DEVELOPMENT OF AN ADAPTIVE KALMAN FILTER FOR ESTIMATION IN CHEMICAL PLANT

APPENDICES AND BIBLIOGRAPHY.

A thesis submitted for the degree of

Doctor of Philosophy

by

ROBERT NORMAN WEBB

Department of Chemical Engineering
University of Aston in Birmingham

October 1977

APPENDICES

APPENDIX A

RESULTS OF THE SIMULATION STUDIES OF THE KALMAN FILTER

CONTENTS

FIGURE A-1

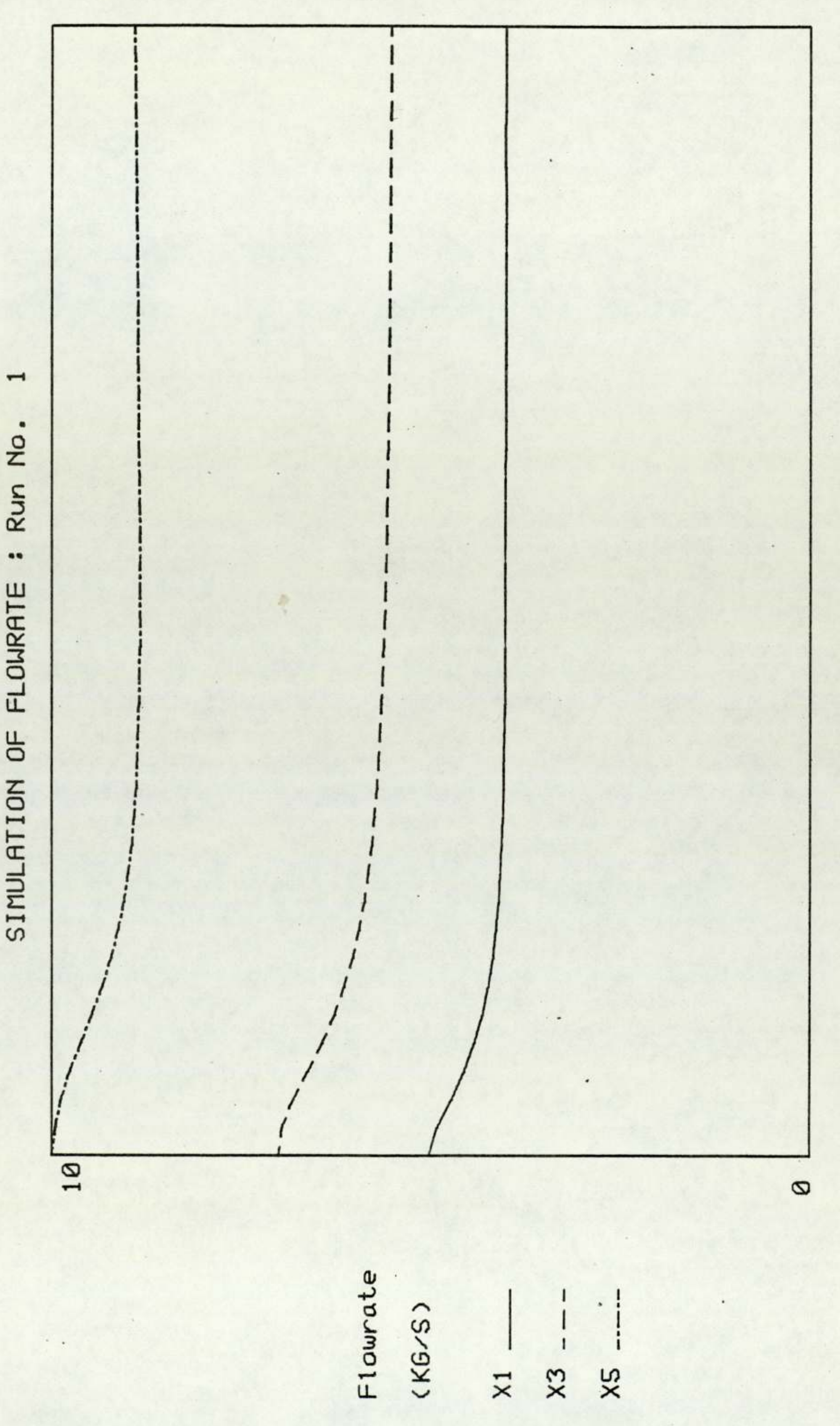

SIMULATION OF FLOWRATE : Run No. 1
10
0
Flowrate
(KG/S)
X1
X3
X5
0
10
Time (HRS.)

FIGURE A-2

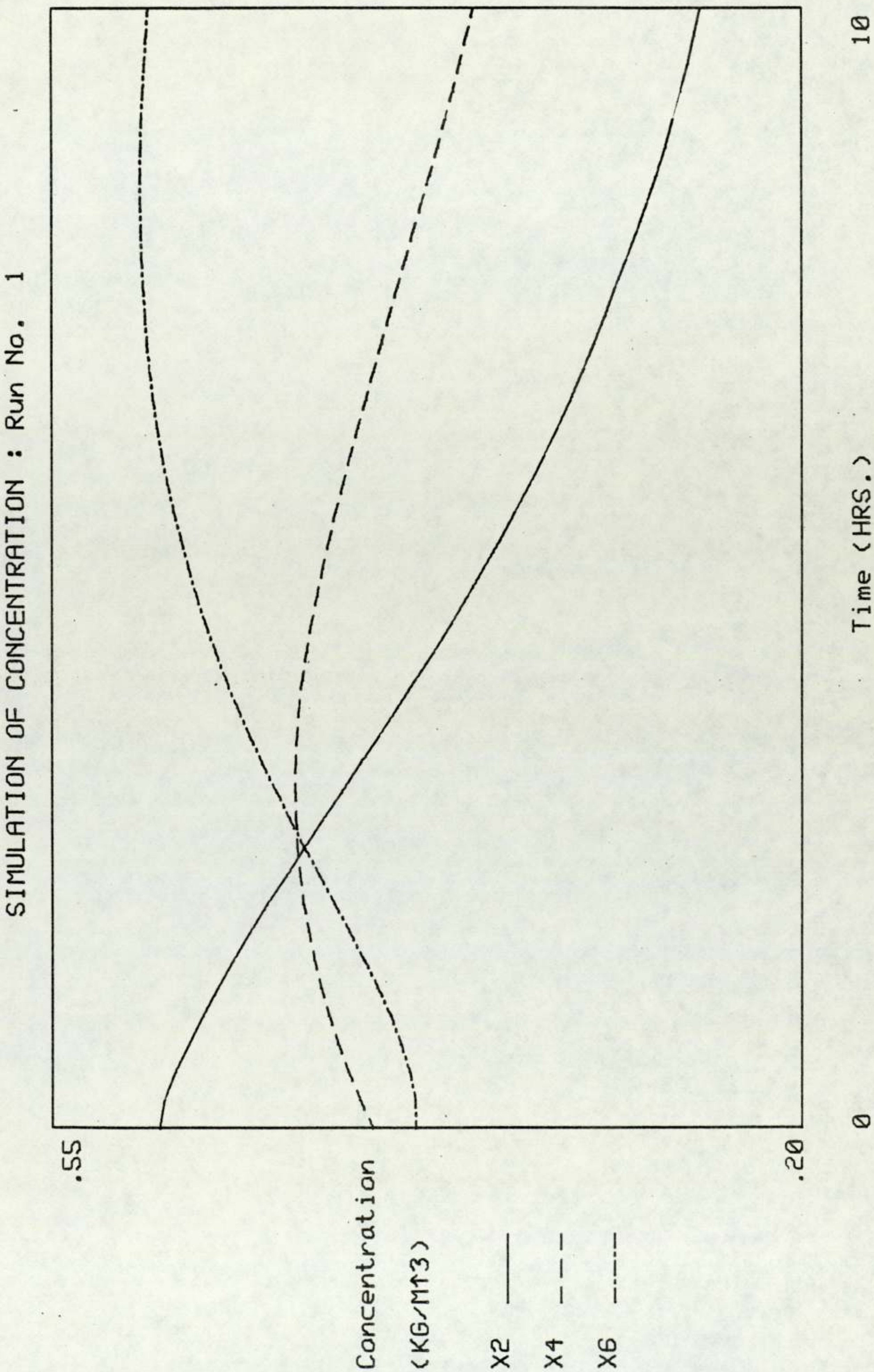
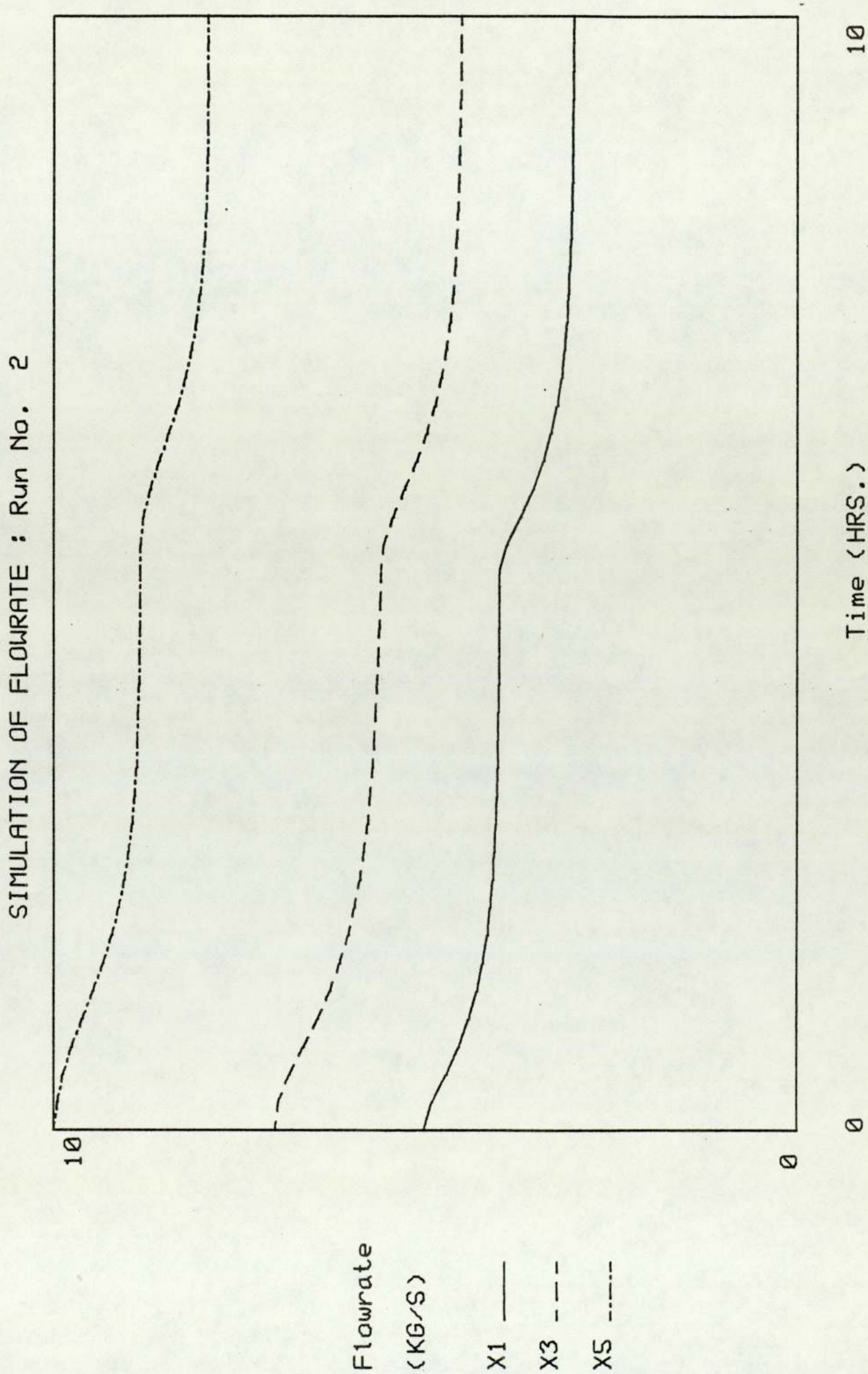

SIMULATION OF CONCENTRATION : Run No. 1
.55
.20
0
10
Time (HRS.)
Concentration
(KG/M↑3)
X2
X4
X6

FIGURE A-3



SIMULATION OF FLOWRATE : Run No. 2
Flowrate
(KG/S)
X1
X3
X5
10
0
0
10
Time (HRS.)

FIGURE A-4

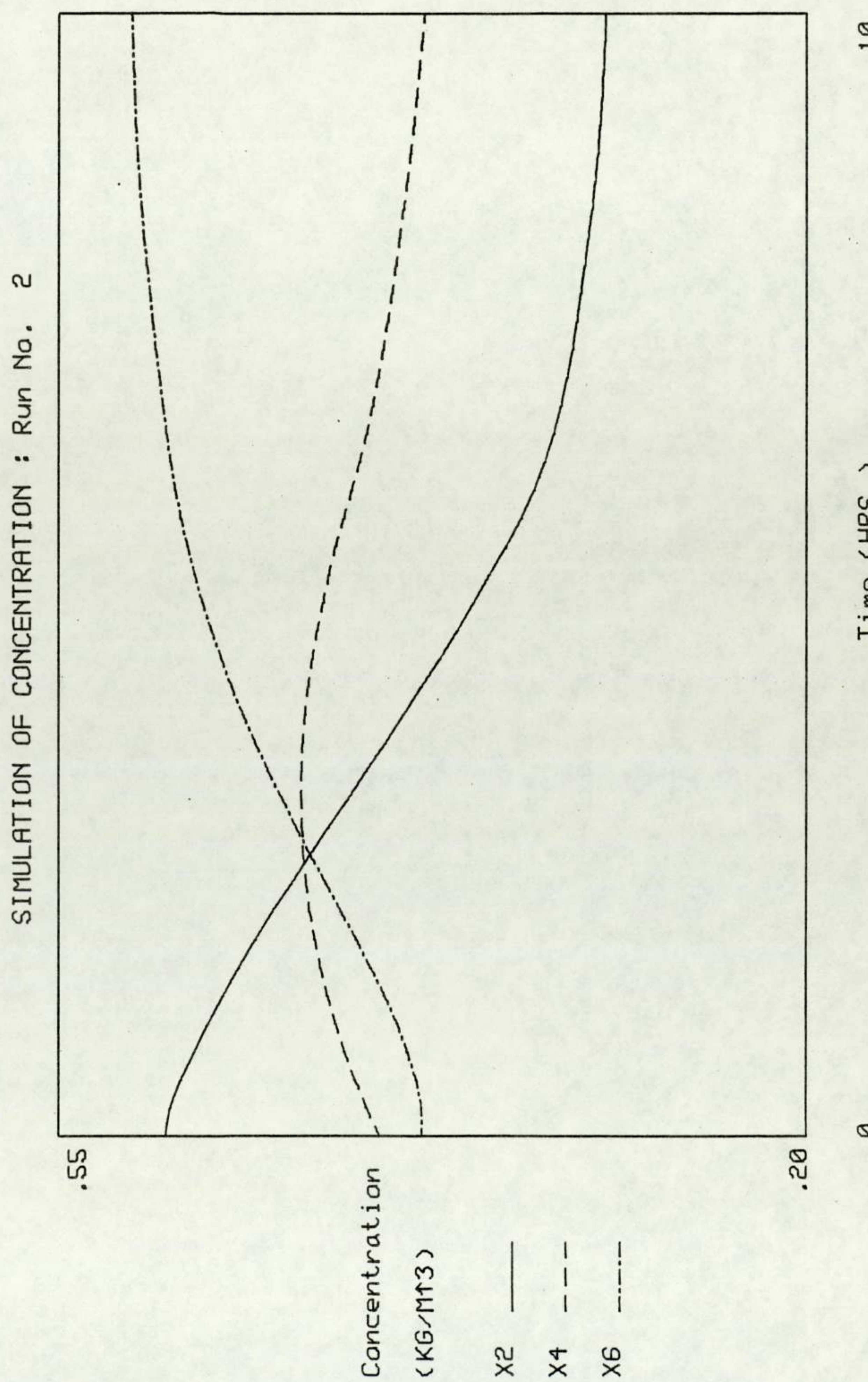
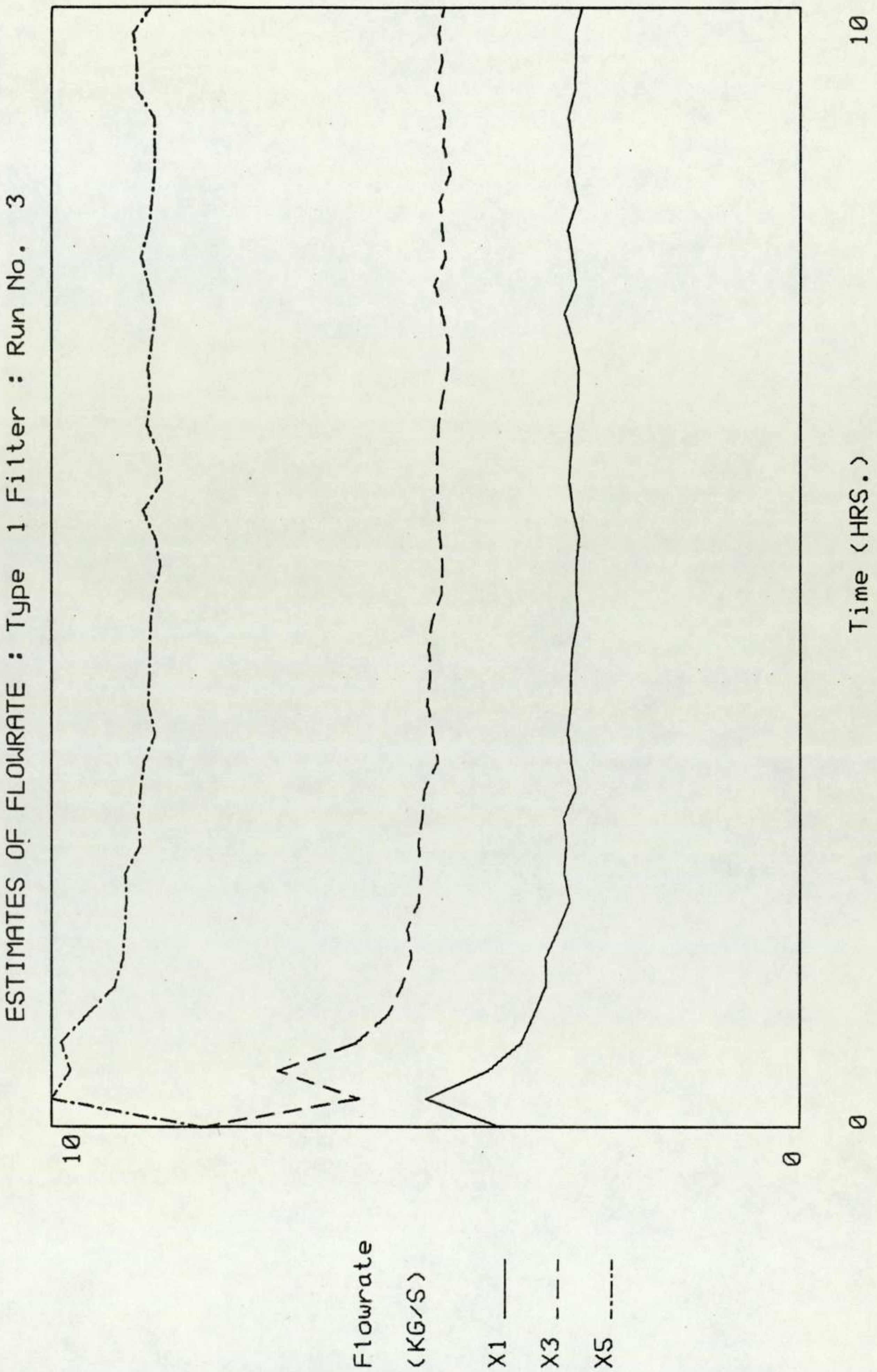
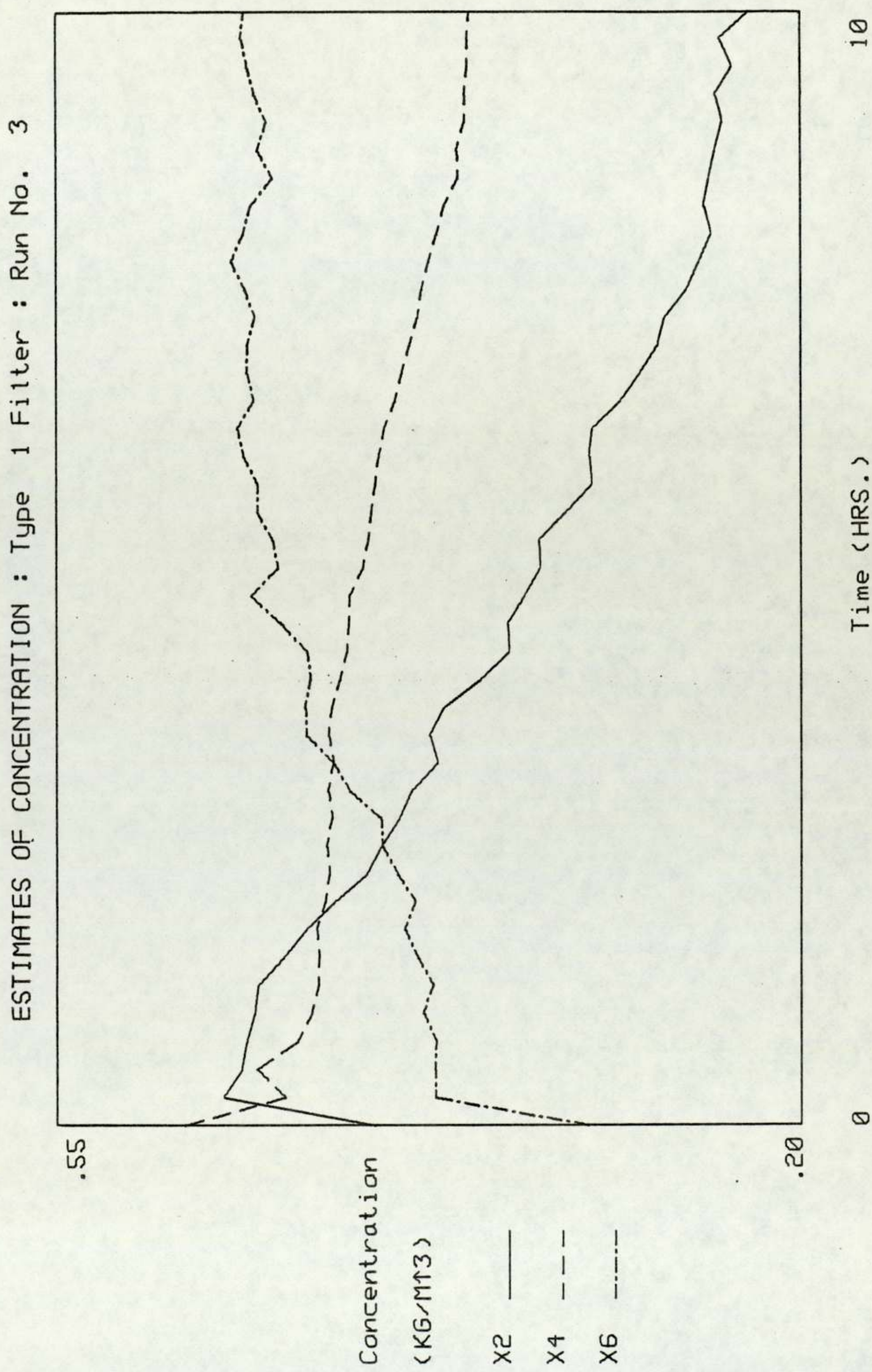

SIMULATION OF CONCENTRATION : Run No. 2
.55
.20
Concentration
(KG/M↑3)
X2
X4
X6
0
10
Time (HRS.)

FIGURE A-5



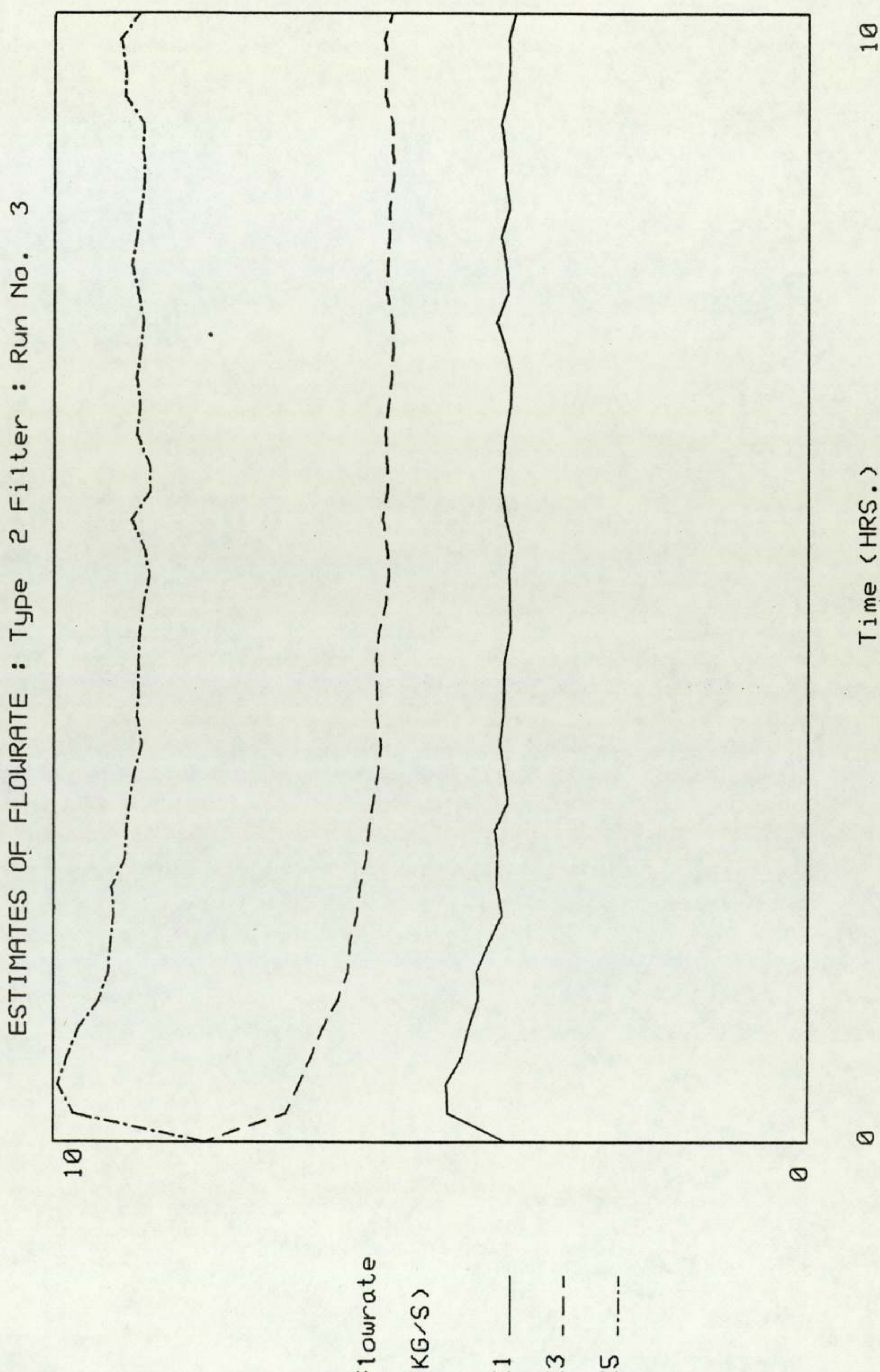
ESTIMATES OF FLOWRATE : Type 1 Filter : Run No. 3
Flowrate
(KG/S)
X1
X3
X5
10
0
0
10
Time (HRS.)

FIGURE A-6



ESTIMATES OF CONCENTRATION : Type 1 Filter : Run No. 3
Concentration
(KG/M↑3)
.55
.20
0
10
Time (HRS.)
X2
X4
X6

FIGURE A-7



ESTIMATES OF FLOWRATE : Type 2 Filter : Run No. 3
Flowrate
(KG/S)
10
0
0
10
Time (HRS.)
X1
X3
X5

FIGURE A-8

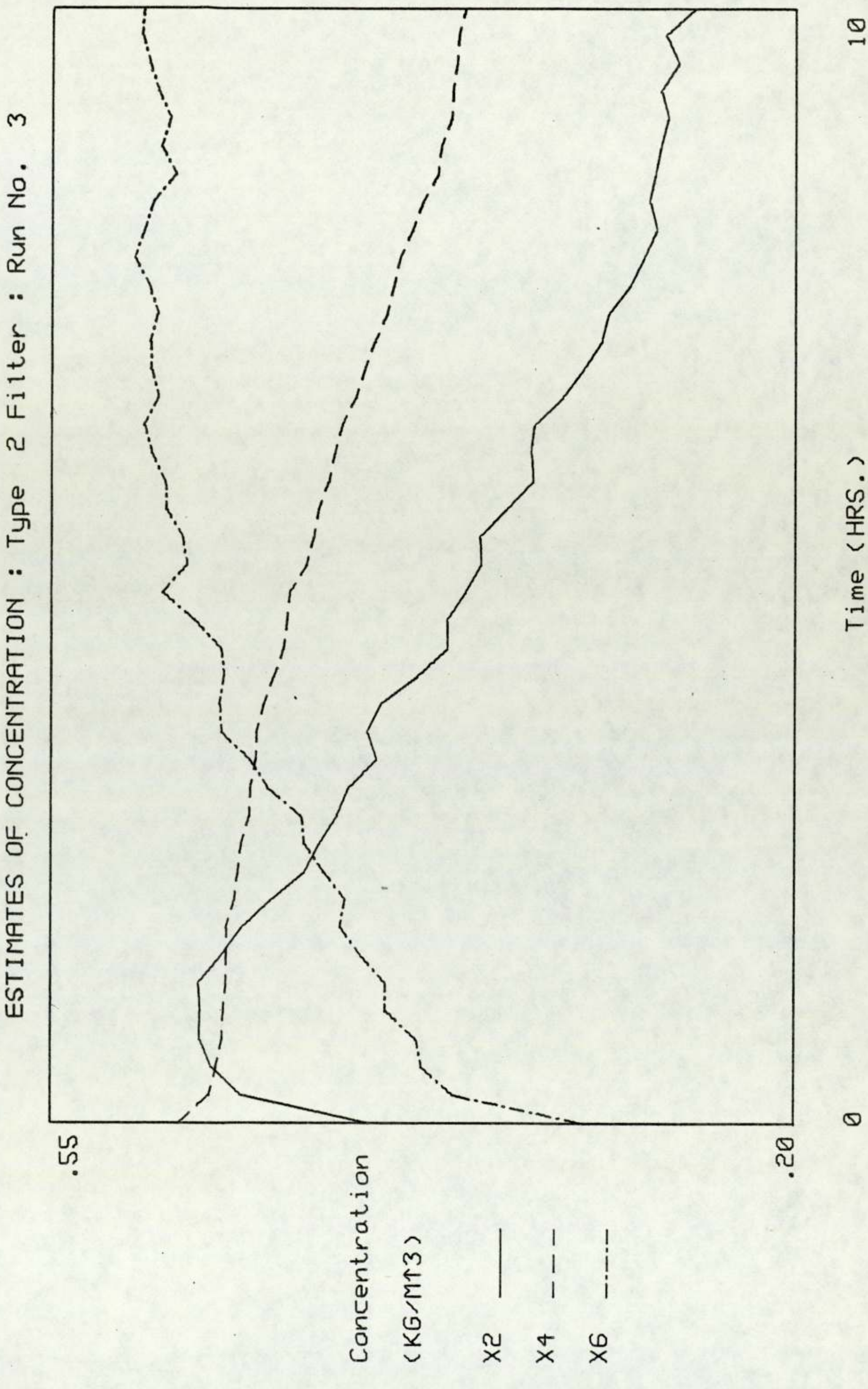

ESTIMATES OF CONCENTRATION : Type 2 Filter : Run No. 3
Concentration
(KG/M↑3)
.55
.20
0
10
Time (HRS.)
X2
X4
X6

FIGURE A-9

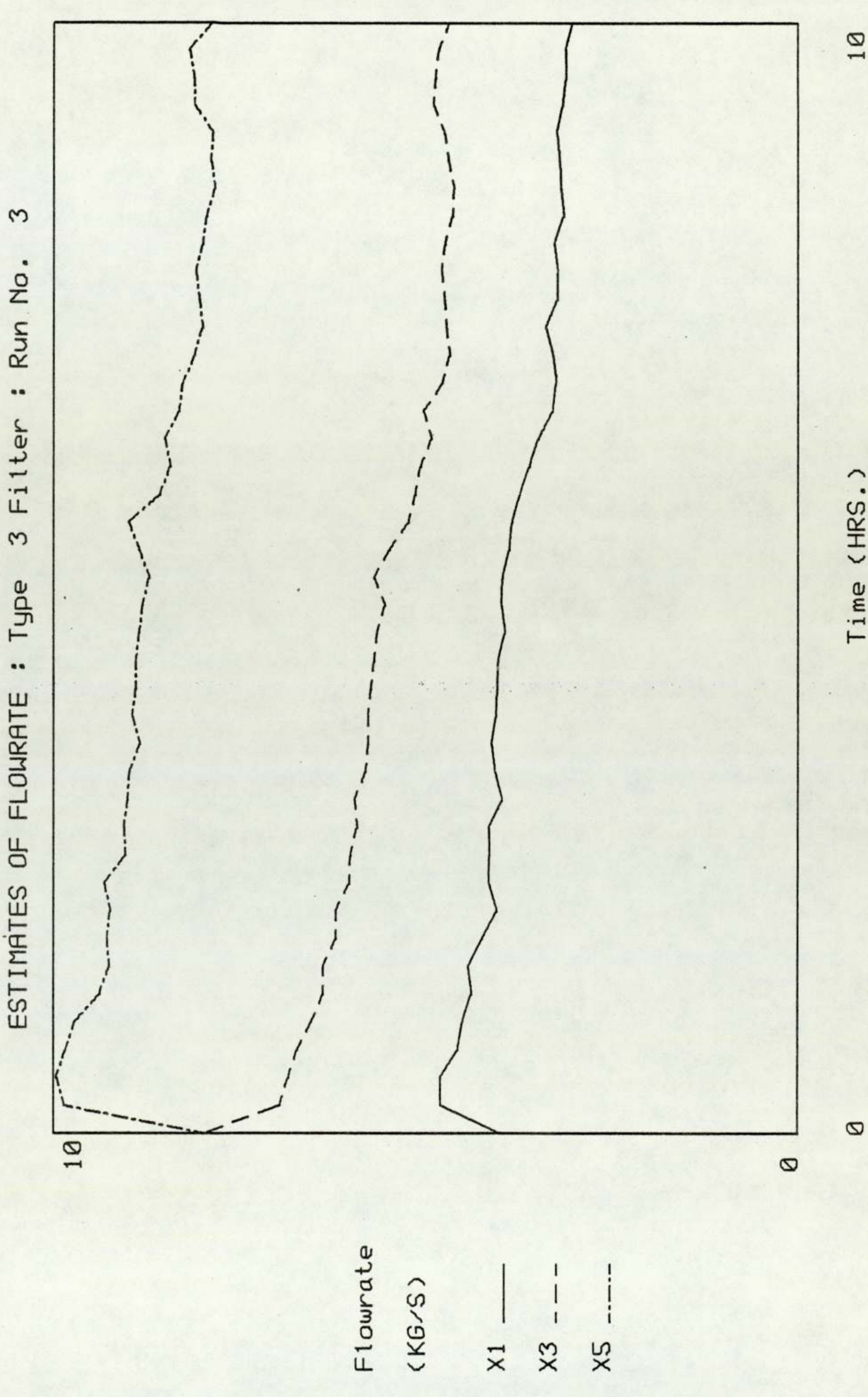

ESTIMATES OF FLOWRATE : Type 3 Filter : Run No. 3
Flowrate
(KG/S)
X1
X3
X5
10
0
0
10
Time (HRS.)

FIGURE A-10

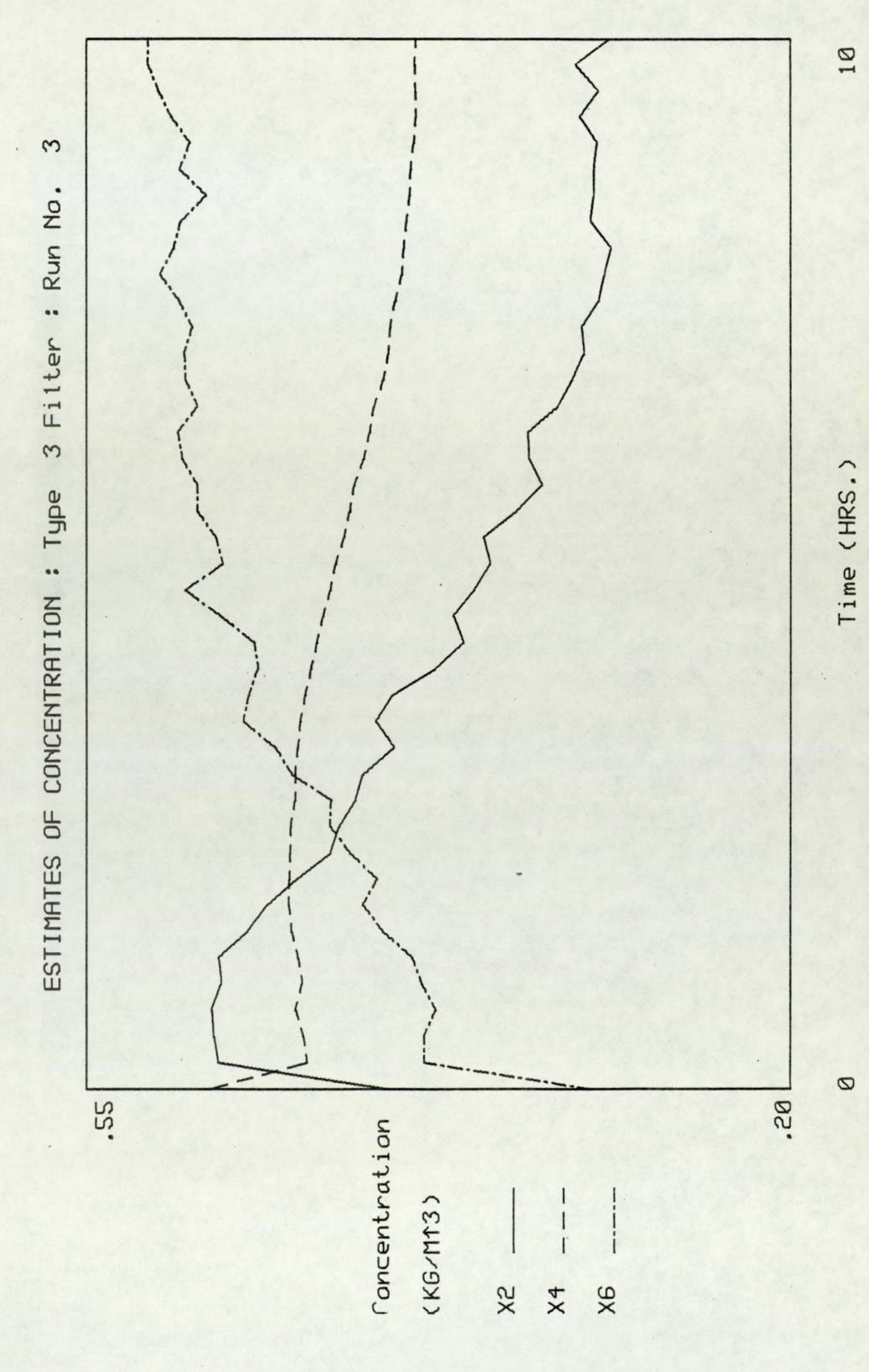

ESTIMATES OF CONCENTRATION : Type 3 Filter : Run No. 3
.55
.20
0
10
Time (HRS.)
Concentration
(KG/M↑3)
X2
X4
X6

FIGURE A-11

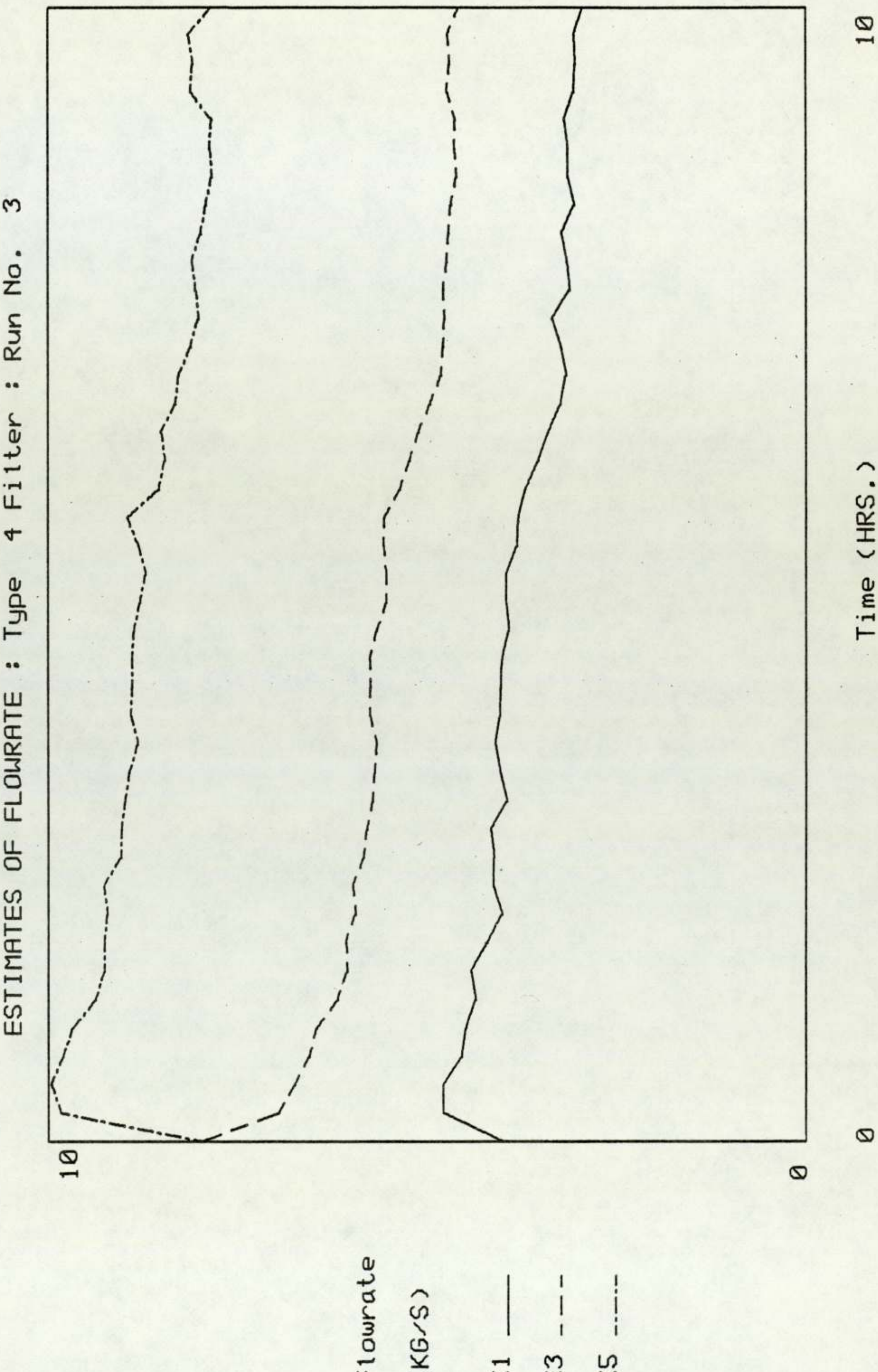

ESTIMATES OF FLOWRATE : Type 4 Filter : Run No. 3
10
0
0
10
Time (HRS.)
Flowrate
(KG/S)
X1
X3
X5

FIGURE A-12

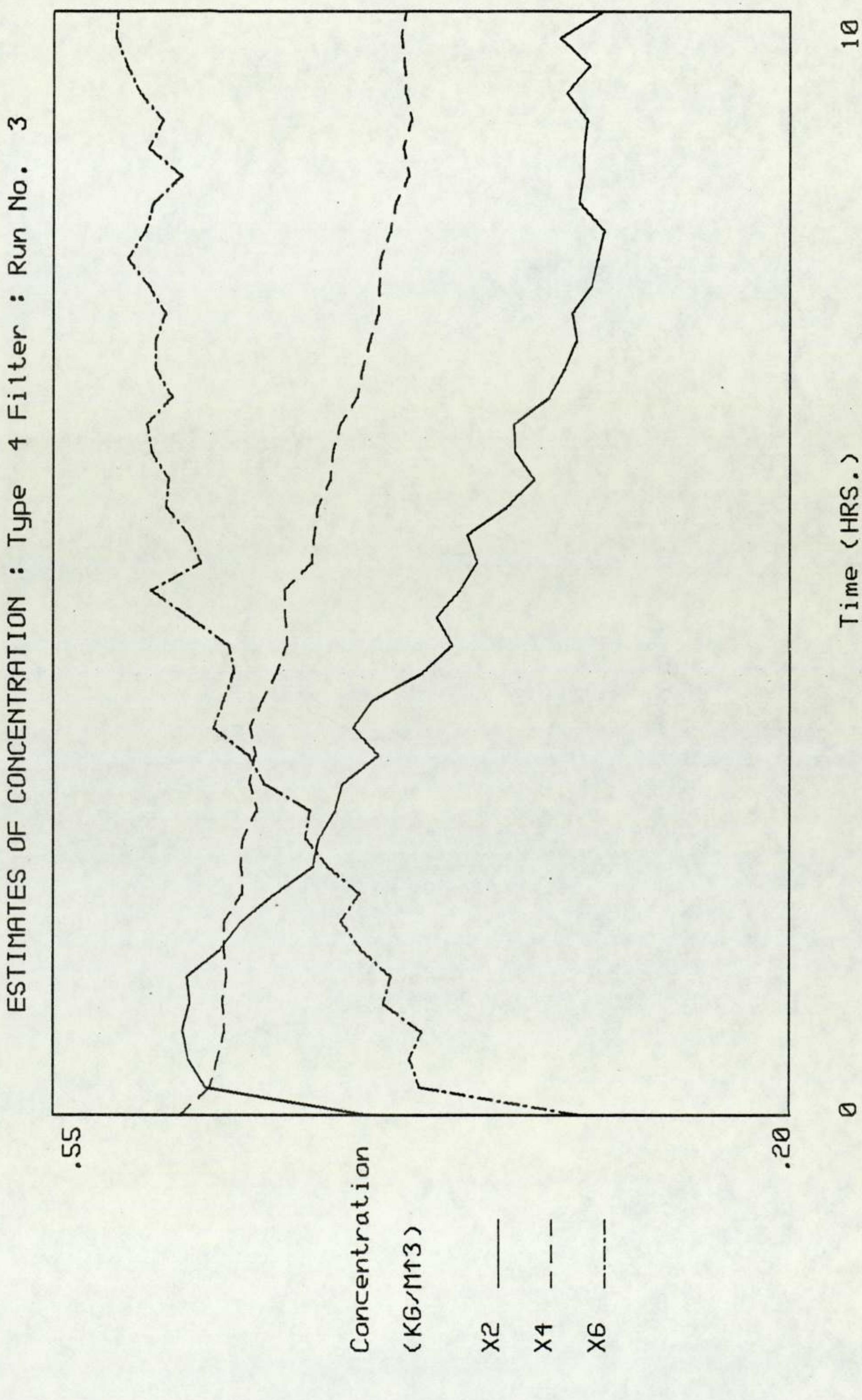

ESTIMATES OF CONCENTRATION : Type 4 Filter : Run No. 3
.55
.20
Concentration
(KG/M↑3)
X2
X4
X6
0
10
Time (HRS.)

