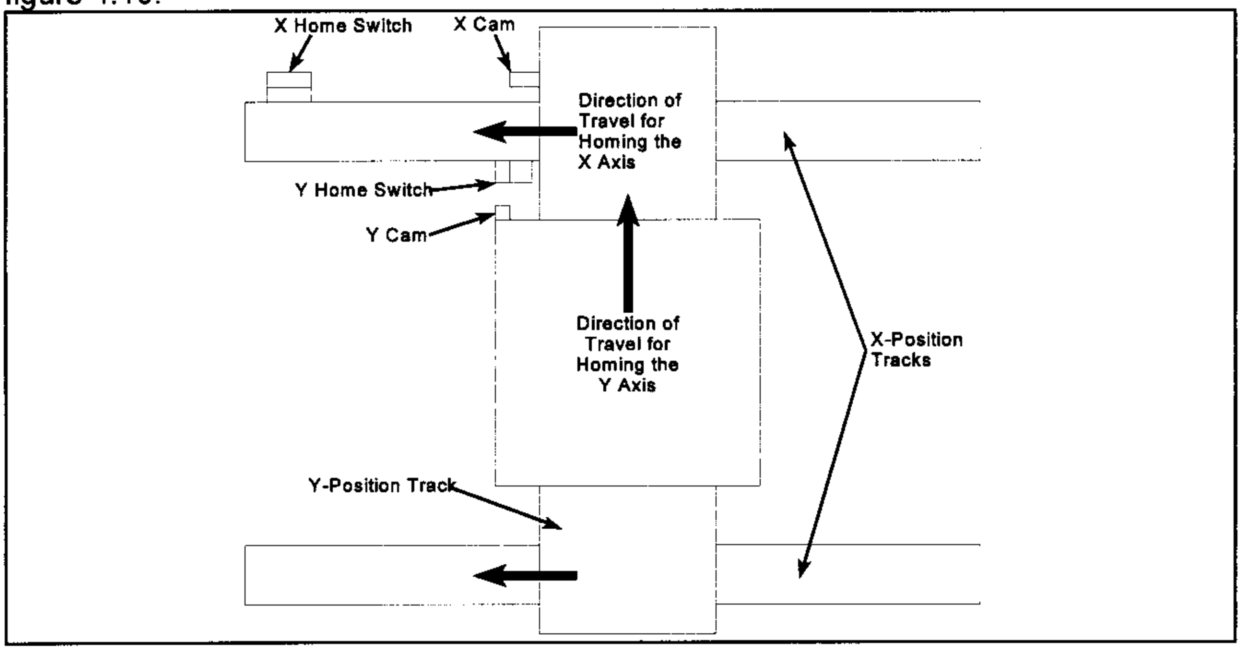
This will start the standard positioning stage home mode, described in section 4.7.

[ESC] Finished:

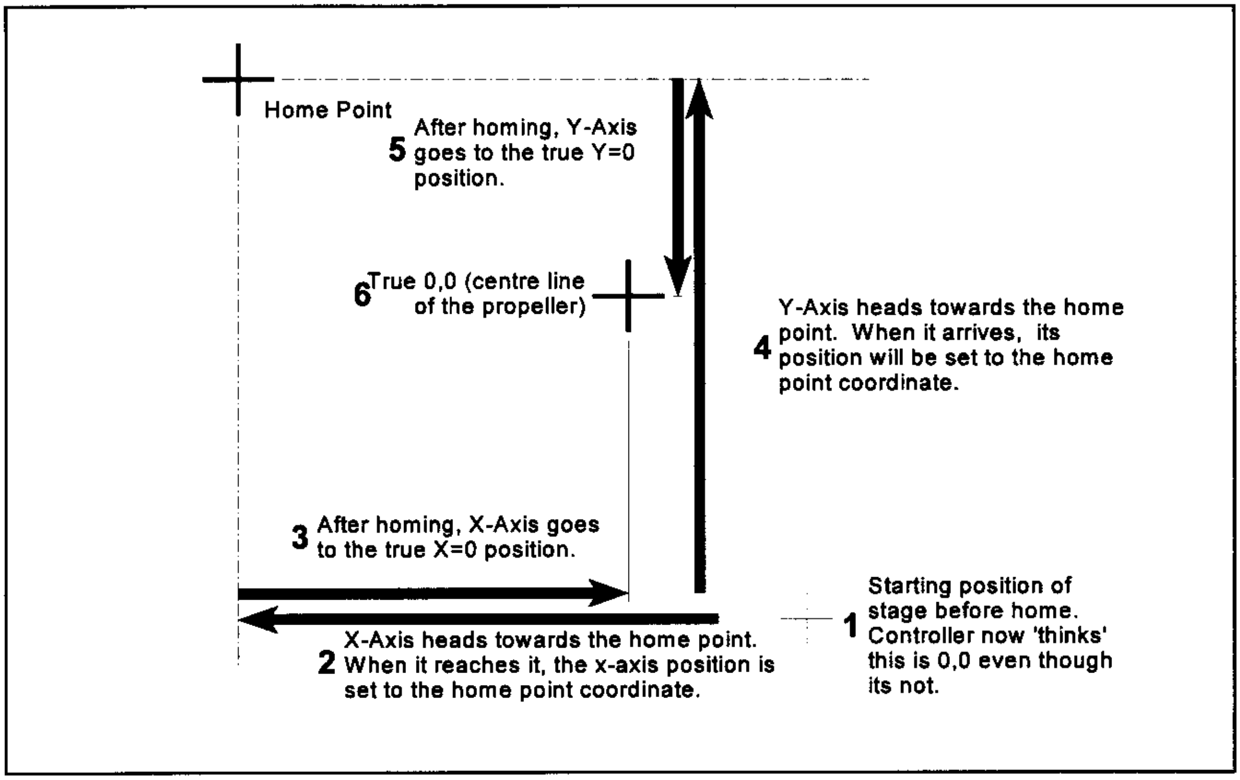
This will return to the *Main Menu* or to the *Execute .RUN File* screen depending on which routine called *Control Motor*.

4.8 Initialize and Home

Initialize and Home is how the positioning stage keeps track of its position. Two home switches are located in the upper left hand corner of the positioning stage, as shown in figure 4.10.



The Unidex motor uses a simple, hardware based homing system. The software jogs the frame towards the home point. When the home point switch is triggered the software readjusts its position reading and encoder reading to read the real world value of the home point. For example, assume that the propeller centreline is in the middle of the positioning stage. This corresponds to position 0,0 with reference to the propeller centreline. However, when the positioning controller is switched on whatever position it happens to be at will be called zero. For example, if the position was the lower right corner, the controller would call this 0,0 even though it not the propeller centreline. The home point position (upper left corner) has a position with reference to the real propeller centre line. Assume that it is -100,100 in the x/y from the 0,0 origin of the propeller centreline. When the stage is homed each axis will move to the home point. When they hit the home point their values will be reset to -100,100 which is the proper coordinate of the home point. Now, all other positions are referenced to the home point. The position 0,0 will now be the real propeller centre line. This idea is illustrated in figure 4.11.



`WAKE.BAS` has a built in homing routine and a built in calibration routine for determining what the starting value of the home point is relative to the propeller centre line. To home the positioning stage select *Initialize and Home* from the *main menu. Initialize and Home* can also be accessed from *Control Motor* and *Execute a .RUN File* command screens using the **[H] Home** option. The input/output window will clear and the follwoing prompt will be displayed:

```
Home will use WAKE_MM.CAL as the calibration file.
Press [N] if you wish to use a different one, [ESC] to quit home,
or press any other key to continue.
```

Calibration files contain the location of propeller centrelines. There would be one calibration file for every different model or set-up. These files are explained further in this section. Pressing **[N]** will prompt for the new calibration file to use. Pressing **[ESC]** will skip the initialization and homing and return to the previous screen, either the *Main Menu*, *Control Motor* or *Execute a .RUN File.* Pressing any other key will start the motor initialiization and homing process.

The first part of the process is intitialization. This resets the controller and its position and encoder counts. It also flushes all input and output command buffers, stops any currently executing commands, removes any unit scaling factors and erases all velocity and acceleration settings. The program will then proceed to set the units and velocities and acceleration. Once this is done, intitialization is complete. The program will display the message:

```
Initializing System....Done.
```

After initialization, the axes will be homed. They are homed sequentially to avoid possible damage to the probe because sometimes the home point itself can be too close to the model hull. The x axis is homed first. The program will display the message:

```
Homing X axis motor....
```

The x axis motor will automatically jog towards the x home point. The settings window will track the motor as it moves, but no plot is made to the probe position window. When it reaches the x home point it will reset itself to the required x value. It will then move to the x=0 origin position. During all of this procedure the y axis will not move. Once the x axis arrives at the origin the y axis will home itself and move to the y=0 position.

The initialization process has to be performed every time the positioning stage controller or computer is turned off and on. It should also be done occassionly during a series of tests. Homing should also be done whenever the encoder errors start to become too high. This will ensure accuracy in the positioning device.

