NASA Contractor Report 165698

AN OPERATIONS MANUAL FOR THE SPINNING MODE SYNTHESIZER
IN THE LANGLEY AIRCRAFT NOISE REDUCTION LABORATORY

Daniel L. Palumbo

KENTRON INTERNATIONAL, INC.
Hampton Technical Center
   an LTV company
Hampton, Virginia 23666

NASA

National Aeronautics and
Space Administration

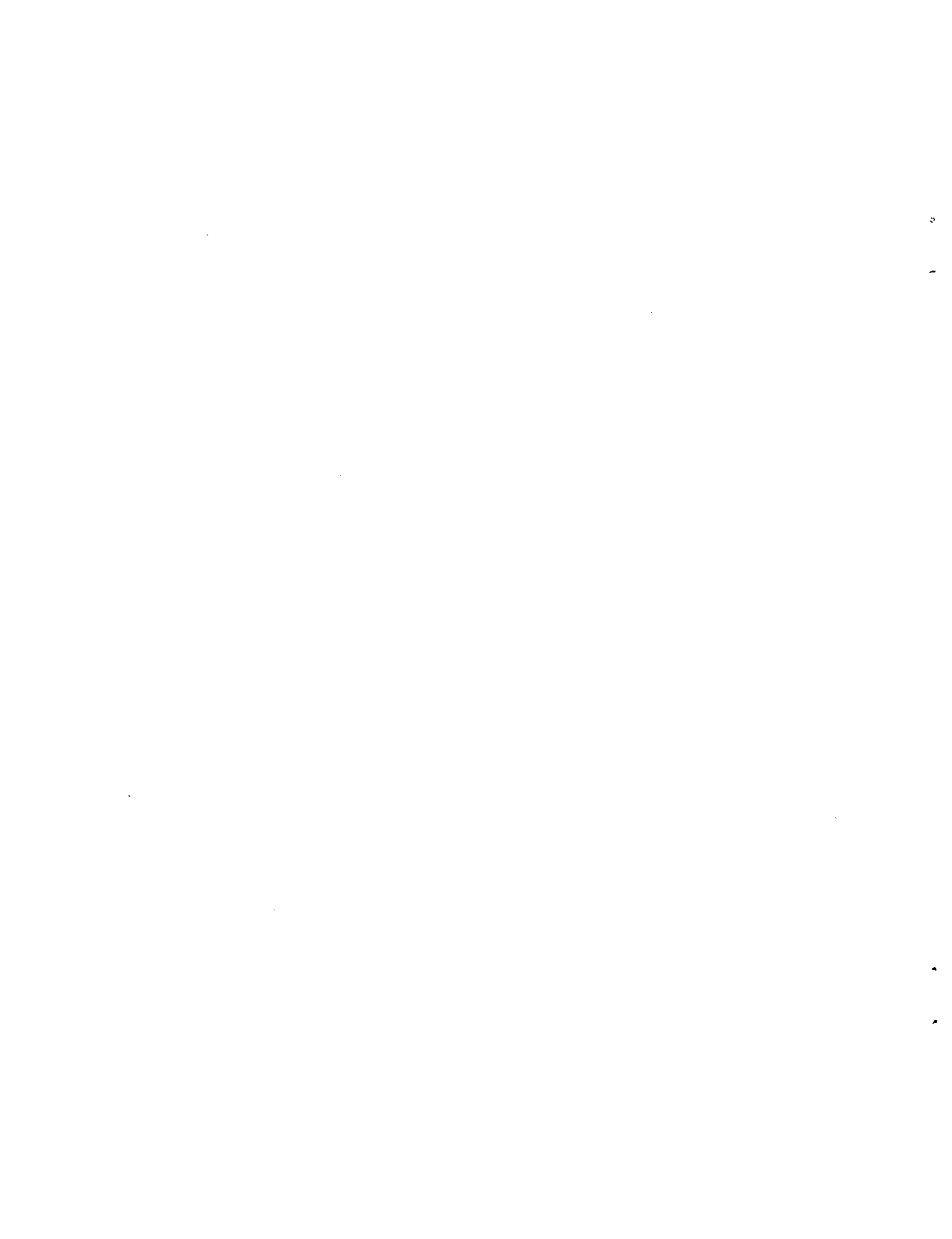**Langley Research Center**
Hampton, Virginia 23665

TABLE OF CONTENTS

TABLE OF CONTENTS
(continued)

1.0 INTRODUCTION TO THE SPINNING MODE SYNTHESIZER

Fan rotors mounted in static test rigs as an acoustic source suffer from many problems. These include variations of the source acoustic field with inflow distortions, an extremely complicated source structure, and limited control over the relative amplitudes and phases of individual modes or their frequencies. The use of fan rotors as a noise source is, therefore, undesirable in studies of theoretical propagation models or in tests of acoustic liner properties, since the structure of the source must be accurately known to insure consistent results. This need for a dependable and controllable noise source led the Acoustic and Noise Reduction Division (ANRD) to contract with the Lockeed Georgia Company to develop the Spinning Mode Synthesizer (SMS). After the installation of SMS in January 1977, Kentron International, Inc., Hampton, Virginia, was contracted to provide support in refining the system. It is this level of revision, as of June 1980, that is the basis of this report.

Turbofan noise is composed of a series of fundamental acoustical modes. SMS is capable of creating and controlling 13 azimuthal spinning modes (mode 0 up to mode 12) independently or concurrently, spinning positively or negatively, at any amplitude or phase angle. The modes are produced by 24 acoustic drivers equispaced circumferentially around the 0.3 meter diameter flow duct. The amplitudes of the real and imaginary parts of each driver signal are controlled by a 16 bit computer. The acoustic field is monitored by 48 microphones mounted flush in the duct wall at an axial position 0.2 meters from the drivers. The amplitude and phase of the signal at each microphone are tabulated in an array to define the produced modal content. This pressure field is compared to an ideal result in an optimization algorithm that adjusts the driver settings until the system error is minimized. At this point SMS suspends and data may be collected.

2.0 OPERATING INSTRUCTIONS

The software for SMS operates in a 16 bit computer which supports multi-users and multi-programming. Each user has a log-on name and an associated User Identification Code (UIC). Log-on name SMS will attach the user to the SMS files on UIC [10,12]. SMS requires a 132 character/line terminal and is usually run from terminal 'TT2' in the control room, Room 227, of Building 1208, the Langley Aircraft Noise Reduction Laboratory (ANRL). Mode shape parameters and

optimization variables are input through the terminal. Eight control switches provide additional program control, (Table I). Most switches apply to specific program functions and will be described later. The first seven switches are asserted when raised. The eighth or STOP switch is a pushbutton, and its state is generally unknown until the program is run. The STOP switch will turn off the drivers and halt the program. SMS is usually run to either create, optimize and store a mode shape or to recreate or recall a mode shape so other analysis programs can be run. In the following sections describing operator input on TT2's keyboard CR refers to the RETURN or ENTER key on the terminal. Input to the program is distinguished from data and prompting output by double quotes.

2.1 CREATING A MODE SHAPE

When creating a mode shape it has been common practice to insert an anechoic terminator into the duct. This eliminates large reflectons from the duct termination, resulting in faster convergence of the acoustic field.

2.1.1 Starting the SMS Program

Log-on to the computer under SMS. To start the program, enter

"RUN SMS   CR "

SMS will be loaded into the computer and start execution by requesting the user's name (up to 48 characters) and asking if he wishes to recall a mode shape.

NAME:
"RJS   CR "
DO YOU WISH TO RECALL A MODE SHAPE? " CR "

If the response is not "YES", a "NO" is assumed.

2.1.2 Defining the Frequency

SMS can set either the frequency or the frequency parameter. The frequency parameter, KA, is defined as

$$KA = 2\pi FA/C$$

where:

π = 3.14159

F = frequency in hertz

$$C = \text{speed of sound}$$
$$A = \text{radius of the duct}$$

and is therefore a dimensionless quantity. The program is capable of stepping the frequency parameter or frequency through a number of increments for any mode shape. This option is controlled by switch 3 and is rarely used due to the length of time involved in optimizing any one shape. In the following example the frequency parameter is set to 4.0 with no increments or steps.

```
MODIFY FREQUENCY SEQUENCE? "Y CR"
SET FREQUENCY PARAMETER? "Y CR"
INITIAL WAVE #/FREQ: 0.0 INCREMENT: 0.0 NO. OF STEPS: 0
NEW? "4.0 CR"
```

2.1.3 Selecting the Mode Shape

Up to 13 modes may be set individually or concurrently. Each mode is defined by its spinning mode order, the direction it is spinning and its amplitude and phase. The following example will request a mode 1 spinning positive at 120.0 dB with no phase shift.

```
MODIFY MODE SHAPE? "Y CR "
0 MODES; CHANGE? "Y CR "
TO: "2 CR "
```

	SPINNING +		SPINNING −		
MODE	AMP(dB)	PHASE	AMP(dB)	PHASE	YES (NEW VECTOR)
0	0.0	0.0	0.0	0.0	" CR "
1	0.0	0.0	0.0	0.0	"Y 120.0 CR "

To set a mode of order n, n+1 modes must be requested. SMS will calculate and display on the terminal the ideal amplitudes and phases at each microphone for the mode shape entered.

Consider, for example, modes with combinations of ± spinning mode components. For a positive spinning mode order 2 with 120 dB level at 45 degrees phase combined with a positive spinning mode order 3 with 125 dB level at 0 degrees phase and a negative spinning mode order 3 with the same level but at 90 degrees phase, one would input the following:

3

```
MODIFY MODE SHAPE?   "Y CR"
O MODES:  CHANGE?   "Y CR"
TO:  "4 CR"
```

MODE	AMP (dB)	PHASE	AMP (dB)	PHASE	YES (NEW VECTOR)
0	0.0	0.0	0.0	0.0	"CR"
1	0.0	0.0	0.0	0.0	"CR"
2	0.0	0.0	0.0	0.0	"Y 120., 45. CR"
3	0.0	0.0	0.0	0.0	"Y 125., 0., 125., 90. CR"

2.1.4 The Target Error Function

The target error function, TEF, is calculated as

$$TEF = 0.0096 \; AMAX^2 * XN^2$$

where:

XN = the percent error

$AMAX$ = the maximum amplitude

SMS then prompts the user to establish XN, the percent of the maximum amplitude
to be used to calculate TEF. 5.0 Percent is Default.

```
5.0 PERCENT ERROR - NEW VALUE? " CR "
TARGET ERROR = 2330.3
```

The system error (PCOST) is calculated as the sum of the square of the difference
between the real and imaginery parts of the acquired and ideal mode shape arrays.
When PCOST drops below TEF, the target has been met.

