CONTRACTOR REPORT

TRANSONIC COMPRESSOR: PROGRAM SYSTEM TECHO FOR DATA ACQUISITION AND ON-LINE REDUCTION

H. ZEBNER
BDM CORPORATION
P.O. Box 2019
Monterey, CA 93940

October 1980

Interim Report for Period
October 1979 – September 1980

Approved for public release; distribution unlimited

Prepared for:
Naval Postgraduate School
Monterey, California 93940
The work reported herein was carried out for the Naval Postgraduate School by the BDM Corporation under Work Order 426, under Contract Number N00014-78-C-0204. The work contributed to the program entitled Transonic Compressor Investigations funded in part by Work Request N00019-80-N00-51199 and under the cognizance of Dr. H. J. Mueller, and to the program entitled Axial Compressor Flow Fields funded in part by Work Request N00014-79-C-07890 and under the cognizance of Mr. J. E. Patton, Jr.

The author is a candidate for the degree D. Ing under Prof. H. Gallus at Rhein-Westf. Techn. Hochschule, 51 Aschen, W. Germany.

This report was prepared by:

[Signature]

BDM Corporation
P.O. Box 2019
Monterey, CA 93940

Publication of the report does not constitute approval of the sponsor for the findings or conclusions. It is published for information and for the exchange and stimulation of ideas.
**Title:** Transonic Compressor: Program System TXCO for Data Acquisition and On-Line Reduction

**Author(s):** H. Zebner

**Performing Organization:** BMI Corporation

**Contractor Report:** October 1979 - September 1980

**Report Date:** October 1980

**Number of Pages:** 279

**Distribution Statement:** Approved for Public Release; distribution unlimited

**Abstract:**

A system of data acquisition and reduction programs, TXCO is described. The programs were written for the transonic compressor test facility at the NPS Turbopropulsion Laboratory which is served by an HP1000 series computer operating under RTE-IVB. However, the structure of the program system (strict separation of acquisition and reduction, store raw data as acquired, routines to verify the data system, etc.) is of more general interest, and allows the system to be applied to any test rig. The introduction of a "program control..."
20. Array accelerates execution and provides means for communication between programs, which otherwise execute individually.
ABSTRACT

A system of data acquisition and reduction programs, TXCO is described. The programs were written for the transonic compressor test facility at the NPS Turbopropulsion Laboratory which is served by an HP1000 series computer operating under RTE-IVB. However, the structure of the program system (strict separation of acquisition and reduction, store raw data as acquired, routines to verify the data system, etc.) is of more general interest, and allows the system to be applied to any test rig. The introduction of a "program control array" accelerates execution and provides means for communication between programs, which otherwise execute individually.
ACKNOWLEDGMENT

This study was made possible by Professor Ray Shreeve, Director of the Turbopropulsion Laboratory (TPL) of the Department of Aeronautics, and by Professor Dr. Ing Heinz Gallus, Institute for Aero Engines and Turbomachines of the Technical University Aachen, West Germany; to both I express my thanks. Jim Hammer, the TPL Laboratory Manager, contributed to this program system by letting me share his brilliant knowledge and vast experience in the areas of instrumentation and turbomachines.
# TABLE OF CONTENTS

<table>
<thead>
<tr>
<th>Section</th>
<th>Page No.</th>
</tr>
</thead>
<tbody>
<tr>
<td>1. INTRODUCTION</td>
<td>6</td>
</tr>
<tr>
<td>2. GUIDE TO THE PROGRAM DESCRIPTIONS</td>
<td>10</td>
</tr>
<tr>
<td>3. PROGRAM TXCOØ</td>
<td>13</td>
</tr>
<tr>
<td>3.1. Flow Chart of Program TXCOØ</td>
<td>13</td>
</tr>
<tr>
<td>3.2. Program Listing, TXCOØ</td>
<td>29</td>
</tr>
<tr>
<td>4. PROGRAM TXCOL</td>
<td>43</td>
</tr>
<tr>
<td>4.1. Description</td>
<td>43</td>
</tr>
<tr>
<td>4.2. Subroutine ABSRV</td>
<td>53</td>
</tr>
<tr>
<td>4.3. Subroutine CALIB</td>
<td>62</td>
</tr>
<tr>
<td>4.4. Subroutine FREER</td>
<td>70</td>
</tr>
<tr>
<td>4.5. Subroutine PACER</td>
<td>84</td>
</tr>
<tr>
<td>4.6. Program Listing, TXCO1</td>
<td>104</td>
</tr>
<tr>
<td>5. PROGRAM TXCO2</td>
<td>133</td>
</tr>
<tr>
<td>5.1. Description</td>
<td>133</td>
</tr>
<tr>
<td>5.2. Subroutine COMB</td>
<td>140</td>
</tr>
<tr>
<td>5.3. Subroutine STDY</td>
<td>154</td>
</tr>
<tr>
<td>5.4. Program Listing, TXCO2</td>
<td>169</td>
</tr>
<tr>
<td>6. PROGRAM TXCO3</td>
<td>186</td>
</tr>
<tr>
<td>6.1. Description</td>
<td>186</td>
</tr>
<tr>
<td>6.2. Subroutine CHECK</td>
<td>192</td>
</tr>
<tr>
<td>6.3. Subroutine CHNGE</td>
<td>209</td>
</tr>
<tr>
<td>6.4. Program Listing, TXCO3</td>
<td>218</td>
</tr>
<tr>
<td>7. UTILITY SUBROUTINE PACKAGE TXCOU</td>
<td>229</td>
</tr>
<tr>
<td>7.1. Description</td>
<td>229</td>
</tr>
<tr>
<td>7.2. Program Listing, TXCOU</td>
<td>241</td>
</tr>
<tr>
<td>8. DATA REDUCTION PROGRAMS</td>
<td>259</td>
</tr>
<tr>
<td>9. CONCLUSIONS AND RECOMMENDATIONS</td>
<td>260</td>
</tr>
<tr>
<td>APPENDIX A. DATA ACQUISITION WORK SHEETS</td>
<td>262</td>
</tr>
<tr>
<td>A1. Data Locations</td>
<td>263</td>
</tr>
<tr>
<td>A2. Steady State Data Array</td>
<td>265</td>
</tr>
<tr>
<td>A3. Program Control Array (CNTRL)</td>
<td>267</td>
</tr>
<tr>
<td>A4. Paced Data Array</td>
<td>269</td>
</tr>
<tr>
<td>APPENDIX B. LINEAR APPROXIMATION BY METHOD OF LEAST SQUARES</td>
<td>270</td>
</tr>
<tr>
<td>REFERENCES</td>
<td>273</td>
</tr>
</tbody>
</table>
1. INTRODUCTION

This report describes a system of data acquisition and reduction programs, designed to acquire data from the Turbo-propulsion Laboratory's transonic compressor test facility. The computer hardware consists of an HP21MX mini computer with various peripheral devices. Both steady-state and high speed data are required to be taken as scheduled during a compressor test. The entire hardware configuration is shown in Fig. 1.

At the outset, the system of computer programs was required to do the following:

(i) Control via the "Interface Bus", measurement devices such as Scanivalves (S/V's), Scanivalve controllers, scanners, digital voltmeters (DVM), digital counters, analog to digital converters, and the acquisition timing device called PACER.

(ii) Perform data acquisition as efficiently as possible, store data in disc files, and document the test conditions.

(iii) Provide a means to check the data system (e.g., SUBROUTINE CHECK, Section 6.2).

(iv) Provide a means to verify the raw data (e.g., SUBROUTINE PICTR, called from SUBROUTINE PACER, Section 4.5. PICTR uses the auxiliary terminal to display the acquired wave form).
(v) Provide a means for the operator to communicate interactively. Since the operator at the system console is usually the investigator or research engineer and not a computer specialist, the program flow and the programmed interactive messages were required to be clear, logical and easy to understand.

The demand to speed up the data acquisition conflicted with the requirement of keeping the dialogue between program and operator clear. Interactive programs necessarily have extensive input-output operations which slow down the execution of the program. A reasonable compromise between these two choices was the introduction of a "program control array", CNTRL, whose elements - once pre-assigned - relieved the operator from entering routine decisions (e.g., telling the subroutines FREER and PACER how many Kulite signals are to be recorded and where to locate them; see Appendix A3: CNTRL(23) through CNTRL(246)). Additionally the control array provides accounting data (e.g., the sequential number for raw data files).

In the present report complete documentation is given of the program system "TXCO". The system consists of a "father" program, TXCO#, which, in operation, calls on a series of "son" programs TXC01, TXC02 or TXC03.

The father program, TXC0#, offers the investigator a menu of program branches to be scheduled according to a single digital entry as follows:
1. Survey using the type 'A' and the type 'B' ABSRV Kulite semiconductor pressure probes (Ref 1 and 2).
2. On-line calibration type 'A' and type 'B' CALIB probe.
3. Acquisition of high speed data through the FREER fast A/D converter, which is operated in free run mode.
4. Acquisition of high speed data through the PACER fast A/D converter, which now is controlled by a timing device, the PACER (Ref 3).
5. Radial flow survey using a temperature-pneumatic four hole COMBINATION PROBE.
6. Acquisition of all steady state data. STDY
7. Check the instrumentation. CHECK
8. Change the program control array. CHNGE
9. Reduce high speed data from the 'A' - 'B' REDAB probe system. REDAB uses the data gathered by ABSRV.
10. Reduce flow data from the combination probe. REDCO REDCO uses the data gathered by COMB.
11. Reduce steady state data and add this REDST operating point to the compressor performance map. REDST uses the data gathered by STDY.
The investigator selects the desired program module by entering the appropriate number between 1 and 11. Entering 12 halts the program. Subroutines ABSRV, CALIB, FREER and PACER - they handle the high speed data - are contained in PROGRAM TXCO1 (Section 4). Subroutines COMB and STDY - they handle the steady state data - are contained in PROGRAM TXCO2 (Section 5). Subroutines CHECK and CHNGE - they are used by the operator to control the program flow and verify the data system - are contained in PROGRAM TXCO3 (Section 6). After the select code is entered, and verified either by entering an additional parameter or tapping the RETURN key, the "father" program suspends its operation while the desired "son" program (TXCO1, TXCO2 or TXCO3) executes. The entire TXCO-system works interactively with the operator and displays as many informative messages as possible.

The program descriptions in the following sections explain, in user-manual form, how to handle each subroutine. The descriptions often resemble each other, which in the interests of utility was deliberately not avoided. A compressor failure prevented the author from using the programs for compressor test runs. The report is therefore presented with only a very short section of conclusions and recommendations. The program system is not considered to have been perfected, since little experience has been gained with its operation other than in "dry" runs.
2. GUIDE TO THE PROGRAM DESCRIPTIONS

Detailed descriptions of the programs are given in the following sections. First, in Section 3, a flow chart and listing are given for the father program TXCO0. Then, the descriptions given in Sections 4 through 6 (of TXCO1 through TXCO3) are structured as follows:

PROGRAM XXXX (or SUBROUTINE XXXX): PURPOSE:
A brief description of the purpose of this particular program module is given, and its capabilities and restrictions are noted.

ARGUMENTS: If the program module is a subroutine, which is called with parameters, the parameter list is explained.

EXTERNALS: The externals of each program module are listed. This information is necessary when loading the relocatable binary version (indicated by the % sign as first character of the disc file name under the RTE-IV operating system).

COMMON BLOCKS: The members of the COMMON blocks and their length in 32-bit words are listed and explained.

Mnemonic Abbreviations: The mnemonic acronyms which each program module uses are listed and explained.

ERROR MESSAGES: If a salvageable error occurs during the execution of a program module, an error message with suggestions for how to resolve the problem are described.
PROCEDURE: This subsection, which should always be used together with the flow chart, describes how to go through the program module. Hints for how to utilize all program features are given.

DATA FILE: The date file name is explained for all program modules, which save data. The first two characters are typical for the type of data which the file contains; for example,

<table>
<thead>
<tr>
<th>Data File Name</th>
<th>Type of Data</th>
<th>Created By</th>
</tr>
</thead>
<tbody>
<tr>
<td>T1rrss</td>
<td>'A'-'B' probe survey</td>
<td>ABSRV</td>
</tr>
<tr>
<td>T2rrss</td>
<td>free run sample</td>
<td>FREER</td>
</tr>
<tr>
<td>T3rrss</td>
<td>paced run sample</td>
<td>PACER</td>
</tr>
<tr>
<td>T4rrss</td>
<td>all raw steady state data</td>
<td>STDY</td>
</tr>
<tr>
<td>T5rrss</td>
<td>combination probe survey</td>
<td>COMB</td>
</tr>
</tbody>
</table>

rr — # of test run
ss — sequential # of data file type

The following modules are synchronized through the data file:

<table>
<thead>
<tr>
<th>Data Reduction Program</th>
<th>File Name</th>
<th>Data Acquisition Program</th>
</tr>
</thead>
<tbody>
<tr>
<td>REDAB</td>
<td>↔ T1rrss</td>
<td>↔ ABSRV</td>
</tr>
<tr>
<td>REDST</td>
<td>↔ T4rrss</td>
<td>↔ STDY</td>
</tr>
<tr>
<td>REDCO</td>
<td>↔ T5rrss</td>
<td>↔ COMB</td>
</tr>
</tbody>
</table>

VARIABLES: All variables, their type (REAL or INTEGER) and length (only if the variable is used as an array), together with a brief description, are listed.
The flow chart and a FORTRAN-IV compiler listing of the program module complete each description.

The source codes of programs TXCO0, TXCO1, TXCO2 and TXCO3 are available in the disc files &TXCO0, &TXCO1, &TXCO2 and &TXCO3. Since TXCO1, TXCO2 and TXCO3 use common sub-routines and functions, the latter are grouped together in file &TXCOU, where the "U" indicates the following "utility" program modules:

- **ACQN**: Positions S/V and reads DVM.
- **CNTL**: Closes scanner channel.
- **CURVE**: Compute linear curve fit.
- **ICON**: Converts two-digit INTEGER to ASCII-string.
- **IPORT**: Interrogates S/V.
- **PICTR**: Use CRT to display the acquired data.
- **REWRF**: Data transfer disc ↔ array.
- **RPACE**: Triggers A/D through PACER.
- **SCANR**: Closes scanner channel and reads SVM, counter.
- **TIME**: Gets date and time ↔ ASCII-string.
- **WAIT**: Causes a defined time delay.

When loading %TXCO1, %TXCO2 or %TXCO3, the relocatable binary utility file %TXCOU must also be loaded in order to satisfy the externals. The modules of TXCOU are described in Section 7, but in less detail than the programs in Sections 4 through 6.
3. **PROGRAM TXCOS**

3.1. **FLOW CHART PROGRAM TXCOS:**

```
START

Define COMMON block TMP.

Define COMMON block CIBUF.

Define COMMON block CONTR.

DO ASSIGNMENT.
LI=LOG U(TESSIN)

Clear screen.
WRITE(1,16) START

IF(1  .NE. 1) GOTO 2

Read program control array from disc file into CMR.
CALL SBUMP(-1,1)

1

2
```
1.
Redefine LI
LI = CNTFL(19)

2.
Get date & time.
CALL TIME

3.
Display menu on terminal, indicated by LI.
FORMAT # 101

4.
Read select code ISLCO;
Write on screen.
READ (LI, NE) ISLCO

5.
IF ISLCO < 1 OR ISLCO > NE

6.
Write select code in CNTFL(NE) prior to scheduling a Sun program.
CNTFL(NE) = ISLCO
Schedule some program, whose name is contained in PR (S)
CALL EXEC

Error Return from the EXEC CALL GO TO 17

O.K. Return from the EXEC call is GO TO 12

Read contents of A and B Register CALL AREG2D10

Display error message and show contents of A and B registers at the time of failure

8

16
Z

Ask for verification from Read and Compare Data.

Write program control away
CTRL back into disc file
CALL REWRAP(1,2)

STOP 0001
A - B - probe system survey

Z

Ask for verification read and compare figure.

With name of scan program (PC65) into integer array PR(3) (PCHANGEO)

CALL CODE WRITE (MF, 10)

L
On line calibration KULITE
(type A and type B probe)

while name of sam
program (7201)
into integer array
PR(2) 

CALL CODE
KULITE(PR, 101)

M
Acquisition of high speed data in free run mode

While name of son program (TNCCH) into integer array (FOCH) (FORMAT 48).

CALL COSE WERE (PP, 422)
Acquisition of high speed data in paced run mode

Z

Ask for
Random number; verification, read
and compare then.

Start Run

Write name of son program (PR301)
into integer array PR30 (Program #403).

CALL CODE 1 WRITE (PR, 403)

O
Combination probe survey

Z

Wait for verification, check and compare T0ぬ1.

while name of son program (TXCOZ) into integer array PR (Q) (POSITIONAL)
CALL CODE MUREST (PR, 42)

P
Acquisition of steady state data

Z

F

Ask whether to conduct a short (1) or long (2) run. Based on component I0,U,M.

Yes

\( \text{Trans} < 0 \)

\( \text{Trans} > 0 \)

No

\[ \text{DELAY} = \text{J0,UM} \]

Write name of run program (xxxx2) into integer array P10(3) (remember 100)

CALL CODE

WRITE (PRI, 100)

Q
Check the documentation

2

Ask where to direct output of check routines. Read and compare IDUM.

Yes

IDUM = 1

No

Yes

IDUM = 6

No

Yes

IDUM = 18

No

10

CTRL(10) = IDUM

While name of scan program (TICOS) into integer array FNS (FORMAT #2)
CALL CODE WRITE (FNS, 112)
Change the program control
gress CNTRL on line

Z

H

Go for
verification. End
and compare
ITEM

ITEM = Z

Write name of son
program (72003)
into integer array
PRC1 (FORTRAN).

CALL CODE
WRITE (PA, 112)

S
Reduce high speed data from A-B probe system

\[ Z \]

Ask for verification. Read and compare IDmun.

If mun = EM mun.

Write name of sin program (READ) into integer array PRC (PRIORITIES).

CALL CODE
WRITE (MP, 15)

\[ T \]
Reduce combination probe data

J

2

Task for Verification Read & compare IDMN.

IDMN = IDMN + 2

Write name of son program /REDOCS into integer array PR(2) (Format 44).

CALL CODE WRITE (PR, 147)

U
Reduce steady state data

Z

K

Not for verification. Read & Compare Z

Znew - Z

While name of son program (RECON) into integer array

CALL CODE WRITE(MH, IH)

V

Page 28
3.2. PROGRAM LISTING, TXCOG (updated version, 20 September 1982)

PAGE 0001 FTN. 9:32 AM MON., 20 SEP., 1982

0001  FTN4,L
0002  BLOCK DATA
0003    * FMP / IDCB(144),ITILE(3),ISIZE(2),IBECU,ICR
0004    COMMON / FMP / IDCB,IFILE,ISIZE,IBECU,ICR
0005  INTEGER IDCB(144),IFILE(3),ISIZE(2)
0006  END

FTN4 COMPILER: HP92060-16092 REV. 2001 (791101)
** NO WARNINGS ** NO ERRORS **
BLOCK COMMON FMP      SIZE = 00151
BLOCK DATA
* / CIBUF / IFUF(1664)
COMMON / CIBUF / IBUF
INTEGER IBUF(1664)
END

FTN4 COMPILER: HP92060-16092 REV. 2001 (791101)
** NO WARNINGS ** NO ERRORS **
BLOCK COMMON CIBUF SIZE = 01664
0012 BLOCK DATA
0013 *, / CONTR / CNTRL(256)
0014 COMMON / CONTR / CNTRL
0015 INTEGER CNTRL(256)
0016 END

FIN4 COMPILER: HP92060-16092 REV. 2001 (791101)
** NO WARNINGS ** NO ERRORS **
BLOCK COMMON CONTR SIZE = 00256
PROGRAM TXCOO (3,99)

Data acquisition transsonic compressor.

COMMON / CONTR / CNTRL

INTEGER CNTRL(256),IDCB(144),PR(3)

DATA NOLF /0065378/
DATA NOICR /0000338,0404338/
DATA ICLR /0155248,0155158
DATA START /0155108,0155128,0155018/

C 1::'( )R M TXCOO X ,)
100 FORMAT (3A2"TXCOO : START")
101 FORMAT (I"lrmnsonic Comressor Investigation: Test run
103 FORMAT (12)
104 FORMAT (12)
105 FORMAT (12)
106 FORMAT (12)
107 FORMAT (12)
108 FORMAT (12)
109 FORMAT (12)
110 FORMAT (12)
111 FORMAT (12)
112 FORMAT (12)
113 FORMAT (12)
114 FORMAT (12)
115 FORMAT (12)
116 FORMAT (12)

102 FORMAT (12)
103 FORMAT (12)
104 FORMAT (12)
105 FORMAT (12)
106 FORMAT (12)
107 FORMAT (12)
108 FORMAT (12)
109 FORMAT (12)
110 FORMAT (12)
111 FORMAT (12)
112 FORMAT (12)
113 FORMAT (12)
114 FORMAT (12)
115 FORMAT (12)
116 FORMAT (12)

* Select desired program module! Enter select code

* "2A2"

* "2A2"

* "2A2"

* "2A2"

* "2A2"

* "2A2"

* "2A2"

* "2A2"

* "2A2"

* "2A2"

* "2A2"

* "2A2"

* "2A2"

* "2A2"

* "2A2"

* "2A2"

* "2A2"

* "2A2"

* "2A2"

* "2A2"

* "2A2"

* "2A2"

* "2A2"

* "2A2"

* "2A2"

* "2A2"

* "2A2"

* "2A2"

* "2A2"

* "2A2"

* "2A2"

* "2A2"

* "2A2"

* "2A2"

* "2A2"

* "2A2"

* "2A2"

* "2A2"

* "2A2"

* "2A2"

* "2A2"
117 FORMAT ("REDCO ")
118 FORMAT (" Reduce data from the combination probe? Verify!"2A2)
119 FORMAT (" REDST ")
120 FORMAT (" Reduce steady state data? Verify!"2A2)
121 FORMAT (" TXCO 0: SCHEDULE "2A2,A1", CNTRL(50) ="I2)
122 FORMAT (" TXCO 0: FAILED TO SCHEDULE "2A2,A1", A REGISTER IS"O7"
123 FORMAT (" STOP 0001 ? Verify!"2A2)
124 FORMAT (9X="20"A2)
125 FORMAT ((3A2))
126 C
127 FORMAT (9X="20"A2)
128 C
129 FORMAT (9X="20"A2)
130 C
131 FORMAT (9X="20"A2)
132 C
133 FORMAT (9X="20"A2)
134 C
135 FORMAT (9X="20"A2)
136 C
137 FORMAT (9X="20"A2)
138 C
139 FORMAT (9X="20"A2)
140 C
141 FORMAT (9X="20"A2)
142 C
143 FORMAT (9X="20"A2)
144 C
145 FORMAT (9X="20"A2)
146 C
147 FORMAT (9X="20"A2)
148 C
149 FORMAT (9X="20"A2)
150 C
151 FORMAT (9X="20"A2)
152 C
153 FORMAT (9X="20"A2)
154 C
155 FORMAT (9X="20"A2)
156 C
157 FORMAT (9X="20"A2)
158 C
159 FORMAT (9X="20"A2)
160 C
161 FORMAT (9X="20"A2)
162 C
163 FORMAT (9X="20"A2)
164 C
165 FORMAT (9X="20"A2)
166 C
167 FORMAT (9X="20"A2)
168 C
169 FORMAT (9X="20"A2)
170 C
171 FORMAT (9X="20"A2)
172 C
173 FORMAT (9X="20"A2)
174 C
175 FORMAT (9X="20"A2)
176 C
177 FORMAT (9X="20"A2)
178 C
179 FORMAT (9X="20"A2)
180 C
181 FORMAT (9X="20"A2)
182 C
183 FORMAT (9X="20"A2)
184 C
185 FORMAT (9X="20"A2)
186 C
187 FORMAT (9X="20"A2)
188 C
189 FORMAT (9X="20"A2)
190 C
191 FORMAT (9X="20"A2)
192 C
193 FORMAT (9X="20"A2)
194 C
195 FORMAT (9X="20"A2)
196 C
197 FORMAT (9X="20"A2)
198 C
199 FORMAT (9X="20"A2)
200 C
201 FORMAT (9X="20"A2)
0155 C
0156 C
0157 C
0158 C
0159 C
0160 C
0161 C
0162 WRITE (LI, 109) NOCR
0163 READ (LI, 149) IDUM
0164 WRITE (LI, 149) ICLR
0165 IF ( IDUM .NE. 2H ) GO TO 02
0166 WRITE (PR, 103)
0167 GO TO 15
0168
0169 C
0170 C
0171 C
0172 C
0173 C
0174 WRITE (LI, 109) NOCR
0175 READ (LI, 149) IDUM
0176 WRITE (LI, 149) ICLR
0177 IF ( IDUM .NE. 2H ) GO TO 02
0178 CALL CODE
0179 WRITE (PR, 100)
0180 GO TO 15
0181
0182 C
0183 C
0184 C
0185 C
0186 C
0187 C
0188 WRITE (LI, 110) NOCR
0189 READ (LI, 111) IDUM
0190 WRITE (LI, 149) ICLR
0191 IF ( IDUM .LT. 0 .OR. IDUM .GT. 1 ) GO TO 02
0192 CALL CODE
0193 WRITE (PR, 100)
0194 GO TO 15
0195
0196 C
0197 C
0198 C
0199 C
0200 C
0201 WRITE (LI, 113) NOCR
0202 READ (LI, 102) IDUM
0203 WRITE (LI, 149) ICLR
0204 IF ( IDUM .EQ. 1 ) GO TO 10
0205 IF ( IDUM .EQ. 6 ) GO TO 10
0206 IF ( IDUM .EQ. 18 ) GO TO 10
0207 GO TO 02
0208 CNTRL(SI) = IDUM
0209 CALL CODE
0210 WRITE (PR, 112)
0211 GO TO 15
0212
0213 C
0214 C
0215 C
0216 C
0217 C
0218 WRITE (LI, 114) NOCR
0219 READ (LI, 149) IDUM
0220 WRITE (LI, 149) ICLR
0221 IF ( IDUM .NE. 2H ) GO TO 02
0222 CALL CODE
0223 WRITE (PR, 112)

0155 C
0156 C
0157 C
0158 C
0159 C
0160 C
0161 C
0162 WRITE (LI, 109) NOCR
0163 READ (LI, 149) IDUM
0164 WRITE (LI, 149) ICLR
0165 IF ( IDUM .NE. 2H ) GO TO 02
0166 WRITE (PR, 103)
0167 GO TO 15
0168
0169 C
0170 C
0171 C
0172 C
0173 C
0174 WRITE (LI, 109) NOCR
0175 READ (LI, 149) IDUM
0176 WRITE (LI, 149) ICLR
0177 IF ( IDUM .NE. 2H ) GO TO 02
0178 CALL CODE
0179 WRITE (PR, 100)
0180 GO TO 15
0181
0182 C
0183 C
0184 C
0185 C
0186 C
0187 C
0188 WRITE (LI, 110) NOCR
0189 READ (LI, 111) IDUM
0190 WRITE (LI, 149) ICLR
0191 IF ( IDUM .LT. 0 .OR. IDUM .GT. 1 ) GO TO 02
0192 CALL CODE
0193 WRITE (PR, 100)
0194 GO TO 15
0195
0196 C
0197 C
0198 C
0199 C
0200 C
0201 WRITE (LI, 113) NOCR
0202 READ (LI, 102) IDUM
0203 WRITE (LI, 149) ICLR
0204 IF ( IDUM .EQ. 1 ) GO TO 10
0205 IF ( IDUM .EQ. 6 ) GO TO 10
0206 IF ( IDUM .EQ. 18 ) GO TO 10
0207 GO TO 02
0208 CNTRL(SI) = IDUM
0209 CALL CODE
0210 WRITE (PR, 112)
0211 GO TO 15
0212
0213 C
0214 C
0215 C
0216 C
0217 C
0218 WRITE (LI, 114) NOCR
0219 READ (LI, 149) IDUM
0220 WRITE (LI, 149) ICLR
0221 IF ( IDUM .NE. 2H ) GO TO 02
0222 CALL CODE
0223 WRITE (PR, 112)

0155 C
0156 C
0157 C
0158 C
0159 C
0160 C
0161 C
0162 WRITE (LI, 109) NOCR
0163 READ (LI, 149) IDUM
0164 WRITE (LI, 149) ICLR
0165 IF ( IDUM .NE. 2H ) GO TO 02
0166 WRITE (PR, 103)
0167 GO TO 15
0168
0169 C
0170 C
0171 C
0172 C
0173 C
0174 WRITE (LI, 109) NOCR
0175 READ (LI, 149) IDUM
0176 WRITE (LI, 149) ICLR
0177 IF ( IDUM .NE. 2H ) GO TO 02
0178 CALL CODE
0179 WRITE (PR, 100)
0180 GO TO 15
0181
0182 C
0183 C
0184 C
0185 C
0186 C
0187 C
0188 WRITE (LI, 110) NOCR
0189 READ (LI, 111) IDUM
0190 WRITE (LI, 149) ICLR
0191 IF ( IDUM .LT. 0 .OR. IDUM .GT. 1 ) GO TO 02
0192 CALL CODE
0193 WRITE (PR, 100)
0194 GO TO 15
0195
0196 C
0197 C
0198 C
0199 C
0200 C
0201 WRITE (LI, 113) NOCR
0202 READ (LI, 102) IDUM
0203 WRITE (LI, 149) ICLR
0204 IF ( IDUM .EQ. 1 ) GO TO 10
0205 IF ( IDUM .EQ. 6 ) GO TO 10
0206 IF ( IDUM .EQ. 18 ) GO TO 10
0207 GO TO 02
0208 CNTRL(SI) = IDUM
0209 CALL CODE
0210 WRITE (PR, 112)
0211 GO TO 15
0212
0213 C
0214 C
0215 C
0216 C
0217 C
0218 WRITE (LI, 114) NOCR
0219 READ (LI, 149) IDUM
0220 WRITE (LI, 149) ICLR
0221 IF ( IDUM .NE. 2H ) GO TO 02
0222 CALL CODE
0223 WRITE (PR, 112)

Acquisition of high speed data in paced run mode.
Combination probe survey.
Acquisition of steady state data.
Check the instrumentation.
Change the program control array CNTRL on line.
GO TO 15

12 WRITE (LI, 118) NOCA
  READ (LI, 149) IDUM
  WRITE (LI, 149) ICLR
  IF (IDUM .NE. 2H) GO TO 02
  CALL CODE
  WRITE (PR, 115)
  GO TO 15

Reduce high speed data from A-B-probe system.

13 WRITE (LI, 118) NOCA
  READ (LI, 149) IDUM
  WRITE (LI, 149) ICLR
  IF (IDUM .NE. 2H) GO TO 02
  CALL CODE
  WRITE (PR, 117)
  GO TO 15

Reduce combination probe data.

Reduce steady state data.

14 WRITE (LI, 121) NOCA
  READ (LI, 149) IDUM
  WRITE (LI, 149) ICLR
  IF (IDUM .NE. 2H) GO TO 02
  CALL CODE
  WRITE (PR, 119)
  GO TO 15

Schedule desired son program.

15 WRITE (LI, 121) PR, CNTRL(50)
  ICODE = 9
  CALL REWRF (1, 2)
  CALL EXEC (ICODE+100000B, PR, IDC9, IDCBS)
  GO TO 17

16 GO TO 10

17 CALL ABREGR (IA, IB)
  WRITE (LI, 149) (ICLR, I=1, 2)
  WRITE (LI, 122) PR, IA, IB, NOLF, PR
  GO TO 01

19 WRITE (LI, 123) NOCA
  READ (LI, 149) IDUM
  WRITE (LI, 149) ICLR
  IF (IDUM .NE. 2H) GO TO 02
  CALL REWRF (1, 2)
  WRITE (LI, 124) NOLF
  STOP 0001
  END
INTEGER FUNCTION ICON (I,N)

Converts integer numbers into ASCII string.

Author: Robert N. Geepfarth
Date: January 31, 1979

Because of the simplicity of the program the program description is included in this box.

I, N ... integer numbers to be added.
IC ... integer number to be converted into ASCII.
ICON ... 2-character ASCII string to be returned

* Converts integer to ASCII-string.

100 FORMAT (I2)
IC = I+N
IF ( IC .LT. 10 ) GO TO 01
CALL CODE
WRITE (ICON,100) IC
RETURN
01 ICON = IC+30060B
RETURN
END
SUBROUTINE REWRF (IREWR, IWHATA)

This subroutine reads (IREWR = +1) or writes (IREWR = -1) of an array specified by IWHATA.

Author: Hans M. Zebner

Date: February 08, 1990

Detailed program description is available in TXCO log; the comment statementsotch to the flow chart explanations.

::...

Transfer disc array.

COMMON / CONTR / CNTRL

COMMON / CA / A

COMMON / FMP / IDCBIFILEISIZEISECUICR

REAL A(256)

INTEGER IBLIF(1664)

INTEGER IDCB(144), IFILE(3), ISIZE(2)

INTEGER NOLF, NOCR, ICLR(3)

DATA NOLF /6, 06337B/

DATA NOCR /0000338, 040433B/

DATA ICLR /00SS24B, 01SSISB, 006S378/

FORMAT (REWRF : ARRAY IBUF (1664) DISC FILE IBUFF

*;O0:26")

FORMAT (REWRF : DISC FILE IBUFF:00:26 ARRAY IBUF(1664)*)

FORMAT (REWRF : ARRAY CNTRL(256) DISC FILE CNTRLF:00:26"

.....

CALL OPEN (IDCBIERR, -3A2, "I2", "I2", "I4"

*00:26)

CALL LOCF (IDCB, IERR IDUMKIDUMIDUM, ISIZE(1), I

DIJM,1I)UM ISIZE(2)) failed STOP 42X"

CALL READF (IDCBIERRIBUF,-3A2,-2,-"12") failed;

STOP 27X"

CALL WRITF (IDCBIERRIBUF, I3", "I2", "I2")

failed; STOP *26X"

CALL READF (IDCBIERRCNTL,"I3","12","I2") failed;

STOP 27X"

CALL WRITF (IDCBIERRCNTL,"I3","I2","I2")

failed; STOP 26X"

CALL READF (IDCBIERR,A, "I3", "I2", "I2") failed;

STOP 27X"

CALL WRITF (IDCBIERR,A, "I3", "I2", "I2")

failed; STOP 26X"

CALL READF (IDCBIERR,A, "I3", "I2", "I2")

failed; STOP 26X"

CALL CLOSE (IDCBIERR) failed; STOP 40X"

LI = LOGLU (I3ESSN)

ISECU = 0

ICR = 26

IF ( IWHATA .LT. 1 .OR. IWHATA .GT. 2 ) GO TO 40

GO TO (10, 20) IWHATA

Integer array IBUF being written back and forth.
0388  10 CALL CODE
0389   WRITE (IFILE,108)
0390  CALL OPEN ((IDCB,IERR,IFILE,IOPTN,ISECU,ICR,IDCBS))
0391     IF ( IERR .GE. 0 ) GO TO 11
0392     WRITE (LI,121) IFILE,IOPTN,ISECU,ICR,IDCBS
0393    STOP 1
0394  11 CALL LOCF ((IDCB,IERR,IMID,IMID,ISIZE(1),IMID,IMID,ISIZE(2)))
0395   IF ( IERR .GE. 0 ) GO TO 12
0396   WRITE (LI,122)
0397    STOP 2
0398  12 CALL RWD (IDCB,IERR)
0399   IF ( IERR .GE. 0 ) GO TO 13
0400   WRITE (LI,123)
0401    STOP 3
0402  13 | SIZE(1) | = | SIZE(1)/2 |
0403     | SIZE(1) = | SIZE(1)*SIZE(2) |
0404     | ( IREWR .EQ. -1 ) GO TO 14 |
0405     | ( IREWR .EQ. +1 ) GO TO 15 |
0406  14 CALL READF ((IDCB,IERR,IBUF,IL))
0407     IF ( IERR .GE. 0 ) WRITE (LI,102)
0408     WRITE (LI,124) IL,LEN,NUM
0409    STOP 4
0410  15 CALL WRTF (IDCB,IERR,IBUF,IL)
0411   IF ( IERR .GE. 0 ) WRITE (LI,101)
0412   WRITE (LI,125) IL,LEN,NUM
0413    STOP 5
0414  16 CALL CLOSE (IDCB,IERR,0)
0415   IF ( IERR .GE. 0 ) GO TO 17
0416   WRITE (LI,130)
0417    STOP 6
0418  17 RETURN
0419
0420 C....................................................
0421 C....................................................
0422 C....................................................
0423 C....................................................
0424 C....................................................
0425 C....................................................
0426 C....................................................
0427 C....................................................
0428 C....................................................
0429 C....................................................
0430 C....................................................
0431 C....................................................
0432 C....................................................
0433 C....................................................
0434 C....................................................
0435 C....................................................
0436 C....................................................
0437 C....................................................
0438 C....................................................
0439 C....................................................
0440 C....................................................
0441 C....................................................
0442 C....................................................
0443 C....................................................
0444 C....................................................
0445 C....................................................
0446 C....................................................
0447 C....................................................
0448 C....................................................
0449 C....................................................
0450 C....................................................
0451 C....................................................
0452 C....................................................
0453 C....................................................
0454 C....................................................
0455 C....................................................
0456 C....................................................