Calibration of the unit is accomplished through calibration (`.CAL`) files. These calibrations contain the coordinates of home point with reference to the propeller centre line. These coordinates will change from model to model because they represent the relationship between the position of the stage and the position of the propeller centre line. A sample `.CAL` file is shown in figure 4.12.

```
** Quest Calibration - Starboard Propeller
**
** Units: mm      Date:03-15-1995     Time:15:30:46
X_HOME_POS    ,                -110.8476
Y_HOME_POS    ,                 82.20314
X_START_POSITION (After Homing),         0
Y_START_POSITION (After Homing),         0
```

Figure 4.11 - Typical calibration file.

The first three lines, preceded with '`**`' are comments containing the model name, date of the calibration and any other required information. The next two lines contain the x and y home positions, in this case x = `-110.8476` and y = `82.20314`. This means that with reference to the propeller centre line, the home point is at `-110.8576, 82.20314`. The last two lines indicate the starting position after a home. This specifeis the x and y position where the control will go after homing the axes, in this case to the origin, 0,0.

For the above calibration file, the following home steps would be followed by the software:

1) Home the x-axis. When the home switch is triggered make the x-position -110.8476.

2) Now that the computer knows the position of the x axis is a -110.8476 at the home point (where the probe is now), send the probe to x=0, the start point.

3) Home the y-axis. When the home swtich is triggered, make the y-position 82.20314.

4) Now that the computer knows that its at the home point, and the coordinate is 82.20314, send the probe to y=0.

5) Both the x and y positions are now at the true origin, 0,0 of the propeller centre line.

4.8.1 Calibration

The positioning system is calibrated using the **[C] Calibrate** option in the *Control Menu* screen (found through the *Main Menu*). Before any calibration can be done the positioning frame must be homed. After the stage has been homed, the probe should be positioned using the jog and the **[SPACE] go to position** controls in the *Control Motor* screen. Once the probe is lined up to the centre line (ie. the new origin) of the propeller press **[C]** to calibrate. The input/output window will clear and the program will prompt for the name of the calibration file:

```
Enter new calibration file name: █
```

After a valid file name is entered, WAKE.BAS will calculate the new position of the home point relative to the new origin. This will be displayed to the screen:

```
The new home point is now: X:  -125.7495mm  y:    94.8955mm
Press any key to continue...
```

Pressing any key displays another program prompt:

```
After regular homes the controller will send the probe to a starting
location. Typically, this would be the origin, 0,0.  However, you may
specify a different point as the start point.

Enter x positon of start point after homing [Default = 0.000]: █
```

This allows the specification of starting points other than zero. For example, if a shaft was present the starting point could not be the origin because the probe would hit the shaft. In these cases, the starting point would be specified somewhere other than the origin. This has no effect on the accuracy of the positioning system.

After entering tha x and then the y positions of the start point, the program will prompt for the entry of two comment lines to be added to the beginning of the calibration file. Once this is completed the computer will signal that the file has been saved. Then it will attempt to home the motors using the new calibration file:

```
The controller will now re-home the motors with the new calibration file
Press any key to start...
```

This will start the *Initialize and Home* screen using the new calibration file instead of the default calibration file.

▲ **WARNING!** After creating a new calibration file for a particular model, the `WAKE.CFG` file should be edited to change the default calibration file. This will ensure that every time the motor is homed, the proper calibration file is used.

4.11 Interactive Mode

This mode allows direct communications with the Unidex controller through its built-in ASCII command langauge. All communication is done through the input/output screen:

```
Interactive Mode is now engaged...press [ESC] to quit.
Note: Units used are currently: mm
WY
PC38 ver 1.60-4E
```

The units for all commands will be set to the current units setting unless a reset command is issued in the interactive mode. The `WY` command gets the board version number of the Unidex controller. Pressing **[ESC]** returns the program back to the main menu.

4.12 Set Acc. and Vel.

This command sets the x and y accelerations and veloccities used by the controller. The following menu will displayed in the input/output window:

```
Current Settings:
     Xvel    50.0000 mm/s
     Yvel    50.0000 mm/s
     Xacc   50.0000 mm/s
     Yacc   50.0000 mm/s
     Finished
```

Selecting any of the four settings will display a prompt next to the menu where a new value may be entered. Changing the setting to a value beyond the maximum rates of the controller will cause the setting to made it maximum value. Negative values will be taken as absolute values. The *Finished* option exits the *Set Acc. and Vel.* menu.

4.13 Set Units

This command sets the unit system used by `WAKE.BAS`. This will display a menu in the menu window displaying the various available unit system. Picking one of the unit systems will make it the active one. All other settings will then be automatically converted to the new unit system. Selecting *Finished* exits the unit selection menu without making a change.

5.0

Matlab WKS_FAIR Data Fairing & Plotting

Topics:

- ❑ Setting up Matlab and starting WKS_FAIR
- ❑ Loading and saving the data files
- ❑ Fairing curves through the data points
- ❑ Plotting the data
- ❑ Calculating the Fourier expansion for the data

A new data fairing and plotting routine has been developed for manipulating wake survey results. This routine was implemented in Matlab 4.1.1c for OpenVMS on the DEC Alpha platform. Matlab was chosen for many reasons: user interfaces are easy to set up and operate; it supports mouse input and multiple windows; it has convenient spline curve routines; it has excellent printer output and support and any routines written in Matlab for OpenVMS can be easily ported to other Matlab systems, such as Windows, Macintosh or Unix. Another important reason for choosing Matlab on the Alpha machines is that it keeps all of the wake survey analysis programs and procedures on one platform (as opposed to using Matlab for Windows).

5.1 Setting up Matlab and WKS_FAIR

Matlab can only be accessed on the DEC Alpha platforms (GNOME, GOLEM, GRYNCH, GHOST and TROLL). To set up Matlab on your account you need to add a [.MATLAB] directory to your account in your HOME directory:

```
$ home                              ! Go to your HOME directory
$ create /directory [.matlab]       ! Create the Matlab directory
```

Into this directory you need to create a `STARTUP.M` file. Files that end in `.M` are used as command files in Matlab. They are ASCII text files made up of Matlab commands. They are similar to VMS DCL `.COM` files and can only be run inside Matlab by typing the name of the file. The `STARTUP.M` file is automatically executed by Matlab every time you start matlab Matlab. It can execute any valid Matlab commands which you add to it. It is the Matlab equivalent to your `LOGIN.COM` file in VMS and it is your personal start-up file. This file can be created by using a text editor:

```
$ eve startup.m          ! Edit the file STARTUP.M. The editor will create
                           a blank file if STARTUP.M does not exist.
```

The editor will start up with the `STARTUP.M` file. Next, you need to add a path statement to the file using the editor. This statement tells Matlab where the various `.M` files used by `WKS_FAIR` are located. To set the paths add the following lines:

```
%
% WKS_FAIR and FAIR Paths
%
     path(path, 'wks:[fair]')     % Sets the WKS_FAIR path
     path(path, 'mfiles:[fair]') % Sets the FAIR path
     disp('WKS_FAIR and FAIR added to user search path')
```

The %'s are Matlab comments so the first three lines are only comments. The first `path` statement sets the path for the `WKS_FAIR` Matlab command files. The second `path` statement sets the path for the `FAIR` command files which are used by `WKS_FAIR` to do the actual spline curve fitting. The final statement is `disp` which prints the message '`WKS_FAIR and FAIR added to user search path`' when Matlab is first started.

▲ ***Warning!*** Matlab is case sensitive. That is, it distinguishes between upper and lower case letters. For example, `DISP` is not the same command as `disp`. All of Matlab's commands and the `WKS_FAIR` and `FAIR` commands use lower case letters. Also, Maltab always uses single quotes, such as `'` rather than double quotes `"`. Be careful when adding these lines to your `STARTUP.M` file.

Once these lines have been added to the `STARTUP.M` file it can be saved. Next, you must ask the system administrator to copy the file `MATLAB.DAT` to your `[.MATLAB]` directory. This file is used by Matlab to adjust the default settings of your Matlab sessions. This step is not necessary to run Matlab but it is a good idea to have it copied to your directory anyway.

The final step before using Matlab is setting the display. If you are on a local node, such as sitting in front of the machine you are logged on to, you do not need to set the display options. However, if you are using a remote node you must set the display and security options. For example, if you are on the local node MAROON, (ie. sitting in front of it) and are logged into GNOME, you must set the display in the GNOME window to MAROON:

```
GNOME $ set display/create/node=MAROON     ! Set display node to MAROON
```

This sets the display to MAROON so that any graphics windows from GNOME will be output to your screen on MAROON. **NOTE:** The security option under your session manager on MAROON has to be adjusted to allow output from your user name on GNOME to appear on MAROON.