2.1.5 The Signal Enhancement Parameters

Data taken from 48 source microphones are used to determine the modal
content of the acoustic field. To increase the accuracy of the data, it is
averaged before it is analyzed. When flow is present in the duct, the acoustic
field may fluctuate with time causing the entire mode structure to change as the
48 microphones are being sampled. To eliminate this problem, two parameters
are defined. One, the source microphone sweep parameter, MAVS, is the number of
passes to make through all 48 microphones. The other is the number of averages
to perform at each microphone for each pass, and is defined as the averages per

4

sweep parameter MAVT. Performing the averages as multiple sweeps spreads each microphone's acquisition time out over the total time to do all the averaging and therefore reduces the contribution of flow dependent fluctuations and yields a more consistent result. The value of these parameters determines how long an optimization will take and choosing the least number of averages to yield accurate results for any given mode shape/flow rate condition is an intuitive matter. Switch 5 will cause the acquisition routine to pause to allow the user to enter these two values before each pass. During the optimization routine when repeated acquisitions are made, the user could define the driver increment, INCDR, to be zero and by raising switch 5, adjust the sweep and avg/sweep parameters, MAVS and MAVT, respectively, until the PCOST is consistent, indicating a stable result (see section 2.1.8 Optimization Algorithm).

SOURCE MIC SWEEPS: 3 AVG/SWEEP: 1 NEW: " CR "

The default is 3 sweeps with one average at each source microphone per sweep. This is adequate for no flow work. If flow is present, a response of 5,5 will give good results. Both numbers must be input if either is changed.

2.1.6 The Reference Temperature

If SMS is setting the frequency parameter, it will need the ambient temperature to calculate the setup frequency. The temperature is input in degrees Celsius.

ENTER REF TEMP(C)

"23.0 CR "

WAVE # 4.0 CHANGED TO 1480.0 HZ AT ROOM TEMP OF 23 "C

At this point all the input parameters have been defined. A summary of these parameters is printed before optimization begins.

2.1.7 Initial Driver Settings

The Initial Driver Setting routine (INID) is used to get a good mode setup before optimization begins. INID will first determine the setting for each driver to obtain a value within +/- .5 dB of the calculated ideal amplitude of the corresponding microphone (taken to be the microphone number double that of the driver number). If the target amplitude is too high to be set, a warning

5

message is printed. This error is usually irrecoverable in INID. To halt the process use the stop switch. If the condition persists, more information may be obtained by raising switch 2. The overall gain and real and imaginery parts of the individual gains will be printed each time a driver setting is changed.

After each driver setting has been determined, all drivers are turned on. The resultant mode shape is compared to the ideal and adjusted until the average amplitude difference at each microphone is less than 1 dB and the average phase difference is less than 3 degrees. This process may not converge and to continue to the next phase of optimization raise switch 1. To stop the program, press the stop switch and raise switch 1. No matter how INID is exited, the driver settings that minimized the system error are used as the basis of the optimization.

2.1.8 The Optimization Algorithm

The optimization algorithm, CKMODE, attempts to bring the system error PCOST below the target error function TEF. CKMODE begins by listing on the terminal a table of amplitudes and phases at each microphone and an expansion of this circumferential pressure distribution to +/- 10 spinning modes. The driver inputs may be uniformly phase shifted at this point to achieve the desired phase angle (usually zero) of a dominant mode.

ENTER PHASE SHIFT+/- DEGREES " CR "

A response of zero or CR will not alter any values and the program will proceed. A nonzero response will phase shift the driver setting array and repeat the section.

The first step in the optimization process is to tabulate the effect on PCOST of incrementing each driver by a constant factor, INCDR. As each component of each driver is incremented, the difference between the original PCOST and the resultant PCOST is stored. After all the drivers have been incremented and the difference array constructed, the array is multiplied by a variable, EPS, and reapplied to all the driver settings simultaneously. By changing EPS, CKMODE then adjusts the amount of correction added to the drivers until a minimum PCOST is found. The process is repeated until TEF is reached or switch 6 is raised.

6

The amount of driver increment, INCDR, and the initial value of the multiplication variable, EPS, can be altered to suit the requirements of a particular mode setup.

<div align="center">INCDR= 20   INIT.   EPS = .100E 1 NEW?   " CR "</div>

The default values of 20,1 work well in most circumstances. Note that INDCR must be entered as an integer and EPS as a real number.

As CKMODE is varying EPS it is trying to fit the resultant PCOSTS to a parabola. If a good fit is not found, the message

<div align="center">FAILED TO CONVERGE</div>

is displayed. Switch 6 can then be raised to exit CKMODE or lowered to try the optimization with a new set of parameters. Switch 6 terminates the optimization at any point in CKMODE.

2.1.9 Storing a Mode Shape

After the mode setup has been optimized, it can be stored for recall later.

DO YOU WISH TO STORE MODE SHAPE   Y/ CR   ? "Y CR "

ENTER 8 CHARACTER FILE NAME
"KA400M1P CR "

ENTER 110 CHARCTER COMMENTS
"FREQUENCY PARAMETER 4.0   MODE 1 POSITIVE CR "

SMS keeps a record of all file names in use. If a duplicate name is entered, a message is printed and a new name is requested.

SMS will finally suspend itself to allow the user to run his analysis routines. Typing "UNS CR " will resume SMS. If the stop button has been pushed, SMS will shut down; otherwise a modal expansion of the present state of the system is printed and SMS suspends.

2.2 RECALLING A MODE SHAPE

A typical test requiring SMS involves setting up a particular mode and flow conditions, and then running one or more data acquisition programs. The data from these programs should be stored with the user's UIC. To start SMS from the user's UIC enter

"RUN [10,12] SMS CR "

SMS will be loaded into the mini-computer and begin execution by requesting the user's name (up to 48 characters) and asking if he wishes to recall a mode shape.

NAME:

"RJS CR"

DO YOU WISH TO RECALL A MODE SHAPE? "Y CR "

ENTER 8 CHARACTER FILE NAME "KA400M1P CR "

If the name entered does not exist, an error message is printed and the question is repeated. If changes must be made to the recalled mode shape, the user can modify all input parameters.

MODIFY? "Y CR "

A positive response continues with section 2.1.2, Defining the Frequency. A negative response continues with section 2.1.4, The Target Error Function.

3.0 SYSTEM HARDWARE AND HARDWARE DRIVERS

Before attempting a detailed analysis of the Spinning Mode Synthesizer's main software modules, a description of the system's harware devices and their software interfaces will be presented.

The overall system layout is illustrated in Figure 1. The hardware is interfaced to the mini-computer's data bus through three 16 bit input/output cards (DR11A, DR11B and DR11C shown on Figure 2), a 12 bit analog-to-digital converter (ATD), a general purpose instrumentation bus (GPIB), and an asynchronous current loop terminal interface (DL11). Except for the terminal, which interfaces through the DL11, and the frequency synthesizer, which is on the instrumentation bus, all devices are attached through the Computer Interface Console (CIC). The CIC connects to the acoustic drivers' power amplifiers and the Test Facility Console (TFC) in the ANRL Applications Area, Room 299, and the switch panel in the Control Room, Room 227.

3.1 THE COMPUTER INTERFACE CONSOLE

Figure 3 shows the CIC and the devices attached to it in Room 300. The output of the frequency synthesizer is divided into four signals by a phase

shifter. These signals have a phase angle relative to each other of 0, 90, 180, and 270 degrees. The four signals are fed into the phase attenuator in rack 2 of the CIC through connector P4 where their amplitudes are adjusted by a master gain setting. The four signals, along with digital control lines, are bussed to the 24 gain/phase boards where they are added to produce the desired amplitudes and phases.

The digital control lines for the gain/phase logic come from DR11A, through connector P1 of rack 2, and are buffered by bus drivers before exiting through connector P4 to rack 1 through P2. Control switch panel settings come through connector P3 to current sensors. The remaining digital lines are driven to current sensors in the TFC.

The analog signal selected by the multiplexer in the TFC comes through P2 of rack 2 where it is interfaced by a differential receiver. The buffered signal passes through P1 to a tracking filter and then to the analog to digital converter. The tracking filter's center frequency is controlled by phase 0 and has a selectable bandwidth with 200 Hz used in SMS. To detect clipping of the signal before the tracking filter, an overload indicator monitors the signal.

3.1.1 Setting the Frequency (CKFRQ and GFRQ)

The frequency synthesizer is interfaced to the mini-computer over the GPIB. SMS uses subroutine CKFRQ to set the source signal frequency and calculate the acquisition rate necessary to acquire the specified number of samples per cycle.

          CALL CKFRQ(F,USEC,XNS,IT,NSPC)

where   F   = frequency in Hertz (input)
               USEC = signal period in microseconds (output)
               XNS = scaled angle increment per sample (output)
               IT   = sample interval in microseconds (output)
               NSPC = number of samples per cycle (input)

The frequency is encoded into the following ASCII command string which is sent to the synthesizer by subroutine TALK.

          FU1   - function 1, sinewave
          AM1.VR   - amplitude 1.0 VRMS
          PH0.DE   - phase 0.0 degrees

```
RF2           - rear panel output enabled
OFO.VO        - offset voltage 0.0 volts
FRnnnnn.OHZ   - frequency in Hertz
AC            - do auto calibrate
*             - termiantor
```

After a ten-second wait for the synthesizer to set up and calibrate, sub-routine GFRQ is called to check the frequency

CALL GFRQ(TK2)

where TK2 is the period of the signal in microseconds. GFRQ uses the analog-to-digital converter's internal timing function to average the period of the trigger signal (phase 0) for 20 cycles. If the period is not accurate, an error message is printed and the command string is reissued.

The sample time, IT, is calculated from the ratio of the signal period to the number of samples per cycle (NSPC). The analog-to-digital converter is limited to a 40 kHz acquisition rate during 12 bit conversions, so IT is limited to 25 microseconds. At the default value for NSPC of 16, IT begins being limited at a source frequency of 2500 Hz. This of course lowers the NSPC and, therefore, the accuracy with which the amplitude and phase of the signal can be determined may suffer. The amplitude and phase routine, CMPAPH does a least squares fit to a sine wave and will work quite well if the input data are a clean wave. XNS is the angle increment scaled, so one cycle per sample would give the maximum unsigned 16 bit integer value of 65535.