39
PAGE 0012  REWRF  9:32 AM MON., 20 SEP., 1982

0457  26 CALL CLOSE (IDCB,IERR,0)
0458  IF (IERR .GE. 0) GO TO 27
0460  STOP 16
0461  27 RETURN

0470

0475 C C  Real array A being written back and forth.

0477 C C

0480  30 CALL WRITE (IFILE,i10)
0482  IF (IERR .GE. 0) GO TO 31
0484  WRITE (LI,121) IFILE,ISECU,ICR,IDCB
0494  STOP 21
0496  31 CALL LDCF (IDCB,IERR,NUM,NUM,ISIZE(1),NUM,NUM,ISIZE(2))
0498  IF (IERR .GE. 0) GO TO 32
0499  WRITE (LI,122)
0500  32 CALL RWDNF (IDCB,IERR)
0502  IF (IERR .GE. 0) GO TO 33
0503  WRITE (LI,123)
0504  33 ISIZE(1) = ISIZE(1)/2
0506  IL = ISIZE(1)*ISIZE(2)
0508  IF (IREWR .EQ. -1) GO TO 34
0510  IF (IREWR .EQ. +1) GO TO 35
0512  34 CALL READF (IDCB,IERR,IL)
0514  IF (IERR .GE. 0) WRITE (LI,106)
0515  IF (IERR .GE. 0) GO TO 36
0516  WRITE (LI,125) IL,LEN,NUM
0522  STOP 22
0524  35 CALL WRITF (IDCB,IERR,AIL)
0526  IF (IERR .GE. 0) WRITE (LI,105)
0528  IF (IERR .GE. 0) GO TO 37
0530  WRITE (LI,129) IL,LEN,NUM
0536  STOP 25
0538  36 CALL CLOSE (IDCB,IERR,0)
0540  IF (IERR .GE. 0) GO TO 38
0541  WRITE (LI,130)
0543  37 RETURN
0545

0549 C C

0598  40 WRITE (LI,107) IWHATA
0600  IWHATA = -IWHATA
0601  RETURN
0602  END

FTN4 COMPILER: HP92060-16092 REV. 2001 (791101)

** NO WARNINGS ** NO ERRORS ** PROGRAM = 01146 COMMON = 00000

40
SUBROUTINE TIME (IMON, IDAY, IYEAR, IHOUR, IMIN, ISEC)

Get date and time and convert the variables to ASCII string.

COMMON / CONT / CNTRL

INTEGER ITIME(5)
INTEGER CNTRL(256)

901 FORMAT ("ERROR DETECTED IN PROGRAM TIME/
" "CALL EXEC (11,ITIME)/

IMON = 2H##
IDAY = 2H##
IYEAR = 2H##
IHOUR = 2H##
IMIN = 2H##
ISEC = 2H##
CALL EXEC (11+1000000B,ITIME)
GO TO 02
GO TO 03
CALL ABREG (IA,IB)
01 GO TO 03
02 CALL ABREG (IA,IB)
03 IMON = ICON(CNTRL(1),0)
IDAY = ICON(CNTRL(2),0)
IYEAR = ICON(CNTRL(3),0)
IHOUR = ICON(ITIME(4),0)
IMIN = ICON(ITIME(3),0)
ISEC = ICON(ITIME(2),0)
RETURN
04 WRITE (6, 901) IA,IB
RETURN
END

FTN4 COMPILER: HP92060-16092 REV. 2001 (791101)

** NO WARNINGS ** NO ERRORS ** PROGRAM = 00146 COMMON = 00000
<table>
<thead>
<tr>
<th>Symbol</th>
<th>Description</th>
<th>Notes</th>
</tr>
</thead>
<tbody>
<tr>
<td>FMP</td>
<td>26042-26270</td>
<td></td>
</tr>
<tr>
<td>CBUF</td>
<td>31471-32070</td>
<td></td>
</tr>
<tr>
<td>CTRL</td>
<td>31471-32070</td>
<td></td>
</tr>
<tr>
<td>TXCO0</td>
<td>32071-36015</td>
<td>Data acquisition transonic compressor.</td>
</tr>
<tr>
<td>ICON</td>
<td>36016-36061</td>
<td>Converts integer to ASCII-string.</td>
</tr>
<tr>
<td>RFWRF</td>
<td>36852-40253</td>
<td></td>
</tr>
<tr>
<td>TIME</td>
<td>40254-40475</td>
<td>Gets date and time ASCII-string.</td>
</tr>
<tr>
<td>LOGLU</td>
<td>40476-40553</td>
<td></td>
</tr>
<tr>
<td>READF</td>
<td>40554-41544</td>
<td></td>
</tr>
<tr>
<td>OPEN</td>
<td>41545-42107</td>
<td></td>
</tr>
<tr>
<td>CLOSE</td>
<td>42108-42324</td>
<td></td>
</tr>
<tr>
<td>Save</td>
<td>42325-42417</td>
<td></td>
</tr>
<tr>
<td>LOCF</td>
<td>42418-42724</td>
<td></td>
</tr>
<tr>
<td>AREC</td>
<td>42726-43053</td>
<td></td>
</tr>
<tr>
<td>NWDF</td>
<td>43056-43070</td>
<td></td>
</tr>
<tr>
<td>LURC</td>
<td>43071-43100</td>
<td></td>
</tr>
<tr>
<td>CLRO</td>
<td>43101-43124</td>
<td></td>
</tr>
<tr>
<td>FMTIO</td>
<td>43125-44756</td>
<td></td>
</tr>
<tr>
<td>IFITY</td>
<td>44757-45036</td>
<td></td>
</tr>
<tr>
<td>DADS</td>
<td>45037-45146</td>
<td></td>
</tr>
<tr>
<td>DMP</td>
<td>45147-45314</td>
<td></td>
</tr>
<tr>
<td>.compareTo</td>
<td>45315-45633</td>
<td></td>
</tr>
<tr>
<td>SESSN</td>
<td>45634-45772</td>
<td></td>
</tr>
<tr>
<td>R/W#</td>
<td>45774-45781</td>
<td></td>
</tr>
<tr>
<td>F. P.</td>
<td>45782-45785</td>
<td></td>
</tr>
<tr>
<td>LNC</td>
<td>45786-45834</td>
<td></td>
</tr>
<tr>
<td>PAUSE</td>
<td>45835-46132</td>
<td></td>
</tr>
<tr>
<td>SEKI</td>
<td>46133-46250</td>
<td></td>
</tr>
<tr>
<td>$OPEN</td>
<td>46251-46311</td>
<td></td>
</tr>
<tr>
<td>RWD#</td>
<td>46312-46466</td>
<td></td>
</tr>
<tr>
<td>DNM</td>
<td>46467-47034</td>
<td></td>
</tr>
<tr>
<td>DIN</td>
<td>47035-47157</td>
<td></td>
</tr>
<tr>
<td>PRO</td>
<td>47158-47165</td>
<td></td>
</tr>
<tr>
<td>RMT</td>
<td>47166-47177</td>
<td></td>
</tr>
<tr>
<td>FMT.E</td>
<td>47178-47200</td>
<td></td>
</tr>
<tr>
<td>O/M</td>
<td>47201-47204</td>
<td></td>
</tr>
<tr>
<td>$SEP</td>
<td>47205-47226</td>
<td></td>
</tr>
<tr>
<td>RMPAR</td>
<td>47227-53053</td>
<td></td>
</tr>
<tr>
<td>LTRU</td>
<td>53054-53225</td>
<td></td>
</tr>
<tr>
<td>HTRU</td>
<td>53226-53245</td>
<td></td>
</tr>
<tr>
<td>LBT</td>
<td>53246-53324</td>
<td></td>
</tr>
</tbody>
</table>

12 PAGES RELOCATED
4. PROGRAM TXCO1

4.1. DESCRIPTION

TXCO1 is a son program of the father program TXCO0, by which it is scheduled, if one of the following operations is desired:

1 - A - B - probe system survey
2 - On line calibration, KULITE type 'A' and 'B' probes
3 - Acquisition of high speed data in free run mode
4 - Acquisition of high speed data in paced run mode.

When scheduled by TXCO0, which suspends operation while the son program TXCO1 executes, the program TXCO1, reads the program control array from the disc, sets the HP interface bus and the measurement and control devices to remote control and programs the Digital Voltmeter (DVM), the scanners and the counter. CNTRL (50) is the actual decision variable to select and call the subroutine, which performs the desired operation. When this subroutine has terminated, the interface bus and the devices are released from remote control and the control array is written into a disc file, so that the next TXCO module can read it. The correct termination of each subroutine can be verified by checking the stop codes. Note, that each stop coding ending on 77 indicates correct execution of a subroutine.

<table>
<thead>
<tr>
<th>CNTRL (50)</th>
<th>Subroutine</th>
<th>STOP Code</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>ABSRV</td>
<td>TXCO1 : STOP 0177</td>
</tr>
<tr>
<td>2</td>
<td>CALIB</td>
<td>TXCO1 : STOP 0277</td>
</tr>
<tr>
<td>3</td>
<td>FREER</td>
<td>TXCO1 : STOP 0377</td>
</tr>
<tr>
<td>4</td>
<td>PACER</td>
<td>TXCO1 : STOP 0477</td>
</tr>
</tbody>
</table>
Any other STOP code indicates an error and utilizing a program list the operator can trace the problem. The first two digits of the STOP code are typical for the subroutines. An example: the program stops at STOP code 0304; the first two digits read 3 and this tells the operator that it was subroutine FREER which ran into trouble, because the ending two digits read 04, which is different from 77; a program list uncovers that the failure occurred while writing into a disc file using FMP (File Management Package) subroutine WRITF near line 1005. STOP codes are crucial for a complex program system in order to rapidly detect and salvage problems, even during a test run.

**EXTERNALS:** REWRF, ABRT, RMOTE, ABSRV, CALIB, FREER, PACER, CLEAR, LOCL

**COMMON BLOCKS:** CONTR, CIBUF, FMP

FORTRAN conventions for the HP21MX computer request COMMON blocks to be predefined in a BLOCK DATA subroutine prior to using a COMMON block in a program, subroutine or function.

<table>
<thead>
<tr>
<th>BLOCK DATA subroutine</th>
<th>arrays &amp; variables</th>
<th>length in words</th>
</tr>
</thead>
<tbody>
<tr>
<td>CONTR</td>
<td>CNTRL</td>
<td>400B = 256</td>
</tr>
<tr>
<td>CIBUF</td>
<td>IBUF</td>
<td>3200B = 1664</td>
</tr>
<tr>
<td>FMP</td>
<td>IDCB, IFILE, ISIZE, ISECU, ICR</td>
<td>227B = 151</td>
</tr>
</tbody>
</table>

The COMMON block CONTR allocates the space for the control array CNTRL. A key to decode the individual elements of CNTRL can be found in the Appendix. COMMON block CNTRL is designed to take the largest raw data array - IBUF (1664) in subroutine
FREER - even if other subroutines only partially use the space, allocated by the block CIBUF. The arrays and variables allocated by the COMMON block FMP are frequently used for the data transfer from and to the disc. Since each individual subroutine saves the data prior to terminating, more than one subroutine or function may use the same buffer area.

MNEMONIC ABBREVIATIONS: None

ERROR MESSAGES: If CNTRL (50) is less than 1 or greater than 4, no subroutine can be selected and the program terminates, outputting an error message (FORMAT 102) to the terminal.

PROCEDURE: For more detailed information study the flow chart and the information given in the section PURPOSE.

DATA FILE: None

VARIABLES IN BLOCK DATA CONTR:
  CNTRL (256) integer program control array.

VARIABLES IN BLOCK DATA CIBUF:
  IBUF (1664) integer buffer array for the raw data.

VARIABLES IN BLOCK DATA FMP:
  IDCB (144) integer data control block.
  IFILE (3) integer array to contain file name.
  ISIZE (2) integer array to contain # of records in the first and record length in 16-bit-words in the second word.
**VARIABLES IN PROGRAM TXCO1:**

<table>
<thead>
<tr>
<th>Variable</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>ISECU</td>
<td>integer</td>
<td>security code of data file.</td>
</tr>
<tr>
<td>ICR</td>
<td>integer</td>
<td>cartridge reference number, where data file is located.</td>
</tr>
<tr>
<td>CNTRL (256)</td>
<td>integer</td>
<td>program control array.</td>
</tr>
<tr>
<td>NOLF</td>
<td>integer</td>
<td>suppresses line feed.</td>
</tr>
<tr>
<td>LI</td>
<td>integer</td>
<td>LU3 of standard input device (terminal).</td>
</tr>
<tr>
<td>STOP</td>
<td>integer</td>
<td>control variable to select STOP code.</td>
</tr>
<tr>
<td>X1</td>
<td>real</td>
<td></td>
</tr>
<tr>
<td>X2</td>
<td>real</td>
<td></td>
</tr>
<tr>
<td>X3</td>
<td>real</td>
<td></td>
</tr>
</tbody>
</table>

dummy variables.
FLOW CHART PROGRAM TYCO:

START

Define COMMON block FMP.

Define COMMON block CIBUF.

Define COMMON block CONTR.

Read program control array from disc file into CTRL.

CALL REWRF

I/O Assignment.
LI-CTRL (18)
Set interface bus & devices to remote control

CALL ABRTY(2)
CALL RLLRTE(8)
CALL RLRT(10)
CALL RLRT(12)
CALL RLRT(15)

Preprogram devices.
WRITE (8, 901)
WRITE (10, 1001)
WRITE (12, 201)
WRITE (15, 501)

Define start code.
ISTOP=CNTRL(90)

---
A) LU Assignments:
8. Scanner #1
10. Digital Voltmeter DVM
12. Counter
15. Scanner #2
7. HP-IB
CALL A

CALL ABSEY

CALL CALLS

CALL FEEDER

CALL BACER

Release interface
bus and devices
from remote
control.

CALL CLEAR (? , 1)
CALL LOC ( ? )

Write program
control array
back into disc
file CHRL.

CALL REINF

STOP INT

STOP G77

STOP G77

STOP G77

STOP G77

Display error message
END
START

Read control array in subroutine REWRT from disc

I/O assignment
LI=CNTRL(19)

IF CTRL(20)<4 OR CTRL(20)>4

Set interface bus and devices to remote control

ISTOP=CNTRL(38)

CALL ABSRV

YES

CONTROL(6) - 1

1

2

NO
1

CALL CALLB

CNTRL(39) = 3

NO

CALL FREE

CNTRL(30) = 3

NO

CALL PACE

CNTRL(30) = 4

NO

Release interface
bus and devices
from remote control.

Write control array
in subroutine RETURN
back to drive file.

STOP 0177

STOP = 1

3

2
STOP 0277

IF R = 2

STOP 03020

IF STOP = 3

STOP 0670

IF STOP = 4

Alert, param in subroutine has not been initialized

END
4.2. **SUBROUTINE ABSRV:**

**PURPOSE:** Acquisition of high speed data from the 1-stage axial compressor using miniaturized probes equipped with KULITE semiconductor pressure transducers.

**ARGUMENTS:** None

**EXTERNALS:** CALIB, TIME, REAT, PURGE, OPEN, WRITE, POSNT, CLOSE, SCANR, PACER.

**COMMON BLOCKS:** CIBUF, CONTR, FMP. For detailed explanation refer to the TXCO1 description.

**MNEMONIC ABBREVIATIONS:**

RE ... Repeat data acquisition of this yaw position.

NE ... Proceed to next yaw position.

EN ... End data acquisition at various yaw positions.

UP ... Update position readings of probes prior to data taking.

TA ... Initialisation command to take data.

PU ... Allow purge of an existing data file.

**ERROR MESSAGES:** If the number of yaw positions exceeds the previously defined number, the program terminates the subroutine correctly (in order to save the already acquired data) and displays an error message (FORMAT 118). The total # of possible yaw positions is input prior to creating the raw data file, so that latter can be created at the desired length.
PROCEDURE: For more detailed information, study the flow chart. After having read the accounting data, assigned the I/O references and preset the raw data array, ABSRV asks the operator, whether the 'A'- 'B'- probe system has been calibrated on line. If the answer is NO, ABSRV calls the subroutine CALIB, which controls the calibration. Then the calibration results are entered and the operator is asked to input the number of different yaw positions. Based on this information a raw data file of the appropriate length will be created and positioned. If the file with the automatically determined name already exists, the operator either allows overwriting the existing file (Input: PU) or renames the current data file (Input: any alphabetic character other than T). Prior to taking data the position of the probes is scanned and displayed. This control loop can be repeated by keying UP. Inputting TA initializes the data acquisition by subroutine PACER. Upon completion of the scan the operator can repeat this scan (Input: RE), proceed to the next point (Input: NE) with a different yaw position of both 'A' and 'B' probe. If the operator accidentally has decided to proceed to a probe position beyond the previously specified number, ABSRV displays an error message and terminates the subroutine correctly, i.e. saves the data in file, closes the file and writes the accounting data back into the control array.

DATA FILE: For more detailed information, study the following flow chart. The default file name is Tlrss (rr ... ASCII converted run #, ss ... ASCII converted sequential #).
**VARIABLES:**

<table>
<thead>
<tr>
<th>Variable</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>IBUF</td>
<td>integer</td>
<td>buffer array</td>
</tr>
<tr>
<td>CNTRL</td>
<td>integer</td>
<td>control array</td>
</tr>
<tr>
<td>IDCB</td>
<td>integer</td>
<td>data control block, used for FMP calls</td>
</tr>
<tr>
<td>IFILE</td>
<td>integer</td>
<td>array to contain current file name calls</td>
</tr>
<tr>
<td>ISIZE</td>
<td>integer</td>
<td>specifies # of records and record length</td>
</tr>
<tr>
<td>ISECU</td>
<td>integer</td>
<td>security code of data file</td>
</tr>
<tr>
<td>JSECU</td>
<td>integer</td>
<td>ASCII-converted security code</td>
</tr>
<tr>
<td>ICR</td>
<td>integer</td>
<td>cartridge reference number, when data file is located</td>
</tr>
<tr>
<td>JCR</td>
<td>integer</td>
<td>ASCII converted cartridge reference number</td>
</tr>
<tr>
<td>POS</td>
<td>real</td>
<td>array to contain probe positions</td>
</tr>
<tr>
<td>RBUF</td>
<td>real</td>
<td>data array, set equivalent to IBUF</td>
</tr>
<tr>
<td>NOLE</td>
<td>integer</td>
<td>suppresses line feed</td>
</tr>
<tr>
<td>NOCR</td>
<td>integer</td>
<td>suppresses line feed and carriage return</td>
</tr>
<tr>
<td>ICLR</td>
<td>integer</td>
<td>clear line above cursor</td>
</tr>
<tr>
<td>IDCBS</td>
<td>integer</td>
<td>length of data control block IDCB</td>
</tr>
<tr>
<td>IPAGE</td>
<td>integer</td>
<td>count of current page</td>
</tr>
<tr>
<td>IDOC</td>
<td>integer</td>
<td>count of current program run</td>
</tr>
<tr>
<td>IDOCF</td>
<td>integer</td>
<td>count of current data file</td>
</tr>
<tr>
<td>IL</td>
<td>integer</td>
<td>number of words to be transferred in FMP calls</td>
</tr>
</tbody>
</table>

55
<table>
<thead>
<tr>
<th>Variable</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>ITYPE</td>
<td>integer</td>
<td>type of data file</td>
</tr>
<tr>
<td>IFRST</td>
<td>integer</td>
<td>standard for the first two characters of file name</td>
</tr>
<tr>
<td>LI</td>
<td>integer</td>
<td>LU3 of standard input device (terminal)</td>
</tr>
<tr>
<td>LO</td>
<td>integer</td>
<td>LU# of standard output device (line position)</td>
</tr>
<tr>
<td>LS1</td>
<td>integer</td>
<td>LU# of scanner #1</td>
</tr>
<tr>
<td>LS2</td>
<td>integer</td>
<td>LU# of scanner #2</td>
</tr>
<tr>
<td>ICAL</td>
<td>integer</td>
<td>decision parameter</td>
</tr>
<tr>
<td>IDUM</td>
<td>integer</td>
<td>decision variable</td>
</tr>
<tr>
<td>SLOPEA</td>
<td>real</td>
<td>slope of linear curve fit for A probe calibration</td>
</tr>
<tr>
<td>SECONA</td>
<td>real</td>
<td>intercept of linear curve fit for A probe calibration</td>
</tr>
<tr>
<td>SLOPEB</td>
<td>real</td>
<td>slope of linear curve fit for B probe calibration</td>
</tr>
<tr>
<td>SECONB</td>
<td>real</td>
<td>intercept of linear curve fit for B probe calibration</td>
</tr>
<tr>
<td>AVRGEA</td>
<td>real</td>
<td>average voltage A probe, when aligned to flow</td>
</tr>
<tr>
<td>AVRGB</td>
<td>real</td>
<td>average voltage B probe, when aligned to flow</td>
</tr>
<tr>
<td>PBARO</td>
<td>real</td>
<td>barometric pressure</td>
</tr>
<tr>
<td>NPOS</td>
<td>integer</td>
<td>number of different yaw positions 'A'-'B' survey</td>
</tr>
<tr>
<td>IERR</td>
<td>integer</td>
<td>error flag (FMP package)</td>
</tr>
<tr>
<td>Variable</td>
<td>Type</td>
<td>Description</td>
</tr>
<tr>
<td>----------</td>
<td>--------</td>
<td>--------------------------------------------------</td>
</tr>
<tr>
<td>IMON</td>
<td>integer</td>
<td>ASCII converted month of the year</td>
</tr>
<tr>
<td>IDAY</td>
<td>integer</td>
<td>ASCII converted day of the month</td>
</tr>
<tr>
<td>IHOUR</td>
<td>integer</td>
<td>ASCII converted hour of the day (24 hr clock)</td>
</tr>
<tr>
<td>IMIN</td>
<td>integer</td>
<td>ASCII converted minute of the hour</td>
</tr>
<tr>
<td>IYEAR</td>
<td>integer</td>
<td>ASCII converted last two digits of current year</td>
</tr>
<tr>
<td>IFRST</td>
<td>integer</td>
<td>temporary buffer variable</td>
</tr>
<tr>
<td>NEW</td>
<td>integer</td>
<td>scratch variable for change of file name</td>
</tr>
<tr>
<td>IPOS</td>
<td>integer</td>
<td>current yaw position count</td>
</tr>
<tr>
<td>IREC</td>
<td>integer</td>
<td>record positioning variable</td>
</tr>
</tbody>
</table>
ENTRY

Accounting

I/O assignments preset data array

In the A, B, probe system calibrated?

YES

CALL CALIB
- calibrate A, B, probe system on line

NO

Enter calibration results; Print reading

/
Initialize data acquisition by entering a new position NOG. Create file of appropriate size.

IPOS = 0

Output error message.

Set CNTRL (23) and CNTRL (75) to 1 and position file.
Load, correct and output (on terminal)
Probe position and case angle

Acquire data in subroutine PACER
TDR = starting second

File and check IDUH 8

STOP 0111

STOP 0110
6

Write calibration results and other additional data in the directory of the data file; close file.

Terminate subroutine; write accounting data back into control array.

RETURN
4.3. **SUBROUTINE CALIB**

**PURPOSE:** Control the on-line calibration for the A-B- probe system. This includes data acquisition and storage as well as approximating the calibration results.

**ARGUMENTS:** None

**EXTERNALS:** TIME, FREER, PACER, CURVE

**COMMON BLOCKS:** CONTR. For detailed explanation refer to the TXCOl description.

**MNEMONIC ABBREVIATIONS:**

- **RE** ... Repeat this point
- **EN** ... End the on-line probe calibrations

**ERROR MESSAGES:** If no calibration is performed, the subroutine outputs a warning (FORMAT 108) and terminates; this can happen, if at the first decision to be made the operator inputs EN.
If less than two points with different reference pressures are taken, the subroutine outputs an error message (FORMAT log) and terminates.
Both messages, if studied carefully, tell the operator how to avoid mistakes.

**PROCEDURE:** For more detailed information, study the flow chart. After having read the accounting data and assigned the I/O references, CALIB asks the operator to input a
number (which, when the program was debugged, was the digital multimeter read-out displaying the analog voltage of either 'A' or 'B' probe). This input initializes the data acquisition at the first reference pressure. Then the program reminds the operator to switch the pacer to free run mode. The operator responds by pressing the return key and the program calls subroutine FREER. Average voltage from both 'A' and 'B' probe, together with the KULITE reference pressure are written into the arrays AVOLT, BVOLT and RPRES, respectively. The operator then decides whether to repeat the measurements at this reference pressure (Input : RE), end the calibration (Input : EN) or proceed to the next point (Input : any numerical value). If the calibration is to be terminated, the operator is reminded to switch the pacer to paced run mode and, with the reference pressure unchanged, a paced scan is taken from both 'A' and 'B' probe (using PACER). Then subroutine CURVE computes an average linear curve fit through the data points (AVOLT vs. RPRES and BVOLT vs. RPRES respectively). In both cases slope and intercept are printed. Note, that the intercept is meaningless, but required in subroutine CURVE, which uses a least squares algorithm. CALIB then terminates and writes the accounting data back into the control array.

DATA FILE: Handled by subroutines FREER (Section 4.4) and PACER (Section 4.5).
### VARIABLES:

<table>
<thead>
<tr>
<th>Variable</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>CNTRL(256)</td>
<td>integer</td>
<td>control array</td>
</tr>
<tr>
<td>AVOLT(10)</td>
<td>real</td>
<td>array to contain average voltages from A probe</td>
</tr>
<tr>
<td>BVOLT(10)</td>
<td>real</td>
<td>array to contain average voltages from B probe</td>
</tr>
<tr>
<td>RPRES(10)</td>
<td>real</td>
<td>array to contain reference pressures</td>
</tr>
<tr>
<td>DMM(10)</td>
<td>real</td>
<td>array to contain additional data (e.g.: DMM read outs)</td>
</tr>
<tr>
<td>NOLF</td>
<td>integer</td>
<td>suppresses line Feed</td>
</tr>
<tr>
<td>NOCR(2)</td>
<td>integer</td>
<td>suppresses line Feed and carriage return</td>
</tr>
<tr>
<td>ICLR(3)</td>
<td>integer</td>
<td>clears line above cursor</td>
</tr>
<tr>
<td>ITIME(5)</td>
<td>integer</td>
<td>array to contain ASCII converted date and time</td>
</tr>
<tr>
<td>IO(5)</td>
<td>integer</td>
<td>scratch array</td>
</tr>
<tr>
<td>IPAGE</td>
<td>integer</td>
<td>count of current page</td>
</tr>
<tr>
<td>LI</td>
<td>integer</td>
<td>LU# of standard input device (terminal)</td>
</tr>
<tr>
<td>LO</td>
<td>integer</td>
<td>LU# of standard output device (line printer)</td>
</tr>
<tr>
<td>IMON</td>
<td>integer</td>
<td>ASCII converted month of the year</td>
</tr>
<tr>
<td>IDAY</td>
<td>integer</td>
<td>ASCII converted day of the month</td>
</tr>
<tr>
<td>IYEAR</td>
<td>integer</td>
<td>ASCII converted last two digits</td>
</tr>
<tr>
<td>IHOUR</td>
<td>integer</td>
<td>ASCII converted hour of the day (24 hr clock)</td>
</tr>
<tr>
<td>Variable</td>
<td>Type</td>
<td>Description</td>
</tr>
<tr>
<td>----------</td>
<td>-------</td>
<td>-------------</td>
</tr>
<tr>
<td>IMIN</td>
<td>integer</td>
<td>ASCII converted minute of the hour</td>
</tr>
<tr>
<td>IPTS</td>
<td>integer</td>
<td>variable count total # of calibration positions</td>
</tr>
<tr>
<td>IPREV</td>
<td>integer</td>
<td>variable temporary stores contents of CNTRL (219)</td>
</tr>
<tr>
<td>IDUM</td>
<td>integer</td>
<td>scratch variable</td>
</tr>
<tr>
<td>DM</td>
<td>real</td>
<td>variable used to decode value from array IO</td>
</tr>
<tr>
<td>IREC</td>
<td>integer</td>
<td>starting record for paced run-data array</td>
</tr>
<tr>
<td>SLOPE</td>
<td>real</td>
<td>slope of linear curve fits</td>
</tr>
<tr>
<td>SECON</td>
<td>real</td>
<td>intercept of linear curve fit (as from CURVE)</td>
</tr>
</tbody>
</table>
FLOW CHART SUBROUTINE CALIB

ENTRY

Accounting

I/O Assignments, print heading.

Confin the on-line calibration to the two A/D channels with A: B: Signal.

Select next step: read and check I0.

STOP 0201

IO: any measured input

Decode DM from IO(5)

1

2
Calculate linear curve fit through A and B data points; done by subroutine CURVE

Terminate subroutine: unique accounting data back into control array. Reset CNTRL(85) to the previous value

RETURN

Error: only one data point is not sufficient to derive slope.

Error: no collision performed at all.

Error processing.
SUBROUTINE FREER:

PURPOSE: Control data acquisition from HP 5610 A/D converter, store data in file and documentation, perform calculation of average voltage.

ARGUMENTS: AVOLT, BVOLT, PREFR

AVOLT real average voltage from 'A'- probe, based on NRPT3 points
BVOLT real average voltage from 'B'- probe, based on NRPT3 points
PREFR real reference pressure for KULITE transducers (raw data format)

EXTERNALS: TIME, ICON, SCANR, EXEC, ABREG, CREAT, OPEN, PURGE, WRITF, CLOSE

COMMON BLOCKS: CIBUF, CONTR, FMP. For detailed description refer to the TXCO1 description

MNEMONIC ABBREVIATIONS:

PU ... Allow purge of an existing data file

ERROR MESSAGES: If the EXEC call to read the voltages from the A/D converter is not executed correctly, an error return occurs as follows:

70
CALL EXEC (1 + 100000B, 20, IBUF(1), NRPT2, ICHNL, 4)
GO TO 11
GO TO 12
10 CALL ABREG (IA, IB)  look, what's in the registers.
GO TO 21
:
:
:
:
Error returns ....

21 WRITE (6, 901) NRPT2, ICHNL output error message,
WRITE (6, 902) IA, IB then terminate subroutine
GO TO 20

PROCEDURE: For more detailed information, study the flow chart.
The subroutine reads the accounting data from the control array
and defines FMP parameters (FMP : File Management Package,
manipulates disc Files). Next the I/O references are assigned
and all words of the raw data array are preset to be 177777B.
If CNTRL (37) is set to 1, the heading for the Free Run document-
ation page is printed. If CNTRL (38) is set to one, a key to
the printout is printed. Then the data acquisition loop starts
and executes NRPT1 times (NRPT1 = CNTRL (230) = number of KULITE
signals to be acquired; maximum is 16). Should the sequential
number for the data file name become greater than 99, the first
two characters of the file name are changed from T2 (default)
to S2 and the count is reset to zero. Additional data is
acquired and the probe positions are read and written into the
variable IOXM. Prior to the data acquisition all unused
elements of the data array are set to zero. Utilizing the EXEC
call NRP2 measurements are performed and the A/D digital
output is written into array IBUF, starting at address of word IBUF (1). ICHNL specifies the selected A/D analog input channel. The 4 in the parameter list causes the A/D converter to dump data into the CPU as fast as possible via DMA (Direct Memory Access). If an error occurs, its reason is enquired (see preceding segment ERROR MESSAGES). To calculate the average voltage, all words of IBUF must be anded with IMASK, because bits 0 through 5 are used to control the data transfer.

\[
\begin{align*}
\text{bit} & \quad 15 & 12 & 9 & 6 & 3 & 0 \\
\text{IBUF(J2)} & \quad 1111001011100010 \quad = \quad 171342B \\
\text{IMASK} & \quad 1111111111000000 \quad = \quad 177700B \quad = \quad -64 \\
\text{IBUF(J2)} & \quad \text{IAND} \ (\text{IBUF(J2)}, \ \text{IMASK}) \\
\text{IBUF(J2)} & \quad 1111001011000000 \quad = \quad 171300B \quad = \quad -3392 \\
\end{align*}
\]
To derive the voltage, IBUF(J2) must first be divided by the maximum value which can be transferred by a 16-bit word when the bits 0 through 5 do not contain data; this word is

\[
011111111111000000 \quad = \quad 077700B \quad = \quad 32704
\]
This bit configuration corresponds to the full scale voltage (FSVLT) of \(+1\) Volt. When no bit is set, the voltage is 0 (ensured by calibration). Thus if the integer, IBUF(J2) is divided by 32 704 and multiplied by unity the voltage is obtained. Since the bits 0 through 5 are not used for data, the maximum voltage resolution of the A/D converter is

\[
R = \frac{100B}{77700B} \times \frac{64}{32704} \times \frac{1}{511} \times \text{FSVLT} \\
R \quad = \quad .001956947 \text{ Volt, if FSVLT} = +1.0 \text{ V}
\]

72
The voltage associated with the bit configuration

\[ \text{1 1 1 1 0 0 1 0 1 1 0 0 0 0 0} = 171300B = -3392 \]

is \(-0.103718 \text{ Volt} = -\frac{3392}{32704} \times 1.0 \text{ Volt}\). In the Subroutine, however, the division through 32704 and the multiplication with FSVLT is executed after all the voltages from NRPT2 points are added in order to compute the average voltage. The average voltage then is written into the variable AVOLT or BVOLT, depending on which probe has been selected. The data then are saved in a file. If a file with the automatically determined name already exists, the operator either allows overwriting the existing file (Input : PU) or renames the current data file (Input : any alphabetic character other than T). This is the only interactive manipulation in the subroutine. The data acquisition loop terminates, printing the most important data. Accounting data are written back into the control array and the subroutine returns control to the calling program.

**DATA FILE:** The data file consist of 13 records with a length of 128 words each, so that 1664 = (128 * 13) words can be stored. The default file name is T2rrss (rr ... ASCII converted run #, ss ... ASCII converted sequential #).

**VARIABLES:**

- **IBUF** (1664) integer buffer array for raw data
- **CNTRL** (256) integer control array
- **IDCB** (144) integer data control block, used for FMB calls
IFILE (3) integer array to contain current file name
ISIZE (2) integer specifies # of records and record length
ISECU integer security code of data file
ICR integer cartridge reference number, where data file is located
NOLF integer suppresses line feed
NOCR (2) integer suppresses line feed and carriage return
ICLR (3) integer clear line above cursor
IOXIM (9) integer array, where 'A' and 'B' probe positions are written into in ASCII code
FSVLT real full scale voltage of A/D converter
IDCBS integer length of data control block IDCB
IPAGE integer counts of current page
IDOC integer counts, how often this subroutine is called
IDOCF integer count of current data file sequential #
ITYPE integer type of data file
IFRST integer standard for the first two characters of file name
ISP integer decision variable, used to space the output
<table>
<thead>
<tr>
<th>Variable</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>IL</td>
<td>integer</td>
<td>number of words to be transferred in FMP calls</td>
</tr>
<tr>
<td>LI</td>
<td>integer</td>
<td>LU# of standard input device (terminal)</td>
</tr>
<tr>
<td>LO</td>
<td>integer</td>
<td>LU# of standard output device (line printer)</td>
</tr>
<tr>
<td>LS1</td>
<td>integer</td>
<td>LU# of scanner #1</td>
</tr>
<tr>
<td>LS2</td>
<td>integer</td>
<td>LU# of scanner #2</td>
</tr>
<tr>
<td>ISV1</td>
<td>integer</td>
<td>code # of S/V controller #1</td>
</tr>
<tr>
<td>ISV4</td>
<td>integer</td>
<td>code # of S/V controller #2</td>
</tr>
<tr>
<td>NRPT1</td>
<td>integer</td>
<td># of various KULITE signals to be acquired</td>
</tr>
<tr>
<td>NRPT2</td>
<td>integer</td>
<td>total # of point, taken from each KULITE signal</td>
</tr>
<tr>
<td>NRPT3</td>
<td>integer</td>
<td>= NRPT2+ : DO loop start address</td>
</tr>
<tr>
<td>IMASK</td>
<td>integer</td>
<td>masking variable</td>
</tr>
<tr>
<td>IW</td>
<td>integer</td>
<td>controls time delay between closing S/V port and reading voltage</td>
</tr>
<tr>
<td>IMON</td>
<td>integer</td>
<td>ASCII converted month of the year</td>
</tr>
<tr>
<td>IDAY</td>
<td>integer</td>
<td>ASCII converted day of the month</td>
</tr>
<tr>
<td>IHOUR</td>
<td>integer</td>
<td>ASCII converted hour of the day (24 hr clock)</td>
</tr>
<tr>
<td>IMIN</td>
<td>integer</td>
<td>ASCII converted minute of the hour</td>
</tr>
<tr>
<td>IYEAR</td>
<td>integer</td>
<td>ASCII converted last two digits of current year</td>
</tr>
<tr>
<td>ICHNL</td>
<td>integer</td>
<td>A/D input channel to be selected</td>
</tr>
<tr>
<td>FREQ</td>
<td>real</td>
<td>RPM of the transonic compressor</td>
</tr>
</tbody>
</table>
CIM  real  immersion of the combination probe
CYAW  real  yaw angle of the combination probe
PREF  real  KULITE reference pressure
PREFR  real  KULITE reference pressure as returned to the calling routine (either CALIB, ABSRV or TXCO1)
P1  real  
P23  real  pressures P1, P23 & P4 from calibration probe
P4  real  
E  real  Temperature reading from sensor ahead of rotor (in mV)
DE  real  Differential temperature reading from station ahead of rotor across rotor
XIM  real  Immersion of the KULITE probe
YAW  real  Yaw angle of the KULITE probe
IA  integer  Variable to contain contents of A register
IB  integer  Variable to contain contents of B register
AVRGE  real  KULITE output average voltage after amplification and A/D conversion
AVOLT  real  'A' probe output average voltage
BVOLT  real  'B' probe output average voltage
ISP  integer  control variable to space output
IDUM  integer  decision variable
<table>
<thead>
<tr>
<th>Variable</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>IERR</td>
<td>integer</td>
<td>error flag returned from FMP calls</td>
</tr>
<tr>
<td>JSECU</td>
<td>integer</td>
<td>ASCII converted security code</td>
</tr>
<tr>
<td>JCR</td>
<td>integer</td>
<td>ASCII converted cartridge reference number</td>
</tr>
<tr>
<td>NEW</td>
<td>integer</td>
<td>variable to contain changed first two characters of raw data file name</td>
</tr>
</tbody>
</table>
FLOW CHART: SUBROUTINE FREER

ENTRY

Accounting

I/O Assignments; print data array

CTRL(37) = 1

Print heading

CTRL(39) = 1

Print legend for tabulated output

/
DO 10 J1 = 1, NRPFI, 1

Define A/D channel
ICMP = CMPAR(230+2)

Define sequential number of read-out IDOC
and data file-IDOCF
IDOC = IDOC + 1
IDOCF = IDOCF + 1

IF IDOCF < 100 THEN

Channel IFRST from 78 (standard) to 52, Subtract 150 from IDOCF
IFRST* = 2*IFRST
IDOCF = IDOCF - 100

ELSE

END

2

3
Define data file name
IDB(1), IDST
IDB(1) = IDST

Acquire additional data required for reduction procedure

ICHNL = 0

YES

ICHNL = 1

NO

Get 'A' probe position and write values into IDXIM

NO

WALL KNOTS
Indicate this in variable IDXIM

YES

Get 'B' probe position and write values into IDXIM

3

4
1. Proceed unused elements of data array IBUF

2. QRTZ = 1664

3. NO

4. Write 0 into all the elements, that later will not contain a value.

5. Read voltages from A/D converter into data array IOUT; starting element for direct memory access is IOUT(1); A/D is operated in free run mode.