FIGURE A-13

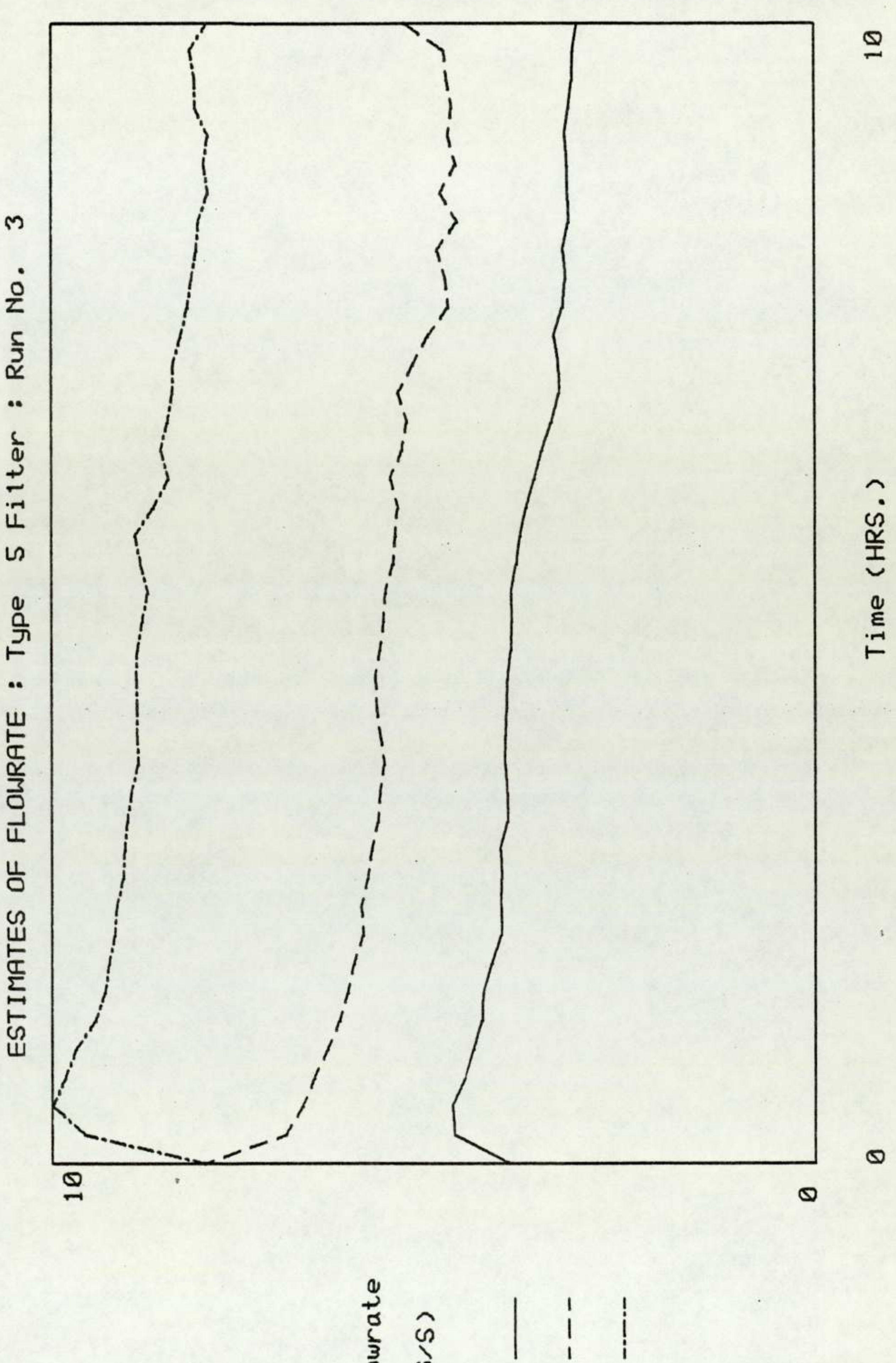
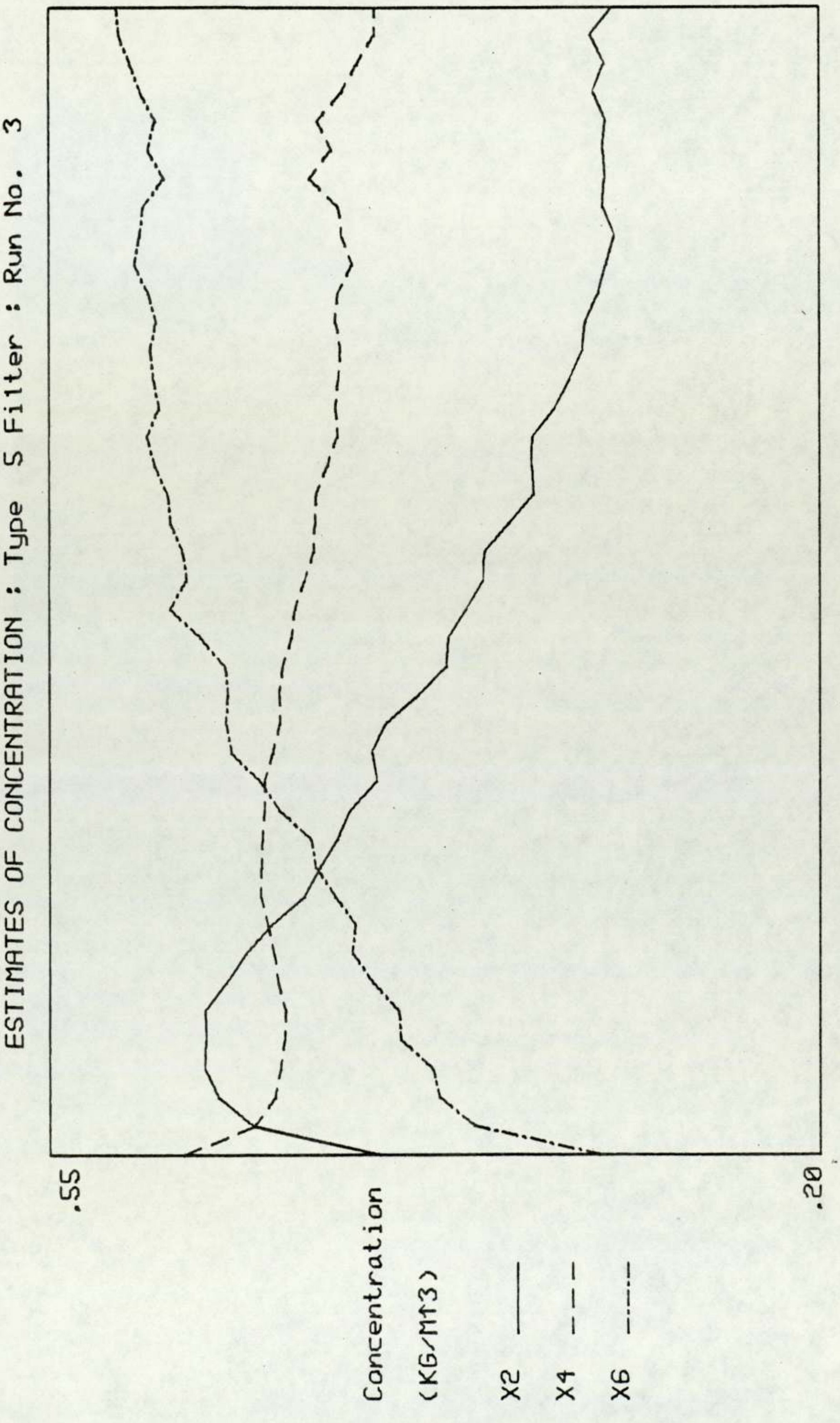
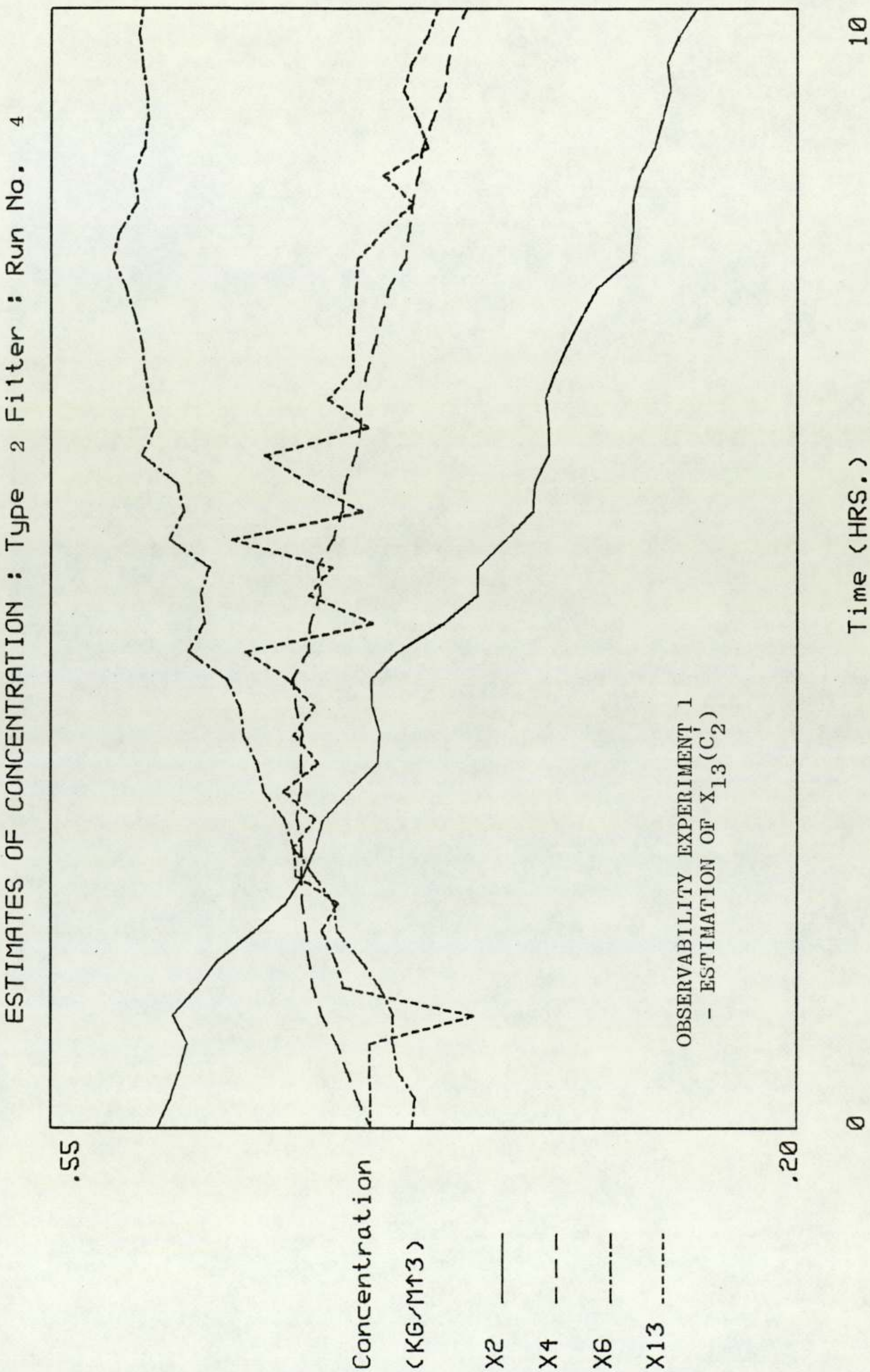

ESTIMATES OF FLOWRATE : Type 5 Filter : Run No. 3
Flowrate
(KG/S)
10
0
0
10
Time (HRS.)
X1
X3
X5

FIGURE A-14



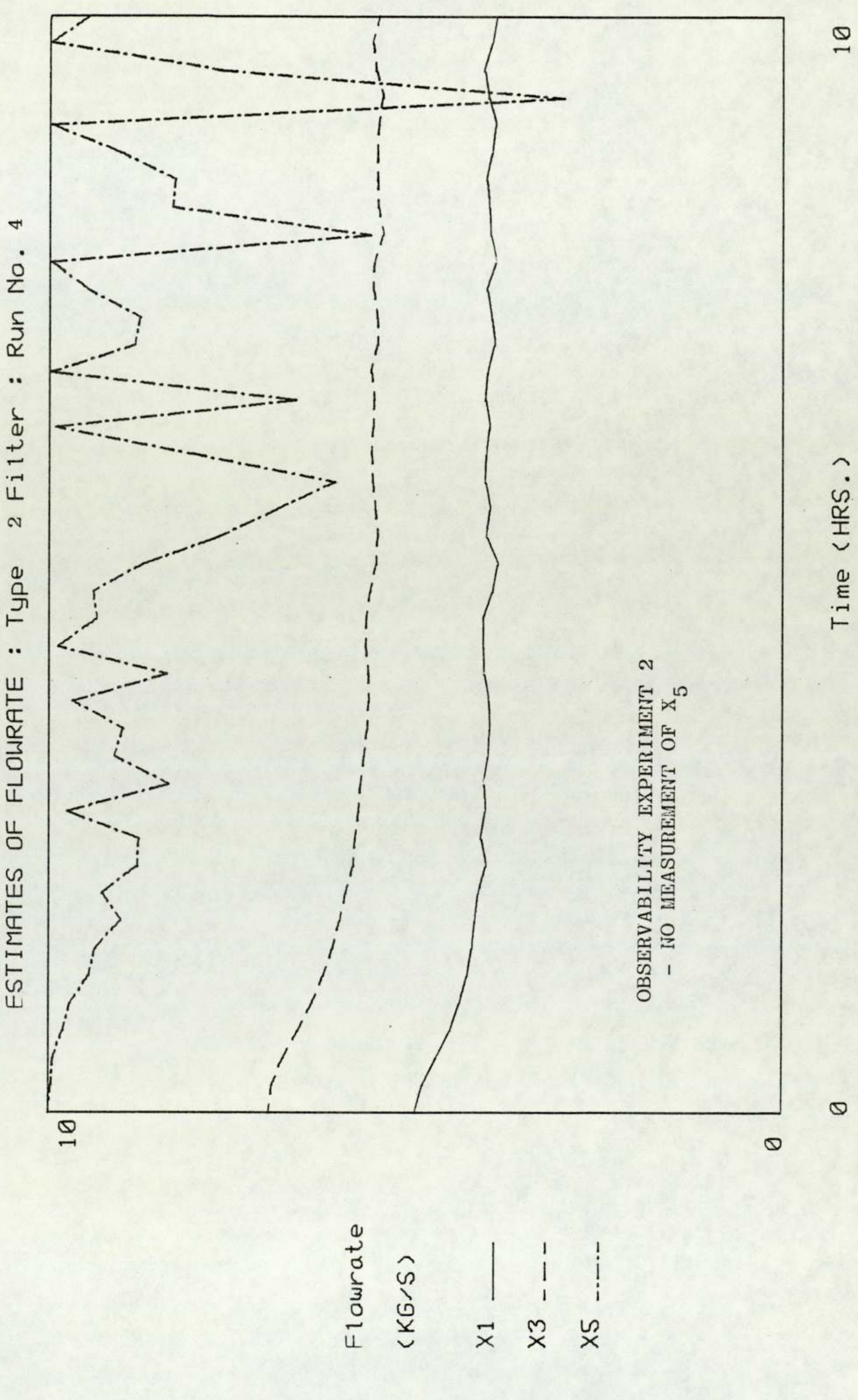
ESTIMATES OF CONCENTRATION : Type S Filter : Run No. 3
.55
.20
Concentration
(KG/M↑3)
0
10
Time (HRS.)
X2
X4
X6

FIGURE A-15



ESTIMATES OF CONCENTRATION : Type 2 Filter : Run No. 4
.55
.20
Concentration
(KG/M↑3)
X2
X4
X6
X13
0
10
Time (HRS.)
OBSERVABILITY EXPERIMENT 1
- ESTIMATION OF $X_{13}(C_2')$

FIGURE A-16


ESTIMATES OF FLOWRATE : Type 2 Filter : Run No. 4
OBSERVABILITY EXPERIMENT 2
- NO MEASUREMENT OF X5
Flowrate
(KG/S)
X1
X3
X5
10
0
0
10
Time (HRS.)

APPENDIX B

ASSEMBLER LEVEL PROGRAMMING OF THE H316 COMPUTER

CONTENTS

TABLE B-1.1 - SUMMARY OF DAP-16MOD2 INSTRUCTIONS

MNEMONIC	MEANING	MNEMONIC	MEANING
ADD	Binary add to A reg	OCT	Octal constant
ALR	Logical left rotate A	OTA	Output from A to peripheral
ALS	Arithmetic left shift A	OTK	Output keys
ANA	Logical AND to A		
AOA	Add one to A	SKP	Unconditional skip
ARR	Logical right rotate A	SKS	Skip if sense line set
ARS	Arithmetic right shift A	SMI	Skip if A negative
CAS	Compare and skip	SMK	Set mask
CRA	Clear A	SNZ	Skip if A not zero
		SPL	Skip if A positive
DIV	High speed divide	SSM	Set sigh minus
ENB	Enable CPU interrupts	STA	Store A
ERA	Exclusive OR to A	STX	Store index register
IAB	Interchange A and B	SUB	Subtract
IMA	Interchange Memory and A	SZE	Skip if zero
INA	Input to A from peripheral	TCA	Two's complement A
INH	Inhibit CPU Interrupts		

TABLE B-1.1 - continued

MNEMONIC	MEANING
INK	Input keys
IRS	Increment, replace and skip
JMP	Unconditional Jump
JST	Jump and store current location
LDA	Load A
LDX	Load index register
LLL	Long left shift of A and B
LRL	Long right shift of A and B
MPY	High Speed Multiply
NOP	No operation
OCP	Output control pulse

TABLE B-1.2 - SUMMARY OF DAP-16MOD2 PSEUDO OPERATIONS

MNEMONIC	MEANING
ABS	Instructions following to have absolute addresses
BCI	Binary coded information
BSZ	Block set to zero
DAC	Define address constant
ORG	The location of the next instruction
REL	Instructions following to have relative addresses
SETB	Set base
VFD	Variable field descriptor
XAC	External address constant
**	Zero address code
*	Indirect operation (when in op. code)
*	Address of this operation (when in address field)
'222	Octal constant
=1	Literal constant

B-2 PROGRAMMING INSTRUCTIONS FOR HADIOS

The following is a summary of the programming instructions used to access the HADIOS controller and HADIOS subinterfaces.

B-2.1 HADIOS CONTROLLER

SKS 'XXYY
- skip if controller not interrupting
- YY is the device address of the controller (70)
- XX is the controller highway address line (17)

B-2.2 DIGITAL INPUT

INA 'IIYY
- input to 'A' register
- II is the digital input highway address line (10)

B-2.3 DIGITAL OUTPUT

OTA 'OOYY
- Output from 'A' register
- OO is the digital output highway address line (13)

B-2.4 COUNTER INPUT

INA 'CCYY	- input to 'A' register from counter input 8 bit register* *Counter Register is automatically cleared after an INA.
	- CC is the first counter input highway address line (O2)
OTA 'DDYY	- Output from 'A' register to preset the counter input 8 bit register
	- DD is the second counter input highway address line (O3)
SKS 'CCYY	- skip if counter not interrupting
OCP 'CCYY	- enable half-full counter interrupt
OCP 'DDYY	- reset interrupt mode

B-2.5 ANALOGUE INPUT

OCP 'AAYY	- start A.D.C. conversion cycle
	- AA is the first analogue input highway address line (OO)
INA 'AAYY	- input data to the 'A' register if conversion is complete - data enters the 10 most significant bits of the 'A' register

OTA 'BBYY - output set up word to A.D.C.
- BB is the second analogue input highway address line (01)

The set up word has the following structure;

(1) Bits 1,2,3,4,9 and 10 are ignored.

(2) Bit 5 - set external trigger mode.

(3) Bit 6 - set sequential address mode

(4) Bit 7 - set free run mode

(5) Bit 8 - set direct data transfer mode

(6) Bits 11&12 - Multiplexer number (0 to 3)

(7) Bits 13-16 - Channel number (0 to 15)

Thus, the usual set up word for the system used in this research is

0 000 010 000 MMN NNN

where, MM = 00 or 01

and, N NNN = 00 to 15_8

B-2.6 <u>CONTROLLER INTERRUPT BIT</u>

When using the HADIOS controller in an interrupt mode it is necessary to set the corresponding interrupt flip-flop. To do this an SMK '20 instruction is issued with bit 13 set in the 'A' register.

B-3 INPUT/OUTPUT MODIFICATION PROGRAM

The program shown in Table B-3.1 permits the following input/output facilities in BASIC programs;

(1) The setting of sense switch 3 causes data to be input from the high speed paper tape reader when an INPUT statement is executed.

(2) The setting of sense switch 4 causes results to be output at the paper tape punch when a PRINT statement is executed.

The I/O MOD program can only be loaded into sector 0.

TABLE B-3.1. - INPUT/OUTPUT MODIFICATION PROGRAM

```
                          * I/O MOD - RELOCATEABLE       PAGE        1

                          * I/O MOD - RELOCATEABLE
                          *
                                 REL
00000     101004          AA     SS3
00001     0 01 00004             JMP    *+3
00002     140040                 CRA
00003     0 04 00105             STA    '105
00004     0 02 00406             LDA    '406
00005    -0 01 00006             JMP*   *+1
00006     004143                 OCT    4143
00007     0 12 00105      BB     IRS    '105
00010     140040                 CRA
00011     0 04 00106             STA    '106
00012    -0 10 00014             JST*   *+2
00013    -0 01 00015             JMP*   *+2
00014     003065                 OCT    3065
00015     004575                 OCT    4575
00016     101002          CC     SS4
00017     100000                 SKP
00020     0 12 00106             IRS    '106
00021    -0 10 00023             JST*   *+2
00022    -0 01 00024             JMP*   *+2
00023     003047                 OCT    3047
00024     004212                 OCT    4212
00025     0 12 00105      DD     IRS    '105
00026     140040                 CRA
00027     0 04 00106             STA    '106
00030    -0 01 00031             JMP*   *+1
00031     005245                 OCT    5245
                                 ABS
                                 ORG    '4142
04142     0 01 00000             JMP    AA
                                 ORG    '4211
04211     0 01 00016             JMP    CC
                                 ORG    '4574
04574     0 01 00007             JMP    BB
                                 ORG    '5244
05244     0 01 00025             JMP    DD
                                 END

AA      000000    BB      000007    CC      000016    DD      000025

0000 WARNING OR ERROR FLAGS
DAP-16 MOD 2    REV. C    01-26-71
AC
```

B-4 BASE SECTOR RELOCATION PROGRAM

The program shown in Table B-4.1 enables the relocation of Base Sector from sector 0 to sector 37 when FORTRAN and/or DAP-16 subroutines are called from a BASIC program. The program should normally be loaded into either sector 36 or 37 and on no account should it cross a sector boundary. Two further restrictions on its use are that location o of the sector into which the program is loaded (e.g. '37000) should not be used by any other program and in situations where interrupts of the central processor may occur, use of this program is not advised. When using this program up to 20 subroutines, each with a maximum of 19 arguments, may be called from a BASIC program.

TABLE B-4.1. - BASE SECTOR RELOCATION PROGRAM

```
4012  JMP*   '716
 716  DAC    STRT
 442  DEC    21

STRT  LDA    AA
      STA    AA+1
      LDA    BB
      STA    BB+1
      LDA*   AA+1
      ADD    CC
      STA*   BB+1
 REP  IRS    AA+1
      IRS    BB+1
      LDA*   AA+1
      CAS    CC+1
      SKP
      JMP    *+3
      STA*   BB+1
      JMP    REP
      LDA    DD
      STA*   BB+1
      LDA    DD+1
      SMK    '1320
      STX    0
      JMP    ARG
RTRN  CRA
```

TABLE B-4.1 - continued

```
        SMK     '1320
        LDX     0
        JMP*    RTN
ERR     DAC     **
        CRA
        SMK     '1320
        LDX     0
        JMP*    ERTN
AA      OCT     230
        OCT     0
BB      DAC     ARG
        OCT     0
CC      OCT     213
        JMP*    '550
DD      JMP     RTRN
        OCT     37000
RTN     OCT     4013
ERTN    OCT     4064
SR      DAC     ERR
                (19 locations)
        DAC     ERR
ARG     BSZ     19
```

B-5 PUNCH INTERRUPT MODIFICATION

A hardware modification to the High Speed Reader/Punch interface, which allows the enabling and disenabling of interrupts caused by the punch, has meant that the following program modification is necessary when using the Input/Output Modification Program (See Appendix B-4)

```
5353    JMP     AA
  AA    OCP     2
        LDA*    *+2
        JMP*    *+2
        OCT     5357
        OCT     5354
```

APPENDIX C

THE HADIOS EXECUTIVE PACKAGE MK.2

CONTENTS

C-1 HADIOS EXECUTIVE PROGRAM REVISION O1

The HADIOS EXECUTIVE PROGRAM can be best understood by following its normal order of execution. The first of the two entry points to the executive is the BASIC Entry Point. The statement CALL(1,...) in the BASIC program will cause the execution of subroutine 1 : see Figure C-1.1. In subroutine 1 the various tasks necessary before scanning are carried out and then the Dispatcher is entered to await the first real time clock interrupt which occurs after 100ms.

When the first clock interrupt occurs the HADIOS EXECUTIVE PROGRAM is entered at the second entry point which is known as the Interrupt Entry Point. The first action here is to decode the interrupt, i.e. determine which device has interrupted. As shown in Figure C-1.2 this will cause the correct section of interrupt response code to be executed. When the real time clock interrupts the HADIOS subinterfaces are interrogated as required by the devices selected parameter (P_1). Figure C-1.3 shows the clock interrupt response program and as can be seen after all the requested devices have been

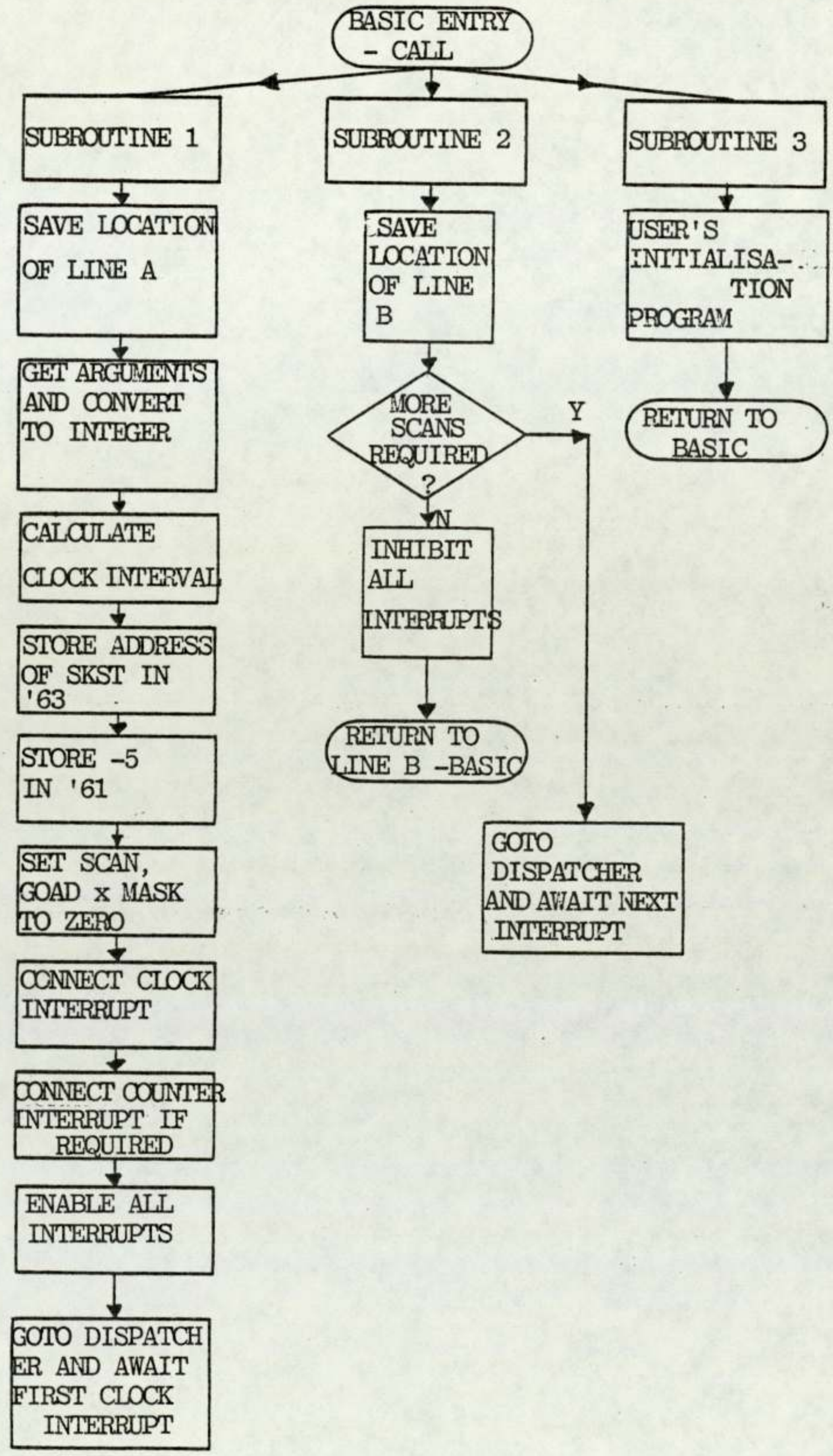


FIGURE C-1.1 - SUBROUTINES CALLED FROM BASIC

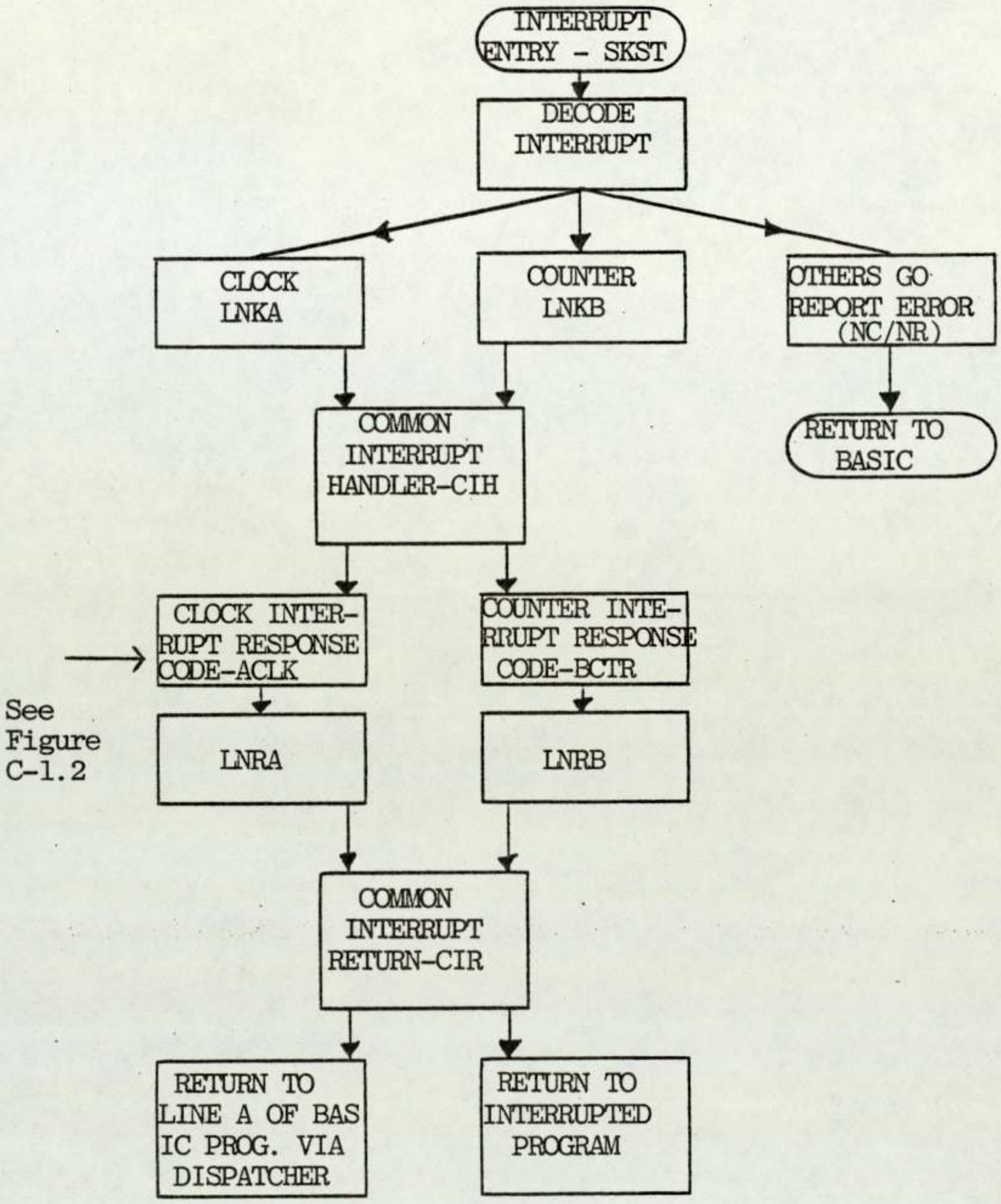


FIGURE C-1.2 - INTERRUPT HANDLING

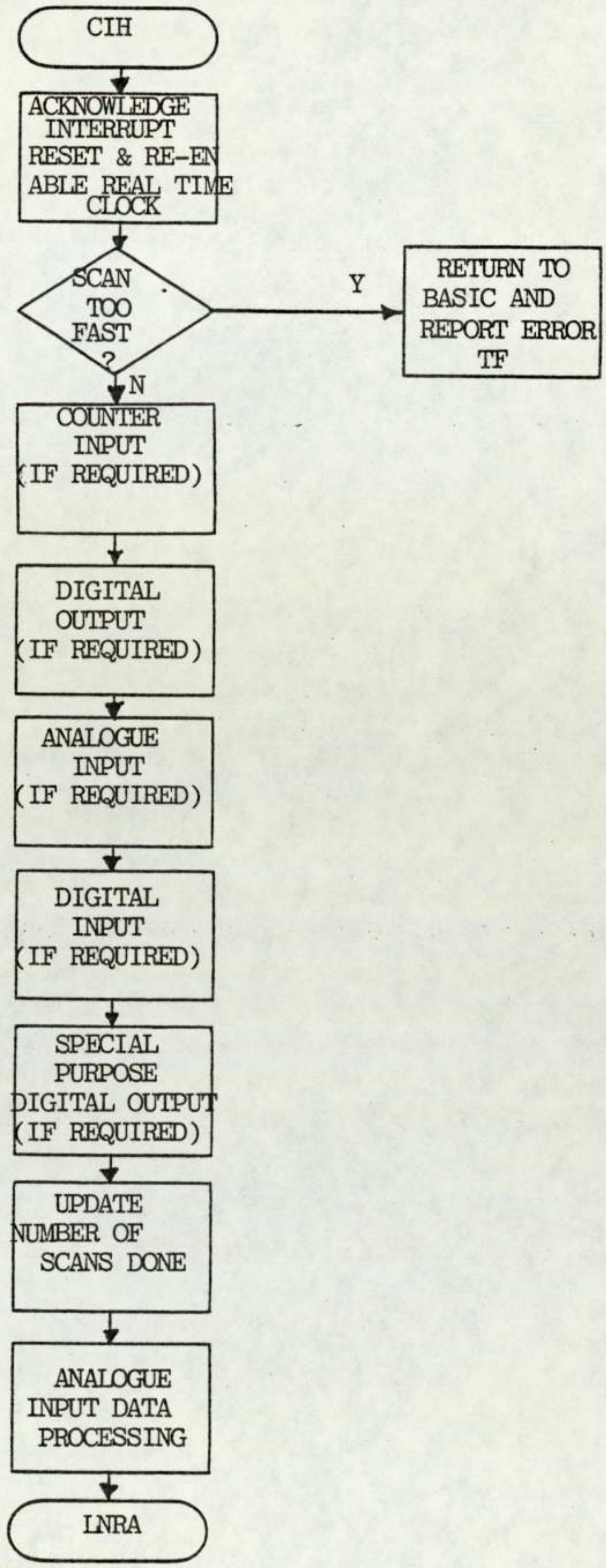


FIGURE C-1.3 - CLOCK INTERRUPT RESPONSE PROGRAM serviced the number of scans done counter is updated and execution continues at line A (see section 6.5.2) of the BASIC program.

The Inter Scan Basic Processing is now carried out following which the CALL(2) statement causes the program to move to subroutine 2. The number of scans done is now tested and if more are required the program moves to the dispatcher to await the next interrupt. On the other hand if the number of scans requested have been carried out then all interrupts are disconnected and execution continues at line B (see section 6.5.2) of the BASIC program.

When the counter half-full interrupt occurs the sequence of events is similar to that which occurs at a clock interrupt, see Figure C-1.2, but for this case after the interrupt has been serviced execution continues from the point in the program at which the interrupt occurred.

A listing of the HADIOS EXECUTIVE PROGRAM is given in Table C-1.1.

TABLE C-1.1 - THE HADIOS EXECUTIVE PROGRAM


```
                            * HADIOS EXECUTIVE. 16.06.76
                            *         REV 01
                            *
                                  REL
           000070           DATA EQU    '70
           000170           ANAG EQU    '170
           000270           CTR  EQU    '270
           000370           SET  EQU    '370
           001070           DIGI EQU    '1070
           001370           DIGO EQU    '1370
           001770           HAD  EQU    '1770
           000020           CLK  EQU    '20
           000220           SCLK EQU    '220
           000034           SIP  EQU    '34
           000037           SBP  EQU    '37
           000230           IBUF EQU    '230
           000515           CJST EQU    '515
           004064           SS   EQU    '4064
           000675           C12  EQU    '675
           000676           C21  EQU    '676
           000653           L22  EQU    '653
           000654           H22  EQU    '654
           000670           D22  EQU    '670
           000674           M22  EQU    '674
                            *
                            * INTERRUPT ENTRY POINT
                            *
     00000 0 000000         SKST DAC    **
     00001 34 0020               SKS    CLK
     00002 0 01 00013            JMP    LNKA
     00003 34 1770               SKS    HAD
     00004 0 01 00007            JMP    CONT
     00005 0 10 00707            JST    ERCL
     00006 147322                BCI    1,NR                  NOT RECOGNISED
                            *
                            * HADIOS INTERRUPTING
                            *
     00007 34 0270          CONT SKS    CTR
     00010 0 01 00026            JMP    LNKB
     00011 0 10 00707            JST    ERCL
     00012 147303                BCI    1,NC                  NOT COUNTER
                            *
                            * CLOCK INTERRUPT LINK
                            *
     00013 0 10 00041       LNKA JST    CIH
     00014 000000                BSZ    5
     00021 000001                OCT    1
     00022 0 000014              DAC    LNKA+1
                            *
                            * CLOCK INTERRUPT RETURN
                            *
     00023 0 000000         LNRA DAC    **
     00024 0 35 00022            LDX    *-2
     00025 0 01 00066            JMP    CIR
                            *
                            * COUNTER INTERRUPT LINK
                            *
```

TABLE C-1.1 continued


```
00026   0 10 00041 LNKB JST   CIH
00027   000000          BSZ   5
00034   000010          OCT   10
00035   0 000027        DAC   LNKB+1
                   *
                   * COUNTER INTERRUPT RETURN
                   *
00036   0 000000   LNRB DAC   **
00037   0 35 00035      LDX   *-2
00040   0 01 00066      JMP   CIR
                   *
                   * COMMON INTERRUPT HANDLER
                   *
00041   0 000000   CIH  DAC   **
00042  -0 15 00041      STX*  CIH
00043   0 35 00041      LDX   CIH
00044   1 13 00001      IMA   1,1
00045   000043          INK
00046   000005          SGL
00047   1 04 00002      STA   2,1
00050   0 02 00000      LDA   SKST
00051   1 04 00004      STA   4,1
00052   1 02 00005      LDA   5,1
00053   140401          CMA
00054   0 03 00130      ANA   MASK
00055   0 04 00130      STA   MASK
00056   74 0020         SMK   '20
00057   1 02 00007      LDA   7,1
00060   0 04 00065      STA   CIHA
00061   000201          IAB
00062   1 04 00003      STA   3,1
00063   000011          DXA
00064  -0 01 00065      JMP*  CIHA
00065   000000     CIHA BSZ   1
                   *
                   * COMMON INTERRUPT RETURN
                   *
00066   1 02 00003 CIR  LDA   3,1
00067   000201          IAB
00070   1 02 00005      LDA   5,1
00071   140401          CMA
00072   001001          INH
00073   0 03 00130      ANA   MASK
00074   1 05 00005      ERA   5,1
00075   0 04 00130      STA   MASK
00076   74 0020         SMK   '20
00077   1 02 00004      LDA   4,1
00100   0 04 00126      STA   CIRA
00101   1 02 00000      LDA   0,1
00102   0 04 00127      STA   CIRX
00103   1 02 00002      LDA   2,1
00104   171020          OTK
00105   1 13 00001      IMA   1,1
00106   0 35 00127      LDX   CIRX
00107   000401          ENB
00110  -0 01 00126      JMP*  CIRA
                   *
```

TABLE C-1.1 continued


```
                           * COMMON INTERRUPT INITIATOR
                           *
00111   0 000000           CII  DAC    **
00112  -0 35 00111              LDX*   CII
00113   0 12 00111              IRS    CII
00114   1 02 00000              LDA    0,1
00115   140401                  CMA
00116   0 03 00130              ANA    MASK
00117   1 05 00000              ERA    0,1
00120   74 0020                 SMK    '20
00121   0 04 00130              STA    MASK
00122  -0 02 00111              LDA*   CII
00123   0 12 00111              IRS    CII
00124   1 04 00002              STA    2,1
00125  -0 01 00111              JMP*   CII
00126   000000             CIRA BSZ    1
00127   000000             CIRX BSZ    1
00130   000000             MASK BSZ    1
00131   005243             ERR  OCT    5243
                           *
                           * DISPATCHER
                           *
00132   0 35 00775         DISP LDX    =-5
00133   1 02 00157              LDA    GOAD+5,1
00134   100040                  SZE
00135   0 01 00145              JMP    DISQ
00136   100010                  SR2
00137   0 01 00143              JMP    STOP
00140   0 12 00000              IRS    0
00141   0 01 00133              JMP    DISP+1
00142   0 01 00132              JMP    DISP
00143   0 10 00707         STOP JST    ERCL
00144   152711                  BCI    1,UI                  USER INTERFERED
00145   0 04 00151         DISQ STA    DISS
00146   140040                  CRA
00147   1 04 00157              STA    GOAD+5,1
00150  -0 01 00151              JMP*   DISS
00151   000000             DISS BSZ    1
00152   000000             GOAD BSZ    5
                           *
                           * BASIC ENTRY POINT
                           *
00157   0 02 00230         CALL LDA    IBUF
00160   0 07 00515              SUB    CJST
00161   0 04 00000              STA    0
00162  -1 01 00162              JMP*   TABL-1,1
00163   0 000175           TABL DAC    SUB1
00164   0 000307                DAC    SUB2
00165   0 000000                XAC    MYS1
00166   0 004064                DAC    SS
00167   0 004064                DAC    SS
00170   0 004064                DAC    SS
00171   0 004064                DAC    SS
00172   0 004064                DAC    SS
00173   0 004064                DAC    SS
00174   0 004064                DAC    SS
                           *
```

TABLE C-1.1 continued


```
                          * SUBROUTINE 1
                          *
00175    0 02 00034  SUB1 LDA     SIP
00176    0 04 00345       STA     SIP1
00177    0 02 00037       LDA     SBP
00200    0 04 00346       STA     SBP1
00201    0 02 00342       LDA     NEXT
00202    0 04 00204       STA     *+2
00203    0 01 00205       JMP     *+2
00204    0 000000         DAC     **
00205   -0 10 00344       JST*    FAT
00206    000013           DEC     11
00207    000000      PARS BSZ     11
00222    0 35 00773       LDX     =-10
00223    0 15 00351       STX     KT
00224   -0 10 00653       JST*    L22
00225   -1 000221         DAC*    PARS+10,1
00226   -0 10 00676       JST*    C21
00227    0 01 00305       JMP     ERI
00230    0 35 00351       LDX     KT
00231    1 04 00340       STA     CTRS+10,1
00232    0 12 00000       IRS     0
00233    0 01 00223       JMP     *-8
00234   -0 10 00653       JST*    L22
00235   -0 000210         DAC*    PARS+1
00236   -0 10 00674       JST*    M22
00237    0 000352         DAC     F50
00240   -0 10 00676       JST*    C21
00241    0 01 00305       JMP     ERI
00242    140407           TCA
00243    0 04 00343       STA     CINT
00244    0 02 00340       LDA     DACS
00245    0 04 00063       STA     '63
00246    0 02 00341       LDA     FRST
00247    0 04 00061       STA     '61
00250    0 35 00775       LDX     =-5
00251    140040           CRA
00252    0 04 00130       STA     MASK
00253    0 04 00700       STA     SCAN
00254    1 04 00157       STA     GOAD+5,1
00255    0 12 00000       IRS     0
00256    0 01 00254       JMP     *-2
00257    0 10 00111       JST     CII
00260    0 000021         DAC     LNRA-2
00261    0 000354         DAC     ACLK
00262    14 0020          OCP     CLK
00263    0 02 00326       LDA     CTRS
00264    0 03 00772       ANA     ='10
00265    101040           SNZ
00266    0 01 00303       JMP     INT
00267    0 02 00335       LDA     CTRS+7
00270    74 0370          OTA     SET
00271    0 01 00270       JMP     *-1
00272    0 02 00334       LDA     CTRS+6
00273    0 11 00771       CAS     =1
00274    0 01 00277       JMP     *+3
00275    101000           NOP
```

TABLE C-1.1 continued


```
00276   0 01 00303        JMP   INT
00277   0 10 00111        JST   CII
00300   0 000034          DAC   LNRB-2
00301   0 000716          DAC   BCTR
00302   14 0270           OCP   CTR
00303   000401      INT   ENB
00304   0 01 00132        JMP   DISP
00305   0 10 00707  ERI   JST   ERCL
00306   151311            BCI   1,RI                    REAL TO INTEGER
                    *
                    * SUBROUTINE 2
                    *
00307   0 02 00034  SUB2  LDA   SIP
00310   0 04 00347        STA   SIP2
00311   0 02 00037        LDA   SBP
00312   0 04 00350        STA   SBP2
00313   0 02 00700        LDA   SCAN
00314   0 07 00330        SUB   CTRS+2
00315   100040            SZE
00316   0 01 00132        JMP   DISP
00317   14 0220           OCP   SCLK
00320   14 0370           OCP   SET
00321   0 02 00347        LDA   SIP2
00322   0 04 00034        STA   SIP
00323   0 02 00350        LDA   SBP2
00324   0 04 00037        STA   SBP
00325  -0 01 00677        JMP*  CL06
00326   000000      CTRS  BSZ   10
00340   0 000000    DACS  DAC   SKST
00341   177777      FRST  DEC   -1
00342   0 000231    NEXT  DAC   IBUF+1
00343   000000      CINT  BSZ   1
00344   0 000000    FAT   XAC   F$AT
00345   000000      SIP1  BSZ   1
00346   000000      SBP1  BSZ   1
00347   000000      SIP2  BSZ   1
00350   000000      SBP2  BSZ   1
00351   000000      KT    BSZ   1
00352   041544      F50   DEC   50.
00353   000000
                    *
                    * CLOCK INTERRUPT RESPONSE CODE
                    *
00354   14 0220     ACLK  OCP   SCLK
00355   0 02 00343        LDA   CINT
00356   0 04 00061        STA   '61
00357   14 0020           OCP   CLK
00360   000401            ENB
00361   0 02 00020        LDA   LNKA+5
00362   0 07 00604        SUB   BSTP
00363   100400            SPL
00364   0 01 00705        JMP   STP
00365   0 02 00326        LDA   CTRS
00366   0 03 00772        ANA   ='10
00367   101040            SNZ
00370   0 01 00422        JMP   A1
                    *
```

TABLE C-1.1 continued


```
                          * COUNTER INPUT
                          *
00371   0 02 00221              LDA     PARS+10
00372   0 06 00774              ADD     =66
00373   0 04 00404              STA     *+9
00374   140040                  CRA
00375   001001                  INH
00376   0 04 00755              STA     FLAG
00377   54 0270                 INA     CTR
00400   0 01 00377              JMP     *-1
00401   0 04 00701              STA     TEMP
00402  -0 10 00675              JST*    C12
00403  -0 10 00654              JST*    H22
00404   0 000000                DAC     **
00405   0 02 00334              LDA     CTRS+6
00406   0 11 00771              CAS     =1
00407   0 01 00411              JMP     *+2
00410   0 01 00413              JMP     *+3
00411   0 02 00335              LDA     CTRS+7
00412   0 01 00414              JMP     *+2
00413   0 02 00701              LDA     TEMP
00414   74 0370                 OTA     SET
00415   0 01 00414              JMP     *-1
00416   0 02 00061              LDA     '61
00417   0 04 00702              STA     TEMP+1
00420   101000                  NOP
00421   000401                  ENB
                          *
                          * DIGITAL OUTPUT
                          *
00422   0 02 00326 A1           LDA     CTRS
00423   0 03 00770              ANA     ='1000
00424   101040                  SNZ
00425   0 01 00473              JMP     A1A
00426   0 02 00337              LDA     CTRS+9
00427   0 11 00771              CAS     =1
00430   0 01 00470              JMP     PRB
00431   0 01 00443              JMP     ARRY
00432   0 02 00221              LDA     PARS+10
00433   0 06 00767              ADD     =70
00434   0 04 00436              STA     *+2
00435  -0 10 00653              JST*    L22
00436   0 000000                DAC     **
00437  -0 10 00676              JST*    C21
00440   0 01 00305              JMP     ERI
00441   0 01 00471              JMP     OTP
00442   000000         WORD     BSZ     1
00443   0 02 00221     ARRY     LDA     PARS+10
00444   0 06 00766              ADD     =106
00445   0 04 00605              STA     LOCB
00446   140040                  CRA
00447   0 04 00442              STA     WORD
00450   0 35 00765              LDX     =-16
00451   0 02 00442     REP2     LDA     WORD
00452   0415 77                 ALS     1
00453   0 04 00442              STA     WORD
00454  -0 02 00605              LDA*    LOCB
```

TABLE C-1.1 continued


```
00455   101040              SNZ
00456   0 01 00462          JMP    *+4
00457   0 02 00442          LDA    WORD
00460   141206              AOA
00461   0 04 00442          STA    WORD
00462   0 12 00605          IRS    LOCB
00463   0 12 00605          IRS    LOCB
00464   0 12 00000          IRS    0
00465   0 01 00451          JMP    REP2
00466   0 02 00442          LDA    WORD
00467   0 01 00471          JMP    OTP
00470   0 02 00442  PRB     LDA    WORD
00471   74 1370     OTP     OTA    DIGO
00472   0 01 00471          JMP    *-1
                    *
                    * ANALOGUE SCAN
                    *
00473   0 02 00326  A1A     LDA    CTRS
00474   0 03 00771          ANA    =1
00475   101040              SNZ
00476   0 01 00607          JMP    A2
00477   0 35 00764          LDX    =-32
00500   140040              CRA
00501   -0 04 00576         STA*   AB32
00502   0 12 00000          IRS    0
00503   0 01 00501          JMP    *-2
00504   0 02 00331          LDA    CTRS+3
00505   140407              TCA
00506   0 04 00600          STA    ENS
00507   0 02 00333          LDA    CTRS+5
00510   140407              TCA
00511   0 04 00601          STA    LAST
00512   0 02 00332  MUX1    LDA    CTRS+4
00513   0 07 00763          SUB    =16
00514   101400              SMI
00515   0 01 00545          JMP    MUX2
00516   0 02 00602          LDA    M1
00517   0 05 00332          ERA    CTRS+4
00520   74 0170             OTA    ANAG
00521   000000              HLT
00522   0 07 00602          SUB    M1
00523   0 07 00763          SUB    =16
00524   0 04 00000          STA    0
00525   14 0070     MU1     OCP    DATA
00526   140040              CRA
00527   54 0070             INA    DATA
00530   0 01 00527          JMP    *-1
00531   0404 72             LGR    6
00532   -0 06 00577         ADD*   AB16
00533   -0 04 00577         STA*   AB16
00534   0 02 00601          LDA    LAST
00535   0 06 00763          ADD    =16
00536   0 06 00000          ADD    0
00537   101040              SNZ
00540   0 01 00573          JMP    ESBL
00541   0 12 00000          IRS    0
00542   0 01 00525          JMP    MU1
```

TABLE C-1.1 continued


```
00543  140040              CRA
00544  0 01 00547          JMP   *+3
00545  0 02 00332 MUX2 LDA   CTRS+4
00546  0 07 00763          SUB   =16
00547  0 05 00603          ERA   M2
00550  74 0170             OTA   ANAG
00551  000000              HLT
00552  0 07 00603          SUB   M2
00553  0 07 00763          SUB   =16
00554  0 04 00000          STA   0
00555  14 0070     MU2     OCP   DATA
00556  140040              CRA
00557  54 0070             INA   DATA
00560  0 01 00557          JMP   *-1
00561  0404 72             LGR   6
00562  -0 06 00576         ADD*  AB32
00563  -0 04 00576         STA*  AB32
00564  0 02 00601          LDA   LAST
00565  0 06 00762          ADD   =32
00566  0 06 00000          ADD   0
00567  101040              SNZ
00570  0 01 00573          JMP   ESBL
00571  0 12 00000          IRS   0
00572  0 01 00555          JMP   MU2
00573  0 12 00600 ESBL IRS   ENS
00574  0 01 00512          JMP   MUX1
00575  0 01 00607          JMP   A2
00576  1 001072    AB32 DAC   ABUF+32,1
00577  1 001052    AB16 DAC   ABUF+16,1
00600  000000      ENS  BSZ   1
00601  000000      LAST BSZ   1
00602  002000      M1   OCT   2000
00603  002020      M2   OCT   2020
00604  0 000000    BSTP XAC   C$12
00605  000000      LOCB BSZ   2
                   *
                   * DIGITAL INPUT
                   *
00607  0 02 00326 A2   LDA   CTRS
00610  0 03 00761      ANA   ='100
00611  101040          SNZ
00612  0 01 00655      JMP   A3
00613  54 1070         INA   DIGI
00614  0 01 00613      JMP   *-1
00615  0 04 00606      STA   LOCB+1
00616  0 02 00336      LDA   CTRS+8
00617  100040          SZE
00620  0 01 00631      JMP   IP1
00621  0 02 00221      LDA   PARS+10
00622  0 06 00760      ADD   =68
00623  0 04 00627      STA   *+4
00624  0 02 00606      LDA   LOCB+1
00625  -0 10 00675     JST*  C12
00626  -0 10 00654     JST*  H22
00627  0 000000        DAC   **
00630  0 01 00655      JMP   A3
00631  0 35 00765 IP1  LDX   =-16
```

TABLE C-1.1 continued


```
00632  0 02 00221       LDA   PARS+10
00633  0 06 00757       ADD   =74
00634  0 04 00650       STA   LOCA
00635  140040           CRA
00636  000201           IAB
00637  0 02 00606 REP1  LDA   LOCB+1
00640  0416 77          ALR   1
00641  0 04 00606       STA   LOCB+1
00642  101001           SSC
00643  0 01 00646       JMP   *+3
00644  0 02 00756       LDA   ='40300
00645  0 01 00647       JMP   *+2
00646  140040           CRA
00647  -0 10 00654      JST*  H22
00650  0 000000   LOCA  DAC   **
00651  0 12 00650       IRS   LOCA
00652  0 12 00650       IRS   LOCA
00653  0 12 00000       IRS   0
00654  0 01 00637       JMP   REP1
                  *
                  * SPECIAL PURPOSE DIGITAL OUTPUT
                  *
00655  0 02 00337 A3    LDA   CTRS+9
00656  0 11 00771       CAS   =1
00657  -0 10 00703      JST*  SPSR
00660  101000           NOP
                  *
                  * ANALOGUE DATA PROCESSING
                  *      (IF REQUIRED)
                  *
00661  0 02 00326       LDA   CTRS
00662  0 03 00771       ANA   =1
00663  101040           SNZ
00664  0 01 00666       JMP   A4
00665  -0 10 00704      JST*  ANPR
                  *
                  * UPDATE SCANS DONE
                  *
00666  0 12 00700 A4    IRS   SCAN
00667  0 02 00345       LDA   SIP1
00670  0 04 00034       STA   SIP
00671  0 02 00346       LDA   SBP1
00672  0 04 00037       STA   SBP
00673  0 02 00677       LDA   CL06
00674  0 04 00152       STA   GOAD
00675  0 10 00023       JST   LNRA
00676  0 01 00354       JMP   ACLK
00677  004013     CL06  OCT   4013
00700  000000     SCAN  BSZ   1
00701  000000     TEMP  BSZ   2
00703  0 000000   SPSR  XAC   MYSR
00704  0 001000   ANPR  DAC   ANP1
00705  0 10 00707 STP   JST   ERCL
00706  152306           BCI   1,TF                SCAN TOO FAST
                  *
                  * ERROR RETURN TO BASIC
                  *
```

TABLE C-1.1 continued


```
00707   0 000000    ERCL DAC    **
00710  -0 02 00707       LDA*   ERCL
00711   0 04 00715       STA    *+4
00712   14 0220          OCP    SCLK
00713   14 0370          OCP    SET
00714  -0 10 00131       JST*   ERR
00715   0 000000         DAC    **
                    *
                    * COUNTER INTERRUPT
                    *   RESPONSE CODE
                    *
00716   14 0370     BCTR OCP    SET
00717   0 02 00335       LDA    CTRS+7
00720   74 0370          OTA    SET
00721   0 01 00720       JMP    *-1
00722   14 0270          OCP    CTR
00723   000401           ENB
00724   0 02 00221       LDA    PARS+10
00725   0 06 00761       ADD    =64
00726   0 04 00742       STA    *+12
00727   0 06 00772       ADD    =8
00730   0 04 00747       STA    *+15
00731   0 02 00755       LDA    FLAG
00732   100040           SZE
00733   0 01 00743       JMP    *+8
00734   0 02 00061       LDA    '61
00735   0 07 00702       SUB    TEMP+1
00736  -0 10 00752       JST*   C12A
00737  -0 10 00753       JST*   D22A
00740   0 000352         DAC    F50
00741  -0 10 00754       JST*   H22A
00742   0 000000         DAC    **
00743   0 12 00755       IRS    FLAG
00744   0 02 00755       LDA    FLAG
00745  -0 10 00752       JST*   C12A
00746  -0 10 00754       JST*   H22A
00747   0 000000         DAC    **
00750   0 10 00036       JST    LNRB
00751   0 01 00716       JMP    BCTR
00752   0 000000    C12A XAC    C$12
00753   0 000000    D22A XAC    D$22
00754   0 000000    H22A XAC    H$22
00755   000000      FLAG BSZ    1
00756   040300           FIN
00757   000112
00760   000104
00761   000100
00762   000040
00763   000020
00764   177740
00765   177760
00766   000152
00767   000106
00770   001000
00771   000001
00772   000010
00773   177766
```

TABLE C-1.1 continued


```
     00774    000102
     00775    177773

                                 *
                                 * ANALOGUE DATA PROCESSING
                                 *
                                      ORG    '1000
     01000     0 000000    ANP1  DAC    **
     01001    -0 02 01076          LDA*   CT4
     01002     0 06 01072          ADD    BF
     01003     0 04 01073          STA    ADBF
     01004    -0 02 01100          LDA*   PR10
     01005    -0 06 01076          ADD*   CT4
     01006    -0 06 01076          ADD*   CT4
     01007     0 04 01074          STA    DBF
     01010    -0 02 01077          LDA*   CT5
     01011     0 06 01101          ADD    =1
     01012     0 06 01072          ADD    BF
     01013     0 04 01075          STA    LBF
     01014    -0 02 01073    REP   LDA*   ADBF
     01015    -0 10 00675          JST*   C12
     01016    -0 10 00670          JST*   D22
     01017    -0 000212            DAC*   PARS+3
     01020    -0 10 00654          JST*   H22
     01021    -0 001074            DAC*   DBF
     01022     0 12 01073          IRS    ADBF
     01023     0 12 01074          IRS    DBF
     01024     0 12 01074          IRS    DBF
     01025     0 02 01073          LDA    ADBF
     01026     0 07 01075          SUB    LBF
     01027     100400              SPL
     01030     0 01 01014          JMP    REP
     01031    -0 01 01000          JMP*   ANP1
     01032     000000      ABUF  BSZ    32
     01072     0 001032    BF    DAC    ABUF
     01073     000000      ADBF  BSZ    1
     01074     000000      DBF   BSZ    1
     01075     000000      LBF   BSZ    1
     01076     0 000332    CT4   DAC    CTRS+4
     01077     0 000333    CT5   DAC    CTRS+5
     01100     0 000221    PR10  DAC    PARS+10
     01101     000001              FIN

                                 *
                                 * PATCH IN BASIC
                                 *   CALLING SEQUENCE
                                 *
                                      ABS
                                      ORG    '4012
     04012    -0 01 00716          JMP*   '716
                                      ORG    '716
     00716     0 000157            DAC    CALL
                                      ORG    '551
     00551     0 000707            DAC    ERCL
                                      END

A1       000422     A1A     000473     A2       000607     A3       000655
```

TABLE C-1.1 continued

A4	000666	AB16	000577	AB32	000576	ABUF	001032
ACLK	000354	ADBF	001073	ANAG	000170A	ANP1	001000
ANPR	000704	ARRY	000443	BCTR	000716	BF	001072
BSTP	000604	C12	000675A	C12A	000752	C21	000676A
CALL	000157	CIH	000041	CIHA	000065	CII	000111
CINT	000343	CIR	000066	CIRA	000126	CIRX	000127
CJST	000515A	CL06	000677	CLK	000020A	CONT	000007
CT4	001076	CT5	001077	CTR	000270A	CTRS	000326
D22	000670A	D22A	000753	DACS	000340	DATA	000070A
DBF	001074	DIGI	001070A	DIGO	001370A	DISP	000132
DISQ	000145	DISS	000151	ENS	000600	ERCL	000707
ERI	000305	ERR	000131	ESBL	000573	F50	000352
FAT	000344	FLAG	000755	FRST	000341	GOAD	000152
H22	000654A	H22A	000754	HAD	001770A	IBUF	000230A
INT	000303	IP1	000631	KT	000351	L22	000653A
LAST	000601	LBF	001075	LNKA	000013	LNKB	000026
LNRA	000023	LNRB	000036	LOCA	000650	LOCB	000605
M1	000602	M2	000603	M22	000674A	MASK	000130
MU1	000525	MU2	000555	MUX1	000512	MUX2	000545
NEXT	000342	OTP	000471	PARS	000207	PR10	001100
PRB	000470	REP	001014	REP1	000637	REP2	000451
SBP	000037A	SBP1	000346	SBP2	000350	SCAN	000700
SCLK	000220A	SET	000370A	SIP	000034A	SIP1	000345
SIP2	000347	SKST	000000	SPSR	000703	SS	004064A
STOP	000143	STP	000705	SUB1	000175	SUB2	000307
TABL	000163	TEMP	000701	WORD	000442		

0000 WARNING OR ERROR FLAGS
DAP-16 MOD 2 REV. C 01-26-71

C

C-2 RELOCATION OF ADT1-8 TABLE

During normal operation of the BASIC-16 Interpreter there are no interrupts of the central processor and so there is no need to reserve locations '61 and '63 for real time clock and interrupt handling. However, this is not the case when using the HADIOS EXECUTIVE PACKAGE and so the ADT1-8 table (assign dimensioned variable table) must be relocated from locations '61 to '70, to locations '757 to '766. This is done by the program shown in Table C-2.1.