Now Matlab can be started from any directory. For example, if you wanted to analyze the data in the directory where you have just finished analyzing a test you would simply type `matlab` at the VMS command line. When Matlab starts, the title will be displayed in the terminal window. Then a small graphics window will displayed with the Matlab logo. Finally, after a short wait, the following lines will be displayed:

```
Commands to get started: intro, demo, help help
Commands for more information: help, whatsnew, info, subscribe

WKS_FAIR and Fair added to user search path
>>
```

You will now be at the Matlab command prompt. At this prompt you can run any Matlab `.M` file by simply typing its name.

5.2 Starting WKS_FAIR

Once Matlab is running, WKS_FAIR can be started:

```
>> wks_fair
```

This will clear the command screen (the main terminal screen where you started Matlab) and will display the `WKS_FAIR` title screen. Also, in the upper right corner a graphical, push button menu will appear. The menu options are:

`Load Wake Survey Data File`

`Save Faired Data in .FRD File(s)`

`Fair Data`

`Clear All Data`

`Plot Data`

`Fourier Transform`

`Finished`

Any of these options can be selected by clicking on the menu item. In the screen another set of lines showing which wake survey data files have been loaded is displayed:

```
Currently Loaded Data Files:------------------------------
No Files Currently Loaded
----------------------------------------------------------
```

This menu is called the *Main Menu*. It is used to access all of the main functions of `WKS_FAIR`. The analysis and plotting procedure is outlined in sections 5.3 to 5.10. As each option is being used, simple instructions will be displayed in the main control window.

▲ ***NOTE:*** `WKS_FAIR` does not erase any data when it is started. Therefore, if you cause and error while using and it sends you back the >> matlab prompt, you can start the program again using `WKS_FAIR`. All of the data files and analysis you have done so far should be saved and you continue where you left off. Errors can often occur for simple reason like not selecting a filename when loading a file or by entering a character rather than a number to a prompt.

`WKS_FAIR` uses a spline curve fitted through a series of control points. These control points can be manipulated and saved for each data file.

5.3 Load Wake Survey Data File

WKS_FAIR is used to analyze the output radii files from the WAKE_SORT_RADII VMS command. Each file contains the axial, tangential and radial velocities and angular position of the sample points for one radius. These individual radii files need to be loaded one by one into Matlab. This is done using the Load Wake Survey Data File option from the *Main Menu*. Selecting this command will bring up a VMS file requester window and cause the *Main Menu* to disappear. This windows can be used to load the required data file by double clicking on the filename or by clicking on the file name and pressing OK. The file requester will be filtered by the *RAD*.DAT filter so that only files with the word RAD and the extension .DAT will be shown. This would show the radii files created by WAKE_SORT_RADII which outputs its data using the _RAD*n*.DAT addition.

Notice that before while the file request window is displayed, the following message is displayed in the command window:

```
Pick the raw data file *RAD*.DAT.  If a faired data file (*.FRD) exists
the program will load that, if requested.
```

WKS_FAIR stores the control points to the spline curve fits to the data in faired data files, which have the extension .FRD. If WKS_FAIR detects one of these files for this particular data file it will ask you if you want to load it:

```
Found a faired data file (.FRD) for this data.
Would you like to load it? [Default = Y]:
```

The existence of a faired data file means that the data has previously been loaded into WKS_FAIR and a faired data file saved for it. If no faired data file exists for this file, the control points are set to be the same as the data points.

After the file has been loaded, the velocities versus theta will be plotted to a new figure window and you will be returned back to the *Main Menu*, which will reappear. In the command, the name of the file you just loaded is displayed under the Currently Loaded Data Files header.

Repeat the load process until all of the required data files have been loaded. In general, if you were analyzing a test you would load in the data for the test radii. You should start by loading radii 1, the largest radii, then load radii 2 and so on. However, you could also load in data for the same radius but at different model speeds. This is useful for seeing the differences caused by velocity changes. All of the analysis techniques described below can be used with data for the same radius but different velocities. However, this section uses the example of different radii at the same speed. Generally, curve fairing would be done using this type of data set. Once the curve fairing is done and the control points saved in a faired data file, these files can be loaded at any time and they will not need to be faired but they can be plotted against each other and so on.

5.4 Save Faired Data in .FRD File(s)

This option will save the current control points to all of the currently loaded data files. A separate `.FRD` file will be created for each loaded data file. After you have faired your data you should saved the control points using this procedure so that the next time you can load them in with the `LOAD` menu option. The command windows will indicates that all of the faired data files have been saved for each data file (if you have not faired some of the data the control points will be saved as the data points).

5.5 Fair Data

The *Fair Data* option allows the user to interactively control a spline curve fit through the data by adjusting the control points of the spline. When selected from the *Main Menu* another menu will appear with the names of the currently loaded data files (except if only one data file is loaded, in which case the program will skip to the next menu). Pick the name of the data file you want to fair. Then another menu will appear with the following options:

```
Fair Vx/Vt Wake Data
Fair Vt/Vs Wake Data
Fair Vr/Vs Wake Data
Finished
```

This allows you to choose which velocity you want to fair. Generally, you fair all of them one by one. After you finished fairing one you will be returned to this menu. You can fair another one or choose `Finished` to return to the *Main Menu*.

When you fair one of the velocities the velocity versus theta plot will appear in a new figure. The above menu will replaced by a new one:

```
Redraw
Zoom In
Zoom Out
Zoom Set
Move Points
Delete Points
Insert Points
Set Slope
Finished
```

This menu is used to perform all of the manipulation of the spline. The figure window with the data being faired will contain cyan '+'s for the actual data points and white 'x's for the spline control points. If it is the first time fairing this particular data, the spline control points and the data points will be the same and will appear as white '*'s on the screen. The control points ('x's) can be moved and changed to change the spline curve but the original data points ('+'s) cannot be changed.

Redraw

This command will recompute the spline through the control points. As changes are made to the control points, the spline is not automatically recomputed because it would take too long. Therefore, use `Redraw` when you want to see the effect of any changes you have made. `Redraw` will not change the scale of the drawing.

Zoom In:

Allows the user to zoom in on the data. When zoom is selected, pick any two points on the plot, using the *left mouse button*, to create a box which will be the new limits of the plot. When the mouse pointer is on the plot figure a cross will appear. As you pick each point a small white dot will be placed there. When you have picked the second point the plot will zoom to the window you selected. You can continue to zoom in on the data as much as is necessary.

Zoom Out:

Will zoom out to show the entire plot. It is an auto scaling function.

Zoom Set:

This command allows the user to enter the x and y maximums and minimums of the plot. It will then expand the plot to the new limits. The limits are enter through the keyboard and the command window. The defaults for each prompt would be the current x and y limits.

Move Points:

This command allows you to move any of the control points in the y-direction only. First, pick the control point you want to move using the *left mouse button.* (Pressing the *right mouse button* while positioned on the plot figure will return you to the menu without moving any points) When the mouse pointer is on the figure plot it appears as a cross. Move the pointer to the point you want to move (you don't need to click exactly on it, just close enough). A small, white circle should appear around the control point you selected (Remember, you can only move the control points, which are the 'x's). Next, pick the new y-position for that point. The program will ignore the x coordinate of the position you pick so you only need to worry about the new y position. After you move a point, you can move another by clicking near the new point with the *left mouse button* and then you can pick a new position for that point. When you are finished moving points, click the *right mouse button* while positioned on the figure plot. This will return you to the menu.

Delete Points:

This command will delete any user selected control points. When you select the command, pick the point you want to delete using the *left mouse button*. A small white circle will appear around the point. If you want to delete that point press the *left mouse button* again and the point will deleted. If you don't want to delete the point, press the *right mouse button* and the point will not be deleted. Then another point can be selected using the *left mouse button* again. Or, if you are finished, use the *right mouse button*.

Insert Points:

This command will new control points at positions specified by the mouse. When this option is selected you can insert a point anywhere on the figure plot using the *left mouse button*. Points can continue to be inserted until the *right mouse button* is pressed, which will return you to the menu.

Set Slope:

This command allows the user to set the slope at the two endpoints using either graphical mouse input or by entering the slope as a number. For wake survey data, the end points will have the same value and the same slope because they are the same physical point. Therefore, you only need to specify one slope and the program will apply it to both end points.

When selected, the following prompt will appear:

```
[E]nter slope or pick with [M]ouse ? [Default = M]:
```

Entering 'E' will bring up a prompt asking for the new slope as a numerical value. Entering 'M' allows you to use the mouse to pick the slope:

First, pick the first point of the slope on the figure plot. A small white circle should appear. Then pick the second point of the slope. Another white circle will appear and the two circles will be connected making a slope. The slope will be calculated and applied to both end points.

Finished:

Returns you back to the '`Which data to fair`' menu.

5.6 Clear All Data

This command clears all of the data files and control points in memory and allows you start over again. It will prompt you to make sure you want to erase the memory. After using this commands, data files will have to loaded again through the *load* command in the *Main Menu*.

5.7 Plot Data

This is the main data plotting routine. It can be used to plot either faired or unfaired data. That is, you can load the data using the *Load* option and go straight to the plotting routine without ever fairing the data using the *Fair* option, meaning it would be unfaired data.