6. CALL EXEC
Inquire contents of A and B registers for error processing.

Correct return from EXEC call.

Calculate average voltage based on 10000 points:
1. Step: Mask the raw data
2. Step: Convert from two's complement integer to real
3. Step: refer to A/D range
   000000B = 0  mV
   777777B = 1111V

Write average voltages into variables A/DOUT & B/DOUT.

Error return from EXEC call.

Write error message & contents of A and B register.
Stop data acquisition loop.
List data.

Do loop completed? NO

Print bottom of data documentation page.

20

Terminate subroutine; write accounting data back into control array.

RETURN
4.5. **SUBROUTINE PACER:**

**PURPOSE:** Control data acquisition from HP 5640 A/D converter if this device is triggered by the pacer, store data in file and document all steps.

**ARGUMENTS:** IREC

IREC integer starting record #, where raw KULITE and additional data are written

**EXTERNALS:** TIME, ICON, SCANR, ACQN, RSPACE, PICTR, CREAT, OPEN, PURGE, WRITF, CLOSE

**COMMON BLOCKS:** CIBUF, CONTR, FMP. For detailed description refer to the TXCO1 description.

**MNEMONIC ABBREVIATIONS:**

PU ... Allow purge of an existing data file

RE ... Repeat data acquisition

**ERROR MESSAGES:** None

**PROCEDURE:** For more detailed information, study the flow chart! The subroutine reads the accounting data from the control array and defines FMP parameters (FMP: file management package, manipulates disc files; refer to the HP manuals for more information). Next the I/O references are assigned and the words of the data array are present. If CNTRL (37) is set to 1, the heading for the paced run documentation page is printed. If CNTRL (38) is set to 1, a key to the print
out is printed. Then the data acquisition loop starts and executes NRPT1 times (NRPT1 = CNTRL(230) = number of KULITE signals to be acquired; maximum is 16). If CNTRL(39) is set to 1 (i.e.: subroutine PACER is called from subroutine ABSRV, which takes care of creating/opening, positioning and closing files), the accounting of the data file names is skipped. If otherwise, i.e. CNTRL(39) is not equal to 1, the sequential number for the data file name exceeds 99, the first two characters of the file name are changed from T3 (default) to S3 and the count is reset to zero. Additional data is acquired and the probe positions are read and written into the variable IOXIM. Since the KULITE probes are mounted in physically different positions (the phase angle is 90°, i.e. 2½ times 40°, where 40° is double the rotor inter-blade angle), and the signals must be converted from the same point in the rotor blade wake, the IBLADE for the 'B' probe has to be increased by the appropriate amount, which is 576 (see sketch).
The operator is then informed that the system is ready for the next data scan. Depressing the RETURN key starts the data acquisition. Pacer mode (1 or 2), selected blade pair, increment to step through the 256 blade passage locations and the number of measurement repetitions at each location (i.e. at each IBLADE) are read from the control array. If the pacer is operated in mode 2 (i.e. selects a specific blade pair), the bit 15 is set by adding IADD = 100000B to the start and stop address. Refer to the RPACE description for details concerning how the data acquisition is performed. Not only the voltages, through subroutine RPACE, but also additional data are written into the raw data array. Some of the data are multiplied by 1,000,000 in order to be able to store all valid digits in integer constants and the average voltage AVRBE is set equivalent to the array IAVRGE(2) by an EQUIVALENCE statement.

Date and time are written into the raw data array also. If CNTRL(40) is set to 1, the wave as acquired is displayed on the terminal, which is selected by its logical unit number LA. Refer to the detailed description of subroutine PICTR for further information on how this is achieved; i.e. to use a non-graphics
terminal for plotting. The resolution of the terminal plot is very limited. The option to display the just-acquired periodic high speed signal is designed to give the operator an opportunity to immediately verify the correctness of the data acquisition. Connecting a lead from KULITE amplifier output to an oscilloscope gives the investigator the chance to check digitized data against original analog data. If an error is encountered, the data scan may be repeated (Input: ... RE). Depressing the RETURN key causes the subroutine to proceed to the next task, the storing of the data. File name, ASCII converted security code and ASCII converted cartridge reference number are written into the raw data array. The raw data file is either created/opened and closed by subroutine PACER (CNTRL (39) is not equal 1) or this subroutine is called from subroutine ABSRV, which already has created/opened and positioned the raw data file and will close it (CNTRL (39) is set to 1). If, in the first mode, the automatically determined file name already exists, the operator either allows overwriting the existing file (Input: PU) or renames the current data file name (Input: any alphabetic character other than T). The starting record number is also written in the data array. If CNTRL (39) is not equal 1, the raw data file is closed and the data acquisition loop stops printing all the additional data on the documentation page. The accounting data are written back into the control array and the subroutine terminates.
**DATA FILE:** For more detailed information, study the key to the raw data file following this description. The default file name is T3rrss (rr ... ASCII converted run #, ss ... ASCII converted sequential #).

**VARIABLES:**

- **IBUF (1664)** integer buffer array
- **CNTRL (256)** integer control array
- **IDCB (144)** integer data control block; used for FMP calls
- **IFILE (3)** integer array to contain current file name
- **ISIZE (2)** integer specifies # of records and record length in words
- **ISECU** integer security code of data file
- **JSECU** integer ASCII converted security code
- **ICR** integer cartridge reference number, where data file is located
- **JCR** integer ASCII converted cartridge reference number
- **NOLF** integer suppresses line feed on terminal
- **NOCR (2)** integer suppresses line feed and carriage return on terminal
- **ICLR (3)** integer clear line above cursor
- **IBUFL (384)** integer raw data array, set equivalent to IBUF
- **IOXIM (9)** integer array to contain probe positions in ASCII code

88
IAVRGE (2)  integer  array to contain average voltage, set equivalent to AVRGE
IDCBS  integer  length of data control block IDCB in words (here: 144)
IPAGE  integer  count of current page
IDOC  integer  count of current program run
IDOCF  integer  count of current data file sequential #
IL  integer  number of words to be transferred in FMP calls
ITYPE  integer  type of data file (here: 1)
IFIRST  integer  standard for the first two characters of data file name
ISP  integer  control variable, used to space the output
LI  integer  LU# of standard input device (system console)
LO  integer  LU# of standard output device (line printer)
LA  integer  LU# of auxiliary output device (auxiliary terminal)
LS1  integer  LU# of scanner 1
LS2  integer  LU# of scanner 2
ISV1  integer  number of S/V controller 1
ISV4  integer  number of S/V controller 2
NRPT1  integer  number of KULITE measurements ('A', 'B', case KULITES)
<table>
<thead>
<tr>
<th>Variable</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>IW</td>
<td>integer</td>
<td>time delay between closing S/V part and reading transducer voltage from DVM</td>
</tr>
<tr>
<td>IMON</td>
<td>integer</td>
<td>ASCII converted month of the year</td>
</tr>
<tr>
<td>IDAY</td>
<td>integer</td>
<td>ASCII converted day of the month</td>
</tr>
<tr>
<td>IYEAR</td>
<td>integer</td>
<td>ASCII converted last two digits of current year</td>
</tr>
<tr>
<td>FREQ</td>
<td>real</td>
<td>RPM of the compressor</td>
</tr>
<tr>
<td>CIM</td>
<td>real</td>
<td>immersion of the combination probe</td>
</tr>
<tr>
<td>CYAW</td>
<td>real</td>
<td>yaw angle of the combination probe</td>
</tr>
<tr>
<td>PREF</td>
<td>real</td>
<td>reference pressure for the KULITE probes</td>
</tr>
<tr>
<td>P1</td>
<td>real</td>
<td>pressure $P_1$ from the combination probe</td>
</tr>
<tr>
<td>P23</td>
<td>real</td>
<td>pressure $P_{23}$ from the combination probe</td>
</tr>
<tr>
<td>P4</td>
<td>real</td>
<td>pressure $P_4$ from the combination probe</td>
</tr>
<tr>
<td>E</td>
<td>real</td>
<td>thermocouple output, Station 'O'</td>
</tr>
<tr>
<td>DE</td>
<td>real</td>
<td>thermocouple differential output from 'O' across rotor</td>
</tr>
<tr>
<td>XIM</td>
<td>real</td>
<td>immersion of either 'A'- or 'B' probe</td>
</tr>
<tr>
<td>YAW</td>
<td>real</td>
<td>yaw angle of either 'A'- or 'B' probe</td>
</tr>
<tr>
<td>IADD</td>
<td>integer</td>
<td>variable to be added to start and stop address for paced run to 90</td>
</tr>
</tbody>
</table>
compensate phase angle between these probes

<table>
<thead>
<tr>
<th>Variable</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>IDUM</td>
<td>integer</td>
<td>decision variable</td>
</tr>
<tr>
<td>IPAMO</td>
<td>integer</td>
<td>pacer mode (1 or 2)</td>
</tr>
<tr>
<td>IPAIR</td>
<td>integer</td>
<td>selected blade pair</td>
</tr>
<tr>
<td>ISTART</td>
<td>integer</td>
<td>start address for paced run</td>
</tr>
<tr>
<td>ISTOP</td>
<td>integer</td>
<td>stop address for paced run</td>
</tr>
<tr>
<td>INCR</td>
<td>integer</td>
<td>increment for paced run</td>
</tr>
<tr>
<td>IRPT</td>
<td>integer</td>
<td>number of repetitions at each IBLADE</td>
</tr>
<tr>
<td>J111</td>
<td>integer</td>
<td>dummy variable</td>
</tr>
<tr>
<td>J222</td>
<td>integer</td>
<td>dummy variable</td>
</tr>
<tr>
<td>DUM</td>
<td>real</td>
<td>dummy variable</td>
</tr>
<tr>
<td>IERR</td>
<td>integer</td>
<td>error flag used by FMP calls</td>
</tr>
<tr>
<td>NEW</td>
<td>integer</td>
<td>scratch variable used to change file name</td>
</tr>
</tbody>
</table>
FLOW CHART SUBROUTINE PACER

ENTRY

\[ \text{CNTRL}(23) = 1 \]

\[ \text{CNTRL}(27) = 1 \]

Accounting

I/O Assignments; preset data array.

Print heading

/

YES

NO

YES

NO
1

CNTRL(38) = 1

NO

YES

Print legend for tabulated output

DO J = 1, NRPY, 1

Define A/I0 channel
IMNL = CNTRL(220 + J)

CNTRL(52) = 1

YES

Define sequential number of print on IDOC and data FILE IDOC
IDOC = IDOC + 1
IDOC = IDOC + 1

2

3

4

93
2

NO

IDOCF < 100

YES

Change IPRST
from T2 (standard)
to S2. Subtract
100 from IDOCF
IPRST = 2
IDOCF = IDOCF - 100

Define data file
name:
IPRST = IPRST
FILE(1) = IPRST
FILE(2) = IDOCF

Acquire additional
data required for
reduction procedure

3

4

5

6
Diagram shows a flowchart with steps:

1. Define phase mode
2. Check if phase mode is 2
3. If yes, continue
4. Compute phase angle between A and B
5. Print angle
6. Increment counter
7. Print counter
8. Continue

Legend:
- Define phase mode
- Check if phase mode is 2
- Compute phase angle
- Print angle
- Increment counter
- Print counter
- Continue
Acquire high speed data in fast run mode.

CALL FRACE

While additional data into data array

CTRL.(40) = 1

Display the just acquired data on auxiliary terminal CRT.

CALL PICTR
No - Code ZxJrcu

IRE < 0

CALL PURGE

Purge the file with the already existing file name

CALL CREAT

Create data file

STOP 0606

11

13

NO

SAVE data in file. Define security code ZxJrcu and cartridge reference number IER; write these data and file name into data array IER.

CTRL(13) #1

NO

CALL CREAT

Create data file

YES

IRE > 0

12

13

14

15

16

4
Terminate subroutine, write accounting data back into control array

RETURN
4.6. PROGRAM LISTING TXCO1

PAGE 0001 FTN. 2:47 PM MON., 25 AUG., 1980

0001 FTN4,L
0002 BLOCK DATA
0003 * / FMP / IDCBC(144),IFILE(3),ISIZE(2),ISECU,ICR
0004 COMMON / FMP / IDCBC,IFILE,ISIZE,ISECU,ICR
0005 INTEGER IDCBC(144),IFILE(3),ISIZE(2)
0006 END

FTN4 COMPILER: HP92060-16092 REV. 1926 (790430)
** NO WARNINGS ** NO ERRORS **
BLOCK COMMON FMP SIZE = 001S1
BLOCK DATA

COMMON CIBUF / BUF

INTEGER BUF(1664)

END

FTN4 COMPILER: HP92060-16092 REV. 1926 (790430)

** NO WARNINGS **

** NO ERRORS **

BLOCK COMMON CIBUF SIZE = 01664
PAGE 0003 FTN. 2:47 PM MON., 25 AUG., 1980

0012   BLOCK DATA
0013       *, 'CONTR' / CNTRL(256)
0014   COMMON / CONTR / CNTRL
0015   INTEGER CNTRL(256)
0016   END

FTN4 COMPILER: HP92060-16092.REV. 1926 (790430)
** NO WARNINGS ** NO ERRORS **
BLOCK COMMON CONTR SIZE = 00256
PROGRAM TXCO1 (3,99)

The operating system RTE IV B requests the data acquisition program TXCO for the one stage transonic compressor to be split into several programs scheduled by the father program TXCO. This son program TXCO1 consists of the subroutines ABRTX, CALIB, FREER and PACER. These handle the acquisition of high speed data. The data transfer between the father and son program takes place via the control array file CNTRL (disc file CNTRLF) and the data array IBUFF (disc file IBUFF).

The utility subroutines ACON, CNTL, CURVE, ICON, IPORT, PICTR, REWRF, RPACE, SCANR, TIME and WAIT are added.

Author: Hans M. Zeberner
Date: March 12, 1990

A detailed program description is available in the TXCO log.

Comment statements and statement numbers in the source code match to the program description. This program is part of the TXCO transonic compressor investigation program system.

COMMON / CONTR / CNTRL

INTEGER CNTRL(256)

DATA NOLF /0065371/

FORMAT (9X"A2")
10 FORMAT (*TXCOO, PROGRAM ABORTED! NO SUBROUTINE HAS BEEN INITIALIZED.")
901 FORMAT ("I17M3A1HOT3")
1201 FORMAT ("PF4GWT")
1501 FORMAT ("CA")

CALL CLEAR (7,1)
CALL LOCL (7)
CALL REWRF (1,2)

* First son program of father program TXCO:

CALL CLEAR (19,2)
IF (CNTRL(50).LT. 1 .OR. CNTRL(50).GT. 4 ) GO TO 65

......

Set interface bus and devices to remote control.

......

CALL ABRT (7,1)
CALL RMOTE (2)
CALL RMOTE (10)
CALL RMOTE (12)
CALL RMOTE (15)
WRITE (7,861)
WRITE (18,1001)
WRITE (18,1201)
WRITE (18,1301)

......

Call subroutine indicated by CNTRL(50).

ISTEP = CNTRL(50)
IF ( CNTRL(50) .EQ. 3 ) CALL CALIB
IF ( CNTRL(50) .EQ. 3 ) CALL FREER (X1,X2,X3)
IF ( CNTRL(50) .EQ. 4 ) CALL PACER (1)

......

Release interface bus and devices from remote control.

CALL CLEAR (7,1)
CALL LOCL (7)
CALL REWRF (1,2)
WRITE (LI,101) MOLF
GO TO (01,02,03,04) ISTOP
01 STOP 0177
02 STOP 0277
03 STOP 0377
04 STOP 0477
05 WRITE (LI,102)
END

FTN4 COMPILER: HP92860-16692 REV. 1926 (790430)
** NO WARNINGS ** NO ERRORS ** PROGRAM = 08257 COMMON = 08888
SUBROUTINE ABSRV

Subroutine to acquire high speed data from the 1-stage axial transonic compressor using miniaturized probes equipped with KULITE semiconductor pressure transducers.

Author: Hans Zebner
Date: August 12, 1980

A detailed program description is available in the TXCO log. Comment statements and statement numbers in the source code match the program description. This subroutine is part of the TXCO transonic compressor investigation program system.

Subroutine to acquire high speed data from the 'A' - 'B' - probe system.

COMMON / CBUF / IBUF
COMMON / CNTRL / IDIC, IFILE, ISIZE, ISEC, ICR

INTEGER IBUF(144), IFILE(3), ISIZE(2)
INTEGER CTRL(256), IDIC(144), IFILE(3), ISIZE(2)
REAL POS(7), RBUF(64)

INTEGER NOLF, NOCR(2), XCLR(3)

INTEGER IDCBS
DATA NOLF /0065373/
DATA NOCR /0000339, 0404333/
DATA XCLR /0155243, 0065373/
DATA IDCBS /144/

COMMON / CBUF / IBUF
COMMON / CNTRL / IDIC, IFILE, ISIZE, ISEC, ICR

INTEGER IBUF(144), IFILE(3), ISIZE(2)
INTEGER CTRL(256), IDIC(144), IFILE(3), ISIZE(2)
REAL POS(7), RBUF(64)

INTEGER NOLF, NOCR(2), XCLR(3)

INTEGER IDCBS
DATA NOLF /0065373/
DATA NOCR /0000339, 0404333/
DATA XCLR /0155243, 0065373/
DATA IDCBS /144/

C FORMAT ABSRV START

101 FORMAT (" Did you calibrate the type 'A' and type 'B' probes?")
102 FORMAT ("")
103 FORMAT (" 79X" "A2/"")
104 FORMAT (" Since you forgot to calibrate these nice probes, I will do it right now."")
105 FORMAT (" Enter the RETURN key to continue the execution of the program:"")
106 FORMAT (" Enter the following results from the on line call"
107 FORMAT (" of the probes."")
108 FORMAT (" Type 'A' probe"
109 FORMAT (" Probe"
110 FORMAT (" SLOPE 1 SLOPE 2"
111 FORMAT (" Type 'B' probe"
112 FORMAT (" SLOPE 1 SLOPE 2"
113 FORMAT (" Type 'C' probe"
114 FORMAT (" Type 'D' probe"
115 FORMAT (" Type 'E' probe"
116 FORMAT (" Type 'F' probe"
117 FORMAT (" Type 'G' probe"
118 FORMAT (" Type 'H' probe"
119 FORMAT (" Type 'I' probe"
120 FORMAT (" Type 'J' probe"
121 FORMAT (" Type 'K' probe"
122 FORMAT (" Type 'L' probe"
123 FORMAT (" Type 'M' probe"
124 FORMAT (" Type 'N' probe"
125 FORMAT (" Type 'O' probe"
126 FORMAT (" Type 'P' probe"
127 FORMAT (" Type 'Q' probe"
128 FORMAT (" Type 'R' probe"
129 FORMAT (" Type 'S' probe"
130 FORMAT (" Type 'T' probe"
131 FORMAT (" Type 'U' probe"
132 FORMAT (" Type 'V' probe"
133 FORMAT (" Type 'W' probe"
134 FORMAT (" Type 'X' probe"
135 FORMAT (" Type 'Y' probe"
136 FORMAT (" Type 'Z' probe"
137 FORMAT (" Type 'a' probe"
138 FORMAT (" Type 'b' probe"
139 FORMAT (" Type 'c' probe"
140 FORMAT (" Type 'd' probe"
141 FORMAT (" Type 'e' probe"
142 FORMAT (" Type 'f' probe"
143 FORMAT (" Type 'g' probe"
144 FORMAT (" Type 'h' probe"
145 FORMAT (" Type 'i' probe"
146 FORMAT (" Type 'j' probe"
147 FORMAT (" Type 'k' probe"
148 FORMAT (" Type 'l' probe"
149 FORMAT (" Type 'm' probe"
150 FORMAT (" Type 'n' probe"
151 FORMAT (" Type 'o' probe"
152 FORMAT (" Type 'p' probe"
153 FORMAT (" Type 'q' probe"
154 FORMAT (" Type 'r' probe"
155 FORMAT (" Type 's' probe"
156 FORMAT (" Type 't' probe"
157 FORMAT (" Type 'u' probe"
158 FORMAT (" Type 'v' probe"
159 FORMAT (" Type 'w' probe"
160 FORMAT (" Type 'x' probe"
161 FORMAT (" Type 'y' probe"
162 FORMAT (" Type 'z' probe"
163 FORMAT (" Type 'A' probe"
164 FORMAT (" Type 'B' probe"
165 FORMAT (" Type 'C' probe"
166 FORMAT (" Type 'D' probe"
167 FORMAT (" Type 'E' probe"
168 FORMAT (" Type 'F' probe"
169 FORMAT (" Type 'G' probe"
170 FORMAT (" Type 'H' probe"
171 FORMAT (" Type 'I' probe"
172 FORMAT (" Type 'J' probe"
173 FORMAT (" Type 'K' probe"
174 FORMAT (" Type 'L' probe"
175 FORMAT (" Type 'M' probe"
176 FORMAT (" Type 'N' probe"
177 FORMAT (" Type 'O' probe"
178 FORMAT (" Type 'P' probe"
179 FORMAT (" Type 'Q' probe"
180 FORMAT (" Type 'R' probe"
181 FORMAT (" Type 'S' probe"
182 FORMAT (" Type 'T' probe"
183 FORMAT (" Type 'U' probe"
184 FORMAT (" Type 'V' probe"
185 FORMAT (" Type 'W' probe"
186 FORMAT (" Type 'X' probe"
187 FORMAT (" Type 'Y' probe"
188 FORMAT (" Type 'Z' probe"
189 FORMAT (" Type 'a' probe"
190 FORMAT (" Type 'b' probe"
191 FORMAT (" Type 'c' probe"
192 FORMAT (" Type 'd' probe"
193 FORMAT (" Type 'e' probe"
194 FORMAT (" Type 'f' probe"
195 FORMAT (" Type 'g' probe"
196 FORMAT (" Type 'h' probe"
197 FORMAT (" Type 'i' probe"
198 FORMAT (" Type 'j' probe"
199 FORMAT (" Type 'k' probe"
200 FORMAT (" Type 'l' probe"
201 FORMAT (" Type 'm' probe"
202 FORMAT (" Type 'n' probe"
203 FORMAT (" Type 'o' probe"
204 FORMAT (" Type 'p' probe"
205 FORMAT (" Type 'q' probe"
206 FORMAT (" Type 'r' probe"
207 FORMAT (" Type 's' probe"
208 FORMAT (" Type 't' probe"
209 FORMAT (" Type 'u' probe"
210 FORMAT (" Type 'v' probe"
211 FORMAT (" Type 'w' probe"
212 FORMAT (" Type 'x' probe"
213 FORMAT (" Type 'y' probe"
214 FORMAT (" Type 'z' probe"

109
**HE to proceed to the next point** "A2/"

**EN to terminate the A/-/B/-probe survey at**

**3rd radius/operating point** "A2/"

"3X**, "4X**, "5X**, "6X**, "7X**, "8X**

118 FORMAT (** Error: You want to proceed to the "I2" position

but the data array only "B"="A2/

"A2" can store 12 positions, that you defined previously!"37X"

**A2**

149 FORMAT (**(3A2)**)

601 FORMAT (1H ,15(1H ),33HTransonic Compressor Test Run **,I7)

602 FORMAT (1H ,28(1H ),6HDate: ,A2,1H/,A2,1H/,A2)

603 FORMAT (1H ,"","",2B(1H ),6HTime: ,A2,1H/,A2,3H h,/////)

604 FORMAT (1H ,"","","","",3X)

**"A", "Y". ** Probes Page** "I3"**, **

605 FORMAT (" **,41X**"I2". Yaw position")

606 FORMAT (" **,42X**"I2". Yaw position**" **,72X")

607 FORMAT (**,28(1H ),6HTime: ,A2,1H/,A2,3H h**)

---

110
FORMATS A B S R V S T O P

Accounting.

IFACE = CNTRL(212)
IDOC = CNTRL(213)
IFACE = IFACE + 1
IDOC = IDOC + 1
ISZED = CNTRL(31)
ICR = CNTRL(30)
IFM = ICOM(CNTRL(4), 0)
ITYPE = 128
IFIRST = 2H1

I/O Assignments; preset data array.

L1 = CNTRL(19)
L0 = CNTRL(20)
L11 = CNTRL(71)
LO2 = CNTRL(72)
LO 01 I=I+1: 768
01 IBUF(I)= 0250528

Ask operator, whether the 'A'- 'B'-probe system has been calibrated on line.

WRITE (LI, 101) MODCA
READ (LI, 102) ICAL
WRITE (LI, 149) ICLR
IF (ICAL .EQ. 2H89) STOP. 0101
IF (ICAL .NE. 2H89) GO TO 82

111
Call subroutine CALIB to calibrate 'A'-'B'-probe system.

```fortran
CALL subroutine CALIB to calibrate 'A'-'B'-probe system.

WRITE (LI, 103) (N0FL, I=1,4)
READ (LI, 102) IDUM
WRITE (LI, 149) (ICLR, I=1,7)
IF (IDUM .LT. 2) STOP 0102
WRITE (LI, 104) 21st.
CALL CALIB

Enter calibration results; print heading.

WRITE (LI, 105) N0FL
READ (LI, 106) SLOPEA, SLOPEB, SLOPEA, SLOPEB, AVGREA, AVGREB, FBARD
CALL TIME (INMON, IYEAR, IDAY, IHOUR, IMIN)
WRITE (LO, 601) CNTRL (4)
WRITE (LO, 603) IYEAR, IDAY
WRITE (LO, 604) IHOUR, IMIN
WRITE (LO, 605) IPAGE

Initialize data acquisition; create raw data file of the correct size. If the file name assigned to this data set already exists, the operator decides whether to purge the already existing file (FL) or change this file name.

WRITE (LI, 107) NAME
READ (LI, 1) NPOS
WRITE (LI, 149) ICLR
ISIZE (I) = 1*NPOD
IF (IDOCF) LT. 100) GO TO 03
IF (IDOCF) = IDOCF-100
IF (IFILE) = IFIRST
IF (IFILE) = ICONF(100)
CALL CREATE (IDOCF, IFILE, ISIZE, ITYPE, ISECUR, ICR, IDCBS)
IF (IFILE) .LT. 0) GO TO 06
WRITE (LI, 108) IFILE
READ (LI, 102) IDUM
WRITE (LI, 149) ICLR, I=1,7
IF (IDOCF) .LT. 2) STOP 0103
IF (IDOCF) .LT. 2) STOP 0105
IDOCF = ICONF(100)
WRITE (LI, 149) IFILE, ISECUR, ICR
CALL PURGE (IDOCF, IFILE, ISECUR, ICR)
IF (IFILE) .LT. 0) STOP 0104
GO TO 06
IF (IFILE) = NEW
WRITE (NEW, 110) IDUM
WRITE (LI, 111) NEW, IFILE(2), IFILE(3)
IF (IFILE) = NEW
GO TO 06
CALL OPEN (IDOCF, ISECUR, IFILE, IOPM, ISECUR, ICR, IDCBS)
IF (IFILE) = NEW
GO TO 06
IF (IFILE) = NEW
STORE
ISECUR = ICONF(ISSECUR)
JCR = ICONF(JCR)
WRITE (LI, 112) IFILE, ISECUR, JCR, ITYPE, ISIZE
```

112
Position the raw data file is done by subroutine ABSRV. The actual data are written in the data file by subroutine PACER. CNTRL(37) is set to 1 in order to suppress printing a heading in subroutine PACER. CNTRL(39) is set to 1 to tell subroutine PACER not to create/open and close a new data file.

08 IPOS = IPOS+1
09 IF ( IPOS .GT. NPOS ) GO TO 19
10 CNTRL(37) = IPOS
11 CNTRL(39) = IPOS
12 IREC = 2+(IPOS-1)*8
13 CALL POSMT (IDCB, IERM, IREC)
14 IF ( IERM .LT. 0 ) STOP 0167

Check position of probes before acquiring data.

15 WRITE (LI, 113) NOLP
16 DO 10 J=1,35
17 POS(12) = SCANL(LB1,J,IC)
18 IF ( POS(12) .NE. POS(J) ) GO TO 14
19 WRITE (LI, 114) NOCR
20 READ (LI, 102) IDUM
21 IF ( IDUM .EQ. 2HNE ) GO TO 10
22 IF ( IDUM .EQ. 2HMA ) GO TO 15
23 GO TO 14

Acquire data in subroutine PACER.

24 WRITE (LI, 115) (POS(J), J=1,7,1), NOCR
25 READ (LI, 102) IDUM
26 WRITE (LI, 149) (ICLR, I=1,12)
27 IF ( IDUM .EQ. 2HUP ) GO TO 10
28 IF ( IDUM .EQ. 2HMA ) GO TO 15
29 GO TO 14

Select the next step:

RE repeat the data acquisition at this yaw position
EN terminate the survey at this operating point

30 WRITE (LI, 117) NOLP; IPOS; NOLP; I=1,8; 1; NOCR
31 READ (LI, 102) IDUM
32 WRITE (LI, 149) (ICLR, I=1,12)
33 IF ( IDUM .EQ. 2HNE ) GO TO 08
IF (IBUF .EQ. -999) GO TO 19
GO TO 16

Stop data acquisition. Write additional data (i.e. barometric pressure, calibration results, number of points and yaw positions into first record (Directory) of the data file.

17 CONTINUE
GO TO 18
18 IBUF(I) = 4250528
IBUF(2) = NPOS
RBUF(2) = PBARO
RBUF(3) = PFF
RBUF(4) = AVGOEA
RBUF(5) = AVGEB
RBUF(6) = SLOPEA
RBUF(7) = SECONDA
RBUF(8) = SLOPEB
RBUF(9) = SECONDB
CALL TIME (IBUF(96),IBUF(104),IBUF(112),IBUF(120),IBUF(128))
IREC = 1
CALL WRITE (IDCB, IERR, IBUF, IL, IREC)
IF (IERR .LT. 0) STOP 8112
CALL CLOSE (IDCB, IERR)
IF (IERR .LT. 0) STOP 8113
WRITE (LO, 607) IBUF(120), IBUF(128)

Terminate subroutine; write accounting variables back into control array.

CNTL(212) = IPAGE
CNTL(213) = IDCBO
CNTL(50) = -1
RETURN

Error return.

19 WRITE (LI, 'i18') NPOS, NOLF, NPOS, NOLF
GO TO 17
END

FTN4 COMPILER: HP92060-16692 REV. 1926 (790430)

SS NO WARNINGS SS NO ERRORS SS PROGRAM = 02498 COMMON = 00000
SUBROUTINE CALIB

* subroutine to control the on line calibration of the KULITE type 'A'-"B"-probe system.

Author: Hans Zehnder
Date: August 13, 1980

A detailed program description is available in the TXCO log.
Comment statements and statement numbers in the source code match to the program description. This subroutine is part of the TXCO transonic compressor investigation program system.

* On-line calibration of KULITE probes.

COMMON / CONTRL / CNTRL
INTEGER CNTRL(256)

REAL AVOLT(10), BVOLT(10), RPRES(10), DMM(10)
INTEGER NOCR(2), ICLR(3), ITIME(5), IO(5)

DATA M12 /0065372/
DATA ICLR /0000338, 0404338/
DATA ICLR /0155248, 0155158, 0065372/}

FORMATS CALIB START
101 FORMAT ("9X"="A2/
202 FORMAT (="KULITE pressure transducers: Input DMM =A2/
303 FORMAT (="MULTIPLIER read out to initialize calibration, RE
404 FORMAT (="To repeat this part of the "A2/
505 FORMAT (="calibration or EN to terminate the on line calibr
606 FORMAT (="ation!"="A2/
707 FORMAT (="A2/
808 FORMAT (="CALIB : CALL P/R"("I2")"
909 FORMAT (="Error: You did not perform a calibration at all
1010 FORMAT (="Error: Please, ask yourself honestly, whether j
1111 FORMAT (="Just one point is sufficient"="A2/
1212 FORMAT (="to give an accurate calibration curve fit? I frankly doubt
1313 FORMAT (="i1=(1S="A2/
1414 FORMAT (=""="A2/
1515 FORMAT (=""="A2/
1616 FORMAT (=""="A2/
1717 FORMAT (=""="A2/
1818 FORMAT (=""="A2/
1919 FORMAT (=""="A2/
2020 FORMAT (=""="A2/
2121 FORMAT (=""="A2/
2222 FORMAT (=""="A2/
2323 FORMAT (=""="A2/
2424 FORMAT (=""="A2/
2525 FORMAT (=""="A2/
2626 FORMAT (=""="A2/
2727 FORMAT (=""="A2/
2828 FORMAT (=""="A2/
2929 FORMAT (=""="A2/
3030 FORMAT (=""="A2/
3131 FORMAT (=""="A2/
3232 FORMAT (=""="A2/
3333 FORMAT (=""="A2/
3434 FORMAT (=""="A2/
3535 FORMAT (=""="A2/
3636 FORMAT (=""="A2/
3737 FORMAT (=""="A2/
3838 FORMAT (=""="A2/
3939 FORMAT (=""="A2/
4040 FORMAT (=""="A2/
4141 FORMAT (=""="A2/
4242 FORMAT (=""="A2/
4343 FORMAT (=""="A2/
4444 FORMAT (=""="A2/
4545 FORMAT (=""="A2/
4646 FORMAT (=""="A2/
4747 FORMAT (=""="A2/
4848 FORMAT (=""="A2/
4949 FORMAT (=""="A2/
5050 FORMAT (=""="A2/
5151 FORMAT (=""="A2/
5252 FORMAT (=""="A2/
5353 FORMAT (=""="A2/
5454 FORMAT (=""="A2/
5555 FORMAT (=""="A2/
5656 FORMAT (=""="A2/
5757 FORMAT (=""="A2/
5858 FORMAT (=""="A2/
5959 FORMAT (=""="A2/
6060 FORMAT (=""="A2/
6161 FORMAT (=""="A2/
6262 FORMAT (=""="A2/
6363 FORMAT (=""="A2/
6464 FORMAT (=""="A2/
6565 FORMAT (=""="A2/
6666 FORMAT (=""="A2/
6767 FORMAT (=""="A2/
6868 FORMAT (=""="A2/
6969 FORMAT (=""="A2/
7070 FORMAT (=""="A2/
7171 FORMAT (=""="A2/
7272 FORMAT (=""="A2/
7373 FORMAT (=""="A2/
7474 FORMAT (=""="A2/
7575 FORMAT (=""="A2/
7676 FORMAT (=""="A2/
7777 FORMAT (=""="A2/
7878 FORMAT (=""="A2/
7979 FORMAT (=""="A2/
8080 FORMAT (=""="A2/
8181 FORMAT (=""="A2/
8282 FORMAT (=""="A2/
8383 FORMAT (=""="A2/
8484 FORMAT (=""="A2/
8585 FORMAT (=""="A2/
8686 FORMAT (=""="A2/
8787 FORMAT (=""="A2/
8888 FORMAT (=""="A2/
8989 FORMAT (=""="A2/
9090 FORMAT (=""="A2/
9191 FORMAT (=""="A2/
9292 FORMAT (=""="A2/
9393 FORMAT (=""="A2/
9494 FORMAT (=""="A2/
9595 FORMAT (=""="A2/
9696 FORMAT (=""="A2/
9797 FORMAT (=""="A2/
9898 FORMAT (=""="A2/
9999 FORMAT (=""="A2/
0000 FORMAT (=""="A2/
0101 FORMAT (=""="A2/
0202 FORMAT (=""="A2/
0303 FORMAT (=""="A2/
0404 FORMAT (=""="A2/
0505 FORMAT (=""="A2/
0606 FORMAT (=""="A2/
0707 FORMAT (=""="A2/
0808 FORMAT (=""="A2/
0909 FORMAT (=""="A2/
1010 FORMAT (=""="A2/
1111 FORMAT (=""="A2/
1212 FORMAT (=""="A2/
1313 FORMAT (=""="A2/
1414 FORMAT (=""="A2/
1515 FORMAT (=""="A2/
1616 FORMAT (=""="A2/
1717 FORMAT (=""="A2/
1818 FORMAT (=""="A2/
1919 FORMAT (=""="A2/
2020 FORMAT (=""="A2/
2121 FORMAT (=""="A2/
2222 FORMAT (=""="A2/
2323 FORMAT (=""="A2/
2424 FORMAT (=""="A2/
2525 FORMAT (=""="A2/
2626 FORMAT (=""="A2/
2727 FORMAT (=""="A2/
2828 FORMAT (=""="A2/
2929 FORMAT (=""="A2/
3030 FORMAT (=""="A2/
3131 FORMAT (=""="A2/
3232 FORMAT (=""="A2/
3333 FORMAT (=""="A2/
3434 FORMAT (=""="A2/
3535 FORMAT (=""="A2/
3636 FORMAT (=""="A2/
3737 FORMAT (=""="A2/
3838 FORMAT (=""="A2/
3939 FORMAT (=""="A2/
4040 FORMAT (=""="A2/
4141 FORMAT (=""="A2/
4242 FORMAT (=""="A2/
4343 FORMAT (=""="A2/
4444 FORMAT (=""="A2/
4545 FORMAT (=""="A2/
4646 FORMAT (=""="A2/
4747 FORMAT (=""="A2/
4848 FORMAT (=""="A2/
4949 FORMAT (=""="A2/
5050 FORMAT (=""="A2/
5151 FORMAT (=""="A2/
5252 FORMAT (=""="A2/
5353 FORMAT (=""="A2/
5454 FORMAT (=""="A2/
5555 FORMAT (=""="A2/
5656 FORMAT (=""="A2/
5757 FORMAT (=""="A2/
5858 FORMAT (=""="A2/
5959 FORMAT (=""="A2/
6060 FORMAT (=""="A2/
6161 FORMAT (=""="A2/
FORMATS CALIB STOP

Accounting.
Assignments; print heading.

\[ \text{IPAGE} = \text{CNTRL}(214) \]
\[ \text{IPAGE} = \text{IPAGE} + 1 \]

Limit on line calibration calibration to two A/D channels, that contain the type 'A' and 'B' probe output voltage.