TABLE C-2.1 - ADT1-8 RELOCATION


```
                             * ADT1-8 RELOCATION
                             *
                                     ORG    '4757
     04757   0 04 00761              STA    '761
                                     ORG    '4761
     04761   0 04 00762              STA    '762
                                     ORG    '4763
     04763   0 04 00764              STA    '764
     04764   0 15 00763              STX    '763
                                     ORG    '4766
     04766   0 04 00766              STA    '766
                                     ORG    '5006
     05006   000764                  OCT    764
     05007   0 04 00764              STA    '764
     05010   0 12 00762              IRS    '762
                                     ORG    '5023
     05023   0 02 00764              LDA    '764
                                     ORG    '5025
     05025   0 06 00762              ADD    '762
                                     ORG    '5027
     05027   0 04 00765              STA    '765
                                     ORG    '5034
     05034   0 04 00757              STA    '757
     05035   0 07 00765              SUB    '765
                                     ORG    '5037
     05037   0 04 00760              STA    '760
     05040  -0 02 00757              LDA*   '757
     05041  -0 04 00760              STA*   '760
     05042   0 12 00757              IRS    '757
     05043   0 12 00760              IRS    '760
                                     ORG    '5045
     05045   0 07 00757              SUB    '757
                                     ORG    '5050
     05050   0 02 00760              LDA    '760
                                     ORG    '5061
     05061   0 07 00765              SUB    '765
                                     ORG    '5065
     05065   0 02 00764              LDA    '764
                                     ORG    '5070
     05070   0 04 00757              STA    '757
                                     ORG    '5074
     05074   0 12 00757              IRS    '757
                                     ORG    '5076
     05076   0 02 00762              LDA    '762
                                     ORG    '5100
     05100   0 04 00757              STA    '757
                                     ORG    '5104
     05104   0 12 00757              IRS    '757
                                     ORG    '5112
     05112   0 02 00762              LDA    '762
                                     ORG    '5114
     05114   0 02 00761              LDA    '761
                                     ORG    '5123
     05123   0 02 00766              LDA    '766
                                     ORG    '5126
     05126   0 35 00763              LDX    '763
                                     END
```

C-3 SUBROUTINES FØER, FØHT

The special version of subroutine FØER shown in Table C-3.1 has three functions,

(i) real time clock and counter input interrupts are inhibited,

(ii) the mnemonic 'FT' is printed out on the VDU or teletype, and

(iii) the program returns to the BASIC-16 error reporting routine to display the error mnemonic in the normal BASIC manner.

Subroutine FØHT differs only in the third of the above functions which for this routine is that the central processor halts.

TABLE C-3.1. - SUBROUTINES FER, FHT


```
                         *
                         *    F$ER FOR USE WITH HADIOS EXECUTIVE
                         *
                              SUBR   F$HT,HALT
                              SUBR   F$ER,ERR
                              REL
00000   0 000000     ERR      DAC    **
00001   14 0220               OCP    '220
00002   14 0370               OCP    '370
00003  -0 02 00000            LDA*   ERR
00004   0 04 00053            STA    TEMP
00005  -0 02 00053            LDA*   TEMP
00006   0 04 00013            STA    ERR1+1
00007   0 02 00055            LDA    FT
00010   0 04 00056            STA    OP
00011   0 10 00031            JST    CONT
00012  -0 10 00054   ERR1     JST*   TEMP+1
00013   000000                OCT    0
00014   0 000000     HALT     DAC    **
00015   14 0220               OCP    '220
00016   14 0370               OCP    '370
00017   000201                IAB
00020  -0 02 00014            LDA*   HALT
00021   0 04 00053            STA    TEMP
00022  -0 02 00053            LDA*   TEMP
00023   0 04 00056            STA    OP
00024   0 10 00031            JST    CONT
00025   000201                IAB
00026   000000                HLT
00027   0 12 00014            IRS    HALT
00030  -0 01 00014            JMP*   HALT
00031   0 000000     CONT     DAC    **
00032   34 0104               SKS    '104
00033   0 01 00032            JMP    *-1
00034   14 0104               OCP    '104
00035   0 02 00052            LDA    CRLF
00036   0 10 00042            JST    OUT
00037   0 02 00056            LDA    OP
00040   0 10 00042            JST    OUT
00041  -0 01 00031            JMP*   CONT
00042   0 000000     OUT      DAC    **
00043   0416 70               ALR    8
00044   74 0004               OTA    4
00045   0 01 00044            JMP    *-1
00046   0416 70               ALR    8
00047   74 0004               OTA    4
00050   0 01 00047            JMP    *-1
00051  -0 01 00042            JMP*   OUT
00052   106612       CRLF     OCT    106612
00053   000000       TEMP     OCT    0
00054   005243                OCT    5243
00055   143324       FT       OCT    143324
00056   000000       OP       OCT    0
                              FIN
                              END

CONT   000031    CRLF   000052    ERR   000000    ERR1   000012
```

C-4 SUBROUTINES MYS1, MYSR

The subroutines shown in Table C-4.1 are dummy routines loaded with the HADIOS EXECUTIVE PROGRAM in case either of the special purpose digital output subroutines are mistakenly called when they have not been loaded.

TABLE C-4.1. - SUBROUTINES MYS1, MYSR


```
                                * SUBR MYS1 & MYSR - DUMMY
                                *
                                *  THESE ARE DUMMY ROUTINES
                                *  LOADED TO GENERATE ERROR
                                *  CONDITIONS IF THE HADIOS
                                *  EXEC. CALLS THE SPECIAL
                                *  PURPOSE DIGITAL OUTPUT
                                *  ROUTINES WHEN THEY HAVE
                                *  NOT BEEN LOADED.
                                *
                                *  MYS1 - CALLED FROM BASIC.
                                *  MYSR - CALLED BY THE
                                *              HADIOS EXEC.
                                *
                                      SUBR  MYS1,A1
                                      SUBR  MYSR,A2
                                      REL
00000   -0 10 00005 A1    JST*   ERCL
00001     151661          BCI    1,S1
00002    0 000000   A2    DAC    **
00003   -0 10 00005       JST*   ERCL
00004     151722          BCI    1,SR
00005     036707    ERCL  OCT    36707
                                      END

A1      000000      A2      000002      ERCL    000005

0000 WARNING OR ERROR FLAGS
DAP-16 MOD 2     REV. C     01-26-71
```

C-5 CONSTRUCTION OF THE HADIOS EXECUTIVE PACKAGE MK. 2

The following procedure was used when constructing the HADIOS EXECUTIVE PACKAGE MK.2. When the phrase 'load an object tape' is used it should be taken to mean that the object loader, LDR-APM, is used.

(1) Clear the computer's memory.

(2) Load an SLST of the 30 000 DEBUG into sectors 30-35 inclusive.

(3) Load an SLST of BASIC-16 into sectors 0-7 inclusive.

(4) Load an SLST of LDR-APM Rev E (entry point 16000_8) into sectors 13-16 inclusive.

(5) Load an object tape of PALAP into sector 17.

- P = 16000_8, A = 17000_8.

(6) Enter DEBUG (P = 30000_8) and carry out the following checks/modifications:

(a) Modify the BASIC-16 initialisation message (locations 166_8 to 203_8) to,

'HADIOS EXEC. MK.2 22.6.76 <CR,LF> '

(b) Modify the BASIC High Address initialisation procedure,

location (octal)	contents (octal)
7240 to 7243	NOP
7245	JMP 7301
7367	35337

(c) Modify location 2540_8 to 052726_8 to correct a fault in the BASIC-16 Interpreter, see reference (C.1).

(d) Check the following locations in LDR-APM,

location 15000_8 should contain 37777_8 - COMMON base

location 13167_8 should contain NOP - enables object tapes to be loaded from magnetic tape cassette.

(e) Make the modification outlined in Appendix B-5 to permit punch interrupts when using the I/O modification program (see Appendix B-2). The address AA, which is specified in Appendix B-5 is 770_8.

(7) Load an object tape of the program to relocate the ADT1-8 table from locations 61_8-70_8 to 757_8-766_i (see Appendix C-2).

- P = 16000_8.

(8) Load an object tape of the INPUT/OUTPUT Modification Program into locations 720_8-751_8

- P = 16000_8, A = 720_8

(9) Load object tapes of the HADIOS EXECUTIVE PROGRAM REVISION 01 (see Appendix C-1), the required Fortran Libraries, the special versions of subroutines F$ER and F$HT (see Appendix C-3) and the dummy versions of subroutines MYS1 and MYSR (see Appendix C-4). Loading proceeds as follows,

(a) P=16000_8, A=36000_8 - load HADIOS EXEC Rev. 01.

(b) P=16003_8, A=37102_8, B0$37574_8$

- load M$22, N$22, ARG$

(these routines having been previously obtained from the standard Fortran Library tape).

(c) P=16003_8, A=35340_8, B=35773_8 - load C$12, S$22, F$AT (these routines having been previously obtained from the standard Fortran Library tape).

(d) Press start - load special versions of F$ER and F$HT.

(e) Press start - load MYSR and MYS1.

(f) P=16002 - take a memory map (see Table C-5.1)

(10) Use PALAP to punch out an SLST of the contents of memory between locations 100_8 and 37577_8.

TABLE C-5.1 - MEMORY MAP OF THE LOADING PROCEDURE USED WHEN MAKING THE HADIOS EXECUTIVE PROGRAM MK.2.

```
*LOW    00551
*START  00551
*HIGH   37572
*NAMES  12677
*COMN   37777
*BASE   35775
*BASE   37575
C$12    35340
S$22    35372
A$22    35400
F$AT    35622
F$ER    35704
F$HT    35720
MYSI    35764
MYSR    35766
M$22X   37102
M$22    37102
D$22X   37263
D$22    37263
N$22    37522
H$22    37534
ARG$    37552

LC
```

C-6. ADDITION OF OTHER SUBROUTINES TO THE HADIOS EXECUTIVE PACKAGE MK.2.

In this version of the HADIOS EXECUTIVE PACKAGE provision has been made for the loading of additional subroutines required by the user. Two restrictions which apply to the addition of other subroutines are,

(i) Additional subroutines must be loaded below location 35377_8 using the loader incorporated in the package - entry point $P=16000_8$.

(ii) When these additional subroutines use COMMON, the COMMON base specified by the memory map of that particular load must be above location 37600_8.

Normally additional subroutines can be split into the following categories:

(a) A subroutine written in DAP-16MOD2 Assembler to initialise a special function directly concerned with the HADIOS EXECUTIVE PROGRAM. This subroutine can be accessed by a CALL (3, ---) in the BASIC program; this CALL statement should be included in the Initialisation section of the BASIC program.

This subroutine is entered from the HADIOS EXECUTIVE PROGRAM by a JMP (not a JST) through a local address word to location 36165_8. Thus the address which is hand patched into location 36165_8 is the first executable instruction of the subroutine and not a DAC **. Similarly an exit from the subroutine is achieved by a JMP* a local address word containing 4013_8 and not a JMP* the entry point.

One further restriction on the structure of this subroutine is that if the BASIC CALL statement has arguments (e.g. CALL (3,N,X)) then a pseudo subroutine entry point must be used so that F$AT can be used to fetch the argument addresses from the buffer in the BASIC-16 Interpreter.

As a result of the above restrictions, the overall structure of subroutine 3 will be,

```
IBUF  EQU   '230
  AA  ----------  First executable
                  instruction
      ----------

      ----------

      LDA   NEXT  See line 259 of Table C-1.1

      STA   *+2

      JMP   *+2

  BB  DAC   **    Pseudo subroutine entry
                  point
```

```
      JST*   FAT
      DEC    2
N     BSZ    1
X     BSZ    1
      ---------
      ---------
      ---------
      ---------
      JMP*   CLO6
CLO6  OCT    4013
FAT   XAC    F$AT
```

In the above example AA is the entry point of subroutine 3 and the address of this location should be stored in location 36165_8.

(b) A subroutine written in DAP-16MOD2 Assembler to generate a binary integer for transmission to experimental equipment as a 16 bit pattern by the digital output subinterface. This subroutine should be written using the conventional structure for DAP-16 subroutines and must store the binary integer to be output in location 36442_8 (WORD). This type of subroutine is not called directly from the BASIC program but is accessed by selecting the digital output option (see Table 6.1) and setting P_{10}=2 (see section 6.5.1).

(c) Additional subroutines written in FORTRAN IV or DAP-16MOD2 Assembler, e.g. Graphical or Matrix routines. These subroutines should be written using conventional structures and when loaded, the start addressed of the routines which are to be called from the BASIC program should be stored into locations 521_8 to 527_8, as required. In addition, the address 230_8 (DAC IBUF) should be stored into location 36166_8 and onwards for as many subroutines as are called.

There can be up to 7 of this type of subroutine and they will be numbered from 4 to 10 at the BASIC level.

C-6.1. NOTES

(i) When routines from the Fortran libraries are required by any of these additional subroutines they should be loaded as well. No attempts should be made to use the Fortran libraries loaded with HADIOS EXECUTIVE PROGRAM as these are required by the Counter Input half-full interrupt sections of the program.

(ii) The special version of subroutine FER, FHT should be used as this returns the program to the BASIC command mode when an execution error is detected by the Fortran libraries.

(iii) When BASE is required loading should take place by the second method described in section 6.2.5.1. Base section relocation is not advisable as this can be affected by the interrupts of the CPU.

C-7. ERROR MESSAGES GENERATED BY THE HADIOS EXECUTIVE PROGRAM Rev. 01

Error Message Format : ERROR XY LINE NNN,

where, NNN is a line number in the BASIC program.

and, XY is one of mnemonics defined by 1 to 7 below

1. TF - the clock interrupt frequency (P_2) is too small and so instructions in the Inter-scan Basic Processing section of the BASIC program were still being executed when a clock interrupt took place. Clearly this would lead to a backlog of processing at the BASIC level if it were allowed to go unchecked and so execution is terminated.

2. NR - an unidentifiable interrupt has occurred and execution is terminated.

3. NC - the HADIOS controller has interrupted the central processor but the interrupting device was not the counter. Execution is terminated.

4. UI - the execution of the program has been

terminated by user interference. This is caused by the user setting sense switch 2 which not only terminates execution but also disenables all interrupts.

5. RI - a real to integer conversion error has occurred in the HADIOS EXECUTIVE PROGRAM, i.e. conversion of a real number outside the range -32768 to +32767 has been attempted. This error can also occur in the Fortran libraries but this will result in a different error message - see 8.

6. S1 - An attempt has been made to CALL subroutine 3 when a special purpose routine has not been loaded.

7. SR - the HADIOS EXECUTIVE PROGRAM has attempted to CALL a special purpose digital output subroutine when it has not been loaded, i.e. digital output has been selected (see Table 6.1) and P_{10}=2 when no special subroutine is present.

8. Error messages preceded by a line containing the mnemonic FT represent execution errors which occurred in the Fortran libraries.

APPENDIX D

OP-16 REAL TIME OPERATING SYSTEM

CONTENTS

D-1 EXECUTIVE SYSTEM FUNCTIONS - DAP-16 PROGRAMMING DETAILS

Programs contained within an OP-16 system communicate with the Executive (and, in turn with other programs) by means of system functions. There are 8 standard system functions used within OP-16 systems and these are briefly discussed below together with the DAP-16 programming details.

D-1.1. FUNCTION 1 - REQUEST PROGRAM

Location	Contents	Comment
P	JST* XLNK	Function handler entrance
P+1	DEC 1	Function number
P+2	BCI 1, <A2>	Name of requested program
P+3	DAC <error return point>	
P+4	OCT <communication parameter>	
P+5		Normal return point

The requested program is not started up by this function but merely requested. The Executive Scheduler will start the requested program as soon as this is possible. All system functions enter the executive at a common entry point, XLNK.

D-1.2. FUNCTION 2 - SCHEDULE LABEL

Location	Contents	Comments
P	JST* XLNK	Function handler entrance
P+1	DEC 2	Function number
P+2	BCI 1,<A2>	Name of program in which label is to be scheduled
P+3	DAC <error return point>	
P+4	DAC <label to be scheduled>	
P+5		Normal return point

The major use of the Schedule Label function is to allow a program which services another to call the first one back after its service is complete. The program in which the label is scheduled must be in an active state.

D-1.3 FUNCTION 3 - CONNECT CLOCK

Location	Contents	Comments
P	JST* XLNK	Function handler entrance
P+1	DEC 3	Function number

P+2	BCI 1,<A2>	Name of program
P+3	DAC <error return point>	
P+4	DEC <time until first execution>	
P+5	DEC <interval between subsequent executions>	
P+6	DEC <base frequency>	
P+7		Normal return point

This function is used to connect a program to the clock for automatic initiation by the Executive Real Time Clock program. The requested program is first executed after the delay specified by location P+4 and thereafter on a periodic basis as specified by P+5.

D-1.4 FUNCTION 4 - DISCONNECT CLOCK

Location	Contents	Comments
P	JST* XLNK	Function handler entrance
P+1	DEC 4	Function number
P+2	BCI 1,<A2>	Name of program to be disconnected
P+3	DAC <error return point>	
P+4	DEC <base frequency>	
P+5		Normal return point

This function requests the executive to stop the periodic execution of a program connected to the Real Time Clock program.

D-1.5 FUNCTION 5 - CONNECT INTERRUPT

Location	Contents	Comments
P	JST* XLNK	Function handler entrance
P+1	DEC 5	Function number
P+2	DEC <interrupt reference number>	
P+3	DAC <error return point>	
P+4	DAC <start of interrupt response code>	
P+5		Normal return point

This function informs the Executive Interrupt Handler that the program wishes to respond to any interrupts generated by the peripheral device designated by the interrupt reference number. When such interrupts occur the executive will direct the program to the response code specified by the contents of location P+4.

D-1.6 FUNCTION 6 - DISCONNECT INTERRUPT

Location	Contents	Comments
P	JST* XLNK	Function handler entrance

P+1	DEC 6	Function number
P+2	DEC <interrupt reference number>	
P+3		Return point

This function informs the Executive that the calling program no longer wishes to respond to the named interrupt.

D-1.7 FUNCTION 7 - TERMINATE

Location	Contents	Comments
P	JST* XLNK	Function handler entrance
P+1	DEC 7	Function number

This function enables the program to inform the executive when it has finished execution. Control is returned to the Executive with no return to the program. If the program attempts to terminate with an interrupt still connected, an error message will be printed and the program disabled. Otherwise, all parameters associated with the running of the program will be reset.

D-1.8 FUNCTION 8 - WAIT

Location	Contents	Comments
P	JST* XLNK	Function handler entrance

P+1 DEC 8 Function number

This function informs the Executive that the program wishes to suspend execution until it is restarted by a schedule label function or by its interrupt response code.

D-2 EXTENDED HONEYWELL FORTRAN

In Revision J of the standard Series 16 FORTRAN IV Compiler, new capabilities were added to facilitate the writing of Fortran programs to run under the OP-16 operating system. This appendix describes the programming rules for this revision of the compiler.

D-2.1. ADDITIONAL COMPILER FEATURES

The additional features included in Revision J of the compiler are as follows:

1. OP-16 Statements
2. In-line Assembly Code
3. Octal Constants
4. Special DATA statement capabilities
5. The compiler library generation option
6. Additions to the FORTRAN MATHS. Library

D-2.2. OP-16 STATEMENTS

Throughout this section parameters are defined only once and so in some cases it may be necessary to refer to earlier definitions for an explanation.

D-2.2.1. THE HEADER STATEMENT

This statement is used to generate the program header for programs which are to be run under OP-16 and it has the following form:

HEADER N(P), LN

where, N is 1 or 2 character alphanumeric name,
LN is the number of words to generate for scheduled labels (default 4),
and, P is the communication parameter (optional)

D-2.2.2. REQUEST STATEMENT

This statement is used to generate a Request Program executive call (function 1) and it has the following form:

REQUEST I(P) ,, E

where, I is any integer expression, which is evaluated and used as the name of the program being requested,
and, E is the statement number to which control is to be passed if the request fails.

D-2.2.3. SCHEDULE STATEMENT

This statement is used to generate a schedule label executive call (function 2) and it has the following form:

SCHEDULE L(I) , E

where, L is the label to be scheduled. This may either be a statement number or an integer expression.

D-2.2.4. CONNECT CLOCK STATEMENT

This statement is used to generate a connect clock executive call (function 3) and it has the following form:

CONNECT CLOCK I (J,K), IB, E

where, J is time of first execution,
K is the interval between successive executions of the program,
and, IB is the Base frequency (see reference 6.8).

D-2.2.5. DISCONNECT CLOCK STATEMENT

This statement is used to generate a disconnect clock executive request (function 4) and it has the following form:

DISCONNECT CLOCK I (IB), E

D-2.2.6. CONNECT INTERRUPT STATEMENT

This statement is used to generate a connect interrupt executive function request (function 5) and it has the following form:

CONNECT INTERRUPT N(R),

where, R is the interrupt reference number.

D-2.2.7. DISCONNECT INTERRUPT STATEMENT

This statement is used to generate a disconnect interrupt executive request (function 6) and it has the following form:

DISCONNECT INTERRUPT N

D-2.2.8. TERMINATE STATEMENT

This statement is used to generate a terminate program executive request (function 7) and it has the following form:

TERMINATE

D-2.2.9. WAIT STATEMENT

This statement is used to generate a program wait executive request (function 8) and it has the following form:

WAIT

When such a statement is used the next executable statement must have a line number.

D-2.2.10. PRINT ERROR STATEMENT

This statement is used to generate a call to the system's error print routine (EPMOD) and has the following form:

```
PRINT ERROR I(IE)
```

where, IE is the error code.

D-2.2.11. INTERRUPT BLOCK STATEMENT

This statement is used to declare that a section of a program is interrupt response code. The statement has the following form,

```
INTERRUPT BLOCK N
```

D-2.2.12. INTERRUPT RETURN STATEMENT

This statement is used to exit from interrupt response code and has the following form:

```
INTERRUPT RETURN K
```

D-2.3. IN-LINE ASSEMBLY CODE

Revision J of the Fortran Compiler incorporates a small assembler which is capable of processing all I/O, shift and generic instructions. A line of assembly code consists of an 'A' in column one and a legal 3 character instruction mnemonic beginning in column 7, e.g.

Column 1	Column 7, 8, 9
A	SSM

D-2.4. OCTAL CONSTANTS

An octal constant may now be used anywhere an integer constant is permitted by preceding the six digit number with a colon (:), e.g.

: 17710

D-2.5. DATA STATEMENT ENHANCEMENTS

To facilitate compile time set up of OP-16 driver parameter lists, two extensions have been made to the data statement syntax. First, an integer variable may be initialised to the address of a statement by placing the statement number preceded by a dollar sign ($) in the constant list, e.g.

```
DATA I|$200|
```

The second extension is used to initialise one variable to the address of another variable, e.g.

```
      DATA N/M/
```

This would be equivalent to the assembly language statement:

```
      N DAC M
```

D-2.6. COMPILER LIBRARY GENERATION FACILITY

The special features available are,

(1) Register load:

```
      = (I+10)/2
```

loads the A register with (I+10)/2

(2) Register store:

```
      I =
```

will store the contents of the A register in I.

(3) Register test:

```
      IF ( ) S1, S2, S3
```

Execution proceeds to statements S1, S2 or S3 depending on the contents of the 'A' register.

D-2.7. FORTRAN LIBRARY ADDITIONS

Three routines have been added to the standard FORTRAN MATHS. Library to facilitate the writing of OP-16 programs in FORTRAN.

(1) Function LOC (arg).

This function returns as its value the address of its argument.

(2) Function IFETCH (arg).

This function returns as its value the contents of the memory location whose address is specified by the argument.

(3) Subroutine ISTORE (arg1, arg2)

This subroutine is used to set the memory location whose address is contained in the first argument to the value contained in the second argument.

N.B. All arguments for the above additions must be of the integer type.

D-3. FUNCTIONS SUPPORTED BY THE UTILITY PROGRAM (ONLCUP)

D-3.1. PRINT TIME

Format : SF = PT (CR)

Function : Prints the system time in (decimal) hours and minutes, based on the 24-hour clock.

D-3.2. REPLACE TIME

Format : SF = RT <hours>, <minutes> (CR)

Function : Replaces the system time with the (decimal) hours and minutes specified.

D-3.3. REQUEST PROGRAM

Format : SF = RP <name>, <parameter> (CR)

Function : Requests Executive System Function 1, i.e. request the execution of a program and optionally passes an octal parameter to it.

D-3.4. CONNECT CLOCK

Format : SF = CC <name>, <initial> , <interval>, <base> (CR)

Function : Connects a program to the system clock for automatic periodic execution.

D-3.5. DISCONNECT CLOCK

Format : SF = DC <name>, <base> (CR)

Function : Disconnects a program from the system clock.

D-3.6. REPLACE CORE

Format : SF = RC <CSA> (CR)

Function : Prints and optionally replaces the contents of an area of core starting at a specified address in core.

D-3.7. PRINT CORE

Format : SF = PC <CSA>, <CEA> (CR)

Function : Prints the contents of core between the specified addresses.

D-3.8. PRINT LIMITS

Format : SF = PL (CR)

Function : Prints the core protection limits. Utility functions are prevented from modifying core locations outside these limits.

D-3.9. REPLACE LIMITS

Format : SF = RL, <low limit>, <high limit> (CR)

Function : Replaces the core protection limits
with those specified.

D-3.10. TRANSFER - CORE TO PAPER TAPE

Format : SF = TR COPP <CSA>, <CEA> (CR)

Function : Transfers a specified area of core to paper tape in the form of binary information.

D-3.11. TRANSFER - PAPER TAPE TO CORE

Format : SF = TR PRCO <CSA>, <CEA> (CR)

Function : Transfers binary information from paper tape to a specified area of core.

D-3.12. VERIFY - PAPER TAPE AGAINST CORE

Format : SF = VE PRCO <CSA>, <CEA> (CR)

Function : Verifies the information from paper tape against a specified area of core.

For details of the various characters and abbreviations associated with the utility program see Table D-3.1.

TABLE D-3.1. - CHARACTERS OR ABBREVIATIONS ASSOCIATED WITH ONLCUP

CHARACTER OR ABBREVIATION	MEANING
name	Name of a system program (two ASCII characters)
initial	Time (decimal) until first execution
interval	Interval (decimal) between subsequent executions
base	Base frequency as follows: 0, initial - absolute time of day interval - minutes. 1, initial and interval in 50 ms units 2, initial and interval in secs 3, initial and interval in minutes
<CSA>	Core Start Address
<CEA>	Core End Address
(CR)	Carriage return - terminates a command line
/	Deletes the current line and causes SF = to be output
$	Requests the utility program
!	Terminates the utility program when typed in response to SF =

APPENDIX E

THE OLDFP EXECUTIVE

CONTENTS

E-1 HADIOS DRIVERS SUPERVISORY PROGRAM (H1-H6)

Table E-1.1 shows a listing of the supervisory program, described in section 6.6.1, which controls the requests of the standard Honeywell drivers HAD-ANI and HAD-DIO.

TABLE E-1.1 - HADIOS DRIVERS SUPERVISORY PROGRAM


```
                         * HADIOS DRIVER INTERFACES
                         *  (H1-H6)
                         *
                         * 15.02.77
                         *
                                SUBR  H1
                                SUBR  H2
                                SUBR  H3
                                SUBR  H4
                                SUBR  H5
                                SUBR  H6
                                ENT   BUFD
                                ENT   ENS1
                                ENT   ADBF
                                ENT   PCTR
                         *
                         * DIGITAL INPUT/OUTPUT
                         *
                                REL
                         * DIGITAL INPUT
00000   177777                  DEC   -1
00001   000000                  BSZ   5
00006   0 10 00417 H1           JST   PARA
00007   0 02 00020              LDA   PN1
00010   0 04 00150              STA   PAR
00011   0 02 00021              LDA   ST1
00012   0 04 00153              STA   PAR+3
00013   0 02 00022              LDA   DEV1
00014   0 04 00154              STA   PAR+4
00015   0 02 00024              LDA   BFD1
00016   0 04 00155              STA   PAR+5
00017   0 01 00134              JMP   RP
00020   144261     PN1          BCI   1,H1
00021   100001     ST1          OCT   100001
00022   0 000023   DEV1         DAC   *+1
00023   001070                  OCT   1070
00024   0 000156   BFD1         DAC   BUFD
                         * DIGITAL OUTPUT
00025   177777                  DEC   -1
00026   000000                  BSZ   5
00033   0 10 00417 H2           JST   PARA
00034   0 02 00045              LDA   PN2
00035   0 04 00150              STA   PAR
00036   0 02 00046              LDA   ST2
00037   0 04 00153              STA   PAR+3
00040   0 02 00051              LDA   BFD2
00041   0 04 00155              STA   PAR+5
00042   0 02 00047              LDA   DEV2
00043   0 04 00154              STA   PAR+4
00044   0 01 00134              JMP   RP
00045   144262     PN2          BCI   1,H2
00046   000001     ST2          OCT   1
00047   0 000050   DEV2         DAC   *+1
00050   001370                  OCT   1370
00051   0 000157   BFD2         DAC   BUFD+1
                         * COUNTER INPUT
00052   177777                  DEC   -1
```

TABLE E-1.1 - continued


```
00053   000000               BSZ    5
00060   0 10 00417  H3       JST    PARA
00061   0 02 00072           LDA    PN3
00062   0 04 00150           STA    PAR
00063   0 02 00073           LDA    ST3
00064   0 04 00153           STA    PAR+3
00065   0 02 00074           LDA    DEV3
00066   0 04 00154           STA    PAR+4
00067   0 02 00076           LDA    BFD3
00070   0 04 00155           STA    PAR+5
00071   0 01 00134           JMP    RP
00072   144263      PN3      BCI    1,H3
00073   100001      ST3      OCT    100001
00074   0 000075    DEV3     DAC    *+1
00075   000270               OCT    270
00076   0 000160    BFD3     DAC    BUFD+2
                    * PRESET COUNTER
00077   177777               DEC    -1
00100   000000               BSZ    5
00105   0 10 00417  H4       JST    PARA
00106   0 02 00120           LDA    PN4
00107   0 04 00150           STA    PAR
00110   0 02 00121           LDA    ST4
00111   0 04 00153           STA    PAR+3
00112   0 02 00122           LDA    DEV4
00113   0 04 00154           STA    PAR+4
00114   0 02 00124           LDA    BFD4
00115   0 04 00155           STA    PAR+5
00116   14 0270              OCP    '270
00117   0 01 00134           JMP    RP
00120   144264      PN4      BCI    1,H4
00121   000001      ST4      OCT    1
00122   0 000123    DEV4     DAC    *+1
00123   000370               OCT    370
00124   0 000161    BFD4     DAC    BUFD+3
                    * COUNTER INT RESPONSE
00125   0 000000    H5       DAC    **
00126   14 0370              OCP    '370
00127   0 10 00401           JST    TIME
00130   0 000166             DAC    BUFD+8
00131   0 02 00133           LDA    RTN
00132   -0 01 00125          JMP*   H5
00133   000000      RTN      BSZ    1
                    * REQUEST PROG - HD
00134   -0 10 00400 RP       JST*   XLNK
00135   000001               DEC    1
00136   144304               BCI    1,HD
00137   0 000431             DAC    ER1
00140   0 000150             DAC    PAR
                    * WAIT
00141   -0 10 00400          JST*   XLNK
00142   000010               DEC    8
                    * READ TIME
00143   001001      RT       INH
00144   0 10 00401           JST    TIME
00145   0 000162             DAC    BUFD+4
00146   000401               ENB
```

TABLE E-1.1 - continued


```
                             * SCHEDULE LABEL FOR RETURN
00147   0 01 00371           JMP   LB
00150   0 00 00000  PAR      ***   **
00151   0 000143             DAC   RT
00152   0 000433             DAC   ER2
00153   0 00 00000           ***   **
00154   0 00 00000           ***   **
00155   0 00 00000           ***   **
00156   000000      BUFD     BSZ   12
        000161      PCTR     EQU   BUFD+3
                             *
                             * ANALOGUE INPUT
                             *
00172   177777               DEC   -1
00173   000000               BSZ   5
00200   0 10 00417  H6       JST   PARA
                             * CLEAR BUFA
00201   0 35 00462           LDX   =-17
00202   140040               CRA
00203   1 04 00311           STA   BUFA+17,1
00204   0 12 00000           IRS   0
00205   0 01 00203           JMP   *-2
                             * INITIALISE ENSEMBLE COUNT
00206   0 02 00341           LDA   ENS1
00207   140407               TCA
00210   0 04 00340           STA   ENS
                             * REQUEST PROG MULTIPLEXER 0
00211   0 02 00332  MUX      LDA   STP1
00212   0 04 00334           STA   STP
00213   0 02 00461           LDA   =16
00214   0 04 00365           STA   PAR1+3
00215   0 02 00335           LDA   DAT1
00216   0 04 00370           STA   PAR1+6
00217   0 10 00351           JST   RPRG
                             * REQUEST PROG MULTIPLEXER 1
00220   0 02 00333           LDA   STP2
00221   0 04 00334           STA   STP
00222   0 02 00460           LDA   =1
00223   0 04 00365           STA   PAR1+3
00224   0 02 00336           LDA   DAT2
00225   0 04 00370           STA   PAR1+6
00226   0 10 00351           JST   RPRG
                             * STORE DATA
00227   0 35 00462           LDX   =-17
00230   1 02 00332  STDA     LDA   BUFB+17,1
00231   0404 72              LGR   6
00232   1 06 00311           ADD   BUFA+17,1
00233   1 04 00311           STA   BUFA+17,1
00234   0 12 00000           IRS   0
00235   0 01 00230           JMP   STDA
                             * TEST FOR ANOTHER SCAN
00236   0 12 00340           IRS   ENS
00237   0 01 00211           JMP   MUX
                             * READ TIME
00240   001001               INH
00241   0 10 00401           JST   TIME
00242   0 000162             DAC   BUFD+4
```

TABLE E-1.1 - continued


```
00243    000401            ENB
                      * ANALOGUE DATA PROCESSING
00244    0 02 00341        LDA   ENS1
00245   -0 10 00344        JST*  C12
00246   -0 10 00346        JST*  H22
00247    0 000342          DAC   ENSR
00250    0 02 00350        LDA   ADBF
00251    0 04 00347        STA   ADRS
00252    0 02 00462        LDA   =-17
00253    0 04 00340        STA   ENS
00254    0 35 00340   REP  LDX   ENS
00255    1 02 00311        LDA   BUFA+17,1
00256   -0 10 00344        JST*  C12
00257   -0 10 00345        JST*  D22
00260    0 000342          DAC   ENSR
00261   -0 10 00346        JST*  H22
00262   -0 000347          DAC*  ADRS
00263    0 12 00347        IRS   ADRS
00264    0 12 00347        IRS   ADRS
00265    0 12 00340        IRS   ENS
00266    0 01 00254        JMP   REP
                      * SCHEDULE LABEL FOR RETURN
00267    0 01 00371        JMP   LB
00270    000000       BUFA BSZ   17
00311    000000       BUFB BSZ   17
00332    002000       STP1 OCT   2000
00333    002020       STP2 OCT   2020
00334    000000       STP  BSZ   1
00335    0 000311     DAT1 DAC   BUFB
00336    0 000331     DAT2 DAC   BUFB+16
00337    000170       DADA OCT   170
00340    000000       ENS  BSZ   1
00341    000012       ENS1 DEC   10
00342    000000       ENSR BSZ   2
00344    0 000000     C12  XAC   C$12
00345    0 000000     D22  XAC   D$22
00346    0 000000     H22  XAC   H$22
00347    000000       ADRS BSZ   1
00350    016407       ADBF OCT   16407
                      * REQUEST PROG - HI
00351    0 000000     RPRG DAC   **
00352   -0 10 00400        JST*  XLNK
00353    000001            DEC   1
00354    144311            BCI   1,HI
00355    0 000435          DAC   ER3
00356    0 000362          DAC   PAR1
00357   -0 10 00400        JST*  XLNK
00360    000010            DEC   8
00361   -0 01 00351   RTRN JMP*  RPRG
00362    144266       PAR1 BCI   1,H6
00363    0 000361          DAC   RTRN
00364    0 000437          DAC   ER4
00365    0 00 00000        ***   **
00366    0 000334          DAC   STP
00367    0 000337          DAC   DADA
00370    0 00 00000        ***   **
                      *
```

TABLE E-1.1 - continued


```
                       * SCHEDULE LABEL
                       *
00371  -0 10 00400 LB    JST*  XLNK
00372   000002           DEC   2
00373   144324           BCI   1,HT
00374   0 000441         DAC   ER5
00375   000000           BSZ   1
                       *
                       * TERMINATE
                       *
00376  -0 10 00400 TE    JST*  XLNK
00377   000007           DEC   7
00400   101001     XLNK OCT    101001
                       *
                       * SUBROUTINE TO READ TIME
                       *
00401   0 000000   TIME DAC    **
00402  -0 02 00401       LDA*  TIME
00403   0 06 00415       ADD   INX4
00404   0 04 00414       STA   ADDR
00405   0 35 00457       LDX   =-4
00406  -0 02 00416       LDA*  CLK
00407  -0 04 00414       STA*  ADDR
00410   0 12 00000       IRS   0
00411   0 01 00406       JMP   *-3
00412   0 12 00401       IRS   TIME
00413  -0 01 00401       JMP*  TIME
00414   000000     ADDR BSZ    1
00415   040004     INX4 VFD    2,1,11,0,3,4
00416   041007     CLK  VFD    2,1,4,0,10,'1007
                       *
                       * SUBROUTINE PARA
                       *     (FETCH RETURN ADDRESS)
                       *
00417   0 000000   PARA DAC    **
00420   0 02 00417       LDA   *-1
00421   0 07 00456       SUB   =2
00422   0 04 00430       STA   TEMP
00423  -0 02 00430       LDA*  TEMP
00424   0 04 00430       STA   TEMP
00425  -0 02 00430       LDA*  TEMP
00426   0 04 00375       STA   LB+4
00427  -0 01 00417       JMP*  PARA
00430   000000     TEMP BSZ    1
                       *
                       * HERE TO REPORT ERRORS
                       *
00431   0 02 00450 ER1   LDA   ERR1
00432   0 01 00442       JMP   ERR
00433   0 02 00451 ER2   LDA   ERR2
00434   0 01 00442       JMP   ERR
00435   0 02 00452 ER3   LDA   ERR3
00436   0 01 00442       JMP   ERR
00437   0 02 00453 ER4   LDA   ERR4
00440   0 01 00442       JMP   ERR
00441   0 02 00454 ER5   LDA   ERR5
00442   0 35 00447 ERR   LDX   NAME
```

TABLE E-1.1 - continued

```
                              * HADIOS DRIVER INTERFACES    PAGE         6

     00443   001001             INH
     00444  -0 10 00455         JST*   ER
     00445   000401             ENB
     00446   0 01 00376         JMP    TE
     00447   144301        NAME BCI    1,HA
     00450   001110        ERR1 OCT    1110
     00451   001120        ERR2 OCT    1120
     00452   001130        ERR3 OCT    1130
     00453   001140        ERR4 OCT    1140
     00454   001150        ERR5 OCT    1150
     00455   0 000000      ER   XAC    EROR
     00456   000002             END
     00457   177774
     00460   000001
     00461   000020
     00462   177757
```

ADBF	000350	ADDR	000414	ADRS	000347	BFD1	000024
BFD2	000051	BFD3	000076	BFD4	000124	BUFA	000270
BUFB	000311	BUFD	000156	C12	000344	CLK	000416
D22	000345	DADA	000337	DAT1	000335	DAT2	000336
DEV1	000022	DEV2	000047	DEV3	000074	DEV4	000122
ENS	000340	ENS1	000341	ENSR	000342	ER	000455
ER1	000431	ER2	000433	ER3	000435	ER4	000437
ER5	000441	ERR	000442	ERR1	000450	ERR2	000451
ERR3	000452	ERR4	000453	ERR5	000454	H1	000006
H2	000033	H22	000346	H3	000060	H4	000105
H5	000125	H6	000200	INX4	000415	LB	000371
MUX	000211	NAME	000447	PAR	000150	PAR1	000362
PARA	000417	PCTR	000161	PN1	000020	PN2	000045
PN3	000072	PN4	000120	REP	000254	RP	000134
RPRG	000351	RT	000143	RTN	000133	RTRN	000361
ST1	000021	ST2	000046	ST3	000073	ST4	000121
STDA	000230	STP	000334	STP1	000332	STP2	000333
TE	000376	TEMP	000430	TIME	000401	XLNK	000400

```
0000 WARNING OR ERROR FLAGS
DAP-16 MOD 2     REV. C    01-26-71
```

E-2 EXECUTIVE CONFIGURATION MODULE (XCOM)

The Executive Configuration Module shown in Table E-2.1 consists of a series of tables which, amongst other things, list the programs and interrupts known to the Executive.

The writing of XCOM is perhaps the most difficult stage in the construction of any OP-16 system. If this module is not written correctly the Executive will be unable to function correctly and as a result the entire system will not operate as the user intended. The writing of XCOM is covered in the manufacturer's user's guide (E.1) Table E.2.2 shows a list of the components included in XCOM and Appendix E-5 reviews some of the more obscure points encountered during the implementation of the OLDFP Executive.

TABLE E-2.1 - EXECUTIVE CONFIGURATION MODULE (XCOM)


```
                         * XCOM   EXEC CONFIG MODULE
                         *
                         * O.L.D.F.P.
                         *
                         *  24.02.77
                         *
                         * HEADER
                         *
                                SUBR  XPLT
                                SUBR  XPET
                                SUBR  XIDT
                                SUBR  XID1
                                SUBR  XID2
                                SUBR  XPCT
                                SUBR  XCUT
                                SUBR  XIVT
                                SUBR  XLPT
                                SUBR  XFET
                                SUBR  XDCT
                                SUBR  XSPT
                                SUBR  XPFP
                                SUBR  XEXA
                                SUBR  XINT
                                SUBR  ER
                                SUBR  KB1
                                SUBR  CLK2
                                SUBR  CLK3
                                SUBR  MSD,XPLT
                                SUBR  LO1
           000000        LO1    EQU   0
                                REL
                         * XPLT   EXEC PROG LIST TABLE
           000000        XPLT   EQU   *
00000      141714        CL     BCI   1,CL
00001      0 000000             XAC   CL
00002      000221               OCT   221
00003      000000               OCT   0
00004      142720        ER     BCI   1,EP
00005      0 000000             XAC   EP
00006      000000               BSZ   1
00007      000001               OCT   1
00010      004000               OCT   4000
00011      145702        KB     BCI   1,KB
00012      0 000000             XAC   KB
00013      000000               BSZ   1
00014      000001               OCT   1
00015      002000               OCT   2000
00016      140723        AS     BCI   1,AS
00017      0 000000             XAC   AS
00020      000000               BSZ   1
00021      000003               OCT   3
00022      004000               OCT   4000
00023      000101               OCT   101
00024      150320        PP     BCI   1,PP
00025      0 000000             XAC   PP
00026      000000               BSZ   1
00027      000002               OCT   2
```

TABLE E-2.1 - continued

```
                            * XCOM   EXEC CONFIG MODULE  PAGE        2

00030   000111             OCT    111
00031   144261        H1   BCI    1,H1
00032   0 000000           XAC    H1
00033   000000             BSZ    1
00034   000003             OCT    3
00035   001000             OCT    1000
00036   000104             OCT    104
00037   144262        H2   BCI    1,H2
00040   0 000000           XAC    H2
00041   000000             BSZ    1
00042   000003             OCT    3
00043   001000             OCT    1000
00044   000105             OCT    105
00045   144263        H3   BCI    1,H3
00046   0 000000           XAC    H3
00047   000000             BSZ    1
00050   000003             OCT    3
00051   001000             OCT    1000
00052   000106             OCT    106
00053   144264        H4   BCI    1,H4
00054   0 000000           XAC    H4
00055   000000             BSZ    1
00056   000003             OCT    3
00057   001000             OCT    1000
00060   000107             OCT    107
00061   144265        H5   BCI    1,H5
00062   0 000000           XAC    H5
00063   000000             BSZ    1
00064   000000             OCT    0
00065   144266        H6   BCI    1,H6
00066   0 000000           XAC    H6
00067   000000             BSZ    1
00070   000002             OCT    2
00071   000110             OCT    110
00072   144304        HD   BCI    1,HD
00073   0 000000           XAC    HD
00074   000000             BSZ    1
00075   000002             OCT    2
00076   000102             OCT    102
00077   144311        HI   BCI    1,HI
00100   0 000000           XAC    HI
00101   000000             BSZ    1
00102   000002             OCT    2
00103   000103             OCT    103
00104   144324        HT   BCI    1,HT
00105   0 000000           XAC    HT
00106   000000             BSZ    1
00107   000001             OCT    1
00110   002000             OCT    2000
00111   144716        IN   BCI    1,IN
00112   0 000000           XAC    IN
00113   000000             BSZ    1
00114   000001             OCT    1
00115   002000             OCT    2000
00116   000000        FF   OCT    0
                      * XPET   EXEC PROG ENTRY TABLE
        000117        XPET EQU    *
```

TABLE E-2.1 - continued

```
                              * XCOM   EXEC CONFIG MODULE  PAGE      3

00117     0 000000   CL1  DAC   CL
00120     0 000016        DAC   AS
00121     0 000004        DAC   ER
00122     0 000061        DAC   HS
00123     0 000024        DAC   PP
00124     0 000104        DAC   HT
00125     0 000077        DAC   HI
00126     0 000072        DAC   HD
00127     0 000053        DAC   H4
00130     0 000031        DAC   H1
00131     0 000065        DAC   H6
00132     0 000037        DAC   H2
00133     0 000045        DAC   H3
00134     0 000111        DAC   IN
00135     0 000011   KB1  DAC   KB
00136     0 000116        DAC   FF
                     * XIDT   EXEC INT DEF TABLE
          000137     XIDT EQU   *
00137     34 0020    XS01 SKS   '20
00140     0 01 00143      JMP   *+3
00141     0 01 00150      JMP   XS02
00142     0 000000        XAC   CL
00143    -0 10 00142 XL01 JST*  *-1
00144    -0 10 00206      JST*  IH20
00145     0 000117        DAC   CL1
00146     000012     CLK2 DEC   10
00147     177773     CLK3 DEC   -5
00150     34 0404    XS02 SKS   '404
00151     0 01 00154      JMP   *+3
00152     0 01 00162      JMP   XS03
00153     0 000000        XAC   AI
00154    -0 10 00153 XL02 JST*  *-1
00155    -0 10 00206      JST*  IH20
00156     0 000000        DAC   **
00157     000004          DEC   4
00160     000001          OCT   1
00161     000040          OCT   40
00162     34 0270    XS03 SKS   '270
00163     101000          NOP
00164     0 01 00174      JMP   XS04
00165     0 000000        XAC   IH40
00166    -0 10 00165 XL03 JST*  *-1
00167    -0 10 00206      JST*  IH20
00170     0 000000        DAC   **
00171     000022          DEC   18
00172     000001          OCT   1
00173     000010          OCT   10
00174     34 0402    XS04 SKS   '402
00175     101000          NOP
00176    -0 10 00207      JST*  IH40
00177     0 000000        XAC   IH40
00200    -0 10 00177 XL04 JST*  *-1
00201    -0 10 00206      JST*  IH20
00202     0 000000        DAC   **
00203     000003          DEC   3
00204     000001          OCT   1
00205     000100          OCT   100
```

TABLE E-2.1 - continued


```
00206   0 000000   IH20 XAC   IH20
00207   0 000000   IH40 XAC   IH40
00210   000001     XID1 DEC   1
00211   000041          OCT   41
00212   74 0020         SMK   '20
00213   177774     XID2 DEC   -4
00214   0 000143        DAC   XL01
00215   0 000154        DAC   XL02
00216   0 000166        DAC   XL03
00217   0 000200        DAC   XL04
                   * XPCT     EXEC PROG COMM TABLE
00220   000001     XPCT DEC   1
00221   000011          DEC   9
00222   0 000000        XAC   FIFO
00223   0 000235        DAC   BF1
00224   0 000244        DAC   BF2
00225   0 000254        DAC   BF3
00226   0 000264        DAC   BF4
00227   0 000267        DAC   BF5
00230   0 000272        DAC   BF6
00231   0 000275        DAC   BF7
00232   0 000300        DAC   BF10
00233   0 000303        DAC   BF11
00234   0 000312        DAC   XPCE
00235   0 000237   BF1  DAC   *+2
00236   0 000237        DAC   *+1
00237   000000          BSZ   5
00244   0 000246   BF2  DAC   *+2
00245   0 000246        DAC   *+1
00246   000000          BSZ   6
00254   0 000256   BF3  DAC   *+2
00255   0 000256        DAC   *+1
00256   000000          BSZ   6
00264   0 000266   BF4  DAC   *+2
00265   0 000266        DAC   *+1
00266   000000          BSZ   1
00267   0 000271   BF5  DAC   *+2
00270   0 000271        DAC   *+1
00271   000000          BSZ   1
00272   0 000274   BF6  DAC   *+2
00273   0 000274        DAC   *+1
00274   000000          BSZ   1
00275   0 000277   BF7  DAC   *+2
00276   0 000277        DAC   *+1
00277   000000          BSZ   1
00300   0 000302   BF10 DAC   *+2
00301   0 000302        DAC   *+1
00302   000000          BSZ   1
00303   0 000305   BF11 DAC   *+2
00304   0 000305        DAC   *+1
00305   000000          BSZ   5
00312   000000     XPCE BSZ   1
                   * XCUT     EXEC CLK USERS TABLE
00313   0 000317   XCUT DAC   *+4
00314   000000          BSZ   3
00317   0 000324        DAC   *+5
00320   000000          BSZ   4
```

TABLE E-2.1 - continued

```
                        * XCOM   EXEC CONFIG MODULE  PAGE        5

00324    0 000331           DAC   *+5
00325    000000             BSZ   4
00331    0 000336           DAC   *+5
00332    000000             BSZ   4
00336    0 000343           DAC   *+5
00337    000000             BSZ   4
00343    000000             BSZ   5
                    * XIVT   EXEC INT VARS TABLE
00350    000005     XIVT DEC   5
00351    000000             BSZ   30
                    * XLPT   EXEC LABEL PARS TABLE
00407    000000     XLPT BSZ   6
00415    177777             DEC   -1
                    * XFET   EXEC FUNC ENTRY TABLE
00416    000010     XFET DEC   8
00417   -0 000000           XAC*  RP
00420   -0 000000           XAC*  SL
00421   -0 000000           XAC*  CC
00422   -0 000000           XAC*  DC
00423    0 000000           XAC   CI
00424    0 000000           XAC   DI
00425    0 000000           XAC   TE
00426    0 000000           XAC   WA
                    * XINT   EXEC INIT LOC
00427    0 000000   XINT XAC   SC
                    * XDCT   EXEC DEV CONFIG TABLE
00430    000000     XDCT BSZ   1
00431    000001             DEC   1
                    * XSPT   EXEC SPEC PARS TABLE
00432    000001     XSPT OCT   1
00433    001023     XPFP OCT   1023
00434    000000     XEXA OCT   0
00435    000000             OCT   0
00436    037777             OCT   37777
                            END

AS      000016   BF1     000235   BF10    000300   BF11    000303
BF2     000244   BF3     000254   BF4     000264   BF5     000267
BF6     000272   BF7     000275   CL      000000   CL1     000117
CLK2    000146   CLK3    000147   ER      000004   FF      000116
H1      000031   H2      000037   H3      000045   H4      000053
H5      000061   H6      000065   HD      000072   HI      000077
HT      000104   IH20    000206   IH40    000207   IN      000111
KB      000011   KB1     000135   LO1     000000A  PP      000024
XCUT    000313   XDCT    000430   XEXA    000434   XFET    000416
XID1    000210   XID2    000213   XIDT    000137   XINT    000427
XIVT    000350   XL01    000143   XL02    000154   XL03    000166
XL04    000200   XLPT    000407   XPCE    000312   XPCT    000220
XPET    000117   XPFP    000433   XPLT    000000   XS01    000137
XS02    000150   XS03    000162   XS04    000174   XSPT    000432

0000 WARNING OR ERROR FLAGS
DAP-16 MOD 2     REV. C     01-26-71
```

TABLE E-2.2 - COMPONENTS OF XCOM

TABLE	EXPLANATION
XPLT (line 32 of Table E-2.1)	The Executive Program List Table defines all the programs in the system and is central to all operations of the executive. There must be an entry for every program in the system, including device drivers.
XPET (line 113 of Table E-2.1)	The Executive Program Entry Table consists of a list of DAC pointers to the first word of each XPLT entry. It is the order of these pointers which determines the priority of the programs to which they relate. In a situation where two or more programs are scheduled simultaneously, the one with the highest priority will be run first (the program with the highest priority will be the one at the top of the table - in this case CL, see line 115 of Table E-2.1).
XIDT (line 131 of Table E-2.1)	The Executive Interrupt Definition Table contains the information necessary to identify an interrupt when it occurs and to cause control to be transferred to the user's interrupt response code. It consists of three parts. The first (labelled XIDT) contains all of the data specific to an interrupt. The second (labelled XID1) is used by the Executive to keep track of the interrupt status of the system. The third (labelled XID2) is used by the Executive to locate the interrupt data in XIDT.
XPCT (line 182 of Table E-2.1)	The Executive Program Communication Table establishes communication buffers for all of the programs requiring communication. The parameter passing subroutine used in this system is FIFO - see line 185 of Table E-2.1.