When the *Plot Data* option is chosen, another menu will be displayed:

```
Plot Vx / Vs
Plot Vt / Vs
Plot Vr / Vs
Plot Taylor Wake Fraction
Vector Plot of Vt / Vs and Vr / Vs
Contour Plot of Vx / Vs
Print a Figure
Finished
```

5.7.1 Plot Vx / Vs, Plot Vt / Vs, Plot Vr / Vs and Plot Taylor Wake Fraction

All of these options will produce a plot of the parameter versus the angle theta for each different radii, for example Vr / Vs versus theta. First, a prompt will be displayed asking whether the plot will be printed as a portrait or landscape plot. A simple two dimensional rectangular plot will then be produced in a new figure window showing both the actual data and the spline curves. When the plot is done another menu called '`Add Titles and Notes`'. This menu allows you to annotate the graph and add legends and notes. It is explained in detail in section **5.7.5**. When the graph is finished, it can be printed using '`Print a Figure`', explained in section **5.7.4**.

5.7.2 Vector Plot of Vt / Vs and Vr / Vs

This option will produce a vector plot of the Vt/Vs and Vr/Vs velocities at the various points around the propeller. When first started, it will ask you if the plot will be printed as landscape or portrait. Then it will ask what the degree spacing will be between each vector along the axis. Usually, 15 degree spacing is used:

```
Enter degree spacing for vectors [Default = 15]:
```

Finally, it will ask you for the scaling factor. This is the scale factor relating the size of the vector arrows to the size of the plot. The larger the scale factor, the larger the vector arrows will be for a given velocity. Generally, for tests with low values of tangential and radial velocities, `300` is a suitable scale factor.

The program will now generate the vector plot. Red vectors will be plotted at the angle interval around each radius.

The program will now prompt with:

```
Enter font size in points [Default = 10]:
```

This is the font size that will be used for any labels added to the graph.

```
Add radii circles to the graph? (Y/N) [Default = Y]:
```

This will plot a circle at each radii in the test data.

```
Add angle labels to the graph (Y/N) [Default = Y]:
```

This will label the angle of each vector position around the circumference of the largest circle.

```
Add angle partition lines to graph (Y/N) [Default = Y]:
```

This will draw a partition line from the smallest circle to the largest circle at the angles of each vector location.

```
Pick position for scale symbol to go.
```

Now the mouse can be used to pick a position for the scale symbol to be placed (usually the lower, right corner of the plot). The scale symbol is a vector of a given length drawn on the plot to indicate the scale of the vectors. Once you have picked the position, the velocity of the scale symbol is required:

```
Enter the reference value (ie. scale will plot an arrow
for the velocity you specify) [Default = 0.1]:
```

Usually 0.1 is a good scale value. The program will plot an vector for you velocity and underneath it, it will put the text V/Vs=0.1, or whatever velocity you specify.

The vector plot is now finished. The '`Add Titles and Notes`' menu will now appear, which allows you to annotate the graph and add legends and notes. It is explained in detail in section **5.7.5**. When the graph is finished, it can be printed using '`Print a Figure`', explained in section **5.7.4**. **Do not use the *Legend* command for vector plots. To list the radii use the *Add Note* command and add the data descriptions as follows:**

```
Radii:          ! Note Title
r/R = 1.10      ! Note Line 1
r/R = 0.90      ! Note Line 2
r/R = 0.70      ! Note Line 3
r/R = 0.50      ! Note Line 4
r/R = 0.30      ! Note Line 5
```

5.7.3 Contour Plot of Vx / Vs

This command performs contour plots of either Vx/Vs or Taylor wake fraction. Contour plots require at least two loaded data files. This file make an unevenly spaced grid of data points from the spline curves and interpolates them to a rectangular grid that Matlab can use to create contour plots with.

First, the program will ask whether the file will be printed as a portrait or landscape. Then it will ask if you want to plot Vx/Vs or the Taylor Wake Fraction. Next it will ask for the degree spacing for the theta interpolation:

```
Enter degree spacing for theta interpolation [Default = 10]:
```

This determines the angular spacing between each interpolation line on the input grid. The lower the number the better the interpolation but the longer the interpolation will take. Generally 10 provides a good trade-off between accuracy of the interpolation and speed.

```
Enter number of interpolated radii [Default = 9]:
```

Similar to theta spacing above, but determines the number of interpolation points in the radial direction of the input grid. The higher this number, the better the interpolation but the longer the interpolation will take. 9 is a good tradeoff value.

```
Number of nodes [Default = 100]:
```

This determines the number of nodes in the rectangular output grid. More nodes will provide a finer interpolation but computation time will rise. 100 is a good tradeoff number.

The program will begin to compute the contour. When it is finished it will show the following prompt:

```
Set custom contour levels (Y/N) [Default = N]:
```

This would allow you to set whatever contour levels you want. However, the default contour levels are set as 0.05 and work well for most applications, so usually you would answer 'N' to this question. The program will then plot the contours to a new window. Another series of prompts will appear:

```
Enter label font size in points. [Default=10]:
```

This is the font size that will be used for any labels added to the graph.

```
Add radii circles to the graph (Y/N) [Default = Y]:
```

This will draw circles at each radius in the test data.

```
Add angle labels to the graph (Y/N) [Default = Y]:
```

This will add angle labels at each angle around the circumference of the circle. Answering 'Y' and the program will prompt for the angular spacing of the labels and then apply the labels to the graph. This will also eliminate the x and y axis from being drawn on the graph.

```
Add angle partition lines to the graph (Y/N) [Default = Y]:
```

This option will add partition lines from slightly larger than the largest diameter to slightly smaller than the smallest diameter. Answering 'Y' to this question will display another prompt asking for the angular spacing of the partition lines.

```
Enter CONTOUR label font size in points. [Default=8]:
```

This is the font size for the labels applied to each contour label. After it is entered the program will enter the contour label mode. This allows you to label each contour line with it value. Pressing the *middle mouse button* at the end of a contour line will place the value directly at the end of the contour line. Pressing the *left mouse bottom* anywhere on the contour line will go to the arrow define mode. That is, the point you just picked on the contour line will be the head of an arrow. Next pick the end of the arrow in the appropriate place. Finally, pick the end of the flat leader line either to the left or the right of the arrow. The text will be placed accordingly. Press the right mouse button or the press the ENTER key to exit the contour define mode. **Be very careful: Once you exit this mode you cannot go back.**

The contour plot is now completed. The '`Add Titles and Notes`' menu will now appear, which allows you to annotate the graph and add legends and notes. It is explained in detail in section **5.7.5**. When the graph is finished, it can be printed using '`Print a Figure`', explained in section **5.7.4**. **Do not use the *Legend* command for contour plots. To list the radii use the *Add Note* command and add the data descriptions as follows:**

`Radii:`	! Note Title
`r/R = 1.10`	! Note Line 1
`r/R = 0.90`	! Note Line 2
`r/R = 0.70`	! Note Line 3
`r/R = 0.50`	! Note Line 4
`r/R = 0.30`	! Note Line 5

5.7.5 Print a Figure

This command will print any figure to a PostScript printer. Currently, the software will only print to LASER7, the PostScript printer in the first floor computer room. This can only be changed by the system manager.

When the command is first issued, the current printer name will displayed:

```
Note: The figure will be sent with the following command:

PRINT/DELETE/QUEUE=LASER7

If you wish to change this command line you must edit your
printopt.m command file.
```

```
Do you wish to continue printing? (y/n) [Default = Y]:
```

This shows the command line which will send the figure to the printer LASER7. The printer must be a PostScript printer. Currently the only available PostScript printer at IMD is LASER7. To change the printer, the PRINTOPT.M in the WKS:[FAIR] applications directory has to edited in the applications directory. This can only be done by the system manager. However, there is no reason to change this because there are not other suitable printers that can work with the VMS version of Matlab.

If you decide to continue printing, another prompt will appear:

```
[P]rint, [S]ave as print file or [B]oth? [Default=P]:
```

This allows you to print the figure directly to the printer, or save it as print file or do both. WKS_FAIR does not let you save your figures. If any editing is required at a later date the data files and the faired data files would have to be loaded in again and the plot regenerated. Therefore, it may be convenient to save the files as print files so they can be printed at any time after.

Next the program will prompt for the figure number to print. Enter the figure number of the plot you wish to display and the figure will become the current window. The computer will then prompt for the orientation of the plot, either landscape or portrait. This should be the same as the orientation you entered when you first generated the plots.

If you chose to print the file (or to print and save it) the following notice will appear:

```
Job started on LASER7
```

This indicates that WKS_FAIR has sent your print to the printer. If you chose to save the file as a print file (or save it and print it) another prompt will appear:

```
Enter file name for the printer file:
```

When you enter the name, do not use an extension. WKS_FAIR will add the extension .EPS to the file. this indicates that it is an enhanced post script file. For example, if you enter VS10_CONTOUR as the file name it produce a file called VS10_CONTOUR.EPS. To print this file use the following command:

```
$ print /queue=laser7 vs10_contour.eps
```

After the file has been printer and/or saved, you will be returned to the main plot menu.