The current value of CNTRL(230) is temporarily stored in the variable ITEMP.

\[ \text{IPAGE} = \text{CNTRL}(230) \]
\[ \text{CNTRL}(230) = 2 \]

Select the next step:

BE  Repeat this point
EN  Terminate the on line calibration
any number  Proceed to the next point

01 WRITE (LI, 101) (WOLF,1=1,5,1)
READ (LI, 102) IO,
WRITE (LI, 149) (IT, IP = 1, 8, 1)
IF (IDU(1), EQ. 2H$N) GO TO 01
IF (IDU(1), EQ. 2HN) STOP 0201
IDU = 0
GO 02 I = 5,1
IF (IDU, NE. 2M) IDU = 1
CALL CODE
READ (IO, 103) DM

Take free run data at a defined reference pressure.

03 IPTS = IPTS + 1
DM(IPTS) = DM
04 WRITE (LI, 104)
WRITE (LI, 105) NOCR
READ (LI, 102) IDUM
WRITE (LI, 149) ICLR
IF (IDUM, EQ. 2H$N) STOP 0202
CNTRL(13) = IPTS
WRITE (LO, 605) IPTS
CALL FREER (AVOLT(IPTS), BVOLT(IPTS), RPRES(IPTS))
Select the next step:

EA Repeat this point
any number Proceed to the next point

05 WRITE (LI, 101) (WOLF, I=1,5,I)
WRITE (LI, 100) IDM
IF (100) :EQ. 2HAE) GO TO 04
IF (100) :EQ. 2HGA) STOP 0203
IDUM = 8
06 IF (101) NE: 2H } IDUM = 8
CALL CDE
READ (IO,103) DM
GO TO 03

Take paced run data at one defined reference pressure.

07 IF (`IPTS' :EQ. 'I';) GO TO 11
CNTBL(37) = 1
WRITE (LI, 106) NOCR
READ (LI, 102) IDUM
WRITE (LI, 149) ICLR
IF (IDUM :EQ. 2HAE) STOP 0204
IREC
WRITE (LI, 107) IREC
WRITE (LO, 606)
CALL PACER (IREC)

Calculate linear curve fit through data points.

WRITE (LO, 605)
DO 66 I=1,IPTS,1
08 WRITE (LO, 600) I, AVOLT(I), BVOLT(I), TVPRES(I), I, DMM(I)
CALL CURVE (IPTS, AVOLT, BVOLT, TVPRES, SLOPE, SECON)
WRITE (LO, 605) SLOPE, SECON
CALL CURVE (IPTS, AVOLT, TVPRES, SLOPE, SECON)
WRITE (LO, 610) SLOPE, SECON

Terminate subroutine: Write accounting variables back into control array; set CNTBL(13) back to its previous value.

09 CNTBL(37) = 1
CNTBL(38) = 1
CNTBL(214) = IPAGE
CNTBL(213) = IMON'IYEP, IYEAR, IMON, IMIN)
WRITE (LO, 611) IMON, IYEP, IYEAR, IMIN
RETURN
683  Error returns.
684
685  10 WRITE (LI, 109) 'NOLF'
686  GO TO 09
687  11 WRITE (LI, 109) ('NOLF',I=1,2,1)
688  GO TO 09
689
690  END

FTN4 COMPILER: HP92060-16092 REV. 1926 (790430)

## NO WARNINGS ## NO ERRORS ## PROGRAM = 01525    COMMON = 00000
SUBROUTINE FREER (AVOLTBVQLTPREFR)

Subroutine to acquire data using the HP 5610A A/D converter.

C If the A/D converter is operated in free run mode.

Author: Hans Zebner

Date: August 14, 1980

A detailed program description is available in the TXCO log.

Comment statements and statement numbers in the source code
match to that program description. This subroutine is part of
the TXCO transonic compressor investigation program system.

C Takes data from KULITE probes; A/D free run mode.

COMMON / CIBUF / IBUF
COMMON / CONTR / CNTRL
COMMON / FMP / IDCBLIFILEISIZEISECUICR
INTEGER IBUF(16), CNTRL(256)
INTEGER IDCB(L44), IFILE(3), ISIZE(2)
INTEGER NOLF, NOCR, ICLR(3), IOXIM(9)

DATA NOLF / 0065371/
DATA NOCR / 0000000, 0400338/
DATA ICLR / 0999129, 0155159, 0065379/
DATA LSPLT / 1.0/
DATA IDCBS / 141/

FREIR START

101 FORMAT (** acquires additional data**).27X"A2)
102 FORMAT (** acquires additional data**).27X"A2)
103 FORMAT (** acquires additional data**).27X"A2)
104 FORMAT (** acquires additional data**).27X"A2)
105 FORMAT (** acquires additional data**).27X"A2)
106 FORMAT (** acquires additional data**).27X"A2)
107 FORMAT (** acquires additional data**).27X"A2)
108 FORMAT (** acquires additional data**).27X"A2)

FREIR: Prints purge to and enters any channel.

FREIR: Prints purge to and enters any channel.

FREIR: Prints purge to and enters any channel.

FREIR: Prints purge to and enters any channel.

FREIR: Prints purge to and enters any channel.

FREIR: Prints purge to and enters any channel.

FREIR: Prints purge to and enters any channel.

FREIR: Prints purge to and enters any channel.

FREIR: Prints purge to and enters any channel.

FREIR: Prints purge to and enters any channel.

FREIR: Prints purge to and enters any channel.

FREIR: Prints purge to and enters any channel.

FREIR: Prints purge to and enters any channel.

FREIR: Prints purge to and enters any channel.

FREIR: Prints purge to and enters any channel.

FREIR: Prints purge to and enters any channel.

FREIR: Prints purge to and enters any channel.

FREIR: Prints purge to and enters any channel.

FREIR: Prints purge to and enters any channel.

FREIR: Prints purge to and enters any channel.

FREIR: Prints purge to and enters any channel.

FREIR: Prints purge to and enters any channel.

FREIR: Prints purge to and enters any channel.

FREIR: Prints purge to and enters any channel.

FREIR: Prints purge to and enters any channel.

FREIR: Prints purge to and enters any channel.

FREIR: Prints purge to and enters any channel.

FREIR: Prints purge to and enters any channel.

FREIR: Prints purge to and enters any channel.

FREIR: Prints purge to and enters any channel.

FREIR: Prints purge to and enters any channel.

FREIR: Prints purge to and enters any channel.

FREIR: Prints purge to and enters any channel.

FREIR: Prints purge to and enters any channel.

FREIR: Prints purge to and enters any channel.

FREIR: Prints purge to and enters any channel.

FREIR: Prints purge to and enters any channel.

FREIR: Prints purge to and enters any channel.

FREIR: Prints purge to and enters any channel.

FREIR: Prints purge to and enters any channel.

FREIR: Prints purge to and enters any channel.

FREIR: Prints purge to and enters any channel.

FREIR: Prints purge to and enters any channel.

FREIR: Prints purge to and enters any channel.

FREIR: Prints purge to and enters any channel.

FREIR: Prints purge to and enters any channel.

FREIR: Prints purge to and enters any channel.

FREIR: Prints purge to and enters any channel.

FREIR: Prints purge to and enters any channel.

FREIR: Prints purge to and enters any channel.

FREIR: Prints purge to and enters any channel.

FREIR: Prints purge to and enters any channel.

FREIR: Prints purge to and enters any channel.

FREIR: Prints purge to and enters any channel.

FREIR: Prints purge to and enters any channel.

FREIR: Prints purge to and enters any channel.

FREIR: Prints purge to and enters any channel.

FREIR: Prints purge to and enters any channel.

FREIR: Prints purge to and enters any channel.

FREIR: Prints purge to and enters any channel.

FREIR: Prints purge to and enters any channel.

FREIR: Prints purge to and enters any channel.

FREIR: Prints purge to and enters any channel.

FREIR: Prints purge to and enters any channel.

FREIR: Prints purge to and enters any channel.

FREIR: Prints purge to and enters any channel.

FREIR: Prints purge to and enters any channel.

FREIR: Prints purge to and enters any channel.

FREIR: Prints purge to and enters any channel.

FREIR: Prints purge to and enters any channel.

FREIR: Prints purge to and enters any channel.

FREIR: Prints purge to and enters any channel.

FREIR: Prints purge to and enters any channel.

FREIR: Prints purge to and enters any channel.

FREIR: Prints purge to and enters any channel.

FREIR: Prints purge to and enters any channel.

FREIR: Prints purge to and enters any channel.

FREIR: Prints purge to and enters any channel.

FREIR: Prints purge to and enters any channel.

FREIR: Prints purge to and enters any channel.

FREIR: Prints purge to and enters any channel.

FREIR: Prints purge to and enters any channel.

FREIR: Prints purge to and enters any channel.

FREIR: Prints purge to and enters any channel.

FREIR: Prints purge to and enters any channel.

FREIR: Prints purge to and enters any channel.

FREIR: Prints purge to and enters any channel.

FREIR: Prints purge to and enters any channel.

FREIR: Prints purge to and enters any channel.

FREIR: Prints purge to and enters any channel.

FREIR: Prints purge to and enters any channel.

FREIR: Prints purge to and enters any channel.

FREIR: Prints purge to and enters any channel.

FREIR: Prints purge to and enters any channel.

FREIR: Prints purge to and enters any channel.

FREIR: Prints purge to and enters any channel.

FREIR: Prints purge to and enters any channel.

FREIR: Prints purge to and enters any channel.

FREIR: Prints purge to and enters any channel.

FREIR: Prints purge to and enters any channel.

FREIR: Prints purge to and enters any channel.

FREIR: Prints purge to and enters any channel.

FREIR: Prints purge to and enters any channel.

FREIR: Prints purge to and enters any channel.

FREIR: Prints purge to and enters any channel.

FREIR: Prints purge to and enters any channel.

FREIR: Prints purge to and enters any channel.

FREIR: Prints purge to and enters any channel.

FREIR: Prints purge to and enters any channel.

FREIR: Prints purge to and enters any channel.

FREIR: Prints purge to and enters any channel.

FREIR: Prints purge to and enters any channel.

FREIR: Prints purge to and enters any channel.

FREIR: Prints purge to and enters any channel.

FREIR: Prints purge to and enters any channel.

FREIR: Prints purge to and enters any channel.

FREIR: Prints purge to and enters any channel.

FREIR: Prints purge to and enters any channel.

FREIR: Prints purge to and enters any channel.

FREIR: Prints purge to and enters any channel.

FREIR: Prints purge to and enters any channel.

FREIR: Prints purge to and enters any channel.

FREIR: Prints purge to and enters any channel.

FREIR: Prints purge to and enters any channel.

FREIR: Prints purge to and enters any channel.

FREIR: Prints purge to and enters any channel.

FREIR: Prints purge to and enters any channel.

FREIR: Prints purge to and enters any channel.

FREIR: Prints purge to and enters any channel.

FREIR: Prints purge to and enters any channel.

FREIR: Prints purge to and enters any channel.

FREIR: Prints purge to and enters any channel.

FREIR: Prints purge to and enters any channel.

FREIR: Prints purge to and enters any channel.

FREIR: Prints purge to and enters any channel.

FREIR: Prints purge to and enters any channel.

FREIR: Prints purge to and enters any channel.

FREIR: Prints purge to and enters any channel.
0747  901 FORMAT (" ERROR DETECTED IN PROGRAM FREER")
0748  *  CALL EXEC (1, 20, BUF(1), "ID = "12", 4")
0749  902 FORMAT (" A REGISTER IS " & A" B REGISTER IS " & B")
0771  C FORMATS FREER STOP
Accounting.

IMAGE = CNTRL(216)
IDOF = CNTRL(317)
IPAGE = IPAGE+1
ISCIO = CNTRL(33)
IRC = CNTRL(30)
ISIZE(4) = 138
ITYPE(2) = ICON(CNTRL(4),0)
STRT = 3
ILT = 1664

I/O Assignments; preset data array.

DO 01 I=1,1664 1
   01 IBUF(I) = 177777B
   LD = CNTRL(39)
   LB1 = CNTRL(71)
   LB2 = CNTRL(71)
   LB3 = CNTRL(41)
   IB54 = CNTRL(64)
   IB57 = CNTRL(230)
   NPT2 = CNTRL(251)
   NPT3 = NPT2-1
   IMASK = 177700B
   IU = CNTRL(250)

Print heading, unless CNTRL(37) is set to 1.

IF ( CNTRL(37) ) EQ. 1 ) GO TO 02
   CALL TIME (IMON, IDAY, IYEAR, IHOUR, IMIN)
   WRITE (LOG, 601) CNTRL(4)
   WRITE (LOG, 602) IMON, IDAY, IYEAR
   WRITE (LOG, 603) IHOUR, IMIN
   WRITE (LOG, 604) IPAGE
   02 IF ( CNTRL(36) ) EQ. 1 ) WRITE (LOG, 605)

Start data acquisition loop.

DO 19 JI=1,NPT3,1
   WRITE (LI, 101) NDFC
   IDOF = IDOC-1
   IDOCF = IDOCF+1
   IF (IDOCF .LT. 100 ) GO TO 03
   19 WRITE(JI,101)
      IF (NPT3 = 1) GO TO 03
      IDOCF = IDOCF-100
      IF (FILE(3) ) ICON(IDOCF,0)

122
Acquire additional data required for reduction procedure.

*PROC = SCANR(LB1,19,2)
FRGR = FRGR10,0
*GRM = SCANR(LB1,33,1)
*GRP = SCANR(LB1,33,1)
*PRFR = PREP
*PR1 = ACOM(ISV4, 3, IW)
*PR3 = ACOM(ISV4, 4, IW)
*PR4 = ACOM(ISV4, 4, IW)
DE = SCANR(LB2,19,1)

Get correct probe positions.

IF (ICHNL .EQ. 0 ) GO TO 04
IF (ICHNL .EQ. 1 ) GO TO 05

Type 'A' KULITE probe (on A/D input channel 0).

04 XIM = SCANR(LB1,33,1)
YAW = SCANR(LB1,33,1)
CALL CODE
WRITE (IOXIM,102) XIM,YAW
GO TO 07

Type 'B' KULITE probe (on A/D input channel 1).

05 XIM = SCANR(LB1,33,1)
YAW = SCANR(LB1,33,1)
CALL CODE
WRITE (IOXIM,103) XIM,YAW
GO TO 07

Wall KULITE (on A/D input channels 2 and higher).

06 CALL CODE
WRITE (IOXIM,104)

Preset unused elements of data array IBUF.

07 IF ( NRPT2 .EQ. 1664 ) GO TO 09
DO 08 J2=NRPT3,1664,1
08  IBUF(J2) = 0

Read voltages from A/D converter into data array IBUF;
starting element is IBUF(I); A/D is operated in the
free run mode (4).

09  WRITE (LI, '105') NRPT2
    CALL EXEC (I+100000,20,IBUF(1),NRPT2,ICHNL,4)
    GO TO 11
10  CALL ABREG (IA,IB)
    GO TO 21

Calculate average voltage.

12  WRITE (LI, '105') NRPT2
    DO 13 J2=1,9
    IBUF(J2) = AND(IBUF(J2),IMASK)
13  AVERAGE = 0.0
    DO 14 J2=1,9 NRPT2
    AVERAGE = AVERAGE+FLOAT(IBUF(J2))
14  AVERAGE = F6LOAT(/AVERAGE/32768.0)/NRPT2

Write average KULITE voltage (after amplification and
conversion to digital notation) into AVOLT and BVOLT.
Depending on what signal has been digitized, the type 'A'
probe is on A/D input channel 0 and the type 'B' on 1.

IF ( ICHNL.EQ. 0 ) AVOLT = AVERAGE
IF ( ICHNL.EQ. 1 ) BVOLT = AVERAGE

IF ( ISP .LT. 5 ) GO TO 15
15  ISP = ISP + 1
    IF ( CNTVL(77) .NE. 1 ) WRITE (LO, 606)

Save data in file.

16  WRITE (LI, '107') IPSL, NDL, NRPT2,
    CALL CHENT (ICDB,IER,FILE,IBIZE,ITYPE,ISBCU,ICR,IDCBS)
    IF ( IERR .EQ. 0 ) GO TO 16
    WRITE (LI, '106') IFILE
    READ (LI, '109') IDOM
    WRITE (LI, '112') (ICLB,I=1,3,4)
    IF ( IBUF .EQ. 255) STOP 1391
    IBUCU = ICON(ISBCU,0)
    WRITE (LI, '113') IBUCU,ICR
    CALL PURGE (ICDB,IER,IBUCU,ICR)
    IF ( IERR .EQ. 0 ) STOP 1392
    GO TO 17
17  CALL CODE

124
WRITE (NEW,110) IDUM
WRITE (NEW,111) IDUM,NEW,IFILE(1),IFILE(2)
GO TO 16

18 CALL OPEN (IDCB, IERR, IFILE, IOPN, IBECU, ICR, IDCBS)
       IF ( IERR .LT. 6 ) STOP 6363
CALL WRITE (IDCB, IERR, Ibuf) !
       IF ( IERR .LT. 6 ) STOP 6364
CALL CLOSE (IDCB, IERR, 6)
       IF ( IERR .LT. 6 ) STOP 6365
WRITE (LT, 146) ICR,
JSECO = ICON(JSECO, 0)
JCR = ICON(ICR, 0)

Step data acquisition loop.

19 WRITE (LO, 609) IDOC, ICHNL, KFT2, CIN, CYAN, P1, P23, P4, E, BE, TOXIN, P
       & REP, FREQ, AURGE, JFILE, JSECU, JCR, IREC

Terminate subroutine; write accounting variables back
into control array.

IF (CTRL(37) .LT. 1) GO TO 20
CALL TIME (IMON, IDAY, IYEAR, IHOUR, IMIN)
WRITE (LO, 608) IHOUR, IMIN

20 CTRL(36) = 3
CTRL(30) = 3
CTRL(216) = IFAGE
CTRL(217) = IDOC
RETURN

Error returns from EXEC calls; output error message to the
line printer and look what's in the A and B register.

21 WRITE (*, 901) KFT2, ICHNL
WRITE (*, 902) IA, IB
GO TO 20
END
PACE 0026

FTH.

14

t........................ .....................
...
, ..................
SUDROUTtNE
PACER (tREC)

:

LOSS.
Io
C
1
61

2:47 PM

MON., 2S

Subroutine to acquire data using the HP S61OA AID converter
if the AID converter Is triggered through the pacer (poced
run node). Hans Zobfer
Aulhort
1990
1
August
Date:
Cnacho the progra
th

6S

eC

tkesdat
PM

desriPtio n

U~i

pobe;

h
stia

tssorIn

eooeep
rnoi

eC

Z
1069

AUG., 1980

/

svbroutint Is
$

on
•

raot

rt

syeoeon.eo

Mun
ode.

paced

COMMON / CIDUF / !rnJF
COMMON / CONTR / CNTRL
FMON / FMP
If
/ IDCDIFILEISIZEISECUICR

a

INTEGER ISUF(1664)

107Z

IN

1074
to??7

G R

NDCB(R44),ICFLE(3),ISIZ

()

E(2)

INTEGER NOLFNOCR(2),ICLR(3),IBUFI(384),IOXIN(9),IAVRGE(2)

10

EQUIVALENCE (I3UF(1),ICUFi(1)),(IAVJRGE(l),AVRGE)

t081
16982

DATA NOLF
DATA NOCR

/006S37/
/00003330404339/
166
f~24.OlS1I9,G6S37U/
108s
C
"f R Pr.
X
PACER
START
1086
101 FORMAT ('4IX'acquiring additional data required for reducliom

C*n

too?

~t3Cdur O10XOA2)

FORAT ('AFB.6,F9.6)
FORHAT C¢BQFS.6 F9.6)
FORMAT ( SH
Wall KULITE)
FORMAT ('"/0079X''/, The next signal to be 1dAtized
in paced run moade is on A D input channelO13."/
i
K Plug in a lead from the amplilier output to the oscilloscope
U/
* If you desire
... forget the C
S to onitor the data acquisition.
Conparator output sional.
a/'79X'/
C' Press CR to continue the progran execution!"3SX"

lei
103
104
10S

fl
109

1091

1092
1093
094
107
1096
L097

*u/W'I79X"W/)

is

106FO AT
(0021X'displaving wave fort on terninal LUO"12,20X"
0 AIMHTA2
107 FORMAT (00/'79X00/0 CHECK digitized output on a
Cuziliary console against amplylfer outut fed into a/
*' A/D inUt channel "I2" . Press CR if data a
Cre OK *
f an error Is suspected
/
*0 type RE to repeat this dat a cquisition9'3X"/

It q.
1161
1i0
1103
1134
1105

"

1112
J113

14

1117

~1f

Is$ FORMAT ("26Xstorino data in file 3A2 26X'Af)
Fi
03A20 already exst
Type 0,
*'PU
purge
1,.naft.*38X)
Ttecia.any char-",
.
c8er r or enter
/.0 to allow
t~t
110 FRMAT (
A
: PUpR1Er3'b :f2 !32)
III FORMAT (A1034)
t12 FORMAT (" PACER : File noe 03A2" succesfully changed to "342)

09 9 F RMAT ('WARNING

1109

FORMAT (ZIA2)
J1 141F81MAT (/"A2))

it

F RMAT 3~
601 FORMAT (IH .i (H
1

f* FORMAT~~(10 ~
1M //
8

(1

,

~

.
66S FORMAT
AD (/l0:"
I

*

cre-

12

Cnent

C

),33HTranlsonic Coaressor Test Rvn 0 '17)
/II / /
A2 )~
at: 2,M
H~ ),&H)8/t/I

O "l,

p '.

j/tX;072X80/
/S" Input

pacer

blade

start

SIX
0 kllonsO/WX0
aPle channel
7
coun,
2X / mode
bation probei:fi4XO/

pair

count

stop

# rep e-a/

126

NN

in

.


$ 1X"{
 immersion yaw g. pressure RPM voltage $
$^Tt$ 1X"Probe reference rotor average$
$^Tx$ 1X"starting"$
$^t$ data in file record e"/1X="72X")
$609$ FORMAT (1H )
$610$ FORMAT (1X="72X"/1X="BIG"/1X="7F9.6,9X"
$^T$/9A2,9.9,19,F9.6,6X,3A2:"A2:"A2,19="/" 72X")
$611$ FORMAT ("

*,28(1H ),6HTime: ,A2,1H.,A2,3H h)
FORMATS PACTER STOP

Accounting.

IF ('CNTRL(31) .EQ. 1') GO TO $1
IPAGE = CNTRL(18)
IDATE = CNTRL(19)
IDOC = CNTRL(20)
ISIZE = CNTRL(30)
ISIZE2(2) = C8
ITYPE = C8
IF(2) = FCON(CNTRL(4),0)

01 I$FRST = 2HT3
02 IL = 3B4

I/O Assignments; preset data array.

02 IBUFI(I) = 1777778
03 IBUFI(I) = 256,296,1
03 IBUFI(I) = 0
01 I$ = CNTRL(19)
02 LO = CNTRL(20)
03 IA = CNTRL(21)
04 IA$ = CNTRL(22)
05 IA$ = CNTRL(23)
06 IA$ = CNTRL(24)
05 IA$ = CNTRL(25)

Print heading, unless CNTRL(37) is set to 1.

IF ('CNTRL(37) .EQ. 1') GO TO $4
CALL TIME (IMON, IDAY, IYEAR, IHOUR, IMIN)
WRITE (LO, 401) CNTRL(4)
WRITE (LO, 402) CNTRL(4)
WRITE (LO, 403) CNTRL(4)
WRITE (LO, 404) IPAGE
04 IF ('CNTRL(38) .EQ. 1') WRITE (LO, 405)

C
Start data acquisition loop.

DO 19 J1=1,NAPI

WRITE (LI, 101) MOLF
ICHNL = CNTRL(230+J1)
IF ( CNTRL(235) .EQ. 1 ) GO TO 06
IDOCF = IDOCF+1
IF ( IDOCF .LT. 100 ) GO TO 05
IF ( FTST = 2 ) GO TO 06
IDOCF = IDOCF-100
IFILE(1) = IFST
IFILE(3) = ICON(IDOCF,0)

Acquire additional data required for reduction procedure.

FREQ = SCANR(LS1,17;2)
FREQ = FREQ/10.0
DIM = SCANR(LS1,30;1)
CYAW = SCANR(LS1,31;1)
PREF = SCANR(LS1,32;1)
P1 = ACQN(ISV4, 3,1W)
P3 = ACQN(ISV4, 5,1W)
DE = SCANR(LS2,19;1)

Get correct probe position.

IF ( ICHNL .EQ. 3 ) GO TO 03
IF ( ICHNL .EQ. 1 ) GO TO 08
GO TO 09

Type 'A' KULITE probe (on A/D input channel 0).

XIM = SCANR(LS1,33;1)
YAW = SCANR(LS1,35;1)
CALL CODE
WRITE (10XIM,102) XIM,YAW
IADD = 0
IRED(100) = 1
GO TO 10

Type 'B' KULITE probe (on A/D input channel 1).

XIM = SCANR(LS1,34;1)
YAW = SCANR(LS1,35;1)
CALL CODE
WRITE (10XIM,103) XIM,YAW
IADD = 576
IRED(100) = 2
GO TO 10

129
Wall KULITE (on A/D input channels 2 and higher).

09 CALL CASE
WRITE (IDOIM,104)
IDUMF1(294) = 9
IOM = 99

Inform operator about next data scan; wait for ready message; continue.

10 WRITE (LI,149) IDUM
READ (LI,149) IDUM
WRITE (LI,149) (ICLR,F=110,1)
IF ( IDUM .EQ. 2H8E-7 ) STOP 0401

Acquire high speed data in paced run mode.

IFAMO = CALR(L220)
IPAIR = CNTRL(221)
IF ( IFAMO .EQ. 3 ) ISTART = CNTRL(222); IABB
INCR = CNTRL(223)
CALL RPACE (ICHLN,IPAMO,IPAIR,ISTART,INCR,ISTOP,IRPT,AVERCE,0,0)

Write additional data into data array.

IDUMF1(265) = IDOC
IDUMF1(266) = IPAMO
IDUMF1(268) = IPAIR
IDUMF1(269) = ISTART
IDUMF1(270) = INCR
IDUMF1(271) = ISTOP
IDUMF1(272) = IRPT
IDUMF1(273) = CIM
IOM = 1000000
IDUMF1(275) = P23
IOM = 1000000
IDUMF1(277) = P4
IOM = 1000000
IDUMF1(278) = DE
IOM = 1000000
IDUMF1(281) = TIM
IOM = 1000000
IDUMF1(283) = PREF
IOM = 1000000
IDUMF1(284) = RPM
IDUMF1(290) = CNTRL(4)
IDUMF1(292) = CNTRL(6)
IDUMF1(293) = IAVRCE(1)
CALL TIME (IBUF1(352), IBUF1(360), IBUF1(360), IBUF1(376), IBUF1(384))

Display the just acquired wave on terminal CRT, if
CNTRL(40) is set to 1. The character used for the "plot"
is defined by CNTRL(249).

WRITE (LI, 148) IDUM
WRITE (LI, 149) ICLR
IF ( CNTRL(40) .NE. 1 ) GO TO 11
WRITE (LI, 150) LA, HOLP
I111 = 1
J222 = 1
CALL PICTR (LA, IDOC, J111, J222, CNTRL(249), DUM)

Select the next step:
RE  Repeat this point
anything else  Proceed to the next point
11 IF ( ISP .LT. 5 ) GO TO 12
ISP = 0
IF ( CNTRL(37) .NE. 1 ) WRITE (LO, 609)
ISP = ISP+1

Save data in file. There are two options. The raw data file
is either created/opened and closed by subroutine PACER
(CNTRL(39) is set to anything but to 1) or this subroutine
is called from subroutine ABSRU, which already has created/
opened and positioned the raw data file and will close it
(CNTRL(39) is set to 1). In both cases the raw data are
written in file by this subroutine PACER.

13 WRITE (LI, 168) 'FILE, NOLP'
FILE = ICOM(SECU(5))
FILE = ICOM(SECU(6))
IBUF1(282) = IFILE(1)
IBUF1(280) = IFILE(2)
IBUF1(281) = IFILE(3)
IBUF1(263) = JCR
IF ( CNTRL(39) .EQ. 1 ) GO TO 16
CALL CREAT (IDCB, IERR, IFILE, SIZE, ITYPE, SECUC, ICR, IDCBS)
WRITE (LI, 169) 'FILE'
READ (LI, 149) IDUM
WRITE (LI, 149) (ICLR, I=1, B, 1)
IF ( IDUM .EQ. 2HRE ) STOP 8402
IF ( IDUM .EQ. 2HRE ) GO TO 86
12 IF ( ISP .LT. 5 ) GO TO 12
ISP = 0
IF ( CNTRL(37) .NE. 1 ) WRITE (LO, 609)
ISP = ISP+1
14 CALL CODE
15 WRITE (NEW,111) IDUM
16 WRITE (NEW,111) IFILE,NEW,IFILE(2),IFILE(3)
17 GO TO 13
18 CALL OPEN (IDCB,IERR,IFILE,IPHTN,ISECU,ICR,IDCBS)
19 IF ( IERR .LT. 8 ) STOP 0403
20 IREC = 1
21 GO TO 17
22 IF ( ICHNL .EQ. 0 ) IREC = IREC+3
23 CALL WRITE (IDCB,IERR,IBUF1,IL,IREC)
24 IF (. LT. 0 ) STOP 0406
25 IF ( CTRL(39) .EQ. 1 ) GO TO 18
26 CALL CLOSE (IDCB,IERR,39)
27 IF ( IERR .LT. 0 ) STOP 0407
28 IRECU = IREC(ISCU,0)
29 JREC = IREC(ICR,0)
30 WRITE (LI,140) ICLR
31 STOP data acquisition loop.
32
33 WRITE (LO,610) (IBUF1(128),265,272,1),CIM,CYAM,PI,P23,P4,E,DE
34 IXIM,PREFR,FREQ,AVERAGE,IFILE,ISECU,ICR,IREC
35 STOP
36 Terminate subroutine; write accounting data back
37 into control array.
38
39 IF ( CTRL(37) .EQ. 1 ) GO TO 28
40 CALL TIME (IMON,IDAY,YEAR,IMON,IMIN)
41 WRITE (LO,611) IMON,IMIN
42 20 CTRL(37) = -4
43 CTRL(39) = 1
44 CTRL(50) = 4
45 CTRL(218) = IFACE
46 RETURN
47 END
48
49 COMMON
50*COMMON*000 0
51*COMMON*000 0

FTN4 COMPILER: HP92069-16092 REV. 1926 (790430)
** NO WARNINGS ** NO ERRORS ** PROGRAM = 02561 COMMON = 00000

132
5. **PROGRAM TXCO2**

5.1. **DESCRIPTION**

TXCO2 is a son program of the father program TXCOØ, by which it is scheduled if one of the following operations is desired:

5 - Radial survey using the combination probe

6 - Scan through all steady state data

When scheduled by TXCOØ, which suspends operation while the son program TXCO2 executes, the program TXCO2 reads the program control array from the disc, sets the HP interface bus and the measurement and control devices to remote control and programs the digital voltmeter (DVM), the scanners and the counter. CNTRL(50) is the actual decision variable to select and call the subroutine, which performs the desired operation. When this subroutine has terminated, the interface bus and the devices are released from remote control and the control array is written into a disc file, so that the next TXCO module can read it. The correct termination of each subroutine can be verified by checking the stop codes. Note that each stop code ending in 77 indicates correct execution of a subroutine.

<table>
<thead>
<tr>
<th>CNTRL(50)</th>
<th>Subroutine</th>
<th>STOP Code</th>
</tr>
</thead>
<tbody>
<tr>
<td>5</td>
<td>COMB</td>
<td>TXCO2 : STOP 0577</td>
</tr>
<tr>
<td>6</td>
<td>STDY</td>
<td>TXCO2 : STOP 0677</td>
</tr>
</tbody>
</table>

Any other STOP code indicates a mistake and with the help of a program list the operator can trace the problem. The first two digits of the STOP code identify the subroutine. An example: if the program stops at STOP code 0604, the first
two digits read 6 and tells the operator that it was subroutine STDY which encountered problems. The last two digits read 04 (no error would give 77). A program list reveals that the failure occurred after attempting to purge an existing data file using FMP (File Management Package) subroutine PURGE near line 752. Maybe the cartridge, where the raw data are directed, has not been mounted with the MC-command from FMGR. STOP codes are crucial to a complex program system in order to rapidly detect and salvage problems, even during a test run.

EXTERNALS: REWRF, ABERT, RMOTE, COMB, STDY, CLEAR, LOCL

COMMON BLOCKS: FMP, CIBUF, CONTR

FORTRAN conventions for the HP 21 MX computer request COMMON blocks to be predefined in a BLOCK DATA subroutine prior to using a COMMON block in a program, subroutine or function.

<table>
<thead>
<tr>
<th>BLOCK DATA Subroutine</th>
<th>Arrays &amp; Variables</th>
<th>Length in Words</th>
</tr>
</thead>
<tbody>
<tr>
<td>FMP</td>
<td>IDCB,IFILE,ISIZE,ISECIA,ICR</td>
<td>227B = 15110</td>
</tr>
<tr>
<td>CIBUF</td>
<td>IBUF</td>
<td>3200B = 1664</td>
</tr>
<tr>
<td>CONTR</td>
<td>CNTRL</td>
<td>400B = 256</td>
</tr>
</tbody>
</table>

The arrays and variables allocated by the COMMON block FMP are frequently used for the data transfer from and to the disc. COMMON block CIBUF is designed to take the largest raw data array in the TXCO data acquisition and reduction system - IBUF(1664) in subroutine FREER. The largest data array in TXCO2 is PDAT (24, 21) with 1008 = 2*21*24 words. The TXCO2 subroutines only partially use the COMMON area. The COMMON
block CONTR allocates the space for the control array CNTRL. Since each individual subroutine saves the data prior to terminating, the buffer area for the raw data can be shared by more than one subroutine or function.

MNEMONIC ABBREVIATIONS: None

ERROR MESSAGES: If CNTRL(50) is less than 5 or greater than 6, no subroutine can be selected and the program terminates outputting an error message (FORMAT 102) to the standard input device; i.e. the terminal.

PROCEDURE: For more detailed information study the flow chart and the information given in the subroutine descriptions.

DATAFILE: None

VARIABLES IN BLOCK DATA FMP:

- IDC (144) integer data control block
- IFILE (3) integer array to contain file name
- ISIZE (2) integer array to contain # of records in the first and record length in the second 16-bit word
- ISECU integer security code of data file
- ICR integer cartridge reference number, where data file is located
VARIABLES IN BLOCK DATA CIBUF:
IBUF (1664) integer buffer array for the raw data

VARIABLES IN BLOCK DATA CONTR:
CNTRL (256) integer program control array

VARIABLES IN PROGRAM TXCO2:
CNTRL (256) integer program control array
NOLF integer suppresses line feed
LI integer LU# of standard input device (terminal)
ISTOP integer control variable to select STOP code
3

Release interface
bus and devices
from remote
control.
CALL CLEAR (?)
CALL LOC ( ?)

4

Write program
control array
back into disc
file CTRLF.
CALL REWRF

STOP 05

STOP 06

Display error
message.
END
5.2. **SUBROUTINE COMB:**

**PURPOSE:** Acquisition of flow data from the transonic 1-stage axial compressor using a pneumatic 4-hole combination probe. The data necessary for the reduction procedure (PROGRAM REDCO: Reduce Combination probe data) are recorded also. Up to 24 different radial positions can be recorded. Taking more than one scan at one and the same radial position should be avoided, because the reduction program (originally written by R. Shreeve for the Laboratory's HP 9830 calculator and rewritten by F. Neuhoff for the more advanced HP 21 MX computer) is not set up for this condition.

**ARGUMENTS:** None

**EXTERNALS:** TIME, SCANR, ACQN, CREAT, PURGE, OPEN, WRITF, CLOSE

**COMMON BLOCKS:** CIBUF, CONTR, FMP. For detailed explanation refer to the TXCO2 description.

**MNEMONIC ABBREVIATIONS:**

RE ... Repeat data acquisition at this radial position.
NE ... Proceed to the next radial position.
EN ... End survey at this operating condition.
UP ... Update position readings of probes prior to data taking.
TA ... Initialisation command to take data.
TR ... Transfer raw data to HP 9830.
ST ... Store raw data in 21 MX disc file.
PU ... Allow purge of an existing data file.
ERROR MESSAGES: None

PROCEDURE: For more detailed information, study the flow chart. After having assigned the accounting data, assigned the I/O references, COMB asks the operator whether the radial survey takes place ahead of the (IPOS = 1) rotor or behind the (IPOS = 2) rotor. IPOS later will be used to identify the correct data port (see Appendix A.1: Data Locations). The raw data array IBUF - which is set equivalent to real array PDAT (Prob Data; used instead of IBUF, which is an integer array) is first preset with zeroes. Before the operator goes ahead and allows the subroutine to gather data (Input: TA; see key to raw data array), he can monitor the probe positions by updating its reading (Input: UP), until the probe is manually set to the desired position. Upon completion of the data scan the acquired data are printed and the next step depends on the operator's decision. If a preliminary check reveals erroneous data, the scan at this radial position should be repeated (Input: RE). If the data are correct, the operator either proceeds to the next radial position (Input: NE) or terminates the radial combination probe survey (Input: EN) at this operating condition. The subroutine then asks where to dump the data. When this routine was developed the data reduction program for the combination probe was not available in the 21 MX system, hence the option to transfer the data to the 9830 calculator (Input: TR) was used. But the data can as well be stored in a 21 MX disc file (Input: ST). If the raw data file with the automatically
determined name already exists, the operator either allows overwriting the existing file (Input: PU) or renames the current data file (Input: any alphabetic character other than T). The subroutine terminates printing the data file name at the bottom of the data documentation page.