TABLE E-2.2 - continued

TABLE	EXPLANATION
XCUT (line 224 of Table E-2.1)	The Executive Clock User's Table contains information on all programs currently connected to the Real Time Clock. This system contains five entries and so five programs can be connected to the clock consecutively.
XIVT (line 236 of Table E-2.1)	The Executive Interrupted Variables Table is where the Executive stores the registers and status of interrupted programs. The maximum number of programs which may be interrupted is determined by the first entry in this table - line 237. In this case the maximum number of programs is 5.
XLPT (line 239 of Table E-2.1)	The Executive Label Parameter Table is where the Executive temporarily stores labels that have been scheduled by interrupt response code. The first entry (line 240) determines the number of labels which may be scheduled in this manner at any one time - in this case 3 i.e. two locations are required per label.
XFET XINT XDCT XSPT	Refer to user manual (E.1)

E-3 SPECIAL VERSION OF THE ERROR PRINT MODULE (EPMOD)

All of the peripheral device drivers supplied as standard software by the manufacturer are interrupt drivers. As a result of this, the Error Print Module (EPMOD), which is a type of ASR driver, outputs information in a way controlled by the interrupts generated by the teletype or VDU. In the case of the VDU the response is so rapid that another interrupt is generated before EPMOD can process the next character to be output. This causes the same character to be output a number of times and the resultant error message is difficult to decipher. To overcome this problem the standard form of EPMOD was modified so that if the next character is not ready, the code to sound the bell is output. This results in the correct error message being printed but accompanied by the bell being sounded a number of times - the user is left in no doubt that an error has occurred. Table E-3.1 shows a listing of the modified program (the modifications made are to be found between lines 143 and 174). The modified program can also be used when output to the teletype is required.

TABLE E-3.1 - SPECIAL VERSION OF EPMOD FOR USE WITH THE VDU


```
                            * EPMOD - MODIFIED FOR VDU
                            * -----
                            *
                            * 08.10.76
                            *
                            *  PROGRAM HEADER
                                   REL
                                   SUBR   EP
                                   SUBR   ED
00000   177777                     DEC    -1
00001   000000                     BSZ    2
                            *
                            *  CONNECT ASR INTERRUPT TO THIS PGM.
00003  -0 10 00265 EP              JST*   XLNK
00004   000005                     DEC    5
00005   000004                     DEC    4
00006   0 000110                   DAC    EP11
00007   0 000154                   DAC    EP20
                            *
                            *  GET ERROR CODE & NAME & SAVE FOR PRINTING
00010   0 35 00250 EP3             LDX    EPP1
00011   1 02 00000                 LDA    0,1
00012   101040                     SNZ
00013   0 01 00105                 JMP    EP10
00014   000401                     ENB
00015   0 04 00252                 STA    EPS1
00016   1 02 00001                 LDA    1,1
00017   0 04 00253                 STA    EPS2
00020   140040                     CRA
00021   1 04 00000                 STA    0,1
00022   1 04 00001                 STA    1,1
00023   0 04 00263                 STA    EPF1
                            *
                            *  HERE TO PRINT MESSAGE CHARACTERS
00024   0 02 00257                 LDA    EPK1
00025   001001                     INH
00026   14 0104                    OCP    '104
00027   0 10 00145                 JST    WAIT
                            *
00030   0 02 00260                 LDA    EPK2
00031   0 10 00145                 JST    WAIT
                            *
00032   0 02 00262                 LDA    EPK4
00033   0 10 00145                 JST    WAIT
                            *
00034   0 02 00261                 LDA    EPK3
00035   0 10 00145                 JST    WAIT
                            *
00036   0 02 00252                 LDA    EPS1
00037   0400 61                    LRL    15
00040   0 10 00200                 JST    PO
                            *
00041   0 35 00275                 LDX    =-5
00042   0 02 00252 EP1             LDA    EPS1
00043   0400 63                    LRL    13
00044   0 15 00254                 STX    EPS3
00045   0 10 00200                 JST    PO
```

TABLE E-3.1 - continued


```
00046   0 35 00254        LDX    EPS3
00047   0 12 00000        IRS    0
00050   0 01 00042        JMP    EP1
                    *
00051   0 02 00253        LDA    EPS2
00052   141340            ICA
00053   0 10 00145        JST    WAIT
00054   0 02 00253        LDA    EPS2
00055   0 10 00145        JST    WAIT
00056   140040            CRA
00057   0 10 00145        JST    WAIT
                    *
                    *  HERE TO PUSH UP REQUESTS IN STACK
00060   0 35 00250        LDX    EPP1
00061   001001      EP2   INH
00062   1 02 00002        LDA    2,1
00063   0 07 00274        SUB    =-1
00064   100040            SZE
00065   1 02 00002        LDA    2,1
00066   1 04 00000        STA    0,1
00067   1 02 00003        LDA    3,1
00070   1 04 00001        STA    1,1
00071   1 02 00000        LDA    0,1
00072   101040            SNZ
00073   0 01 00100        JMP    EP5
00074   000401            ENB
00075   0 12 00000        IRS    0
00076   0 12 00000        IRS    0
00077   0 01 00061        JMP    EP2
                    *
                    *  DECREMENT 'IN' POINTER FOR ENTRIES
00100   0 02 00251  EP5   LDA    EPP3
00101   0 07 00273        SUB    =2
00102   0 04 00251        STA    EPP3
                    *
                    *  GO TO TEST FOR MORE MESSAGES TO PRINT
00103   001001            INH
00104   0 01 00010        JMP    EP3
                    *
                    *  HERE WHEN ALL MESSAGES PRINTED
                    *  DISCONNECT INTERRUPT
00105  -0 10 00265  EP10  JST*   XLNK
00106   000006            DEC    6
00107   000004            DEC    4
00110  -0 10 00265  EP11  JST*   XLNK
00111   000007            DEC    7
                    *
                    *  HERE TO SAVE ERROR DATA FOR PRINTING
00112   0 000000    ED    DAC    **
00113   100002            SR4
00114  -0 01 00112        JMP*   ED
00115   000201            IAB
00116  -0 02 00251        LDA*   EPP3
00117   100040            SZE
00120   0 01 00143        JMP    ED20
00121   000201            IAB
00122   101040            SNZ
```

TABLE E-3.1 - continued


```
00123   140401          CMA
00124   140100          SSP
00125  -0 04 00251      STA*   EPP3
00126   0 12 00251      IRS    EPP3
00127  -0 15 00251      STX*   EPP3
00130   0 12 00251      IRS    EPP3
                   *
                   *  HERE TO REQUEST FOR PGM EP
00131   0 02 00264      LDA    ER
00132   0 06 00273      ADD    =2
00133   0 04 00144      STA    ED21
00134  -0 02 00144      LDA*   ED21
00135   0 03 00272      ANA    ='377
00136   100040          SZE
00137  -0 01 00112      JMP*   ED
                   *
                   *  HERE TO REQUEST FOR PGM EP
00140  -0 02 00144      LDA*   ED21
00141   0 05 00271      ERA    ='101
00142  -0 04 00144      STA*   ED21
00143  -0 01 00112 ED20 JMP*   ED
00144   000000     ED21 BSZ    1
                   *
                   *  SUBROUTINE WAIT
                   *
                   *   PUTS A REGISTER CONTENTS INTO EPS5 FOR
                   *   PRINTING , DOES WAIT FUNCTION FOR INT-
                   *   ERRUPT FROM ASR AND SETS FLAG
                   *
00145   0 000000   WAIT DAC    **
00146   001001          INH
00147   0 04 00256      STA    EPS5
00150   0 02 00270      LDA    =1
00151   0 04 00266      STA    FLAG
00152  -0 10 00265      JST*   XLNK
00153   000010          DEC    8
                   *
                   *
                   *  HERE TO RESPOND TO INTERRUPTS FROM ASR
00154   0 000000   EP20 DAC    **
00155   0 02 00266      LDA    FLAG
00156   101040          SNZ
00157   0 01 00171      JMP    EPA2
00160   140040          CRA
00161   0 04 00266      STA    FLAG
00162   0 02 00256      LDA    EPS5
00163   101040          SNZ
00164   0 01 00176      JMP    EP21
00165   74 0004         OTA    4
00166   101000          NOP
00167   0 02 00145 EP22 LDA    WAIT
00170  -0 01 00154      JMP*   EP20
                   *
                   *  HERE IF CHARACTER NOT READY
00171   0 02 00262 EPA2 LDA    EPK4
00172   74 0004         OTA    4
00173   101000          NOP
```

TABLE E-3.1 - continued


```
00174    140040           CRA
00175   -0 01 00154       JMP*  EP20
                    *
                    *   HERE TO REMOVE ASR INTERRUPT
00176    14 0004    EP21  OCP   4
00177    0 01 00167       JMP   EP22
                    *
                    *   SUBROUTINE PO (PRINT OCTAL VALUES)
00200    0 000000   PO    DAC   **
00201    0 04 00222       STA   POS1
00202    100040           SZE
00203    0 01 00207       JMP   PO1
00204    0 02 00263       LDA   EPF1
00205    101040           SNZ
00206    0 01 00217       JMP   PO2
00207    0 12 00263 PO1   IRS   EPF1
00210    0 02 00222       LDA   POS1
00211    0 06 00267       ADD   ='260
00212    000201           IAB
00213    0 04 00252       STA   EPS1
00214    000201           IAB
00215    0 10 00145       JST   WAIT
00216   -0 01 00200       JMP*  PO
00217    000201     PO2   IAB
00220    0 04 00252       STA   EPS1
00221   -0 01 00200       JMP*  PO
                    *
                    *   DATA LOCATION
00222    000000     POS1  OCT   0
00223    000000     EPT1  BSZ   20
00247    177777           DEC   -1
00250    0 000223   EPP1  DAC   EPT1
00251    0 000223   EPP3  DAC   EPT1
00252    000000     EPS1  OCT   0
00253    000000     EPS2  OCT   0
00254    000000     EPS3  OCT   0
00255    000000     EPS4  OCT   0
00256    000000     EPS5  OCT   0
00257    000215     EPK1  OCT   215
00260    000212     EPK2  OCT   212
00261    120305     EPK3  BCI   1, E
00262    000207     EPK4  OCT   207
00263    000000     EPF1  OCT   0
00264    0 000000   ER    XAC   ER
00265    101001     XLNK  OCT   101001
00266    000000     FLAG  OCT   0
00267    000260           END
00270    000001
00271    000101
00272    000377
00273    000002
00274    177777
00275    177773


ED      000112    ED20   000143    ED21   000144    EP     000003
EP1     000042    EP10   000105    EP11   000110    EP2    000061
```

TABLE E-3.1 - continued

EP20	000154	EP21	000176	EP22	000167	EP3	000010
EP5	000100	EPA2	000171	EPF1	000263	EPK1	000257
EPK2	000260	EPK3	000261	EPK4	000262	EPP1	000250
EPP3	000251	EPS1	000252	EPS2	000253	EPS3	000254
EPS4	000255	EPS5	000256	EPT1	000223	ER	000264
FLAG	000266	PO	000200	PO1	000207	PO2	000217
POS1	000222	WAIT	000145	XLNK	000265		

0000 WARNING OR ERROR FLAGS
DAP-16 MOD 2 REV. C 01-26-71
C

E-4 EXECUTIVE PROGRAM LINKS

As the OLDFP Executive is loaded into the computer's memory in a number of separate sections, it is necessary to satisfy the demands made by XCOM for the start addresses of programs which are to be controlled by the RTX-16 Executive. This is done by the program shown in Table E-4.1.

TABLE E-4.1 - OLDFP EXECUTIVE PROGRAM LINKS


```
                            * O.L.D.F.P. - LINKS - EXEC.
                            *
                            * 24.02.77
                            *
                                 SUBR  H1
                                 SUBR  H2
                                 SUBR  H3
                                 SUBR  H4
                                 SUBR  H5
                                 SUBR  H6
                                 SUBR  HI
                                 SUBR  HD
                                 SUBR  AS
                                 SUBR  PP
                                 SUBR  KB
              011006        H1   EQU   '11006
              011033        H2   EQU   '11033
              011060        H3   EQU   '11060
              011105        H4   EQU   '11105
              011125        H5   EQU   '11125
              011200        H6   EQU   '11200
              010546        HI   EQU   '10546
              012006        HD   EQU   '12006
              012416        AS   EQU   '12416
              016006        PP   EQU   '16006
              014003        KB   EQU   '14003
                                 END

AS    012416A   H1    011006A   H2    011033A   H3    011060A
H4    011105A   H5    011125A   H6    011200A   HD    012006A
HI    010546A   KB    014003A   PP    016006A

0000 WARNING OR ERROR FLAGS
DAP-16 MOD 2    REV. C    01-26-71
```

E-5 REVIEW OF OBSCURE POINTS ENCOUNTERED DURING THE IMPLEMENTATION OF THE OLDFP EXECUTIVE

During the implementation of the OLDFP Executive a number of points have had to be resolved as they are not covered adequately by the Manufacturer's User's Guide (E.1). It is hoped that the following sections provide clearer explanations of these topics.

E-5.1. COMMUNICATION

It is often necessary for different programs to communicate with one another and when this is so an entry is necessary in the XPLT table. In such cases the 6th word of the entry in XPLT should specify which queuing subroutine and which buffer in XPCT is to be used. For example, H1 requires communication so that it can obtain from the calling program, the return address to which it will schedule a return, via the schedule label function, after its services have been completed. Thus, line 64 of Table E-2.1 contains the entry 104_8, which specifies that the first and indeed the only queueing subroutine FIFO, and the fourth buffer in XPCT are used.

E-5.2 COORDINATION

In situations where two or more programs require the same services (in this context, the word service should be taken to mean one of the following:- Fortran library subroutines, Peripheral Device drivers, Peripheral Devices or core) they must be prevented from running concurrently. This is done by using the coordination option, which when required is implemented by the 5th word of the XPLT table. For example, since both of the Fortran Programs, HT and IN, are loaded together, they share some of the Fortran library subroutines which they require and clearly can not be executed concurrently. Thus, for both entries in the XPLT table, i.e. lines 106 and 111 of Table E-2.1, the value 2000_8 is inserted to prevent them running at the same time.

Care is needed when using the coordination option not to confuse peripheral devices with their drivers. To illustrate this point, although the Utility Program and both of the Fortran Programs (lines 47, 106 and 111 of Table E-2.1) all require the ASR driver, they do <u>not</u> require the ASR teletype itself. They must be coordinated only because they all require the ASR driver. However, the ASR driver and EPMOD (lines 42 and 52 of Table E-2.1) both require the teletype and so they must be coordinated.

E-5.3. SCHEDULE LABEL FUNCTION

This function is often used by device drivers to return control to the calling program after their services are complete. An example of this is given in the HADIOS Drivers Supervisory Program (lines 229 to 235 of Table E-1.1). It should be remembered, however, that to do this the driver must first fetch the return address (lines 260 to 272 of Table E-1.1) and that there must be communication between the calling program and the driver - see line 18 of Table E-13.1.

E-5.4. INTERRUPT

When a user supplies his own interrupt response code for a peripheral device the following rules must be adhered to in order to avoid serious Executive errors:-

(i) The program which connects interrupt to the response code must not be terminated until a Disconnect Interrupt function for that interrupt has been carried out.

(ii) If a label is scheduled by the interrupt response code the label must be in a program which is active.

E-6 EXECUTIVE MODULE LOADING PROCEDURE

The following loading procedure was used to construct the Executive Module (see Figure 6.3) of the OLDFP Executive.

(1) Clear the computer's memory

(2) Load an SLST of LDR-APM Revision E (Entry point, $P=37000_8$)

(3) Load an object tape of EXEC-A,

$P=37000_8$ $A=1000_8$

(4) Load an object tape of XCOM (See Appendix E-2),

Press start.

(5) Load an object tape of FIFO,

Press start.

(6) Load an object tape of FORTSY,

$P=37003_8$, $A=3766_8$

(7) Load routines S$2J, JBASE, F$ER, N$22, F$AT from RFMATH,

Press start (five times).

(8) Load routine M$2J from RFMATH,

$P=37003_8$, $A=1000_8$

(9) Load an object tape of the special EPMOD (See Appendix E-3),

$P=37003_8$, $A=13460_8$.

(10) Load an object tape of the Executive Program Links (See Appendix E-4),

Press start.

A memory map of the loading procedure is given in Table E-6.1.

TABLE E-6.1 - MEMORY MAP OF THE LOADING PROCEDURE USED IN MAKING THE EXECUTIVE MODULE

*LOW	00000	XLNK	02000	XID2	03347
*START	01000	FE10	02057	XPCT	03354
*HIGH	16006	FE20	02104	XCUT	03447
*NAMES	33036	FERR	02123	XIVT	03504
*COMN	37777	XSP1	02131	XLPT	03543
*BASE	03133	LB	02134	XFET	03552
*BASE	02776	RP	02167	XINT	03563
*BASE	04675	RPRO	02174	XDCT	03564
*BASE	05727	SL	02264	XSPT	03566
*BASE	06725	SLBL	02275	XPFP	03567
*BASE	07254	CC	02342	XEXA	03570
LO1	00000	DC	02444	FIFO	03574
*BASE	01774	CI	02516	SYSF	03766
EXEC	01000	DI	02626	OPED	04000
XPIC	01011	TE	02641	FDLS	07000
XPEP	01020	WA	03000	CONV	07260
XSSA	01023	AI	03063	S$2J	07320
OPTRAC	01037**	XPLT	03134	A$2J	07336
XHPT	01041**	MSD	03134	JBASE	07644
INP1	01044	ER	03140	JBAS	07644
SC	01103	HT	03241**	F$ER	07662
IH	01344	IN	03246**	N$22	07676
IH20	01373	XPET	03253	F$AT	07710
IHP1	01510	KB1	03271	H$2J	10000
IH40	01514	XIDT	03273	D$2J	10230
IHSC	01522	CLK2	03302	HI	10546
CLH	01526	CLK3	03303	H1	11006
CL	01530	XID1	03344	H2	11033

TABLE E-6.1 - continued

H3	11060
H4	11105
H5	11125
H6	11200
HD	12006
A3	12416
EP	13463
ED	13572
KB	14003
PP	16006
MR	

E-7 HADIOS DRIVERS LOADING PROCEDURE

The following loading procedure was used to construct the Hadios Driver Program (See Figure 6.3 and Section 6.6.1) used in the OLDFP Executive.

(1) Load an object tape of HAD-ANI,

$P=37000_8$, $A=10540_8$

(2) With the tape reader empty,

$P=37003_8$, $A=11000_8$

This is necessary as the next routine is to be force loaded and a program break is also required.

(3) Load an object tape of H1-H6,

P=37004

(4) Load the OP-16 Maths. library OPFRT2H,

P=37004.

(5) Load the standard Integer/Real Maths. library IRLIB,

Press start.

(6) Load the OP-16 Maths. library OPFRT3,

Press start.

(7) With the tape reader empty,

$P=37003_8$, $A=12000_8$.

(8) Load an object tape of HAD-DIO,

$P=37004_8$.

(9) Load an object tape of XLOCS,

Press start.

A memory map of the loading procedure is given in Table E-7.1.

TABLE E-7.1 - MEMORY MAP OF THE LOADING PROCEDURE USED IN MAKING THE HADIOS DRIVER PROGRAM

*LOW	00000	XPLP	01013	H2	11033
*START	10540	XPSP	01014	H3	11060
*HIGH	12166	XCCW	01015	H4	11105
*NAMES	33170	XEPP	01016	H5	11125
*COMN	37777	EROR	01016	BUFD	11156
COF2	00000	XLOD	01017	PCTR	11161
COF1	00000	XPEP	01020	H6	11200
CIF2	00000	XSYS	01021	ENS1	11341
CIF1	00000	XICF	01022	ADBF	11350
HTI2	00000	XSSA	01023	A$22	11464
HTI1	00000	XHLT	01024	S$22	11466
LPI2	00000	XPLT	01025	M$22	11470
LPI1	00000	XPCT	01026	D$22	11472
ASF2	00000	XCUT	01027	C$12	11574
ASF1	00000	XIDT	01030	N$22	11626
ROCB	00000	XID1	01031	H$22	11640
ROCA	00000	XID2	01032	ARG$	11656
XSTR	01000	XIVT	01033	ADDR	11700
XLNK	01001	XINT	01034	HD	12006
XPFP	01002	XDCT	01035		
XMIL	01003	XSPT	01036	LC	
XSEC	01004	XOPT	01037		
XMIN	01005	XCLC	01040		
XHR	01006	XHPT	01041		
XDAY	01007	XPET	01042		
XMSD	01010	XEXA	01043		
XPIC	01011	HI	01546		
XJBS	01012	H1	11006		

E-8 ASR DRIVER LOADING PROCEDURE

The following loading procedure was used to construct the ASR Driver Program (See Figure 6.3) used in the OLDFP Executive.

(1) Load an object tape of ASRD-S,

$P=37000_8$, $A=12410_8$.

(2) Load an object tape of ASF1 and ASF2,

$P=37003_8$, $A=13000_8$, $B=13450_8$

(3) Load an object tape of XLOCS,

Press start.

A memory map of the loading procedure is given in Table E-8.1.

TABLE E-8.1 - MEMORY MAP OF THE LOADING PROCEDURE USED IN MAKING THE ASR DRIVER PROGRAM

*LOW	00000	XPSP	01014	
*START	12410	XCCW	01015	LC
*HIGH	13440	XEPP	01016	
*NAMES	33366	EROR	01016	
*COMN	37777	XLOD	01017	
COF2	00000	XPEP	01020	
COF1	00000	XSYS	01021	
CIF2	00000	XICF	01022	
CIF1	00000	XSSA	01023	
MTI2	00000	XHLT	01024	
MTI1	00000	XPLT	01025	
LDI2	00000	XPCT	01026	
LPI1	00000	XCUT	01027	
ROCB	00000	XIDT	01030	
ROCB	00000	XID1	01031	
*BASE	13451	XID2	01032	
XSTR	01000	XIVT	01033	
XLNK	01001	XINT	01034	
XPFP	01002	XDCT	01035	
XMIL	01003	XSPT	01036	
XSEC	01004	XOPT	01037	
XMIN	01005	XCLC	01040	
XHR	01006	XHPT	01041	
XDAY	01007	XPET	01042	
XMSD	01010	XEXA	01043	
SPIC	01011	AS	12416	
XJBS	01012	ASF1	13000	
XPLP	01013	ASF2	13162	

E-9 UTILITY PROGRAM (ONLCUP) LOADING PROCEDURE

The loading procedure for the utility program (ONLCUP) is the same as shown in the manufacturer's user guide (E.2). As can be seen from the memory map given in Table E-9.1 the program was loaded into sectors 14 and 15 of the computer's memory.

TABLE E-9.1 - MEMORY MAP OF THE LOADING PROCEDURE USED IN MAKING THE UTILITY PROGRAM (ONLCUP)

*LOW	00000	XPIC	01011	KB	14003
*START	14000	XJBS	01012	FD2	14042
*HIGH	15756	XPLP	01013	ATSL	14061
*NAMES	32467	XPSP	01014	ERR	14071
*COMN	37777	XCCW	01015	TERM	14076
COF2	00000	XEPP	01016	RPRG	14100
COF1	00000	EROR	01016	PNME	14103
CIF2	00000	XLOD	01017	PAR3	14114
CIF1	00000	XPEP	01020	PAR4	14115
MTI2	00000	XSYS	01021	PAR5	14116
MTI1	00000	XICF	01022	PAR6	14117
LPI2	00000	XSSA	01023	FE	14120
LPI1	00000	XHLT	01024	RT1	14120
ASF2	00000	XPLT	01025	ER	14121
ASF1	00000	XPCT	01026	PRT	14126
ROCB	00000	XCUT	01027	DRIV	14126
ROCA	00000	XIDT	01030	C1	14130
*BASE	15664	XID1	01031	A1	14131
*BASE	14777	XID2	01032	KBD	14133
XSTR	01000	XIVT	01033	CRLF	14140
XLNK	01001	XINT	01034	COMA	14145
XPFP	01002	XDCT	01035	CRCK	14152
XMIL	01003	XSPT	01036	SP	14162
XSEC	01004	XOPT	01037	P1	14164
XMIN	01005	XCLC	01040	P2	14165
XHR	01006	XHPT	01041	CSA	14165
XDAY	01007	XPET	01042	P3	14166
XMSD	01010	XEXA	01043	CEA	14166

TABLE E-9.1 - continued

S1	14167	PUT	15442
S2	14170	CMN	15456
S3	14171	CL	15472
LL	14172	AKS	15506
HL	14173	APS	15513
AP	14177	PRS	15534
AK	14200	PPS	15541
IDN	14210	B2	15556
B1	14210	CKSH	15616
ODN	14211	EOF	15617
TR	14250		
FE	14312	MR	
ODK	14351**		
IDK	14354**		
VDK	14363**		
PT	14366		
RT	14413		
RP	14502		
CC	14532		
DC	14562		
RC	14600		
PC	14634		
PL	14714		
RL	14727		
OP	15000		
IP	15071		
VP	15127		
OE	15274		
OD	15320		
AE	15360		
DE	15376		
GET	15430		

E-10 PUNCH DRIVER LOADING PROCEDURE

The following loading procedure was used to construct the Paper Tape Punch Driver Program (see Figure 6.3) used in the OLDFP Executive.

(1) Load an object tape of HSPD-H,

$P=37000_8$, $A=16000_8$

(2) Load an object tape of C$OPF (contained on the standard Honeywell general input/output library, GEN-IOL),

Press start.

(3) Load an object tape of XLOCS,

Press start.

A memory map of the loading procedure is given in Table E-10.1.

TABLE E-10.1 - MEMORY MAP OF THE LOADING PROCEDURE USED IN MAKING THE PUNCH DRIVER PROGRAM

*LOW	00000	XPLP	01013	C$OP	16346
*START	16000	XPSP	01014		
*HIGH	16354	XCCW	01015	LC	
*NAMES	33355	XEPP	01016		
*COMN	37777	EROR	01016		
COF2	00000	XLOD	01017		
COF1	00000	XPEP	01020		
CIF2	00000	XSYS	01021		
CIF1	00000	XICF	01022		
MTI2	00000	XSSA	01023		
MTI1	00000	XHLT	01024		
LPI2	00000	XPLT	01025		
LPI1	00000	XPCT	01026		
ASF2	00000	XCUT	01027		
ASF1	00000	XIDT	01030		
ROCB	00000	XID1	01031		
ROCA	00000	XID2	01032		
XSTR	01000	XIVT	01033		
XLNK	01001	XINT	01034		
XPFP	01002	XDCT	01035		
XMIL	01003	XSPT	01036		
XSEC	01004	XOPT	01037		
XMIN	01005	XCLC	01040		
XHR	01006	XHPT	01041		
XDAY	01007	XPET	01042		
XMSD	01010	XEXA	01043		
XPIC	01011	PP	16006		
XJBS	01012	C$OPF	16346		

E-11 CORRECTIONS TO THE STANDARD SOFTWARE

Before the OLDFP Executive can be used the following three corrections to the standard software should be made.

E-11.1 CORRECTION TO H$2J

The following correction is necessary to ensure that the address of the calling program is not lost when M$2J and D$2J are entered with a zero numerator.

location	contents
M$2J + 12_8	JMP '772
M$2J + 242_8	JMP '770
770	STX* '774
771	JMP* '775
772	STX* '1774
773	JMP* '776
774	OCT 10535
775	OCT 10476
776	OCT 10046

E-11.2 CORRECTION TO HSPD-H

The following correction is necessary to prevent bit 8 being punched out in 'trailer tape' by the paper tape punch driver program.

location	contents
PP + 145_8	JMP PP+211_8

E-11.3 MODIFICATION TO THE REAL TIME CLOCK FREQUENCY

The fundamental unit of time in the Executive Real Time Clock Program is 0.1 sec and with this resolution it was found that the frequency of the signal being monitored by the Hadios counter input subinterface could not be determined accurately. To overcome this problem the resolution of the Executive Time Clock Program was changed to 0.02 secs by making the following modifications.

location	contents	
XMIL (1003_8)	61_8	(50_{10})
CLK2 (3302_8)	62_8	(49_{10})
CLK3 (3303_8)	177777_8	(-1)

E-12 ERROR MESSAGES GENERATED BY THE OLDFP EXECUTIVE.

Tables E-12.1 to E-12.6 list the errors which may occur when the OLDFP Executive is running. With the exception of those shown in Table E-12.6 these errors all indicate a serious malfunction either by one of the programs in the system or by a peripheral device and therefore they should be investigated immediately.

The general error message format is,

E NNNN XX

where, NNNN is the error identification number,
and, XX is a two letter program name.

Special formats apply to tables E-12.3.1, E-12.3.2 and E-12.6.

TABLE E-12.1 - ERROR MESSAGES GENERATED BY EXECA

ERROR MESSAGE	MEANING
E1 XX	a) Named program (xx) has asked for a function involving another program, but the requested program can not be found in the Executive Program List Table (XPLT). b) Named program has used an illegal function number.
E2 XX	Program has tried to schedule a label (function 2) in the named program (xx) which is either not active or which already has the maximum number of labels scheduled.
E3 XX	Named program (xx) has tried to connect the clock, but the Executive Clock User's Table (XCUT) is full.
E4 XX	Named program (xx) has tried to disconnect the clock without its having been connected.
E7 XX	Named program (xx) has attempted to terminate with an interrupt still connected. Program (xx) will be disabled; it can only be re-enabled by operator intervention.
E11 $$	An unidentified interrupt has occurred. The interrupt is ignored.
E12 XX	Named program (xx), which is interrupt driven has tried to schedule a label from its interrupt response code to its non-interrupt code, but the XLPT table is full.
E13 XX	The Executive has tried to schedule a label (from XLPT) in the named program (xx) which is not active or which already has the maximum number of labels scheduled.

TABLE E-12.2 - ERROR MESSAGES GENERATED BY THE OP-16 FORTRAN INPUT/OUTPUT EDITOR (OPED)

ERROR MESSAGE	MEANING
701XX	Digit precedes opening bracket
702XX	No opening bracket
703XX	No decimal point
704XX	Illegal character
705XX	Integer precedes -
706XX	Integer precedes/
707XX	Too many nested brackets
710XX	Non integer
711XX	Out of range for fixed point
712XX	Out of range for integer
713XX	Not T or F
714XX	Field width exceeded

TABLE E-12.3.1 - ERROR MESSAGES GENERATED BY THE RE-ENTRANT FLOATING POINT MATHS LIBRARY (RFMATH)

MNEMONIC	MEANING	ROUTINE
DZ	Division by zero	D$2J
SA	Arithmetic overflow (result > 2**127)	S$2J, A$2J
SD	Divisor unnormalised	D$2J
SM	Arithmetic overflow during multiplication or division	D$2J, M$2J

Error Message Format : E77 YY, where YY is the two letter Fortran object time error mnemonic - this format also applies to the errors in Table E-10.3.2.

TABLE E-12.3.2 - ERROR MESSAGES GENERATED BY THE STANDARD MATHS LIBRARY

MNEMONIC	MEANING	ROUTINE
AD	Over/underflow in double precision	A$66, S$66, A$66X,S$66X
DL	Negative or zero argument	DLOG, DLOG10, DLOG2
DT	Both arguments are zero	DATAN2
EQ	Exponential overflow adding integer to double precision exponent	A$81
EX	Exponential overflow during exponentiation	EXP
GO	Incorrect control variable in a GOTO statement	G$GA
II	First argument zero, second argument negative	E$11, E$11X
IM	Overflow/underflow during integer multiplication	M$11, M$11X
IZ	Integer division by zero	D$11, D$11X
LG	Log of negative or zero argument	ALOG, ALOG10, ALOGX
MD	Double precision multiplication or division overflow/underflow	D$66, M$66, D$66X, M$66X
PZ	Double precision division by zero	D$66, D$66X
RI	Integer too large when converted from real to integer	C$21
SQ	Negative argument	SQRT, SQRTX

TABLE E-12.4 - ERROR MESSAGES GENERATED BY THE HADIOS DRIVER PROGRAM

ERROR MESSAGE	MEANING	REPORTING PROGRAM
E600XX	When the driver attempted to use an input or output device, the device was found to be non-operational. Bit 1 of the device address will be set, and control will only be returned to the user's error address on completion of all other requested transfers.	HAD-DIO
E601XX	When the driver attempted to schedule a label to return control to the user's program an executive error occurred.	HAD-DIO
E610XX	When the driver attempted to schedule a label to return control to the user's program an executive error occurred.	HAD-ANI
E611XX	When the driver attempted to output the set up word to an analogue input, the device was found to be non-operational.	HAD-ANI
E612XX	When the driver attempted to input data from an analogue input, the device was found to be non-operational.	HAD-ANI
E1110HA	When HAD-DIO was requested an executive error occurred	H1-H6
E1120HA	Error return from HAD-DIO	H1-H6
E1130HA	When HAD-ANI was requested an executive error occurred	H1-H6
E1140HA	Error return from HAD-ANI	H1-H6
E1150HA	When the supervisory program attempted to schedule a label to return control to the user's program an executive error occurred	H1-H6

TABLE E-12.5 - ERROR MESSAGES GENERATED BY THE TELETYPE (ASRD-S) AND PUNCH (HSPD-H) DRIVER PROGRAMS

ERROR MESSAGE	MEANING	PROGRAM
E110XX	The named program has requested ASR reader/punch option. This option is not supported by this version of the ASR driver.	ASRD-S
E111XX	When requested by the named program the ASR was found to be busy or not ready.	ASRD-S
E112XX	When the named program requested data input the data was lost	ASRD-S
E70XX	When requested by the named program the punch was found to be busy or not ready.	HSPD-H

TABLE E-12.6 - ERROR MESSAGES GENERATED BY THE UTILITY PROGRAM (ONLCUP)

ERROR MESSAGE	MEANING
FE	Format or function error; parameter entered incorrectly.
LE	Limit error; attempt to modify core outside core protection limits.
HE	Header error; incorrect header record for the particular type of paper tape transfer requested.
CE	Checksum error detected during a paper tape function.
E100 KB	The utility program has been unable to complete the request and has been terminated.

E-13 USE OF THE OLDFP EXECUTIVE

The program shown in Table E-13.1 was written to test the OLDFP Executive. It contains most of the features discussed in Appendix D-2 as well as calling on all of the peripheral device drivers included in this version of the Executive. The loading procedure for this program will not be discussed here since it is not only straight-forward but also it is felt by the author that the techniques involved in constructing such programs is adequately covered in Chapter 9 and Appendix G.

Perhaps the easiest way of explaining how the program operates is to refer to Table E-13.2, which shows a log of a typical operating session. Following the loading of the two SLST's (one being the OLDFP Executive and the other the Fortran program) the system is started up in the usual manner by setting the P/Y register to 1000_8 and pressing the start button. The executive then performs its own initialisation procedures and waits for the first job request. In the case of the OLDFP Executive, the first program can

TABLE E-13.1 - EXAMPLE FORTRAN PROGRAM TO TEST THE OLDFP EXECUTIVE.

```
      HEADER HT                                               PAGE      1

      HEADER HT
      DIMENSION A(20),B(20),C(20),D(20),IP(5)
      LOGICAL ISTART,ITERM
      REAL ITIM1,ITIM
      EXTERNAL BUFA,BUFD
      DATA IH1,IH2,IH3,IH4,IH5,IH6/2HH1,2HH2,2HH3,2HH4,2HH5,2HH6/
      DATA IR1,IR2,IR3,IR4,IR6,IR7/$10,$30,$51,$35,$40,$70/
      DATA IHT/2HHT/
      DATA IEND/10/
      DATA IP(1),IP(2),IP(3),IP(4),IP(5)/2HHT,$69,$71,40,B/
      DATA IPP/2HPP/
      DATA ICH1,ICH2/1,17/
C
C DIGITAL INPUT
C
      ISTART=.FALSE.
      ITERM=.FALSE.
      REQUEST IH1[IR1],,110
      WAIT
   10 IDIGI=IFETCH(LOC(BUFD))
      IF(IDIGI.GE.16384) ISTART=.TRUE.
      IF(IDIGI.LT.0) ITERM=.TRUE.
C
C DIGITAL OUTPUT
C
      IDIGO=0
      IF(ITERM) GOTO 71
      IF(ISTART) GOTO 22
      ISCAN=0
      GOTO 24
   22 ISCAN=ISCAN+1
      IF(ISCAN.GT.IEND) GOTO 71
      IDIGO=IDIGO+ISCAN
      IDIGO=IDIGO+16384
   24 =IDIGO
A     @SSM
      IDIGO=
      CALL ISTORE(LOC(BUFD)+1,IDIGO)
      REQUEST IH2[IR2],,120
      WAIT
C
C CONNECT COUNTER INTERRUPT
C
   30 IF(.NOT.ISTART) GOTO 69
      IF(ISCAN.NE.1) GOTO 32
      WRITE(1,1200)
 1200 FORMAT(18H INPUT 1 FOR PUNCH)
      READ(1,2000) IFLAG
 2000 FORMAT(I1)
   32 REQUEST IH4[IR4],,140
      WAIT
   35 CONNECT INTERRUPT H5[18],150
      ITIM=FLOAT(IFETCH(LOC(BUFD)+4))+50.*(FLOAT(IFETCH(LOC(BUFD)
     1 +5))+60.*(FLOAT(IFETCH(LOC(BUFD)+6))+60.*FLOAT(IFETCH(LOC
     2 (BUFD)+7))))
C
C ANALOGUE INPUT
```

TABLE E-13.1 - continued

```
      HEADER HT                                                   PAGE     2


C
      REQUEST IH6[IR6],,160
      WAIT
C
C READ TIME
C
   40 IHRS=IFETCH(LOC(BUFD)+7)
      IMIN=IFETCH(LOC(BUFD)+6)
      ISEC=IFETCH(LOC(BUFD)+5)
C
C DATA PROCESSING
C
      DO 50 I=ICH1,ICH2
      B(I)=A(I)
   50 CONTINUE
C
C GET COUNTER FREQUENCY
C
      DISCONNECT INTERRUPT 18
      ITIM1=FLOAT(IFETCH(LOC(BUFD)+8))+50.*(FLOAT(IFETCH(LOC(BUFD)
     1 +9))+60.*(FLOAT(IFETCH(LOC(BUFD)+10))+60.*FLOAT(IFETCH(LOC
     2 (BUFD)+11))))
      IF(ITIM1.GE.ITIM) GOTO 55
      REQUEST IH3[IR3],,130
      WAIT
   51 IVAL=IFETCH(LOC(BUFD)+2)
      ITIM1=FLOAT(IFETCH(LOC(BUFD)+4))+50.*(FLOAT(IFETCH(LOC(BUFD)
     1 +5))+60.*(FLOAT(IFETCH(LOC(BUFD)+6))+60.*FLOAT(IFETCH(LOC
     2 (BUFD)+7))))
      GOTO 56
   55 IVAL=127
   56 IVAL1=IFETCH(LOC(BUFD)+3)
      IF(ABS(ITIM1-ITIM).LT.0.0001) GOTO 57
      CRATE=50.*FLOAT(IVAL-IVAL1)/(ITIM1-ITIM)
      GOTO 58
   57 CRATE=999999.9
C
C OUTPUT (ASR)
C
   58 IF(ISCAN.NE.1) GOTO 62
      WRITE(1,1000)
 1000 FORMAT(///1H ,10X,33HOP16 DYNAMIC LOGGING PROGRAM MK 1/
     11H ,11X,33(1H=))
   62 WRITE(1,1010) ISCAN
 1010 FORMAT(///9H SCAN NO.,I3/1H ,11(1H-))
      WRITE(1,1020)IHRS,IMIN,ISEC
 1020 FORMAT(//5H TIME,5X,I2,1H.,I2,1H.,I2)
      WRITE(1,1030)
 1030 FORMAT(//12H CHANNEL NO.,7X,5HVALUE,5X,15HCONVERTED VALUE/)
      DO 65 I=ICH1,ICH2
      IM1=I-1
      WRITE(1,1040) IM1,A(I),B(I)
 1040 FORMAT(1H ,5X,I2,10X,F8.4,5X,E12.6)
   65 CONTINUE
      WRITE(1,1100) CRATE
 1100 FORMAT(//18H COUNTER FREQUENCY,F11.2,4H(HZ))
      WRITE(1,1110) ITIM,ITIM1,IVAL1,IVAL
```

TABLE E-13.1 - continued

```
        HEADER HT                                                        PAGE        3

 1110 FORMAT(1H ,2(E20.14,1X),2(I5,1X))
C
C OUTPUT (HSP)
C
      IF(IFLAG.NE.1) GOTO 69
      REQUEST IPP[IP],,190
      WAIT
C
C TERMINATE THIS SCAN
C
   69 TERMINATE
C
C TERMINATE THE PERIODIC EXECUTION OF THIS PROGRAM
C
   71 IDIGO=0
      ISCAN=0
      CALL ISTORE(LOC(BUFD)+1,IDIGO)
      REQUEST IH2[IR7],,170
      WAIT
   70 WRITE(1,1060)
 1060 FORMAT(///36H **** HT DISCONNECTED FROM THE CLOCK)
      DISCONNECT CLOCK IHT[2],180
   72 TERMINATE
C
C HERE TO REPORT ERRORS
C
  110 PRINT ERROR IHT[:3010]
      GOTO 71
  120 PRINT ERROR IHT[:3020]
      GOTO 71
  130 PRINT ERROR IHT[:3030]
      GOTO 71
  140 PRINT ERROR IHT[:3040]
      GOTO 71
  150 PRINT ERROR IHT[:3050]
      GOTO 71
  160 PRINT ERROR IHT[:3060]
      GOTO 71
  170 PRINT ERROR IHT[:3070]
      GOTO 70
  180 PRINT ERROR IHT[:3100]
      GOTO 72
  190 PRINT ERROR IHT[:3200]
      GOTO 71
      END
$0   @
ND OF JOB
```

only be initiated by first entering the utility program and requesting its execution - subsequently this need not be the case. The utility program is entered by typing an $ character at the teletype or VDU. The system then responds by outputting SF=; see Table E-13.2, page 1. The user may now issue any of the commands allowed by the utility program ONLCUP - see Appendix D-3. In this particular session the following functions are used - see Table E-13.2, page 1:-

(i) Set the real time clock to 14 hours 10 minutes.

(ii) Print the time.

(iii) Transfer from core to paper tape locations 100_8 to 477_8.

(iv) Use the Replace Core function to modify the contents of location 3241_8 to 16406_8. This is the location in which the OLDFP Executive expects to find the start address of program HT.

(v)&(vi) Modify the analogue ensemble and the counter preset value - these are data values required by the Hadios Drivers Supervisory program - see Appendix E-1.

(vii)&(viii) The extent of core which can be modified using ONLCUP is now set to locations 16400_8 to 37777_8. This information is then printed out using the Print Limits function. This action prevents the user from accidentally modifying the OLDFP Executive.

(ix) The user now connects program HT to the clock. The initial interval is 60 units, subsequent intervals are 35 units and base frequency 2 specifies the units of time to be seconds.

(x) Finally, to enable program HT to run, the user exits from the utility program by typing an exclamation mark.

In this session two additional features of the utility program are displayed, these being

(i) FE - a function error message is output when the command TC is input because this is not an allowed command,

(ii) The use of the slash character (/) to terminate the replace core function.

After the initial delay of 60 seconds program HT is initiated for the first time by the executive. Table E-13.1 shows that the first action by this program is to access the digital input value to see whether the user wishes the program to start reading and printing out the values of the analogue and counter inputs, ISTART=.TRUE., or to disconnect the program from the clock, ITERM=.TRUE., or merely to terminate this particular run of program. It is via the digital input that the user communicates with the program: setting bit 2 of the digital input box causes ISTART to be set to .TRUE., whereas setting bit 1 causes ITERM to be set to .TRUE. The next section of the program, digital output, is used to inform the user what is happening. Bit 1 being lit means that the program HT is connected to the clock, bit 2 being lit means that the program is reading and printing out the values of the analogue and counter inputs. The remainder of the bits indicates the number of times the program has read the analogue and counter inputs (referred to as scans).

As can be seen from Table E-13.2, pages 2 to 4, the program performs 3 scans and is then disconnected from the clock by setting bit 1 of the digital input box; see page 5.

The user now reenters the utility program and modifies the analogue ensemble before initiating another run of program HT. Note that a Limit Error (LE) is obtained when an attempt is made to modify a core location outside of the limits set on page 1.

TABLE E-13.2 - LOG OF OPERATING SESSION

$	enter utility program
SF=RT 14,10	set real time clock (i)
SF=PT	print time (ii)
1410	
SF=T:	function error
FE	
SF=TR COPP 100,477	transfer to paper tape (iii)
SF=RC 3241	
3241 0 16406	set start address of HT (iv)
3242 2000 /	
SF=RC 11341	
11341 12 5	set analogue ensemble to 5 (v)
11342 40720 /	
SF=RC 11161	
11161 0 50	set counter preset value to 50_8 (vi)
11162 14 /	
SF=RL 16400,37777	set limits (vii)
SF=PL	print limits (viii)
16400 37777	
SF=PT	print time
1411	
SF=CC HT,60,35,2	connect program HT to the clock (ix)
SF=!	and exit (x)

TABLE E-13.2 - continued

INPUT 1 FOR PUNCH1

OP16 DYNAMIC LOGGING PROGRAM MK 1

SCAN NO 1

TIME 14.13. 8

CHANNEL NO.	VALUE	CONVERTED VALUE
0	177.2000	0.177200E 03
1	172.8000	0.172800E 03
2	166.6000	0.166600E 03
3	162.6000	0.162600E 03
4	155.2000	0.155200E 03
5	162.8000	0.162800E 03
6	164.8000	0.164800E 03
7	182.4000	0.182400E 03
8	177.2000	0.177200E 03
9	174.0000	0.174000E 03
10	0.0000	0.000000E 00
11	0.0000	0.000000E 00
12	34.8000	0.348000E 02
13	27.2000	0.272000E 02
14	127.0000	0.127000E 03
15	367.4000	0.367400E 03
16	0.0000	0.000000E 00

TABLE E-13.2 - continued

--

COUNTER FREQUENCY 541.67(HZ)
0.25594319999936E 07 0.25594379999961E 07 40 105

SCAN NO 2

TIME 14.13.45

CHANNEL NO.	VALUE	CONVERTED VALUE
0	177.0000	0.177000E 03
1	171.6000	0.171600E 03
2	167.0000	0.167000E 03
3	163.0000	0.163000E 03
4	155.2000	0.155200E 03
5	162.4000	0.162400E 03
6	165.2000	0.165200E 03
7	182.8000	0.182800E 03
8	176.8000	0.176800E 03
9	173.6000	0.173600E 03
10	0.0000	0.000000E 00
11	0.0000	0.000000E 00
12	35.0000	0.350000E 02
13	27.2000	0.272000E 02
14	127.2000	0.127200E 03
15	367.7999	0.367800E 03
16	0.0000	0.000000E 00

COUNTER FREQUENCY 541.67(HZ)

TABLE E-13.2 - continued

0.25612679999949E 07 0.25612739999928E 07 40 105

SCAN NO. 3

TIME 14.14.16

CHANNEL NO.	VALUE	CONVERTED VALUE
0	176.6000	0.176600E 03
1	171.4000	0.171400E 03
2	166.6000	0.166600E 03
3	162.2000	0.162200E 03
4	155.4000	0.155400E 03
5	162.6000	0.162600E 03
6	164.8000	0.164800E 03
7	183.2000	0.183200E 03
8	175.2000	0.175200E 03
9	173.8000	0.173800E 03
10	0.0000	0.000000E 00
11	0.0000	0.000000E 00
12	35.0000	0.350000E 02
13	27.6000	0.276000E 02
14	126.8000	0.126800E 03
15	367.6000	0.367600E 03
16	0.0000	0.000000E 00

COUNTER FREQUENCY 650.00(HZ)
0.25628009999968E 07 0.25628059999939E 07 40 105

TABLE E-13.2 - continued

```
**** HT DISCONNECTED FROM THE CLOCK

$
SF=PT
1415
SF=RC :1341
LE
SF=RL 0,37777
SF=RC 11341                    set analogue ensemble to 12₈(10₁₀)
 11341       5 12
 11342  40720 /
SF=CC HT,10,45,2               second run of program HT
SF=!
```

APPENDIX F

LISTINGS OF BASIC COMPUTER PROGRAMS

CONTENTS

TABLE F-1.1 - INSTRUMENT CALIBRATION PROGRAM

```
10  REM       ON-LINE INSTRUMENT CALIBRATION PROGRAM
11  REM       ========================================
15  DIM A(68)
20  CALL (4,1,0,0)
25  CALL (9,2)
30  D=9:D1,D2=0
35  N1=0:N2=31
40  PRINT "   ","ON-LINE CALIBRATION PROGRAM"
45  PRINT "   ","============================"
50  PRINT : PRINT
55  PRINT "INPUT SCAN INTERVAL";: INPUT I: PRINT
60  PRINT "INPUT NUMBER OF SCANS";: INPUT S1: PRINT
65  PRINT "INPUT TYPE OF COUNTER SCAN";: INPUT C1: PRINT
70  PRINT "INPUT COUNTER PRESET VALUE";: INPUT V: PRINT
75  PRINT "INPUT NUMBER OF CHANNEL TO BE CALIBRATED";: INPUT K1: PRINT
80  IF K1>68 GOTO 195
85  PRINT "INPUT ENSEMBLE";: INPUT E
90  CALL (9,3)
95  CALL (9,2)
100  N3,A2=0
105  PRINT "CHANNEL NO.";K1
110  PRINT "----------------"
115  PRINT
120  CALL (1,D,I,S1,E,N1,N2,C1,V,D1,D2,A(0))
125  IF K1=16 GOTO 145
130  IF K1=32 GOTO 155
135  PRINT A(K1)
140  GOTO 160
145  PRINT A(K1),A(32),A(33),A(36)
150  GOTO 160
155  PRINT A(K1),A(33),A(36)
160  N3=N3+1
165  A2=A2+A(K1)
170  CALL (2)
175  A2=A2/N3
180  PRINT : PRINT "AVERAGE READING IS";A2
185  PRINT
190  GOTO 75
195  END
```

TABLE F-2.1 - PRESSURE TO TEMPERATURE PROGRAM

```
?LIST

900   REM      PRESSURE TO TEMPERATURE CONVERSION - TEST PROGRAM
905   REM      ==========================================-=-=====
940   PRINT "TEMP.(DEG)"," PRESS.(KPA)"
945   PRINT
950   FOR P9=20,120,10
1000   REM              PRESSURE TO TEMPERATURE SUBROUTINE
1001   REM              ----------------------------------
1005   T9=LOG(P9/101.325)/13.3185
1010   T8=T9
1015   FOR I=1,4
1020   T7=T9+((.1299*T8+.6445)*T8+1.976)*T8*T8/13.3185
1025   IF ABS(T7-T8)<.1E-04 GOTO 1040
1030   T8=T7
1035   NEXT I
1040   T9=373.15/(1-T7)-273.15
1045   PRINT P9,T9
1050   NEXT P9
1055   END

?RUN
TEMP.(DEG)      PRESS.(KPA)

   20             60.0842
   30             69.1222
   40             75.8838
   50             81.3436
   60             85.9523
   70             89.9578
   80             93.5114
   90             96.7128
  100             99.6316
  110            102.318
  120            104.81

1055 EXIT
?
```

TABLE F-3.1 - STEADY STATE LOGGING PROGRAM

```
10   REM         STEADY STATE LOGGING PROGRAM - MK.3
11   REM         ==================================
15   DIM A(68),B(6,7),C(20),D(20),G(20),H(20),M(20),T(20),V(20)
19   Z6=0
20   Z9=0: GOSUB 1150
25   IF Z6=1 GOTO 40
35   PRINT "INPUT DATE (DAYS,MNTHS.,YRS.)";: INPUT D1,D2,D3: PRINT
40   PRINT "INPUT TIME (HRS.,MINS.,SECS.)";: INPUT O1,O2,O3: PRINT
44   REM
45   REM                        INITIALISATION SECTION
46   REM                        ----------------------
50   PRINT "INPUT INTERVAL BETWEEN SCANS (SECS.)";: INPUT I8: PRINT
55   PRINT "INPUT TOTAL NUMBER OF SCANS";: INPUT S: PRINT
57   PRINT "INPUT 1 IF HARDCOPIES REQUIRED";: INPUT Z8: PRINT
58   RESTORE
60   READ D,E,N1,N2,C,P,I1,I2,A1,A2,A3,A4,A5
64   REM
65   REM                        READ INSTRUMENT CALIBRATION DATA
66   REM                        --------------------------------
70   FOR I=0,17
75   READ C(I),D(I)
80   NEXT I
84   REM
85   REM                       INITIALISE ARRAYS
86   REM                       -----------------
90   FOR I=0,20
95   A(I),G(I),H(I),M(I),T(I),V(I)=0
100  NEXT I
119  REM
120  REM                        SCANNING SECTION
121  REM                        ----------------
125  PRINT : PRINT "SCANNING HAS STARTED":N9=1
130  PRINT "********************"
135  CALL (1,D,I8,S,E,N1,N2,C,P,I1,I2,A(0))
140  CALL (9,4)
145  FOR I=0,16
150  G(I)=G(I)+A(I)
155  H(I)=H(I)+C(I)+D(I)*A(I)
160  M(I)=M(I)+(C(I)+D(I)*A(I))↑2
165  NEXT I
166  IF A(32)=0 THEN A9=0: GOTO 170
167  A9=(127-P)/A(32)
170  G(17)=G(17)+A9
175  H(17)=H(17)+C(17)+D(17)*A9
180  M(17)=M(17)+(C(17)+D(17)*A9)↑2
182  IF N9=1 THEN H6=H(11)
183  N9=N9+1
185  CALL (2)
187  H7=H(11)
190  PRINT : PRINT "SCANNING HAS FINISHED"
195  PRINT "*********************"
214  REM
215  REM                        INPUT UNMEASURED VARIABLES
216  REM                        --------------------------
220  PRINT : PRINT "INPUT COOLING WATER RATE (KG/MIN)";: INPUT M2
221  PRINT
225  PRINT "INPUT CONDENSATE RATE (SECS/KG)";: INPUT M1: PRINT
```