5.7.6 Apply Notes and Labels to the Plot

This command menu will be displayed after every plot. It allows you to change the plot title, axis labels, add notes, and annotations to the splines. The menu will be displayed as follows:

```
Add Title
Add X Label
Add Y Label
Change Scale
Add Note(s)
Move Last Note
Add Last Legend
Move Last Legend
Annotate Spline(s)
Finished
```

Add Title:

Changes the title of the plot. Enter the new title at the prompt and it will become the new graph title.

Add X Label:

Changes the x label of the plot. Enter the new x label at the prompt and it will become the new x label. (The x axis is Theta, measured in degrees)

Add Y Label:

Changes the y label of the plot. Enter the new y label at the prompt and it will become the new y label. (The y axis is one of the velocities, which are non-dimensional)

Change Scale:

This option allows the user to set the maximum and minimum x and y values. The plot will be scaled to these new settings.

Add Note(s):

Allows the user to enter a multi-line note on the plot. First, you will prompted for the font size of the note in points. Then, enter the title of the graph or press [ENTER] for no title. The computer will then ask you to pick the location of the note on the plot. The location is where the title will go or the first line of the note if there is not title. All of the other lines of the note will be displayed under the first line. After you pick the position, the title will be displayed, if you entered one. Next, you will be prompted for the first line of the note. After entering the line it will appear on the plot. You will then be prompted for the next line. This process will continue until you press enter on a blank line. After the note is finished being entered, you will be returned to the add notes menu. You can add as many notes as you want.

Move Last Note:

This command allows you to change the position of the last note that you entered. When you choose this command, first pick a base point near the note. Then choose a new position for this base point and the note will be moved from your first base point to this new base point.

Add Legend:

Allows you to add a legend indicating which raw data points belong to which data set. First, you have to enter the font size of the legend in points. Then, you have to enter the title of the legend (or press [ENTER] for none). Now, pick the position for the legend location on the plot. The title, if there is one, will be drawn on the plot. Next, the program will display the first file name in the series. Enter the legend description for this file name and it will be entered into the computer. For example, if the file name was `WKS_TEST_RAD2.DAT` and you knew that this was for the data r/R = 0.90 then you would enter `r/R = 0.90` as the legend description. This process repeats for all of the data files loaded into `WKS_FAIR`.
Should not be used for contour and vector plots.

Move Last Legend:

This command allows you to change the position of the last legend that you entered. When you choose this command, first pick a base point near the legend. Then choose a new position for this base point and the legend will be moved from your first base point to this new base point.

Annotate Splines:

This option allows you to add arrow labels to the data splines. First, you will be prompted for the label font size, in points. Then the data file name for the first spline will displayed. This spline will also be turned dark blue. You can now enter the name of description for this file name, such as `r/R = 0.50` Then, using the mouse, you have to pick a position on the spline curve for the arrow head to be located. Next, pick two positions which determine a horizontal line which will be drawn next to the label text.

Finished:

Returns to the plot data menu.

5.8 Perform Fourier Expansion

This module will compute the standard Fourier series expansion to the spline curves for each of the velocity. When first started, the command will display a menu of all of the currently loaded data files. Pick the file you want to find the Fourier expansion of. You will then be prompted for the number of terms in the expansion. The more terms will yield a better approximation. Generally, 30 to 35 terms will provide a reasonable fit to the curves. The Fourier expansion will then be computed. When it is completed, the spline curves for the axial, tangential and radial velocities will be plotted on a new window. The Fourier expansion curves will also be plotted as well. If the expansion curves are good fits, they should all appear over the top of each other and look like only three lines, rather than 6 lines.

After the curves have been plotted to the window, the program will display percent error statistics for each curve fit. It will then ask you if you want to plot the percent error as a function of the angle theta for each fit. Answering 'YES' will produce another plot of the percent error against the angle theta. The program will ask if this is an acceptable fit. If it is not, the program will ask for a new number of terms for the Fourier expansion and the process will repeat itself. If it is acceptable, you will be prompted for the name of the output file. This will be an ASCII file which contains the Fourier expansion coefficients for the axial, tangential and radial velocities. After the file has been saved, you will be returned to the main menu.

6.0

Wake Survey VMS Calibration Software

Topics:

- ❑ Setting up for calibrations
- ❑ Using the VMS Wake Survey Calibration Software
- ❑ Smoothing and storing the calibration tables

Wake survey calibrations are performed by moving the probe through combinations of yaw and pitch angles and outputting calibration tables of the Q, R and Pcal parameters. The wake survey calibration programs use the name WAKE_CAL_xxx for all command names. Many of these commands and procedures are similar to the standard wake survey test procedure.

This explanation is set up as a tutorial, unlike the other sections of this manual. For our example, we will calibrate the conical head probe number 3, or C3. We will use the 2-psi transducers.

First, you must add the command:

```
$ WAKE_CAL_START :== @WKS:[CAL]WAKE_CAL_START
```

to your `LOGIN.COM` file in your home directory. After you make the change log in to the machine again to make the changes active.

First, we must make a directory to perform the data analysis in. Assume we are in a main directory call `[MAIN]`:. We can create a sub-directory here:

```
$ create /directory [.wks_cal_c3]          ! Create a storage directory
```

This will create a directory in the `[MAIN]` directory. Next, change to this directory and create another directory to store the temporary raw GEDAP files:

```
$ cd [.wks_cal_c3]
$ create /directory [.raw]       ! Create another directory for the raw data
```

Next, you can start the wake calibration software by typing:

```
$ WAKE_CAL_START             ! Start the wake survey calibration software
```

This will copy the file WAKE_CAL_SET.COM to the current directory, `[.WKS_CAL_C3]`. This file contains all of the program settings for the wake survey calibrations. Edit it by typing:

```
$ eve wake_CAL_set.com      ! Edit the settings file
```

This file is almost identical to the WAKE_SET.COM file used for wake survey tests. While in this file you can edit the following global variables:

Set the raw location of the permanent raw DAS or GEDAP files:

```
WAKE_SOURCE_LOCATION :== TOW
```

Set the test name of the permanent raw DAS or GEDAP files:

```
WAKE_TEST_NAME :== WAKECAL
```

Set the storage directory for the temporary raw GEDAP files. They will be stored in the `[.RAW]` directory we created above:

```
WAKE_STORAGE_DIRECTORY :== [MAIN.WKS_CAL_C3.RAW]
```

Set the printer where any raw channel plots will be sent. Use the name NONE for no plots:

```
WAKE_PRINTER :== NONE
```

Set the name of the output file root name. All output files such as the STATS file and the LOG file will have this name plus `_STATS.DAT` or `_LOG.DAT`:

```
WAKE_CAL_OUTPUT_FILE :== WKS_CAL_C3
```

Set the data stream numbers for each channel:

```
WAKE_CARRIAGE_SPEED_CHAN :== 7
WAKE_PROBE1_PRES_CHAN :== 1
. . . . . .
WAKE_PITCH_CHAN :== 8
WAKE_HEIGHT_CHAN :== 6
```

Set the list of channels used. Channels can be excluded:

```
WAKE_TOTAL_CHANNELS :== 1-9
```

Set the total number of channels:

```
WAKE_NUMBER_OF_CHANNELS :== 9
```

Once all of these settings have been chosen, exit EVE using *ctrl-Z*. Now, purge the directory and run the WAKE_CAL_SET.COM file by typing the command at the prompt:

```
$ purge            ! Purge old versions
$ wake_cal_set     ! Make the settings active
```

Now, we can convert any of the DAS[1] runs into GEDAP runs and store them in our `[.RAW]` directory:

---

[1]*Programmers Note:* Presently, the data files are converted from DAS format to GEDAP using DAS_GEDAP and placed in sub-directories of the raw data directory. However, if the data was acquired in the GEDAP format, a utility would be required to copy these GEDAP files from TOW or the carriage acquisition computer to this raw storage directory in the same format which DAS_GEDAP writes them. This is how the wake survey calibration software expects to find the data.

```
$ wake_cal_convert 24 45          ! Convert runs 24 to 45
```

This will convert the data files from DAS to GEDAP and store them in sub-directories in the [.RAW] directory. Now, the calibrations runs can be analyzed using:

```
$ wake_cal_run 24 45              ! Analyze runs 24 to 45
```

This will start the main analysis program. First, you pick the time segments using the interactive segment selection utility. Make sure that the first segment you pick is the zero tare segment (carriage speed is zero). The program will begin to tare the data. Next it will ask you for the nominal yaw and pitch angles for each segment. It will display the actual yaw and pitch readings for each segment so that you can have an idea of what the yaw and pitch were:

```
The actual yaw angle was -24.667 and the pitch angle was +14.857.
Enter nominal yaw angle: -25
Enter nominal pitch angle: 15
You entered -25 for the yaw and 15 for the pitch.
Is this correct (Y/N): YES
```

After you enter each set of nominal angles, the program will confirm if you are sure they are right. This helps prevent data from going to the wrong location in the calibration tables. The wake survey calibration tables are indexed by nominal yaw and pitch, so entering the wrong set of angles will make the tables erroneous. If you enter a yaw and pitch angle that are not in the table matrix, the data will be saved in the stats file for future use, but it will not be put into the array tables.