**DATA FILE:** The default file name is T5rrss (rr ... ASCII converted run #; ss ... ASCII converted sequential #).

**VARIABLES:**

- **IBUF (1664)** integer buffer array
- **CNTRL (256)** integer control array
- **IDCB (144)** integer data control block, used for FMP calls
- **IFILE (3)** integer array to contain current file name
- **ISIZE (2)** integer specifies # of records and record length
- **ISECU** integer security code of data file
- **ICR** integer cartridge reference number, where the raw data file is located
- **JSECU** integer ASCII converted security code
- **JCR** integer ASCII converted cartridge reference number
- **PDAT(21,24)** real raw data array, set equivalent to IBUF
<table>
<thead>
<tr>
<th>Variable</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>POS (7)</td>
<td>real</td>
<td>array to contain probe positions</td>
</tr>
<tr>
<td>NOLF</td>
<td>integer</td>
<td>suppresses line feed</td>
</tr>
<tr>
<td>NOCR (2)</td>
<td>integer</td>
<td>suppresses line feed and carriage return</td>
</tr>
<tr>
<td>ICLR (3)</td>
<td>integer</td>
<td>clears line above cursor</td>
</tr>
<tr>
<td>IDCBS</td>
<td>integer</td>
<td>length of data control block</td>
</tr>
<tr>
<td>IENTR</td>
<td>integer</td>
<td>multiple entry flag</td>
</tr>
<tr>
<td>IDOC</td>
<td>integer</td>
<td>count of current program run</td>
</tr>
<tr>
<td>IDOCF</td>
<td>integer</td>
<td>count of current data file</td>
</tr>
<tr>
<td>ITYPE</td>
<td>integer</td>
<td>type of raw data file</td>
</tr>
<tr>
<td>IL</td>
<td>integer</td>
<td>number of words to be transferred in FMP calls</td>
</tr>
<tr>
<td>LI</td>
<td>integer</td>
<td>LU# of standard input device (terminal)</td>
</tr>
<tr>
<td>LO</td>
<td>integer</td>
<td>LU# of standard output device (line printer)</td>
</tr>
<tr>
<td>IPOS</td>
<td>integer</td>
<td>Flag to indicate measurement location</td>
</tr>
<tr>
<td>IMON</td>
<td>integer</td>
<td>ASCII converted month of current year</td>
</tr>
<tr>
<td>IYEAR</td>
<td>integer</td>
<td>ASCII converted last two digits of current year</td>
</tr>
<tr>
<td>IDAY</td>
<td>integer</td>
<td>ASCII converted day of the month</td>
</tr>
<tr>
<td>IHOUR</td>
<td>integer</td>
<td>ASCII converted hour of the day (24 h clock)</td>
</tr>
<tr>
<td>IMIN</td>
<td>integer</td>
<td>ASCII converted minute of the hour</td>
</tr>
<tr>
<td>Variable</td>
<td>Type</td>
<td>Description</td>
</tr>
<tr>
<td>----------</td>
<td>------</td>
<td>-------------</td>
</tr>
<tr>
<td>J1</td>
<td>integer</td>
<td>Subscript for data array PDAT</td>
</tr>
<tr>
<td>IS</td>
<td>integer</td>
<td>LU# of the selected scanner</td>
</tr>
<tr>
<td>IC</td>
<td>integer</td>
<td>Instrument code (DVM ... 1 and digital counter ... 3)</td>
</tr>
<tr>
<td>I2</td>
<td>integer</td>
<td>Subscript for position array POS</td>
</tr>
<tr>
<td>J3</td>
<td>integer</td>
<td>Contains channel of desired scanner</td>
</tr>
<tr>
<td>IDUM</td>
<td>integer</td>
<td>Decision variable</td>
</tr>
<tr>
<td>IW</td>
<td>integer</td>
<td>Determines delay in tens of milliseconds between closing S/V port and DVM reading</td>
</tr>
<tr>
<td>JO</td>
<td>integer</td>
<td>Number of selected S/V</td>
</tr>
<tr>
<td>SUM</td>
<td>real</td>
<td>Variable used to compute average</td>
</tr>
<tr>
<td>ISYNCH</td>
<td>integer</td>
<td>Synchronisation variable to coordinate data transfer</td>
</tr>
<tr>
<td>NEW</td>
<td>integer</td>
<td>Scratch variable used to rename files</td>
</tr>
</tbody>
</table>

Key to data array PDAT

- PDAT (1,J1) Barometric pressure $\bar{\text{BARO}}$
- PDAT (2,J1) KULITE reference pressure
- PDAT (3,J1) Combination probe pressure $p_1$
- PDAT (4,J1) Combination probe pressure $p_{23}$
- PDAT (5,J1) Combination probe pressure $p_4$
- PDAT (6,J1) Total pressure ahead of compressor $p_c$
- PDAT (7,J1) Static port in casing #2, $S_2$

144
PDAT (8,J1)  Static port in casing #13, $S_{13}$
PDAT (9,J1)  Average reading of 4 static ports in Hub
             (#2 thru 5) $(H_1 + H_2 + H_3 + H_4)/4$
PDAT (10,J1) Pressure ahead of compressor flow meter
              orifice $P_1$ nozzle compr
PDAT (11,J1) Temperature ahead of compressor flow meter
              orifice $T_1$ nozzle compr
PDAT (12,J1) Pressure drop across compressor flow meter
              orifice $P_{\text{nozzle compr}}$
PDAT (13,J1) Temperature reading from reference probe $T_{\text{ref}}$
PDAT (14,J1) Differential temperature reading from the
              combination probe to the reference probe $T_{\text{probe}}$
PDAT (15,J1) Radial immersion of the combination probe
PDAT (16,J1) Yaw angle of the combination probe
PDAT (17,J1) Case angle
PDAT (18,J1) Compressor RPM
PDAT (19,J1) Test run #
PDAT (20,J1) Test # of this run
PDAT (21,J1) Point # of this test

$J_1 = 1 \ldots 24$ indicates # of radial position.
Program runs on
L1

1. Set position index 2 in data array 
   MAT[24,2] to zero.

2. Set time and print 
   heading of data 
   documentation page.

3. Output L1 to L1A.

4. Output L2 to L1A, 
   counter on L1.

5. If POS < 1, 
   go to 2, else 
   go to 6.

6. If POS > 2, 
   go to 7, else 
   go to 8.

7. Reset raw data 
   array with zeroes.

8. If loop completed, 
   go to 5.


10. Output L1 to L1A, 
    counter on L1.

11. If POS < 1, 
    go to 2, else 
    go to 6.

12. If POS > 2, 
    go to 7, else 
    go to 8.
3. Refer to Appendix A1: worksheet data locations. The location of immersion and jaw angle for the combination probe depends on where to take the (Cycl. 148
4

5

6

7

=IHM
=IHM
=IHM

Compare IDUM

4911

Take data via

S/V = 1.

Take data via

S/V # 2.

Repeat the
data scan
at this radial
position.

Proceed
to the
next radial
position.

9

10

Take data via

scanner #1(25-6)

Take data via

scanner #2 (35-6)

Write accounting
date and case
angle (POST #1)
to data array
and print all data
from this scan.
ASK OPERATOR TO DECIDE ON THE NEXT STEP; READ THE DECISION VARIABLE IDUM

ASK OPERATOR, WHO THEN TO TRANSFER THE DATA TO HP38C (TR) OR SAVE THEM IN 21RR DATA FILE (ST); READ DECISION VARIABLE IDUM.
13

Wait for 3820 change procedure to be completed; enter RETURN to continue.

Print message, that data transfer to DP 3820 has been completed.

14

Open data file
CALL OPEN

IF ERR < 0 THEN
WRITE DATA IN FILE
CALL WRITE

IF ERR < 0 THEN
CLOSE DATA FILE
CALL CLOSE

IF ERR < 0 THEN
STOP

STOP
5.3. **SUBROUTINE STDY:**

**PURPOSE:** Acquisition of flow data from the transonic 1-stage axial compressor using the steady state instrumentation. All data, gathered by this subroutine, will be used for the reduction (PROGRAM REDST: Reduce Steady state data; see section 8.3 of this report). The data array is designed to both resemble the data source location matrix (see Appendix A.1) and to discriminate groups of similar data by blank lines (Appendix A.2). CH3(1) through CH3(5), which contain all the pressures needed to calculate the flow rates, and CH3(6), which is left blank, separates this group of data from the next one. The reason is to allow the investigator a quick check and verification of crucial data. The "units" of the readings depend on the amplifier settings, but usually each channel is calibrated to allow the operator to read voltages as a quantity in engineering units. As far as possible, amplifier drift is traced by the program (CH1(1), CH2(1) etc.). (The author is indebted to Laboratory's manager, Mr. Jim Hammer, who, with admirable patience, instructed the author in how best to handle data and data systems.)

**ARGUMENTS:** IRUN; if IRUN is set to 0 (zero), taking pressure readings from the Scanivalves (S/V) is skipped. This option was needed when the subroutine was first written so that frequent debugging runs did not put additional loads on the S/V's. The standard entry is: IRUN = 1. Only then will the reduction program REDST perform correctly.
EXTERNALS:  TIME, ACQN, SCANR, CREAT, PURGE, OPEN, WRITF, CLOSE

COMMON BLOCKS: CIBUF, CONTR, FMP. For detailed explanation refer to the TXCO2 description

MNEMONIC ABBREVIATIONS:
RP ... Repeat data acquisition at this operating condition
RT ... End data acquisition and return to calling program
PU ... Allow purge of an existing data file

ERROR MESSAGES:  None

PROCEDURE: For more detailed information, study the flow chart.
After having assigned the accounting data and defined the I/O references, STDY presets all elements of the raw data array with -0.999999, which definitely never will occur as a data reading. Next, unless IRUN equals 0 (zero), the pressures on Scanivalves 1 and 4 are read. The voltages from scanner #1 (LU# = 8) and scanner #2 (LU# 15) are read next, not depending on the value of IRUN. Note, that the allocation for the voltages in the raw data array provides blocks of similar data, separated by blank lines. A set of control parameters (CNTRL (1) thru CNTRL (6); and CNTRL (15) and the case angle - which needs to be put in manually - completes the steady state data. Then the raw data is printed to allow the operator to look at and to verify the newly acquired data. The data scan can either be repeated (Input: RP) or the subroutine terminates (Input: ST) storing the data in a type 1 disc file. If the automatically determined name for the data file already exists,
the operator either allows overwriting the existing file (Input: PU) or renames the current data file (Input: any alphabetic character other than T). Finally data file name and time are printed at the bottom of the data documentation page.

DATA FILE: The default file name is T4rrss (rr ... ASCII converted run #; ss ... ASCII converted sequential #); see Appendix A.2: Steady State Data Array. CH4 (1) through CH4 (26) are not used, because the reduction program will write its results into these slots.

VARIABLES:

<table>
<thead>
<tr>
<th>Variable</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>IBUF</td>
<td>integer</td>
<td>buffer array</td>
</tr>
<tr>
<td>CNTRL</td>
<td>integer</td>
<td>control array</td>
</tr>
<tr>
<td>IDCBO</td>
<td>integer</td>
<td>data control block, used for FMP calls</td>
</tr>
<tr>
<td>IFILE</td>
<td>integer</td>
<td>array to contain current file name</td>
</tr>
<tr>
<td>ISIZE</td>
<td>integer</td>
<td>specifies # of records and record length</td>
</tr>
<tr>
<td>ISECU</td>
<td>integer</td>
<td>security code of data file</td>
</tr>
<tr>
<td>ICR</td>
<td>integer</td>
<td>cartridge reference number, where the raw data file is located</td>
</tr>
<tr>
<td>JSECU</td>
<td>integer</td>
<td>ASCII converted security code</td>
</tr>
<tr>
<td>JCR</td>
<td>integer</td>
<td>ASCII converted cartridge reference number</td>
</tr>
<tr>
<td>DATA</td>
<td>real</td>
<td>raw data array, set equivalent to IBUF</td>
</tr>
</tbody>
</table>

156
<table>
<thead>
<tr>
<th>Variable</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>NOLF</td>
<td>integer</td>
<td>suppresses line feed</td>
</tr>
<tr>
<td>NOCR (2)</td>
<td>integer</td>
<td>suppresses line feed and carriage RETURN</td>
</tr>
<tr>
<td>ICLR (3)</td>
<td>integer</td>
<td>clears line above cursor</td>
</tr>
<tr>
<td>IENTR</td>
<td>integer</td>
<td>multiple entry flag</td>
</tr>
<tr>
<td>IDOC</td>
<td>integer</td>
<td>count of current program run</td>
</tr>
<tr>
<td>IDOCF</td>
<td>integer</td>
<td>count of current data file sequential #</td>
</tr>
<tr>
<td>ITYPE</td>
<td>integer</td>
<td>type of raw data file</td>
</tr>
<tr>
<td>IL</td>
<td>integer</td>
<td>number of words to be transferred in FMP calls</td>
</tr>
<tr>
<td>IFRST</td>
<td>integer</td>
<td>temporary buffer variable</td>
</tr>
<tr>
<td>LI</td>
<td>integer</td>
<td>LU # of standard input device (terminal)</td>
</tr>
<tr>
<td>LO</td>
<td>integer</td>
<td>LU # of standard output device (line printer)</td>
</tr>
<tr>
<td>J1</td>
<td>integer</td>
<td>subscript for data array DATA, specifies channel</td>
</tr>
<tr>
<td>J2</td>
<td>integer</td>
<td>subscript for data array DATA</td>
</tr>
<tr>
<td>IRUN</td>
<td>integer</td>
<td>control variable</td>
</tr>
<tr>
<td>JO</td>
<td>integer</td>
<td>number of selected S/V</td>
</tr>
<tr>
<td>IW</td>
<td>integer</td>
<td>determines delay in tens of milliseconds between closing S/V port and DVM reading</td>
</tr>
<tr>
<td>TAREl</td>
<td>real</td>
<td>drift of amplifier S/V#1 during test run</td>
</tr>
</tbody>
</table>
TARE2   real     drift of amplifier S/V #4
during test run

IS      integer   LU # of the selected scanner

IC      integer   instrument code (DVM ... 1
and digital counter ... 2)

J3      integer   contains channel of scanner

NO(2)   integer   ASCII text to be printed, if
value of DATA (J2,J1) =
                 -.999899

IDUM    integer   decision variable

IMON    integer   ASCII converted month of current
                 year

IDAY    integer   ASCII converted day of the month

IYEAR   integer   ASCII converted last two digits
                 of current year

IHOUR   integer   ASCII converted hour of the day
                 (24 h clock)

IMIN    integer   ASCII converted minute of the
                 hour

NEW     integer   scratch variable used to
               rename files

158
FLOW CHART SUBROUTINE STOY

ENTRY

Is the subroutine entered for first time & (time < 01)?

NO

YES

Set the entry flag and define parameters; set count variables IDOC & ZDOC to zero.

Define the rest of the parameters in program JXXX (m) & counter (16); get date & time and print header of data documentation page; increase sequential number of count variables.
Print bottom of the data documentation page.

Read IRUN.

Ask operator, whether to perform a short (IRUN=0) or long (IRUN=1) pass.

Program JVM.

Do 06 J=1,40,1

Do 06 JX=1,40,1

Preset data array DATA (40, 4).

IRUN = 0?

Do have completed?

Do 06 IRUN=0

3

4
Aggure pressure readings from the SV - SV-counter - Store - 3864 data chain.

SV #1 (J = 1)
- raw data
- new channel 1 (J = 1)
- Define IV

If \( J = 1 \) then

- Display all data & parameters.

- CO loop completed?
  - No
  - YES

Else

- Continue

Continue
DO 10 J=1,6,1

Data(30,30) = CONV(30,30)

Display all data of parameters.

00 loop completed?

YET

Subtract bare from pressure readings in order to compensate amplifier drift during test run.

100
Acquire data (pressure, temperature, measured force, speed, probe positions) from scanner-ath system (IC=1) and scanner - counter system (IC=2). Since groups of similar data are processed by a block here, this flow chart may contain errors from mapping all the logical steps. Total the new data before channel 3 (B=2) is filled.

Fill up row data array channel 4 (B=4) with data from control variable list or read from the control array, CTRL; manually input data into each row of MAR(4), (0).

If MAR(4,30) = "STOP", STOP 0001
Print, acquired data,
the printout is
spaced into blocks
of five; if the
particular data
point is -39883,
n/a is printed to
indicate that this
point is not being
used.

Get date & time
and define FMP
parameters JSEC
& JCR.

IDOCF < 100

No

Yes

Change IFAST
from 7A (standard)
to 5A, subtract
100 from IDOCF.
JFAST = 2H54
IDOCF = IDOCF - 100
Define name of the raw data file (DPFILE).

Ask operator whether to repeat this point or terminate the subroutine and return control to the calling program.

Read and check IOM4.

If IOM4 = 2, IOM7 = IOM4.

If IOM4 = 5 or IOM4 = 6, STOP 0604.

If IOM4 = 6, STOP 0608.

Store acquired data in a type 1 disc file.

IERR < 0

14

15

16
Create data file
CALL CREAT

IF ERR > 0
   NO
   IF study
      1st file
      READ 1
      READ 2
   THEN
      STOP 0602
      36
   ELSE
      37
      OPEN data file
      CALL OPEN
   ENDIF

Purge file with the already existing file name.
CALL PURGE
- IERR < 0
  - STOP 0606

- Write data array DATA completely into data file
  - CALL WRITE

- IERR < 0
  - STOP 0606

- Close data file
  - CALL CLOSE

- IERR < 0
  - STOP 0607

- 18
5.4. PROGRAM LISTING TXCO2

PAGE 0001 FTN. 4:12 PM TUE., 23 SEP., 1980

0001 FTN4,L
0002 BLOCK DATA
0003 $/FMP$/IDCB(144),IFILE(3),ISIZE(2),ISECU,ICR
0004 COMMON/FMP$/IDCB,IFILE,ISIZE,ISECU,ICR
0005 INTEGER(IDCB(144),IFILE(3),ISIZE(2)
0006 END

FTN4 COMPILER: HP92060-16092 REV. 1926 (790430)
** NO WARNINGS ** NO ERRORS **
BLOCK COMMON FMP SIZE = 001S1

169
0007    BLOCK DATA
0008    Z/CIBUF/IUF(1664)
0009    COMMON /CIBUF /IBUF
0010    INTEGER IBUF(1664)
0011    END

FTN4 COMPILER: HP92060-16092 REV. 1926 (790439)
** NO WARNINGS ** NO ERRORS **
BLOCK COMMON CIBUF SIZE = 01664
BLOCK DATA
& / CONTR / CNTRL(256)
COMMON / CONTR / CNTRL
INTEGER CNTRL(256)
END
PROGRAM TXCO2 (3,99)

The operating system RTE IV B requests the data acquisition program TXCO for the one stage transonic compressor to be split into several programs scheduled by the father program TXCO. This son program TXCO2 consists of the subroutines COMB and STDY. These codes handle the acquisition of the steady state data and a survey conducted with the combination probe. The data transfer between father and son program takes place via the control array CNTRL (disc file CNTRLF) and the data array IBUF (disc file IBUFF).

The utility subroutines ACQ1N, CTUR, CURVE, ICON, DPORT, PICTR, REWRF, RFACE, SCAN, TIME and WAIT are added.

Author: Hans M. Zebner

Date: March 12, 1980

A detailed program description is available in the TXCO log.

COMMON / CONTR / CNTRL

INTEGER CNTRL(256)

DATA MOLF /046537/

101 FORMAT (9X**20X**A2)
102 FORMAT (** TXCO2 PROGRAM ABORTED! NO SUBROUTINE HAS BEEN INITIALIZED.**)
001 FORMAT (**CA**)
1001 FORMAT (**ERRNO1403**)
1201 FORMAT (**PFC01**)
1501 FORMAT (**CA**)

CALL REWRF (1,2)
CALL CNTRL(1)
IF ( CNTRL(SO) .LT. 5 .OR. CNTRL(SO) .GT. 6 ) GO TO 03

CALL CMON (2,3)
CALL CMON (8)
CALL CMON (10)
CALL CMON (12)
CALL CMON (14)

CALL WRITE (10,1001)
CALL WRITE (12,1001)
CALL WRITE (15,1001)

Call subroutine indicated by CNTRL(SO).

IFSTOP = CNTRL(SO)
IF ( CNTRL(SO) .EQ. 5 ) CALL COMB
IF ( CNTRL(SO) .EQ. 6 ) CALL STDY(CNTRL(S1))

Release interface bus and devices from remote control.

CALL CLEAR (7,1)
CALL LOC3L (7)
CALL REWRF (1,2)
WRITE (LI, 101) NOLF
GO TO (01, 02) ISTOP
STOP 0577
END

FTN4 COMPILER: HP92060-16092 REV. 1926 (790430)
** NO WARNINGS ** NO ERRORS ** PROGRAM = 00221 COMMON = 00000
SUBROUTINE COMB

Accepts data from the transonic 1-stage axial compressor.
Gathers data required for a probe survey with the
combination probe. The raw data then are transferred to
the HP 9830 calculator for reduction.

017 COMMON / CIBUF / IBUF
018 COMMON / CONTR / CTNRL
019 COMMON / FDP / IDCBS,IFILE,ISIZE,ISBIC,ICR
020 REAL PDAT(29,24),POS(3)
021 INTEGER CNTRL(256),IDCB(144),IFILE(3),ISIZE(2)
022 INTEGER IBUF(164)
023 INTEGER NOLF,NOCR(2),ICLR(3)
024 EQUIVALENCE (IBUF(1),PDAT(I,1))
025 DATA NOLF /006537B/
026 DATA NOCR /006533B,040433B/
027 DATA ICLR /015524B,015515B,006537B/
028 DATA IDCBS /144/

C FORMATS

100 FORMAT (A15)
101 FORMAT ("WARNING: file "3A2" already exists! Type PU to "
102 FORMAT ("...leave purge or enter any char".),")
103 FORMAT ("Filename change successful "3A2" changed to "3A2")
104 FORMAT ("26X* Gathering probe data "26X"A2")
105 FORMAT ("26X* Storing data in file "3A2,26X"A2")
106 FORMAT (A2)

210 FORMAT (" Make sure that the 9830 receiver program runs! Type 
211 FORMAT ("YES to continue! "A2)
212 FORMAT ("79X/2A2" Data transfer completed. Print transferr 
213 FORMAT ("ed data? Enter YES or NO! "3A2)
214 FORMAT (" Repeat data transfer? Enter YES or NO! 
215 FORMAT (" Waiting for 9830 storage procedure. Type C 
216 FORMAT (" to continue! "3A2)
217 FORMAT (" Check synchronisation of master and slave program! 
218 FORMAT (" Type N to continue! "3A2)
219 FORMAT ("10X Probe survey at this constellation completed"17 
220 FORMAT ("3X="A2/
221 FORMAT (" Type TB to transfer the data to HP 9830 calculator"/
222 FORMAT (" Type TA to terminate the survey at this constellation"/
223 FORMAT ("26X"Check raw data of this scan! "26X"A2/
224 FORMAT (" Type NE to repeat this point! "26X"A2/
225 FORMAT (" Type NG to proceed to the next point! "26X"A2/
226 FORMAT (" Type TM to terminate the survey at this constellation"/
227 FORMAT ("
228 FORMAT ("26X"Combination "3X"Yaw Angle"
229 FORMAT ("24X"Inches "3X"
230 FORMAT (" Type "A" probe "F10.3","A" probe "F10.3"
231 FORMAT (" Type "B" probe "F10.3","B" probe "F10.3"
232 FORMAT (" Type UP to update those readings! "26X"
233 FORMAT (" Type TA to take a data set at this constellation"/
234 FORMAT (" Enter case angle ="34X,2A2)
235 FORMAT (" in this combination probe survey done before (", 
236 FORMAT (" after (2) the nof Rrer? "2A2)
237 FORMAT (" PDAT(I2,I2) = ACQN("I2","I2","I3") has been e 
238 FORMAT (" selected result is "F10.6"A2)
239 FORMAT (" PDAT(I2,I2) = SCANR("I2","I2","I1") has been exe 
240 FORMAT ("ed result is "F10.6"A2)
241 FORMAT (" 
242 FORMAT (" 
243 FORMAT (" 
244 FORMAT (" 
245 FORMAT (" 
246 FORMAT (" 
247 FORMAT (" 
248 FORMAT (" 
249 FORMAT (" 
250 FORMAT (" 
251 FORMAT (" 
252 FORMAT (" 
253 FORMAT (" 
254 FORMAT (" 
255 FORMAT (" 
256 FORMAT (" 
257 FORMAT (" 
258 FORMAT (" 
259 FORMAT (" 
260 FORMAT (" 
261 FORMAT (" 
262 FORMAT (" 
263 FORMAT (" 
264 FORMAT (" 
265 FORMAT (" 
266 FORMAT (" 
267 FORMAT (" 
268 FORMAT (" 
269 FORMAT (" 
270 FORMAT (" 
271 FORMAT (" 
272 FORMAT (" 
273 FORMAT (" 
274 FORMAT (" 
275 FORMAT (" 
276 FORMAT (" 
277 FORMAT (" 
278 FORMAT (" 
279 FORMAT (" 
280 FORMAT (" 
281 FORMAT (" 
282 FORMAT (" 
283 FORMAT (" 
284 FORMAT (" 
285 FORMAT (" 
286 FORMAT (" 
287 FORMAT (" 
288 FORMAT (" 
289 FORMAT (" 
290 FORMAT (" 
291 FORMAT (" 
292 FORMAT (" 
293 FORMAT (" 
294 FORMAT (" 
295 FORMAT (" 
296 FORMAT (" 
297 FORMAT (" 
298 FORMAT (" 
299 FORMAT (" 
300 FORMAT (" 
301 FORMAT (" 
302 FORMAT (" 
303 FORMAT (" 
304 FORMAT (" 
305 FORMAT (" 
306 FORMAT (" 
307 FORMAT (" 
308 FORMAT (" 
309 FORMAT (" 
310 FORMAT (" 
311 FORMAT (" 
312 FORMAT (" 
313 FORMAT (" 
314 FORMAT (" 
315 FORMAT (" 
316 FORMAT (" 
317 FORMAT (" 
318 FORMAT (" 
319 FORMAT (" 
320 FORMAT (" 
321 FORMAT (" 
322 FORMAT (" 
323 FORMAT (" 
324 FORMAT (" 
325 FORMAT (" 
326 FORMAT (" 
327 FORMAT (" 
328 FORMAT (" 
329 FORMAT (" 
330 FORMAT (" 
331 FORMAT (" 
332 FORMAT (" 
333 FORMAT (" 
334 FORMAT (" 
335 FORMAT (" 
336 FORMAT (" 
337 FORMAT (" 
338 FORMAT (" 
339 FORMA
FORMAT (IN 4 (IN 4)},33 Transonic Compressor Test Run #17)

FORMAT (IN (IN 4 (IN 4) 6) Date: A12 (A1/4A12/1A2)

FORMAT (IN (IN 4 (IN 4) 6) Time: A12 (A1/4A12/1A2, 3H h, )

FORMAT (IN (IN 4 (IN 4) 6) "Constellation "13","/1H .9F10.6/

FORMAT (IN (IN 4 (IN 4) 6) Date transferred to HP 9830 file "10x")

FORMAT (IN (IN 4 (IN 4) 6) "Data saved in file "3A2"; "A2"; "A2")

FORMAT (IN (IN 4 (IN 4) 6) "")
IF (IPOS .EQ. 2) IS = 30
IF (IPOS .EQ. 1) IS = 38
POS(IS) = SCANR(IS, J3, IC)

IF (IPOS .EQ. 2) IS = 31
IF (IPOS .EQ. 1) IS = 39
POS(IS) = SCANR(IS, J3, IC)

DO 24 J3 = 32, 35, 1
POS(IS) = SCANR(IS, J3, IC)

21 TE = 21
DO 22 TE = 1, 5, 2
POS(IS) = POS(IS) * 1000.
DO 23 TE = 2, 5, 2
POS(IS) = POS(IS) * 10000.
WRITE (LI, 119) NOCR
READ (LI, 149) ICLR
WRITE (LI, 149) (ICLR, I = 1, 11)
IF (IDUM .EQ. 2HUP) GO TO 4711
WRITE (LI, 119) POS(7), NOCR
READ (LI, 149) IDUM
WRITE (LI, 149) (ICLR, I = 1, 11)
Gather data recorded via S/V#1 (J0=1).

J0 = 1  
J2 = 6

PDAT(J2, J1) = ACQN(J0, J3, IW)
WRITE (LI, 121) J2, J1, J0, J3, IW, PDAT(J2, J1), NOLF
J3 = 9
J0 = 7

PDAT(J2, J1) = ACQN(J0, J3, IW)
WRITE (LI, 121) J2, J1, J0, J3, IW, PDAT(J2, J1), NOLF

DO 1 J3 = 30, 33
J2 = 9

PDAT(J2, J1) = ACQN(J0, J3, IW)
WRITE (LI, 121) J2, J1, J0, J3, IW, PDAT(J2, J1), NOLF

1 SUM = SUM + PDAT(J2, J1)
PDAT(J2, J1) = SUM/4.

Gather data recorded via S/V#4 (J0=4).

J0 = 4
IF ( IPOS .EQ. 2 ) J3 = 3
J2 = 3

PDAT(J2, J1) = ACQN(J0, J3, IW)
WRITE (LI, 121) J2, J1, J0, J3, IW, PDAT(J2, J1), NOLF
IF ( IPOS .EQ. 1 ) J3 = 9
J2 = 4

PDAT(J2, J1) = ACQN(J0, J3, IW)
WRITE (LI, 121) J2, J1, J0, J3, IW, PDAT(J2, J1), NOLF
IF ( IPOS .EQ. 2 ) J3 = 9
J2 = 5

PDAT(J2, J1) = ACQN(J0, J3, IW)
WRITE (LI, 121) J2, J1, J0, J3, IW, PDAT(J2, J1), NOLF

Gather data recorded via scanner#1 (IW=8).

IC = 2
J3 = 17
J2 = 18

PDAT(J2, J1) = SCANNR(IS, J3, IC)
WRITE (LI, 122) J2, J1, IS, J3, IC, PDAT(J2, J1), NOLF
IC = 1
J3 = 25
J2 = 1

PDAT(J2, J1) = SCANNR(IS, J3, IC)
WRITE (LI, 122) J2, J1, IS, J3, IC, PDAT(J2, J1), NOLF
J3 = 26
J2 = 13

PDAT(J2, J1) = SCANNR(IS, J3, IC)
WRITE (LI, 122) J2, J1, IS, J3, IC, PDAT(J2, J1), NOLF
J3 = 27
J2 = 2

PDAT(J2, J1) = SCANNR(IS, J3, IC)
WRITE (LI, 122) J2, J1, IS, J3, IC, PDAT(J2, J1), NOLF
IF ( IPOS .EQ. 1 ) J3 = 30
J2 = 15

PDAT(J2, J1) = SCANNR(IS, J3, IC)
WRITE (LI, 122) J2, J1, IS, J3, IC, PDAT(J2, J1), NOLF
IF ( IPOS .EQ. 1 ) J3 = 36
J2 = 19

PDAT(J2, J1) = SCANNR(IS, J3, IC)
WRITE (LI, 122) J2, J1, IS, J3, IC, PDAT(J2, J1), NOLF
IF ( IPOS .EQ. 1 ) J3 = 44

177
J2 = 16
PDAT(J2,J1) = SCNR(IS,J3,IC)
WRITE (L1,122) J2,J1,IS,J3,IC,PDAT(J2,J1),NOLF
J3 = 37
J2 = 37
PDAT(J2,J1) = SCNR(IS,J3,IC)
WRITE (L1,122) J2,J1,IS,J3,IC,PDAT(J2,J1),NOLF
J3 = 5
DO 2 J3=4,5
PDAT(J2,J1) = SCNR(IS,J3,JC)
WRITE (L1,122) J2,J1,IS,J3,IC,PDAT(J2,J1),NOLF
2 SUM = SUM+PDAT(J2,J1)
PDAT(J2,J1) = SUM/2.
J3 = 18
J2 = J3
PDAT(J2,J1) = SCNR(IS,J3,IC)
WRITE (L1,122) J2,J1,IS,J3,IC,PDAT(J2,J1),NOLF
J3 = 19
J2 = 14
PDAT(J2,J1) = SCNR(IS,J3,IC)
WRITE (L1,122) J2,J1,IS,J3,IC,PDAT(J2,J1),NOLF
WRITE (L1,122) J2,J1,IS,J3,IC,PDAT(J2,J1),NOLF
1 Gather data recorded via scanner (IS=15).

DO 5 IJ=1,14
PDAT(19, IJ) = CNTRL(I)
PDAT(20, IJ) = CNTRL(S)
PDAT(21, IJ) = CNTRL(6)
PDAT(22, IJ) = POS(7)
WRITE (L1,119) IJ,PDAT(I,J1),I=1,21
50 WRITE (L1,119) NOLF,NOCR
WRITE (L1,149) (ICLR,I=1,6)
IF (IDUM .EQ. 'SHRE') GO TO 471
IF (IDUM .EQ. 'HARE') GO TO 56
IF (IDUM .EQ. 'HEN') GO TO 51
GO TO 50
51 IFILE(1) = 2HTS
IFILE(2) = ICON(CNTRL(4),0)
IFILE(3) = ICON(IDOCF,0)
IFSEC = ICON(ISECU,0)
ICR = ICON(ICR,0)
WRITE (L1,119) NOLF,IFILE,IFSEC,ICR,NOCR
READ (L1,149) IDUM
WRITE (L1,149) (ICLR,I=1,5)
IF (IDUM .EQ. 'SHRE') GO TO 52
IF (IDUM .EQ. 'HARE') GO TO 53
GO TO 51
52 ISYNCH = 9830

Data transfer to HP 9830 for reduction. No storage on 21MX!

71 WRITE (L1,210) NOLF
READ (L1,149) IDUM
WRITE (L1,149) ICLR
IF (IDUM .NE. 'SHRE') GO TO 71
WRITE (L7,201) ISYNCH
WRITE (L1,210) NOLF
READ (L1,149) IDUM
WRITE (L1,149) ICLR
DO 72 J=1,24
WRITE (L0,189) (PDAT(I,J),I=1,21),J=1,24
72

178
**Save data on 21MX disc. No transfer to HP 9830.**

```fortran
53 CONTINUE

418 WRITE (LI,105) (IFILE(J2),J2=1,3),NOLF
   CALL CREATE (IDCB,IERR,IFILE,J2=1,3),TYPE,ISECU,ICR,IDCBS)
   IF (IERR .LT. 0) GO TO 420

419 WRITE (LI,104) IDUM
   WRITE (LI,149) ICLR
   IF (IDUM .NE. 2HPU) GO TO 419

420 CALL CODE
   WRITE (NEW,100) IDUM
   WRITE (LI,102) (IFILE(J2),J2=1,3),NEW,(IFILE(J2),J2=2,3)
   IFILE(1) = NEW
   GO TO 418

425 CALL OPEN (IDCB,IERR,IFILE,IOPTN,ISECU,ICR,IDCBS)
   IF (IERR .LT. 0) STOP 16
   CALL WRITE (IDCB,IERR,IFILE,1)
   IF (IERR .LT. 0) STOP 17
   CALL CLOSE (IDCB,IERR,0)
   IF (IERR .LT. 0) STOP 20

432 WRITE (LO,607) IFILE,ISECU,ICR

433 CALL TIME (IMON,IDAY,IFILE,ICR,ICR,IMIN)
   WRITE (LO,619) IMON,IMIN

END
```

**FTN4 COMPILER: HP92060-16092 REV. 1926 (790430)**

**NO WARNINGS ** NO ERRORS ** PROGRAM = 03462 ** COMMON = 00000

179
SUBROUTINE STDY (IRUN)

Acquires data from the transonic 1-stage axial compressor.

This subroutine takes care of the steady state data, either
in a long or a short run (IRUN=1 resp. 0).

* Takes data from steady state system.