TABLE F-3.1 - continued

```
230  PRINT "INPUT STEAM CONDENSATE RATE (SECS/KG)";: INPUT MS: PRINT
235  M1=1/M1:MS=1/MS
240  M2=M2/60
249  REM
250  REM                        CALCULATE STATISTICS
251  REM                        --------------------
255  FOR I=0, 17
260  G(I)=G(I)/S
265  H(I)=H(I)/S
270  M(I)=M(I)/S
275  M(I)=SQR(ABS(H(I)↑2-M(I)))
280  NEXT I
284  REM
285  REM              OUTPUT MEASUREMENTS, CONVERTED VALUES & STD. DEV.
286  REM              -------------------------------------------------
290  Z9=1: GOSUB 1150
320  PRINT " CHANNEL","MEASUREMENT"," CONV. VALUE"," STD.DEV.": PRINT
325  FOR I=0, 17
330  PRINT I,G(I),H(I),M(I)
335  M(I),G(I)=0
340  NEXT I
359  REM
360  REM                        SET UP HEAT & MASS BALANCE EQUATIONS
361  REM                        ------------------------------------
365  T(1)=H(2):T(2)=H(3):T(3)=H(9):T(4)=H(8)
370  T(8)=H(6):T(11)=H(7):T(12)=H(0):T(13)=H(1)
375  T(14)=H(5):T(15)=H(4):T(5)=100
380  M(2)=H(17):M(8)=H(16):M(12),M(13)=M2
385  V6=H(10)
387  M(11)=M1
390  H1=H(1)
395  P9=H(13): GOSUB 1000:T(9)=T9
400  P9=H(5): GOSUB 1000:T(10)=T9
405  P9=H(14): GOSUB 1000:T(6)=T9
410  C1=4.1868
419  REM
420  REM                        SOLUTION FOR FIRST EFFECT
421  REM                        -------------------------
425  M(1)=M(2)
430  V(3)=M(2)-M(8)
440  T9=T(3): GOSUB 1100:G(3)=G9
445  H(3)=V(3)*G(3)
450  H(1)=M(1)*C1*T(1)
455  H(2)=M(2)*C1*T(2)
460  T9=T(4): GOSUB 100:G(4)=G9
465  M(4)=(H(1)+H(3)-H(2)-V(3)*G(4))/(C1*T(4)-G(4))
470  V(4)=V(3)-M(4)
475  H(4)=V(4)*G(4)+M(4)*C1*T(4)
485  T(7)=(T(8)+T(3))/2
490  M(7)=M(8)
495  V(7)=V(3)
500  T9=T(7): GOSUB 1100:G(7)=G9
505  H(7)=V(7)*G(7)+M(7)*C*T(7)
510  T9=T(6): GOSUB 1100:G(6)=G9
515  V(6)=(H(7)-H(2))/(G(6)-C1*T(5))
520  M(5)=V(6)
525  H(5)=M(5)*C1*T(5)
```

TABLE F-3.1 - continued

```
527  H(6)=V(6)*G(6)
529  REM
530  REM                        HEAT LOSS FROM FIRST EFFECT
531  REM                        ---------------------------
535  H1=(MS-V(6))*G(6)
539  REM
540  REM                        SET UP MATRIX FOR 2ND. EFFECT SOLUTION
541  REM                        --------------------------------------
543  Y9=1
545  H(8)=M(8)*C1*T(8)
550  H(12)=M(12)*C1*T(12)
555  H(13)=M(13)*C1*T(13)
560  T9=T(9): GOSUB 1100:G(9)=G9
565  T9=T(10): GOSUB 1100:G(10)=G9
570  FOR I=0,5: FOR J=0,6
575  B(I,J)=0: NEXT J: NEXT I
580  B(1,1),B(1,3),B(1,4)=1
585  B(2,1)=T(10)*C1:B(2,2)=C1*(T(14)-T(15)):B(2,4)=G(10)
590  B(3,3),B(3,5)=1
595  B(4,1),B(4,4)=1
600  B(5,2)=C1*(T(15)-T(14)):B(5,3)=G(9):B(5,5)=T(15)*C1
605  B(1,6)=M(11):B(2,6)=H(4):B(3,6)=M(8)
610  B(4,6)=M(4)+V(4):B(5,6)=H(8)
619  REM
620  REM                        GAUSS-JORDAN ELIMINATION
621  REM                        ------------------------
623  N=5
625  FOR I=1,N
630  FOR J=I,N
635  IF ABS(B(J,I))>=.1E-02 GOTO 650
640  NEXT J
645  PRINT : PRINT "**** INVERSION ERROR": STOP
650  M9=B(I,I):I9=I
655  IF I+1>N GOTO 703
660  FOR J=I+1,N
665  IF ABS(B(I,J))<ABS(M9) GOTO 675
670  I9=J:M9=B(J,I)
675  NEXT J
685  IF I9=I GOTO 703
690  FOR K=I,N+1
695  Q=B(I,K):B(I,K)=B(I9,K):B(I9,K)=Q
700  NEXT K
703  M7=B(I,I)
705  FOR J=1,N+1:B(I,J)=B(I,J)/M7: NEXT J
710  FOR K=1,N
715  IF K=I GOTO 735
717  M8=B(K,I)
720  FOR J=1,N+1
725  B(K,J)=B(K,J)-M8*B(I,J)
730  NEXT J
735  NEXT K
740  NEXT I
749  REM
750  REM                        SOLUTION FOR SECOND EFFECT
751  REM                        --------------------------
755  M(10)=B(1,6):M(14),M(15)=B(2,6):V(9)=B(3,6)
760  V(10)=B(4,6):M(16)=B(5,6)
```

TABLE F-3.1 - continued

```
760  V(10)=B(4,6):M(16)=B(5,6)
765  T(16)=T(15)
770  H(9)=V(9)*G(9)
775  H(10)=V(10)*G(10)+M(10)*C1*T(10)
780  H(11)=M(11)*C1*T(11)
785  H(14)=M(14)*C1*T(14)
790  H(15)=M(15)*C1*T(15)
795  H(16)=M(16)*C1*T(16)
819  REM
820  REM                         HEAT LOSS FOR 2ND. EFFECT
821  REM                         -------------------------
823  M9=M1
824  M1=(M(12)*C1*(T(13)-T(12))+M(10)*C1*T(10)+V(10)*G(10)+V(9)*G(9))
825  M1=M1/(C1*10*T(11))
826  H2=(M(11)-M1)*C1*T(11)
829  REM
830  REM                         HEAT TRANSFER COEFFICIENTS
831  REM                         --------------------------
835  T1=(T(2)-T(1))/LOG((T(4)-T(1))/(T(4)-T(2)))
845  Q1=M(1)*C1*(T(2)-T(1))
850  X1=Q1/(A1*T1)
855  L6=-2.4068*T(6)+2501.6
860  T2=(T(5)+T(6))/2-(T(7)+T(2))/2:Q2=V(6)*(C1*(T(6)-T(5))+L6)
865  X2=Q2/(A2*T2)
870  T3=(T(14)-T(15))/LOG((T(10)-T(15))/(T(10)-T(14)))
875  Q3=M(15)*C1*(T(14)-T(15))
880  X3=Q3/(A3*T3)
885  T4=(T(13)-T(12))/LOG((T(11)-T(12))/(T(11)-T(13)))
890  Q4=M(12)*C1*(T(14)-T(15))
895  X4=Q4/(T4*A4)
899  REM
900  REM                        OUTPUT RESULTS
901  REM                        --------------
905  Z9=2: GOSUB 1150
910  PRINT " STREAM"," LIQUID"," VAPOUR","  TEMP."," ENTHALPY"
915  PRINT "        "," (KG/S)"," (KG/S)","  (DEG)","  (KJ/S)"
920  PRINT : FOR I=1,16
925  PRINT I,M(I),V(I),T(I),H(I)
930  NEXT I
933  Z9=3: GOSUB 1150
935  PRINT : PRINT "COMPARISON OF MEASURED & CALCULATED VALUES"
940  PRINT "------------------------------------------"
945  PRINT : PRINT "                   "," MEASURED"," CALCULATED"
955  PRINT "CONDENSATE   ",M(11),M1,"(KG/S)"
956  PRINT "STEAM        ",V6,V(6),"(KG/S)"
957  PRINT "STEAM COND.  ",M5,V(6),"(KG/S)"
959  M6=(H7-H6)*A5↑2*3.14159/((S-1)*18*4)
960  PRINT "ACCUMULATION ",M6,M(16),"(KG/S)"
965  PRINT : PRINT "HEAT LOSSES": PRINT "-----------": PRINT
967  H9=H(6)+H(1)+M2*C1*T(12)
970  PRINT "HEAT LOSS FROM FIRST EFFECT ",H1,"(KJ/S)"
975  PRINT "HEAT LOSS FROM SECOND EFFECT",H2,"(KJ/S)"
980  PRINT "TOTAL HEAT LOSS             ",H1+H2,"(KJ/S)"
981  PRINT "HEAT LOSS FROM EVAPORATOR   ",(H1+H2)*100/H9,"(%)"
982  PRINT : PRINT "HEAT TRANSFER COEFFICIENTS"
984  PRINT "--------------------------": PRINT
986  PRINT "PREHEATER  ",X1,"(KW/M↑2*K)"
```

TABLE F-3.1 - continued

```
988  PRINT "1ST. EFFECT",X2,"(KW/M↑2*K)"
990  PRINT "2ND. EFFECT",X3,"(KW/M↑2*K)"
992  PRINT "CONDENSER  ",X4,"(KW/M↑2*K)"
995  Z9=4: GOSUB 1150
996  GOTO 1400
999  REM
1000  REM                     PRESSURE TO TEMPERATURE SUBROUTINE
1001  REM                     ----------------------------------
1005  T9=LOG(P9/101.325)/13.3185
1010  T8=T9
1015  FOR I=1,4
1020  T7=T9+((.1299*T8+.6445)*T8+1.976)*T8*T8/13.3185
1025  IF ABS(T7-T8)<.1E-04 GOTO 1040
1030  T8=T7
1035  NEXT I
1040  T9=373.15/(1-T7)-273.15
1045  RETURN
1099  REM
1100  REM                     SPECIFIC VAPOUR ENTHALPY SUBROUTINE
1101  REM                     -----------------------------------
1105  G9=1.7798*T9+2501.6
1110  RETURN
1149  REM
1150  REM                     OUTPUT TITLE
1151  REM                     ------------
1155  IF Z9=0 GOTO 1175
1160  IF Z8=1 GOTO 1170
1165  CALL (4,0,0,0): PRINT : GOTO 1180
1170  CALL (9,1)
1175  CALL (4,1,0,0)
1180  REM
1183  PRINT "STEADY STATE LOGGING PROGRAM   - PAGE";Z9
1185  PRINT "============================================="
1190  PRINT
1191  IF Z9=0 GOTO 1198
1192  PRINT "DATE";D1;"/";D2;"/";D3: PRINT
1194  PRINT "TIME";O1;":";O2;":";O3
1196  PRINT : PRINT
1198  RETURN
1199  REM
1200  REM
1201  REM                     DATA
1202  REM                     ----
1205  REM          D,E,N1,N2,C,P,I1,I2,A1,A2,A3,A4,A5
1210  DATA 9,20,0,16,2,100,0,0,.38,.72,.672,1.15,.408
1215  REM                     C(I),D(I)
1220  DATA -1.2576,.119093
1225  DATA -1.2338,.119093
1230  DATA -1.3624,.119093
1235  DATA -1.2767,.119093
1240  DATA -1.2433,.119093
1245  DATA -1.1433,.119093
1250  DATA -1.2576,.119093
1255  DATA -1.2243,.119093
1260  DATA -1.2719,.119093
1265  DATA -1.1481,.119093
1270  DATA .91467E-02,.637724E-04
```

```
1275  DATA -.146455,.28159E-02
1280  DATA -2.68874,.222261
1285  DATA -4.02142,.216409
1290  DATA 98.174,.137993
1295  DATA -9.29804,.29318
1300  DATA -.293712E-02,.472866E-04
1305  DATA .25E-03,.695E-04
1394  REM
1395  REM       DYNAMIC LOG
1396  REM       -----------
1400  PRINT : PRINT "INPUT 1 IF DYNAMIC LOG REQD.";: INPUT Z6: PRINT
1405  IF Z6<>1 GOTO 1485
1410  PRINT "INPUT INTERVAL BETWEEN SCANS (SECS.)";: INPUT I8: PRINT
1415  PRINT "INPUT TOTAL NUMBER OF SCANS";: INPUT S: PRINT
1420  PRINT "SET SENSE SWITCH 4";: INPUT Z6
1425  CALL (1,D,I8,S,E,N1,N2,C,P,I1,I2,A(0))
1430  CALL (9,4)
1440  FOR I=0,16
1445  H(I)=C(I)+D(I)*A(I)
1450  PRINT I,",",H(I)
1455  NEXT I
1460  IF A(32)=0 THEN A9=0: GOTO 1465
1462  A9=(127-P)/A(32)
1465  H(17)=C(17)+D(17)*A9
1470  PRINT 17,",",H(17)
1475  CALL (2)
1480  STOP
1485  PRINT : PRINT "INPUT 1 TO RUN AGAIN";: INPUT Z6
1490  IF Z6=1 GOTO 20
1500  STOP
1510  END
```

TABLE F-4.1 - DYNAMIC MODEL SIMULATION PROGRAM

```
10   REM                    SIMULATION OF EVAPORATOR DYNAMIC MODEL
11   REM                    ======================================
15   DIM M(17),V(17),T(17),D(7),R(17),L(17),A(17),Z(17)
16   DIM E(7)
19   REM
20   REM                    READ PARAMETERS & PHYSICAL DATA
21   REM                    -------------------------------
24   REM                    VOLUMES - V
25   READ V1,V2,V3,V4,V5,V6
29   REM                    MASSES - W
30   READ W1,W2,W3,W4,W5,W6
34   REM                    SPECIFIC HEATS - C
35   READ C1,C2,C3,C4
39   REM                    AREAS - A
40   READ A1,A2,A3,A4,A5
44   REM                    LIQUID DENSITY
45   READ R1
49   REM                    HEAT TRANSFER CORRELATIONS
50   READ L1,M1,N1,O1
55   READ L2,M2,N2,O2
60   READ L3,M3,N3,O3
65   READ L4,M4,N4,O4
69   REM                    DENSITY/PRESSURE CORRELATION - B
70   READ B1
74   REM                    PRINT OUT FLAG
75   READ F9
99   REM
100  REM                    ASP DATA
101  REM                    --------
105  READ H,P,E,K
107  N9=INT(P/10+.5)
109  REM
110  REM                    READ INITIAL CONDITIONS,DATE,TIME
111  REM                    ---------------------------------
112  READ Z9,Z8,Z7,Z6,Z5,Z4
115  FOR I=1,16
120  READ M(I),V(I),T(I)
125  NEXT I
130  READ H1
135  READ P6,P7
140  T9=T(4): GOSUB 1100:P5=P9
145  P1=P5:P2=P6:P3=P7
149  REM
150  REM                    INITIALISE ASP & PRINT TITLE
151  REM                    ----------------------------
155  CALL (1,T)
157  I2=1
158  E9=1
160  IF F9=1 GOTO 180
162  PRINT "SIMULATION OF EVAPORATOR DYNAMIC MODEL"
164  PRINT "======================================"
166  PRINT : PRINT : PRINT "DATE";Z9;"/";Z8;"/";Z7
168  PRINT "TIME";Z6;":";Z5;":";Z4
170  PRINT : PRINT : PRINT "STREAM"," LIQUID","  VAPOUR","   TEMP.",
171  PRINT "MEASUREMENT"
172  PRINT "    "," (KG/S)","  (KG/S)","   (DEG.C)"
179  REM
```

TABLE F-4.1 - continued

```
180  REM                    CALCULATE HEAT TRANSFER COEFFICIENTS
181  REM                    ------------------------------------
185  U1=L1+M1*(T(4)-(T(1)+T(2))/2)+N1*M(1)+O1*V(3)
190  U2=L2+M2*(T(5)-(T(2)+T(7))/2)+N2*M(2)+O2*V(6)
195  U3=L3+M3*(T(10)-(T(15)+T(14))/2)+N3*M(15)+O3*V(4)
200  U4=L4+M4*(T(11)-(T(12)+T(13))/2)+N4*M(12)+O4*(V(10)+V(9))
204  REM
205  REM                    CALCULATE LATENT HEATS
206  REM                    ----------------------
210  FOR I=1,16
215  L(I)=-2.4068*T(I)+2501.6
220  NEXT I
224  REM
225  REM                    CALCULATE VAPOUR DENSITIES
226  REM                    --------------------------
230  FOR I=1,16
233  IF T(I)<0 GOTO 240
235  R(I)=EXP(1.93*LOG(T(I))-3.1487)/1000
240  NEXT I
244  REM
245  REM                    CALCULATE RATE OF CHANGE OF PRESSURE
246  REM                    ------------------------------------
250  D1=(P5-P1)/P
255  D2=(P6-P2)/P
260  D3=(P7-P3)/P
279  REM
280  REM                    STEADY STATE EQUATIONS
281  REM                    ----------------------
285  V(7)=(V(6)*L(5)-M(2)*C1*(T(7)-T(2)))/L(3)
286  V(3)=V(7)
287  M(7),M(8)=M(2)-V(3)-D2*V3*B1
291  M(4)=(M(1)*C1*(T(2)-T(1)))/L(4)
293  V(4)=V(3)-M(4)-D1*V2*B1
325  V(9)=(M(14)*C1*(T(14)-T(15))+M(8)*C1*(T(8)-T(15)))/L(15)
340  M(10)=(M(14)*C1*(T(14)-T(15)))/L(10)+M(4)
345  V(10)=M(4)+V(4)-M(10)-D3*V5*B1
350  M(11)=V(9)+M(10)+V(10)
369  REM
370  REM                    DIFFERENTIAL EQUATIONS
371  REM                    ----------------------
375  D(1)=(M(1)*C1*T(1)+U1*A1*(T(4)-(T(2)+T(1))/2)-M(2)*C1*T(2))
376  D(1)=D(1)/(V1*R1*C1+W1*C2)
380  D(2)=V(3)*C1*(T(3)-T(4))+V(3)*(L(3)-L(4))+M(4)*L(4)
381  D(2)=(D(2)-U1*A1*(T(4)-(T(2)+T(1))/2))/(W2*C3+V2*R(4)*C4)
385  D(3)=M(2)*C1*(T(2)-T(7))+L(7)*(M(8)-M(2))
386  D(3)=(D(3)+U2*A2*(T(5)-(T(7)+T(2))/2))/(W3*C2+V3*R(7)*C4)
390  D(4)=M(14)*C1*(T(15)-T(14))+U3*A3*(T(10)-(T(14)+T(15))/2)
391  D(4)=D(4)/(V4*R1*C1+W4*C2)
395  D(5)=M(4)*C1*(T(4)-T(10))+V(4)*C1*(T(4)-T(10))+V(4)*(L(4)-L(10))
396  D(5)=D(5)+L(10)*(M(10)-M(4))-U3*A3*(T(10)-(T(14)+T(15))/2)
397  D(5)=D(5)/(W5*C3+V5*R(10)*C4)
400  D(6)=(M(8)-V(9))/(A5*R1)
405  D(7)=M(12)*C1*(T(12)-T(13))+U4*A4*(T(11)-(T(13)+T(12))/2)
406  D(7)=D(7)/(V6*R1*C1+W6*C2)
407  FOR I=1,7:D(I)=D(I)-E(I): NEXT I
408  REM
409  REM
```

TABLE F-4.1 - continued

```
410  REM                       CALCULATE LOSS TERMS
411  REM                       --------------------
413  IF T>0 GOTO 420
414  IF E9=1 GOTO 417
415  FOR I=1,7:E(I)=D(I):D(I)=0: NEXT I
417  E9=0
419  REM
420  REM                       TEST FOR PRINT OUT & END OF RUN
421  REM                       -------------------------------
423  IF I2=2 THEN I2=1: GOTO 490
425  CALL (2,P,E,I1,I2)
430  IF I2=2 GOTO 550
435  IF I1=2 GOTO 750
479  REM
480  REM                       INTEGRATE INDEPENDENT VARIABLE
481  REM                       ------------------------------
490  CALL (3,T,H,K)
494  REM
495  REM                       INTEGRATE DEPENDENT VARIABLES
496  REM                       -----------------------------
505  CALL (4,T(2),D(1))
510  CALL (4,T(4),D(2))
515  CALL (4,T(7),D(3))
520  CALL (4,T(14),D(4))
525  CALL (4,T(10),D(5))
530  CALL (4,H1,D(6))
535  CALL (4,T(13),D(7))
549  REM
550  REM                       INPUT MEASUREMENTS
551  REM                       ------------------
555  IF I2=1 GOTO 185
557  IF T=0 THEN N8=N9:N9=1
558  FOR J1=1,N9
560  FOR I=0,17
565  INPUT J,A(I)
570  NEXT I
572  NEXT J1
573  IF T=0 THEN N9=N8
575  P1=P5:P2=P6:P3=P7
580  P6=A(12):P7=A(15)
585  T9=A(8): GOSUB 1100:P5=P9
590  M(1),M(2)=A(17)
595  V(6),M(5)=A(10)
600  T(3)=A(9):T(8)=A(6)
605  T(1)=A(2)
610  T(11)=A(7):T(12)=A(0):T(15),T(16)=A(4)
615  Z(2)=A(3):Z(4)=A(8):Z(7)=(A(6)+A(9))/2
620  Z(14)=A(5):Z(13)=A(1):Z(16)=A(11)
625  IF T<>0 GOTO 680
630  T(2)=Z(2):T(4)=Z(4)
635  T(7)=Z(7):T(13)=Z(13)
640  T(14)=Z(14):M(16),H1=Z(16)
679  REM
680  REM                       PRINT OUT SECTION
681  REM                       -----------------
685  IF F9=1 GOTO 695
690  PRINT : PRINT "TIME(SECS)";T: PRINT : GOTO 697
```

TABLE F-4.1 - continued

```
695  PRINT T
697  IF T=Ø THEN H1,Z(16)=1.14576
698  M(16)=H1
7ØØ  FOR I=1,16
7Ø3  IF F9=1 GOTO 72Ø
7Ø5  PRINT I,M(I),V(I),T(I),Z(I)
71Ø  NEXT I
715  GOTO 185
72Ø  FOR I=1,15
725  PRINT T(I),",",Z(I)
73Ø  NEXT I
735  PRINT M(16),",",Z(16)
74Ø  GOTO 185
75Ø  STOP
899  REM
9ØØ  REM                 DATA
9Ø1  REM                 ----
9Ø4  REM                 V
9Ø5  DATA .1593E-Ø2,.7779E-Ø2,.2918E-Ø2,.3947E-Ø2,.151E-Ø1,.6613E-Ø2
9Ø9  REM                 W
91Ø  DATA 4.91,22.3Ø4,8.192,8.922,13.452,14.95
914  REM                 C
915  DATA 4.1868,.3849,.4519,1.9
919  REM                 A(OUTSIDE)
92Ø  DATA .3955,.7238,.672,1.1834,.4Ø8↑2*3.14159/4
924  REM                 R1
925  DATA 985.22
929  REM                 L,M,N,O - H.T.C.
93Ø  DATA .721546E-Ø1,-.168154E-Ø3,8.23371,13.6Ø61
932  DATA 2.321Ø6,-.9523Ø3E-Ø1,7.11163,152.481
934  DATA 1.29675,.434556E-Ø1,.814137,-92.922
936  DATA 1.Ø1167,-.5Ø88Ø8E-Ø2,-1.64215,-.292726E-Ø3
939  REM                 B,F9
94Ø  DATA .6Ø74E-Ø2,Ø
944  REM                 H,P,E,K
945  DATA 5,1Ø,59Ø,4
949  REM                 DATE & TIME
95Ø  DATA 22,3,77,11,3Ø,Ø
954  REM                 INITIAL CONDITIONS
955  DATA .573269E-Ø1,Ø,17.4736
957  DATA .573269E-Ø1,Ø,65.1495
959  DATA Ø,.226856E-Ø1,76.254
961  DATA .5Ø8486E-Ø2,.176ØØ7E-Ø1,74.2Ø33
963  DATA .219599E-Ø1,Ø,1ØØ
965  DATA Ø,.219599E-Ø1,114.927
967  DATA .346413E-Ø1,.226856E-Ø2,75.8355
969  DATA .346413E-Ø1,Ø,75.4169
971  DATA Ø,.761744E-Ø2,62.Ø168
973  DATA .115818E-Ø1,.111Ø38E-Ø1,71.2Ø33
975  DATA .3Ø3Ø3E-Ø1,Ø,57.Ø866
976  DATA .436667,Ø,15.8766
979  DATA .436667,Ø,31.6Ø85
981  DATA 1.55227,Ø,58.1713
983  DATA 1.55227,Ø,55.8925
985  DATA .27Ø239E-Ø1,Ø,55.8925
989  REM                 H
```

TABLE F-4.1 - continued

```
990  DATA 1.12361
994  REM                  P6(12),P7(15)-PRESSURES
995  DATA 44.8479,29.6868
1099  REM
1100  REM                  TEMPERATURE TO PRESSURE SUBROUTINE
1101  REM                  ------------------------------------
1105  T8=1-373.15/(273.15+T9)
1110  P9=101.325*EXP(T8*(13.3185-T8*(1.967+T8*(6445+.1299*T8))))
1115  RETURN
1200  END
```

TABLE F-5.1 - GRAPH PLOTTING PROGRAM

```
10   REM        GRAPHICAL DISPLAY OF ON-LINE FILTERING RESULTS
11   REM        ==============================================
15   DIM A(20),B(20),C(20),D(20)
20   CALL (1,1,0,0)
25   PRINT "R.N.WEBB - ON-LINE KALMAN FILTERING"
30   PRINT "=================================="
35   PRINT : PRINT
40   PRINT "INPUT RUN NO.";: INPUT R1: PRINT
45   PRINT "INPUT TYPE OF FILTER";: INPUT F1: PRINT
54   REM
55   REM        INITIALISATION SECTION
56   REM        ----------------------
60   PRINT "INPUT STREAM NO. OF VARIABLE TO BE PLOTTED";
61   INPUT K1: PRINT
65   PRINT "INPUT VERTICAL MIN. & MAX. COORDINATES FOR VARIABLE";K1;
66   INPUT Y1,Y2: PRINT
75   S=30
80   N=60
84   REM
85   REM        INITIALISE GRAPHICS & SET VIRTUAL WINDOW
86   REM        ----------------------------------------
90   V1=0:V2=1770
95   S1=100:S2=905
100  S3=30:S4=600
105  V3=Y1:V4=Y2-Y1
110  GOSUB 1000
139  REM
140  REM        DRAW AXES
141  REM        ---------
145  T0=0:T9=V2
165  Y9=Y2-Y1:Y0=Y1: GOSUB 1100
169  REM
170  REM        OUTPUT SCALES & TITLES
171  REM        ----------------------
175  T=200:Y=770: GOSUB 900
180  PRINT "ON-LINE KALMAN FILTERING - FILTER";F1;" - RUN";R1
181  T=200:Y=748: GOSUB 900
183  PRINT "================================================="
185  T=70:Y=5: GOSUB 900
190  PRINT T0,"   ","TIME (SECS.)";
195  T=920: GOSUB 900
200  PRINT V2;
230  T=0
240  Y=615: GOSUB 900
245  PRINT Y2
250  Y=30: GOSUB 900
255  PRINT Y1
259  REM
260  REM        OUTPUT GRAPHICAL KEY
261  REM        --------------------
262  L=12:L2=34:L1=14
265  T=150:Y=580: GOSUB 900:T9=400
270  PRINT "ESTIMATE"
273  T=310
275  CALL (3,7,T,Y): CALL (3,9,T9,Y)
280  T=150:Y=550: GOSUB 900
285  PRINT "MEASUREMENT"
```

TABLE F-5.1 - continued

```
287  T=310
290  CALL (3,7,T,Y): CALL (4,3,T9,Y,L2)
295  T=150:Y=520: GOSUB 900
300  PRINT "PREDICTION"
305  T=310
310  CALL (3,7,T,Y): CALL (4,3,T9,Y,L)
315  T=150:Y=490: GOSUB 900
320  PRINT "COVARIANCE"
323  T=310
325  CALL (3,7,T,Y): CALL (4,3,T9,Y,L1)
329  REM
330  REM          OUTPUT LABELS & UNITS
331  REM          ---------------------
365  T=240:Y=700: GOSUB 900
370  IF K1=7 GOTO 385
375  PRINT "TEMPERATURE (DEG.C) - STATE ESTIMATE NO.";K1
382  GOTO 390
385  PRINT "SECOND EFFECT SEPARATOR HEIGHT (METRES)"
390  N1=1
391  T=0
392  FOR I=1,50: CALL (7,10): NEXT I
394  REM
395  REM          INPUT DATA TO BE PLOTTED
396  REM          ------------------------
400  FOR I=1,7: INPUT A(I): NEXT I
405  FOR I=1,7: INPUT B(I): NEXT I
410  FOR I=1,7: INPUT C(I): NEXT I
412  FOR I=1,7
414  IF I=4 GOTO 418
416  INPUT D(I)
418  NEXT I
419  REM
420  REM          PLOTTING SECTION
421  REM          ----------------
425  IF N1=1 GOTO 485
440  CALL (3,1,T-S,H1): CALL (3,3,T,A(K1))
445  CALL (3,1,T-S,H2): CALL (4,1,T,B(K1),L)
450  CALL (3,1,T-S,H3): CALL (4,1,T,C(K1),L1)
455  CALL (3,1,T-S,H4): CALL (4,1,T,D(K1),L2)
479  REM
480  REM          STORE LAST DATA SET
481  REM          -------------------
485  H1=A(K1):H2=B(K1):H3=C(K1):H4=D(K1)
499  REM
500  REM          TEST FOR END OF RUN
501  REM          -------------------
505  N1=N1+1
507  T=T+S
510  IF N1<=N GOTO 395
515  CALL (1,0,0,0): PRINT
520  STOP
899  REM
900  REM          POSITION CURSOR TO OUTPUT A LABEL
901  REM          ---------------------------------
905  CALL (3,7,T,Y)
910  CALL (7,3)
915  RETURN
```

TABLE F-5.1 - continued

```
999   REM
1000  REM        INITIALISE GRAPHICS, SET VIRTUAL & SCREEN WINDOWS
1001  REM        --------------------------------------------------
1005  CALL (1, 1, 0, 0)
1010  CALL (2, 1, V1, V2, V3, V4): REM       SET VIRTUAL WINDOW
1015  CALL (2, 0, S1, S2, S3, S4): REM       SET SCREEN WINDOW
1020  RETURN
1099  REM
1100  REM        DRAW AXES
1101  REM        ---------
1105  CALL (3, 1, T0, Y0)
1110  CALL (3, 3, T0+T9, Y0)
1115  CALL (3, 3, T0+T9, Y0+Y9)
1120  CALL (3, 3, T0, Y0+Y9)
1125  CALL (3, 3, T0, Y0)
1130  RETURN
1200  END
```

TABLE F-6.1 - DETERMINATION OF EIGENVECTORS

```
10   REM          DETERMINATION OF EIGENVECTOR MATRICES
11   REM          =====================================
15   DIM M(17),V(17),T(17),R(17),L(17),A(17),C(10,10),D(10,10),E(10,10)
16   DIM F(10,20)
19   REM
20   REM                          READ PARAMETERS & PHYSICAL DATA
21   REM                          -------------------------------
24   REM                          VOLUMES - V
25   READ V1,V2,V3,V4,V5,V6
29   REM                          MASSES - W
30   READ W1,W2,W3,W4,W5,W6
34   REM                          SPECIFIC HEATS - C
35   READ C1,C2,C3,C4,L9
39   REM                          AREAS - A
40   READ A1,A2,A3,A4,A5
44   REM                          LIQUID DENSITY
45   READ R1
49   REM                          HEAT TRANSFER CORRELATIONS
50   READ L1,M1,N1,O1
55   READ L2,M2,N2,O2
60   READ L3,M3,N3,O3
65   READ L4,M4,N4,O4
69   REM                          DENSITY/PRESSURE CORRELATION - B
70   READ B1
74   REM                          SAMPLING INTERVAL - P
75   READ P
80   T=0
109  REM
110  REM                          READ INITIAL CONDITIONS,DATE,TIME
111  REM                          ---------------------------------
112  READ Z9,Z8,Z7,Z6,Z5,Z4
115  FOR I=1,16
120  READ M(I),V(I),T(I)
125  NEXT I
130  READ H1
135  READ P6,P7
140  T9=T(4): GOSUB 1100:P5=P9
145  P1=P5:P2=P6:P3=P7
149  REM
150  REM                          PRINT TITLES
151  REM                          ------------
155  PRINT "DETERMINATION OF EIGENVECTOR MATRICES"
160  PRINT "====================================="
165  PRINT : PRINT : PRINT "DATE";Z9;"/";Z8;"/";Z7
170  PRINT "TIME";Z6;":";Z5;":";Z4
175  PRINT : PRINT
179  REM
180  REM                          CALCULATE HEAT TRANSFER COEFFICIENTS
181  REM                          ------------------------------------
185  U1=L1+M1*(T(4)-(T(1)+T(2))/2)+N1*M(1)+O1*V(3)
190  U2=L2+M2*(T(5)-(T(2)+T(7))/2)+N2*M(2)+O2*V(6)
195  U3=L3+M3*(T(10)-(T(15)+T(14))/2)+N3*M(15)+O3*V(4)
200  U4=L4+M4*(T(11)-(T(12)+T(13))/2)+N4*M(12)+O4*(V(10)+V(9))
204  REM
205  REM                          CALCULATE LATENT HEATS
206  REM                          ----------------------
210  FOR I=1,16
```

TABLE 6.1 - continued

```
215  L(I)=-2.4068*T(I)+2501.6
220  NEXT I
224  REM
225  REM                          CALCULATE VAPOUR DENSITIES
226  REM                          --------------------------
230  FOR I=1,16
233  IF T(I)<0 GOTO 240
235  R(I)=EXP(1.93*LOG(T(I))-3.1487)/1000
240  NEXT I
244  REM
245  REM                          CALCULATE RATE OF CHANGE OF PRESSURE
246  REM                          ------------------------------------
250  D1=(P5-P1)/P
255  D2=(P6-P2)/P
260  D3=(P7-P3)/P
279  REM
280  REM                          STEADY STATE EQUATIONS
281  REM                          ----------------------
285  V(7)=(V(6)*L(5)-M(2)*C1*(T(7)-T(2)))/L(3)
286  V(3)=V(7)
287  M(7),M(8)=M(2)-V(7)-D2*V3*B1
291  M(4)=(M(1)*C1*(T(2)-T(1)))/L(4)
293  V(4)=V(3)-M(4)-D1*V2*B1
325  V(9)=(M(14)*C1*(T(14)-T(15))+M(8)*C1*(T(8)-T(15)))/L(15)
340  M(10)=(M(14)*C1*(T(14)-T(15)))/L(10)+M(4)
345  V(10)=M(4)+V(4)-M(10)-D3*V5*B1
350  M(11)=V(9)+V(10)+M(10)
369  REM
370  REM                          SET UP COEFFICIENTS MATRIX
371  REM                          --------------------------
385  FOR I=0,6: FOR J=0,6
390  C(I,J)=0
395  NEXT J: NEXT I
400  E1=V1*R1*C1+W1*C2
402  E2=V2*R(4)*C4+W2*C3
404  E3=V3*R(7)*C4+W3*C2
406  E4=V5*R(10)*C4+W5*C3
408  E5=V6*R1*C1+W6*C2
410  E6=V4*R1*C1+W4*C2
412  C(0,0)=(-M(2)*C1-U1*A1/2)/E1
414  C(0,1)=(U1*A1)/E1
416  C(1,0)=(U1*A1/2)/E2
418  C(1,1)=(-V(3)*C1-V(3)*L9+M(4)*L9-U1*A1)/E2
420  C(1,2)=(V(3)*C1+V(3)*L9)/E2
422  C(2,0)=(M(2)*C1+U2*A2/2)/E3
424  C(2,2)=(-M(2)*C1+L9*(M(8)-M(2))-U2*A2/2)/E3
426  C(3,1)=(M(4)*C1+V(4)*(C1+L9))/E4
428  C(3,3)=(-M(4)*C1-V(4)*(C1+L9)+L9*(M(10)-M(4))-U3*A3)/E4
430  C(3,5)=(U3*A3/2)/E4
432  C(4,4)=(-M(12)*C1-U4*A4/2)/E5
434  C(5,3)=(U3*A3)/E6
436  C(5,5)=(-M(14)*C1-U3*A3/2)/E6
438  C(6,6)=1
444  REM
445  REM                          DETERMINATION OF EIGENVALUES & VECTORS
446  REM                          --------------------------------------
447  PRINT : PRINT "TIME(SECS)";T: PRINT "-----------------"
```

TABLE F-6.1 - continued

```
448  PRINT
449  N1=7
450  F1=3:I1=200
452  G1=.1E-06:G2=.1E-02:G3=.1E-03
454  G4=10
456  S1=0:S2=1:G5=1
460  CALL (5,N1,C(Ø,Ø),D(Ø,Ø),E(Ø,Ø),F1,I1,G1,G2,G3,G4,G5,S1,S2,T1,T2,I2
462  IF T1=1 THEN  PRINT "SUB DIAGONAL ELEMENTS TOO LARGE"
464  IF T2=1 THEN  PRINT "EIGENVALUES TOO CLOSE"
466  IF I2>=I1 THEN  PRINT "MAXIMUM ITERATIONS EXCEEDED"
469  REM
470  REM                    INVERT EIGENVECTOR MATRIX
471  REM                    -------------------------
472  E9=1
473  FOR I=Ø,6: FOR J=Ø,6:E(I,J)=E(I,J)*E9: NEXT J: NEXT I
474  R=Ø
475  CALL (6,E(Ø,Ø),F(Ø,Ø),N1,R)
480  IF R=1 THEN  PRINT "NO INVERSE"
489  REM
490  REM                    PRINT OUT SECTION
491  REM                    -----------------
494  PRINT
495  PRINT "COEFFICIENTS MATRIX": PRINT
497  FOR I=Ø,6: FOR J=Ø,6
499  PRINT C(I,J);: NEXT J
501  PRINT : NEXT I
502  PRINT
503  PRINT "EIGENVALUES": PRINT
505  FOR I=Ø,6: PRINT D(I,I);: NEXT I: PRINT
509  PRINT
510  PRINT "EIGENVECTOR MATRIX": PRINT
515  FOR I=Ø,6: FOR J=Ø,6
520  PRINT E(I,J);: NEXT J
525  PRINT : NEXT I
529  PRINT
530  PRINT "INVERSE EIGENVECTOR MATRIX": PRINT
535  FOR I=Ø,6: FOR J=Ø,6
540  PRINT F(I,J);: NEXT J
545  PRINT : NEXT I: PRINT
549  REM
550  REM                    INPUT MEASUREMENTS
551  REM                    ------------------
560  FOR I=Ø,17
565  INPUT J,A(I)
570  NEXT I
575  P1=P5:P2=P6:P3=P7
580  P6=A(12):P7=A(15)
585  T9=A(8): GOSUB 1100:P5=P9
590  M(1),M(2)=A(17)
595  V(6),M(5)=A(10)
600  T(3)=A(9):T(8)=A(6)
605  T(1)=A(2)
610  T(11)=A(7):T(12)=A(Ø):T(15),T(16)=A(4)
615  T(2)=A(3):T(4)=A(8):T(7)=(A(6)+A(9))/2
620  T(14)=A(5):T(13)=A(1):T(16)=A(11)
625  T=T+P
627  IF T>590 THEN  STOP
```

TABLE F-6.1 - continued

```
6?0  GOTO 180
900  REM                    DATA
901  REM                    ----
904  REM                    V
905  DATA .1593E-02,.7779E-02,.2918E-02,.3947E-02,.151E-01,.6613E-02
909  REM                    W
910  DATA 4.91,22.304,8.192,8.922,13.452,14.95
914  REM                    C
915  DATA 4.1868,.3849,.4519,1.9,-2.4068
919  REM                    A(OUTSIDE)
920  DATA .3955,.7238,.672,1.1834,.408↑2*3.14159/4
924  REM                    R1
925  DATA 985.22
929  REM                    L,M,N,O - H.T.C.
930  DATA .721546E-01,-.168154E-03,8.23371,13.6061
932  DATA 2.32106,-.952303E-01,7.11163,152.481
934  DATA 1.29675,.434556E-01,.814137,-92.922
936  DATA 1.01167,-.508808E-02,-1.64215,-.292726E-03
944  REM                    B,P
945  DATA .6074E-02,10
949  REM                    DATE & TIME
950  DATA 22,3,77,11,30,0
954  REM                    INITIAL CONDITIONS
955  DATA .573269E-01,0,17.4736
957  DATA .573269E-01,0,65.1495
959  DATA 0,.226856E-01,76.254
961  DATA .508486E-02,.176007E-01,74.2033
963  DATA .219599E-01,0,100
965  DATA 0,.219599E-01,114.927
967  DATA .346413E-01,.226856E-02,75.8355
969  DATA .346413E-01,0,75.4169
971  DATA 0,.761744E-02,62.0168
973  DATA .115818E-01,.111038E-01,71.2033
975  DATA .30303E-01,0,57.0866
976  DATA .436667,0,15.8766
979  DATA .436667,0,31.6085
981  DATA 1.55227,0,58.1713
983  DATA 1.55227,0,55.8925
985  DATA .270239E-01,0,55.8925
989  REM         H
990  DATA 1.12361
994  REM                    P6(12),P7(15)-PRESSURES
995  DATA 44.8479,29.6868
1099  REM
1100  REM                    TEMPERATURE TO PRESSURE SUBROUTINE
1101  REM                    ----------------------------------
1105  T8=1-373.15/(273.15+T9)
1110  P9=101.325*EXP(T8*(13.3185-T8*(1.967+T8*(6445+.1299*T8))))
1115  RETURN
1200  END
```

APPENDIX G

THE ON-LINE FILTERING PROGRAMS - FILTER2, 3 AND 4

CONTENTS

TABLE G-1.1 - THE INITIALISATION PROGRAM - INIT


```
C O.L.D.F.P. INITIALISATION PROGRAM - INIT
C ------------------------------------------
      HEADER IN
      COMMON/KALM/XE(7),XP(7),Y(7),PE(7,7),R(7,7),Q(7,7),THY(7,7),
     1 GAMMA(7,7),RK(7,14),RM(7,7),N,M,P,
     2 C(7,7),GAM(7),WW(7),F4(7,7),
     3 ALPHA,BETA,ITHETA,MF4,G,CC
      COMMON/MODEL/V1,V2,V3,V4,V5,V6,W1,W2,W3,W4,W5,W6,C1,C2,C3,C4
     1 A1,A2,A3,A4,A5,HA1,HA2,HA3,HA4,HB1,HB2,HB3,HB4,HC1,HC2,HC3,
     2 HC4,HD1,HD2,HD3,HD4,R1,BCNS,T(16),W(16),V(16),H,DP1,DP2,DP3
     3 P12,P15,U1,U2,U3,U4,R4,R7,R10
      COMMON/SCAN/INTR,STEP,NCYCLE,NCOUNT,IPRST,IPRNT,Z(18),D(18)
      EXTERNAL BUFD
      DATA IH1,IH2,IR1,IR2/2HH1,2HH2,$220,$200/
      DATA IIN,IHT/2HIN,2HHT/
 1000 FORMAT(32H OLDFP - TYPE 4 - INITIALISATION/1H ,31(1H=)//)
 1010 FORMAT(40H INTR,NCYCLE,IPRST,IPRNT,STEP - 4I4,F6.2)
 1020 FORMAT(/17H TEMPS.-4(4F7.3/))
 1030 FORMAT(/14H M1,M15-2F10.7)
 1040 FORMAT(/15H M11,M12-2F10.7)
 1050 FORMAT(/9H V6-F10.7)
 1060 FORMAT(/7H H-F7.4)
 1070 FORMAT(/14H P12,P15-2F8.4)
 1080 FORMAT(/9H PE-7F9.4)
 1090 FORMAT(/8H Q-7F9.4)
 1100 FORMAT(/8H R-6F8.6)
 1110 FORMAT(//25H **** READY TO START ****)
 1190 FORMAT(/1H ,4I4,F6.2)
 1200 FORMAT(/4(1H ,4F7.3/))
 1210 FORMAT(/1H ,2F10.7)
 1220 FORMAT(/1H ,F7.4)
 1230 FORMAT(/1H ,2F8.4)
 1240 FORMAT(/1H ,7F9.4)
 1250 FORMAT(/1H ,6F8.6)
 1260 FORMAT(/34H ALPHA,BETA,ITHETA,MF4 - 2F7.4,2I2)
 1270 FORMAT(/1H ,2F7.4,2I2)
 1280 FORMAT(/12H F4-2I3  NO.,I3)
 2000 FORMAT(4I4,F6.2)
 2010 FORMAT(4(4F7.3/))
 2020 FORMAT(2F10.7)
 2050 FORMAT(F7.4)
 2060 FORMAT(2F8.4)
 2070 FORMAT(7F9.4)
 2080 FORMAT(6F8.6)
 2090 FORMAT(2F7.4,2I2)
 2100 FORMAT(2I3)
      WRITE(1,1000)
C
C SET UP COMMON BLOCK SCAN
C
      WRITE(1,1010)
      READ(1,2000) INTR,NCYCLE,IPRST,IPRNT,STEP
      WRITE(1,1190) INTR,NCYCLE,IPRST,IPRNT,STEP
      NCOUNT=1
C
C STORE COUNTER PRESET VALUE
C
```

TABLE G-1.1 - continued


```
      CALL ISTORE(LOC(BUFD)+3,IPRST)
C
C STEADY STATE DATA
C
      WRITE(1,1020)
      READ(1,2010)(T(I),I=1,16)
      WRITE(1,1200) (T(I),I=1,16)
      WRITE(1,1030)
      READ(1,2020) W(1),W(15)
      WRITE(1,1210) W(1),W(15)
      WRITE(1,1040)
      READ(1,2020) W(11),W(12)
      WRITE(1,1210) W(11),W(12)
      WRITE(1,1050)
      READ(1,2020) V(6)
      WRITE(1,1210) V(6)
      WRITE(1,1060)
      READ(1,2050) H
      WRITE(1,1220) H
      WRITE(1,1070)
      READ(1,2060) P12,P15
      WRITE(1,1230) P12,P15
      W(2)=W(1)
      W(14)=W(15)
      W(13)=W(12)
      W(5)=V(6)
      DP1=0.
      DP2=0.
      DP3=0.
C
C SET UP COMMON BLOCK KALM
C
      P=FLOAT(INTR)
      N=7
      M=6
      DO 100 I=1,N
      DO 100 J=1,N
      PE(I,J)=0.
      R(I,J)=0.
      Q(I,J)=0.
      RM(I,J)=0.
      XP(I)=0.
  100 CONTINUE
      WRITE(1,1080)
      READ(1,2070)(PE(I,I),I=1,N)
      WRITE(1,1240) (PE(I,I),I=1,N)
      WRITE(1,1090)
      READ(1,2070)(Q(I,I),I=1,N)
      WRITE(1,1240) (Q(I,I),I=1,N)
      WRITE(1,1100)
      READ(1,2080)(R(I,I),I=1,M)
      WRITE(1,1250) (R(I,I),I=1,M)
      RM(1,1)=1.
      RM(2,2)=1.
      RM(3,3)=1.
      RM(4,5)=1.
      RM(5,6)=1.
```

TABLE G-1.1 - continued

```
C O.L.D.F.P. INITIALISATION PROGRAM - INIT          PAGE        3

        RM(6,7)=1.
        XE(1)=T(2)
        XE(2)=T(4)
        XE(3)=T(7)
        XE(4)=T(10)
        XE(5)=T(13)
        XE(6)=T(14)
        XE(7)=H
C
C SET UP ADAPTIVE SECTION OF COMMON/KALM/
C
        DO 160 I=1,N
        DO 150 J=1,N
        C(I,J)=0.
        F4(I,J)=0.
    150 CONTINUE
        C(I,I)=1.
        WW(I)=0.
        GAM(I)=0.
    160 CONTINUE
        G=0.
        CC=0.
        WRITE(1,1260)
        READ(1,2090) ALPHA,BETA,ITHETA,MF4
        WRITE(1,1270) ALPHA,BETA,ITHETA,MF4
        DO 170 I=1,MF4
        WRITE(1,1280) I
        READ(1,2100) II,JJ
        F4(II,JJ)=1.
    170 CONTINUE
C
C PREPARE TO START
C
        CALL ISTORE(:11373,IIN)
        =0
A     @SSM
        IDIGO=
        CALL ISTORE(LOC(BUFD)+1,IDIGO)
        REQUEST IH2[IR2],,300
        WAIT
    200 WRITE(1,1110)
    210 REQUEST IH1[IR1],,300
        WAIT
    220 IDIGI=IFETCH(LOC(BUFD))
        IF(IDIGI.LT.16384) GOTO 210
        IDEL=10
        CALL ISTORE(:11373,IHT)
        CONNECT CLOCK IHT[IDEL,INTR],2,300
        TERMINATE
C
C ERROR EXIT
C
    300 PRINT ERROR IIN[:3000]
        TERMINATE
        END
$0    @
ND OF JOB
```

TABLE G-2.1 - THE FILTERING PROGRAM - FILTER


```
C O.L.D.F.P MAIN PROGRAM - FILTER
C ---------------------------------
      HEADER HT
      COMMON/MODEL/V1,V2,V3,V4,V5,V6,W1,W2,W3,W4,W5,W6,C1,C2,C3,C4
     1 A1,A2,A3,A4,A5,HA1,HA2,HA3,HA4,HB1,HB2,HB3,HB4,HC1,HC2,HC3,
     2 HC4,HD1,HD2,HD3,HD4,R1,BCNS,T(16),W(16),V(16),H,DP1,DP2,DP3
     3 P12,P15,U1,U2,U3,U4,R4,R7,R10
      COMMON/SCAN/INTR,STEP,NCYCLE,NCOUNT,IPRST,IPRNT,C(18),D(18)
      COMMON/KALM/XE(7),XP(7),Y(7),PE(7,7),R(7,7),Q(7,7),THY(7,7),
     1 GAMMA(7,7),RK(7,14),RM(7,7),N,M,P,
     2 Z(7,7),GAM(7),WW(7),F4(7,7),
     3 ALPHA,BETA,ITHETA,MF4,G,CC
      DIMENSION A(30),IP(5)
      EXTERNAL BUFD
      DATA IH1,IH2,IH3,IH4,IH6,IHT,IPP/2HH1,2HH2,2HH3,2HH4,2HH6,2H
     1 2HPP/
      DATA IR1,IR2,IR3,IR4,IR5,IR6/$800,$100,$500,$200,$910,$300/
      DATA IP(1),IP(2),IP(3),IP(4),IP(5)/2HHT,$720,$900,54,A/
 1001 FORMAT(25H OLDFP - TYPE 4 - RUNNING/1H ,24(1H=)//)
 1010 FORMAT(1H ,7(F9.4,1X))
 1020 FORMAT(1H ,3(F9.4,1X),10X,3(F9.4,1X))
 1030 FORMAT(1H ,7(F9.5,1X))
 1040 FORMAT(32H **** HT DISCONNECTED FROM CLOCK)
 1050 FORMAT(/18H NUMBER OF CYCLES=,I4,4X,
     1     22H CHARACTERS PER CYCLE=,I4)
 1060 FORMAT(//9H SCAN NO.,1X,I4/5H TIME,1X,I2,1H.,I2,1H.,I2/)
 1070 FORMTA(/19H MODEL ERROR VECTOR//7F9.6//)
 1080 FORMAT(//3H G=,F10.6)
C
C DIGITAL OUTPUT
C
      =NCOUNT
A     @SSM
      IDIGO=
      CALL ISTORE(LOC(BUFD)+1,IDIGO)
      REQUEST IH2[IR2],,1000
      WAIT
C
C CONNECT COUNTER INTERRUPT
C
  100 REQUEST IH4[IR4],,1000
      WAIT
  200 CONNECT INTERRUPT H5[18],1000
      TIM1=FLOAT(IFETCH(LOC(BUFD)+4))+50.*(FLOAT(IFETCH(LOC(BUFD)
     1 +5))+60.*(FLOAT(IFETCH(LOC(BUFD)+6))+60.*FLOAT(IFETCH(
     2 LOC(BUFD)+7))))
C
C ANALOGUE INPUT
C
      REQUEST IH6[IR6],,1000
      WAIT
C
C READ TIME
C

  300 IHRS=IFETCH(LOC(BUFD)+7)
      IMIN=IFETCH(LOC(BUFD)+6)
```

TABLE G-2.1 - continued


```
      ISEC=IFETCH(LOC(BUFD)+5)
C
C ANALOGUE DATA PROCESSING
C
      DO 400 I=1,16
  400 A(I)=A(I)*D(I)+C(I)
C
C SET UP DYNAMIC MODEL INPUTS - T,W,V
C
      T(1)=A(3)
      T(3)=A(10)
      T(8)=A(7)
      T(11)=A(8)
      T(12)=A(1)
      T(15)=A(5)
      V(6)=A(11)
      W(5)=A(11)
      T4=1.-373.15/(273.15+A(9))
      AP4=101.325*EXP(T4*(13.3185-T4*(1.967+T4*(0.6645+0.1299*T4))
     1 ))
      IF(NCOUNT.EQ.1) P4=AP4
      DP1=(AP4-P4)/P
      DP2=(A(12)-P12)/P
      DP3=(A(15)-P15)/P
      P12=A(12)
      P15=A(15)
      P4=AP4
C
C GET COUNTER FREQUENCY
C
      TIM2=FLOAT(IFETCH(LOC(BUFD)+8))+50.*(FLOAT(IFETCH(LOC(BUFD)
     1 +9))+60.*(FLOAT(IFETCH(LOC(BUFD)+10))+60.*FLOAT(IFETCH(
     2 LOC(BUFD)+11))))
      DISCONNECT INTERRUPT 18
      IF(TIM1.LT.TIM2) GOTO 510
      REQUEST IH3[IR3],,1000
      WAIT
  500 TIM2=FLOAT(IFETCH(LOC(BUFD)+4))+50.*(FLOAT(IFETCH(LOC(BUFD)
     1 +5))+60.*(FLOAT(IFETCH(LOC(BUFD)+6))+60.*FLOAT(IFETCH(LOC
     2 (BUFD)+7))))
      IVAL=IFETCH(LOC(BUFD)+2)
      GOTO 520
  510 IVAL=127
  520 IF(ABS(TIM2-TIM1).LT.0.00001) GOTO 530
      A(18)=50.*FLOAT(IVAL-IPRST)/(TIM2-TIM1)
      GOTO 540
  530 A(18)=1000000.
  540 A(18)=A(18)*D(18)+C(18)
      W(1)=A(18)
      W(2)=A(18)
C
C SET UP MEASUREMENT VECTOR - Y
C
      Y(1)=A(4)
      Y(2)=A(9)
      Y(3)=(A(10)+A(7))/2.
      Y(4)=A(2)
```

```
      Y(5)=A(6)
      Y(6)=A(12)
C
C ESTIMATION
C
      IF(NCOUNT.EQ.1) GOTO 600
      ALPHA=1./FLOAT(NCOUNT)
      IF(ALPHA.LE.ALPAH1) ALPHA=ALPHA1
      CALL ADAPT
      GOTO 610
  600 ALPHA1=ALPHA
      DO 605 I=1,M
      RK(I,1)=Y(I)
  605 CONTINUE
C
C OUTPUT(ASR)
C
      WRITE(1,1001)
  610 WRITE(1,1060) NCOUNT,IHRS,IMIN,ISEC
      WRITE(1,1010) (XE(I),I=1,N)
      WRITE(1,1010) (XP(I),I=1,N)
      WRITE(1,1020) (Y(I),I=1,M)
      WRITE(1,1030) (PE(I,I),I=1,N)
C
C OUTPUT(HSP)
C
      IF(IPRNT.NE.1) GOTO 720
      DO 700 I=1,N
      A(I)=XE(I)
      IPN=I+N
      A(IPN)=XP(I)
      IP2N=I+2*N
      A(IP2N)=PE(I,I)
      IF(I.GT.M) GOTO 700
      IP3N=I+3*N
      A(IP3N)=Y(I)
  700 CONTINUE
      REQUEST IPP[IP],,1000
      WAIT
C
C PREDICTION
C
  720 CALL PREDIC
C
C STATE TRANSITION MATRIX
C
      CALL TRANS
C
C DIGITAL INPUT
C
      REQUEST IH1[IR1],,1000
      WAIT
  800 IDIGI=IFETCH(LOC(BUFD))
      IF(IDIGI.LT.0) GOTO 900
C
C TEST FOR END OF RUN
C
```