3.1.2 Setting the Gains and Phases(SGNP)

Control of the amplitude and phase of the signal to each power amplifier is achieved by adding a pair of orthogonal signals (real and imaginary parts) derived from the phase shifter. By varying the amplitude of the real and imag-inary parts, any phase or amplitude may be set.

$$A\cos(\omega t + \phi) = A\cos(\phi)\cos(\omega t) - A\sin(\phi)\sin(\omega t)$$

$$= B\cos(\omega t) + C\sin(\omega t)$$

$B\cos(\omega t)$ represents the real part and uses the 0 degree signal for positive amplitudes and the 180 degree signal for negative amplitudes. $C\sin(\omega t)$ is the

10

imaginary part and uses the 90 degree signal for positive amplitudes and the 270 degree signal for negative amplitudes. The resultant signal level and phase are controlled by three gain settings; one setting for a master gain given to the four phase signals in the phase attenuator, and one each for the real and imaginary parts of the signal. The gain is set by an 8 bit digital-to-analog converter where the signal is input as the reference signal, and the 8 bit digital setting controls the output level.

An address decode card enables either the phase attenuater or one of the real or imaginary parts of the 24 gain/phase boards. Addresses 0-47 select one of the real or imaginary parts, and address 63 selects the master gain.

CALL SGNP(IADD,IGAIN)

where   IADD is address; 0-47 or 63
                 IGAIN is gain 0-255

SGNP uses the low byte of DR11A. Control status register (CSR) bit 0 is set to output the address and reset for gain. CSR bit 1 is set for negative gain and reset for positive.

3.1.3 Control Panel Interfaces(GPSW)

The 8 control panel switches are sensed through current sensors (card 5) attached to DR11A's low input byte. Associated with the switches are 8 indicator lights driven by the control panel interface, card 3. The latching circuitry for this card is enabled by address 62 of the gain/phase address decode logic.

CALL GPSW(ISW,ION)

where   ISW is switch number to sense
                 ION is 1 if down or 2 if up.

3.1.4 Current Drivers

Card 4 of rack 2 holds drivers which interface to current sensors in the TFC for the auto gain and multiplexer devices. Refer to the next section on the TFC for explanation of these modules.

3.1.5 Differential Receiver (ATOD)

A differential receiver is used to reduce noise impressed on the analog line from the TFC to the CIC. The signal then goes to a tracking filter. The tracking filter has a 200 Hz bandwidth with the center frequency set by the 0 degree signal from the phase shifter. The filter eliminates flow noise and thereby reduces averaging time in the computer. The actual input to the computer is done by direct memory access transfer through the analog to digital converter. The phase 0 signal is used to trigger the start of the conversion to provide ensemble averaging.

        CALL ATOD(IBUFF,ICHN,IT,NSMP,MODE)

where   IBUFF is data array

        ICHN is channel number to input

        IT   is acquisition sample time in microseconds

        NSMP is number of samples

        MODE is zero for triggered acquisition and 1 for non-triggered

3.1.6 Overload Monitor (LDOVR and OVRST)

The analog-to-digital converter is limited to a range of +/- 5.0 volts. If the signal is clipped at these levels, the software can adjust the auto gain unit until no clipping is present. However, with flow present, the signal after the tracking filter can be less than +/- 5.0 volts, while before the filter the signal is being clipped at the amplifiers in the auto gain unit. The auto gain unit will clip at +/- 13.0 volts. To detect this distortion, the overload indicator monitors the signal and sets bit 7 of DR11C if the level goes above +/- 12.0 volts.

        CALL OVRST
        CALL ATOD(IBUFF,ICHN,IT,NSMP,MODE)

        IF(LDOVR(0).EQ.1)GOTO4

Subroutine OVRST resets the indicator. If the signal transits the limits during acquisition, the indicator will be set. Integer function LDOVR will return a 1 if the indicator is set, or a 0 if not.

12

3.2 THE TEST FACILTY CONSOLE

Although SMS uses the Test Facility Console (TFC) primarily to bus microphone signals to the CIC, the TFC holds added capability originally used by SMS, and is now available in external routines (refer to Section 5). These devices include the auxiliary multiplexer, temperature and pressure transducer amplifiers, and the drum stepper motor drive as shown in Figure 4.

The 48 source microphones have built in FET transistors. The microphone power supply provides the bias voltage for the transistors in 4 quadrants of 12 microphones each. The microphone signals are connected to the first 48 channels of the 64 channel multiplexer. The amplitude of the selected channel is adjusted by the auto gain unit which provides from 12 dB attenuation to 78 dB gain.

3.2.1 The Mutilplexer (SMPX)

The TFC multiplexer has 64 channels. Table II lists the channel assignments. SUBROUTINE SMPX selects the multiplexer channel.

$$CALL\ SMPX(ICHAN)$$

Where ICHAN is channel number, 0-63. Channels 0-7 are set as channels 48-55. Channels 8-15 are set as 40-47 and so on until channels 56-63 which are actually set as channels 56-63. DR11A bits 8-13 hold the multiplexer select lines.

3.2.2 The Auto Gain Unit (SMKG)

The auto gain unit provides from 12 dB attenuation to 78 dB of gain. Subroutine SMKG selects the gain level,

$$CALL\ SMKG(IGN)$$

where IGN is an integer value of 0-15. The gain setting is calculated as

$$GAIN = -12 + (IGN*6)\ dB$$

IGN is set into bits 12-15 of DR11B. CSR bit 0 is used to strobe the setting into the auto gain unit.

3.3 DIAGNOSTICS (SMSDIAG)

SMSDIAG is a diagnostic program devloped to test the operation of SMS hardware. SMSDIAG runs independently of SMS software and is, therefore, useful

13

in determining if a failure in SMS is in the software or hardware. The program uses the mini-computer's front console switches for control input.

> "RUN SMSDIAG CR "
> ODT:TTO
> "G"
> SPINNING MODE SYNTHESIZER
> BASIC DIAGNOSTIC PACKAGE
> ENTER CODE CHARACTER:

Characters A-H call up the tests described below. Figure 5 gives the mini-computer console switch definition for each diagnostic. Entering an 'X' will exit the program. Once a test has begun switch 0 will terminate it.

3.3.1 A - Analog Outputs

The Laboratory Peripheral System, LPS, has 4 channels of digital to analog output in addition to the 8 channels of analog-to-digital input. Although the analog output is no longer used by SMS, the test remains as part of this program. Switches 13-15 select the output channel and switches 1-11 define the output level where 0 sets -5. volts and 4095 sets +5. volts.

3.3.2 B - Gain/Phase Board Logic

This test uses switches 9-15 to select the gain/phase address, switch 8 the sign and switches 1-7 the magnitude. The master gain, address 53, must be set before any other board will ouput a signal.

3.3.3 C - Mpx, Microphone Gains

This test sets the multiplexer and auto gain unit. Switches 8-13 select a multiplexer channel and switches 1-5 a microphone gain. Multiplexer channels 53 and 54 select test signals of a square wave and triangle wave, respectively.

3.3.4 D - Display Control Panel

This test will output on the LPS LED display the octal representation of the panel switch settings in the control room.

14

3.3.5 E - To Test Clock Sequence

This test simulates the timing function used in subroutine GFRQ to determine the oscillator frequency. The test averages 10 periods of the signal present at the LPS ST2 input. The result is displayed in octal on the LED display in 100 microsecond units. For example, a 1 kHz input will display an octal 12. The ST2 trigger level must be adjusted to trigger on the input signal.

3.3.6 F - A/D Transfer

This routine tests the basic analog to digital conversion function of the LPS. Switches 8-13 select the input channel on the LPS. The results are stored in octal on the LED display.

3.3.7 G - DMA Transfer

This test runs a Direct Memory Access acquisition identical to the one used by SMS. Switches 8-13 select the input channel. In addition to displaying the 128 acquired values on the LED display, a print out of information is listed on the terminal. The first value in the buffer is printed, followed by 32 pairs of octal numbers. The first number in the pair is the acquired value. The second number is the difference between the present value and the previous one. If the ST1 trigger function is working correctly, there should be little difference between the two acquisitions for a periodic input. Also listed are the frequency, real and imaginary parts, amplitude and phase, and mean value of the input signal.

3.3.8 H - Overload Indicator Test

This test sets the multiplexer to channel 53 for the square wave signal and steps the auto gain unit from 0 to 15 or until an overload occurs. If the test is unsuccessful, the message 'FAILED TO DETECT OVERLOAD' is printed on the terminal.

3.4 HARDWARE INITIALIZATION (INIT, IEINIT)

Subroutine INIT assigns logical unit numbers 1 through 4 to DR11A, DR11B, DR11C and the LPS. INIT uses a special function of the DR11 driver to set the SMS task privileged. Subroutine IEINIT assigns logical unit number 9 to the GPIB bus.

3.5 ERROR MESSAGE (DISERR)

Subroutine DISERR is found in module INIT and is called to indentify an I/O error after it has been detected.

DEVICE XXX FUNC IO.YYY STATUS CODE - NNN

XXX is a device acronym, YYY is the function and NNN is the error number which is printed on the terminal. Table III contains a device versus I/O function reference. The error number can be resolved by using HELP.

'HELP ERR IO -NNN CR '

For example

DEVICE LPS FUNC IO.CL3 STATUS CODE - 3

means that the LPS had a time out error during the IO.CL3 function used by subroutine GFRQ to determine the input frequency.

4.0 SYSTEM SOFTWARE

Program SMS uses 9 logical unit numbers (LUNs). Logical unit numbers 1 through 3 are assigned to the DR11 interfaces and LUN 4 to the LPS A/D converter in subroutine INIT. LUNs 5 and 6 retain their default assignments of the terminal and printer. LUN 7 is assigned to the microphone calibration file DRMCAL.DAT. LUN 8 is used to access mode shape storage files. LUN 9 is assigned to the GPIB bus in subroutine IEINIT. IEINIT also reserves the oscillator for use by specifying its address, an ASCII 1.

Subroutine STOPIT is called in SMS to sense the stop switch and terminate the program if set. STOPIT is called after each major segment of SMS.

Subroutine SETUP initializes the program's data base. Integer variable IFINID indicates the user has recalled a mode shape if equal to 1, and, therefore, does not need the driver initialization function of INID.

The frequency control loop in SMS is rarely used. Its purpose is to allow stepping the mode shape through a set of frequencies. Array CIPOS holds the initial frequency value and CDPOS is the increment amount. If the user chose to use frequency parameter instead of frequency, JCNTL(5) will be nonzero. If frequency parameter is sensed, subroutine TABS is called to convert it to frequency. Once

16

the frequency is set, subroutine SETCAL interpolates the microphone calibration
file and calculates the calibrated target mode shape array, CATMS.