COMMON /CBUF/ IBUF
COMMON /CNTR/ CNTRL
COMMON /FHP/ IDC8,IFILE,ISIZE,ISECU,ICR
INTEGER IDCC(144),IFILE(3),1812ET2)
INTEGER CNTRL(256)
REAL DATA(48,4)
EQUIVALENCE (IBUF(1),DATA(i,i))
DATA NOLF /006537B/006537B/
DATA ICLR /0155248,0155158,006537B/006537B/
COMMON /IDCB/ INOFL /006537B/

FORMAT STDY

STUDY START

100 FORMAT (I4,A4)
101 FORMAT (*.WARNING: File "3A2" already exists! Try 
*to allow purge or enter any char")
102 FORMAT (*. STUDY : File name "3A2" successfully changed to "3A2")
103 FORMAT (*) STUDY : PU,"3A2":"A2".
104 FORMAT (I2)
105 FORMAT (*.26X*Storing data in file "3A2",26X"A2")
106 FORMAT (* Enter case angle : "A1","A2")
110 FORMAT (*) Look at the just acquired data! Decide and enter"
111 FORMAT (*) Do you want to repeat the data acquisition"
112 FORMAT (*.Do you want"
*1 TN long run"","A2")
115 FORMAT (I4,F6.2)
117 FORMAT (*. DATA("12","11") = ACQN("11","12","13") 
*has been exec 
*ted result is F10.6"A2")
121 FORMAT (*.12A1,7A2) = SCANR("12","13","11") 
*has been exec 
*ted result is F10.6"A2")
127 FORMAT (I1)
130 FORMAT (1362)
601 FORMAT (1H***.//36X 
*steady state data documentation P
602 FORMAT (1H***.//36X 
*Transonic compressor Test Run 0,17)
603 FORMAT (1H***.//36X 
*ShData : A2,1H,A2,1H,A2,1H
604 FORMAT (1H***.//36X 
*8Time : A2,1H,A2,1H,h,//////)
605 FORMAT (1H***.//36X 
*6Time : A2,1H,A2,1H,A2,1H,1H)
606 FORMAT (1H***.//36X 
*7Time : A2,1H,A2,1H)
607 FORMAT (1H***.//36X 
*8Time : A2,1H,A2,1H)
608 FORMAT (1H***.//36X 
*9Time : A2,1H,A2,1H)
609 FORMAT (1H***.//36X 
*10Time : A2,1H,A2,1H)
610 FORMAT (1H***.//36X 
*Long run performed. Above printed data are sav 
*ed in file "3A1","A2","A2". ... no data acquired.")
611 FORMAT (1H***.//36X 
*Short run performed. Above printed data are sa 
*ved in file "3A2","A2","A2". ... no data acquired.")
612 FORMAT (1H***.//36X 
*by A2,1H,A2,1H,1H,h,//////
0499 C FORMAT ("A REGISTER IS "K6"
0500 " B REGISTER IS "K6/")
0501 1001 FORMAT ("PF4G&T")
0502 1291 FORMAT ("PF4G&T")
0505 C FORMATS ST Dy STOP
0506 IF (IENTr .NE. 0 ) GO TO 1
0507 IENTr = 1
0508 I DenC = 0
0509 ISHC F = 0
0510 ISHCU = CNTRL(31)
0511 IL = CNTRL(30)
0512 I TYPE = 1
0513 I R = 384
0514 I DCBS = 144

181
Preset data array DATA(48,4) with the dummy variable -999999 to make trouble shooting easier.

DO 06 I2=1,46
IF (IRIT.EQ.0) GO TO 11

Acquire pressures from cannival-scanner-DVM system. Only performs in a long run.

DO 07 J1=1,40,1
IF ( IRIT.EQ.0 ) GO TO 11

Subtract tare from pressure readings.

Acquire data (pressures, temperatures, torque, speed, probe positions) from scanner-DVM or scanner-counter system.

PAGE 0012  STDY  4:12 PM TUE., 23 SEP., 1990

01 IBSIZE(3) = 3
03 WRITE (10,1001)
05 WRITE (12,1201)
07 IFRTY = 24
09 IRTY = CRTNL(19)
11 CRTL(20)
13 CALL TIME (IMON, IDAY, IYEAR, IHOUR, IMIN)
15 WRITE (LO, 502) CNTRL(4)
17 WRITE (LO, 503) CNTRL(4)
19 WRITE (LO, 504) CNTRL(20)
21 WRITE (LO, 504) CNTRL(20)
23 WRITE (LO, 601)
25 IORCF = 1+IDOCF
27 WRITE (10, 1001)
29 WRITE (12, 1201)
31 IBSIZE(3) = 3
33 WRITE (10,1001)
35 WRITE (12,1201)
37 IFRTY = 24
39 IRTY = CRTNL(19)
41 CRTL(20)
43 CALL TIME (IMON, IDAY, IYEAR, IHOUR, IMIN)
45 WRITE (LO, 502) CNTRL(4)
47 WRITE (LO, 503) CNTRL(4)
49 WRITE (LO, 504) CNTRL(20)
51 WRITE (LO, 504) CNTRL(20)
53 WRITE (LO, 601)
55 IORCF = 1+IDOCF
57 WRITE (10, 1001)
59 WRITE (12, 1201)
61 IBSIZE(3) = 3
63 WRITE (10,1001)
65 WRITE (12,1201)
67 IFRTY = 24
69 IRTY = CRTNL(19)
71 CRTL(20)
73 CALL TIME (IMON, IDAY, IYEAR, IHOUR, IMIN)
75 WRITE (LO, 502) CNTRL(4)
77 WRITE (LO, 503) CNTRL(4)
79 WRITE (LO, 504) CNTRL(20)
81 WRITE (LO, 504) CNTRL(20)
83 WRITE (LO, 601)
85 IORCF = 1+IDOCF
87 WRITE (10, 1001)
89 WRITE (12, 1201)
91 IBSIZE(3) = 3
93 WRITE (10,1001)
95 WRITE (12,1201)
0607 J3 = 0
0609 DO 14 J2 = 14, 17, 1
0611 DATA(J2, J1) = SCANR(IS, J3, IC)
0613 WRITE (LI, 121) J2, J1, IS, J3, IC, DATA(J2, J1), NOLF
0614 J3 = J3 + 1
0615 J2 = J2 + 1
0616 IF DATA(J2, J1) .LE. 9
0618 PRINT acquired data.
0620 14 CONTINUE
0621 IF DATA(J2, J1) .LT. 999
0623 CONTINUE
0625 IF DATA(J2, J1) .EQ. 999
0627 STOP 0601
0629 IF DATA(J2, J1) .LE. 999
0631 IF DATA(J2, J1) .LE. 9
0633 END
WRITE (LO, 605) (J2, J2=1,4)
CALL CODE
WRITE (IDATA, 115) (DATA(J1, J2), J2=1,4)
DO 20 J2=1,4,1
IF (DATA(J1, J2) .EQ. .999999 ) GO TO 20
DATA(J1, J2) = NO(1)
DATA(J1, J2) = NO(2)
DATA(J1, J2) = "H"
20 CONTINUE
WRITE (LO, 606) J1, ((DATA(J3, J2), J3=1,8), J2=1,4), J1
WRITE (IDATA, 115) (DATA(J1, J2), J2=1,4)
DO 25 J2=1,4,1
IF (DATA(J1, J2) .EQ. .999999 ) GO TO 25
DATA(J1, J2) = NO(1)
DATA(J1, J2) = NO(2)
DATA(J1, J2) = "H"
25 CONTINUE
WRITE (LO, 607) ((DATA(J3, J2), J3=1,8), J2=1,4)
WRITE (IDATA, 115) (DATA(J1, J2), J2=1,4)
DO 28 J2=1,4,1
IF (DATA(J1, J2) .EQ. .999999 ) GO TO 28
DATA(J1, J2) = NO(1)
DATA(J1, J2) = NO(2)
DATA(J1, J2) = "H"
28 CONTINUE
WRITE (LO, 608) J1, ((DATA(J3, J2), J3=1,8), J2=1,4), J1
WRITE (IDATA, 115) (DATA(J1, J2), J2=1,4)
DO 29 J2=1,4,1
IF (DATA(J1, J2) .EQ. .999999 ) GO TO 29
DATA(J1, J2) = NO(1)
DATA(J1, J2) = NO(2)
DATA(J1, J2) = "H"
29 CONTINUE
WRITE (LO, 609) (J2, J2=1,4)
READ (LI, 149) IDUM
IF (IDUM .EQ. 2) GO TO 34
IF (IDUM .EQ. 3) GO TO 75
STOP 0602
34 GO TO 11
WRITE (LI, 149) IDUM
WRITE (LI, 149) IDUM
READ (LI, 149) IDUM
WRITE (LO, 604) IHOUR, IMIN
GO TO 05

STORE acquired data on a disc type 1 file.

WRITE (LI, 105) IFILE, FILE
CALL CREAP (IDCB, IERR, IFILE, ISIZE, ITYPE, ISECU, ICR, IDCBS)
IF ( IERR .EQ. 0 ) GO TO 37
WRITE (LI, 103) IFILE
READ (LI, 149) IDUM
WRITE (LI, 149) (ICLR, I=1,3)
IF ( IDUM .EQ. 249 ) STOP 0603
IF ( IDUM .EQ. 250 ) GO TO 36
WRITE (LI, 103) IFILE, ISECU, ICR
CALL PURGE (IDCB, IERR, IFILE, ISECU, ICR)
IF ( IERR .LT. 0 ) STOP 0604
GO TO 35
CALL CODE
WRITE (NEW, 100) IDUM
WRITE (LI, 102) IFILE, NEW, IFILE(2), IFILE(3)
IF (I*) = NEW
GO TO 35
CALL OPEN (IDCB, IERR, IFILE, INPTN, ISECU, ICR, IDCBS)
IF ( IERR .LT. 0 ) STOP 0605
CALL WRITE (IDCB, IERR, DATA, IL)
IF ( IERR .LT. 0 ) STOP 0606
CALL CLOSE (IDCB, IERR, 0)
CALL URITF (ICB, IERR, 1)
STOP 0606
IF ( IERR .EQ. 1 ) WRITE (LO, 610) IFILE, ISECU, ICR, NO
IF ( IERR .EQ. 1 ) WRITE (LO, 611) IFILE, ISECU, ICR, NO
WRITE (LO, 612) IHOUR, IMIN
RETURN
END

FTN4 COMPILER: HP92060-16092 REV, 1926 (790430)
** NO WARNINGS ** NO ERRORS ** PROGRAM = 03193 COMMON = 00000
6. PROGRAM TXCO3

6.1. DESCRIPTION

TXCO3 is a son program of the father program TXCO9, by which it is scheduled if one of the following operations is desired:

7 - Check the instrumentation
8 - Manipulate the program control array CNTRL.

When scheduled by TXCO9, which suspends operation while the son program TXCO3 executes, the program TXCO3 reads the program control array from the disc, sets the HP interface bus and the measurement and control devices to remote control, preprograms the digital voltmeter (DVM), the scanners and the counter. CNTRL(50) is the actual decision variable to select and call the subroutine, which performs the desired operation.

When this subroutine has terminated, the interface bus and the devices are released from remote control and the control array is written into a disc file, so that the next TXCO module can read it. The correct termination of each subroutine can be verified by checking the stop codes. Note that all stop codes ending on 77 indicate correct execution of a subroutine.

<table>
<thead>
<tr>
<th>CNTRL(50)</th>
<th>Subroutine</th>
<th>STOP Code</th>
</tr>
</thead>
<tbody>
<tr>
<td>7</td>
<td>CHECK</td>
<td>TXCO3 : STOP 0777</td>
</tr>
<tr>
<td>8</td>
<td>CHNGE</td>
<td>TXCO3 : STOP 1077</td>
</tr>
</tbody>
</table>

EXTERNALS: REWRF, ABRT, RMOTE, CHECK, CHNGE, CLEAR, LOCL

COMMON BLOCKS: FMP, CIBUF, CONTR.
The FORTRAN-IV compiler for the HP 21 MX computer requests COMMON blocks to be predefined in a BLOCK DATA subroutine prior to using a COMMON block in a program, subroutine or function.

<table>
<thead>
<tr>
<th>BLOCK DATA Subroutine</th>
<th>Arrays &amp; Variables</th>
<th>Length in Words</th>
</tr>
</thead>
<tbody>
<tr>
<td>FMP</td>
<td>IDCBIFILEISEIZIESCUICR</td>
<td>227B = 151.10</td>
</tr>
<tr>
<td>CIBUF</td>
<td>IBUF</td>
<td>3200B = 1664.10</td>
</tr>
<tr>
<td>CONTR</td>
<td>CNTRL</td>
<td>400B = 256.10</td>
</tr>
</tbody>
</table>

The arrays and variables allocated by the COMMON block FMP are frequently used for the data transfer from and to the disc. COMMON block CIBUF is designed to take the largest raw data array in the TXCO data acquisition and reduction system - IBUF(1664) in subroutine FREER. The program modules CHECK and CHNGE do not use the complete area allocated by CIBUF. COMMON block CONTR allocates the space for the control array CNTRL.

MNEMONIC ABBREVIATIONS: None

ERROR MESSAGES: If CNTRL(50) is less than 7 or greater than 8, no subroutine has been selected and the program terminates outputting an error message (FORMAT $102) to the standard input device, i.e. the terminal.

PROCEDURE: For more detailed information, study the flow chart and the information given in the subroutine descriptions.

DATA FILE: None
VARIABLES IN BLOCK DATA FMP:

- ICDB (144) integer data control block
- IFILE (3) integer array to contain file name
- ISIZE (2) integer array to contain # of records in the first and record length in 16-bit-words in the second word
- ISECU integer security code of data file
- ICR integer cartridge reference number, where data file is located

VARIABLES IN BLOCK DATA CIBUF:

- IBUF (1664) integer buffer array for the raw data

VARIABLES IN BLOCK DATA CONTR:

- CNTRL (256) integer program control array

VARIABLES IN PROGRAM TXCO3:

- CNTRL (256) integer program control array
- NOLF integer suppresses line feed
- LI integer LU # of standard input device (terminal)
- ISTOP integer control variable to select STOP code
FLOW CHART PROGRAM TXCOS

START

Define COMMON block FMP.

Define COMMON block CIBUF.

Define COMMON block CNTR.

Read program control array from disk file into CNTRL.
CALL REWRF

I/O Assignment LI=CNTRL(12)
CALL RMOV (12)

1. Set interface but & devices to remote control
   CALL ABRT (7,2)
   CALL RMOFE (8)
   CALL RMONE (10)
   CALL RHFNE (12)
   CALL RMOFE (15)

2. Program devices
   WRITE (8, 201)
   WRITE (10, 1001)
   WRITE (12, 1201)
   WRITE (14, 1501)

3. Define STOP code
   STOP=CHTR (56) - 6

---

1. Interface Bus
2. Scanner #1
3. Digital Voltmeter
4. Counter
5. Scanner #2
TRANSONIC COMPRESSOR: PROGRAM SYSTEM TXCO FOR DATA ACQUISITION --ETC(U)
OCT 80  H ZEBNER

UNCLASSIFIED
CTRL(SR) = 7

CALL CHECK
CALL CHNRED

Release interface
bus and devices
from remote
conrol.
CALL CLEAR (?, U)
CALL LOCAL (?)

Write program
Control Group
look up disk
file CTRLF.
CALL REWF

Error = 1

Error 0???
STD* 10???

Display error message
END
6.2. SUBROUTINE CHECK:

PURPOSE: This subroutine enables the investigator to
(independently from the data acquisition modules ABSRV, CALIB,
FREER, PACER, COMB and STDY) check all data locations to
troubleshoot or verify the transonic compressor test rig measurement system.

ARGUMENTS: LO ; this variable specifies the output unit
where the protocol of the check is directed to. In any case,
the data are displayed on the standard input device (terminal LI) and if LO is equal to LI, double output is suppressed.
The selection of LO = 6 (line printer) is an appropriate choice for a hardcopy of the check protocol.

EXTERNALS: ACQN, SCANR, RPACE

COMMON BLOCK: CONTR; for detailed explanation refer to the
TXCO3 description.

MNEMONIC ABBREVIATIONS: None

ERROR MESSAGES: None

PROCEDURE: For more detailed information, study the flow chart. After having assigned the I/O reference, CHECK asks the operator which particular system should be checked.

<table>
<thead>
<tr>
<th>Instrumentation code</th>
<th>system being checked</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>S/V-S/V controller - scanner - DVM - system</td>
</tr>
</tbody>
</table>

192
<table>
<thead>
<tr>
<th>Instrumentation Code</th>
<th>System Being Checked</th>
</tr>
</thead>
<tbody>
<tr>
<td>2</td>
<td>amplifier - scanner - DVM - system</td>
</tr>
<tr>
<td>3</td>
<td>Pacer</td>
</tr>
</tbody>
</table>

The operator then selects the desired code and the program branches.

i) S/V - S/V controller - scanner - DVM - system

The operator has to input the number (1 thru 5) of the S/V, the low port and the high port. Erroneous input will cause the program to re-request the data. If S/V #2 is selected and either low or high port are odd, they will be increased to the next even number. In increments of 1 (2 resp., if S/V #2 is addressed) the subroutine steps from low to high port, taking a reading of each. The result is displayed and printed immediately. Upon completion the operator is asked whether another check shall be done. The answer is YES or NO, and if YES is entered, SUBROUTINE CHECK is run again from the beginning.

ii) amplifier - scanner - DVM - system

The operator has to input the number (1 or 2) of the scanner, the low channel and the high channel. Erroneous input of the scanner # will cause the program to re-request the data, whereas no check is made whether the boundaries for the scan, low channel ILOW and high channel IHIGH, are correct. In increments of 1 the subroutine steps from low channel to high channel, taking a reading at each port. The result is displayed and printed immediately. Upon completion, the operator is
asked whether another check shall be done. The answer is YES or NO, and if YES is entered, SUBROUTINE CHECK is run again from the beginning.

iii) **Pacer**

The operator has to input the pacer control parameters:

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>ADCHNL</td>
<td>A/D analog input channel to be selected by the A/D converter multiplexer.</td>
</tr>
<tr>
<td>PAMO</td>
<td>Pacer mode</td>
</tr>
<tr>
<td></td>
<td>= 1 allows pacer to trigger A/D conversion at the specified position in any blade interval. The variable PAIR is ignored.</td>
</tr>
<tr>
<td></td>
<td>= 2 causes pacer to select blade pair # PAIR.</td>
</tr>
<tr>
<td>PAIR</td>
<td># of blade pair selected (1 - 9)</td>
</tr>
<tr>
<td>START</td>
<td>Start count to step through blade passage</td>
</tr>
<tr>
<td>INCR</td>
<td>Increment to step through blade passage</td>
</tr>
<tr>
<td>STOP</td>
<td>Stop count to step through blade passage</td>
</tr>
<tr>
<td>REP</td>
<td>Number of repetitions at each individual point</td>
</tr>
</tbody>
</table>
This sketch shows how the pacer control parameters identify the location, where the probe takes data. Upon completion the operator is asked whether another check shall be done. The answer is YES or NO, and if YES is entered, start to read this section SUBROUTINE CHECK again.

DATA FILE: None

VARIABLES:
LO integer LU# of standard output device (line printer)
CNTRL(256) integer program control array
NOLF integer suppresses line feed
NOCR(2) integer suppresses line feed and carriage return
ICLR(3) integer clears line above cursor
ADCHNL  integer  Pacer control parameter  see section iii) for detailed explanation
PAMO   integer  Pacer control parameter
PAIR   integer  Pacer control parameter
START  integer  Pacer control parameter
INCR   integer  Pacer control parameter
STOP   integer  Pacer control parameter
REP    integer  Pacer control parameter
LI     integer  LU# of standard interactive input device (system terminal)

LINES  integer  line count
IDIUM  integer  decision variable
IPORT  integer  # of desired S/V (1 - 5)
ILOW   integer  low port of desired S/V
IHIGH  integer  high port of desired S/V
ISTEP  integer  increment to step from low to high port
IW     integer  delay between closing S/V port and taking the DVM reading in tens of ms.

V      real  pressure reading (raw data)
ISCR   integer  # of desired scanner (1 or 2)
ILOW   integer  low channel of desired scanner
IHIGH  integer  high channel of desired scanner
LS     integer  LU# of the desired scanner
D      real  voltage reading (raw data)
AVRGE  real  average voltage as returned from subroutine RPACE
Ask operator what scanner shall be checked, and choose low and high channel.

Read ISCR, ILOW & IHIGH; Define Lists of the scanners.

ISCR = 1?

\[ ? \]

YES

\[ LS = 3 \]

NO

\[ ISCP = 2 \]

YES

\[ ? \]

NO

...
20

21

22

23

24

25

LS = 15

NO

YES

ISDR < 1

or

ISDR > 2

NO

YES

LO NE LI

NO

Print key to data
print out on out-
put device (LO)

Display key to data
print out on input
device (Terminal 4)
Increase line count.

Prepare ZLOW &
ZHIGH to be DO
loop lower and
upper limit; i.e. add 1 line.
Subhead (one) from loop control variable to obtain channel #
I+I-1

Read voltage into D
J-SCARK (LS; I; 0)

LO.ME.LT
-YES
-NO

Print data on output device (LA)

Display data on terminal (LD); increase line count
26

27

30 (loop completed?) NO

30

28

29

Check amplifier - occur - DMI - System
Ask operator to enter PACER control parameters

Read ABCDE, FGH, IJ, KLM, NOP, QRS, TUV, WXYZ, and Z

Call PACER subroutine with input PACER and direct control output to unit indicated by CO.

CALL PACER

Check PACER

32

31

28

80

8
Ask operator, whether to take another system check.

Read decision variable IDUM.

YES IDUM = 24YE

RETURN
ENTRY

I/O assignment
LI = CNTRL (19)

LO. ME. LE? Yes

Print heading on output device.

No

Reset line count
LINES = 0

Ask operator, what system shall be checked.

Read decision variable IDUM
Ask operator, what S/V shall be checked; and enter low and high part.

Read INPUT, low & DVM; define STEP = 1.

IF INPUT ≤ 1 THEN YES

IF INPUT > 5 THEN NO
S/N #2 is a 2-bit port S/N and counts the ports by even numbers. If ILOW or INISH are odd, they will be changed to the next even value; redefine ISRP = 2.
12

13

14

Print key to date
print out on out-
put device (12)

Display key to date
print out on input
device (Terminal 13)
Increase line count

Do not divide 1120.55

Define delay between
closing 5V port
and taking the
DVM reading in
tens of milliamps
INV=CENTRAL (250)

Read PRESSURE
into V.
V=ACON(PORT1,5,2)

LO NE LI

YES

NO

15

16

17

18

207
6.3. **SUBROUTINE CHNGE:**

**PURPOSE:** Change any element of the program control array CNTRL on line and display any element of CNTRL.

**ARGUMENTS:** None

**EXTERNALS:** CODE, REWRF

**COMMON BLOCK:** CONTR; for detailed explanation refer to the TXCO3 description.

**MNEMONIC ABBREVIATIONS:**

- **C** ... Change CNTRL (i) to new value
- **D** ... Display current value of CNTRL (i)
- **R** ... Return to the calling program

**Note:** C is followed by the value of <<i>> and the new value for <<CNTRL (i)>> and D is followed by the value of <<i>>.

**ERROR MESSAGES:** None

**PROCEDURE:** For more detailed information, study the flow chart. After having assigned the I/O reference and reset the line count, the operator is asked what to do. The input in the first inverse video box specifies where to branch to.

1) **Change CNTRL (i) to new value**  
   **Input:** C

   The control character C is followed by the value of <<i>> and the new value for <<CNTRL (i)>>. Each of the latter data items has to be aligned to the right margin of the two
following inverse video boxes. If the input is to be compiled as ASCII code, the identifier 2H has to precede the two input characters.

**EXAMPLE A** Suppose, the time delay $IW$ between closing a S/V port and taking the DVM reading shall be changed to 1 second, which is $100 \times 10$ milliseconds. Enter

```
C 250 100
```

and press the RETURN key. The subroutine responds displaying the message CNTRL (250) changed from 80 to 100. Where 80 is the previous value of CNTRL (250).

**EXAMPLE B** The character, used to display the just acquired waveform in subroutine PACER shall be changed to the asterisk ($= 2H*$). Enter

```
C 249 2H*
```

blank, because subroutine PICTR outputs this item in Al-Format and press the RETURN key. The subroutine responds displaying

```
CNTRL (249) changed from 2H+$\omega$ to 2H*$\omega$ .
```

where the add sign ($= 2H+\omega$) was previously used for the plot.

ii) **Display actual value of CNTRL (i)** [Input: D]

The control character C is followed by the value of $\langle i \rangle$ and, only if the element CNTRL (i) shall be displayed in ASCII-mode, the string $\langle 2H \rangle$. The data for $\langle i \rangle$ has to be aligned to right margin of the second inverse video box and $\langle 2H \rangle$ has to be centered in the third box.

210
EXAMPLE A) Display the value for the cartridge reference number, where the raw data files are located. Appendix A.3 (Program Control Array) reveals that you have to look into CNTRL(30). Enter

```
D  30
```

and press the RETURN key. The subroutine responds by informing you that

The actual value of CNTRL (30) is 26.

EXmple B) Display the first two characters of the name of the raw data file, which are written into CNTRL (32). Since the file name is ASCII coded, the ASCII-identifier «2H» must not be forgotten. Enter

```
D  32  2H
```

and press the RETURN key. The subroutine responds by informing you, that

The actual value of CNTRL (32) is 2HT5.

This information reveals that the last data acquisition was a combination probe survey, since there all data file names start with «T5».

iii) Return [Input: R]

Enter R: Now the subroutine asks, whether to clear the informative responses, displayed by this subroutine previously. Inputting anything else but NO initializes the program to clear the screen.

Then the subroutine terminates writing the modified control array back into disc file.

211
FLOW CHART SUBROUTINE CHANGE

ENTRY

I/O assignment; Reset line count; LI=CNTPL (18)
ISCR = 0

If

Input instructions from keyboard.

Read IDUM, ICNT & INEW

IDUM = 2HC

YES

NO

2

3

1
Increase line count and write old value of CNTL (IDNT) into variable COLD.
ISRP = ISRP + 1
BLD = CNTL (IDNT)

IF the input supposed to be a number? ANSWER: YES

INCREASE PROGRAM CONTROL ARRAY

1  2  3

IDTN = 1HD

YES

IDTN = 2HR

YES

NO

4  5  6

Charge program control array

7  8

NO

YES
The input shall be written into CNTRL (IOCN) in ASCII format (A2).
CNTRL (IOCN) = NEW U.

Decode CNTRL (IOCN) from the variable INEW in IG format.
CALL CODE
READ (CNTRL, 1) CNTRL (IOCN)

Is the input a number?
NE2H / H H H/H

Display old and new value of CNTRL (IOCN) using IG format (A).
FORMAT 4 10.3

CR-to-daay

YES

NO

YES

NO

YES

NO
Display desired element of program control array

Change program control array

Display old and current value of control variable A2 format 4.3

Decrease the count

Display Current control variable A2 format 4.3
Display current value of CNTCL (IOT) using AL formats.
FORMAT # 107

Terminal. This subroutine: ask, if the terminal CTRP shall be cleared of all above stored data. Read QIDIEM variable ITUR.

YES

ITUR = 2H10

NO

ITUR > 0

Clear Screen by writing ITUR to terminal ISTK times

216
Clear one line above cursor by writing ICUR in A-Formal to term

Write program control array back into disk file CNTRLF

CALL REWRF

RETURN
6.4. PROGRAM LISTING TXCO3

PAGE 0001  FTN.  4:12 PM TUE., 23 SEP., 1980

0001  FTN4,L
0002    BLOCK DATA
0003    & / FMP / IDCB(144),IFILE(3),ISIZE(2),ISECU,ICR
0004    COMMON / FMP / IDCB,IFILE,ISIZE,ISECU,ICR
0005    INTEGER IDCB(144),IFILE(3),ISIZE(2)
0006    END

FTN4 COMPILER: HP92060-16092 REV. 1926 (790430)
## NO WARNINGS ## NO ERRORS ##
BLOCK COMMON FMP SIZE = 00151
BLOCK DATA
COMMON CIDUF / IBUF
INTEGER IBUF(1664)

FTN4 COMPILER: HP92060-16092 REV. 1926 (790430)
** NO WARNINGS ** NO ERRORS **
BLOCK COMMON CIDUF SIZE = 01664
BLOCK DATA
* / CONTR / CNTRL(256)
COMMON / CONTR / CNTRL
INTEGER CNTRL(256)
END

FTM4 COMPILER: HP92660-16072 REV. 1926 (790430)
** NO WARNINGS ** NO ERRORS **
BLOCK COMMON CONTR SIZE = 00256
PROGRAM TXCO3 (3,99)

The operating system RTE IV B requests the data acquisition program TXCO for the one stage transonic compressor to be

formatted into several programs scheduled by the father program TXCO. This son program TXCO3 consists of the subroutines

CHECK and CHNGE. These codes handle the equipment check and

the on-line modification of the control array. The data

transfer between father and son programs take place via the

control array CNTRL (disk file CNTRFL) and the data array

IBUFF (disk file IBUFF).

The utility subroutines ACON, CNTL, CURVE, ICON, IPORT,

PICTR, REWRF, RSPACE, SCANN, TIME and WAIT are added.

Author: Hans M. Zehner

Date: March 12, 1960

A detailed program description is available in the TXCO log.

* Third son program of father program TXCO.

COMMON / CONTR / CNTRL

INTEGER CNTRL(256)

DATA NOLF /006537B/

131 FORMAT (9X="20X",AP)

103 FORMAT (" TXCO3: PROGRAM ABORTED! NO SUBROUTINE HAS BE
XEN INITIALIZED.")

101 FORMAT ("FR7W3A1H0T3")

1201 FORMAT ("PA4G6T")

1501 FORMAT ("CA")

CALL REWRF (-1,2)

IF ( CNTRL(50) .LT. 7 .OR. CNTRL(50) .GT. 8 ) GO TO 63

Set interface bus and devices to remote control.

CALL ABRT (7,1)

CALL RNODE (9)

CALL RNODE (18)

CALL RNODE (19)

CALL RNODE (15)

WRITE (6, 881)

WRITE (10,1081)

WRITE (16,1681)

WRITE (15,1581)

Call subroutine indicated by CNTRL(50).

IF (CNTRL(50) = 6)

881 IF ( CNTRL(50) = 6 ) CALL CHNGE(CNTRL(51))

Release interface bus and devices from remote control.

CALL CLEAR (7,1)

CALL LOCL (7)
CALL REWF (1,2)
WRITE (LI, 102) NOLF
GO TO (81,02) ISTOP
STOP 0777
STOP 0777
WRITE (LI, 102)
END

FTN4 COMPILER: HP92060-16092 REV. 1926 (790430)

# NO WARNINGS # NO ERRORS # PROGRAM = 00220 COMMON = 00000
SUBROUTINE CHECK (LO)

Instrumentation check.

Author: Hans H. Zebner
Date: January 11, 1980
Detailed program description is available in TXCO log; the
comment statements match to the flow chart explanations.

* Checks the instrumentation:

COMMON / CONTR / CNTRL

INTEGER CNTRL(256), NOLF, NOCR(2), ICLR(3)
INTEGER ADCHNLPAM6, pAI, STARTTNCR, STOP, REP

C FORMATS CHECK START CONTROL OUT

182 FORMAT ("SUBROUTINE CHECK CONTROL OUT")

183 FORMAT ("
Enter instrumentation code"
184 "...
185 "...
186 "...
187 "...
188 "...
189 "...
190 "...
191 "...
192 "...
193 "...
194 "...
195 "...
196 "...
197 "...
198 "...
199 "...
200 "...
201 "...
202 "...
203 "...
204 "...
205 "...
206 "...
207 "...
208 "...
209 "...
210 "...
211 "...
212 "...
213 "...
214 "...
215 "...
216 "...
217 "...
218 "...
219 "...
220 "...
221 "...
222 "...
223 "...

Check completed")
149 FORMAT ((3A2))
150 FORMAT (FL8.3A14H173*)
151 C FORMATS CHECK STOP
152
153 C FORMAT (*FIRM3AiHiT3")
154)
155
156 C FORMAT (*FIRM3AiHiT3")
157)
158
159 C FORMAT (*FIRM3AiHiT3")
160)
161
162 C FORMAT (*FIRM3AiHiT3")
163)
164
165 C FORMAT (*FIRM3AiHiT3")
166)
167
168 C FORMAT (*FIRM3AiHiT3")
169)
170
171 C FORMAT (*FIRM3AiHiT3")
172)
173
174 C FORMAT (*FIRM3AiHiT3")
175)
03 IF (.LO_.NE_.LI) WRITE (LO, 103) IPORT
04 WRITE (LI, 103) IPORT
LINES = LINES+3

Read & output voltages.

WRITE (10, 101)
DO 04 I=LOW,IMICH,ISTEP
04 V = AGN(IMPORT,I,IW)
IF ( LO .NE. LI ) WRITE (LO, 104) I,V
LINES = LINES+1
GO TO 07

Check scanner; Input scanner 0, low channel, high channel.

05 WRITE (LI, 105) NOCR
READ (LI, 8) ISCR,LOW,IMICH
WRITE (LI, 149) (ICLR,I=1,2)
IF ( ISCR .EQ. 1 ) LO = 0
IF ( ( ISCR .LT. 1 ) .OR. ( ISCR .GT. 2 ) ) GO TO 05
LINES = LINES+3
LINES = ICR+1
IMICH = IMICH+1

Read & output voltages.

WRITE (10, 101)
DO 06 I=LOW,IMICH
11 = I-1
D = SCAN(LS,11,1)
IF ( LO .NE. L1 ) WRITE (LO, 104) I1,D
WRITE (LI, 104) I1,D
LINES = LINES+1
GO TO 07

Check pacer; Input pacer control parameters.

20 WRITE (LI, 109) (NOLF,I=1,2)
READ (LI, 149) ACHNL,PAHO,PACR,START,INCR,STOP,REP
WRITE (LI, 149) (ICLR,I=1,4)
WRITE (LI, 111) NOLF,ACHNL,PAHO,PACR,START,INCR,STOP,REP,NOLF
LINES = LINES+2
CALL RPACE (ACHERL,PAH0,PAIR,START,INCR,STOP,REP,AVRGE,1,LO)
LINES = LINES+3

Ask whether to perform more test samples.

07 WRITE (LI, 107) NOCR
READ (LI, 107) IDUM
LINES = LINES+1
WRITE (LI, 147) (ICLR,I=1,LINES)
IF ( IDUM .EQ. 6 ) WRITE (LO, 100)
RETURN
END
SUBROUTINE CHNGE

Modify control array CNTRL(256) interactively.

Author: Hans M. Zehner

Date: January 11, 1980

Detailed program description is available in TXCO leg. The

Comment statements match to the flow charts explanations.

Enables user to change control array on-line.

COMMON / CNTRL / CNTRL
             INTEGER CNTRL(256)
             INTEGER INEW(3), NOLF, NOCR(2), ICLR(3)

FORMATS CHNGE START

101 FORMAT (I,1X,14,1X,3A2)
102 FORMAT (16)
103 FORMAT ("CNTRL(13)" changed from "I6" to "I6"")
104 FORMAT ("CNTRL(13)" changed from 2M"A2" to 2M"A2").
105 FORMAT ("Clear screen? YES or NO 3A2")
106 FORMAT ("The actual value of CNTRL(13) is I7")
107 FORMAT ("The actual value of CNTRL(13) is 2M"A2").
108 FORMAT ((3A2))

FORMATS CHNGE STOP

DATA NOLF /000553B,040433B/
DATA ICLR /015524B,015515B,006537B/
DATA ICR /CNTRL(19)/

Input instructions from keyboard.

01 WRITE (LI, 100) NOLF
READ (LI, 149) IDUM, ICR, INEW
WRITE (LI, 149) (ICLR, I=1,8)
IF (IDUM .EQ. 3HD) GO TO 02
IF (IDUM .EQ. 3HR) GO TO 06
GO TO 01

Change desired element of control array CNTRL.

02 ICLR = ICR
IDOL = CNTRL(1CNT)
IF (INEW(2) NE. 2H2H) GO TO 03
CNTRL(1CNT) = INEW(3)
GO TO 04

03 CALL CODE
READ (INEW, 102) CNTRL(1CNT)
04 IF (INEW(2) .EQ. 2H2H) WRITE (LI, 103) ICNT, IDOL, CNTRL(1CNT)
IF (INEW(2) .NE. 2H2H) WRITE (LI, 104) ICNT, IDOL, CNTRL(1CNT)
GO TO 01

C

227
Display desired element of control array CNTRL.

Go to 01

READ (LI, 149) IDUM

IF CI CR .GT. 0 WRITE (LI, 149) (ICLR, I1, ISCR)

R07 I RTE (LZ 149)

END

FTN4 COMILER: P92060-W

REV. 926 (790430)

3 NO WARNINGS

3 NO ERRORS

3 PROGRAM - 0061

COMMON - 0880
7. **UTILITY SUBROUTINE PACKAGE TXCOU**

7.1. **Description**

Subroutines and functions, which are commonly used by either TXCO2, TXCO2 or TXCO3, are contained in the utility package TXCOU (source code is saved in file &TXCOU; relocatable binary code is saved in file %TXCOU). Thus the length of the TXCO1, TXCO2 or TXCO3 source files can be kept to minimum, which allows editing and recompiling TXCO1, -2, -3 separately, which saves time. When loading TXCO1, -2, -3, the load of the utility subroutines has to be included using the multiple search loader command (MS,&TXCOU). Since the utility subroutines and functions are short and straightforward, the comment statements and program explanation included in the code serve to describe them. The present section briefly outlines the utility subroutine package.

<table>
<thead>
<tr>
<th>Name</th>
<th>Purpose</th>
<th>Author</th>
</tr>
</thead>
<tbody>
<tr>
<td>ACQN</td>
<td>Positions Scanivalve (S/V) and reads DVM</td>
<td>Geopfarth</td>
</tr>
<tr>
<td>CNTL</td>
<td>Closes scanner channels, which control the S/V controller, HG78K</td>
<td>Geopfarth</td>
</tr>
<tr>
<td>CURVE</td>
<td>Computes coefficients for a linear curve fit</td>
<td>McGuire</td>
</tr>
<tr>
<td>ICON</td>
<td>Converts a one or two digit integer into a two character ASCII string</td>
<td>Geopfarth</td>
</tr>
</tbody>
</table>

229
<table>
<thead>
<tr>
<th>Name</th>
<th>Purpose</th>
<th>Author</th>
</tr>
</thead>
<tbody>
<tr>
<td>IPORT</td>
<td>Interrogates S/V controller and returns the present port #</td>
<td>Geopfarth</td>
</tr>
<tr>
<td>PICTR</td>
<td>Uses the (24 x 80 dot) CRT of a terminal for a graphics display of data acquired with the PACER</td>
<td>Zebner</td>
</tr>
<tr>
<td>REWRF</td>
<td>Data transfer disc file to program array and vice versa</td>
<td>Zebner</td>
</tr>
<tr>
<td>RPACE</td>
<td>Triggers A/D through the PACER and calculates the average voltage</td>
<td>Zebner</td>
</tr>
<tr>
<td>SCANR</td>
<td>Closes scanner channel and reads the DVM or digital counter</td>
<td>Geopfarth</td>
</tr>
<tr>
<td>TIME</td>
<td>Obtains date and time in ASCII-format</td>
<td>Zebner</td>
</tr>
<tr>
<td>WAIT</td>
<td>Causes a defined time delay</td>
<td>Geopfarth (Original)</td>
</tr>
</tbody>
</table>

**REAL FUNCTION ACQN**

Arguments: INTEGER: IVALVE, IADES, IW

IVALVE - - - Desired S/V #
IADES - - - Desired port # of S/V
IW - - - - Time delay in tens of ms between closing transducer relay and taking the DVM reading

230
Example: The pressure on S/V #4, Port #18 is to be read with the time delay to be 0.5 sec (= 500 ms = 50 x 10 ms). The correct call is

IVALVE = 4
IPORT = 18
IW = 50
PRES = ACQN (IVALVE, IPORT, IW)

or
PRES = ACQN (4, 18, 50)

In both cases the DVM reading is written into the real variable PRES.