TABLE G-2.1 - continued

```
C O.L.D.F.P MAIN PROGRAM - FILTER                          PAGE        4

      NCOUNT=NCOUNT+1
      IF(NCOUNT.GT.NCYCLE) GOTO 900
      TERMINATE
C
C END OF RUN
C
  900 IDIGO=0
      CALL ISTORE(LOC(BUFD)+1,IDIGO)
      REQUEST IH2[IR5],,1000
      WAIT
  910 DISCONNECT CLOCK IHT[2],1011
      WRITE(1,1040)
      WRITE(1,1070) (WW(I),I=1,M)
      WRITE(1,1080) G
      WRITE(1,1050) NCOUNT,IP(4)
  920 TERMINATE
C
C ERROR EXIT
C
 1000 PRINT ERROR IHT[:3100]
      GOTO 900
 1011 PRINT ERROR IHT[:3110]
      GOTO 920
      END
$0   @
```

TABLE G-3.1 - SUBROUTINE KALMAN


```
      SUBROUTINE KALMAN
C
C SUBROUTINE KALMAN IS THE "EXTENDED" VERSION OF THE
C          RECURSIVE DIGITAL FILTER
C
      COMMON/KALM/XE(7),XP(7),Y(7),PE(7,7),R(7,7),Q(7,7),THY(7,7)
     1 ,GAMMA(7,7),RK(7,7),RM(7,7),N,NM,P
      DIMENSION B(7,7),B1(7,14)
      REAL I1,II
      LOGICAL RR
 1000 FORMAT(///28H **** INVERSION ERROR KALMAN)
      I=1
      II=-1.
      I1=1.
C
C ERROR COVARIANCE MATRIX FOR PREDICTIONS
C
      CALL MATMUL(THY,PE,B,N,N,N)
      CALL MATRAP(THY,B1,N,N)
      CALL MATMUL(B,B1,PE,N,N,N)
      CALL MATMUL(GAMMA,Q,B,N,N,N)
      CALL MATRAP(GAMMA,B1,N,N)
      CALL MATMUL(B,B1,RK,N,N,N)
      CALL MATADD(PE,RK,PE,N,N,I1,I1)
C
C FILTER GAIN
C
      CALL MATMUL(RM,PE,B1,NM,N,N)
      CALL MATRAP(RM,THY,NM,N)
      CALL MATMUL(B1,THY,B,NM,N,NM)
      CALL MATADD(B,R,B,NM,NM,I1,I1)
      CALL MATINV(B,B1,NM,RR,DET)
      IF(RR) GOTO 270
      CALL MATMUL(PE,THY,B,N,N,NM)
      CALL MATMUL(B,B1,RK,N,NM,NM)
C
C ESTIMATION
C
      CALL MATMUL(RM,XP,B,NM,N,I)
      CALL MATADD(B,Y,B,NM,I,II,I1)
      CALL MATMUL(RK,B,XE,N,NM,I)
      CALL MATADD(XE,XP,XE,N,I,I1,I1)
      DO 20 I2=1,N
      DO 10 J=1,N
   10 B(J,I2)=0.
   20 B(I2,I2)=1.
C
C ERROR COVARIANCE MATRIX FOR ESTIMATES
C
      CALL MATMUL(RK,RM,B1,N,NM,N)
      CALL MATADD(B1,B,B,N,N,II,I1)
      CALL MATMUL(B,PE,THY,N,N,N)
      CALL MATRAP(B,B1,N,N)
      CALL MATMUL(THY,B1,B,N,N,N)
      CALL MATMUL(RK,R,THY,N,NM,NM)
      CALL MATRAP(RK,B1,N,NM)
      CALL MATMUL(THY,B1,PE,N,NM,N)
```

TABLE G-3.1 - continued

```
      SUBROUTINE KALMAN                                    PAGE      2

      CALL MATADD(B,PE,PE,N,N,I1,I1)
      RETURN
  270 WRITE(1,1000)
      TERMINATE
      END
$0    @
ND OF JOB
```

TABLE G-4.1 - SUBROUTINE ADAPT


```
      SUBROUTINE ADAPT
C
C SUBROUTINE ADAPT IS AN ADAPTIVE FORM OF THE EXTENDED
C    KALMAN FILTER DEVELOPED BY R.NEWMAN AT ASTON UNIVERSITY.
C
      COMMON/KALM/XE(7),XP(7),Y(7),P(7,7),R(7,7),Q(7,7),THY(7,7),
     1 GAMMA(7,7),RK(7,14),RM(7,7),N,M,RINTR,
     2 C(7,7),GAM(7),W(7),F4(7,7),
     3 ALPHA,BETA,ITHETA,MF4,G,CC
      COMMON/SCAN/INTR,STEP,NCYCLE,NCOUNT,IPRST,IPRNT,Z(18),D(18)
      DIMENSION A(7,7),B(7,14),Y1(7),W1(7)
      LOGICAL RR
 1000 FORMAT(////31H **** INVERSION ERROR - ADAPT-W//1H ,
     1 6(6(F8.4,1X)/))
 1010 FORMAT(////31H **** INVERSION ERROR - ADAPT-K)
C
C CALCULATE INITIAL VALUE OF Y-RM*XP,Y1
C
      RONE=1.
      IONE=1
      RMONE=-1.
      RHLF=0.5
      RMHLF=-0.5
      IF(NCOUNT.GT.2) GOTO 100
      CALL MATMUL(RM,XE,A,M,N,IONE)
      CALL MATADD(A,RK,Y1,M,IONE,RMHLF,RHLF)
      BETA1=1.-BETA
C
C CALCULATION OF THE COVARIANCE MATRIX OF THE PREDICTIONS,P
C
  100 CALL MATMUL(THY,P,RK,N,N,N)
      CALL MATRAP(THY,A,N,N)
      CALL MATMUL(RK,A,P,N,N,N)
      CALL MATMUL(GAMMA,Q,RK,N,N,N)
      CALL MATRAP(GAMMA,B,N,N)
      CALL MATMUL(RK,B,A,N,N,N)
      CALL MATADD(P,A,P,N,N,RONE,RONE)
      CALL MATMUL(THY,C,B,N,N,N)
      CALL MATADD(P,B,P,N,N,RONE,RONE)
      CALL MATRAP(C,B,N,N)
      CALL MATMUL(B,A,RK,N,N,N)
      CALL MATADD(P,RK,P,N,N,RONE,RONE)
      CALL MATMUL(GAMMA,F4,A,N,N,M)
      CALL MATRAP(A,RK,N,M)
      CALL MATMUL(A,RK,B,N,M,N)
      DO 150 I=1,N
      DO 150 J=1,N
  150 B(I,J)=B(I,J)*CC
      CALL MATADD(P,B,P,N,N,RONE,RONE)
C
C CALCULATION OF THE DISTURBANCE MATRIX OF THE PREDICTIONS,C
C
      CALL MATMUL(THY,C,A,N,N,N)
      CALL MATADD(A,B,C,N,N,RONE,RONE)
C
C CALCULATION OF THE RESIDUALS VECTOR,GAMMA,
C    & THE ESTIMATE OF THE TRACE OF THE COVARIANCE MATRIX
```

TABLE G-4.1 - continued

```
      SUBROUTINE ADAPT                                      PAGE          2

C     OF THE RESIDUALS,G
C
      CALL MATMUL(RM,XP,A,M,N,IONE)
      CALL MATADD(Y,A,A,M,IONE,RHLF,RMHLF)
      CALL MATRAP(A,B,M,IONE)
      CALL MATMUL(B,A,RK,IONE,M,IONE)
      G1=RK(IONE,IONE)
      CALL MATADD(Y1,A,B,M,IONE,RONE,RONE)
      CALL MATADD(B,GAM,GAM,M,IONE,BETA,BETA1)
      CALL MATRAP(Y1,B,M,IONE)
      CALL MATMUL(B,Y1,RK,IONE,M,IONE)
      G1=G1+RK(IONE,IONE)
      G=G+ALPHA*(G1-G)
      DO 250 I=1,M
  250 Y1(I)=A(I,IONE)
C
C CALCULATION OF MODEL ERRORS VECTOR,W
C
      CALL MATMUL(RM,GAMMA,A,M,N,N)
      CALL MATMUL(A,F4,B,M,N,M)
      CALL MATRAP(B,A,M,M)
      CALL MATMUL(B,A,RK,M,M,M)
      TR1=0.
      DO 200 I=1,M
  200 TR1=TR1+RK(I,I)
      CALL MATMUL(A,B,RK,M,M,M)
      CALL MATINV(RK,B,MF4,RR,DET)
      IF(RR) GOTO 270
      CALL MATMUL(B,A,RK,MF4,MF4,M)
      CALL MATMUL(RK,GAM,W1,MF4,M,IONE)
  203 IF((NCOUNT/ITHETA)*ITHETA.NE.NCOUNT) GO TO 205
      CALL MATADD(W,W1,W,M,IONE,RONE,RONE)
C
C CALCULATION OF THE COMMON DIAGONAL ELEMENT OF THE COVARIANCE
C  MATRIX OF THE MODEL ERRORS,CC
C
  205 CALL MATMUL(RM,P,A,M,N,N)
      CALL MATRAP(RM,B,M,N)
      CALL MATMUL(A,B,RK,M,N,M)
      CALL MATADD(RK,R,A,M,M,RONE,RONE)
      TR2=0.
      DO 210 I=1,M
  210 TR2=TR2+A(I,I)
      CC1=(G-TR2)/TR1
      CC=CC+CC1
      IF(CC.LT.0.) CC=0.
C
C CALCULATION OF FILTER GAIN,RK
C
      CALL MATINV(A,RK,M,RR,DET)
      IF(RR) GOTO 280
      CALL MATMUL(B,RK,A,N,M,M)
      CALL MATMUL(P,A,RK,N,N,M)
C
C CALCULATION OF ESTIMATED STATE VECTOR,XE
C
      CALL MATMUL(RM,XP,A,M,N,IONE)
```

TABLE G-4.1 - continued

```
      SUBROUTINE ADAPT                                          PAGE      3

      CALL MATADD(Y,A,A,M,IONE,RONE,RMONE)
      CALL MATMUL(RK,A,B,N,M,IONE)
      CALL MATADD(XP,B,XE,N,IONE,RONE,RONE)
C
C CALCULATION OF THE COVARIANCE MATRIX OF THE ESTIMATES,P,
C   & THE DISTURBANCE MATRIX OF THE ESTIMATES,C
C
      CALL MATMUL(RK,RM,A,N,M,N)
      DO 240 I=1,N
      DO 230 J=1,N
  230 A(I,J)=-A(I,J)
  240 A(I,I)=A(I,I)+1.
      CALL MATRAP(A,B,N,N)
      CALL MATMUL(A,P,THY,N,N,N)
      CALL MATMUL(THY,B,P,N,N,N)
      CALL MATMUL(A,C,THY,N,N,N)
      CALL MATMUL(THY,B,C,N,N,N)
      CALL MATMUL(RK,R,A,N,M,M)
      CALL MATRAP(RK,B,N,M)
      CALL MATMUL(A,B,THY,N,M,N)
      CALL MATADD(P,THY,P,N,N,RONE,RONE)
      CALL MATADD(C,THY,C,N,N,RONE,RONE)
C
C NORMAL & ERROR EXITS
C
      RETURN
  270 WRITE(1,1000) ((RK(I,J),J=1,MF4),I=1,MF4)
      DO 275 I=1,M
  275 W(I)=0.
      GOTO 203
  280 WRITE(1,1010)
      TERMINATE
      END
$0    @
```

TABLE G-5.1 - SUBROUTINE PREDIC


```
      SUBROUTINE PREDIC
C
C CALCULATION OF PREDICTED STATE BY RK4 INTEGRATION
C
      COMMON/KALM/XE(7),XP(7),Y(7),PE(7,7),R(7,7),Q(7,7),THY(7,7),
     1 GAMMA(7,7),RK(7,14),RM(7,7),N,M,P,
     2 Z(7,7),GAM(7),WW(7),F4(7,7),
     3 ALPHA,BETA,ITHETA,MF4,G,CC
      COMMON/MODEL/V1,V2,V3,V4,V5,V6,W1,W2,W3,W4,W5,W6,C1,C2,C3,C4
     1 A1,A2,A3,A4,A5,HA1,HA2,HA3,HA4,HB1,HB2,HB3,HB4,HC1,HC2,HC3,
     2 HC4,HD1,HD2,HD3,HD4,R1,BCNS,T(16),W(16),V(16),H,DP1,DP2,DP3
     3 P12,P15,U1,U2,U3,U4,R4,R7,R10
      COMMON/SCAN/INTR,STEP,NCYCLE,NCOUNT,IPRST,IPRNT,C(18),D(18)
      DIMENSION DX(7),XA(7),DXA(7),E(7),A(7)
      REAL L
      LOGICAL FLAG
      DATA RVAP/0.112/
      L(T2)=-2.4068*T2+2501.6
      RV(T3)=EXP(1.93*ALOG(T3)-3.1487)/1000.
 1000 FORMAT(//7H NSTEP=,I5)
 1030 FORMAT(//5H XP2=,F10.5)
 1040 FORMAT(//5H XP3=,F10.5)
 1050 FORMAT(//5H XP4=,F10.5)
C
C INITIALISATION
C
      JS4=0
      NSTEP1=0
      FLAG=.FALSE.
      NSTEP=INT(FLOAT(INTR)/STEP+0.5)
      WRITE(1,1000) NSTEP
      T1=0.
      DO 10 I=1,N
      XP(I)=XE(I)
      XA(I)=0.
      DXA(I)=0.
   10 CONTINUE
      IONE=1
      RONE=1.
   20 JS4=JS4+1
      IF(JS4.EQ.5) GOTO 400
C
C CALCULATE HEAT TRANSFER COEFFICIENTS
C
      U1=HA1+HB1*(XP(2)-(T(1)+XP(1))/2.)+HC1*W(1)+HD1*V(3)
      U2=HA2+HB2*(T(5)-(XP(1)+XP(3))/2.)+HC2*W(2)+HD2*V(6)
      U3=HA3+HB3*(XP(4)-(XP(6)+T(15))/2.)+HC3*W(15)+HD3*V(4)
      U4=HA4+HB4*(T(11)-(T(12)+XP(5))/2.)+HC4*W(12)+HD4*(V(10)
     1 +V(9))
C
C CALCULATE VAPOUR DENSITIES
C
      IF(XP(2).LE.0.) GOTO 35
      R4=RV(XP(2))
      GOTO 40
   35 R4=RVAP
      WRITE(1,1030) XP(2)
```

TABLE G-5.1 - continued


```
   40 IF(XP(3).LE.0.) GOTO 45
      R7=RV(XP(3))
      GOTO 50
   45 R7=RVAP
      WRITE(1,1040) XP(4)
   50 IF(XP(4).LE.0.) GOTO 55
      R10=RV(XP(4))
      GOTO 60
   55 R10=RVAP
      WRITE(1,1050) XP(4)
C
C STEADY STATE EQUATIONS
C
   60 V(7)=(V(6)*L(T(5))-W(2)*C1*(XP(3)-XP(1)))/L(XP(3))
      V(3)=V(7)
      W(7)=W(2)-V(7)-DP2*V3*BCNS
      W(8)=W(7)
      W(4)=(W(1)*C1*(XP(1)-T(1)))/L(XP(2))
      V(4)=V(3)-W(4)-DP1*V2*BCNS
      V(9)=(W(14)*C1*(XP(6)-T(15))+W(8)*C1*(T(8)-T(15)))/L(T(15))
      W(10)=(W(14)*C1*(XP(6)-T(15)))/L(XP(4))+W(4)
      V(10)=W(4)+V(4)-W(10)-DP3*V5*BCNS
      W(11)=V(9)+W(10)+V(10)
C
C DIFFERENTIAL EQUATIONS
C
      DX(1)=(W(1)*C1*(T(1)-XP(1))+U1*A1*(XP(2)-(XP(1)+T(1))
     1 /2.))/(V1*R1*C1+W1*C2)
      DX(2)=(V(3)*C1*(T(3)-XP(2))+V(3)*(L(T(3))-L(XP(2)))+
     1 W(4)*L(XP(2))-U1*A1*(XP(2)-(XP(1)+T(1))/2.))/
     2 (W2*C3+V2*R4*C4)
      DX(3)=(W(2)*C1*(XP(2)-XP(3))+L(XP(3))*(W(8)-W(2))+U2*A2
     1 *(T(5)-(XP(3)+XP(2))/2.))/(W3*C2+V3*R7*C4)
      DX(4)=(W(4)*C1*(XP(2)-XP(4))+V(4)*C1*(XP(2)-XP(4))+V(4)*
     1 (L(XP(2))-L(XP(4)))+L(XP(4))*(W(10)-W(4))-U3*A3*(XP(4)-
     2 (XP(6)+T(15))/2.))/(W5*C3+V5*R10*C4)
      DX(5)=(W(12)*C1*(T(12)-XP(5))+U4*A4*(T(11)-(XP(5)+T(12))
     1 /2.))/(V6*R1*C1+W6*C2)
      DX(6)=(W(14)*C1*(T(15)-XP(6))+U3*A3*(XP(4)-(XP(6)+T(15))
     1 /2.))/(V4*R1*C1+W4*C2)
      DX(7)=(W(8)-V(9))/(A5*R1)
C
C CALCULATE LOSS TERMS
C
      IF(NCOUNT.NE.1.OR.FLAG) GOTO 110
      DO 100 I=1,N
      E(I)=DX(I)
  100 CONTINUE
      FLAG=.TRUE.
  110 DO 120 I=1,N
      DX(I)=DX(I)-E(I)
  120 CONTINUE
C
C INTEGRATE INDEPENDENT VARIABLE
C
      IF(JS4.EQ.1) GOTO 202
      IF(JS4.EQ.3) GOTO 204
```

TABLE G-5.1 - continued

```
        SUBROUTINE PREDIC                                         PAGE      3

        GOTO 300
    202 DT=STEP/2.
        T1=T1+DT
        GOTO 300
    204 T1=T1+DT
C
C INTEGRATE DEPENDENT VARIABLES
C
    300 CALL MATMUL(F4,WW,A,N,M,IONE)
        CALL MATADD(DX,A,DX,N,IONE,RONE,RONE)
        DO 310 I=1,N
        GOTO(304,305,306,308),JS4
    304 XA(I)=XP(I)
        DXA(I)=DX(I)
        XP(I)=XP(I)+DX(I)*DT
        GOTO 310
    305 DXA(I)=DXA(I)+2.*DX(I)
        GOTO 307
    306 DXA(I)=DXA(I)+2.*DX(I)
    307 XP(I)=XA(I)+DX(I)*DT
        GOTO 310
    308 DXA(I)=(DXA(I)+DX(I))/6.
        XP(I)=XA(I)+DXA(I)*STEP
    310 CONTINUE
        GOTO 20
C
C TEST FOR END OF PREDICTION INTERVAL
C
    400 NSTEP1=NSTEP1+1
        IF(NSTEP1.EQ.NSTEP) RETURN
        JS4=0
        GOTO 20
        END
$0    @
D OF JOB
```

TABLE G-6.1 - SUBROUTINE TRANS(1)

```
      SUBROUTINE TRANS                                    PAGE       1

      SUBROUTINE TRANS
C
C CALCULATION OF THE STATE TRANSITION MATRIX USING
C                A TRUNCATED TAYLOR SERIES
C
      COMMON/KALM/XE(7),XP(7),Y(7),PE(7,7),R(7,7),Q(7,7),THY(7,7),
     1 GAMMA(7,7),RK(7,14),RM(7,7),N,M,P,
     2 C(7,7),GAM(7),WW(7),F4(7,7),
     3 ALPHA,BETA,ITHETA,MF4,G,CC
      COMMON/MODEL/V1,V2,V3,V4,V5,V6,W1,W2,W3,W4,W5,W6,C1,C2,C3,C4
     1 A1,A2,A3,A4,A5,HA1,HA2,HA3,HA4,HB1,HB2,HB3,HB4,HC1,HC2,HC3,
     2 HC4,HD1,HD2,HD3,HD4,R1,BCNS,T(16),W(16),V(16),H,DP1,DP2,DP3
     3 P12,P15,U1,U2,U3,U4,R4,R7,R10
      REAL L
C
C INITIALISATION
C
      L=-2.4068
      DO 20 I=1,N
      DO 10 J=1,N
      THY(I,J)=0.
      GAMMA(I,J)=0.
   10 CONTINUE
      GAMMA(I,I)=1.
   20 CONTINUE
C
C CALCULATE DIVISORS
C
      DIV1=V1*R1*C1+W1*C2
      DIV2=V2*R4*C4+W2*C3
      DIV3=V3*R7*C4+W3*C2
      DIV4=V5*R10*C4+W5*C3
      DIV5=V6*R1*C1+W6*C2
      DIV6=V4*R1*C1+W4*C2
C
C CALCULATE NON-ZERO ELEMENTS OF THY(I,J)
C
      THY(1,1)=((-U1*A1/2.-W(2)*C1)/DIV1)*P+1.
      THY(1,2)=((U1*A1)/DIV1)*P
      THY(2,1)=((U1*A1/2.)/DIV2)*P
      THY(2,2)=((-V(3)*(C1+L)+W(4)*L-U1*A1)/DIV2)*P+1.
      THY(2,3)=((V(3)*(C1+L))/DIV2)*P
      THY(3,1)=((W(2)*C1+U2*A2/2.)/DIV3)*P
      THY(3,3)=((-W(2)*C1+L*(W(8)-W(2))-U2*A2/2.)/DIV3)*P+1.
      THY(4,2)=((W(4)*C1+V(4)*(C1+L))/DIV4)*P
      THY(4,4)=((-W(4)*C1-V(4)*(C1+L)+L*(W(10)-W(4))-U3*A3)/DIV4)
     1 *P+1.
      THY(4,6)=((U3*A3/2.)/DIV4)*P
      THY(5,5)=((-W(12)*C1-U4*A4/2.)/DIV5)*P+1.
      THY(6,4)=((U3*A3)/DIV6)*P
      THY(6,6)=((-W(14)*C1-U3*A3/2.)/DIV6)*P+1.
      THY(7,7)=1.
      RETURN
      END
 $0   @
ND OF JOB
```

TABLE G-7.1 - SUBROUTINE TRANS(2)


```
      SUBROUTINE TRANS
C
C SUBROUTINE TRANS(2) CALCULATES THE STATE TRANSITION MATRIX
C         ,THY, USING A CANONICAL TRANSFORMATION
C
      COMMON/KALM/XE(7),XP(7),Y(7),PE(7,7),Q(7,7),R(7,7),THY(7,7)
     1 ,GAMMA(7,7),RK(7,7),RM(7,7),N,M,P
      DIMENSION B(7,7),C(7,7),D(7,7),XPT(7),XET(7)
C
C INITIALISATION
C
      IONE=1
C
C TRANSFORMATION OF ESTIMATES
C
      CALL RUTIS(C,D,N)
      CALL MATMUL(D,XE,XET,N,N,IONE)
C
C TRANSFORMATION OF PREDICTIONS
C
      CALL MATMUL(D,XP,XPT,N,N,IONE)
C
C CALCULATION OF TRANSFORMED STATE TRANSITION MATRIX
C
      DO 1ØØ K=1,N
      DO 11Ø J=1,N
  11Ø B(K,J)=Ø.
      B(K,K)=XPT(K)/XET(K)
  1ØØ CONTINUE
C
C RETRANSFORM STATE TRANSITION MATRIX
C
      CALL MATMUL(C,B,RK,N,N,N)
      CALL MATMUL(RK,D,THY,N,N,N)
C
C CALCULATE INTEGRAL STATE TRANSITION MATRIX
C
      DO 115 K=1,N
  115 B(K,K)=B(K,K)*P
C
C RETRANSFORM INTEGRAL STATE TRANSITION MATRIX
C
      CALL MATMUL(C,B,RK,N,N,N)
      CALL MATMUL(RK,D,GAMMA,N,N,N)
      RETURN
      END
$Ø   @
ID OF JOB
```

TABLE G-8.1 - SUBROUTINE RUTIS


```
      SUBROUTINE RUTIS(C,D,N)
C
C SUBROUTINE RUTIS SETS UP THE EIGENVECTOR MATRIX
C         & THE INVERSE EIGENVECTOR MATRIX
C
      DIMENSION C(7,7),D(7,7)
C
C INITIALISATION
C
      DO 10 I=1,N
      DO 10 J=1,N
      C(I,J)=0.
      D(I,J)=0.
   10 CONTINUE
C
C SET UP EIGENVECTOR MATRIX,C
C
      C(1,1)=0.809056
      C(1,2)=0.752564
      C(1,3)=0.33976E-01
      C(2,1)=-0.516571
      C(2,2)=0.425911
      C(2,3)=0.650505E-01
      C(3,1)=-0.280129
      C(3,2)=-0.498526
      C(3,3)=0.997303
      C(4,1)=0.102106E-01
      C(4,2)=-0.568681E-01
      C(4,3)=0.122877E-02
      C(4,4)=0.428303
      C(4,6)=0.959781
      C(5,5)=1.
      C(6,1)=-0.246828E-01
      C(6,2)=-0.220257E-01
      C(6,3)=0.121576E-03
      C(6,4)=-0.903635
      C(6,6)=0.280752
      C(7,7)=1.
C
C SET UP INVERSE EIGENVECTOR MATRIX,D
C
      D(1,1)=0.604202
      D(1,2)=-1.01425
      D(1,3)=0.455721E-02
      D(2,1)=0.656751
      D(2,2)=1.0789
      D(2,3)=-0.927469E-01
      D(3,1)=0.498005
      D(3,2)=0.254425
      D(3,3)=0.969143
      D(4,1)=-0.675935E-02
      D(4,2)=-0.334233E-03
      D(4,3)=0.20273E-04
      D(4,4)=0.284296
      D(4,6)=-0.971892
      D(5,5)=1.
      D(6,1)=0.348642E-01
```

TABLE G-8.1 - continued

```
      SUBROUTINE RUTIS(C,D,N)                               PAGE        2

      D(6,2)=0.745395E-01
      D(6,3)=-0.722997E-02
      D(6,4)=0.915038
      D(6,6)=0.433708
      D(7,7)=1.
      RETURN
      END
 $0   @
D OF JOB
```

TABLE G-9.1 - MATRIX MANIPULATION ROUTINES


```
C OLDFP MATRIX LIBRARY
C ---------------------
      SUBROUTINE MATINV(A,B,N,R,DET)
C
C MATRIX INVERSION BY THE METHOD OF GAUSS-JORDAN
C
      DIMENSION A(7,N),B(7,14)
      LOGICAL R
      REAL MAX
C
C INITIALISATION
C
      DET=1.
      R=.FALSE.
      NRINT=0
      N1=2*N
      DO 1 I=1,N
      DO 2 J=1,N1
      IF(J.GT.N) GOTO 4
      B(I,J)=A(I,J)
      GOTO 2
    4 B(I,J)=0.
    2 CONTINUE
      K=I+N
      B(I,K)=1.
    1 CONTINUE
C
C START ELIMINATION
C
      DO 10 I=1,N
      II=I+1
C
C TEST FOR SINGULARITY
C
      DO 30 J=I,N
      IF(ABS(B(J,I)).GE.0.001) GOTO 15
   30 CONTINUE
      R=.TRUE.
      RETURN
C
C DETERMINE PIVOTAL ROW
C
   15 MAX=B(I,I)
      IPIVOT=I
      IF(II.GT.N) GOTO 20
      DO 11 J=II,N
      IF(ABS(B(J,I)).LT.ABS(MAX)) GOTO 11
      IPIVOT=J
      MAX=B(J,I)
   11 CONTINUE
      IF(IPIVOT.EQ.I) GOTO 20
C
C INTERCHANGE ITH ROW & PIVOTAL ROW
C
      NRINT=NRINT+1
      DO 17 K1=I,N1
      Q=B(I,K1)
```

TABLE G-9.1 - continued


```
      B(I,K1)=B(IPIVOT,K1)
      B(IPIVOT,K1)=Q
   17 CONTINUE
C
C ELIMINATION OF COEFFICIENTS FROM ITH ROW
C
   20 DET=DET*B(I,I)
      DO 40 J=I,N1
      J1=N1+I-J
      B(I,J1)=B(I,J1)/B(I,I)
      DO 50 K1=1,N
      IF(K1.EQ.I)GOTO 50
      B(K1,J1)=B(K1,J1)-B(I,J1)*B(K1,I)
   50 CONTINUE
   40 CONTINUE
   10 CONTINUE
C
C STORE INVERSE IN FIRST HALF OF
C   THE AUGMENTED MATRIX
C
      DO 60 J=1,N
      DO 60 I=1,N
      II=I+N
      B(J,I)=B(J,II)
   60 CONTINUE
C
C CORRECT SIGN OF DETERMINANT FOR ROW INTERCHANGES
C
      IF(NRINT.EQ.0) RETURN
      DET=DET*((-1.)**NRINT)
   70 RETURN
      END
      SUBROUTINE MATMUL(A,B,C,L,N,M)
      DIMENSION A(7,N),B(7,M),C(7,M)
      DO 20 J=1,L
      DO 20 I=1,M
      C(J,I)=0.
      DO 20 K=1,N
      C(J,I)=C(J,I)+A(J,K)*B(K,I)
   20 CONTINUE
      RETURN
      END
      SUBROUTINE MATADD(A,B,C,N,M,P,Q)
      DIMENSION A(7,M),B(7,M),C(7,M)
      DO 20 I=1,N
      DO 20 J=1,M
      C(I,J)=P*A(I,J)+Q*B(I,J)
   20 CONTINUE
      RETURN
      END
      SUBROUTINE MATRAP(A,B,N,M)
      DIMENSION A(7,M),B(7,N)
      DO 10 I=1,N
      DO 10 J=1,M
   10 B(J,I)=A(I,J)
      RETURN
      END
```

TABLE G-10.1 - BLOCK DATA SUBPROGRAM


```
      BLOCK DATA
      COMMON/SCAN/INTR,STEP,NCYCLE,NCOUNT,IPRST,IPRNT,C(18),D(18)
      COMMON/MODEL/V1,V2,V3,V4,V5,V6,W1,W2,W3,W4,W5,W6,C1,C2,C3,C4
     1 A1,A2,A3,A4,A5,HA1,HA2,HA3,HA4,HB1,HB2,HB3,HB4,HC1,HC2,HC3,
     2 HC4,HD1,HD2,HD3,HD4,R1,BCNS,T(16),W(16),V(16),H,DP1,DP2,DP3
     3 P12,P15,U1,U2,U3,U4,R4,R7,R10
      DATA C(1),C(2),C(3),C(4),C(5),C(6),C(7),C(8),C(9),C(10),C(11
     1 C(12),C(13),C(14),C(15),C(16),C(17),C(18)
     2       /-1.2576,-1.2338,-1.3624,-1.2767,-1.2433,-1.1433,-1.25
     3 -1.2243,-1.2719,-1.1481,0.914674E-02,-0.146455,-2.68874,
     4 -4.02142,98.174,-9.29804,-0.293712E-02,0.25E-03/
      DATA D(1),D(2),D(3),D(4),D(5),D(6),D(7),D(8),D(9),D(10),D(11
     1 D(12),D(13),D(14),D(15),D(16),D(17),D(18)
     2       /10*0.1190476,0.637724E-04,0.28159E-02,0.222261,0.2164
     3 0.137993,0.29318,0.472866E-04,0.695E-04/
      DATA V1,V2,V3,V4,V5,V6/0.1593E-02,0.7779E-02,0.2918E-02,
     1 0.3947E-02,0.151E-01,0.6613E-02/
      DATA W1,W2,W3,W4,W5,W6/4.91,22.304,8.192,8.922,13.452,14.95/
      DATA C1,C2,C3,C4/4.1868,0.3849,0.4519,1.9/
      DATA A1,A2,A3,A4,A5/0.3955,0.7238,0.672,1.1834,0.1307/
      DATA HA1,HA2,HA3,HA4/0.721546E-01,2.32106,1.29675,1.01167/
      DATA HB1,HB2,HB3,HB4/-0.168154E-03,-0.952303E-01,0.434556E-0
     1 -0.508808E-02/
      DATA HC1,HC2,HC3,HC4/8.23371,7.11163,0.814137,-1.64215/
      DATA HD1,HD2,HD3,HD4/13.6061,152.481,-92.922,-0.292726E-03/
      DATA R1,BCNS/985.22,0.6074E-02/
      END
 $0   @
ND OF JOB
```

TABLE G-11.1 - PROGRAM LINKS BETWEEN THE ON-LINE FILTERING PROGRAMS AND THE OLDFP EXECUTIVE


```
                           * O.L.D.F.P. - LINKS - FTRN.
                           *
                           *
                                SUBR  H5
                                SUBR  BUFD
                                SUBR  HT
                                SUBR  IN
            011125         H5   EQU   '11125
            011156         BUFD EQU   '11156
            016406         HT   EQU   '16406
            020702         IN   EQU   '20702
                                END

BUFD   011156A   H5     011125A   HT     016406A   IN     020702A

0000 WARNING OR ERROR FLAGS
DAP-16 MOD 2    REV. C    01-26-71
```

G-12 LOADING PROCEDURE FOR FILTER2

The following loading procedure was used to construct an SLST of the on-line filtering program, FILTER2.

(i) Load an SLST of LDR-APM Revision E, entry point 7000_8

(ii) Load an object tape of FILTER,

$P = 7000_8$ $A = 16400_8$ $B = 100_8$

(iii) Load an object tape of BLOCK DATA,

Press start.

(iv) Force load an object tape of INIT,

$P = 7004_8$

(v) Load an object tape of SUBROUTINE KALMAN,

Press start.

(vi) Load an object tape of SUBROUTINE PREDIC,

Press start.

(vii) Load an object tape of SUBROUTINE TRANS(1),

Press start.

(viii) Load an object tape of the Matrix Manipulation routines,

Press start.

(ix) Load Part 1 of the OP-16 Fortran Libraries tape (see Appendix G-15),

Press start.

(x) Load Part 2 of the OP-16 Fortran Libraries tape (see Appendix G-15),

Press start.

(xi) Load an object tape of the Fortran Links program - see Appendix G-11,

(xii) Obtain a memory map of the load - See Table G-12.1,

P = 7002

(xiii) Make the following correction to subroutine F$ER,

location, F$ER + 11_8(32016_8) - contents, NOP

(xiv) Load an SLST of PALAP (entry point 15000_8) and punch out a 2 part SLST of FILTER2,

locations 100_8 to 477_8 and 16400_8 to 37577_8

TABLE G-12.1 - MEMORY MAP OF FILTER2

*LOW	00000	HT	16406
*START	16400	IN	20532
*HIGH	32104	KALMAN	22266
*NAMES	02252	PREDIC	23430
*COMN	36014	TRANS	26026
COF2	00000	MATINV	26762
COF1	00000	MATMUL	27730
CIF2	00000	MATADD	30074
CIF1	00000	MATRAP	30212
MTI2	00000	A$22	30304
MTI1	00000	S$22	30306
LPI2	00000	M$22	30310
LPI1	00000	D$22	30312
ASF2	00000	F$AT	30414
ASF1	00000	F$R1	30500
ROCB	00000	F$W1	30506
ROCA	00000	F$L3	30514
*BASE	00431	F$L1	30514
XSTR	01000	F$L5	30520
XLNK	01001	F$L6	30520
XPFP	01002	F$L2	30520
XMIL	01003	F$CB	30524
XSEC	01004	CON	30537
XMIN	01005	TAG	30564
XHR	01006	EXP	30610
XDAY	01007	ALOG10	31150
XMSD	01010	ALOGX	31160
XPIC	01011	ALOG	31160
XJBS	01012	ABS	31446
XPLP	01013	FLOAT	31460
XPSP	01014	IFIX	31470
XCCW	01015	IDINT	31470
XEPP	01016	INT	31470
EROR	01016	E$21	31500
XLOD	01017	M$11X	31560
XPEP	01020	M$11	31560
XSYS	01021	C$12	31640
XICF	01022	C$21	31672
XSSA	01023	N$22	31724
XHLT	01024	REAL	31736
XPLT	01025	L$22	31736
XPCT	01026	H$22	31746
XCUT	01027	ARG$	31764
XIDT	01030	ADDR	32006
XID1	01031	F$ER	32016
XID2	01032	L$33	32034
XIVT	01033	ISTORE	32050
XINT	01034	IFETCH	32062
XDCT	01035	LOC	32072
XSPT	01036	MODEL	36014
XOPT	01037	SCAN	36324
XCLC	01040	KALM	36443
XHPT	01041		
XPET	01042	LC	
XEXA	01043		
H5	11125		
BUFD	11156		

G-13 LOADING PROCEDURES FOR FILTER3

The following loading procedure was used to construct an SLST of the on-line filtering program, FILTER3.

(i) Steps (i) to (vi) as for FILTER2 - See Appendix G-12.

(vii) Load an object tape of SUBROUTINE TRANS(2).

Press start

(viii) Load an object tape of SUBROUTINE RUTIS,

Press start.

(ix) Load an object tape of the Matrix Manipulation routines,

Prèss start.

(x) Load Part 1 of the OP-16 Fortran Libraries tape,

Press start.

(xi) Load Part 2 of the OP-16 Fortran Libraries tape,

Press start.

(xii) Load an object tape of the Fortran Links program - See Appendix G-11.

(xiii) Obtain a memory map of the load - see Table G-13.1.

(xiv) Make the following correction to subroutine F$ER,

location, F$ER+$11_8$ (33121_8) - contents, NOP.

(xv) Load an SLST of PALAP (entry point 15000_8) and punch out a 2 part SLST of FILTER3,

locations 100_8 to 477_8 and 16400_8 to 37577_8.

TABLE G-13.1 - MEMORY MAP OF FILTER3

*LOW	00000	BUFD	11156
*START	16400	HT	16406
*HIGH	33176	IN	20532
*NAMES	02236	KALMAN	22266
*COMN	36014	PREDIC	23430
CUF2	00000	TRANS	26024
COF1	00000	RUTIS	27060
CIF2	00000	MATINV	30054
CIF1	00000	MATMUL	31022
MTI2	00000	MATADD	31166
MTI1	00000	MATRAP	31304
LPI2	00000	A$22	31376
LPI1	00000	S$22	31400
ASF2	00000	M$22	31402
ASF1	00000	D$22	31404
ROCB	00000	F$AT	31506
ROCA	00000	F$R1	31572
*BASE	00412	F$W1	31600
XSTR	01000	F$L3	31606
XLNK	01001	F$L1	31606
XPFP	01002	F$L5	31612
XMIL	01003	F$L6	31612
XSEC	01004	F$L2	31612
XMIN	01005	F$CB	31616
XHR	01006	CON	31631
XDAY	01007	TAG	31656
XMSD	01010	EXP	31702
XPIC	01011	ALOG10	32242
XJBS	01012	ALOGX	32252
XPLP	01013	ALOG	32252
XPSP	01014	ABS	32540
XCCW	01015	FLOAT	32552
XEPP	01016	IFIX	32562
EROR	01016	IDINT	32562
XLOD	01017	INT	32562
XPEP	01020	E$21	32572
XSYS	01021	M$11X	32652
XICF	01022	M$11	32652
XSSA	01023	C$12	32732
XHLT	01024	C$21	32764
XPLT	01025	N$22	33016
XPCT	01026	REAL	33030
XCUT	01027	L$22	33030
XIDT	01030	H$22	33040
XID1	01031	ARG$	33056
XID2	01032	ADDR	33100
XEVT	01033	F$ER	33110
XINT	01034	L$33	33126
XDCT	01035	ISTORE	33142
XSPT	01036	IFETCH	33154
XOPT	01037	LOC	33164
XCLC	01040	MODEL	36014
XHPT	01041	SCAN	36324
XPET	01042	KALM	36443
XEXA	01043		
HS	11125	LC	

G-14 LOADING PROCEDURE FOR FILTER4

The following loading procedure was used to construct an SLST of the on-line filtering program, FILTER4.

(i) Steps (i) to (iv) as for FILTER2 - See Appendix G-12.

(v) Load an object tape of SUBROUTINE ADAPT,

Press start.

(vi) Steps (vi) to (xi) as for FILTER2 - See Appendix G-12.

(xii) Obtain a memory map of the load - See Table G-14.1.

(xiii) Make the following correction to subroutine F$ER,

location, F$ER+$11_8$ (34001_8) - contents, NOP

(xiv) Load an SLST of PALAP (entry point 15000_8) and punch a 2 part SLST of FILTER4.

locations 100_8 to 577_8 and 16400_8 to 37577_8

TABLE G-14.1 - MEMORY MAP OF FILTER4

*LOW	00000	BUFD	11156
*START	16400	HT	16406
*HIGH	34056	IN	20702
*NAMES	02167	ADAPT	22760
*COMN	35300	PREDIC	25260
COF2	00000	TRANS	27730
COF1	00000	MATINV	30664
CIF2	00000	MATMUL	31632
CIF1	00000	MATADD	31776
MTI2	00000	MATRAP	32114
MTI1	00000	A$22	32206
LPI2	00000	S$22	32210
LPI1	00000	M$22	32212
ASF2	00000	D$22	32214
ASF1	00000	F$AT	32316
ROCB	00000	F$R1	32402
ROCA	00000	F$W1	32410
*BASE	00501	F$L3	32416
XSTR	01000	F$L1	32416
XLNK	01001	F$L5	32422
XPFP	01002	F$L6	32422
XMIL	01003	F$L2	32422
XSEC	01004	F$CB	32426
XMIN	01005	CON	32441
XHR	01006	TAG	32466
XDAY	01007	EXP	32512
XMSD	01010	ALOG10	33052
XPIC	01011	ALOGX	33062
XJBS	01012	ALOG	33062
XPLP	01013	ABS	33350
XPSP	01014	FLOAT	33362
XCCW	01015	IFIX	33372
XEPP	01016	IDINT	33372
EROR	01016	INT	33372
XLOD	01017	E$21	33402
XPEP	01020	M$11X	33462
XSYS	01021	M$11	33462
XICF	01022	D$11X	33542
XSSA	01023	D$11	33542
XHLT	01024	C$12	33612
XPLT	01025	C$21	33644
XPCT	01026	N$22	33676
XCUT	01027	REAL	33710
XIDT	01030	L$22	33710
XID1	01031	H$22	33720
XID2	01032	ARG$	33736
XIVT	01033	ADDR	33760
XINT	01034	F$ER	33770
XDCT	01035	L$33	34006
XSPT	01036	ISTORE	34022
XOPT	01037	IFETCH	34034
XCLC	01040	LOC	34044
XHPT	01041	MODEL	35300
XPET	01042	SCAN	35610
XEXA	01043	KALM	35727
HS	11125		

LC

G-15 OP-16 FORTRAN LIBRARIES TAPE

The OP-16 FORTRAN LIBRARIES TAPE contains the following libraries which are supplied as standard software by the manufacturer:-

	LIBRARY NAME	CONTENTS	NUMBER OF SUBROUTINES
1	CLIB	COMPLEX SUBROUTINES	22
2	DLIB	DOUBLE PRECISION SUBROUTINES	35
3	OPFRT1	TRACE SUBROUTINE	1
4	OPFRT2H	LINKS TO RE-ENTRANT SUBROUTINES AND DRIVERS	29
5	IRLIB	INTEGER/REAL	33
6	OPFRT3	OP-16 SUBROUTINES	16
7	ULIB	UTILITY ROUTINES	20
8	AC1-5	DOUBLE PRECISION ACCUMULATORS	1
9	XLOCS	EXEC-A LOCATIONS	1
		TOTAL	158

The OP-16 Fortran Libraries Tape was constructed using the utility program OBJCHOP (G.1). The tape comes in two parts, the first part containing CLIB, DLIB, OPFRT1 and the first 12 subroutines of OPFRT2H (in that order) and the second part containing the remaining subroutines from OPFRT2H, IRLJB, OPFRT3, ULIB, AC1-5 and XLOCS (in that order). When using this tape it must be remembered that the first subroutine on OPFRT2H (the links with the floating point re-entrant mathematics subroutines) is always loaded because it is preceded by a force load block. A further point to take note of is that this library tape contains XLOCS, an object tape which specifies the key locations in the executive (EXEC-A). Thus, when this tape is loaded with the object tape of a Fortran program the linkages with the executive are automatically satisfied.

TABLE G-16.1 - ERRORS GENERATED BY FILTER2, 3 and 4

ERROR MNEMONIC	MEANING	REPORTING PROGRAM
E3000IN	An Executive error has occurred when program IN attempted to REQUEST either program H1 or program H2. Program IN is terminated.	INIT
E3100HT	An Executive error has occurred when program HT attempted to REQUEST one of the followings programs: H1, H2, H3, H4, H5, H6, PP. Program HT is terminated.	FILTER
E3110HT	An Executive error has occurred when program HT attempted to disconnect itself from the clock. Program HT is terminated.	FILTER

G-17 PROGRAM TO CONVERT BINARY FORMAT TAPES TO ASCII FORMAT TAPES

The purpose of the program shown in Table G-17.1 is to read in a tape containing numbers in binary format and then punch out the same numbers in ASCII format. A rather unusual feature of Fortran which is used in this program is the unformatted READ statement - see line 17 of Table G-17.1. This statement enables numbers in binary format to be read in.

The program was loaded using LDR-APM Revision E (entry point 37000_8) and the following loading procedure:-

(i) Load an object tape of the program shown in Table G-17.1,

$P=37000_8$, $A=1000_8$, $B=100_8$

(ii) Force load an object tape of subroutine I$PO,

$P=37004_8$

(iii) Load the standard Fortran Libraries Tape.

TABLE G-17.1 - PROGRAM TO CONVERT BINARY FORMAT TAPES TO ASCII FORMAT TAPES

```
C CONVERSION OF BINARY TO ASCII                        PAGE     1

C CONVERSION OF BINARY TO ASCII
C
      DIMENSION A(500)
 1000 FORMAT(47H ENTER NO. OF CHARACTERS/CYCLE & NO. OF CYCLES,
     1 5H- 2I2)
 1010 FORMAT(E20.12)
 2000 FORMAT(2I2)
C
C INITIALISATION
C
      WRITE(1,1000)
      READ(1,2000) N,NCYCL
      NCOUNT=1
C
C INPUT
C
    5 READ(2) (A(I),I=1,N)
C
C OUTPUT
C
      DO 10 I=1,N
   10 WRITE(2,1010) A(I)
C
C TEST FOR END OF TAPE
C
      NCOUNT=NCOUNT+1
      IF(NCOUNT.LE.NCYCL) GOTO 5
      STOP
      END
$0   @
D OF JOB
```

APPENDIX H

STEADY STATE ANALYSIS OF THE DOUBLE EFFECT EVAPORATOR

CONTENTS

TABLE H-1.1 - STEADY STATE ANALYSIS EXPERIMENTAL RESULTS - PREHEATER (PH)

No	$T_4-(T_1+T_2)/2$	M_1	V_3	U_{PH}
1	30.791	0.031525	0.0140587	0.631288
2	30.7253	0.031525	0.0128492	0.644656
3	30.07098	0.031525	0.0143586	0.649298
4	30.6803	0.0312644	0.0137707	0.651013
5	32.1187	0.031525	0.0153043	0.576347
6	32.1903	0.031525	0.015454	0.575306
7	32.4991	0.031525	0.0159598	0.565543
8	32.1752	0.031525	0.0169249	0.575806
9	32.3705	0.031525	0.0170445	0.565747
10	32.9218	0.0471625	0.0188125	0.673937
11	32.7518	0.0471625	0.0169983	0.678993
12	32.714	0.0471625	0.0198001	0.680036
13	32.7985	0.0471625	0.018037	0.678288
14	32.8773	0.0471625	0.0180697	0.677936
15	34.6479	0.031525	0.0179933	0.534231
16	34.5988	0.031525	0.0188861	0.54025
17	34.5557	0.031525	0.0188028	0.543882
18	34.5605	0.031525	0.0191836	0.544979
19	34.5019	0.031525	0.0186898	0.546527
20	35.3886	0.0471625	0.019474	0.729083
21	35.3278	0.0471625	0.0195569	0.730318
22	35.2646	0.0471625	0.0192833	0.732172
23	35.4557	0.0471625	0.02089	0.732044
24	35.4328	0.0471625	0.0204153	0.729781
25	36.5988	0.0471625	0.024912	0.793213
26	36.8137	0.0463806	0.0242567	0.787649

TABLE H-1.1 - continued

27	37.1501	0.0463806	0.0229458	0.79492
28	37.3546	0.0463806	0.0249998	0.811615
29	37.515	0.0463806	0.0255457	0.814579
30	37.8034	0.0463806	0.022993	0.798157
31	30.56435	0.031525	0.0204263	0.584486
32	30.3113	0.031525	0.0181033	0.582702
33	30.19395	0.031525	0.0176306	0.582096
34	30.18905	0.031525	0.0170554	0.579674
35	30.11005	0.031525	0.0161998	0.579838

TABLE H-2.1 - STEADY STATE ANALYSIS EXPERIMENTAL RESULTS - FIRST EFFECT (FE)

No	$T_5-(T_2+T_7)/2$	M_2	V_6	U_{FE}
1	32.47815	0.031525	0.0139459	1.58857
2	32.22025	0.031525	0.0125739	1.44598
3	32.1084	0.031525	0.0142444	1.64103
4	31.9808	0.0312644	0.0135923	1.58881
5	32.6208	0.031525	0.0154082	1.81504
6	32.473	0.031525	0.0155725	1.84595
7	32.35615	0.031525	0.0161433	1.93369
8	32.35205	0.031525	0.0172057	2.04884
9	32.4883	0.031525	0.0173588	2.06659
10	37.4298	0.0471625	0.018979	1.9918
11	37.4596	0.0471625	0.0169508	1.77388
12	37.4695	0.0471625	0.0200722	2.09961
13	37.4329	0.0471625	0.0181062	1.89711
14	37.40415	0.0471625	0.0181414	1.90374
15	31.2249	0.031525	0.0183925	2.38279
16	31.02605	0.031525	0.0193671	2.52192
17	30.9082	0.031525	0.0192679	2.51889
18	30.84185	0.031525	0.0196876	2.5809
19	30.83465	0.031525	0.0191381	2.50819
20	32.45975	0.0471625	0.0194432	2.49265
21	32.5437	0.0471625	0.0195333	2.49009
22	32.53445	0.0471625	0.019226	2.45024
23	32.48765	0.0471625	0.0210005	2.67554
24	32.5307	0.0471625	0.0204877	2.61685
25	29.7926	0.0471625	0.025307	3.65843
26	29.2406	0.0463806	0.0245748	3.63176
27	28.525	0.0463806	0.0230725	3.51738

TABLE H-2.1 - continued

28	28.1489	0.0463806	0.0252778	3.91532
29	27.8378	0.0463806	0.0258575	4.05782
30	27.59125	0.0463806	0.023057	3.66107
31	36.59945	0.031525	0.0213185	2.15066
32	37.0407	0.031525	0.0187318	1.85848
33	37.21465	0.031525	0.0182087	1.79588
34	37.3314	0.031525	0.0175719	1.72784
35	37.37765	0.031525	0.0166177	1.63278

TABLE H-3.1 - STEADY STATE ANALYSIS EXPERIMENTAL RESULTS - SECOND EFFECT (SE)

No	$T_{10}-(T_{15}+T_{14})/2$	M_{15}	V_4	U_{SE}
1	11.59385	1.93128	0.0114719	2.56312
2	11.68445	2.02009	0.010234	2.73125
3	11.7523	1.92253	0.0117337	2.28462
4	11.7503	1.9772	0.0111527	2.61207
5	13.42685	2.07081	0.0127497	2.35934
6	13.2267	2.0715	0.0128961	2.35734
7	13.43625	1.91256	0.0134053	2.19322
8	13.13145	1.92166	0.0143664	2.04884
9	13.11565	1.9143	0.0145	1.97173
10	12.62335	2.04682	0.0155388	1.22479
11	12.4437	1.90597	0.0137238	1.73618
12	12.3169	1.837845	0.0165258	0.989686
13	12.34715	2.24273	0.0147602	1.46921
14	12.34405	2.25015	0.0147861	1.46112
15	14.17885	2.24403	0.0153655	1.44319
16	14.1498	2.06075	0.0162418	1.23548
17	14.08005	2.07563	0.0161491	1.26094
18	14.11595	1.99765	0.0165257	1.16745
19	14.0811	1.99226	0.0160315	1.29714
20	13.74605	1.99447	0.0157166	1.14702
21	13.5987	1.994641	0.0158015	1.1414
22	13.5697	1.96573	0.015528	1.21105
23	13.5689	1.941296	0.0171136	0.819439
24	13.451	1.947817	0.0166506	0.942613
25	14.19495	1.990891	0.0207759	0.313469
26	14.00325	1.95082	0.0201343	0.482596
27	13.73765	1.949086	0.0187544	0.810392

TABLE H-3.1 - continued

28	13.7346	1.979743	0.0206994	0.308687
29	13.7491	1.96294	0.0212136	0.177258
30	13.84245	2.03431	0.0187077	0.794109
31	11.12175	2.064717	0.0179893	0.783844
32	11.12085	1.94441	0.0156927	1.4899
33	11.08885	1.9348	0.0152315	1.63802
34	11.06905	1.89056	0.0146635	1.81635
35	11.0914	2.00415	0.013814	2.07212

TABLE H-4.1 - STEADY STATE ANALYSIS EXPERIMENTAL

RESULTS - CONDENSER (CD)

No	$T_{11}-(T_{12}+T_{13})/2$	M_{12}	$V_{10}+V_9$	U_{CD}
1	25.60765	0.433333	0.0120391	0.17357
2	25.39495	0.433333	0.0108246	0.161246
3	25.31275	0.433333	0.0122999	0.189648
4	25.72085	0.433333	0.0117065	0.156428
5	26.1512	0.441667	0.0131146	0.15478
6	24.9983	0.441667	0.0132566	0.159532
7	24.98155	0.441667	0.0137632	0.163337
8	25.10675	0.441667	0.0147138	0.160834
9	25.0256	0.441667	0.0148452	0.163425
10	23.7683	0.441667	0.0163119	0.164756
11	23.6652	0.441667	0.0145005	0.160768
12	23.45755	0.441667	0.0172784	0.164479
13	23.3187	0.441667	0.0155205	0.166031
14	23.19235	0.441667	0.0155438	0.165043
15	25.9365	0.435000	0.0157308	0.165536
16	26.5293	0.435000	0.0165995	0.162018
17	26.96355	0.435000	0.0165096	0.159597
18	27.31245	0.435000	0.016884	0.157615
19	27.55375	0.435000	0.0163954	0.156904
20	27.0136	0.453667	0.0166148	0.155279
21	26.8837	0.453667	0.0166898	0.153605
22	26.88675	0.453667	0.0164191	0.151764
23	26.58035	0.453667	0.0179741	0.149369
24	26.5757	0.453667	0.0175236	0.148954
25	27.93955	0.436667	0.0216349	0.144292
26	29.112	0.436667	0.0209594	0.135525

TABLE H-4.1 - continued

27	30.4769	0.436667	0.0195816	0.1282
28	31.27295	0.436667	0.0215292	0.127331
29	32.0272	0.436667	0.0220348	0.122653
30	32.68445	0.446667	0.0195218	0.12025
31	26.9425	0.446667	0.0183007	0.111216
32	26.1062	0.446667	0.0160184	0.115537
33	25.81365	0.446667	0.0155611	0.113205
34	25.4886	0.446667	0.014998	0.115877
35	25.34415	0.446667	0.0141596	0.119043

TABLE H-5.1 - COMPARISON OF PREDICTED AND EXPERIMENTAL HEAT TRANSFER COEFFICIENTS - PREHEATER (U_{PH})

Run No	PREDICTED VALUE	PERCENTAGE DEVIATION	Run No	PREDICTED VALUE	PERCENTAGE DEVIATION
1	0.617830	2.132	26	0.777889	1.239
2	0.623761	3.241	27	0.759996	4.393
3	0.638542	1.656	28	0.794346	2.128
4	0.649021	0.306	29	0.801747	1.575
5	0.534554	7.251	30	0.760528	4.714
6	0.536579	6.732	31	0.604506	-3.425
7	0.543409	3.914	32	0.572941	1.675
8	0.556594	3.336	33	0.566529	2.674
9	0.558189	1.336	34	0.558704	3.617
10	0.710907	-5.486	35	0.547076	5.650
11	0.686251	-1.069			
12	0.724379	-6.521			
13	0.700376	-3.256			
14	0.700807	-3.374			
15	0.570715	-6.829			
16	0.582872	-7.889			
17	0.581745	-6.962			
18	0.586925	-7.697			
19	0.580217	-6.164			
20	0.719492	1.315			
21	0.72063	1.326			
22	0.716918	2.083			
23	0.738747	-0.916			
24	0.732292	-0.344			
25	0.793279	-0.008			

AVERAGE ABSOLUTE DEVIATION = 3.499%

TABLE H-6.1 - COMPARISON OF PREDICTED AND EXPERIMENTAL HEAT TRANSFER COEFFICIENTS - FIRST EFFECT ($\underline{U}_{FE}$)

Run No	PREDICTED VALUE	PERCENTAGE DEVIATION	Run No	PREDICTED VALUE	PERCENTAGE DEVIATION
1	1.57884	0.613	27	3.45257	1.842
2	1.39419	3.582	38	3.83021	2.174
3	1.65956	-1.129	29	3.99372	1.579
4	1.57043	1.157	30	3.69987	1.059
5	1.81328	0.097	31	2.05086	4.640
6	1.82735	1.007	32	1.8741	-0.840
7	1.92552	0.423	33	1.77777	1.008
8	2.0879	-1.906	34	1.66955	3.373
9	2.09827	-1.533	35	1.61252	1.241
10	1.98595	0.294			
11	1.76246	0.644			
12	2.14886	-2.346			
13	1.85257	2.348			
14	1.86067	2.262			
15	2.3762	0.276			
16	2.54375	-0.865			
17	2.53984	-0.832			
18	2.61016	-1.134			
19	2.52706	-0.752			
20	2.53003	-1.499			
21	2.53577	-1.834			
22	2.4898	-1.614			
23	2.69327	-0.663			
24	2.68254	-2.510			
25	3.67814	-0.539			
26	3.6135	-0.503			

AVERAGE ABSOLUTE DEVIATION = 1.396%

TABLE H-7.1 - COMPARISON OF PREDICTED AND EXPERIMENTAL HEAT TRANSFER COEFFICIENTS - SECOND EFFECT (U_{SE})

Run No	PREDICTED VALUE	PERCENTAGE DEVIATION	Run No	PREDICTED VALUE	PERCENTAGE DEVIATION
1	2.55231	0.422	27	0.760893	6.108
2	2.64793	3.051	28	0.34743	-12.551
3	2.11951	7.227	29	0.199834	-12.786
4	2.63725	-0.964	30	0.757757	4.578
5	2.38142	-0.936	31	0.731037	6.737
6	2.35968	-0.099	32	1.49777	-0.528
7	2.19207	0.052	33	1.69423	-3.432
8	1.9341	5.600	34	1.87296	-3.117
9	1.88244	4.528	35	2.12676	-2.637
10	1.25367	-2.358			
11	1.78832	-3.003			
12	0.978501	1.130			
13	1.47351	-0.292			
14	1.47701	-1.087			
15	1.49792	-3.792			
16	1.26602	-2.472			
17	1.28371	-1.806			
18	1.18679	-1.657			
19	1.30823	-0.855			
20	1.14683	0.017			
21	1.14815	-0.591			
22	1.20375	0.603			
23	0.899683	-7.790			
24	1.02431	-7.866			
25	0.350189	-11.714			
26	0.441382	8.540			

AVERAGE ABSOLUTE DEVIATION = 3.739%

TABLE H-8.1 - COMPARISON OF PREDICTED AND EXPERIMENTAL HEAT TRANSFER COEFFICIENTS - CONDENSER (U_{CD})

Run No	PREDICTED VALUE	PERCENTAGE DEVIATION	Run No	PREDICTED VALUE	PERCENTAGE DEVIATION
1	0.169776	2.186	27	0.131308	-2.424
2	0.170861	-5.963	28	0.127257	0.058
3	0.171276	9.687	29	0.123361	-0.577
4	0.1692	-8.165	30	0.120076	0.145
5	0.153319	0.944	31	0.112765	-1.392
6	0.159185	0.217	32	0.113426	1.827
7	0.159234	2.512	33	0.113656	-0.398
8	0.158633	1.368	34	0.116386	-0.439
9	0.159046	2.679	35	0.120013	-0.815
10	0.16497	-0.129			
11	0.165968	-3.234			
12	0.167024	-1.547			
13	0.167731	-1.024			
14	0.168374	-2.018			
15	0.165359	-0.107			
16	0.162343	-0.200			
17	0.160133	-0.336			
18	0.158358	-0.471			
19	0.15713	-0.144			
20	0.14893	4.089			
21	0.149591	2.613			
22	0.149576	1.442			
23	0.151134	-1.182			
24	0.151158	-1.479			
25	0.144218	0.052			
26	0.138252	-2.012			

AVERAGE ABSOLUTE DEVIATION = 1.808%

H-9 DATA LAYOUT FOR THE MULREG COMPUTER PROGRAM

Data should be entered as follows:

```
9900 DATA N,V,R
9901 DATA M1,N1,P1 ------W1
9902 DATA M2,N2,P2 ------W2


990N DATA Mn,Nn,Pn ------Wn
99XX DATA 1,G1,Q1,Q2,X1,X2,X3 ---X_G1,Y
99XY DATA 2,G2,Q1,Q2,X1,X2,X3 ---X_G2,Y


99ZZ DATA R,GR,Q1,Q2,X1,X2,X3 ---X_GR,Y
```

where, N = Number of Data Sets

V = Number of variables per data set

R = Number of Regressions to be solved in this Run.