TPRNT prints a summary of the test parameters prior to mode shape initial-
ization in INID. After INID, CKMODE applies the optimization algorithm.

Upon returning from CKMODE, the user has the option to store the mode
shape parameters for later use. File INDEX.SMS keeps a list of all past
file names and checks to be sure the new name is unique. All mode shape data
files end with the extension .SMS. File SMSDAT.DAT is written to pass the
acquisition parameters to analysis programs. SMS then stops. This function
is like a 'PAUSE,' but sets the program inactive to release its system resources.
The user may at this time run his analysis programs. When SMS becomes
'UNSTOPPED,' it either exits, if the stop switch is set, or recalculates and
displays the mode shape before stopping again.

4.1 INITIALIZING THE DATA BASE (SETUP)

SETUP begins by setting all drivers and the multiplexer channel to zero.
It then asks for the name of the user, after which there are three paths through
SETUP. If a mode shape is recalled, then it can be used as is or used with
modifications to any of the input parameters. If a mode shape is not recalled,
then it is necessary to input all the parameters required to run. If a mode
shape is recalled, integer variable IFINID is set to a one. This disables the
driver initialization section in INID. Variable III is used locally to convey
the same information and is described later.

The first value to be determined is frequency. This can be set as frequency
or frequency parameter. If frequency parameter is chosen, JCNTL(5) is set to a
one to indicate that the value needs to be converted to frequency before being
used (see section 4.2). The frequency (frequency parameter) is entered as a starting
value (CIPOS), an increment (CDPOS) and a number of steps (NCSTP). The length of
time necessary to optimize a mode definition usually limits any run to only one
frequency.

The next section of SETUP calculates the complex target mode shape array,
CTMS, according to user inputs AP,PHP,AN, and PHN, which are the amplitude and
phase of positive and negative spinning modes for each order mode.

17

$$CTMS(i) = VSTD \sum_{m=1}^{M} [10^{AP(m)/20} e^{ja} + 10^{AN(m)/20} e^{jb}]$$

where exponents a and b are defined by:

a   = (m - 1) PH(i) + PHP(m)

b   = -(m - 1) PH(i) + PHN(m)

and where:

VSTD   = the amplitude of a microphone signal at 0 dB SPL

m   = the mode order index

i   = the microphone index

j   = $\sqrt{-1}$

M   = the maximum spinning order considered

AP(m)   = the amplitude of the + spinning mode

AN(m)   = the amplitude of the - spinning mode

PHP(m) = the phase of the + spinning mode

PHN(m) = the phase of the - spinning mode

PH(i)   = the ith microphone's angular position

After calculation of CTMS, file DRMCAL is read. DRMCAL holds the calibrated value for VSTD,VODB, a list of bad microphones in array IBM and an array of calibration data CALM, for the source microphones. Real variable FACT is the ratio of VODB to VSTD and is used to adjust CTMS. If a mode shape has been recalled and is unmodified, this adjustment has already been applied. Variable III is zero if the mode recalled is unmodified and is otherwise a one.

$$FACT = FACT^{III}$$

This equation results in one if the mode shape was recalled unmodified or as the ratio of VODB/VSTD if not.

The maximum amplitude from CTMS is used to calculate the target error function, TEF.

$$TEF = 0.0096*AMAX^2*XN^2$$

where   XN is the percent error

AMAX is the maximum value found in CTMS

18

and .0096 accounts for the squared per cent and the sum of 96 real and imaginary parts of the 48 microphones.

Before exiting SETUP defines the optimization and enhancement parameters. INCDR is the driver increment used during optimization. IDLIM is the maximum allowed driver setting, leaving room for a positive increment, and EPSI is the step multiplier. MAVS is the number of passes to be made through 48 microphones and MAVT is the number of averages to be performed at each microphone per pass.

4.2 FREQUENCY PARAMETER TO FREQUENCY (TABS)

TABS is called with one argument, VO, the frequency parameter, which is returned as the corresponding frequency. The frequency parameter is the wave number non-dimensionalized by the duct radius. The subroutine asks for a reference temperature in degrees Celsius which is used to determine the speed of sound in feet per second. The frequency is then calculated as

$$VO = 12.*V*C/(2\pi*5.895)$$

where   12 converts the radius from inches to feet

               V is the frequency parameter

               C is the speed of sound

and   5.895 is the duct radius in inches

4.3 CALIBRATING THE MODE SHAPE (SETCAL)

SETCAL takes the complex target mode shape, CTMS, and calculates the calibrated target mode shape, CATMS. Array CALM contains 10 calibration values for each microphone determined at the frequencies in array FRQS. The test frequency F is compared to FRQS to obtain the interpolation ratios D and DR. The calibration value, CAL, is then calculated and applied to CTMS resulting in CATMS. BMC is the boom microphone calibration factor and is no longer used in SMS.

4.4 TEST SUMMARY (TPRNT)

TPRNT is called to display a summary of the run. The user's name and the original and modified date are first displayed. Then the frequency values (CIPOS, CDPOS and NCSTP) and whether they are actual frequency or frequency parameter is printed. Finally, a summary of the mode shape to be set is printed.

19

4.5 ACQUIRING THE DATA (RSM)

Subroutine RSM can acquire data from one to 48 microphones, as determined by its two arguments NMICS, the number of microphones, and, MICS, an array of microphone numbers. The acquisition occurs in three nested do-loops. The outside loop performs MAVS passes over array MICS which is indexed by the center loop. The inner loop sums each microphone MAVT times. The results of each pass are stored in the virtual array RBUF. Averaging in this manner reduces the system sensitivity to small time variations in the acoustic field induced by the flow. Typically about 24 averages total must be performed to produce clean data with flow. This can take close to three minutes. If 24 averages are done at a time on each microphone (MAVS=1, MAVT=24), the acoustic field for microphone #1 might be different from the field when microphone #48 is averaged. The data is therefore not representative of the sound pressure field and difficult to optimize. By spreading out each microphone's data over time (MAVS=8, MAVT=3), this discrepancy is reduced (see section 2.1.5).

When either the multiplexer channel or the auto gain setting is changed, it takes about 20 milliseconds for the tracking filter to settle out. This time delay was determined by measuring the rise time of a signal while the channel of the multiplexer and setting of the auto gain unit was changed. The 'wait for' after SMPX was eliminated by placing the call before the summing of the data. This calculation was measured to take longer than 20 milliseconds.

RSM begins by testing switch 5. If set, the user is prompted to enter values for MAVS,MAVT and gain wait. The gain wait parameter is in TICS which is 10 milliseconds. The disc wait and SMPX wait are no longer used. The first channel is set on the multiplexer before the summing array, RBUF, is cleared and the data lower limit, LLIM, is set at 820 counts, or about 2 volts.

In the acquisition loop, variable IM is assigned the microphone number and the overload flag, OVRFLG, is set to false. The gain flag is no longer used. Before calling ATOD, the overload monitor is reset if this is the first sum for this microphone or if an overload occurred on the last sum. ATOD acquires NP points into IBUF from ATD channel JSYS. ISMPT is the number of microseconds between samples and was calculated in CKFRQ. The system gain is adjusted to eliminate overload of the differential receivers and to produce a filtered

20

signal at the A/D converter of between 2 and 5 volts. If function LDOVR returns a 1, an overload occurred and the gain is reduced. Subroutine MAXDP scans IBUF, subtracts the converter offset, 2048, and returns the absolute maximum value in variable MAX. If MAX is greater than or equal to 2047 counts (+/- 5 volts) the system gain is reduced. If MAX is less than LLIM, the gain is increased unless OVRFLG is set. Due to the tracking filter, it is possible to overload the differential receiver while the filtered signal is less than 2 volts. In this case, LLIM is set to MAX to increase the loop's efficiency.

Once the data pass all the limit tests, a check is made to see if this is the last sum for this microphone during this pass. If it is, the multiplexer is set to the next channel. If the center loop index is on the last microphone, the first microphone is selected for the next pass.

Finally the data are converted to floating point, adjusted by array FACT for the system gain, and summed into RBUF.

When the acquisition is completed, the total number of averages, MAVS*MAVT, is divided out of the sum and the absolute maximum value is determined. This maximum is used to normalize the data to 1900. The normalization improves the resolution in CMPAPH. Variable F is calculated to denormalize the results of CMPAPH.

$$F=1./(RMKG(IZ)*1638.4)$$

where

F = denormalization factor

RMKG = normalizing factor

1638.4 = 32768 counts/5 volts conversion factor from CMPAPH

CMPAPH is then called to curve fit the data and resolve its real and imaginary parts. The values are denormalized and stored in real array IAMS.

4.5.1 Maximum Data Point (MAXDP)

MAXDP is a MACRO subroutine which subtracts the A/D converter offset, 2048, and then searches for and returns the absolute maximum value.

4.5.2 Compute The Amplitude And Phase (CMPAPH)

CMPAPH computes the real and imaginary parts of a sinusoidal signal from digital data.

```
          CALL CMPAPH(IBUF,LNS,LNSF,NP,IRL,IMAG,MEAN)
```

where   IBUF is the input data array

                     LNS is the angle increment per data point (from XNS)

                     LNSF is the fractional remainder of LNS

                     NP   is the number of data points

                     IRL is the real part

                     IMAG is the imaginary part

                     MEAN is the mean

The values IRL,IMAG, and MEAN are scaled so that 1 volt equals 1638.4. CMPAPH
uses a least squares fit of the data to the function

$$Y(i) = A\cos(i\phi) + B\cos(i\phi) + C$$

where   A is the real part

                     B is the imaginery part

                     C is the mean

                     i is the data index

                     ϕ is the angle increment

By conventional least squares reduction techniques the equation is reduced to

$$\begin{bmatrix} S_{cc} & S_{sc} & S_c \\ S_{sc} & S_{ss} & S_s \\ S_c & S_s & N \end{bmatrix} \times \begin{bmatrix} A \\ B \\ C \end{bmatrix} = \begin{bmatrix} S_{xc} \\ S_{xs} \\ S_x \end{bmatrix}$$

where   S denotes the sum over N samples

                     c is the $\cos(i\phi)$

                     s is the $\sin(i\phi)$

                     x is the ith data point

and   Multiple subscripts denote summed products

Inversion of the 3 x 3 matrix provides the means of solving the equation to
yield the complex magnitude and mean value.