It is desirable to step forwards systematically and sequentially through the required parts of a S/V in order to reduce unnecessary wear. Whenever a S/V is scanned, the operator should watch the data system closely. In some cases (e.g. if the HP 9830 is brought on line) the HP-Interface bus and the devices may be downed. If this happens when the program ACQN has closed the scanner channels (on scanner #1) which either resets or advances the S/V the S/V relay will burn out. To prevent damage, the operator must turn off the power to scanner #1 immediately, then bring the data system up again using the UP-command (see HP manuals).

SUBROUTINE CNTL (ICHAN, IDEL, ISTEP, K)

Arguments: INTEGER: ICHAN, IDEL, ISTEP, K

ICHAN --- Channel # of scanner #1 (LU# = 8)
IDELE - - - Number of repetitions to close
the scanner channel

ISTEP - - - Increment to step from 1 through
IDELE

K - - - - Function code

K = 1 Close for 10 ms, wait for
150 ms; Repeat "IDELE" times;
return

K = 2 Close for 10 ms, wait for
4 sec; return

K = 3 Close; return

An example is unnecessary since the only subroutine to use
SUBROUTINE CNTL is the REAL FUNCTION ACON, which is itself
a utility. The user won't have to deal with CNTL.

SUBROUTINE CURVE (N, X, Y, SLOPE, SECON)

Arguments: INTEGER: N
REAL: X(N), Y(N), SLOPE, SECON

N - - - Number of data points
X(N) - - - Abscissa of data points
Y(N) - - - Ordinals of data points
SLOPE - - Slope of linear curve fit
SECON - - Intercept of linear curve fit

Example: Suppose the following (N = 6) pairs of data points
shall be approximated by a linear curve fit:

X(1) = 1.0    Y(1) = 105.6
X(2) = 2.5    Y(2) = 105.4

232
\[ \begin{align*}
X(3) &= 3.0 & Y(3) &= 104.8 \\
X(4) &= 5.0 & Y(4) &= 104.1 \\
X(5) &= 6.5 & Y(5) &= 102.9 \\
X(6) &= 8.0 & Y(6) &= 102.7
\end{align*} \]

The situation is shown in the following sketch:

To obtain the slope and the intercept of the linear curve fit (which is derived using the least squares criterion), program

```
CALL CURVE (6, X, Y, SLOPE, SECON)
```

and the results will be returned from SUBROUTINE CURVE

SLOPE = -.461

SECON = 106.247
INTEGER FUNCTION ICON

Arguments: INTEGER: I, N

I, N - - - Two integer numbers to be added

ICON - - - The result of the addition (which should not exceed two digits), but ASCII converted.

Example: Suppose the data documentation page IPAGE shall be converted to an ASCII-string named JPAGE. The correct call is:

JPAGE = ICON (IPAGE, 0)
or

JPAGE = ICON (0, IPAGE)

The bit structure, if IPAGE is 71₁₀, is as follows:

15

0

IPAGE: 0 0 0 0 0 0 0 0 0 1 0 0 0 1 1 1

JPAGE: 0 0 1 1 0 1 1 1 0 0 1 1 0 0 0 1

ASCII-converted ASCII-converted

seven (7) one (1)

ASCII converted 71

INTEGER FUNCTION IPORT

Arguments: INTEGER: IVALVE

IVALVE - - - Desired S/V (1 through 5)

Example: The call

IWHERE = IPORT(5)

returns the present port # of S/V #5 and writes it into the variable IWHERE.
SUBROUTINE PICTR

Arguments: INTEGER: LO, NUMBER, NEWPG, ICL, ISIGN
            REAL: AMPL

LO - - - Defines terminal LU# (either 1 or 18)
NUMBR - - Identifier to appear in the "drawing"
NEWPG - - No significance! Will be altered by PICTR
ICL - - - No significance! Will be altered by PICTR
ISIGN - - Character to be used for the drawing
AMPL - - Amplitude range of the raw data returned to the calling program

Subroutine PICTR is designed for the TXCOL-subroutine PACER, from which it gets the data through the COMMON block CIBUF. PICTR uses the enhanced display capabilities of the video terminal to produce a "drawing" which, of course, is of limited resolution. It allows the operator to verify the acquired raw paced run data qualitatively by checking the "drawing" against the oscilloscope display. See the listing of subroutine PACER, if an example is needed.

SUBROUTINE REWRF

Arguments: INTEGER: IREWRF, IWHAT

IREWRF - - Determines whether the array indicated by IWHAT shall be read from a disc file into an array or whether it shall be written into a disc file

235
IREWRF = -1  Read data from disc file into array
IREWRF = +1  Write data from array into disc file
IWHATA  --  Specifies the type of data to be transferred
IWHATA = 1  Array IBUF ↔ disc file IBUFF
IWHATA = 2  Array CNTRL ↔ disc file CNTRLF

This subroutine relieves the individual TXCO1, -2, -3 subroutines from the routine task of data transfer between disc and program (CP).

Example: The four applications are:

i) Read the program control array from disc file CNTRLF::26 into array CNTRL:  CALL REWRF (-1, 2)

ii) Write the program control array from array CNTRL into disc file CNTRLF::26:  CALL REWRF (1, 2)

iii) Read the raw data buffer from disc file IBUFF::26 into array IBUF:  CALL REWRF (-1, 1)

iv) Write the raw data buffer from array IBUF into disc file IBUFF::26:  CALL REWRF (1, 1)

SUBROUTINE RPACE
Arguments:  INTEGER:  ADCHNL, PAMO, PAIR, START, INCR,
            STOP, REP, IPRINT, LO

            REAL:  AVRGE

            ADCHNL  --  A/D input channel to be selected (0...15)
            PAMO  --  Pacer mode (1 or 2)
            PAIR  --  Blade pair (1...9)
            START  --  Start location (in counts) for data scan across the blade passage
INCR -- - Step size (in counts) to scan across the blade passage

STOP -- - Stop location (in counts) for data scan across the blade passage.

REP -- - - Number of repetitions at each location in the blade passage

IPRINT -- Flag to decide whether to output intermediate information

IPRINT = 1 Print all intermediate data and suppress pointer at the terminal

IPRINT = 0 Suppress printed output and initialize pointer at the terminal

LO -- - - LU# of device for printed output

AVRGE -- - Average voltage of paced run data

This subroutine is the control program to acquire data from the A/D converter in the synchronized PACER mode. (See also the description of subroutine PACER (Section 4.5), where the synchronized sampling is explained.) Although the (decoded) voltages from the A/D converter are REAL numbers between -1.0 volt and +1.0 volt, the data storage uses an integer array. Before writing the voltages into the data array, they are multiplied by 10,000. Note that if one of the PACER control parameters is out of the defined range, it is set to a default value without outputting any warning. The subroutine contains a large number of I/O statements
which were necessary during the development of the TXCO program system. In order to increase speed, the unnecessary statements should be removed. A further improvement would be a conversion from FORTRAN to ASSEMBLER programming language. An example of a call to subroutine RPACE is contained in the description and listing of subroutine PACER (Section 4.5).

**REAL FUNCTION SCANR**

**Arguments:** INTEGER: LU, ICHAN, K  

LU - - - Logical Unit # of the desired scanner  
(either 8 - scanner #1, or 15 - scanner #2)  
ICHAN - - - Scanner channel (integer)  
K - - - - Instrument code  
K = 1 Read the DVM  
K = 2 Read the digital counter  

**Example A:** To obtain the torque reading from the transonic compressor test rig, which is fed into scanner #1 (LU = 8), channel 36, program  

LU = 8  
ICHAN = 36  
K = 1  
TORQUE = SCANR (LU, ICHAN, K)  
or  
TORQUE = SCANR (8, 36, 1)  

In both cases the DVM reading is written into the real variable TORQUE.
Example B: To obtain the RPM of the Allis-Chalmers (central air supply) compressor, program

\[
\text{LU} = 8 \\
\text{ICHAN} = 15 \\
K = 2 \\
\text{RPMAC} = \text{SCANR} (\text{LU, ICHAN, K})
\]
or

\[
\text{RPMAC} = \text{SCANR} (8, 15, 2)
\]

In both cases the reading of the digital counter is written into the real variable RPMAC.

<table>
<thead>
<tr>
<th>Example</th>
<th>Instrument Code</th>
<th>Instrument Read</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>1</td>
<td>Digital Voltmeter</td>
</tr>
<tr>
<td>B</td>
<td>2</td>
<td>Digital Counter</td>
</tr>
</tbody>
</table>

SUBROUTINE TIME

Arguments: INTEGER: IMON, IDAY, IYEAR, I HOUR, I MIN, ISEC

IMON — ASCII converted month of the year
IYEAR — ASCII converted last two digits of current year
IDAY — ASCII converted day of the month
I HOUR — ASCII converted hour (24-hour clock) of the day
IMIN — ASCII converted minute of the hour
ISEC — ASCII converted second of the minute

IMON, IDAY and IYEAR are obtained from the program control array; I HOUR, IMIN and ISEC are obtained from the system
clock through an EXEC call; all variables are returned to the calling program.

Example: Suppose the system clock is set correctly and the control array CNTRL is defined, i.e., CNTRL was read from the disc file and adjusted to the actual conditions; then the following code

```
603 FORMAT ("Date & Time: "A2"/"A2"/"A2,2X,
   *A2":"A2)
   CALL TIME (IMON,IDAY,IYEAR,IHOUR,IMIN,ISEC)
   WRITE (6,609) IMON,IDAY,IYEAR,IHOUR,IMIN
```

produces the following output:

```
Date & Time: 09/27/89 21:57
```

SUBROUTINE WAIT

Arguments: INTEGER: TWAIT

TWAIT -- Time delay in tens of milliseconds

Example: To cause a defined time delay of 5.7 sec (= 5700 ms = 570 x 10 ms), program

```
ITWAIT = 570
CALL WAIT (ITWAIT)
```

or

```
CALL WAIT (570)
```
REAL FUNCTION ACQN (IVALVE, IADES, IW)

Position scannivalve IVALUE to port IADES and define ACQN
the DVM output voltage. A time delay of (IWAIT*10) ms occurs
between port selection and voltage measurement. The DVM is
triggered by issuing HP-IR subroutine TRIGR.

Author: Robert H. Geopfarth

Date: January 31, 1979

A detailed program description is available in TXCO log. The
variables are:

IVALVE ... Desired S/V.
IADES ... Desired S/V port #.
IAPR ... Present S/V port #.
ICHAN ... ASCII converted scanner channel.
ACQN ... Transducer voltage as read from DVM.
IW ... Time delay factor.

Positions scannivalve and reads DVM. Utilities.

FORMAT ("C")

FORMAT (/"*** ERROR DETECTED IN REAL FUNCTION ACQN;/*/"
** CHECK FOR BAD PARAMETER IN CALL!/*/
** IVALVE ="13" IADES ="13" IW ="14/

ISTEP = 1

IF (IVALVE .LT. 1 .OR. IVALUE .GT. 5 ) GO TO 06
IF (IADES .LT. 1 .OR. IADES .GT. 48 ) GO TO 06

K = 2

ISTEP = 2

IF (IVALVE .EQ. 2 .OR. IVALUE .EQ. 3 ) ISTEP = 2

IF ( IADES .LT. 1 .OR. IADES .GT. 48 ) GO TO 06

IF ( IVALUE .EQ. 2 .OR. IVALUE .EQ. 3 ) ISTEP = 2

Compare present port # to desired port#.

IAPR = IADES(IVALVE)
IDEL = IADES-IAPR
IF (IDEL) 02, 03, 04

Desired port below present port; reset S/V

ICHAN = ICON(IVALVE,4)
GO TO 05

Present port is present port; close X-ducer relay & read

ICHAN = ICON(IVALVE,9)
GO TO 05

Desired port is above present port; advance S/V.

ICHAN = ICON(IVALVE,4)
K = 1

Control S/V.

CALL CHTL (ICHAN, INEL, ISTEP, K)
IF (K .NE. 3) GO TO 01
0077 C 'Pause and read transducer output voltage.
0078 C
0079 C
0080 CALL WAIT (IU)
0081 CALL TRIGR (10)
0082 READ (10, %) DUM
0083 CALL TRIG (10)
0084 READ (10, %) ACQN
0085 WRITE (10, BU)
0086 RETURN
0087
0088 C
0089 C Error encountered; output error message; return.
0090 C
0091 C
0092 06 WRITE (10, 'Error encountered')
0093 RETURN
0094 C END
0095

FTN4 COMPILER: HP92060-16092 REV. 1926 (790430)

** NO WARNINGS ** NO ERRORS ** PROGRAM = 00237 COMMON = 00000
SUBROUTINE CNTL(ICHAN, IDEL, ISTEP, K)

* Closes scanner channel.

THIS PROGRAM Closes scanner channel "ICHAN" in steps of "ISTEP" based upon "K". (ICHAN Must BE an ASCII-converted integer.)

K FUNCTION

1  close for 10-MS wait for 150-MS. repeat "IDEL" times.

2  close for 10-MS wait for 4-SEC return.

3  close return.

Author: R.N. Geopfarth, Lt USN

Date: Jan 79

GO TO (100, 200, 300), K

DO 10 I = 1, ISTEP
     WRITE(S, 60) ICHAN
     CALL WAIT(I)
     WRITE(S, 62)
     CALL WAIT(15)
     CONTINUE
     RETURN

WRITE(S, 60) ICHAN
     CALL WAIT(400)
     RETURN

WRITE(S, 60) ICHAN
     RETURN

FORMAT(A2)

END
SUBROUTINE CURVE (N,X,Y,SLOPE,SECON)

* Compute linear curve fit using least square root method.

Author: Alan G. McGuire

Date: February 21, 1980

A detailed program description is available in TXCO log. The comment statements and statement numbers match to the ones used in the flow chart.

* Computes linear curve fit.

REAL X(N),Y(N)

SUMPO = 0.
SUMEO = 0.
SUMUP = 0.
SUME2 = 0.

DO 1 I=1,N-1
SUMEO = SUMEO + X(I)
SUMUP = SUMUP + (X(I)*Y(I))
1 SUME2 = SUME2 + (X(I)*X(I))

RN = FLOAT(N)

SNUM = (RN*SUMUP) - (SUMEO*SUMPO)
SDEN = (RN*SUME2) - (SUMEO*SUMED)

SLOPE = SNUM/SDEN
SECON = (SUMPO-(SLOPE*SUMEO))/RN

RETURN

END
INTEGER FUNCTION ICON (I,N)

Converts integer numbers into ASCII string.

Date: January 31, 1979

Because of the simplicity of the program the program description is included in this box.

IC  N ... integer numbers to be added.
ICOM ... 2-bit character ASCII string to be returned

* Converts integer to ASCII-string.

100 FORMAT (12)

IC = IC + N
IF ( IC .LT. 10 ) GO TO 01
CALL CODE
WRITE (ICON,100) IC
RETURN
01 ICON = IC + 30060B
RETURN 
END
INTEGER FUNCTION IPORT (IVALUE)

* Interrogates scanivalve.

THIS PROGRAM INTERROGATES SCANIVALVE
"IVALUE" AND CONVERTS PORT ADDRESS
INTO A DECIMAL VALUE.

VARIABLES:

IVALUE = DESIRED S/V
IP = S/V INPUT BUFFER
MSD = MOST SIGNIF. DIGIT
LSD = LEAST SIGNIF. DIGIT
IPORT = DECIMAL S/V ADDRESS

AUTHOR: R. N. GEOPFARTH, LT USN
DATE: DEC 78

CALL EXEC(2,LU,IVALUE*256,-1)
CALL EXEC(1,LU,IP,-1)
IP = IP/256
MSD = IAND(IP/16,7B)
LSD = IAND(IP,7B)
IPORT = 10*MSD + LSD
RETURN
END

FTN4 COMPILER: HP92060-16092 REV. 1926 (790430)

** NO WARNINGS ** NO ERRORS ** PROGRAM = 00064 COMMON = 00000
SUBROUTINE PICTR (LO,NUMBR,NEWPG,ICL,ISIGN,AMPL)

: Use terminal screen for graphic display.

* Use CRT to display the acquired data. *

COMMON / CIFBUF / IBUF
REAL CLR(64)
INTEGER IRUF(164)
INTEGER NOLFNOCR(2),ICLR(3),PLOT(9)
INTEGER BLACK(2),GREY(2),WHITE(2),LN(2),TEXT(20),HJF(3)

DATA NOCR /006339,040439/
DATA ICLR /015S249,01SS15B/
DATA WNIT /054460/
DATA GREY /015446B,062112B/
DATA HJF /01SS10BOiSS12,01SS062/
DATA ICLEAR /2H/

FORMATS PICTR START
1801 FORMAT (2A2,79X&3A2)
1803 FORMAT (A2*44I3"CFigure","3","20A2"
1805 FORMAT (2A2,9X 2A2,45X,2A2,6X,3A2)
1806 FORMAT (A2*13"-CFigure","3","20A2")
1807 FORMAT (2A2,2X,2A2,1X,(2A2,5X,2A2,1X),2A2,6X)
1809 FORMAT ("GF.3,4F16.3")
1817 FORMAT (Q2)
1819 FORMAT (A2*13"-CFigure","3","20A2")

FORMATS PICTR STOP
21 IF ( IFIRST .EQ. 1 ) GO TO 21
IFIRST = 1
IFIRST = 1
NEWPG = 1
21 IF ( NEWPG .EQ. 1 ) ICL = 0
IF ( LO .EQ. 1 ) GO TO 01
IF ( LO .EQ. 1 ) GO TO 02
01 LH(1) = GREY(1)
LH(2) = GREY(2)
GO TO 13
02 LH(1) = BLACK(1)
LH(2) = BLACK(2)
03 BG(1) = WHITE(1)
BG(2) = WHITE(2)
IF ( ICL .NE. 1 ) GO TO 25

Clear dots in frame w/e erasing the frame.

11 ICL = 1
ICOL = 1
CALL CODE
WRITE (PLOT,1819) BLACK(1),ICOL,ICLEAR,NOCR
WRITE (LO,1817) PLOT
DO 13 I=1,21
T = I-11
13 J = T
IF ( CLR(I) .GE. X-0.5 .AND. CLR(I) .LT. X+0.5 ) NUPDN = J
IF = I
CALL CODE
WRITE (PLOT,1819) BLACK(1),ICOL,ICLEAR,NOCR

Get the curve in the format required by PICTR.

25 J=0
DO 44 I=1,256,4
44 CLR(I) = IRUF(I)/10000.
44 XMIN = CLR(1)
44 XMAX = CLR(1)
IF ( ICL .EQ. 1 ) GO TO 43
DO 41 I=1,64
  IF ( CLR(1) .GT. XMAX ) XMAX = CLR(1)
  IF ( CLR(1) .LT. XMIN ) XMIN = CLR(1)
  AMPL = XMAX - XMIN
  IF ( A(5)(XMIN) .GT. XMAX ) AMPL = XMIN
  IF ( AMPL .LE. 0.001 ) AND. AMPL .GT. 0.000 ) AMPL = 0.001
  IF ( AMPL .GT. 0.001 ) AND. AMPL .LT. 0.005 ) AMPL = 0.005
  IF ( AMPL .GT. 0.005 ) AND. AMPL .LT. 0.025 ) AMPL = 0.025
  IF ( AMPL .GT. 0.025 ) AND. AMPL .LT. 0.250 ) AMPL = 0.250
  IF ( AMPL .GT. 0.250 ) AND. AMPL .LT. 1.000 ) AMPL = 1.000
  IF ( AMPL .GT. 1.000 ) AND. AMPL .LT. 10.000 ) AMPL = 10.000

CONTINUE

DO 42 I=1,64
  CLR(I) = CLR(I)*(10.0/AMPL)
  IF ( NEWP .ME. I ) GO TO 31

CONTINUE

05 WRITE (LO,1817) JNF(1),JNF(2),NOLF
  WRITE (LO,1805) BC, BLACK
  DO 46 I=1,9
    WRITE (LO,1805) BC, BLACK

06 WRITE (LO,1805) BC, BLACK
  DO 47 I=1,9
    WRITE (LO,1805) BC, BLACK


CONTINUE

31 ZER0 = 0
  AMP = AMP
  IRM = 0
  IC0L = 0
  WRITE (LO,1803) BLACK(1), IBM, ICOL, NUMBR, TEXT
  IC0L = 2
  WRITE (LO,1806) BLACK(1), IBM, ICOL, AMP
  IC0L = 1
  WRITE (LO,1806) BLACK(1), IBM, ICOL, ZERO
  IC0L = 21
  WRITE (LO,1806) BLACK(1), IBM, ICOL, AMP

CONTINUE

08 CONTINUE

CONTINUE

DO 48 I=1,64
  DO 49 I=1,21
    I = I

09 IF ( CLR(1) .GE. X-0.5 .AND. CLR(1) .LT. X+0.5 ) NUPDN = J
  IF ( CLR(1) .GE. X-0.5 .AND. CLR(1) .LT. X+0.5 ) NUPDN = J
  IC0L = 721
  CALL CODE
  WRITE (PLOT,1817) BLACK(1), IRM, ISIGN, NOCR
  WRITE (PLOT,1817) BLAIC(1), IRM, IC0L, ICLEAR, NOCR
  RETURN
END.

FTN4 COMPILER: HP92060-16872 REV. 1926 (790430)

## NO WARNINGS ## NO ERRORS ## PROGRAM = 01345 COMMON = 00000

249
SUBROUTINE RERRF (IREWR, IWHATA)

This subroutine reads (IREWR = -1) or writes (IREWR = +1) of a array specified by IWHATA.

Author: Hans P. Zebner
Date: February 08, 1980

Detailed program description is available in TXCO leg. The comment statements match to the flow chart explanations.

* Read or write of array.

COMMON /CBUF/ IBUF
COMMON /CON/ CNTRL
C COMMON /A /
COMMON /FMP/ IDCB,IFILE,ISIZE
C COMMON /FMP/ ID崂,NOCR(2),ICLR(3)
DATA NOLF /606S7B/
DATA NOCR /000033B,0404339/
DATA ICLR /01SS249,0tSSiSB,006379/

FORMAT ("REWRF:ARRAY ISUF(b64)
DISC FILE IBUFF")
FORMAT ("REWRF:DISC FILE IBUFF:00:26 ARRAY IUF(t664)"
FORMAT ("REWRF:ERROR RETURN (IWHATA *="13")")
FORMAT ("REWRF:DISC FILE CNTRL:00:12 ARRAY CNTRL(2S6)"
FORMAT ("REWRF:DISC FILE AF:00:26 ARRAY A(2S6)"
FORMAT ("REWRF:DISC FILE AF:00:26 ARRAY A(2S6)"
FORMAT ("REWRF:DISC FILE AF:00:26 ARRAY A(2S6)"
FORMAT ("REWRF:ERROR RETURN (IWHATA *="13")")
CALL OPEN (IDCD, IERR, -3A2-, 12, 12-, 12*, 12*)
failed; STOP*21X")
CALL LOCF (IDCD, IERR, IDU, IDUM, IDUMISIZE(2), IDUMIDUMISIZE(2))
failed; STOP*26X")
CALL READF (IDCB, IERR, A, *13", "I2", "I2")
failed; STOP*26X")
CALL WRITF (IDCB, IERR, CNTRL, "I3", "I2", "I2")
failed; STOP*26X")
CALL CLOSE (IDCD, IERR, 0) failed; STOP*40X**
* 205
POS
Integer array IBUF being written back and forth.

= 0 ISECU = 0 ICR = 26 IF (IWHATA .LT. 1 .OR. IWHATA .GT. 2 ) GO TO 40

10 CALL CODE
WRITE (IFILE, 108)
CALL OPEN (IDCD, IERR, A, IFILE, IOPN, ISECU, ICR, IDCB)
GO* 0 ) GO TO 11
WRITE (LI, 121) IFILE, ISECU, ICR, IDCB
STOP 1
11 CALL LOCF (IDCD, IERR, IDUM, IDUM, IDUMISIZE(1), IDUM, IDUMISIZE(2))

0451 IF ( IERR .GE. 0 ) GO TO 12
0453 STOP 2
0454 CALL Rnf (IDCB,IERR)
0455 IF ( IERR .GE. 0 ) GO TO 13
0456 WRITE (LI, 123)
0457 STOP 3
0458 13 ISIZE (1) = ISIZE (1)/2
0459 IL = ISIZE (1) * ISIZE (2)
0460 IF ( IERR .GE. 2 ) GO TO 14
0461 CALL READF (IDCB,IERR,IBUF,IL)
0462 IF ( IERR .GE. 0 ) WRITE (LI, 102)
0463 IF ( IERR .GE. 0 ) GO TO 16
0464 WRITE (LI, 124) IL,LEN,NUM
0465 STOP 4
0466 14 CALL WRITEF (IDCB,IERR,IBUF,IL)
0467 IF ( IERR .GE. 0 ) WRITE (LI, 101)
0468 IF ( IERR .GE. 0 ) GO TO 16
0469 WRITE (LI, 125) IL,LEN,NUM
0470 STOP 5
0471 15 CALL CLOSE (IDCB,IERR,0)
0472 IF ( IERR .GE. 0 ) GO TO 17
0473 WRITE (LI, 130)
0474 STOP 6
0475 17 RETURN

0481 20 CALL CODE
0482 WRITE (FILE, 109)
0483 CALL OPEN (IDCB,IERR,FILE,IOPTN,ISECU,ICR,IDCB)
0484 WRITE (LI, 121) IFILE,IOPTN,ISECU,ICR,IDCB
0485 STOP 11
0486 21 CALL IOCF (IDCB,IERR,IDUM,IDUM,IDUM,ISIZE(1),IDUM,IDUM,ISIZE(2))
0487 IF ( IERR .GE. 0 ) GO TO 22
0488 WRITE (LI, 122)
0489 STOP 12
0490 22 CALL Rnf (IDCB,IERR)
0491 IF ( IERR .GE. 0 ) GO TO 23
0492 WRITE (LI, 123)
0493 STOP 13
0494 23 ISIZE (1) = ISIZE (1)/2
0495 IL = ISIZE (1) * ISIZE (2)
0496 IF ( IERR .GE. 2 ) GO TO 24
0497 IF ( IERR .GE. 4 ) GO TO 25
0498 CALL READF (IDCB,IERR,CTRL,IL)
0499 IF ( IERR .GE. 0 ) WRITE (LI, 104)
0500 IF ( IERR .GE. 0 ) GO TO 26
0501 WRITE (LI, 126) IL,LEN,NUM
0502 STOP 14
0503 24 CALL WRITEF (IDCB,IERR,CTRL,IL)
0504 IF ( IERR .GE. 0 ) WRITE (LI, 103)
0505 IF ( IERR .GE. 0 ) GO TO 26
0506 WRITE (LI, 127) IL,LEN,NUM
0507 STOP 15
0508 25 CALL CLOSE (IDCB,IERR,0)
0509 IF ( IERR .GE. 0 ) GO TO 27
0510 IF ( IERR .GE. 0 ) GO TO 16
0511 WRITE (LI, 130)
0512 STOP 16
0513 27 RETURN

0521 30 CALL CODE

Integer array CNTRL being written back and forth.

Real array A being written back and forth.
WRITE (IFILE,110)
CALL OPEN (IDCB, IERR, IFILE, IOPTN, ISECU, ICR, IDCBS)
WRITE (LI, 122)
STOP 21
CALL LOCDF (IDCB, IERR, IDUM, IDUM, ISIZE(1), IDUM, IDUM, ISIZE(2))
IF IERR .GE. 0 ) GO TO 32
WRITE (LI, 122)
STOP 22
CALL RWDNF (IDCB, IERR)
IF IERR .GE. 0 ) GO TO 33
WRITE (LI, 122)
STOP 23
ISIZE(1) = ISIZE(1)/2
IL = ISIZE(1) * ISIZE(2)
IF IREW .EQ. -1 ) GO TO 34
CALL READF (IDCB, IERR, A, IL)
IF IERR .GE. 0 ) WRITE (LI, 106)
IF IERR .GE. 0 ) GO TO 36
WRITE (LI, 128) IL, LEN, NUM
STOP 24
CALL WRTIF (IDCB, IERR, A, IL)
IF IERR .GE. 0 ) WRITE (LI, 105)
IF IERR .GE. 0 ) GO TO 36
WRITE (LI, 129) IL, LEN, NUM
STOP 25
CALL CLOSE (IDCB, IERR)
IF IERR .GE. 0 ) GO TO 37
WRITE (LI, 130)
STOP 26
RETURN

COMMON 00000

Error; IWHAT is not defined.

RETURN

IWHAT = -IWHAT
RETURN
END

FTM4 COMPILER: HP902660-16092 REV. 1926 (790438)

## NO WARNINGS ## NO ERRORS ## PROGRAM = 01148 COMMON = 00000

252
SUBROUTINE RPACE (ADCHNL, PAMO, PAIR, START, INCR, STOP, REP, AVRGE, IPRINT)

* Interface program to trigger HP A/D converted through pacer.

* Author: Hans M. Zebner

* Date: March 20, 1990

* Detailed program description is available in TXCO log.

* Triggers A/D through Pacer.

COMMON /CIBUF / BUFR
COMMON / CONTR / CNTRL

INTEGER BUFR(1664), CNTRL(256)

INTEGER ADCHNL, PAMO, PAIR, START, INCR, STOP, REP, NOLF, ICLR(3), WHERE, WH

DATA NOLF /0065279/
DATA ICLR /0065245, 0065515B, 0065328/
DATA MASK /1777009/
DATA FSVLTG /-1E01/

FORMAT (*Q* Enterd SUBROUTINE RPACE (I2, MI1, NI2, INW uI2, uI2, AVRGE, uI2)

* Formats RPACE START

* IF ( LO .EQ. A.D LI ) GO TO 01

IF ( PAMO .GT. 0 .OR. PAIR .LT. 0 )

IF ( START .LT. 2 .OR. PAIR .GT. 2 )

IF ( INCR .LT. 1 .OR. REP .GT. 1 )

IF ( ADCHNL .GT. 27 .OR. ADCHNL .LT. 1 )

IF ( PAMO .GT. 0 .OR. PAIR .LT. 0 )

IF ( START .LT. 2 .OR. PAIR .GT. 2 )

IF ( INCR .LT. 1 .OR. REP .GT. 1 )

Check input variables for logical errors.

Get adjusted START and STOP, depending on selected PAMO.
Start acquisition loop.

AVRGE = 0
WHEREP = 1
DO 10 I = START, STOP, INCR
   I = I + 1
   AVRGE = AVRGE + BUFR(I)
   WHEREP = WHEREP .AND. IPRINT .EQ. 0
10 CONTINUE

Repeat A/D conversion at selected point REP times.

BUFR(I) = 0
DO 10 I = 1, REP
   CALL EXEC(I)
   CALL EXEC(19, IPRM, 1, 1)
   CALL EXEC(20, BUF, I, ADDCNL, 0)
   IBUFR = (AND(INBUF, Mask))
   BUFR = Float(IBUF)/32768. + BUFR
   IF ( IPRINT .EQ. 0 ) GO TO 10
   WRITE (LI, 105) I, IBUFR, BUF, BUFR(I)
   IF ( IPRINT .EQ. 0 ) GO TO 10
10 CONTINUE

BUFR(I) = (BUFR*BUFR)/REP)*10000

AVRGE = AVRGE + BUFR(I)
IF ( IPRINT .EQ. 0 ) GO TO 10
WRITE (LI, 106) AVRGE, I, REP, IBUFR, BUF, BUFR(I)
LINES = LINES + 1
IF ( LINES .LE. 20 ) GO TO 99
WRITE (LI, 149) (ICLR, K = 1, LINES)

LINES = 0
IF ( IPRINT .EQ. 0 ) GO TO 10
WRITE (LI, 106) AVRGE, I, REP, IBUFR, BUF, BUFR(I)
CONTINUE

AVRGE = (AVRGE/I)/10000
IF ( IPRINT .EQ. 0 ) GO TO 11
LINES = LINES+1
WRITE (LI, 149) (ICLR, I=1,LINES),
WRITE (LI, 107) ADCHNL,PAMO,PAIR,START,INCR,STOP,REP,AURGE,IPRINT,
*
IF ( LO .EQ. LI ) GO TO 11
WRITE (LO, 107) ADCHNL,PAMO,PAIR,START,INCR,STOP,REP,AURGE,IPRINT,
*
CONTINUE
IF ( IPRINT .NE. 0 ) GO TO 12
WRITE (LI, 108)
WRITE (LI, 149) ICLR
12 RETURN
END
REAL FUNCTION SCANR (LU, ICHAN, K)

Close relay ICHAN on scanner LU and read the instrument indicated by K.

Author: Robert M. Godfarth
Date: February 31, 1979

Detailed program description is available in TXCO log; the variables are:

LU ... LU# of desired scanner (0 or 15).
ICHAN ... Scanner channel (integer).
IC ... Scanner channel (ASCII).
K ... Instrument code (DYM / Counter - 1).

FTN4 COMPILER: HP92060-16092 REV. 1926 (790430)

** NO WARNINGS ** NO ERRORS ** PROGRAM = 00104 COMMON = 00000
SUBROUTINE TIME (IMON,IDAY,IYEAR,IHOUR,IMIN,ISEC)

; Get date and time and convert the variables to ASCII

COMMON / CNTRL / CNTRL
INTEGER ITIME(S)

901 FORMAT (" ERROR DETECTED IN PROGRAM TIME"/
          " CALL EXEC (II,ITIME)"
IMON = 2H$$
IDAY = 2H$$
IYEAR = 2H$$
IHOUR = 2H$$
IMIN = 2H$$
ISEC = 2H$$
CALL EXEC (II+100000B, ITIME)
GO TO 02
CALL ABREG (IA,IB)
IMON = ICON(CNTRL(1),0)
IDAY = ICON(CNTRL(2),0)
IYEAR = ICON(ITIME(4),0)
IHOUR = ICON(ITIME(3),0)
IMIN = ICON(ITIME(2),0)
ISEC = ICON(ITIME(1),0)
RETURN
WRITE ( 6, 901) IA,IB
END

FTN4 COMPILER: HP92060-16092 REV. 1926 (790430)

** NO WARNINGS ** NO ERRORS ** PROGRAM = 00146 COMMON = 00000
SUBROUTINE WAIT (TWAIT)

Causes a defined time delay of TWAIT*10 milliseconds.

Author: Hans M. Zebner
Date: February 13, 1980

Because of the simplicity of the program the program description is included in this box.

TWAIT ... Desired time delay is (TWAIT*10) milliseconds.
TNOW ... Present time.
TM(5) ... Input time buffer (required for EXEC call).
TSTOP ... Final time.

* Causes a defined time delay. Geopfarth, Zebner

INTEGER TM(5)

01 CALL EXEC (T1, TM)
   TNOW = TM(4)
   TSTOP = TM(1) + TM(2)*100 + TM(3)*6000 + TWAIT

02 CALL EXEC (T1, TM)
   THOUR = TM(1) + TM(2)*100 + TM(3)*6000
   IF ( TNOW .NE. THOUR ) GO TO 01
   IF ( TNOW .LT. TSTOP ) GO TO 02
RETURN
END

** NO WARNINGS ** NO ERRORS ** PROGRAM = 00097 COMMON = 06000

FTN4 COMPILER: HP92060-16092 REV. 1926 (790430)
8. DATA REDUCTION PROGRAMS

Three data reduction programs can be initiated from within the TXCO system. They are the following:

(i) **Program REDAB** (Enter 9)

   This program was written to reduce data from the A-B Kulite probe system following the method given in Ref 2, and outputs distributions of velocity magnitude and flow angles.

(ii) **Program REDCO** (Enter 10)

   This program reduces survey data taken with the combination temperature-pneumatic probe and outputs distributions of pressure rise, temperature rise, Mach number, flow angle and losses.

(iii) **Program REDST** (Enter 11)

   This program reduces data taken from fixed instrumentation and outputs the steady-state performance of the compressor.

The above programs are documented separately.
9. CONCLUSIONS AND RECOMMENDATIONS

The large quantity and variety of instrumentation used in the transonic compressor test facility required that data acquisition programs be provided for the different types of data. This was achieved using a particular program structure. Data reduction programs were strictly separated but geared to the acquisition modules through the use of standard data arrays. The data acquisition programs TXCO1, TXCO2 and TXCO3 have been described in detail and the operator commands are explained. The reduction programs are to be documented separately.

The need for easy-to-understand program control leads to a conflict. If interactive messages which explain the program flow and offer menus giving a selection of next logical steps are included, this introduces extended I/O operations and leads to long programs whose speed in execution is slowed considerably by the I/O's. On the other hand the I/O's may be kept to a minimum, which speeds up execution, but this may also lead to communication gaps between the program and the operator. Since the research on the transonic compressor test rig is carried out in large part by visiting researchers and postgraduate students, it was decided to program closer to the first alternative. However, a very useful compromise was achieved through the introduction of the program control array. Should experience in using the TXCO-system show that
the interactive messages, error processing, or the checking for erroneous operator input are too extensive, then the programs should certainly be trimmed.