After you enter the nominal angles for yaw and pitch the calibration tables will be created. There will be three calibration table files for the Q, R and Pcal parameters and two files for storing the actual yaw and pitch angle for each set of nominal angles. They are named using the WAKE_CAL_OUTPUT_FILE symbol named defined by WAKE_CAL_SET.COM:

```
WKS_CAL_C3_Q.001          ! GEDAP array file for the Q parameters
WKS_CAL_C3_R.001          ! GEDAP array file for the R parameters
WKS_CAL_C3_PCAL.001       ! GEDAP array file for the Pcal parameters
WKS_CAL_C3_YAW.001        ! GEDAP array file for the actual yaw angles
WKS_CAL_C3_PITCH.001      ! GEDAP array file for the actual pitch angles
```

These a GEDAP array files indexed by the nominal yaw and nominal pitch angles. Their contents can be checked using the GEDAP command list:

```
$ list wks_cal_c3_q.001   ! View the GEDAP array file
```

As you continue to analyze runs during the wake survey calibration, these files will be continually built up.

When you first start the analysis the program will automatically create the table files. You will be asked for the sizes of the nominal yaw and pitch grid being used for the calibration:

```
Enter first angle in Yaw matrix in degrees: -40
Enter the Yaw step in degrees: 5
Enter the number of Yaw angles: 17
Enter first angle in Pitch matrix in degrees: -40
Enter the Pitch step in degrees: 5
Enter the number of Pitch angles: 17
```

In this above example, both the yaw and pitch angles of the matrix start a -40 degrees and extend in 17, 5 degree increments up to +40 degrees.

At any time you can view the list of programs you used by:

$ wake_cal_log ! View the LOG file

You can also view the run statistics for the calibration using the command:

$ wake_cal_stats ! View the STATS file

This statistics viewer is not as sophisticated as the one used for wake survey tests, WAKE_STATS. It does not tell you what each parameter is. It merely outputs the raw STATS file. The format of the file is shown below:

```
**Run Number**Segment number*******************************
Run number, Segment Number, Nominal Yaw, Nominal Pitch, T1, T2, T3, T4
Chan No, Chan Mean, Chan Max, Chan Min, Chan Std Dev, Chan RMS
Chan No, Chan Mean, Chan Max, Chan Min, Chan Std Dev, Chan RMS
 .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .
Chan No, Chan Mean, Chan Max, Chan Min, Chan Std Dev, Chan RMS
1, Q, R, PCAL, Theta_pitch, Phi_Roll, Cp1, Cp2, Cp3, Cp4, Cp5
```

T1, T2	- Time segment for the zero tare segment.
T3, T4	- Time segment for the anlysis segment
Chan No - Chan Rms	- Raw channel statistics for all acquired channels
Q, R and PCAL	- Table values for this segment
Theta_pitch - Cp5	- Intermediate calculated values.

If at any time something goes wrong the data can be reanalyzed without having to re-select the time segments and enter the yaw and pitch angles. First, you should delete the STATS, LOG and GEDAP array storage files using:

$ wake_cal_reset ! Will reset the test and delete the output files

This command will make back-up copies of the output files with the addition of _OLD.DAT and then delete the output files. After the output files have been deleted, the runs can be re-analyzed:

```
$ wake_cal_rean_run 24 35        ! Re-analyze runs 24 to 35
```

This will rebuild all of the data files.

When the test is completed and all of the GEDAP array tables have been filled, the data needs to be converted into an ASCII file that can be input into Matlab for data interpolation. This is done using the command:

```
$ wake_cal_gedap_matlab          ! Create an ASCII file for use with Matlab
```

This will create the file WKS_CAL_C3_MATLAB.DAT. This data can now be used with a special Matlab data interpolation routine called WAKE_CAL_SMOOTH.M. In order to use this command you must add the following line to your STARTUP.M file in your home drive:

```
path(path, 'wks:[cal]')    % Set path for the WAKE_CAL_SMOOTH command
```

Now, start Matlab in the directory where the calibration data files you have collected are:

```
$ matlab          ! Start Matlab in the current directory
```

Then, start the smoothing and interpolation program:

```
>> wake_cal_smooth          ! Start the smoothing Matlab command file
```

A file requestor window will appear. Pick the ASCII file we just created, WKS_CAL_C3_MATLAB.DAT. The program will load this file in. Then, it will ask you what the output interpolation matrix will be:

```
Enter first yaw angle in output, smoothed matrix: -40
Enter the Yaw step in output, smoothed matrix: 1
Enter number of Yaw angles in output, smoothed matrix: 81
Enter first pitch angle in output, smoothed matrix: -40
Enter the Pitct step in output, smoothed matrix: 1
Enter number of Pitch angles in output, smoothed matrix: 81
```

This is the spacing of the output, interpolation grid. Generally, a 1 degree spacing grid is adequate. The program will then produce three dimensional surface plots of the three calibration parameters. It will then ask for the file names of the output ASCII files of the interpolated data. Choose the defaults output file names:

Q Parameter File:	WKS_CAL_C3_Q.DAT
R Parameter File:	WKS_CAL_C3_R.DAT
Pcal Parameter File:	WKS_CAL_C3_PCAL.DAT

These output files are then converted into GEDAP array files which will be the final array files. To do this use the command:

```
$ wake_cal_matlab_gedap    ! Convert ASCII Matlab Output to GEDAP files
```

The program will start by prompting for the yaw and pitch grid values of the smoothed, output data:

```
Enter first yaw angle in output, smoothed matrix: -40
Enter the Yaw step in output, smoothed matrix: 1
Enter number of Yaw angles in output, smoothed matrix: 81
Enter first pitch angle in output, smoothed matrix: -40
Enter the Pitct step in output, smoothed matrix: 1
Enter number of Pitch angles in output, smoothed matrix: 81
```

It will then ask for the names of the Matlab ASCII output files for each calibration parameter:

```
Name of Q parameter ASCII Matlab file: wks_cal_c3_q.dat
Name of R parameter ASCII Matlab file: wks_cal_c3_r.dat
Name of Pcal parameter ASCII Matlab file: wks_cal_c3_pcal.dat
```

Then, it will ask you for the root name of the output GEDAP calibration parameter files:

```
Name of output file for parameters: wks_c3
```

Always enter a name that is different from the `WAKE_CAL_OUTPUT_FILE` name defined in your `WAKE_CAL_SET.COM` file. If you use the same name, it will erase some of the original data files. This will output three new files:

```
WKS_C3_Q.001          ! Q Parameter calibration table file
WKS_C3_R.001          ! R Parameter calibration table file
WKS_C3_PCAL.001       ! Pcal Parameter calibration table file
```

This is the final calibration table for this calibration. It should be put into the `WKS:[TABLES]` directory by the system manager (only he has access to this directory). Now, if you want to use this calibration file for a wake survey test, you would specify this file name as your calibration file in your `WAKE_SET.COM` file:

```
WAKE_CALIBRATION_FILE :== WKS:[TABLES]WKS_C3
```

APPENDIX A

Wake Survey Analysis Software (V3.0) Release Notes

To: Spence Butt
Dave Cumming
Pat Dunphy
Keith Mews
Bud Mesh
Antonio Simoes Re
Michael Sullivan
Brian Veitch
Andy Wallace

From: Lawrence Mak

Re: Wake Survey Analysis Software Version 3.0

Date: October 10, 1997

Wake Survey Analysis Software (Version 3.0) is now released. With this version, fairing and 3D plotting can be conducted on both Alpha's and PC's using Matlab Version 5.

I am enclosing herein the following release notes for versions 3.0 and 2.1 for your information:

Version 3.0
This version upgrades the fairing and 3D plotting routines in Matlab. The GEDAP analysis routines are the same as in Version 2.1.

Version 2.1
This version upgrades the wake survey analysis software in GEDAP.

Release Notes for Version 3.0 Wake Survey Analysis Software
by L.M. Mak (NRC) October 8, 1997

1.0 Introduction

A new version of wake survey analysis software (version 3.0) has been installed on both KNARR (IMD's PC server) and Mickey (IMD's Alpha server). This enables wake survey analysis in Matlab to be conducted in the PC or Alpha environment.

The new release should give IMD more flexibility in data analysis and should facilitate the eventual migration from Alpha to PC environment. The new PC Matlab also has better performance than the corresponding Alpha version based on benchmark tests.

2.0 Issues Addressed In This Release

Version 3.0 is a service release which addresses the following main issues in Matlab script files (*.m) used in wake survey analysis. The GEDAP analysis software used in wake survey remains unchanged.