The first time called, CMPAPH calls CIS to return the sine and cosine of
each data point's relative angular position. LNS and LNSF are a double precision
integer representing the angle increment per data point. They were calculated in

22

INID from XNS which was calculated in CKFRQ. CIS uses table look-up and interpolation to determine the sines and cosines. CMPAPH uses these values to calculate the data independent coefficients of the matrix and then stores all the results for later use.

Once the sines and cosines and their associated parameters have been determined, CMPAPH can calculate the data dependent values. CMPAPH then calls INVT to invert the matrix and solve the equation.

INVT is called from CMPAPH with 8 parameters in array IA of common INVTA defining the matrix equation given in section 4.5.2. INVT expands the parameters into a 3 x 3 matrix and calls INVERS to calculate the matrice's inverse. INVT then solves the equation returning the results in array IA.

4.6 THE ERROR FUNCTION, PCOST (GMDSH)

GMDSH is a real function which returns as its value the system error, PCOST. GMDSH begins by constructing the MICS array for RSM by excluding all bad microphone numbers found in array IBM from the list of 48. RSM is then called to do the acquisition on the good microphones. Array ATMS is the calibrated target mode shape referenced as 96 real values, the odd indices being real, the even imaginary. Before calculating the error, the bad microphone values are replaced in the data array IAMS with the target values. The PCOST is then calculated as the sum of the square of the differences between ATMS and IAMS.

4.7 DETERMINING THE MODE SHAPE

Two subroutines are used to display mode shape data. WAMPP calculates and prints the calibrated amplitude and phase at each microphone. MODE resolves the data into an amplitude and phase for spinning mode orders -10 through +10.

4.7.1 The Microphones' Amplitudes And Phases (WAMPP)

WAMPP calculates the microphone calibration factors as in SETCAL and calibrates the data acquired in RSM. These data are then converted to amplitude in dB and phase in degrees for each microphone and displayed on the terminal.

4.7.2 Modal Content (MODE)

Subroutine MODE reduces the data acquired from RSM into an amplitude and phase at the duct wall for azimuthal spinning modes of order -10 to 10.

$$P(m) = \frac{1}{N} \sum_{i=1}^{N} A(i)e^{-jm \, PH(i)}$$

where

P(m) is the complex pressure level at the duct wall

m is the mode index (-10 to +10)

A is the calibrated complex pressure data

PH(i) is the ith microphone angular position

N is the number of microphones (48)

4.8 INITIALIZING THE DRIVERS (INID)

Subroutine INID fills the gap between the initial state of the system, all drivers off, and CKMODE which works best fine tuning a preset system. Making the assumptions that (1) for small changes in the driver setting, the relation between the driver setting and the acoustic field is linear, and (2) the desired amplitudes of sound at the source microphones can be approximated by first determining the individual driver settings that produce amplitudes close to the ideal at their corresponding microphones, the system can be quickly brought close enough to the target error to make it possible for CKMODE to converge in as few passes as necessary.

Subroutine INID first calls CKFRQ to set the frequency. It then calculates the double precision angle increment, LNS and LNSF, and clears the drivers' outputs. If variable IFINID is non-zero, INID calls DINCR to set the recalled driver settings and returns. Otherwise, INID driver initialization occurs in two steps. The first loop addresses one driver at a time, and through a linear approximation iteration, adjusts the driver input corresponding to the microphone (microphone # = 2*driver #) to within +/- 5 dB of its corresponding value in CATMS. Once this is accomplished, all the drivers are turned on and the average amplitude and phase difference between the results and the target is calculated. If the amplitude difference, AMPDFF, is greater than 3 dB, the target is adjusted by AMPDFF and the first loop repeated. If AMPDFF is between 1 and 3 dB the complex driver amplitudes are varied uniformally according to AMPDFF, by the same linear approximation technique as in step one, until the difference is less than 1 dB. When AMPDFF is below 1 dB, the driver settings are rotated until the average phase angle difference, PHDFF, is less than 3 degrees.

24

Before entering the first loop, complex variable CMP is set to (500.,0.) and DELTA is set to 5.0. CMP scales CTMS to a driver setting and DELTA adjusts the target values in array DRAMPS if AMPDFF is greater than 3 dB. Each driver is set to 15 dB below CATMS to allow for the increase in amplitude when they are all turned on at once. Array DRAMPS are the target amplitudes and each is set to 20 dB below CATMS. DELTA is then added to DRAMPS before entering the loop. Array AMPIC is the microphone amplitudes derived directly from CATMS and is used in the second part of INID to compute AMPDFF.

The first approximation begins by adding DELTA to DRAMPS and using the resulting amplitude in dB to calculate the corresponding voltage, DRVOLT. Entering the loop, variable I is the driver index and J the microphone index. The even microphones are used to sense the drivers. If J is found to be a bad microphone in IBM, J-1 is tried. If J-1 fails, INID prints the failing microphone numbers and stops. If a good microphone is found, CMP is then multiplied by the value in CTMS corresponding to the microphone being used. The resulting complex value, C, is used in the call to S1DR to set driver I. Array MICS is set to J and RSM is called to acquire one microphone. The amplitude at the sensing microphone, AMPA, is calculated from IAMS, and converted to dB and set in AMP. If AMP is within +/- 0.5 dB of DRAMPS, the driver setting, C, is stored in CDRI and the loop proceeds to the next driver. If not, the ratio of DRVOLT to AMPA adjusts CMP for the next iteration.

Once all the driver settings are found, DINCR is called to set the drivers and the next level of initialization begins. The PCOST is determined and the driver setting with the minimum PCOST is stored in CDRTMP. Minimizing the average amplitude and phase differences does not always result in the minimum system error. The amplitudes and phases for the target and acquired data are then summed. PDM1 is calculated as the phase shift between microphone two of the target and acquired data. Variable ISHOT provides different paths through this next section. If ISHOT equals 0, PHDFF is set to PDM1 to avoid rotating in the wrong direction on the first pass. If ISHOT equals 1, PHDFF is set to the average difference of the phases and if ISHOT equals 2, INID will exit after setting the drivers to CDRTMP and therefore the minimum PCOST.

INID prints the PCOST, AMPDFF and PHDFF and allows 10 seconds to raise switch one to end the iteration. If AMPDFF is greater than 3 dB , DELTA is set to minus AMPDFF and section one is repeated. If AMPDFF is between 1 and 3 dB, the ratio of the target to the acquired amplitudes is used to adjust the driver array and section 2 is repeated. Once AMPDFF is less than 1 dB, the driver settings are rotated according to PHDFF until PHDFF is less than 3 degrees. If AMPDFF raises above 1 dB it is readjusted. When PHDFF drops below 3 degress, the driver setting with the minimum PCOST, CDRTMP, is set and INID exits.

4.8.1 Set One Driver (S1DR)

S1DR has as its input a driver number and a complex setting. Each driver has a real and imaginery amplitude and an overall gain which is applied to all the drivers. The total gain is the product of the individual setting and the overall gain, IDRG. To achieve the greatest dynamic range for each driver, the overall gain is minimized. This is done by using the greatest of the real and imaginary amplitudes to determine an IDRG which would result in that amplitude being set to IDLIM, the maximum allowable driver setting. As long as the product of the two are equal, the resultant amplitude will be unchanged. If IDRG is greater than 255 or the driver setting excceeds IDLIM, an error message is printed and the variable is set to its corresponding limit. The overall gain and the real and imaginery amplitudes for that driver are then set by SGNP. If switch 2 is set, the driver number, overall gain, and the real and imaginary settings are printed on the terminal.

4.9 OPTIMIZING THE MODE SHAPE (CKMODE)

CKMODE attempts to lower the system error, PCOST, below the target error, TEF. The PCOST can be considered as a function of the 48 real and imaginary driver settings. By varying each element of the driver array and tabulating the extent of the effect on the PCOST, a 48 element vector can be established which is in the direction of decreasing error. As will be shown, the PCOST is a second order function of the driver settings. In CKMODE, once the direction vector is established, three points on the curve are determined by stepping along this vector. Using these three points, a second order polynomial is fit in search of a minimum value for PCOST. If the minimum is less than TEF, CKMODE exits, if not, a new vector is tabulated and the process repeated.

26

Assuming a linear relation between the driver settings and the sound pressure field for small deviations of the driver settings, the PCOST is a second order function of the driver setting array.

$$P = \alpha X + \beta$$

where   P is the sound pressure

      X is the driver setting array

and α, β are the slope and intercept

Given this relationship, the PCOST becomes

$$
\begin{aligned}
PCOST = Y &= (P - P_0)^2 \\
Y &= P^2 - 2P\,P_0 + P_0^2 \\
&= (\alpha X + \beta)^2 - 2P_0(\alpha X + \beta) + P_0^2 \\
&= AX^2 + BX + C
\end{aligned}
\tag{1}
$$

where   Y is the error function

      P is the pressure field

      P_0 is the target pressure field

The mode is then optimized by minimizing the error as defined by equation (1). CKMODE does this by stepping each driver setting by a constant amount, INCDR, and storing the difference between the original PCOST and the PCOST after the step. The resulting array, DRINC, contains values proportional to the amount of change INCDR made in PCOST and in the direction of lowering the PCOST. Three points are then computed; one at the original error, and two at driver settings augmented by DRINC multiplied by EPSI and twice EPSI.

   The driver settings are equal to

$$X = X_0 + \varepsilon D$$

where   X is the variable driver setting

      X_0 is the initial driver setting

      D is the DRINC array value

and ε is EPSI

Substituting this expression in (1) will yield the same second order relationship with different coefficients.

$$Y = a\varepsilon^2 + b\varepsilon + c \tag{2}$$

The minimum of (2) is given by

$$dY/dE = 2a\varepsilon + b = 0$$

$$\varepsilon(\min) = -b/2a \tag{3}$$

The curvature, $2a$, must be greater than zero for a minimum rather than a maximum to exist. Using the three points set, we have

$$Y_0 = c \quad \text{since } \varepsilon_0 = 0 \tag{4}$$

$$Y_1 = a\varepsilon_1^2 + b\varepsilon_1 + c \tag{5}$$

and

$$Y_2 = a\varepsilon_2^2 + b\varepsilon_2 + c \tag{6}$$

Solving equations (4), (5), and (6) for the unknown coefficient a yields,

$$a = (Y_2 - Y_1)/[\varepsilon_2(\varepsilon_2 - \varepsilon_1)] - (Y_1 - Y_0)/(\varepsilon_1\varepsilon_2) \tag{7}$$

or

$$a = T_1 - T_2 \tag{8}$$

where

$$T_1 = (Y_2 - Y_1)/[\varepsilon_2(\varepsilon_2 - \varepsilon_1)] \tag{9}$$

and

$$T_2 = (Y_1 - Y_0)/(\varepsilon_1\varepsilon_2). \tag{10}$$

Solving equations (4), (5), and (6) for the unknown coefficient b yields

$$b = (Y_1 - Y_0)(\varepsilon_2 + \varepsilon_1)/(\varepsilon_1\varepsilon_2) - \varepsilon_1(Y_2 - Y_1)/[\varepsilon_2(\varepsilon_2 - \varepsilon_1) \tag{11}$$

or

$$b = (\varepsilon_1 + \varepsilon_2) T_2 - \varepsilon_1 T_1 \tag{12}$$

All the variables in (8) and (12) are known and, therefore, sufficient to solve (3) for the minimum.