Mn,Nn,Pn---Wn = the complete data set including both dependent and independent variables, for the nth observation.

GR = The number of independent variables in the Rth regression model.

Q1 = The control variable for the variance-covariance matrix, 1 to print, 0 to omit.

Q2 = The control variable for the calculated versus actual with residuals table, 1 to print, 0 to omit.

Xi = The index of position in the data matrix (lines 9901 to 990N) for the ith independent variable in the model.

Y = The index of position in the data matrix for the dependent variable.

APPENDIX J

DYNAMIC ANALYSIS OF THE DOUBLE EFFECT EVAPORATOR

CONTENTS

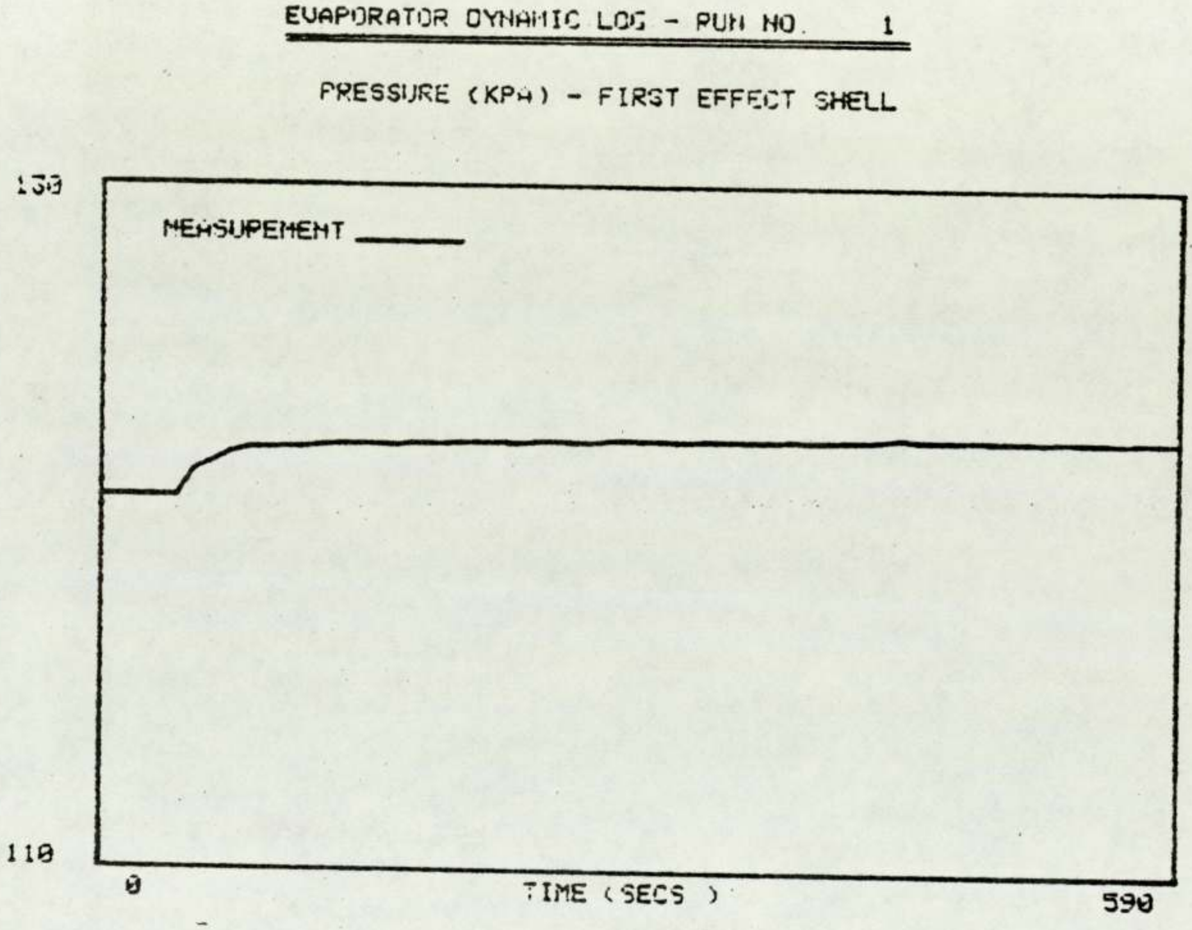


FIGURE J-1.1 - EFFECT OF A STEP CHANGE TO THE STEAM FLOWRATE

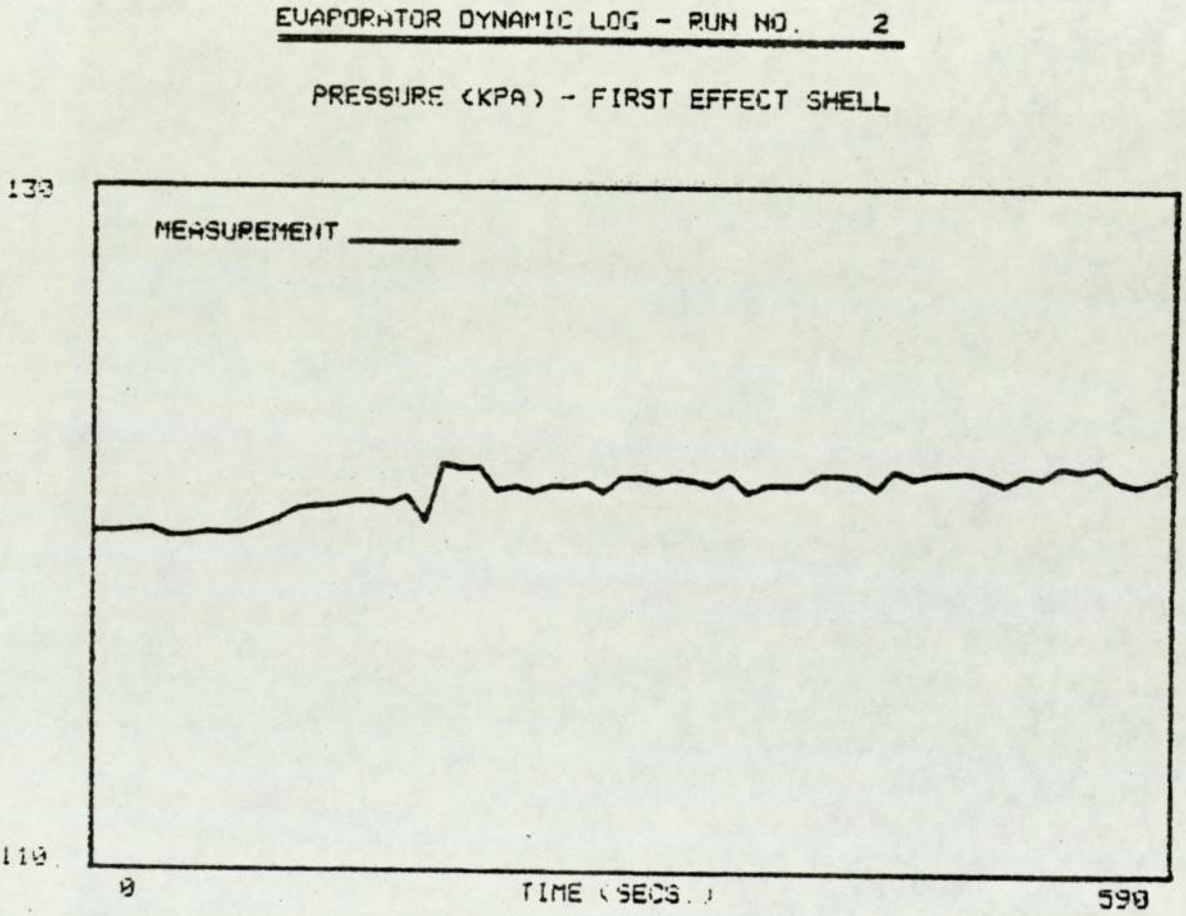


FIGURE J-1.2 - EFFECT OF A STEP CHANGE TO THE FEED FLOWRATE

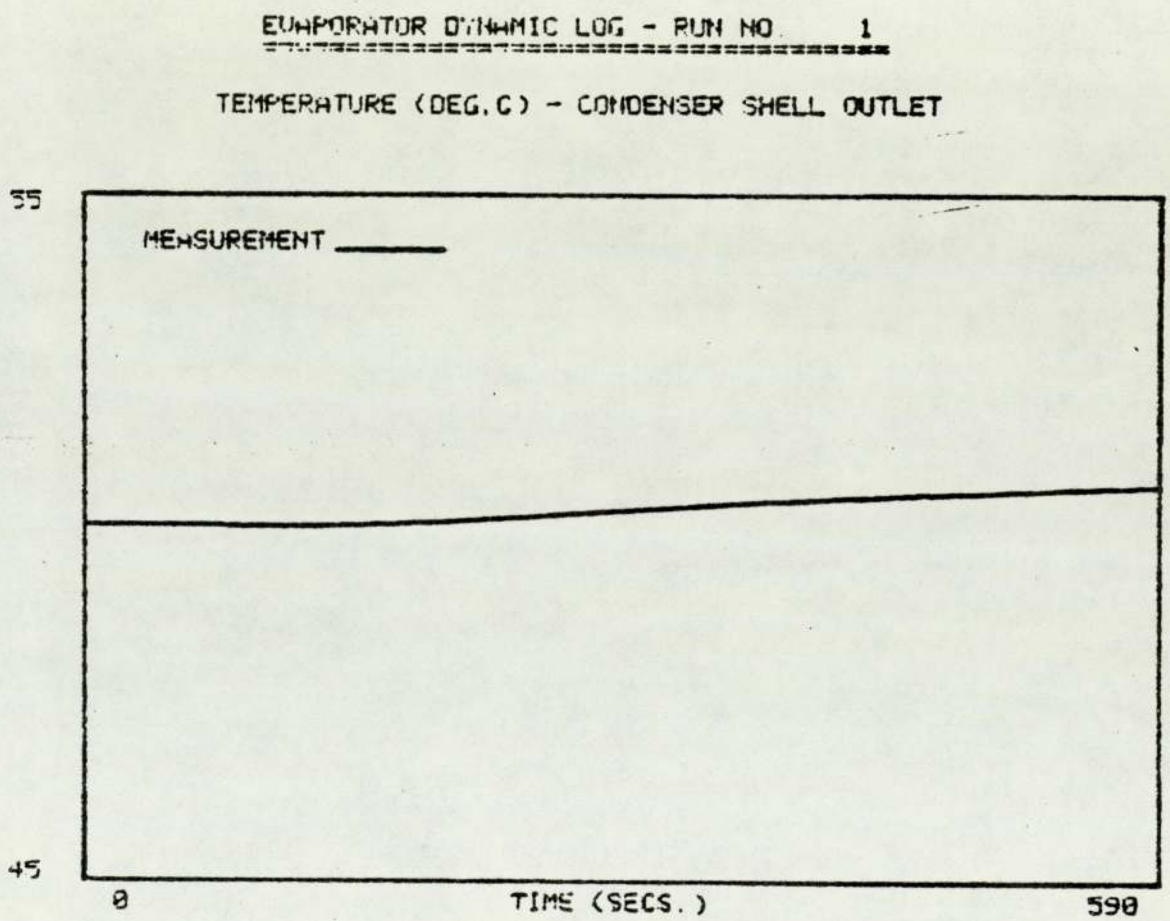


FIGURE J-2.1 - EFFECT ON THE CONDENSER SHELL OUTLET TEMPERATURE OF A STEP CHANGE TO THE STEAM FLOWRATE

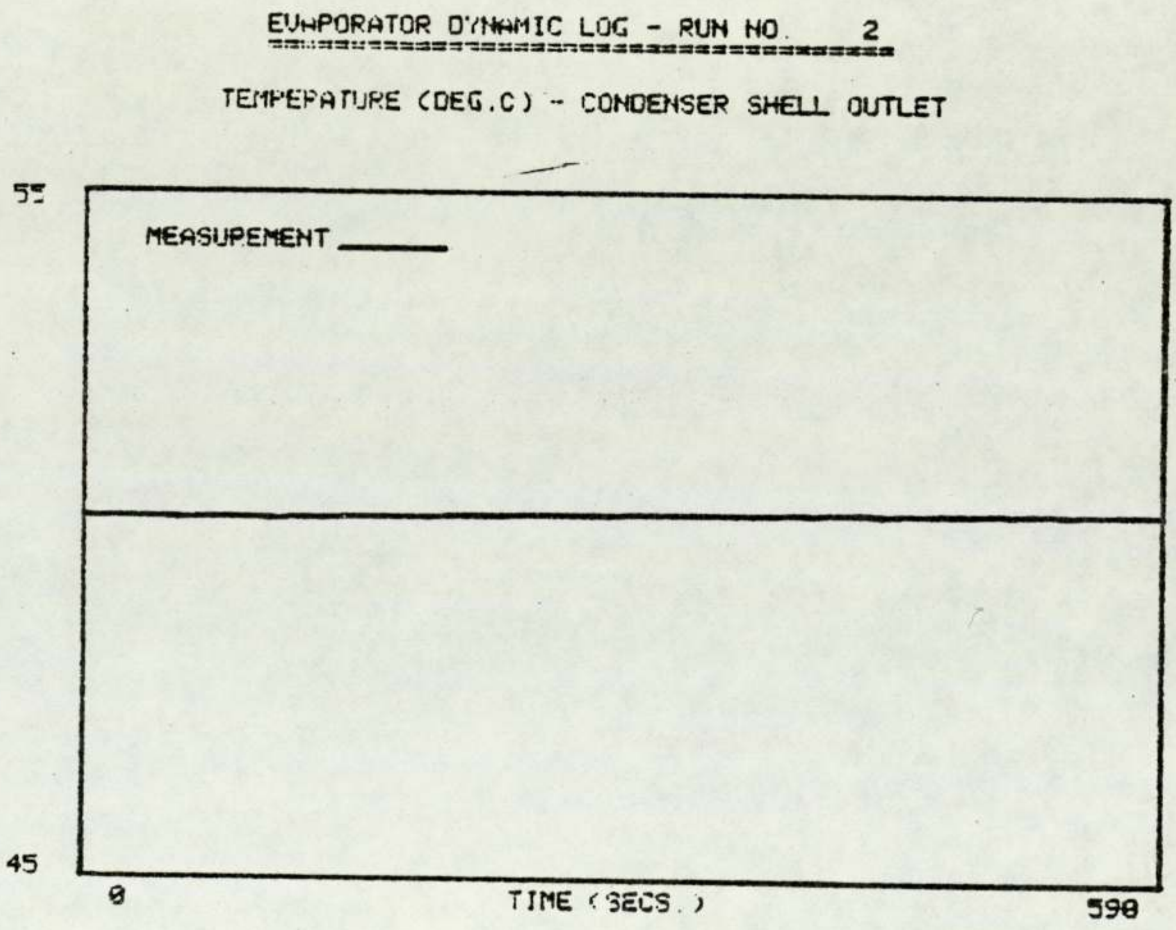


FIGURE J-2.2 - EFFECT ON THE CONDENSER SHELL OUTLET TEMPERATURE OF A STEP CHANGE TO THE FEED FLOWRATE

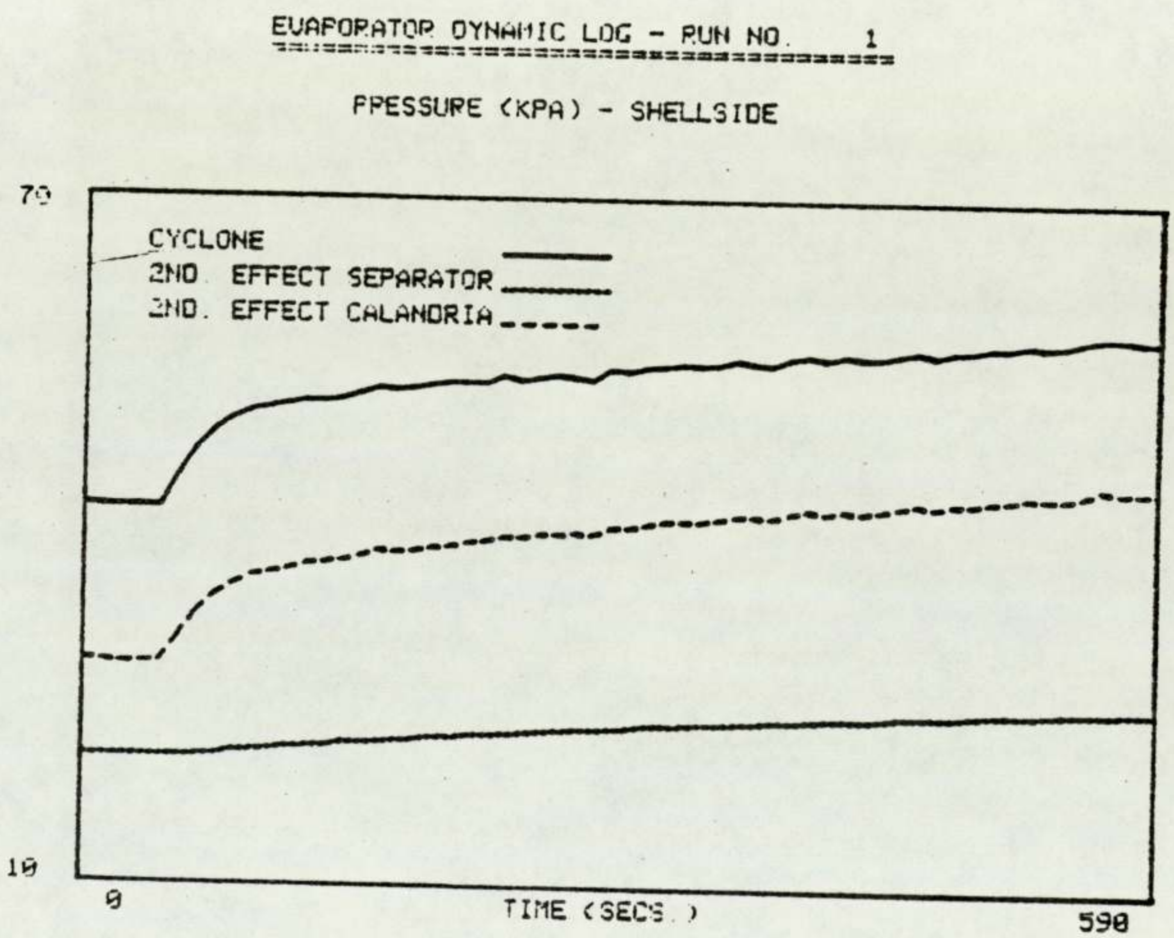


FIGURE J-3.1 - EFFECT ON THE SHELLSIDE PRESSURES OF A STEP CHANGE TO THE STEAM FLOWRATE

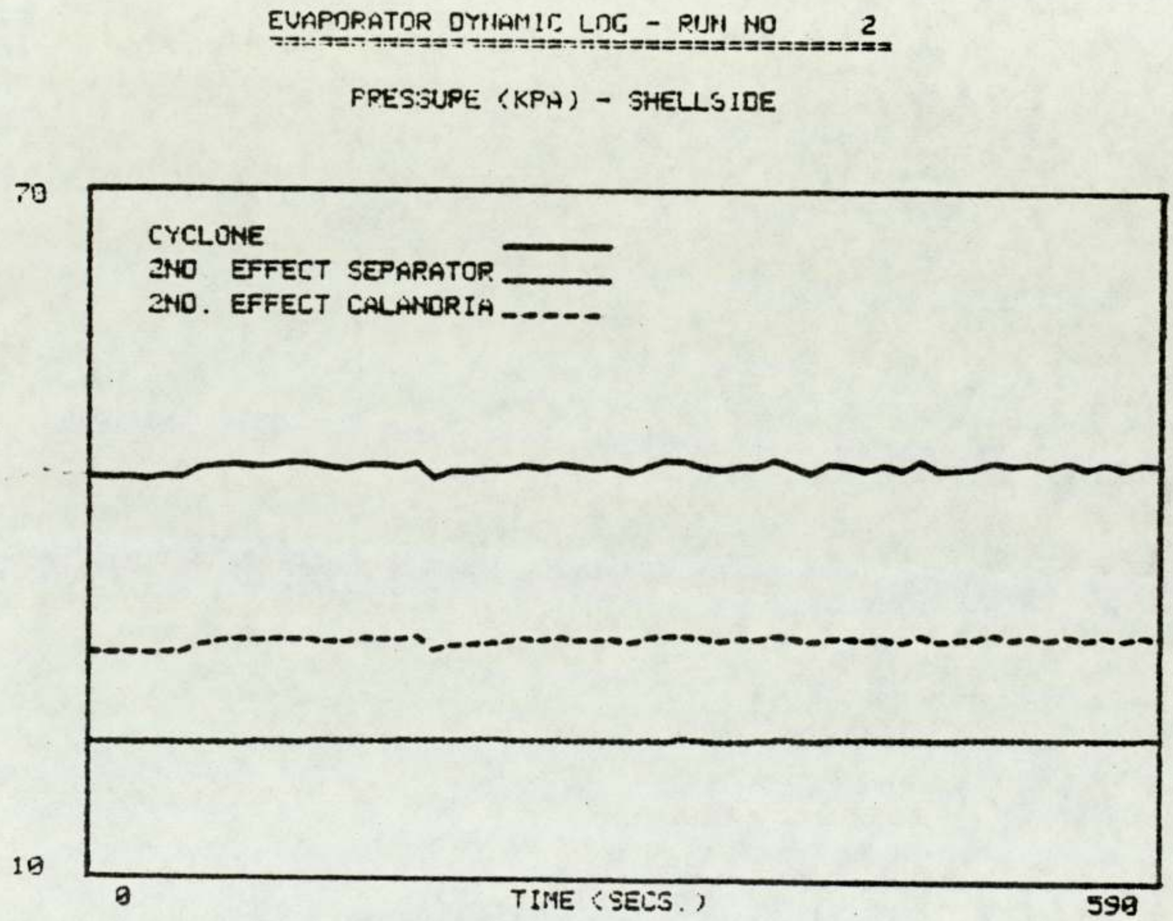


FIGURE J-3.2 - EFFECT ON THE SHELLSIDE PRESSURES OF A STEP CHANGE TO THE FEED FLOWRATE

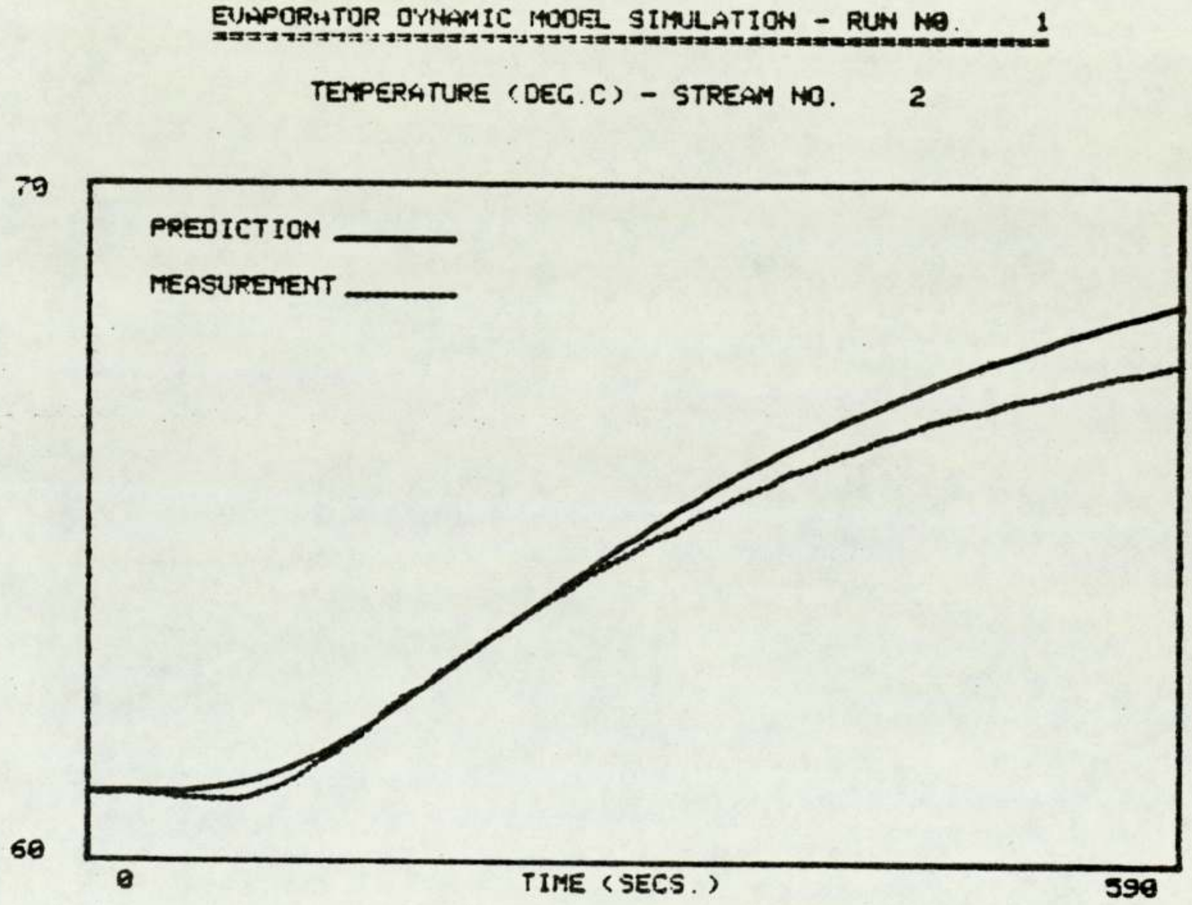


FIGURE J-4.1 - TEMPERATURE - EXIT PREHEATER TUBES

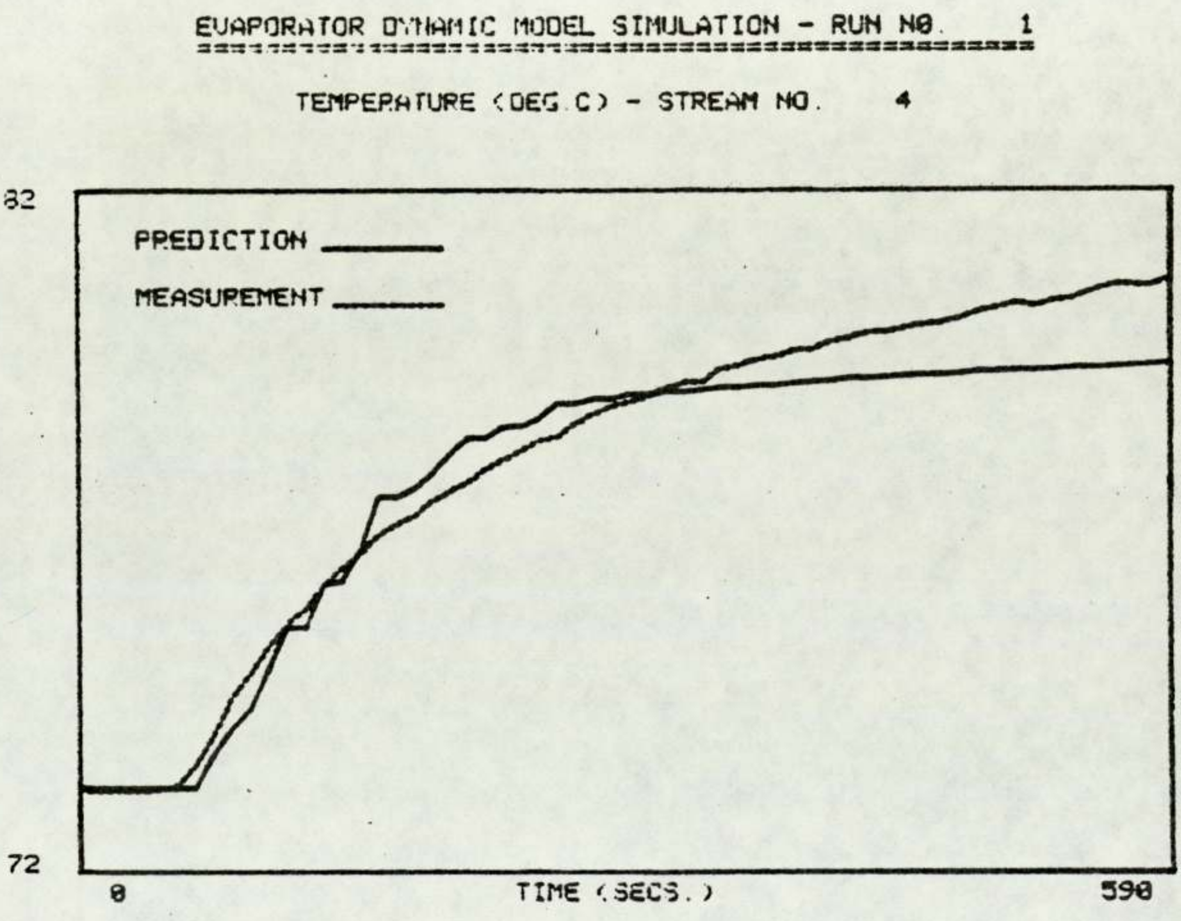


FIGURE J-4.2 - TEMPERATURE - EXIT PREHEATER SHELL

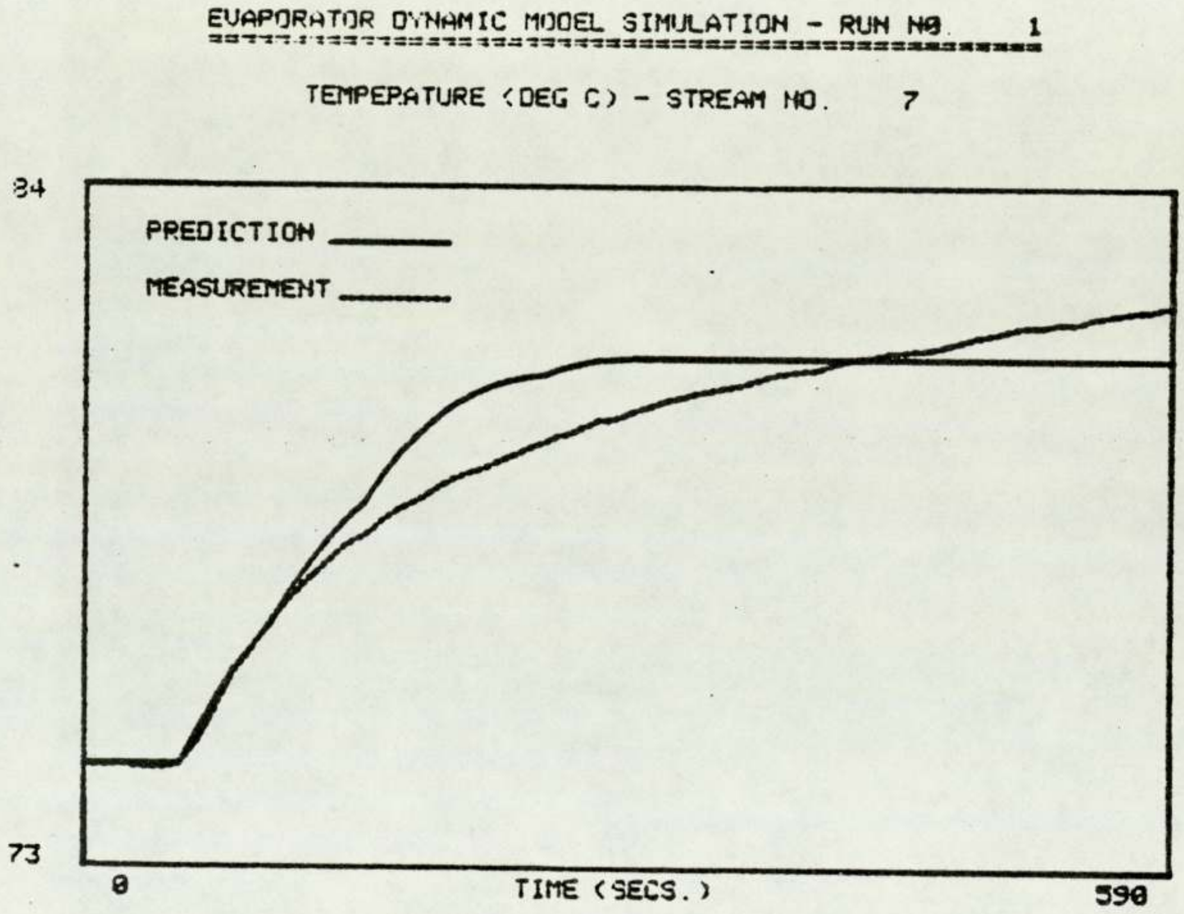


FIGURE J-4.3 - TEMPERATURE - EXIT FIRST EFFECT TUBES

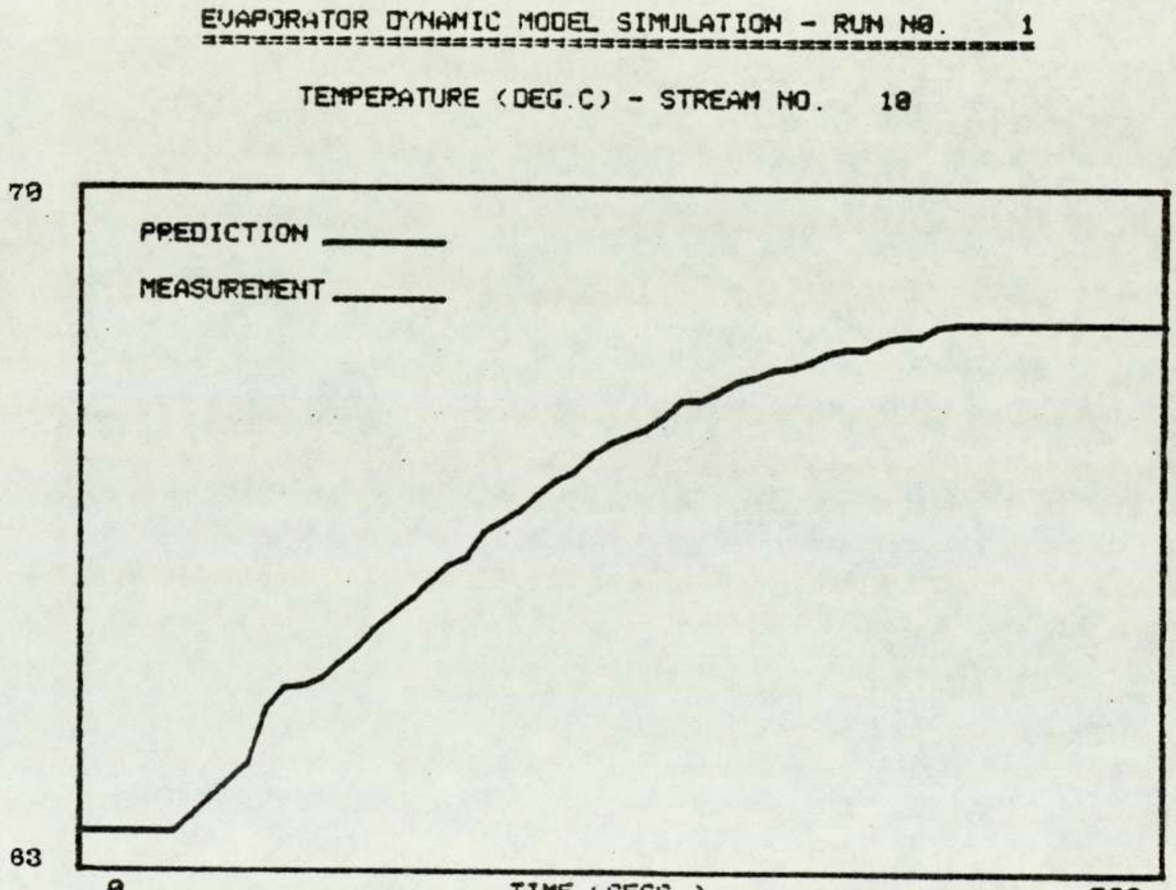


FIGURE J-4.4 - TEMPERATURE - EXIT SECOND EFFECT CALANDRIA SHELL

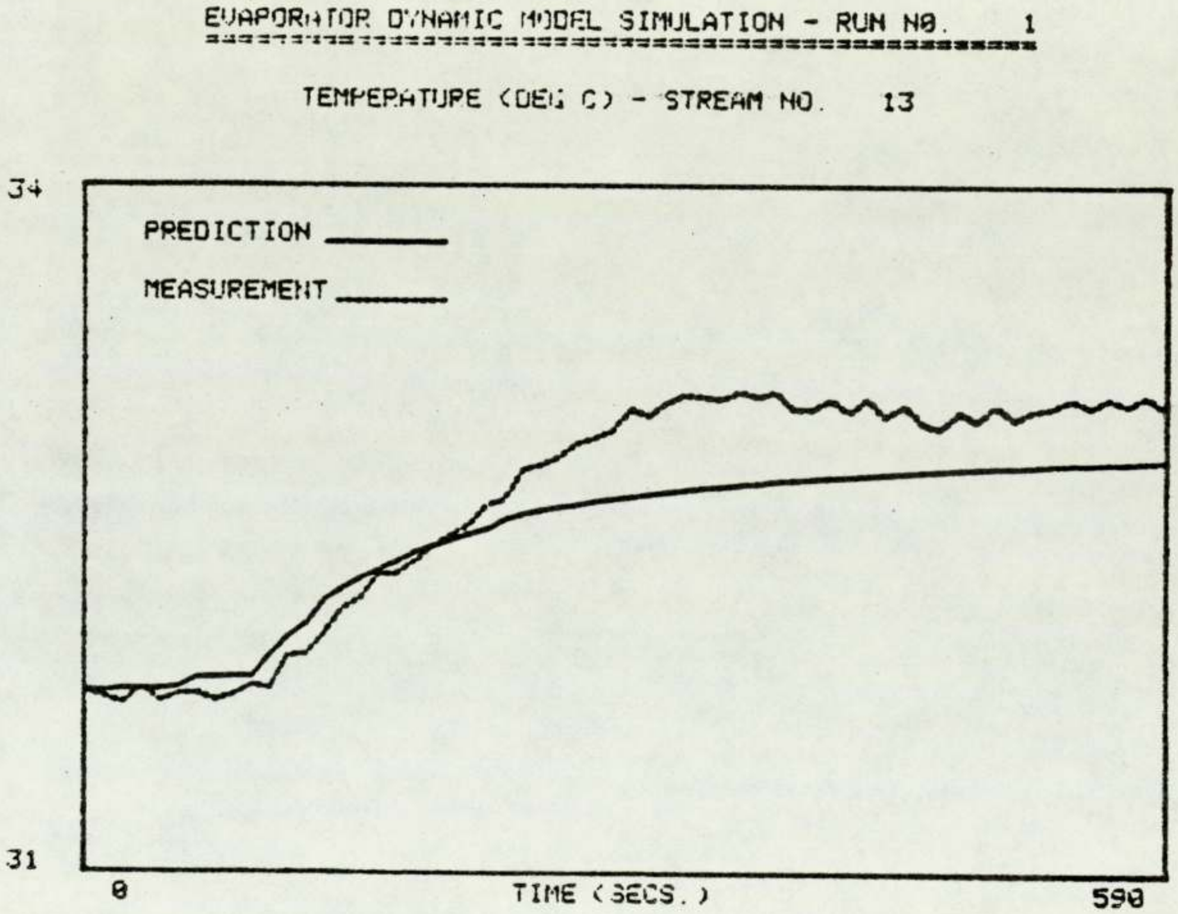


FIGURE J-4.5 - TEMPERATURE - EXIT CONDENSER TUBES

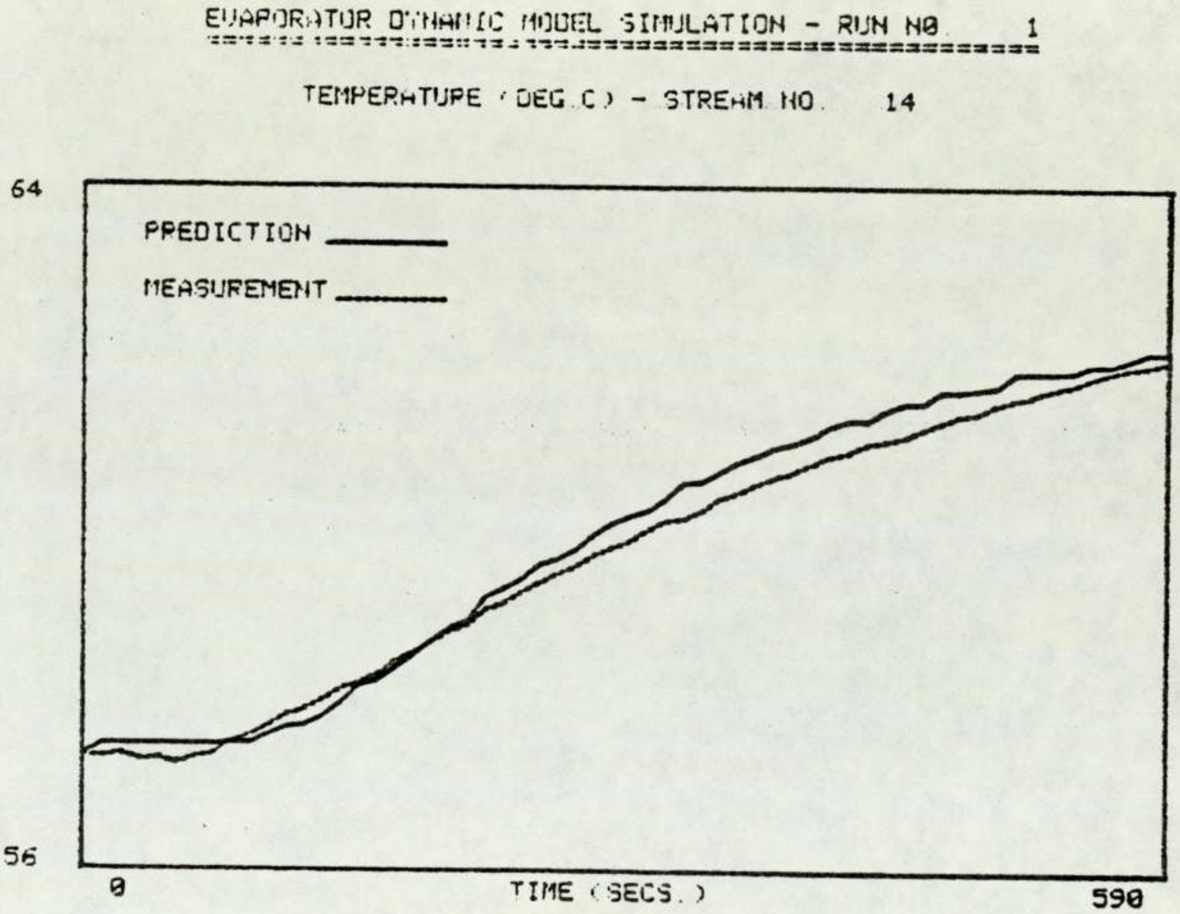


FIGURE J-4.6 - TEMPERATURE - EXIT SECOND EFFECT

CALANDRIA TUBES

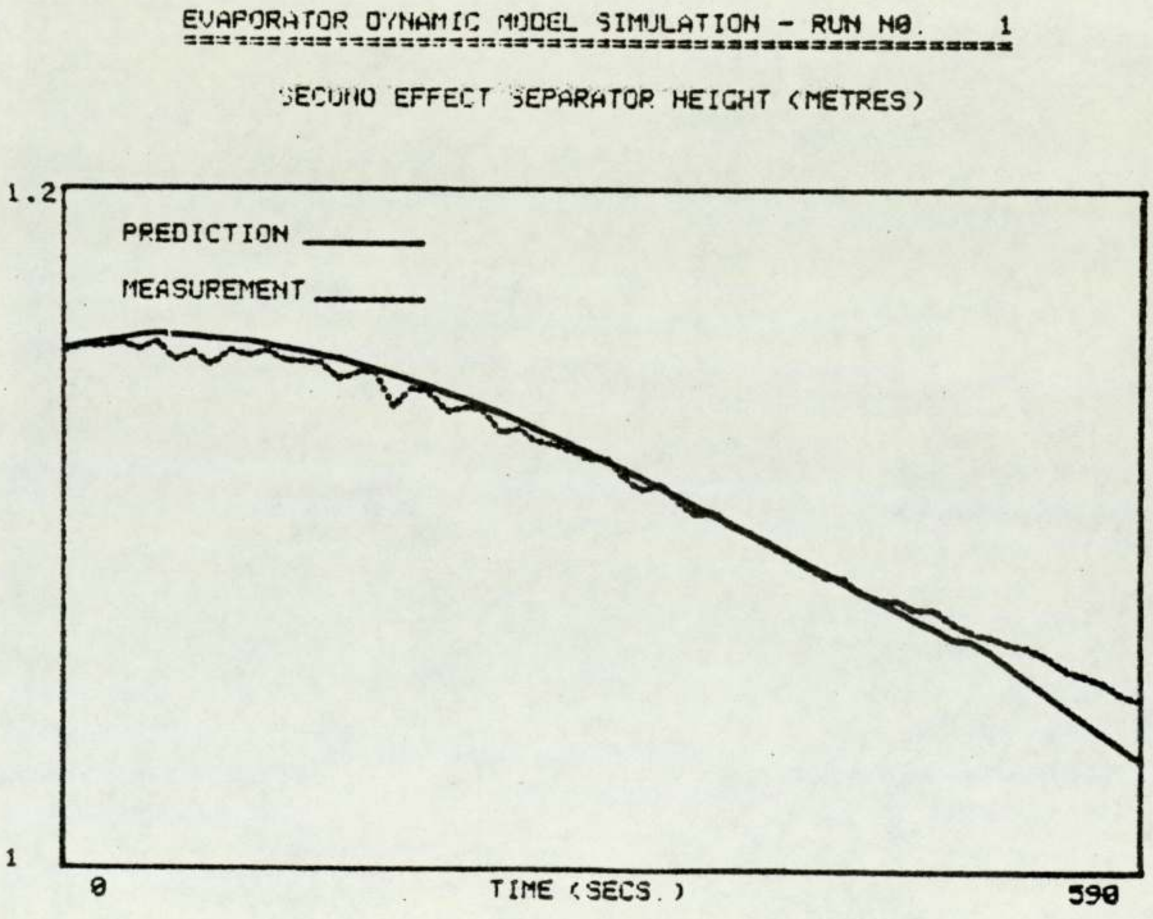

EVAPORATOR DYNAMIC MODEL SIMULATION - RUN NO. 1
SECOND EFFECT SEPARATOR HEIGHT (METRES)
1.2
PREDICTION
MEASUREMENT
1
0
TIME (SECS.)
590