At the time the programs were written, a graphic software package was not present in the operating system and therefore original plotter software was generated. The switch from "home made" to HP-supported graphics is recommended.

Finally, if the instrumentation system is changed, corresponding changes can be introduced into the appropriate program module, or a new one can be added. Also, the same or a similar program system can easily be adapted for use on any other test rig or calibration apparatus in the laboratory.
APPENDIX A. DATA ACQUISITION WORK SHEETS

A.1. Data Locations
A.2. Steady State Data Array
A.3. Program Control Array (CNTRL)
A.4. Paced Data Array
A.1. DATA LOCATIONS

WORK SHEET: DATA LOCATIONS

<table>
<thead>
<tr>
<th>Port</th>
<th>Scanivalve #1</th>
<th>Scanivalve #2</th>
<th>Scanner #1</th>
<th>Scanner #2</th>
<th>Ch</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>PA-PA</td>
<td>PA-PA</td>
<td>Advance S/V</td>
<td>T1 A/C nozzle</td>
<td>0</td>
</tr>
<tr>
<td>2</td>
<td>FCAL-PA</td>
<td>FCAL-PA</td>
<td>&quot;</td>
<td>&quot;</td>
<td>1</td>
</tr>
<tr>
<td>3</td>
<td>P1 nozzle-PA</td>
<td>P1 comb pr-PA</td>
<td>&quot;</td>
<td>&quot;</td>
<td>2</td>
</tr>
<tr>
<td>4</td>
<td>P1 noz th-PA</td>
<td>P23 comb pr-PA</td>
<td>&quot;</td>
<td>&quot;</td>
<td>3</td>
</tr>
<tr>
<td>5</td>
<td>P1 noz f1-PA</td>
<td>P4 comb pr-PA</td>
<td>Reset S/V</td>
<td>T1 comp noz D</td>
<td>4</td>
</tr>
<tr>
<td>6</td>
<td>PBM-PA</td>
<td>BT2-PA</td>
<td>&quot;</td>
<td>&quot;</td>
<td>5</td>
</tr>
<tr>
<td>7</td>
<td>PTO0-PA</td>
<td>PA-PA</td>
<td>&quot;</td>
<td>&quot;</td>
<td>6</td>
</tr>
<tr>
<td>8</td>
<td>S1-PA</td>
<td>PA-PA</td>
<td>&quot;</td>
<td>&quot;</td>
<td>7</td>
</tr>
<tr>
<td>9</td>
<td>S2-PA</td>
<td>K eq-PA</td>
<td>&quot;</td>
<td>&quot;</td>
<td>8</td>
</tr>
<tr>
<td>10</td>
<td>S3-PA</td>
<td>P alpha-PA</td>
<td>Transducer</td>
<td>T cell</td>
<td>9</td>
</tr>
<tr>
<td>11</td>
<td>S4-PA</td>
<td>C7-PA</td>
<td>&quot;</td>
<td>&quot;</td>
<td>10</td>
</tr>
<tr>
<td>12</td>
<td>S5-PA</td>
<td>A1-PA</td>
<td>&quot;</td>
<td>&quot;</td>
<td>11</td>
</tr>
<tr>
<td>13</td>
<td>S6-PA</td>
<td>B1-PA</td>
<td>&quot;</td>
<td>&quot;</td>
<td>12</td>
</tr>
<tr>
<td>14</td>
<td>S7-PA</td>
<td>C1-PA</td>
<td>&quot;</td>
<td>&quot;</td>
<td>13</td>
</tr>
<tr>
<td>15</td>
<td>S8-PA</td>
<td>A2-PA</td>
<td>RPM A/C</td>
<td>&quot;</td>
<td>14</td>
</tr>
<tr>
<td>16</td>
<td>S9-PA</td>
<td>B2-PA</td>
<td>RPM TTR</td>
<td>&quot;</td>
<td>15</td>
</tr>
<tr>
<td>17</td>
<td>S10-PA</td>
<td>C2-PA</td>
<td>RPM TCR</td>
<td>&quot;</td>
<td>16</td>
</tr>
<tr>
<td>18</td>
<td>S11-PA</td>
<td>A3-PA</td>
<td>RPM TTR</td>
<td>&quot;</td>
<td>17</td>
</tr>
<tr>
<td>19</td>
<td>S12-PA</td>
<td>B3-PA</td>
<td>Blade pass freq</td>
<td>&quot;</td>
<td>18</td>
</tr>
<tr>
<td>20</td>
<td>S13-PA</td>
<td>C3-PA</td>
<td>TTR AXF</td>
<td>&quot;</td>
<td>19</td>
</tr>
<tr>
<td>21</td>
<td>S14-PA</td>
<td>A4-PA</td>
<td>TTR CLAF</td>
<td>&quot;</td>
<td>20</td>
</tr>
<tr>
<td>22</td>
<td>S15-PA</td>
<td>B4-PA</td>
<td>TTR N-Mv</td>
<td>&quot;</td>
<td>21</td>
</tr>
<tr>
<td>23</td>
<td>S16-PA</td>
<td>C4-PA</td>
<td>TTR DyTQ</td>
<td>&quot;</td>
<td>22</td>
</tr>
<tr>
<td>24</td>
<td>S17-PA</td>
<td>A5-PA</td>
<td>TTR StTQ</td>
<td>&quot;</td>
<td>23</td>
</tr>
<tr>
<td>25</td>
<td>PA-PA</td>
<td>B5-PA</td>
<td>P barometric</td>
<td>&quot;</td>
<td>24</td>
</tr>
<tr>
<td>26</td>
<td>PCAL-PA</td>
<td>C5-PA</td>
<td>P1 nozzle comp</td>
<td>&quot;</td>
<td>25</td>
</tr>
<tr>
<td>27</td>
<td>S18-PA</td>
<td>A6-PA</td>
<td>P nozzle comp</td>
<td>&quot;</td>
<td>26</td>
</tr>
<tr>
<td>28</td>
<td>S19-PA</td>
<td>B6-PA</td>
<td>P nozzle turb</td>
<td>&quot;</td>
<td>27</td>
</tr>
<tr>
<td>29</td>
<td>H1-PA</td>
<td>C6-PA</td>
<td>P nozzle turb</td>
<td>&quot;</td>
<td>28</td>
</tr>
<tr>
<td>30</td>
<td>H2-PA</td>
<td>A7-PA</td>
<td>rad pos comb pr</td>
<td>&quot;</td>
<td>29</td>
</tr>
<tr>
<td>31</td>
<td>H3-PA</td>
<td>P bearing-PA</td>
<td>yaw comb pr</td>
<td>&quot;</td>
<td>30</td>
</tr>
<tr>
<td>32</td>
<td>H4-PA</td>
<td>P thrust-PA</td>
<td>rad pos 'A' pr</td>
<td>&quot;</td>
<td>31</td>
</tr>
<tr>
<td>33</td>
<td>H5-PA</td>
<td>PT turb in-PA</td>
<td>yaw 'A' pr</td>
<td>&quot;</td>
<td>32</td>
</tr>
<tr>
<td>34</td>
<td>H6-PA</td>
<td>P st out L-PA</td>
<td>rad pos 'B' pr</td>
<td>&quot;</td>
<td>33</td>
</tr>
<tr>
<td>35</td>
<td>H7-PA</td>
<td>P st out R-PA</td>
<td>yaw 'B' pr</td>
<td>&quot;</td>
<td>34</td>
</tr>
<tr>
<td>36</td>
<td>H8-PA</td>
<td>PT ro out L-PA</td>
<td>Torque TCR</td>
<td>&quot;</td>
<td>35</td>
</tr>
<tr>
<td>37</td>
<td>H9-PA</td>
<td>PT ro out R-PA</td>
<td>KUL ref pres</td>
<td>&quot;</td>
<td>36</td>
</tr>
<tr>
<td>38</td>
<td>H10-PA</td>
<td>P ro out L-PA</td>
<td>&quot;</td>
<td>&quot;</td>
<td>37</td>
</tr>
<tr>
<td>39</td>
<td>H11-PA</td>
<td>P ro out R-PA</td>
<td>&quot;</td>
<td>&quot;</td>
<td>38</td>
</tr>
<tr>
<td>40</td>
<td>Diff T1-PA</td>
<td>PA-PA</td>
<td>&quot;</td>
<td>&quot;</td>
<td>39</td>
</tr>
</tbody>
</table>

Port Scanivalve #1 Scanivalve #2 Scanner #1 Scanner #2 Ch

263
<table>
<thead>
<tr>
<th>Port</th>
<th>Scanivalve #1</th>
<th>Scanivalve #2</th>
<th>Scanner #1</th>
<th>Scanner #2</th>
<th>Ch</th>
</tr>
</thead>
<tbody>
<tr>
<td>46</td>
<td>Diff T7-PA</td>
<td></td>
<td></td>
<td>wall KUL</td>
<td>46</td>
</tr>
<tr>
<td>47</td>
<td>Diff T8-PA</td>
<td></td>
<td></td>
<td>&quot;&quot; K10.5</td>
<td>47</td>
</tr>
<tr>
<td>48</td>
<td>Diff T9-PA</td>
<td></td>
<td></td>
<td>&quot;&quot; K11.</td>
<td>48</td>
</tr>
<tr>
<td>49</td>
<td></td>
<td></td>
<td></td>
<td>&quot;&quot; K12.</td>
<td>49</td>
</tr>
<tr>
<td>50</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>50</td>
</tr>
<tr>
<td>51</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>51</td>
</tr>
<tr>
<td>52</td>
<td></td>
<td></td>
<td></td>
<td>'A' KUL pr</td>
<td>52</td>
</tr>
<tr>
<td>53</td>
<td></td>
<td></td>
<td></td>
<td>'B' KUL pr</td>
<td>53</td>
</tr>
<tr>
<td>54</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>54</td>
</tr>
<tr>
<td>55</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>55</td>
</tr>
<tr>
<td>56</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>56</td>
</tr>
<tr>
<td>57</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>57</td>
</tr>
<tr>
<td>58</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>58</td>
</tr>
<tr>
<td>59</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>59</td>
</tr>
<tr>
<td>60</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>60</td>
</tr>
<tr>
<td>61</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>61</td>
</tr>
<tr>
<td>62</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>62</td>
</tr>
<tr>
<td>63</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>63</td>
</tr>
<tr>
<td>64</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>64</td>
</tr>
<tr>
<td>65</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>65</td>
</tr>
<tr>
<td>66</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>66</td>
</tr>
<tr>
<td>67</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>67</td>
</tr>
<tr>
<td>68</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>68</td>
</tr>
<tr>
<td>69</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>69</td>
</tr>
<tr>
<td>70</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>70</td>
</tr>
<tr>
<td>71</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>71</td>
</tr>
<tr>
<td>72</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>72</td>
</tr>
<tr>
<td>73</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>73</td>
</tr>
<tr>
<td>74</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>74</td>
</tr>
<tr>
<td>75</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>75</td>
</tr>
<tr>
<td>76</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>76</td>
</tr>
<tr>
<td>77</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>77</td>
</tr>
<tr>
<td>78</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>78</td>
</tr>
<tr>
<td>79</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>79</td>
</tr>
</tbody>
</table>

264
A.2. STEADY STATE DATA ARRAY

WORK SHEET TO DECODE THE DATA
ARRAY FOR THE STEADY STATE DATA

File name convention: raw steady state data T4RRSS
reduced steady state data T9RRSS

RR = ASCII converted run #
SS = file sequential # from this run

Note: In some cases the letter T (which stands for Transonic compressor) may be changed to any other character in order to prevent overwriting an existing data file.

Example: T40503 is the third data file from test run #5, that contains steady state data. The data reduction program REDST (Reduction steady state data) creates the file T90403 to take the reduced data from this run.

Due to interface bus problems the data acquisition program STDY sometimes has to be aborted, but may already have produced some valid raw data files. When the program is restarted, it tries to write to data files, whose names already exist. To avoid purging these files, the operator then interactively changes the first character of the data file name to U40503, e.g. The reduced data of course are in file U90503.

By the way, T40503 contains good data. You will find this file in cartridge 29 and can use it to get acquainted with REDST.

DATA (48,4)

1 2 3 4

1

48

CH1(48)  CH2(48)  CH3(48)  CH4(48)
<table>
<thead>
<tr>
<th>1</th>
<th>CH1(I)</th>
<th>CH2(I)</th>
<th>CH3(I)</th>
<th>CH4(I)</th>
<th>I</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>PA-PA</td>
<td>PA-PA</td>
<td>P barometric</td>
<td>n/a</td>
<td>1</td>
</tr>
<tr>
<td>2</td>
<td>PCAL-PA</td>
<td>PCAL-PA</td>
<td>P1 nozzle compr</td>
<td>n/a</td>
<td>2</td>
</tr>
<tr>
<td>3</td>
<td>P1 nozzle-PA</td>
<td>P1 comb pr-PA</td>
<td>P nozzle compr</td>
<td>n/a</td>
<td>3</td>
</tr>
<tr>
<td>4</td>
<td>P1 noz th-PA</td>
<td>P23 comb pr-PA</td>
<td>P1 nozzle turb</td>
<td>TT1T</td>
<td>4</td>
</tr>
<tr>
<td>5</td>
<td>P1 noz fl-PA</td>
<td>P4 comb pr-PA</td>
<td>P nozzle turb</td>
<td>TT3T</td>
<td>5</td>
</tr>
<tr>
<td>6</td>
<td>PBM-PA</td>
<td>PT2-PA</td>
<td>n/a</td>
<td>DTTT</td>
<td>6</td>
</tr>
<tr>
<td>7</td>
<td>PTOO-PA</td>
<td>PT1-PA</td>
<td>rad pos comb pr</td>
<td>MFLOT</td>
<td>7</td>
</tr>
<tr>
<td>8</td>
<td>S1-PA</td>
<td>PA-PA</td>
<td>yaw comb pr</td>
<td>HPT</td>
<td>8</td>
</tr>
<tr>
<td>9</td>
<td>S2-PA</td>
<td>K eq-PA</td>
<td>rad pos 'A' pr</td>
<td>HPM</td>
<td>9</td>
</tr>
<tr>
<td>10</td>
<td>S3-PA</td>
<td>P alpha-PA</td>
<td>yaw 'A' pr</td>
<td>PRTT</td>
<td>10</td>
</tr>
<tr>
<td>11</td>
<td>S4-PA</td>
<td>C7-PA</td>
<td>rad pos 'B' pr</td>
<td>TT1C</td>
<td>11</td>
</tr>
<tr>
<td>12</td>
<td>S5-PA</td>
<td>A1-PA</td>
<td>yaw 'B' pr</td>
<td>TT3C</td>
<td>12</td>
</tr>
<tr>
<td>13</td>
<td>S6-PA</td>
<td>B1-PA</td>
<td>n/a</td>
<td>DTTT</td>
<td>13</td>
</tr>
<tr>
<td>14</td>
<td>S7-PA</td>
<td>C1-PA</td>
<td>Tl A/C nozzle</td>
<td>MFLC</td>
<td>14</td>
</tr>
<tr>
<td>15</td>
<td>S8-PA</td>
<td>A2-PA</td>
<td>T turb in</td>
<td>HPC</td>
<td>15</td>
</tr>
<tr>
<td>16</td>
<td>S9-PA</td>
<td>B2-PA</td>
<td>T turb out L</td>
<td>PRCTR</td>
<td>16</td>
</tr>
<tr>
<td>17</td>
<td>S10-PA</td>
<td>C2-PA</td>
<td>T turb out R</td>
<td>RPMCR</td>
<td>17</td>
</tr>
<tr>
<td>18</td>
<td>S11-PA</td>
<td>A3-PA</td>
<td>n/a</td>
<td>MFLCR</td>
<td>18</td>
</tr>
<tr>
<td>19</td>
<td>S12-PA</td>
<td>B3-PA</td>
<td>Tl comp noz D</td>
<td>TOBQR</td>
<td>19</td>
</tr>
<tr>
<td>20</td>
<td>S13-PA</td>
<td>C3-PA</td>
<td>Tl comp noz W</td>
<td>HPMPR</td>
<td>20</td>
</tr>
<tr>
<td>21</td>
<td>S14-PA</td>
<td>A4-PA</td>
<td>T in sta 00</td>
<td>HPCR</td>
<td>21</td>
</tr>
<tr>
<td>22</td>
<td>S15-PA</td>
<td>B4-PA</td>
<td>T out 44</td>
<td>HPTR</td>
<td>22</td>
</tr>
<tr>
<td>23</td>
<td>S16-PA</td>
<td>C4-PA</td>
<td>T out 44</td>
<td>EFFO</td>
<td>23</td>
</tr>
<tr>
<td>24</td>
<td>S17-PA</td>
<td>A5-PA</td>
<td>T out C4</td>
<td>EFF1</td>
<td>24</td>
</tr>
<tr>
<td>25</td>
<td>PA-PA</td>
<td>B5-PA</td>
<td>T cell</td>
<td>EFF2</td>
<td>25</td>
</tr>
<tr>
<td>26</td>
<td>PCAL-PA</td>
<td>C5-PA</td>
<td>n/a</td>
<td>EFF3</td>
<td>26</td>
</tr>
<tr>
<td>27</td>
<td>S18-PA</td>
<td>A6-PA</td>
<td>( \Delta T ) turb L</td>
<td>n/a</td>
<td>27</td>
</tr>
<tr>
<td>28</td>
<td>S19-PA</td>
<td>B6-PA</td>
<td>( \Delta T ) turb R</td>
<td>n/a</td>
<td>28</td>
</tr>
<tr>
<td>29</td>
<td>H1-PA</td>
<td>C6-PA</td>
<td>( \Delta T ) A4</td>
<td>n/a</td>
<td>29</td>
</tr>
<tr>
<td>30</td>
<td>H2-PA</td>
<td>A7-PA</td>
<td>( \Delta T ) B4</td>
<td>T in ref pr</td>
<td>30</td>
</tr>
<tr>
<td>31</td>
<td>H3-PA</td>
<td>P bearing-PA</td>
<td>( \Delta T ) C4</td>
<td>T comb ref</td>
<td>31</td>
</tr>
<tr>
<td>32</td>
<td>H4-PA</td>
<td>P thrust-PA</td>
<td>n/a</td>
<td>n/a</td>
<td>32</td>
</tr>
<tr>
<td>33</td>
<td>H5-PA</td>
<td>PT turb in-PA</td>
<td>KUL ref pres run</td>
<td>#</td>
<td>33</td>
</tr>
<tr>
<td>34</td>
<td>H6-PA</td>
<td>P st out L-PA</td>
<td>wall KUL K6.</td>
<td>test #</td>
<td>34</td>
</tr>
<tr>
<td>35</td>
<td>H7-PA</td>
<td>P st out R-PA</td>
<td>&quot; &quot; K7.</td>
<td>point #</td>
<td>35</td>
</tr>
<tr>
<td>36</td>
<td>H8-PA</td>
<td>PT ro out L-PA</td>
<td>&quot; &quot; K8.</td>
<td>day</td>
<td>36</td>
</tr>
<tr>
<td>37</td>
<td>H9-PA</td>
<td>PT ro out R-PA</td>
<td>&quot; &quot; K8.5</td>
<td>month</td>
<td>37</td>
</tr>
<tr>
<td>38</td>
<td>H10-PA</td>
<td>P ro out L-PA</td>
<td>&quot; &quot; K9.</td>
<td>year</td>
<td>38</td>
</tr>
<tr>
<td>39</td>
<td>H11-PA</td>
<td>P ro out R-PA</td>
<td>&quot; &quot; K9.5</td>
<td>machine code</td>
<td>39</td>
</tr>
<tr>
<td>40</td>
<td>Diff T1-PA</td>
<td>PA-PA</td>
<td>&quot; &quot; K10.</td>
<td>n/a</td>
<td>40</td>
</tr>
<tr>
<td>41</td>
<td>Diff T2-PA</td>
<td>P diff 1-PA</td>
<td>wall KUL K10.5</td>
<td>n/a</td>
<td>41</td>
</tr>
<tr>
<td>42</td>
<td>Diff T3-PA</td>
<td>P diff 2-PA</td>
<td>&quot; &quot; K11.</td>
<td>n/a</td>
<td>42</td>
</tr>
<tr>
<td>43</td>
<td>Diff T4-PA</td>
<td>P diff 3-PA</td>
<td>&quot; &quot; K12.</td>
<td>case angle</td>
<td>43</td>
</tr>
<tr>
<td>44</td>
<td>Diff T5-PA</td>
<td>P diff 4-PA</td>
<td>&quot; &quot; K13.</td>
<td>n/a</td>
<td>44</td>
</tr>
<tr>
<td>45</td>
<td>Diff T6-PA</td>
<td>P diff 5-PA</td>
<td>&quot; &quot; K14.</td>
<td>n/a</td>
<td>45</td>
</tr>
<tr>
<td>46</td>
<td>Diff T7-PA</td>
<td>n/a</td>
<td>n/a</td>
<td>RPM</td>
<td>46</td>
</tr>
<tr>
<td>47</td>
<td>Diff T8-PA</td>
<td>n/a</td>
<td>'A' KUL pr</td>
<td>Torque</td>
<td>47</td>
</tr>
<tr>
<td>48</td>
<td>Diff T9-PA</td>
<td>n/a</td>
<td>'B' KUL pr</td>
<td>n/a</td>
<td>48</td>
</tr>
</tbody>
</table>

266
### A.3. PROGRAM CONTROL ARRAY (CNTRL)

**WORK SHEET TO DECODE/ENCODE THE CONTROL ARRAY CNTRL:**

<table>
<thead>
<tr>
<th>CNTRL(1)</th>
<th>Month of the test run.</th>
</tr>
</thead>
<tbody>
<tr>
<td>CNTRL(2)</td>
<td>Day of the test run.</td>
</tr>
<tr>
<td>CNTRL(3)</td>
<td>Year of the test run.</td>
</tr>
<tr>
<td>CNTRL(4)</td>
<td>Test run #.</td>
</tr>
<tr>
<td>CNTRL(5)</td>
<td>Test # of this run.</td>
</tr>
<tr>
<td>CNTRL(6)</td>
<td>Point # of this test.</td>
</tr>
<tr>
<td>CNTRL(19)</td>
<td>LU# of the standard interactive input device.</td>
</tr>
<tr>
<td>CNTRL(20)</td>
<td>LU# of the standard output device.</td>
</tr>
<tr>
<td>CNTRL(21)</td>
<td>LU# of the optional output device.</td>
</tr>
<tr>
<td>CNTRL(22)</td>
<td>LU# of the plotter.</td>
</tr>
<tr>
<td>CNTRL(30)</td>
<td>Cartridge reference # for data files.</td>
</tr>
<tr>
<td>CNTRL(31)</td>
<td>Security code for data files.</td>
</tr>
<tr>
<td>CNTRL(32)</td>
<td>First and second character of data file name; IFILE(1)</td>
</tr>
<tr>
<td>CNTRL(33)</td>
<td>Third and fourth character of data file name; IFILE(2)</td>
</tr>
<tr>
<td>CNTRL(34)</td>
<td>Fifth and sixth character of data file name; IFILE(3)</td>
</tr>
<tr>
<td>CNTRL(36)</td>
<td>Initializes fast steady state data reduction run, if set to 1.</td>
</tr>
<tr>
<td>CNTRL(37)</td>
<td>Suppresses printing of heading in subroutines FREER and PACER, if set to 1.</td>
</tr>
<tr>
<td>CNTRL(39)</td>
<td>Suppresses creating/opening and closing of files in subroutines FREER and PACER, if set to 1.</td>
</tr>
<tr>
<td>CNTRL(40)</td>
<td>Suppresses analog output of just acquired paced run data to terminal, if set to 1.</td>
</tr>
<tr>
<td>CNTRL(41)</td>
<td>100*Factor to vary size in X-direction of a drawing.</td>
</tr>
<tr>
<td>CNTRL(42)</td>
<td>100*Factor to vary size in Y-direction of a drawing.</td>
</tr>
<tr>
<td>CNTRL(50)</td>
<td>Indicates the son program to be scheduled and the subroutine to be called therefrom.</td>
</tr>
</tbody>
</table>

01 ... Schedule TXCO1 and call ABSRV  
02 ... " " " " CALIB  
03 ... " " " " FREER  
04 ... " " " " PACER  
05 ... Schedule TXCO2 and call COMB  
06 ... " " " " STDY  
07 ... Schedule TXCO3 and call CHECK  
08 ... " " " " CHNGE  
09 ... Schedule REDAB  
10 ... Schedule REDCO  
11 ... Schedule REDST.  

| CNTRL(61) | Number of S/V controller #I. |
| CNTRL(62) | Number of S/V controller #II. |
| CNTRL(63) | Number of S/V controller #III. |
| CNTRL(64) | Number of S/V controller #IV. |
| CNTRL(65) | Number of S/V controller #V. |
| CNTRL(71) | LU# of scanner #1. |
| CNTRL(72) | LU# of scanner #2. |
Accounting variable subroutine ABSRV: output page 

Blade pair (1 - 9), if Pacer is operated in Mode 2.

Start count for data acquisition using Pacer encode.

Increment for data acquisition using Pacer encode.

Stop count for data acquisition using Pacer encode.

A/D input channel for KULITE type 'A' probe.

A/D input channel for KULITE type 'B' probe.

Wall KULITE K6.

K7.

K8.

K8.5.

K9.

K9.5.

K10.

K10.5.

K11.

K12.

K13.

K14.

Character used for analog display in subroutine PICTR.

Total # of multiples of 10ms for S/V controller time delay.

Total # of free run measurements (max. 1664).
### A.4. Paced Data Array

<table>
<thead>
<tr>
<th>Channel 1</th>
<th>Channel 2</th>
<th>Channel 3</th>
<th>Channel 4</th>
<th>Channel 5</th>
<th>Channel 6</th>
<th>Channel 7</th>
<th>Channel 8</th>
<th>Channel 9</th>
<th>Channel 10</th>
<th>Channel 11</th>
<th>Channel 12</th>
</tr>
</thead>
<tbody>
<tr>
<td>( T )</td>
<td>( T )</td>
<td>( T )</td>
<td>( T )</td>
<td>( T )</td>
<td>( T )</td>
<td>( T )</td>
<td>( T )</td>
<td>( T )</td>
<td>( T )</td>
<td>( T )</td>
<td>( T )</td>
</tr>
<tr>
<td>( W )</td>
<td>( W )</td>
<td>( W )</td>
<td>( W )</td>
<td>( W )</td>
<td>( W )</td>
<td>( W )</td>
<td>( W )</td>
<td>( W )</td>
<td>( W )</td>
<td>( W )</td>
<td>( W )</td>
</tr>
</tbody>
</table>

*Note: Each box represents a bit position.*

One 12 bit record.

---

Additional notes:

- One multi-record data file.
- Copy available to DTIC does not permit fully legible reproduction.

Page 269
APPENDIX B. LINEAR APPROXIMATION BY

METHOD OF LEAST SQUARES

Data: $X_i$ and $Y_i$ ; $i = 1, ..., \text{NPNTSI}$

Equation: $Y = C_1 + C_2 \cdot X$

Difference for Each Data Point: $R_i = Y_i - f(X_i) ; i = 1, ..., \text{NPNTSI}$

Sum of Squares of Differences: $R = \sum_{i=1}^{\text{NPNTSI}} R_i^2 = \left( \sum_{i=1}^{\text{NPNTSI}} Y_i - (C_1 + C_2 \cdot X_i) \right)^2$

The value of $R$ depends on the values of the coefficients $C_1$ and $C_2$. In order to determine a minimum value for $R$, the expression for $R$ is partially differentiated with respect to $C_1$ and $C_2$ and the two derivatives are equated to zero. Differentiating,

$$\frac{\partial R}{\partial C_1} = \sum_{i=1}^{\text{NPNTSI}} 2 \cdot \left[ Y_i - (C_1 + C_2 \cdot X_i) \right] \cdot (-1)$$

and
\[
\frac{3R}{3C_2} = \sum_{i=1}^{2R} [Y_i - (C_1 + C_2 \cdot X_i)] \cdot (-X_i)
\]

Setting each expression to zero,
\[
NPNTSI
\cdot \sum_{i=1}^{NPNTSI} (Y_i - C_1 - C_2 \cdot X_i) = 0
\]
\[
NPNTSI
\sum_{i=1}^{NPNTSI} (Y_i \cdot X_i - C_1 \cdot X_i - C_2 \cdot X_i^2) = 0
\]

This gives two equations in which \(C_1\) and \(C_2\) are the only unknowns. Omitting the limits of summation for simplicity,
\[
\sum C_1 + \sum C_2 X_i = \sum Y_i
\]
\[
\sum C_1 X_i + \sum C_2 X_i^2 = \sum Y_i \cdot X_i
\]
or, in matrix notation (note that \(C_1\) and \(C_2\) are constants)
\[
\begin{bmatrix}
NPNTSI \\
\sum X_i \\
\sum X_i^2 \\
\end{bmatrix}
\begin{bmatrix}
C_1 \\
C_2 \\
\end{bmatrix}
= 
\begin{bmatrix}
\sum Y_i \\
\sum Y_i \cdot X_i \\
\end{bmatrix}
\]
or
\[
A \cdot C = B
\]

The components of the matrix \(C\) are obtained using
\[
a_{11} = NPNTSI
\]
\[
a_{12} = a_{21} = \sum X_i
\]
\[
a_{22} = \sum X_i^2
\]
\[
b_1 = \sum Y_i
\]
\[
b_2 = \sum Y_i \cdot X_i
\]
\[ c_1 = \frac{\begin{vmatrix} b_1 & a_{12} \\ b_2 & a_{22} \\ a_{11} & a_{12} \\ a_{12} & a_{22} \end{vmatrix}}{a_{11}a_{22} - a_{12}^2} = \frac{a_{22}b_1 - a_{12}b_2}{a_{11}a_{22} - a_{12}^2} \]

\[ c_2 = \frac{a_{11}b_2 - a_{12}b_1}{a_{11}a_{22} - a_{12}^2} \]

\[ c_1 = \frac{\sum X_i^2 \cdot \sum Y_i - \sum X_i \cdot \sum (Y_i \cdot X_i)}{NPNTSI \cdot \sum X_i^2 - (\sum X_i)^2} \]

\[ c_2 = \frac{NPNTSI \cdot \sum (Y_i \cdot X_i) - \sum X_i \cdot \sum Y_i}{NPNTSI \cdot \sum X_i^2 - (\sum X_i)^2} \]
REFERENCES


<table>
<thead>
<tr>
<th>No.</th>
<th>Name and Address</th>
<th>Copies</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.</td>
<td>Library, Naval Postgraduate School, Monterey, California 93940</td>
<td>4</td>
</tr>
<tr>
<td>2.</td>
<td>Office of Research Administration, Naval Postgraduate School, Monterey, California 93940</td>
<td>1</td>
</tr>
<tr>
<td>3.</td>
<td>Chairman, Department of Aeronautics, Naval Postgraduate School, Monterey, California 93940</td>
<td>1</td>
</tr>
<tr>
<td>4.</td>
<td>Director, Turbopropulsion Laboratory, Department of Aeronautics, Naval Postgraduate School, Monterey, California 93940</td>
<td>30</td>
</tr>
<tr>
<td>5.</td>
<td>Dr. Gerhard Heiche, Naval Air Systems Command, Navy Department, Washington, D.C. 20360</td>
<td>1</td>
</tr>
<tr>
<td>6.</td>
<td>Mr. Karl H. Guttmann, Naval Air Systems Command, Navy Department, Washington, D.C. 20360</td>
<td>1</td>
</tr>
<tr>
<td>7.</td>
<td>Dr. A. D. Wood, Office of Naval Research, Arlington, Virginia 22217</td>
<td>1</td>
</tr>
<tr>
<td>8.</td>
<td>Commanding Officer, Naval Air Propulsion Test Center, Trenton, New Jersey 08628</td>
<td>1</td>
</tr>
</tbody>
</table>
9. National Aeronautics & Space Administration
   Lewis Research Center (Library)
   2100 Brookpark Road
   Cleveland, Ohio 44135

10. CAG Library
    The Boeing Company
    Seattle, Washington 98124

11. Library
    General Electric Company
    Aircraft Engine Technology Division
    DTO Mail Drop H43
    Cincinnati, Ohio 45215

12. Library
    Pratt and Whitney Aircraft
    Post Office Box 2691
    West Palm Beach, Florida 33402

13. Library
    Pratt and Whitney Aircraft
    East Hartford, Connecticut 06108

14. Chief, Fan and Compressor Branch
    Mail Stop 5-9
    NASA Lewis Research Center
    2100 Brookpark Road
    Cleveland, Ohio 44135

15. Prof. D. Adler
    Technion Israel Institute of Technology
    Department of Mechanical Engineering
    Haifa 32000
    ISRAEL

16. Director, Whittle Laboratory
    Department of Engineering
    Cambridge University
    ENGLAND

17. Prof. F. A. E. Breugelmans
    Institut von Karman de la Dynamique des Fluides
    72 Chausee de Waterloo
    1640 Rhode-St. Genese
    BELGIUM

18. Library
    Air Research Mfg. Corporation
    Division of Garrett Corporation
    402 South 36th Street
    Phoenix, Arizona 85034
19. Prof. Jacques Chauvin  
Universite D'Aix-Marseille  
1 Rue Honnorat  
Marseille, FRANCE

20. Mr. James V. Davis  
Teledyne CAE  
1330 Laskey Road  
Toledo, Ohio 43601

21. Dr. Robert P. Dring  
United Technologies Research Labs  
400 Main Street  
Hartford, Connecticut 06108

22. Mr. Jean Fabri  
ONERA  
29, Ave. de la Division Leclerc  
92 Chatillon  
FRANCE

23. Prof. Dr. Ing Heinz E. Gallus  
Lehrstuhl und Institut fur Strahlantiebe und Turbourbeitsmaschinen  
Rhein.-Westf. Techn. Hochschule Aachen  
Templergraben 55  
5100 Aachen, WEST GERMANY

24. Professor J. P. Gostelow  
School of Mechanical Engineering  
The New South Wales Institute of Technology  
AUSTRALIA

25. Dr. Ing. Hans-J. Heinemann  
DFVLR-AVA  
Bunsenstrasse 10  
3400 Gottingen, WEST GERMANY

26. Professor Ch. Hirsch  
Vrije Universiteit Brussel  
Pleinlaan 2  
1050 Brussels, BELGIUM

27. Chairman  
Aeronautics and Astronautics Department  
31-265 Massachusetts Institute of Technology  
Cambridge, Massachusetts 02139

28. Dr. B. Lakshminarayana  
Professor of Aerospace Engineering  
The Pennsylvania State University  
233 Hammond Building  
University Park, Pennsylvania 16802

276
29. Mr. R. A. Langworthy  
Army Aviation Material Laboratories  
Department of the Army  
Fort Eustis, Virginia 23604

30. Prof. Dr. L. G. Napolitano  
Director, Institute of Aerodynamics  
University of Naples  
Viale C. Augusto  
80125 Napoli  
ITALY

31. Prof. Erik Nilsson  
Institutionen for Stromningsmaskinteknik  
Chalmers Tekniska Hogskola  
Fack, 402 20 Goteborg 5  
SWEDEN

32. Prof. Gordon C. Oates  
Department of Aeronautics and Astronautics  
University of Washington  
Seattle, Washington 98105

33. Prof. Walter F. O'Brian  
Mechanical Engineering Department  
Virginia Polytechnic Institute and State University  
Blacksburg, Virginia 24061

34. Dr. P. A. Paranjee  
Head, Propulsion Division  
National Aeronautics Laboratory  
Post Bag 1799  
Bangalore - 17, INDIA

35. R. E. Peacock, Coda 67Pc  
Department of Aeronautics  
Naval Postgraduate School  
Monterey, California 93940

36. Dr. W. Schlachter  
Brown, Boveri-Sulzer Turbomachinery Ltd.  
Dept. TDE  
Escher Wyss Platz  
CH-8023 Zurich, SWITZERLAND

37. Prof. T. H. Okiishi  
Professor of Mechanical Engineering  
208 Mechanical Engineering Building  
Iowa State University  
Ames, Iowa 50011
38. Dr. Fernando Sisto  
Professor and Head of Mechanical Engineering Department  
Stevens Institute of Technology  
Castle Point, Hoboken, New Jersey 07030

39. Dr. Leroy H. Smith, Jr.  
Manager, Compressor and Fan Technology Operation  
General Electric Company  
Aircraft Engine Technology Division  
DTO Mail Drop H43  
Cincinnati, Ohio 45215

40. Dr. W. Tabakoff  
Professor, Department of Aerospace Engineering  
University of Cincinnati  
Cincinnati, Ohio 45221

41. Mr. P. Tramm  
Manager, Research Labs  
Detroit Diesel Allison Division  
General Motors  
P.O. Box 894  
Indianapolis, Indiana 46206

42. Prof. Dr. W. Traupel  
Institut fur Thermische Turbomaschinen  
Eidg. Technische Hochschule  
Sonneckstrasse 3  
8006 Zurich, SWITZERLAND

43. Dr. Arthur J. Wennerstrom  
ARL/LF  
Wright-Patterson AFB  
Dayton, Ohio 45433

44. Dr. H. Weyer  
DFVLR  
Linder Hohe  
505 Porz-Wahn  
WEST GERMANY

45. Mr. P. F. Yaggy  
Director  
U.S. Army Aeronautical Research Laboratory  
AMES Research Center  
Moffett Field, California 94035

278
46. Prof. C. H. Wu
   P.O. Box 2706
   Beijing 100080
   CHINA

47. Director
   Gas Turbine Establishment
   P.O. Box 305
   Jiangyou County
   Sichuan Province
   CHINA

48. Professor Leonhard Fottner
    Department of Aeronautics and Astronautics
    German Armed Forces University
    Hochschule des Bundeswehr
    Werner Heisenbergweg 39
    8014 Neubiberg near Munich
    WEST GERMANY

49. BDM Corporation
    P.O. Box 2019
    Monterey, California 93940