CKMODE begins by setting the miminum improvement in the PCOST, VIMP, to 25% of the target error. The PCOST is computed and the mode shape is displayed by WAMMP and MODE. CKMODE then asks the user if a phase shift is necessary. The driver settings can be rotated to give any desired angle of the mode of interest. If a shift is entered, it is applied to CDRVA, the PCOST is recomputed and the modal distribution redisplayed. Before optimization begins, the user has the opportunity to alter the driver increment, INCDR, and the step multiplier, EPSI. The minimum epsilon, CEPS, is computed as one-fiftieth of EPSI. Switch 6 is used to control CKMODE. It can be set to exit CKMODE and is checked before entering the optimization loop, during the loop and if the process fails to converge.

The first step in optimizing is to apply INCDR to the real and imaginary part of each driver, IADRV, using subroutine DRVINC. Array DRVA holds the original settings. The resulting system error, V, is determined and the difference between the reference PCOST and V is stored in DRINC. INCDR is then removed from IADRV. The values in DRINC are now proportional to the change of the error function for a uniform change in each driver, (i.e., the partial derivative of PCOST with respect to each driver) and in the direction of decreasing error. When all 48 values of DRINC have been determined, the second stage of optimization is entered.

The three points, Y_0, Y_1, Y_2, used to solve the second order equations are measured. The original PCOST(Y_0) is kept in PCM1 to measure the improvement in the error against VIMP after the process completes. The minimum error found is stored in CMIN. DINCR is called to add EPSI* DRINC and 2*EPSI*DRINC to DRVA to get Y_1 and Y_2, respectively. The values Y_0, EPSI(X_1), Y_1, 2*EPSI(X_2), and Y_2 are used to solve equations (8) and (12) for XMIN, the minimum error. Two solutions are flagged as errors. It is possible to fit a negative curvature to the three points, indicating a maximum has been found rather than a minimum. Also, if XMIN is negative, the assumption that DRINC is in the direction of decreasing error is violated. Both these conditions are treated by determining the direction of the three points and doubling EPSI if they are decreasing and halving EPSI if increasing, in an attempt to get a good fit. EPSI may be doubled 5 times and halved once before a test for convergence is made.

The three remaining conditions are

(1) $X_0 < XMIN < X_1$

(2) $X_1 < XMIN < X_2$

(3) $X_2 < XMIN$

The procedure now is to replace X_1 with XMIN and refit the curve. This would continue until one of the states equaled the derived minimum. For condition (1), X_1 is moved to X_2 and XMIN becomes X_1. For condition (2), X_1 is moved to X_0 and XMIN becomes X_1. Since XMIN is outside the bounds of the three points in condition (3), EPSI is doubled and Y_2 is retaken. Again EPSI can be doubled only 5 times.

There are four ways out of the optimization loop: one of the points is less than the target error, one of the points is on the projected minimum, EPSI has been doubled 5 times, and EPSI has been halved to less than CEPS. When one of these conditions is met, CMIN is tested to be less than the previous minimum by at least 25% of TEF. If not, a message is printed saying the process failed to converge. Switch 6 controls whether CKMODE is repeated or not. If CMIN meets the requirements, CKMODE is repeated.

4.9.1 Stepping The Drivers (DRVINC,DINCR)

Subroutine DRVINC is called in the initial phase of CKMODE to add INCDR to the driver settings. DINCR is used to add the product of the direction vector, DRINC, and epsilon, ϵ, to the driver settings. DINCR also selects the over-all gain, IDRG, which will result in the greatest dynamic range for the drivers (as in S1DR). INID uses DINCR for this purpose.

DINCR has four arguments: the original driver setting, FR; the resulting setting TO; the multiplier, EPS; and the increment array, CH. CH is multiplied by EPS and added to FR to give TO. The maximum value in TO is saved in XAM. The overall gain is calculated from XAM. The gain is then used to scale TO before setting the drivers. If the gain is greater than 255 or a driver setting surpasses DLIM, a message is printed warning of driver overflow.

5.0 AUXILIARY SYSTEMS

Once the mode has been set, there are several systems available to aid in data acquisition and analysis. Those unique to SMS are

    1. Auxiliary microphone multiplexer

    2. Pressure scanner and transducer

    3. Spinning mode rake

    4. Radial microphone probes

Among the general purpose laboratory devices are:

    1. BCD meter multiplexer

    2. SD360 FFT processor

    3. GPIB instrumentation bus

5.1 AUXILIARY MICROPHONE MULTIPLEXER (SMPX1)

There are 16 channels available on the auxiliary multiplexer for conditioned microphone input. The auxiliary multiplexer output is fed through SMS multiplexer channel 52. Subroutine SMPX1 will set the auxiliary multiplexer channel.

CALL SMPX1(ICHAN)

where   ICHAN is the channel number 0-15.

If SMPX1 detects an error, the negative of the channel actually set minus one will be returned in ICHAN.

5.2 PRESSURE SCANNER AND TRANSDUCER (ISCANN)

The pressure scanner multiplexes up to 48 pressure ports into the transducer. Integer function ISCANN is used to set the port number.

ISET = ISCANN(ICHAN)

where   ICHAN is the desired port channel 0-47

and   ISET is the port actually set.

The output of the pressure transducer is usually read by a volt meter on the GPIB bus.

5.3 SPINNING MODE RAKE (RAKE)

The spinning mode rake holds four peripherally mounted microphones in a rotating fixture. The rake is driven to a postion by a D.C. motor controlled by an analog controller. The controller has 26 preprogrammed positions. In the present application, 20 postions, 0-19, have been programmed at 9 degree intervals from 0 to 171 degrees.

CALL RAKE(IPOS,IE)

where   IPOS is the postion number, 0-20

and   IE is zero if successful and 1 if not.

RAKE uses subroutines LCSA and LCSADO to interface to the controller (see section 5.5).

5.4 RADIAL MICROPHONE PROBES (RADPOS)

There are four radial microphone probes. The probes are driven by stepper motors controlled by the digital controller. The first two actuators on the controller drive probes capable of traversing the full duct diameter and are mounted at fixed angular positions. The second two actuators drive probes mounted on the SMS drum. The drum is capable of +/-180 degrees rotation. The two probes are mounted 4 inches apart, one up stream from the other, and are capable of .5 diameter transverse. All four microphones are attached to the auxiliary multiplexer.

    CALL RADPOS(R,IZ,IH,IERR)

where   R is the radial position in inches
             IZ is the actuator number (1,2,3 or 4)
             IH = 1; R = R + DN(IZ)
             IH = -1; R = R - OFST(IZ) + DN(IZ)
             IH = 2; R = UP(IZ)
             IERR= -1 if not successful
and   UP(IZ) is the upper limit for actuator IZ
             DN(IZ) is the lower limit
             OFST(IZ) is an offset.

The arrays UP, DN and OFST are passed through labeled common.

    COMMON/LIMITS/UP(4),DN(4),OFST(4)

RADPOS uses routines LCSDG and LCSDR from module LCSMITH to interface to the controller.

5.5 STEPPER MOTOR CONTROLLER (LCSMITH)

Module LCSMITH contains 4 subroutines, two each for the analog and digital controllers. To move the spinning mode rake

    CALL LCSA(IPOS)
    CALL LCSADO

where   LCSA sets the controller to preprogrammed position IPOS
and   LCSADO waits for completion of move .

32

The analog controller has 26 preprogrammed positions. With the spinning mode rake, 21 are programmed, 0-171 degrees at 9 degree intervals, and the last position is to home.

To move one of the radial probes

CALL LCSDG(IZ,IPOS, [ITIME])

where

IZ is the actuator, or probe, number

IPOS is the position scaled 1000 counts/inch

ITIME is an optional timeout in seconds

The position of any probe can be read.

CALL LCSDR(IZ,IPOS)

where

IZ ·is the actuator number

IPOS is the scaled position.

A labeled common is available to allocate the devices' logical unit numbers and examine their status.

COMMON/LCSMIT/LUNA,LUND,IOSTAT(2),IODSTA(2)

where

LUNA,LUND are the LUNS of the analog and digital controllers, respectively

IOSTAT,IODSTA are two word integer status arrays holding device status of the analog and digital controllers, respectively.

The first status word contains the standard I/O status (see RSX11M mini reference). The second word will be 1 if the probe hit the upper limit and -1 if the lower limit.

5.6 THE SMS ROTATING DRUM (DRMPOS,DRMROL)

Radial probes 3 and 4 are mounted on the SMS rotating drum. The drum can be rotated +/-180 degress allowing complete access to two duct cross sections.

CALL DRMROL(ANGLE, [HMCNT])

where

ANGLE is the desired azimuthal position

and

HMCNT is an integer variable returning the number of counts remaining +/- when the drum passed home.

The first call to DRMROL will home the drum. This is necessary because the drum has no position feedback except for the home sensor. Home position is vertical and taken as angle 0.0. Whenever the drum passes home, the remaining counts are passed to HMCNT, if HMCNT was given. HMCNT should be zero, but a value less than four is acceptable. The drum rotates at 12 steps/degree.

Subroutine DRMPOS can be used to rotate the drum and postion the LCS radial probes #3 and #4.

CALL DRMPOS(R,TH,IZ,IH,IERR)

where

R = the radial position in inches, 0-5.895

TH = drum angular position, -180 to +180 degrees

IZ = LCS probe, 3 or 4

IH = ·1 to position probe IZ to R,TH

IH · = 2 to position probe IZ to upper limit

$IERR$ = 1 if successful, -1 if not.

Before calling DRMPOS a logical unit number must be set for the digital controller in labeled common LCSMIT, variable LUND. Subroutine DRMROL assumes LUN 2 has been assigned to DR11B and LUN 4 to the LPS.

5.7 BCD MULTIPLEXER

There are many devices attached to the BCD multiplexer, but of significant consequence here are the three DORIC temperature meters. The thermocouples are movable and, therefore, have no permanent assignment.