- Make all Matlab script files (*.m) run on both PC and Alpha environment.
- Upgraded all Matlab script files (*.m) to conform to tighter coding standards in Matlab Version 5.
- Replaced calls to menu() with calls to newmenu() to avoid excessive CPU hogging.
- Enabled hardcopies to be printed to any printer in the PC environment and Laser1c, Laser7 and Laser13 in the Alpha environment.
- Replaced == [] with isempty().
- Changed code around calls to input(), so proper checking of user input can be done.
- Use strcmp() for string comparison.
- Fixed bugs with unassigned or unreferenced variables.
- Changed colour of display to conform to Matlab Version 5 default where the background colour is white.
- Changed colour of plots and points so they can be seen in white background.
- Fixed a bug in Matlab Version 5.0 print.m in Alpha. (A custom IMD version replaces the version that comes with Matlab Version 5.0)
- Changed the device printopt.m from -dwin to -dwinc, so colour plots can be printed.

3.0 Wake Survey Analysis

Analysis of wake survey test data involves both GEDAP and Matlab. GEDAP is used for front end processing and Matlab is used for fairing and 3D plotting. The front end processing using GEDAP is conducted on ALPHA's. Fairing and 3D plotting using Matlab are conducted either on ALPHA's or on PC.

Matlab uses only the *RAD*.DAT files output from GEDAP as input. If it is desired that fairing and 3D plotting be done on Matlab PC version, these ASCII *RAD*.DAT files can be copied or FTP to a PC.

The following sections describe the setup required to run GEDAP and Matlab wave survey analysis software.

3.1 Setup Procedures and Notes for Wake Survey Analysis using GEDAP on Alpha's (duplicated here from Release Notes for Version 2.1 Wake Survey Analysis Software by L.M. Mak (NRC) February 14, 1997)

- WAKE_SET.COM is no longer copied into user's analysis directory. The new procedures to activate the wake survey analysis software is as follows:
  - Step 1: Go to analysis directory (e.g. DISK$PROJECT2:[PJ96814.WAKE])
  - Step 2: $ Copy WKS:[TEST]WAKE_CONSTANTS.COM [] /LOG
  - Step 3: Edit WAKE_CONSTANTS.COM to modify symbols according to test configuration. In WAKE_CONSTANTS.COM, note that (a) ACQUISITION_SYSTEM symbols has been added to distinguish different data acquisition systems; (b) the meaning of certain symbols definitions are changed when using GDAP data acquisitions (These notes are designated with string "GEDAP Notes:"); and (c) DAC* symbols has been added to identify physical channel assignments for GEDAP acquisition systems.
  - Step 3: $ @WKS:[TEST]WAKE_START
  - Step 4: From now on, everything should appear the same as the original software to users. For example, you can run
    - $ WAKE_CONVERT 1 12
    - $ WAKE_RUN 1 12
    - $ WAKE_SORT_RADII
    - etc.

- Naming Convention of DAC or PDF Files - If you are using GEDAP data acquisition system (GDAC or DAS_CMD), make sure you name the DAC / PDF files in the following format: 'WAKE_TEST_NAME'_00n.DAC or 'WAKE_TEST_NAME'_00n.PDF for each set of tests with n runs. This is because GEDAP does not use TEST_NAME and RUN_NAME as DAS data acquisition system. To maintain backward compatibility with the original design philosophy, the software simulate GEDAP data in DAS format by using generic names. The TEST_NAME is assigned in symbol WAKE_TEST_NAME. The RUN_NAME is composed using the test sequence number, the integer number before the .DAC / PDF file extension. For example, if the WAKE_STORAGE_DIRECTORY is DISK$PROJECT2:[PJ96814.WAKE], the WAKE_TEST_NAME is WAKP and the GDAC filename is WAKP_001.DAC, the output files are stored in

DISK$PROJECT2:[PJ96814.WAKE.WAKP_R1] as WAKP_R1.001 to WAKP_R1.00*. Here, the software implicitly assume the TEST_NAME is WAKP and the RUN_NAME is 1.

- Do not use long names for DAC, PDF and DAS files and WAKE_STORAGE_DIRECTORY. With the way the original wake software is designed, files are stored in and access from directories which is composed based on the WAKE_STORAGE_DIRECTORY, WAKE_TEST_NAME and run name symbols (e.g. DISK$PROJECT2:[PJ96814.WAKE.WAKP_R1]). If the directory name plus the GEDAP file name exceeds 64 characters, GEDAP programs will crash. Note also that when segments are selected, _S* extension is added to each GEDAP file to identify the segment number. This makes your original file names even longer. So, use short names for DAC / PDF files and directories.

- Wake survey analysis software requires 11 channels as input - carriage speed, 5 pressures, x position, y position, pitch angle, heave angle and roll angle. Make sure all channels are acquired.

- After radii have been sorted with WAKE_SORT_RADII and after it is confirmed that everything is fine up to this point, user can optionally use WAKE_DELETE_RUNS to delete GEDAP files in storage directories. For fairing purposes with Matlab, only *RAD*.DAT files are required as input. WAKE_DELETE_RUNS will delete only GEDAP files and remove storage directories. It does not delete *RAD*.DAT, *_DEPTH_CHECKS.DAT, *_LOG.DAT and *_STATS.DAT. However, once GEDAP files are deleted, user will have to re-select segments interactively if they subsequently found any error in processing

3.2 Fairing and 3D Plotting using Matlab on Alpha's

On Alpha's, the new software (Version 3.0) is installed on DISK$AP:[AP.WKS.FAIR3] directory. Bruce has configured all Alpha node to run Matlab Version 5.0, except Pluto which will still run Matlab Version 4.2c until the license expires.

If you log onto Alpha nodes running Matlab Version 5.0, you should configure your Matlab to use Wake Survey Analysis Software Version 3.0 in DISK$AP:[AP.WKS.FAIR3] directory.

If you log onto Pluto running Matlab Version 4.2c, you should configure your Matlab to use Wake Survey Analysis Software Version 2.1 in DISK$AP:[AP.WKS.FAIR2] directory.

Refer to the following instructions to configure your Matlab appropriately.

3.2.1 Setup Procedures for Fairing and 3D Plotting using Matlab On Alpha Nodes Running Matlab 5.0.0.4064

Step 1
If you do not have a Matlab directory in your login directory (i.e. disk$user1:[xxx.matlab] or disk$user:[xxx.matlab] where xxx is your username), then you need to create one using the following commands. If you have this directory, skip to step 2.

```
$ cd sys$login
$ create/dir [.matlab]
```

Step 2
If you do not have a startup.m file in the Matlab directory in step 1, you need to copy one in the directory using the following commands. Startup.m is a user customization file.

```
$ cd sys$login
$ cd [.matlab]
$ copy g2com$:matlab_mickey_startup.m     []startup.m
```

If you already have a startup.m file in the Matlab directory, you have to make sure the following lines are added to it.

The startup.m file should contain the following lines:

```
path(path, 'wks:[fair3]')
path(path, 'wks:[fair3.wfair]')
```

Step 3
Logoff and login again.

Step 4
Start Matlab using the following commands

```
$ matlab
```

At the Matlab prompt, execute the path command as follows:

```
>> path
```

You should see the Matlab directory mentioned in step 1 added to the top of the search path, and wks:[fair3] and wks:[fair3.wfair] added to the bottom of the search path. If not, you have encountered some problems. Contact the Software Engineering Group.

3.2.2 Setup Procedures for Fairing and 3D Plotting using Matlab On Alpha Node Pluto Running Matlab 4.2c

Step 1

If you do not have a Matlab directory in your login directory (i.e. disk$user1:[xxx.matlab] or disk$user:[xxx.matlab] where xxx is your username), then you need to create one using the following commands. If you have this directory, skip to step 2.

```
$ cd sys$login
$ create/dir [.matlab]
```

Step 2

If you do not have a startup.m file in the Matlab directory in step 1, you need to copy one in the directory using the following commands. Startup.m is a user customization file.

```
$ cd sys$login
$ cd [.matlab]
$ copy g2com$:matlab_pluto_startup.m     []startup.m
```

If you already have a startup.m file in the Matlab directory, you have to make sure the following lines are added to it.

The startup.m file should contain the following lines:

```
path(path, 'wks:[fair2]')
path(path, 'wks:[fair2.wfair]')
```

Step 3

Logoff and login again.

Step 4

Start Matlab using the following commands

```
$ matlab
```

At the Matlab prompt, execute the path command as follows:

```
>> path
```

You should see the Matlab directory mentioned in step 1 added to the top of the search path, and wks:[fair2] and wks:[fair2.wfair] added to the bottom of the search path. If not, you have encountered some problems. Contact the Software Engineering Group.

3.3 Fairing and 3D Plotting using Matlab on PC's

On PC's, the new software (Version 3.0) is installed in n:\Software\IMDapps\wks\fair3 directory. Refer to the following instructions to configure your PC Matlab appropriately.