CALL MUXINT(LUN)

.

.

.

CALL DORIC(IDEV,DISP)

where   MUXINT assigns logical unit number LUN to the BCD multiplexer

and   DORIC reads meters IDEV and returns the value in
                   centigrade in DISP.

34

5.8 SD360 FFT PROCESSOR (SDLIB)

The SD360 does dual channel acquisition and analysis. Functions available are spectrum, correlation, coherence, covariance, transfer function and signal enhancement. A library, [10,16] SDLIB, is available for use to remotely control the SD360. For information do HELP SD360 on the computer.

5.9 GPIB INSTRUMENTATION BUS (IELIB)

The GPIB interfaces to any device conforming to IEEE-488 standards. Presently available are the HP 3325 synthesizer, Fluke 8500 DVM, and Fluke 2204 multiplexer. Subroutines to drive these devices are available in [10,17] IELIB. For more information do HELP IELIB on the computer.

6.0 CALIBRATING

Before a test can be run, the acquisition systems must be calibrated. These include the SMS system, the source microphones and the test microphones. While most calibrations are performed by technicians, it is beneficial and sometimes necessary for the researcher to know the procedure and how to apply the results.

6.1 SMS System Calibration

The SMS system calibration factors contain all the components from the source microphones to the LPS. This includes the multiplexer, auto-gain unit, cabling and tracking filter. There is actually a factor for each setting for the auto-gain unit. The calibration is determined by injecting a signal of known amplitude into the system before the multiplexer and measuring the output before the ATD for each setting of the auto-gain unit. The factors are kept in array FACT in SMS block data.

6.2 MICROPHONE CALIBRATION

The SMS source section microphones, plus any test microphones, must be calibrated. The calibrations are taken at a set level and a number of frequencies. A condenser microphone is usually used as a reference and both the reference and the unknown are mounted in a fixture along with a driver. The frequency and the amplitude are set according to the reference microphone. The resulting ratio between the reference and unknown is the calibration factor. The factors are then stored in a file for use by the acquisition programs.

6.2.1 Test Microphone Calibration

Program MC4 is used to calibrate up to 8 microphones at up to 20 frequencies. MC4 uses file CAL.DAT as input and output. The format of CAL.DAT is

$$
\begin{array}{lll}
\text{Record 1} & \text{DATE} & @ \text{ 5A2} \\
\text{2} & \text{SPL,NFR} & @ \text{ f8.3,I3} \\
\text{3} & \text{VODB} & @ \text{ E12.5} \\
\text{4 - on} & \text{F,A,B} & @ \text{ F7.1,F8.4,F7.1}
\end{array}
$$

where   DATE is the ASCII day-month-year the file was written

SPL   is the number of frequencies

VODB is the 0 dB voltage value

F   is the frequency array

A   is the amplitude calibration

B   is the phase calibration

The test microphones and reference microphone are mounted in a fixture along with a driver attached to amplifier 1. MC4 first reads all of CAL into array storage and then asks for the number of micrphones to calibrate and their addresses. The program begins by setting the frequency and SPL according to the values in CAL. The driver is adjusted until the reference microphone sees the SPL specified. The test microphone amplitudes are then taken and stored. When all the frequencies have been taken, a new copy of CAL.DAT is written.

6.2.2 SMS Source Microphone Calibration

All 48 SMS source microphones must be calibrated periodically. The calibration file, DRMCAL.DAT, is formatted as

$$
\begin{array}{lll}
\text{Record} & \text{1} & \text{TITLE,DATE } @24A2 \\
& \text{2} & \text{blank} \\
& \text{3} & \text{blank} \\
& \text{4} & \text{NFR } @ \text{ I5} \\
& \text{5} & \text{FRQS } @ \text{ 10F8.0} \\
& \text{6 - 53} & \text{CAL } @ \text{ 10F8.4} \\
& \text{54} & \text{BOOM } @ \text{ 10F8.4} \\
& \text{55} & \text{NMB } @ \text{ I5} \\
& \text{56} & \text{IBM } @ \text{ I5}
\end{array}
$$

where   TITLE is 'MANUAL CALIBRATION'

                     DATE is the ASCII date

                     NFR   is the number of frequencies

                     FRQS is the frequency array

                     CAL   is the cal factor array

                     BOOM is not used

                     NBM   is the number of bad microphones

                     IBM   is the bad microphone array.

To avoid destroying DRMCAL, an intermediate file, MANCAL.DAT, is used by MC2. Once all 48 microphones have been calibrated program MC3 is run to transfer the data to DRMCAL. File MCAL.DAT is read by MC2 to input NFR and FRQS. MC2 first opens the files and reads MCAL. MC2 then asks for the microphone number to start with and begins calibration of that microphone by setting the first frequency and then adjusting the driver until the reference microphone reads between 3.5 and 4.0 volts. Once this level has been set, the ratio between the source and reference microphones is taken and stored. The process continues until all the frequencies are set. The program then asks if it should continue with the next microphone or retake the present one. If the CAL was good, the next microphone is mounted in the fixture, and the process continues until all the microphones are calibrated. When the last microphone, #48, is completed, MC2 asks if it should save the data in MANCAL, retake the data, or exit. Once MANCAL contains all good data, MC3 can be run to transfer it to DRMCAL.DAT. The bad microphone data, if any, are then added with the EDITOR when necessary.

TABLE I. - CONTROL SWITCH DEFINITIONS

Panel Switch	Function
1	Discontinue INID operation
2	Diagnostics
3	Sequence the test (frequency steps)
4	Not used
5	Input enhancement parameters
6	Control optimization
7	Print driver settings
8	Stop

TABLE II. - MULTIPLEXER CHANNEL ASSIGNMENTS

CHANNEL	FUNCTION
0 - 47	SOURCE SECTION MICROPHONES
48 - 51	UNUSED
52	FAR FIELD MICROPHONE INPUT
53 - 55	TEST SIGNALS
56 - 63	UNUSED

TABLE III. - INPUT OUTPUT DEVICES

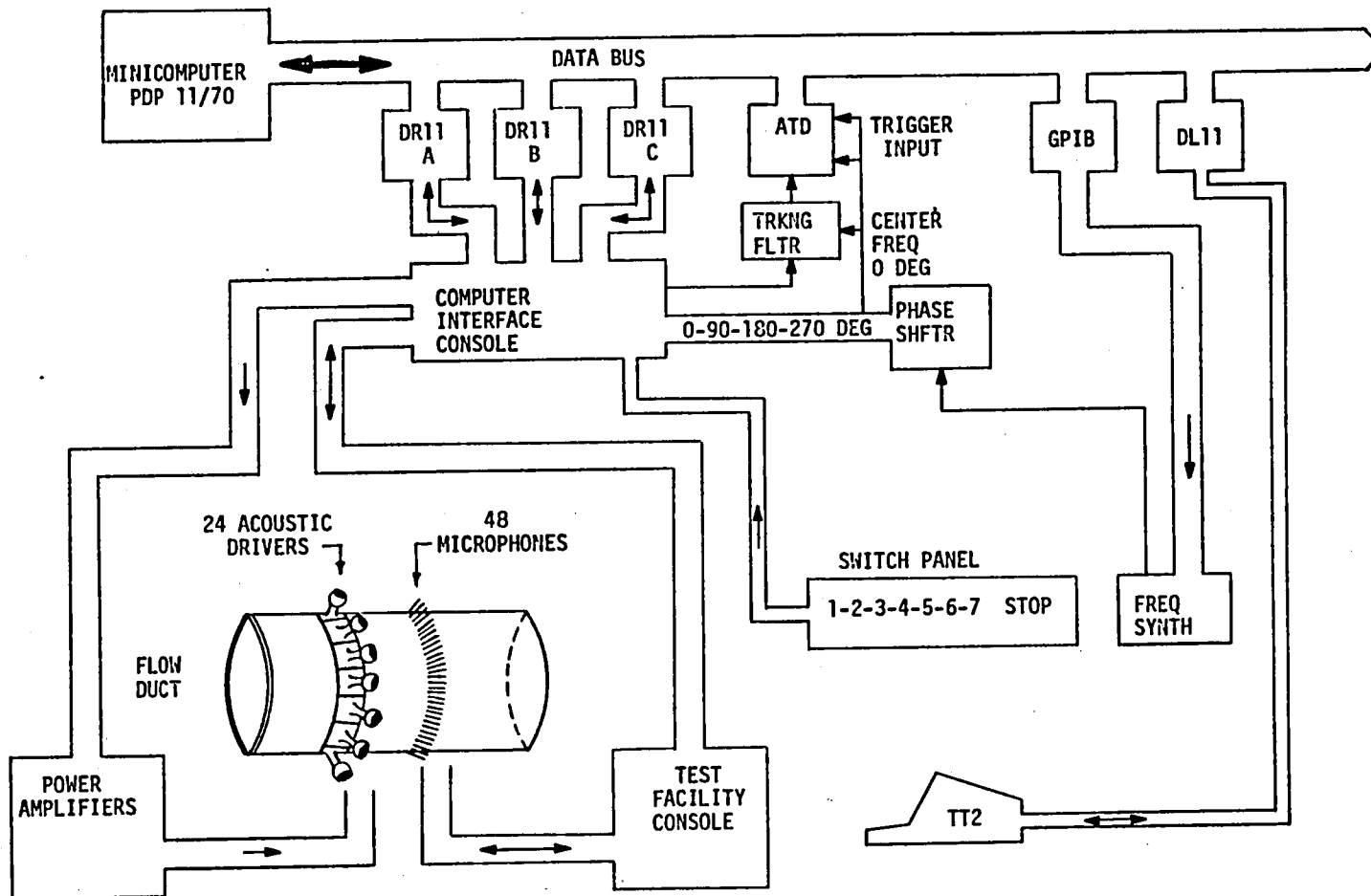| DEVICE	FUNCTION	SUBROUTINE
DR11A	IO.STC IO.WLB IO.RLB	SGNP, GPSW SGNP, SMPX, GPSW GPSW
DR11B	IO.STC IO.WLB	SMKG SMKG
DR11C	IO.WLB IO.RLB	OVRRST LDOVR
LPS	IO.XREG IO.YREG IO.DTOA IO.DMR IO.DMA IO.CLB	SOV SOV SOV ATOD ATOD GFRQ

Figure 1. - Overall system configuration.