3.3.1 Setup Procedures for Fairing and 3D Plotting using Matlab On PC's Running Matlab 5.1.0.421

Step 1

If you do not have a Matlab directory in your C drive (i.e. c:\matlab), then you need to create one using the following commands. If you have this directory, skip to step 2.

```
c:\> c:
c:\> cd\
c:\> md matlab
c:\> cd matlab
```

Step 2

If you do not have a startup.m file in the Matlab directory in step 1, you need to create one in the directory using your text editor. Make sure you enter the following lines. If you already have a startup.m file in the Matlab directory, you have to make sure the following lines are added to it. Startup.m is a user customization file.

The startup.m file should contain the following lines:

```
path(path, 'n:\Software\IMDapps\wks\fair3')
path(path, 'n:\Software\IMDapps\wks\fair3\wfair')
addpath c:\matlab  -begin
```

Step 3

Right mouse click on the Matlab icon, select "Properties" on the pop-up menu. In the "Matlab Properties" dialog, select "Shortcut" tab. In the "Start in:" edit box, enter c:\matlab. When finish, click OK button to close the dialog.

Step 4

Start Matlab by clicking the icon.

At the Matlab prompt, execute the path command as follows:

```
>> path
```

You should see the Matlab directory mentioned in step 1 added to the top of the search path, and n:\Software\IMDapps\wks\fair3 and n:\Software\IMDapps\wks\fair3\wfair added to the bottom of the search path. If not, you have encountered some problems. Contact the Software Engineering Group.

3.4 Executing Fairing and 3D Plotting Routines in Matlab

The procedure is the same on Alpha's and PC's. Start Matlab, type wks_fair and press the enter key.

3.5 Transferring *RAD*.DAT files From Alpha's to PC for Fairing and 3D Plotting

User will only require to follow this if they prefer to conduct fairing and 3D plotting using the PC version of Matlab. For fairing and 3D plotting with Matlab, only *RAD*.DAT files output from GEDAP are required. Therefore, only these ASCII files are needed to be transferred to a PC. The following procedure shows a typical way to tranfer these files from an ALPHA node to a PC.

Step 1:
On the PC, start a MS-DOS window.

Step 2:
Create a desired analysis directory on the PC and go to that directory.

Step 3:
At the PC prompt, type

```
ftp  minnie.imd.nrc.ca
```

and login.

Step 4:
Go to your analysis directory on the ALPHA using the cd command. For example, if your analysis directory is DISK$PROJECT2:[PJ96814.WAKE], type

```
cd DISK$PROJECT2:[PJ96814.WAKE]
```

Step 5:
Specify the transfer mode to be ASCII by typing

```
ascii
```

You should see "Type A ok"

Step 6:
To turn off interactive prompting on multiple commands, type

```
prompt
```

You should see "Interactive mode Off"

Step 7:
Transfer the *RAD*.DAT files by typing

```
mget *RAD*.DAT
```

Step 8:
To disconnect from ALPHA, type

```
quit
```

3.6 Re-assigning logical WKS on Alpha's

WKS is a system logical on Alpha nodes to point to the wake survey analysis software directory. Users normally do not have to re-assign this logical but developers will. To override it, use the following command:

$ define/tran=conceal wks directory_name

For example,

$ define/tran=conceal wks ghost$dka300:[lm.wks.]

3.7 Known Problems with Matlab Version 5.0 on Alpha's

Script file print.m has a bug which does not allow users to specify printers. To overcome this problem, we have replaced it with a IMD customized version. Bruce has installed this version. If you encounter any problem, notify Software Engineering Group.

If you place a menu over a figure, the button may attach to the figure and the figure will not update properly. To overcome this problem, do not place any menu over figures.

Release Notes for Version 2.1 Wake Survey Analysis Software by L.M. Mak (NRC) February 14, 1997

A new version of wake survey analysis software (version 2.1) has been installed on the AP tree in directory DISK$AP:[AP.WKS]. The previous version has been moved to DISK$AP:[AP.WKS_OLD].

Version 2.1 is a maintenance upgrade which addresses the following main issues which Dean brought up to Software Engineering Group and others:

- Enabled analysis of data acquired through DAS, GEDAP DAS_CMD and GEDAP GDAC, with the ability to demultiplex / split multiple runs at a time, like the original software.
- Fixed bug in wks_plta.m which caused Matlab to crash when "plot 1-Vx/Vs" menu option is selected.
- Fixed bug in wks_fair.m, cancel button in load file dialog, which caused Matlab to crash.
- Fixed all GPC files to solve a display problem where Vt/Vs and Vr/Vs are over-writing one another.
- Cleanup all *.com command procedures.
- Added proper indentation and used explicitly calls to GEDAP programs and other command procedures in all *.com command procedures for clarity.
- Upgraded command procedures to use INTERACTIVE_SEGMENT_SELECT4.COM.
- Separated user configurable codes from executable codes in WAKE_SET.COM to prevent users from accidentally modifying executable codes.
- Disabled user access to command procedures which should be called only by command procedures that interface with users. This is to prevent users from accidentally calling command procedures that they shouldn't call directly and cause unexpected problems.

The following changes / cautions should be noted by operators / users with the new version of wake survey analysis software:

- Naming Convention of DAC or PDF Files - If you are using GEDAP data acquisition system (GDAC or DAS_CMD), make sure you name the DAC / PDF files in the following format: 'WAKE_TEST_NAME'_00n.DAC or 'WAKE_TEST_NAME'_00n.PDF for each set of tests with n runs. This is because GEDAP does not use TEST_NAME and RUN_NAME as DAS data acquisition system. To maintain backward compatibility with the original design philosophy, the software simulate GEDAP data in DAS format by using generic names. The TEST_NAME is assigned in symbol WAKE_TEST_NAME. The RUN_NAME is composed using the test sequence number, the integer number before the .DAC / PDF file extension. For example, if the WAKE_STORAGE_DIRECTORY is DISK$PROJECT2:[PJ96814.WAKE], the WAKE_TEST_NAME is WAKP and the GDAC filename is WAKP_001.DAC, the output files are stored in DISK$PROJECT2:[PJ96814.WAKE.WAKP_R1] as WAKP_R1.001 to WAKP_R1.00*. Here, the software implicitly assume the TEST_NAME is WAKP and the RUN_NAME is 1.

- Do not use long names for DAC, PDF and DAS files and WAKE_STORAGE_DIRECTORY. With the way the original wake software is designed, files are stored in and access from directories which is composed based on the WAKE_STORAGE_DIRECTORY, WAKE_TEST_NAME and run name symbols (e.g. DISK$PROJECT2:[PJ96814.WAKE.WAKP_R1]). If the directory name plus the GEDAP file name exceeds 64 characters, GEDAP programs will crash. Note also that when segments are selected, _S* extension is added to each GEDAP file to identify the segment number. This makes your original file names even longer. So, use short names for DAC / PDF files and directories.
- WAKE_SET.COM is no longer copied into user's analysis directory. The new procedures to activate the wake survey analysis software is as follows:
  - Step 1: Go to analysis directory (e.g. DISK$PROJECT2:[PJ96814.WAKE])
  - Step 2: $ Copy WKS:[TEST]WAKE_CONSTANTS.COM [] /LOG
  - Step 3: Edit WAKE_CONSTANTS.COM to modify symbols according to test configuration. In WAKE_CONSTANTS.COM, note that (a) ACQUISITION_SYSTEM symbols has been added to distinguish different data acquisition systems; (b) the meaning of certain symbols definitions are changed when using GDAP data acquisitions (These notes are designated with string "GEDAP Notes:"); and (c) DAC* symbols has been added to identify physical channel assignments for GEDAP acquisition systems.
  - Step 3: $ @WKS:[TEST]WAKE_START
  - Step 4: From now on, everything should appear the same as the original software to users. For example, you can run
    - $ WAKE_CONVERT 1 12
    - $ WAKE_RUN 1 12
    - $ WAKE_SORT_RADII
    - etc.
- Wake survey analysis software requires 11 channels as input - carriage speed, 5 pressures, x position, y position, pitch angle, heave angle and roll angle. Make sure all channels are acquired.
- After radii have been sorted with WAKE_SORT_RADII and after it is confirmed that everything is fine up to this point, user can optionally use WAKE_DELETE_RUNS to delete GEDAP files in storage directories. For fairing purposes with Matlab, only *RAD*.DAT files are required as input. WAKE_DELETE_RUNS will delete only GEDAP files and remove storage directories. It does not delete *RAD*.DAT, *_DEPTH_CHECKS.DAT, *_LOG.DAT and *_STATS.DAT. However, once GEDAP files are deleted,

user will have to re-select segments interactively if they subsequently found any error in processing.

I will ask Dean to confirm for me that all the changes work properly by re-analyzing the wake survey runs in project 96814 - Marineering Type 1000 Buoy Tender. Since the roll channel is missing from that project, the software has to generate an artificial roll channel in WAKE_GDAC_GEDAP.COM. Once the new software is checked out by Dean, I will replace WAKE_GDAC_GEDAP.COM with WAKE_GDAC_GEDAP_REAL.COM which expects roll channel to be present.

Please let me know if you have problem understanding any points in this release notes.