DR11 A-167750

DR11 A-167750	Bit 15	Bit 14	Bit 13	Bit 12	Bit 11	Bit 10	Bit 9	Bit 8	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	CSR 1	CSR 0
OUTPUT			Multiplexer Select						Gain/Phase – Control Panel									
INPUT									Switch Control Panel									

DR11 B-116740

DR11 B-116740	Bit 15	Bit 14	Bit 13	Bit 12	Bit 11	Bit 10	Bit 9	Bit 8	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	CSR 1	CSR 0
OUTPUT	Auto Gain								Drum Stepper Motor									
INPUT		Home Sen											Auxiliary 16 CH. Multiplexer					

DR11 C-167730

DR11 C-167730	Bit 15	Bit 14	Bit 13	Bit 12	Bit 11	Bit 10	Bit 9	Bit 8	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	CSR 1	CSR 0
OUTPUT	Lorch Oscillator						OVLD					Aux. 16 CH. Multiplexer			Scanni-Valve			
INPUT	Servo Tek Status								Scannivalve									

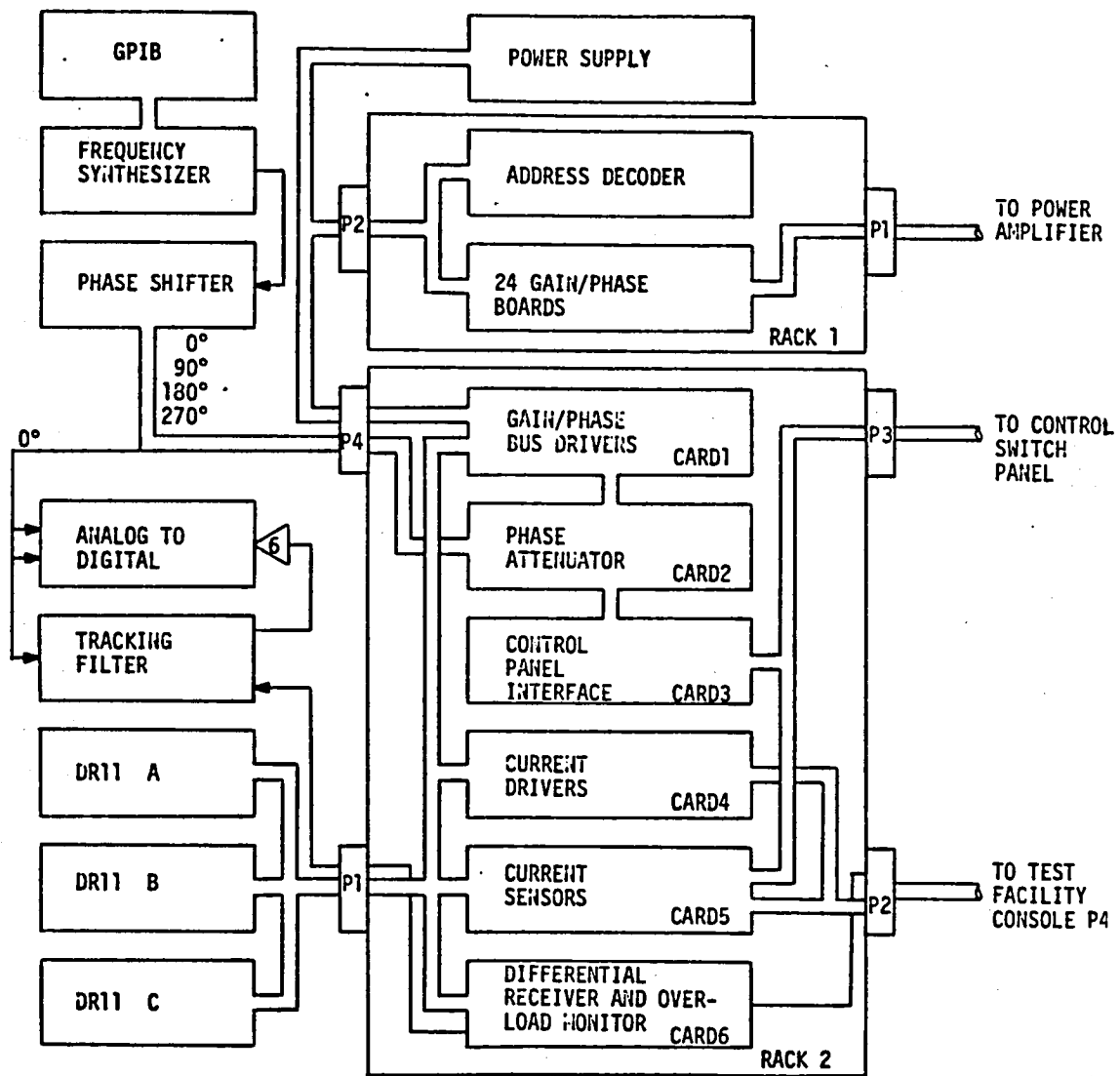Figure 2. – Functions used for DR11 input output cards

Figure 3. - Computer interface console
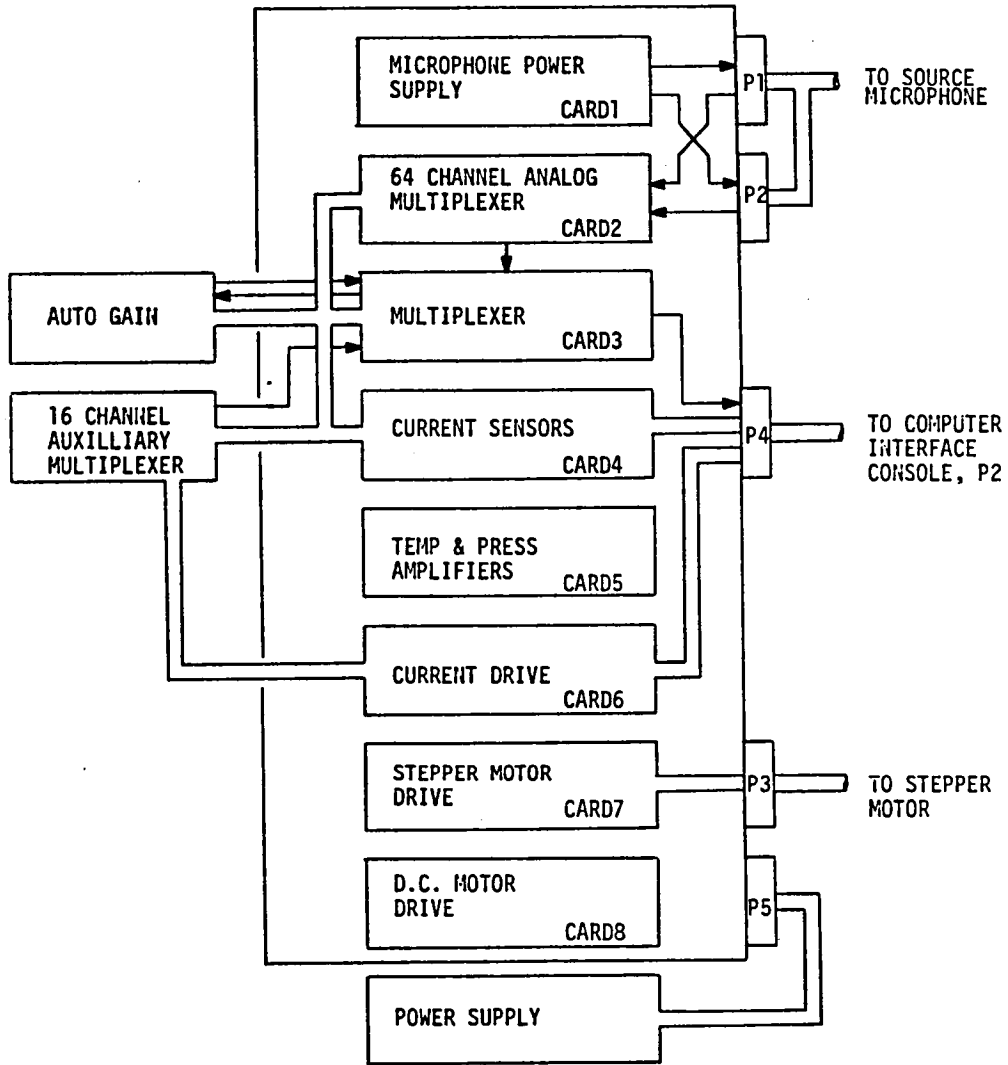
Figure 4. - Test facility console

TEST	Bit 15	Bit 14	Bit 13	Bit 12	Bit 11	Bit 10	Bit 9	Bit 8	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
A-ANALOG OUTPUTS	Channel No.				Output Value 0 = -5 , 4095 = +5											STOP
B-GAIN PHASE	Gain/Phase Address							Sign	. Magnitude							STOP
C-MPX MIC. GAIN		Multiplexer Address									Microphone Gain					STOP
D-DISPLAY CONTROL PANEL																STOP
E-TEST CLOCK																STOP
F-AD TRANSFER		Input Channel														STOP
G-DMA TRANSFER		Input Channel														STOP
H-OVERLOAD																STOP

Figure 5. - Computer switch utilization during diagnostics

1. Report No. NASA CR-165698	2. Government Accession No.	3. Recipient's Catalog No.
4. Title and Subtitle AN OPERATIONS MANUAL FOR THE SPINNING MODE SYNTHESIZER IN THE LANGLEY AIRCRAFT NOISE REDUCTION LABORATORY		5. Report Date March 1981
		6. Performing Organization Code
7. Author(s) Daniel L. Palumbo		8. Performing Organization Report No.
		10. Work Unit No.
9. Performing Organization Name and Address Kentron International, Inc. Hampton Technical Center Hampton, Virginia 23666		11. Contract or Grant No. NAS1-16000
		13. Type of Report and Period Covered Contractor Report
12. Sponsoring Agency Name and Address National Aeronautics and Space Administration Washington, DC 20546		
		14. Army Project No.

15. Supplementary Notes

Langley Technical Monitors: Richard Silcox and Robert Grandle
Topical Report

16. Abstract

This report provides documentation of the current configuration of the Spinning Mode Synthesizer incorporated into the flow duct facility of the Aircraft Noise Reduction Laboratory at NASA Langley Research Center. Included are an introduction to the system, operating instructions, a detailed description of the system, and a users' guide to the data acquisition packages available.

17. Key Words (Suggested by Author(s)) spinning mode synthesizer computer interfaces acoustic data acquisition	18. Distribution Statement Unclassified - Unlimited Subject Category 71		
19. Security Classif. (of this report) Unclassified	20. Security Classif. (of this page) Unclassified	21. No. of Pages 48	22. Price* A03