FIGURE J-4.7

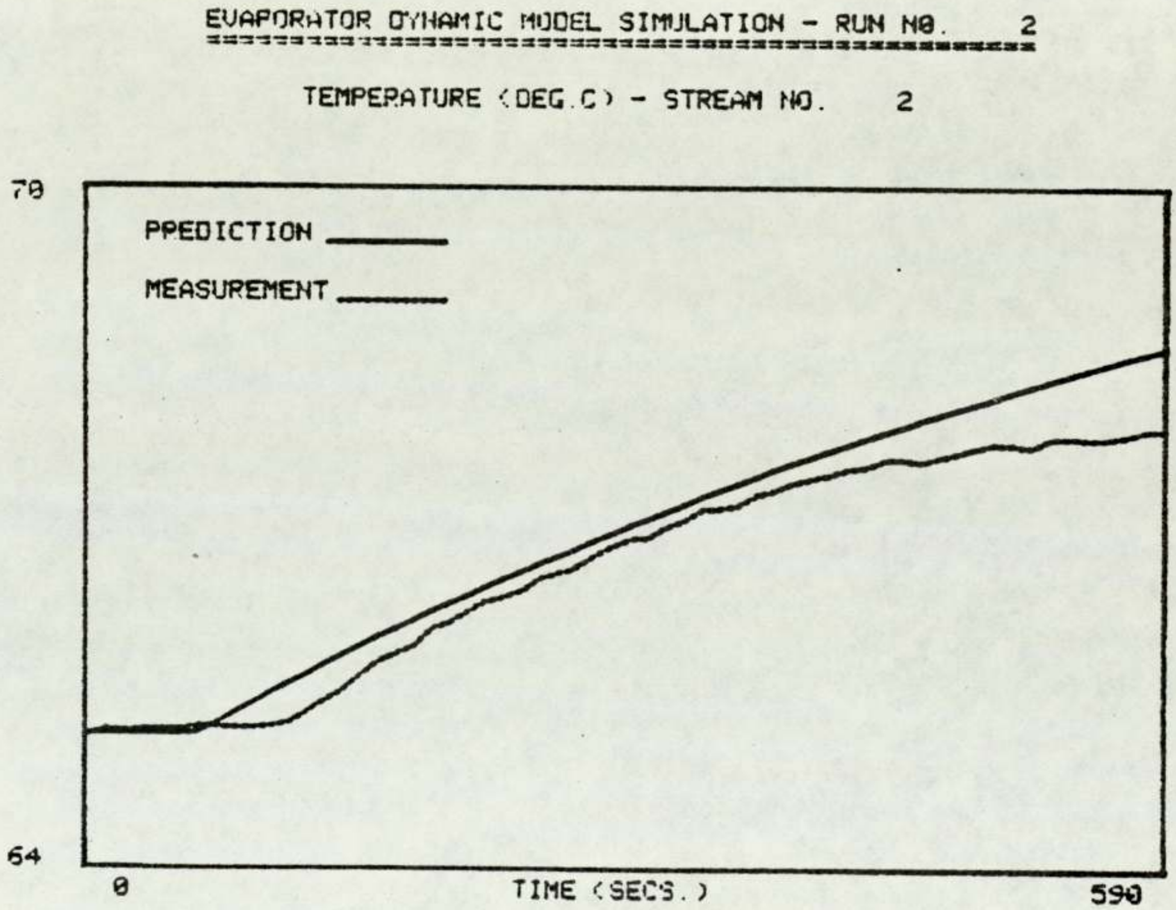


FIGURE J-5.1 - TEMPERATURE - EXIT PREHEATER TUBES

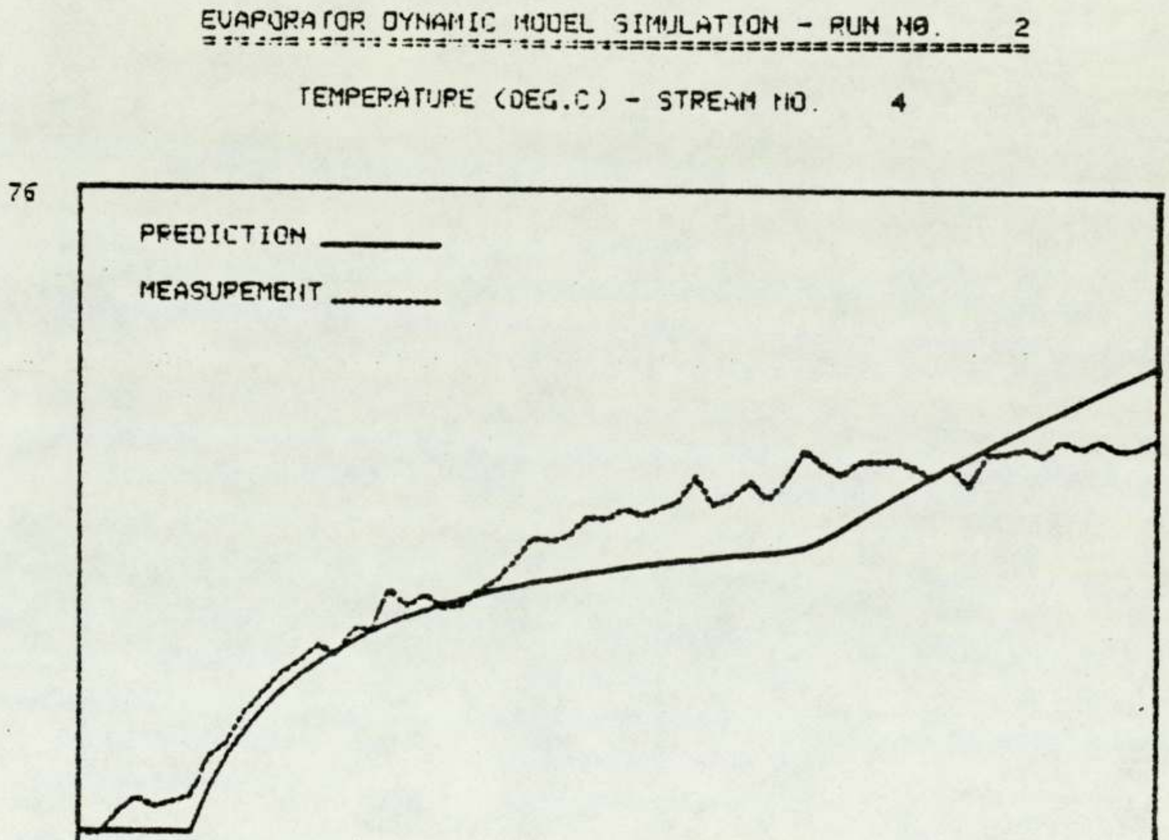


FIGURE J-5.2 - TEMPERATURE - EXIT PREHEATER SHELL

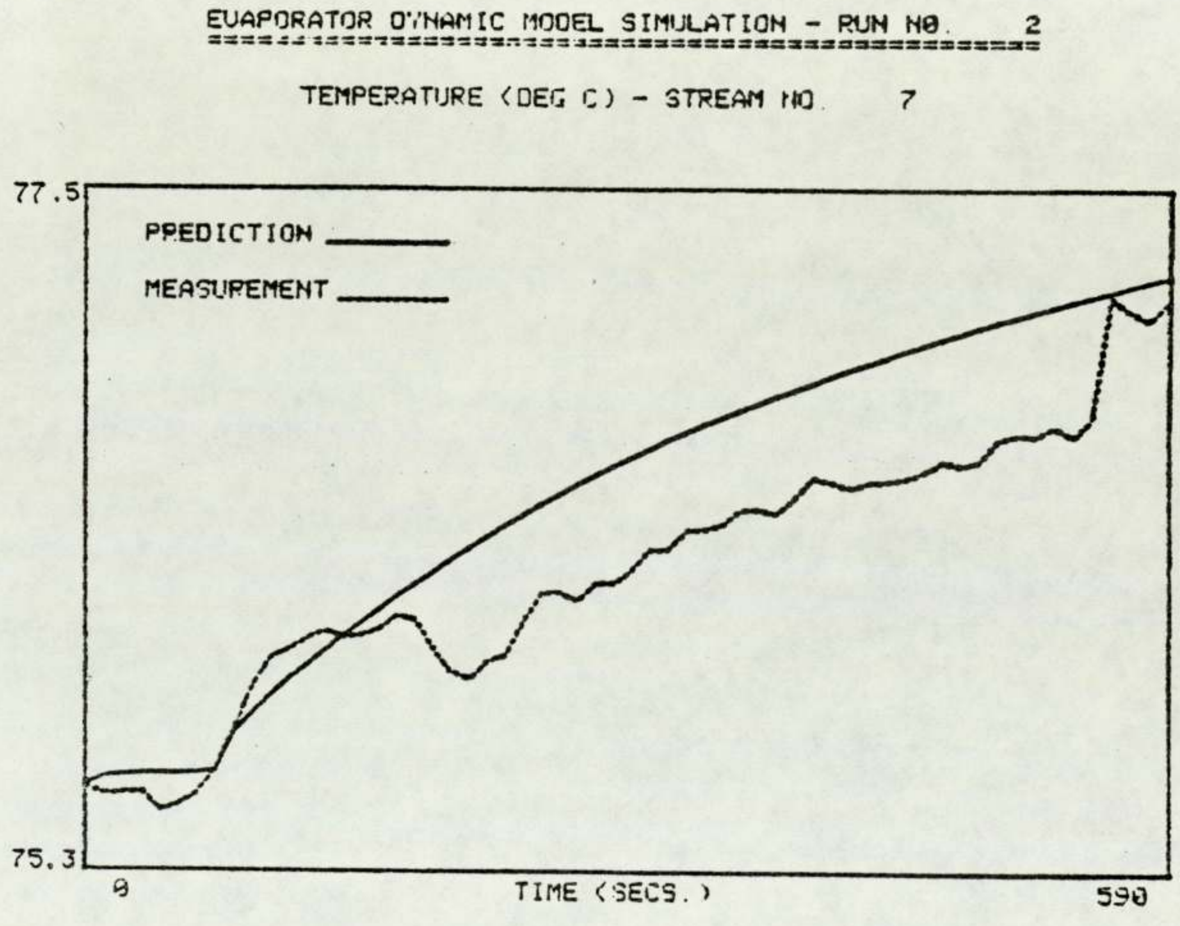


FIGURE J-5.3 - TEMPERATURE - EXIT FIRST EFFECT TUBES

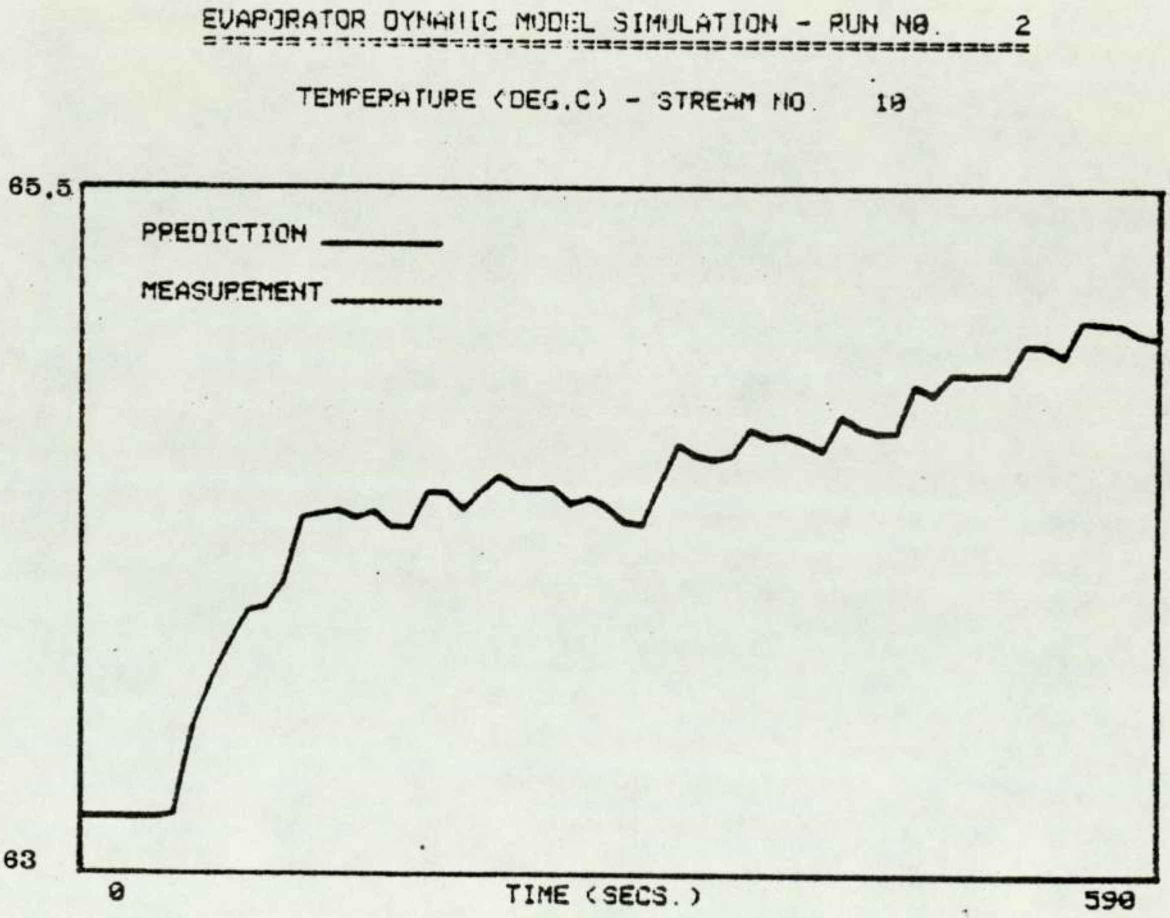


FIGURE J-5.4 - TEMPERATURE - EXIT SECOND EFFECT

CALANDRIA SHELL

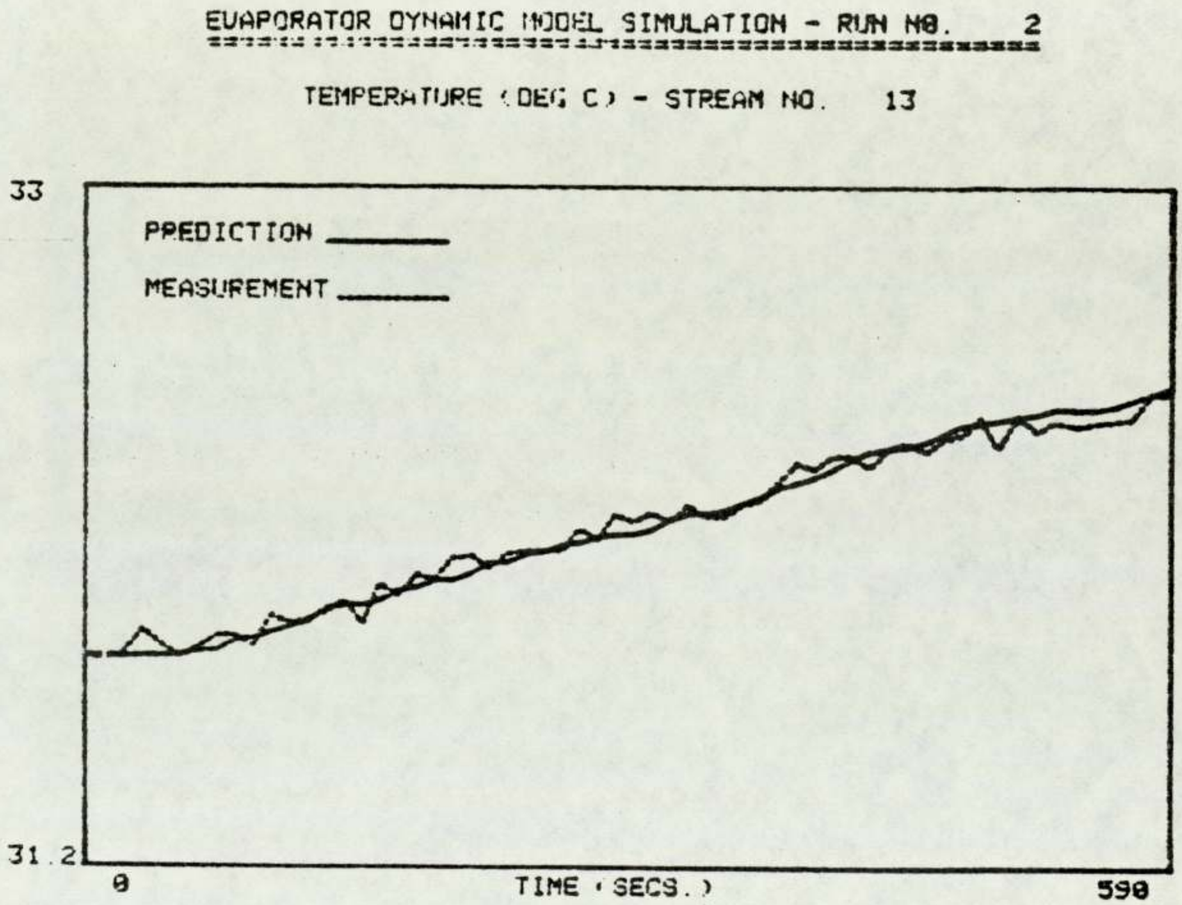


FIGURE J-5.5 - TEMPERATURE - EXIT CONDENSER TUBES

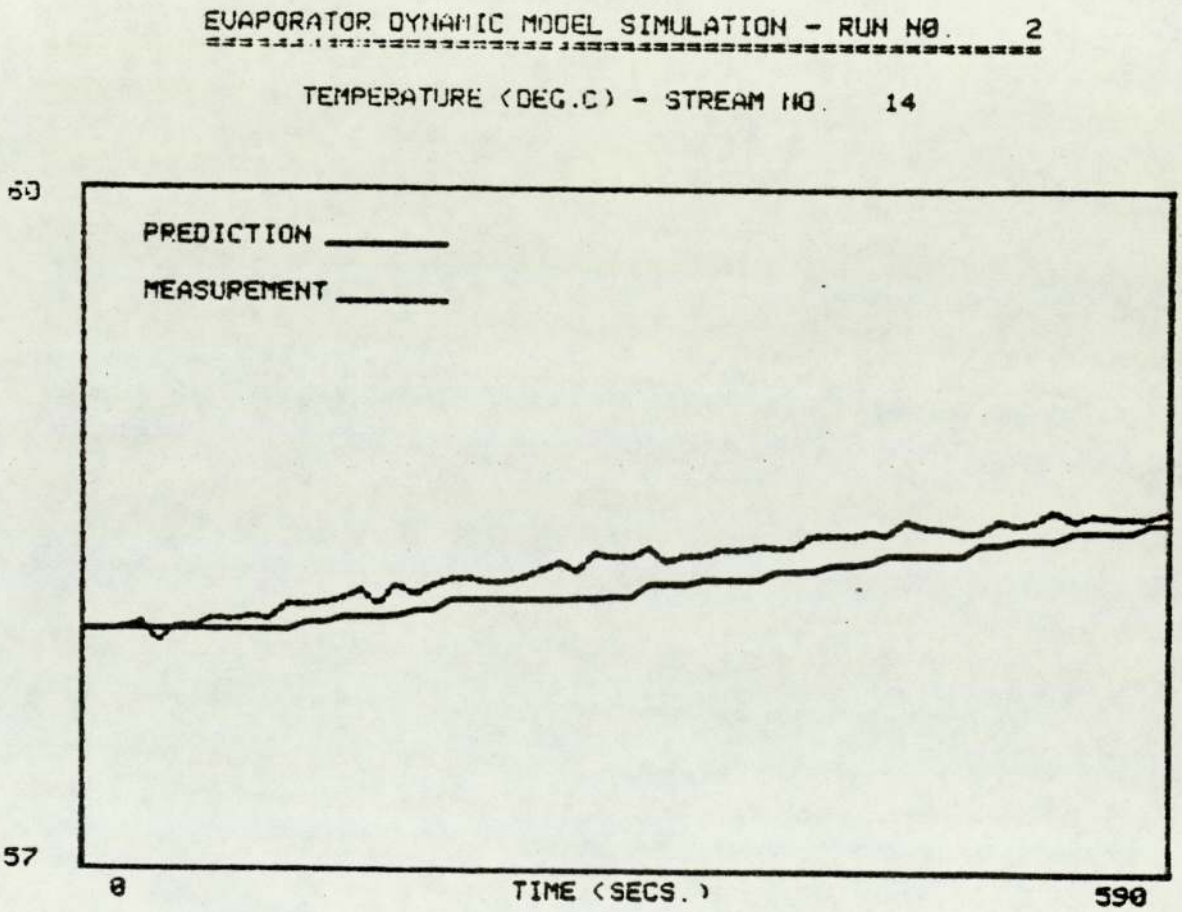


FIGURE J-5.6 - TEMPERATURE - EXIT SECOND EFFECT

CALANDRIA TUBES

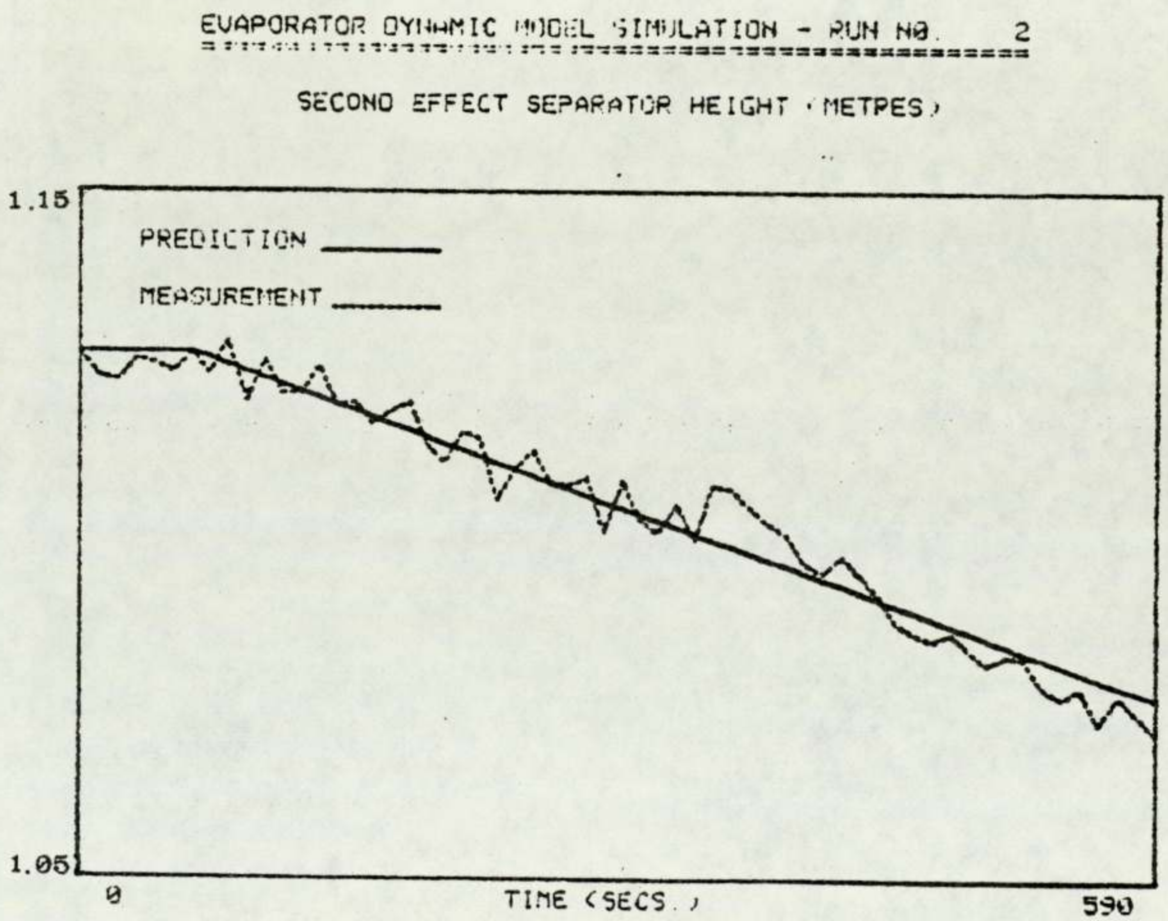
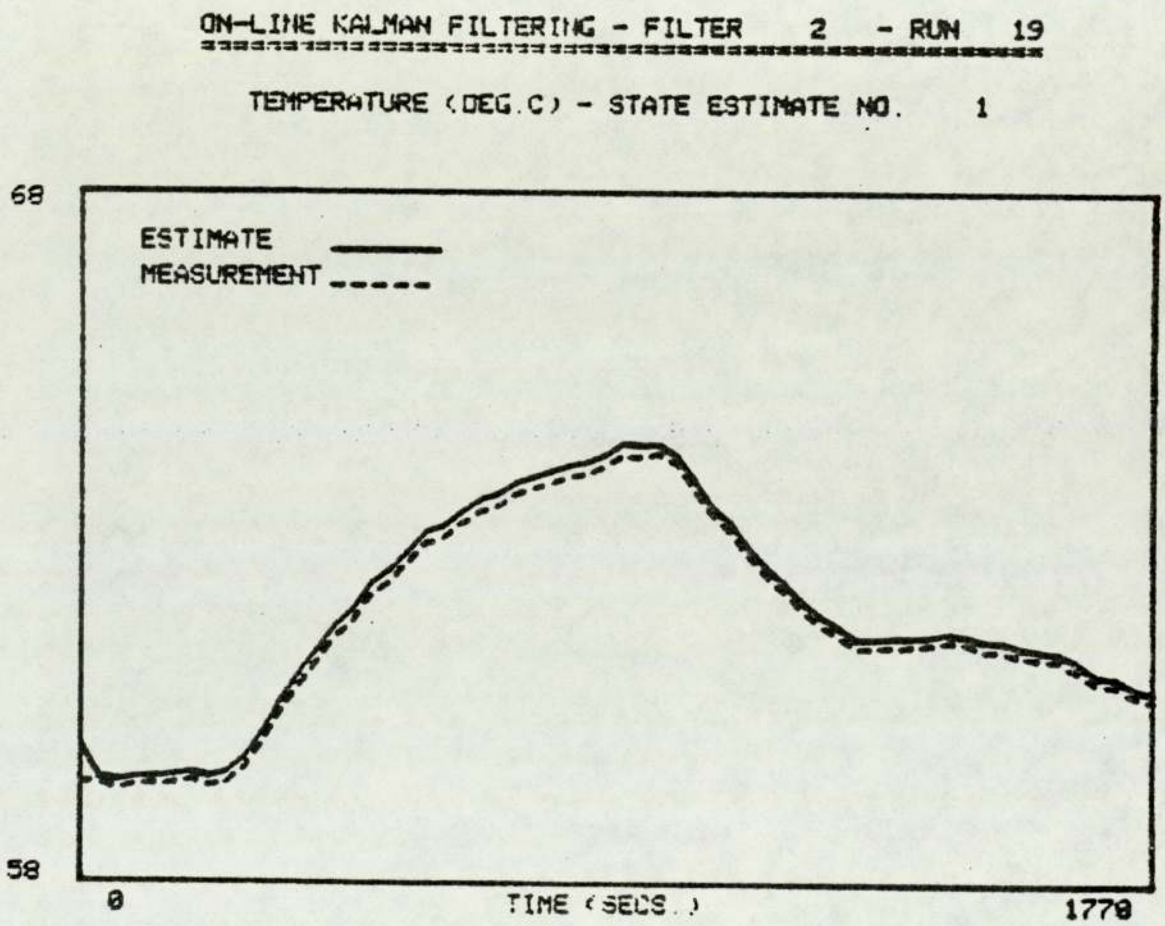

EVAPORATOR DYNAMIC MODEL SIMULATION - RUN NO. 2
SECOND EFFECT SEPARATOR HEIGHT (METRES)
1.15
PREDICTION
MEASUREMENT
1.05
0
TIME (SECS.)
590

FIGURE J-5.7

APPENDIX K

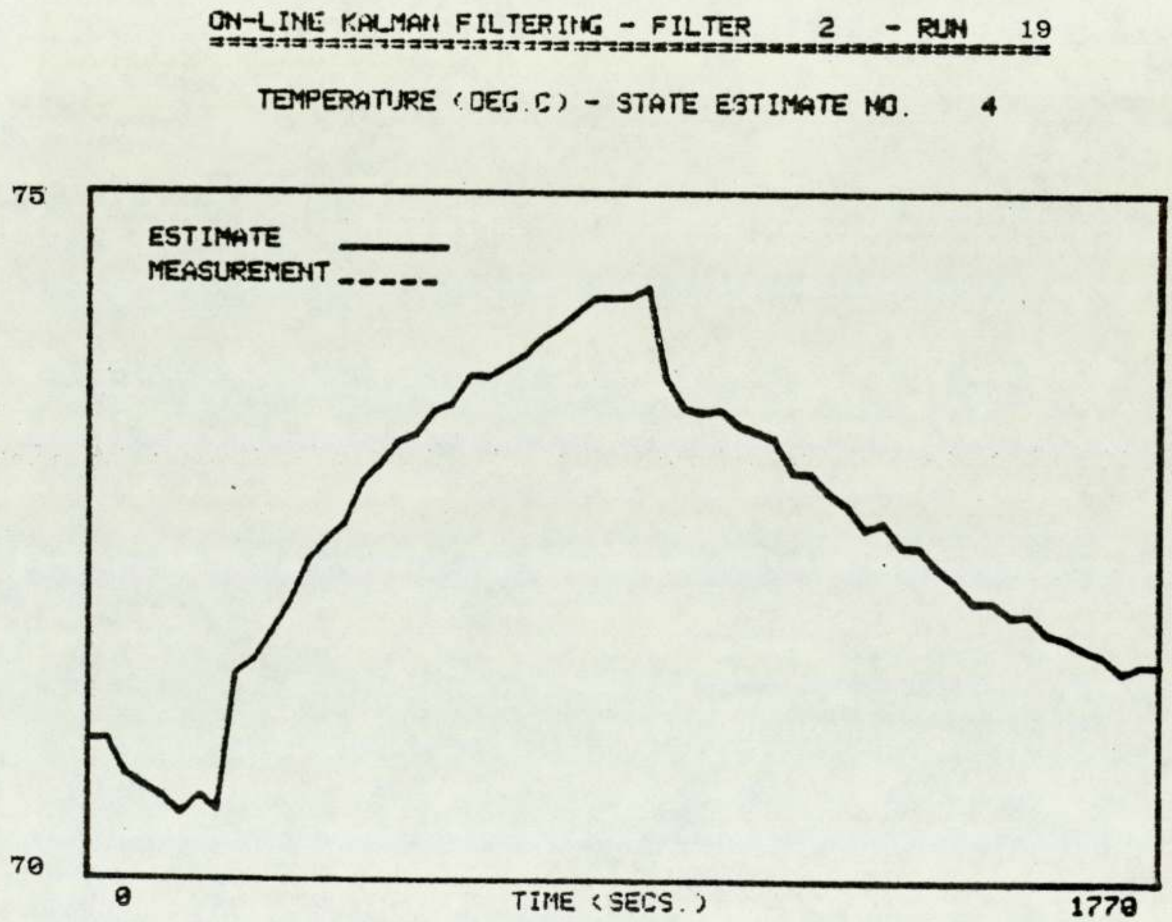
RESULTS OF ON-LINE FILTERING EXPERIMENTS

CONTENTS



Q=1.0*I

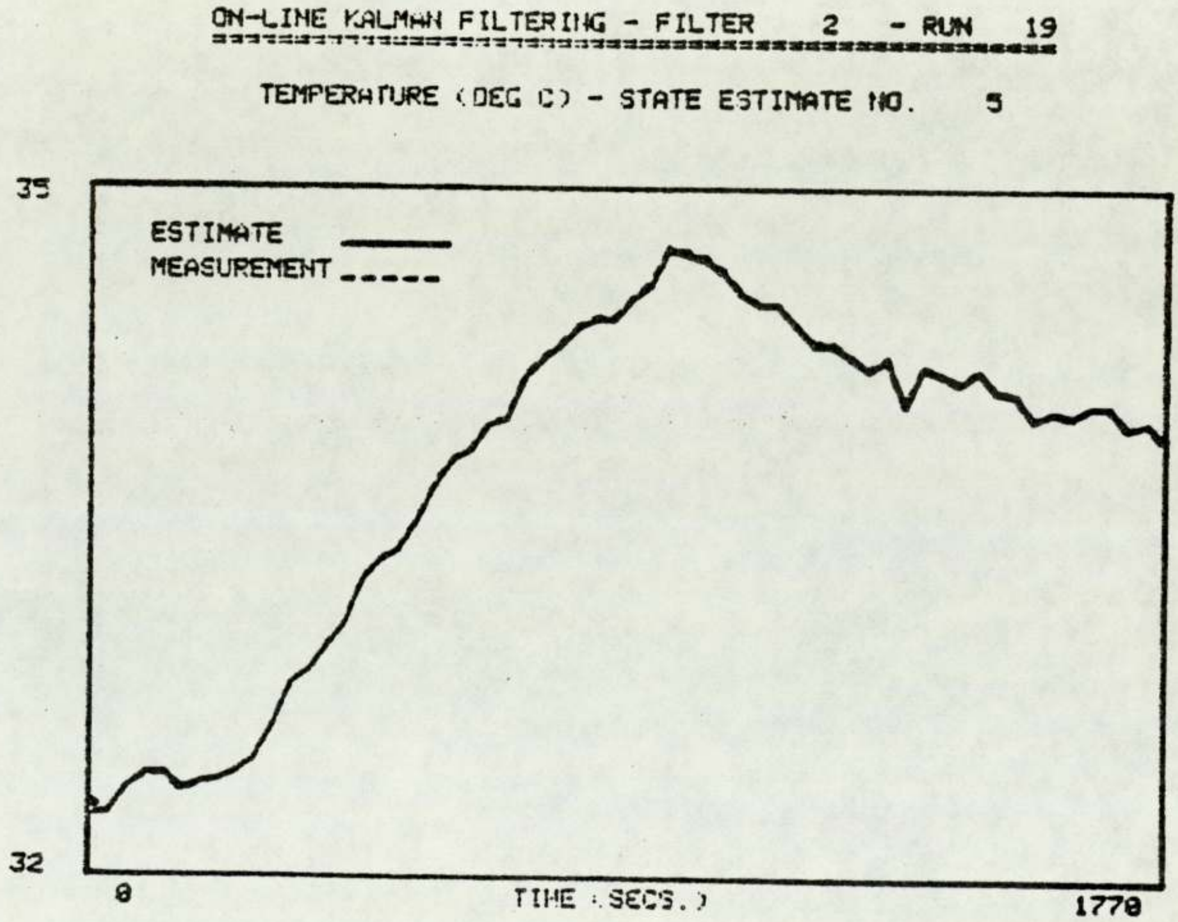
FIGURE K-1.1 - TEMPERATURE - EXIT PREHEATER TUBES



Q=1.0*I

FIGURE K-1.2 - TEMPERATURE - EXIT SECOND EFFECT

CALANDRIA SHELL

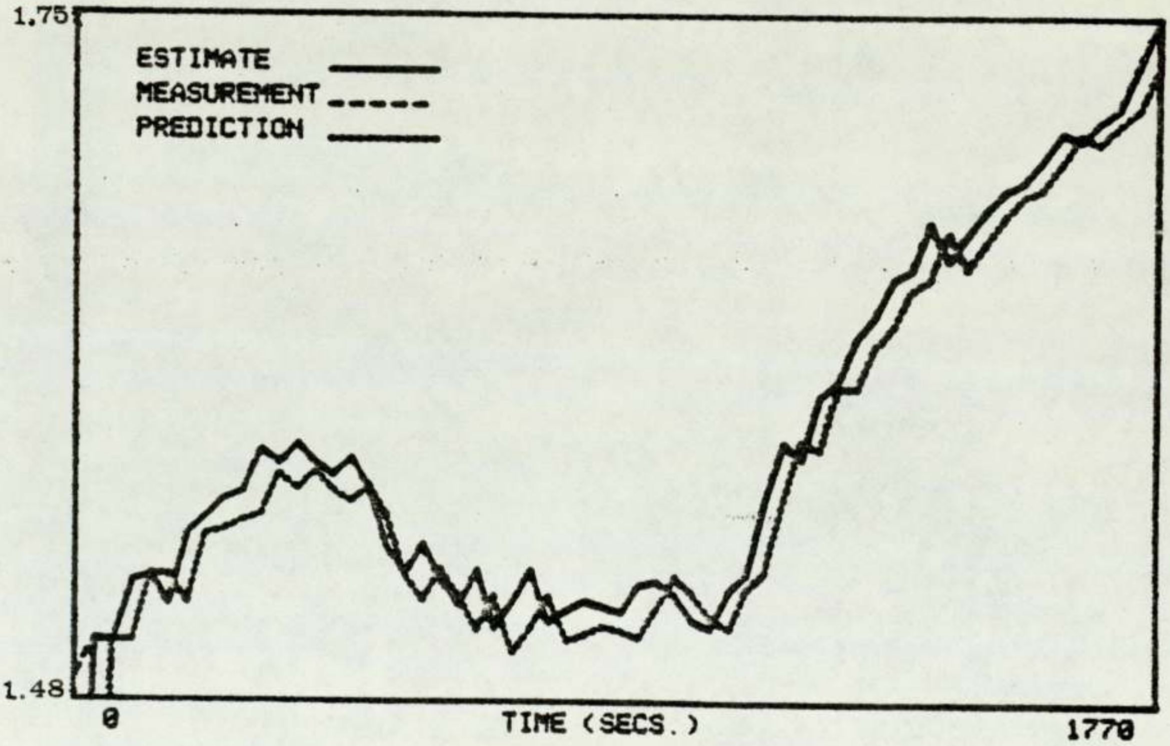


Q=1.0*I

FIGURE K-1.3 - TEMPERATURE - EXIT CONDENSER TUBES

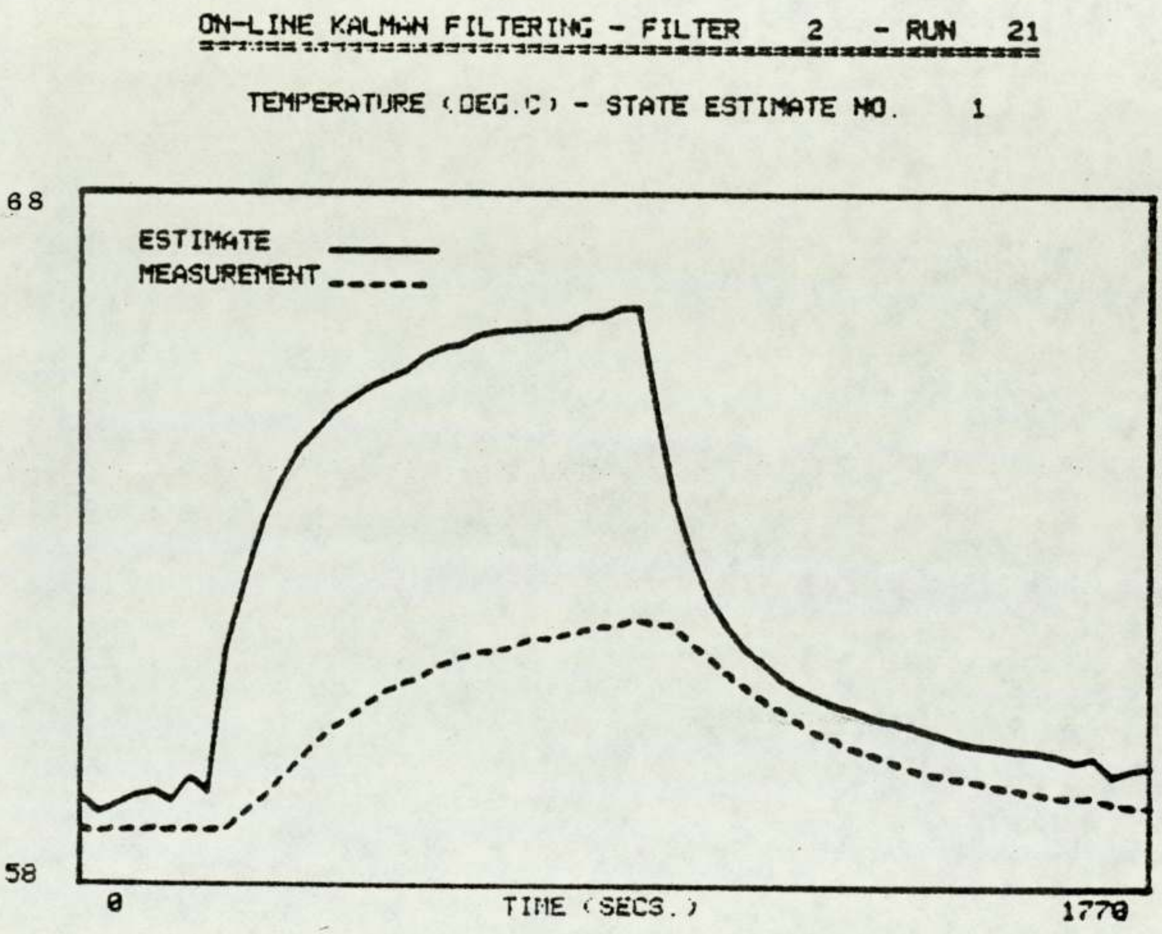
ON-LINE KALMAN FILTERING - FILTER 2 - RUN 19

SECOND EFFECT SEPARATOR HEIGHT (METRES)



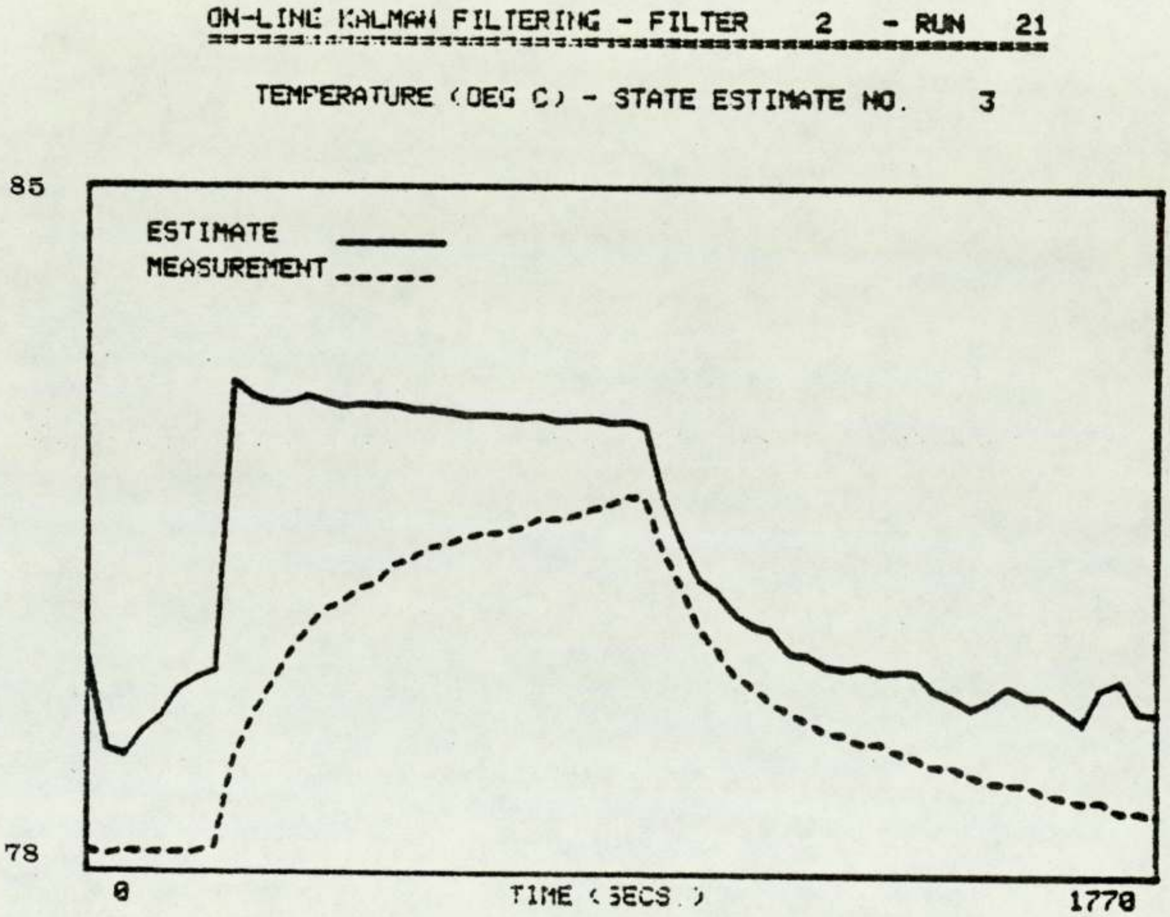
Q=1.O*I

FIGURE K-1.4



Q=0.1*I

FIGURE K-1.5 - TEMPERATURE - EXIT PREHEATER TUBES



Q=0.1*I

FIGURE K-1.6 - TEMPERATURE - EXIT FIRST EFFECT TUBES

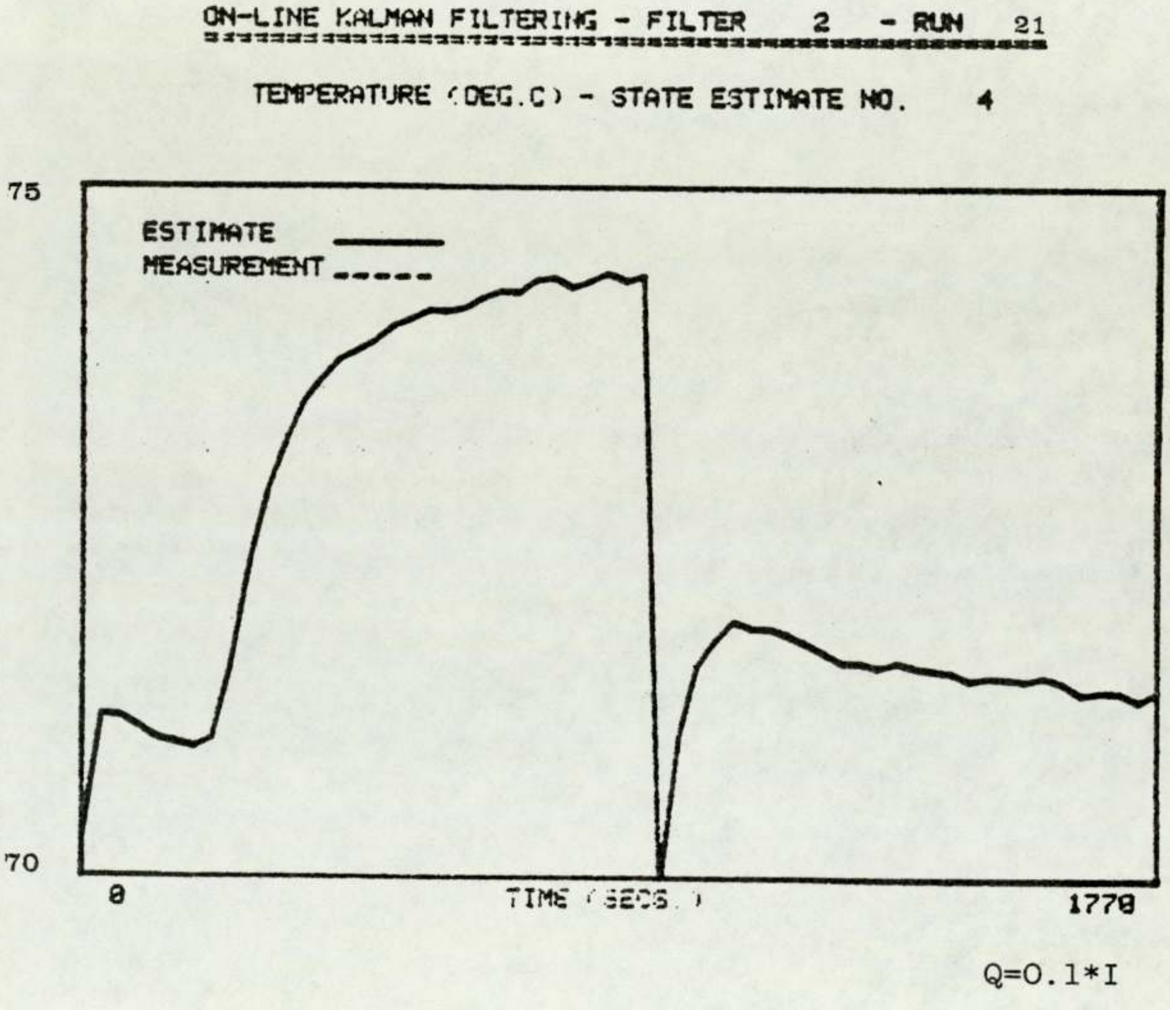
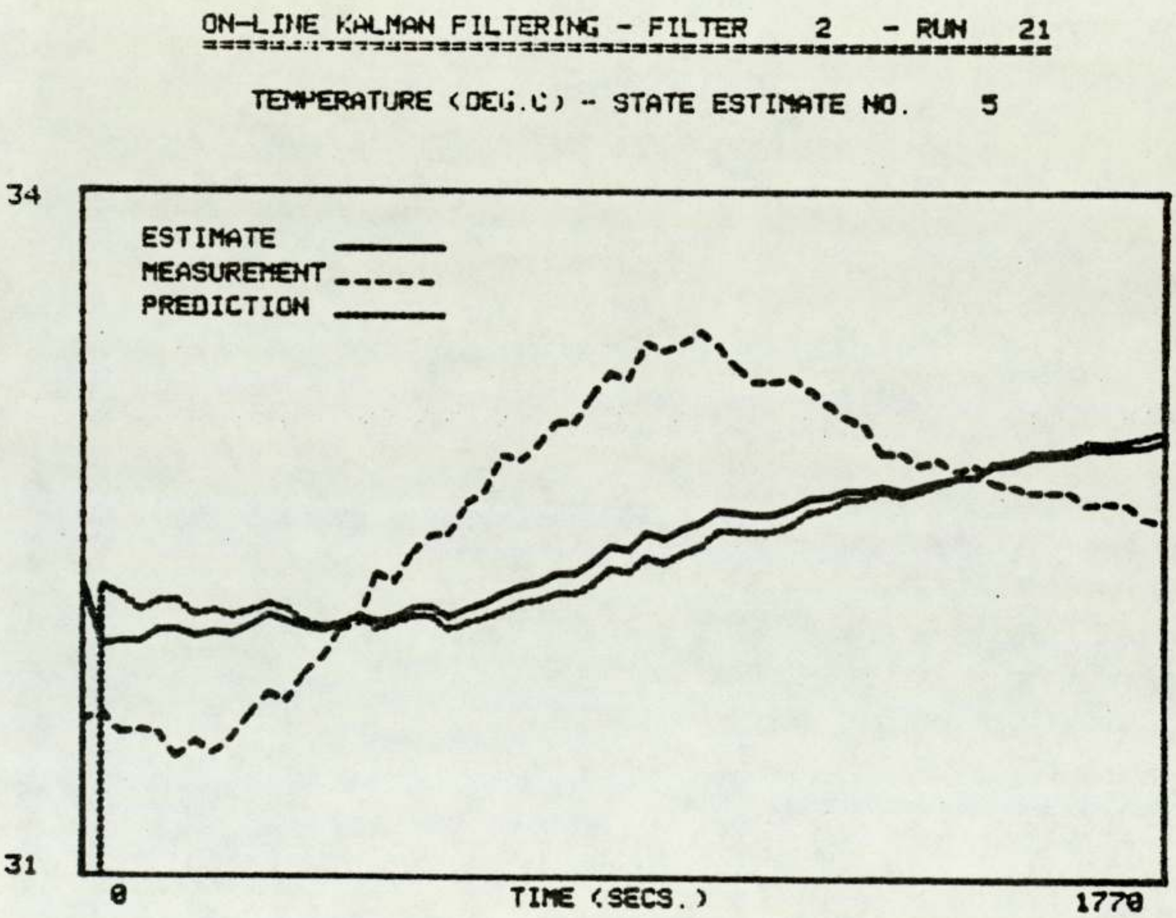


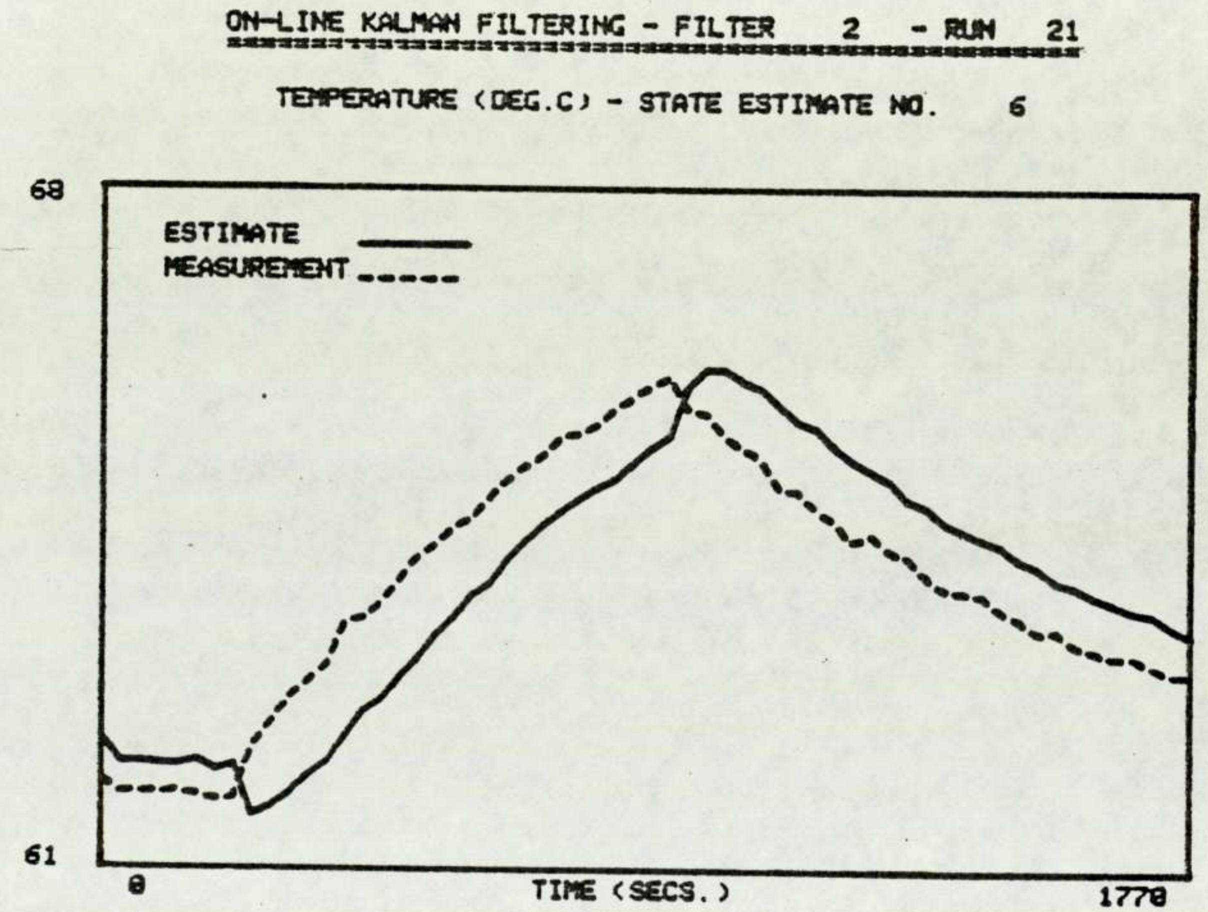
FIGURE K-1.7 - TEMPERATURE - EXIT SECOND EFFECT

CALANDRIA SHELL



Q=0.1*I

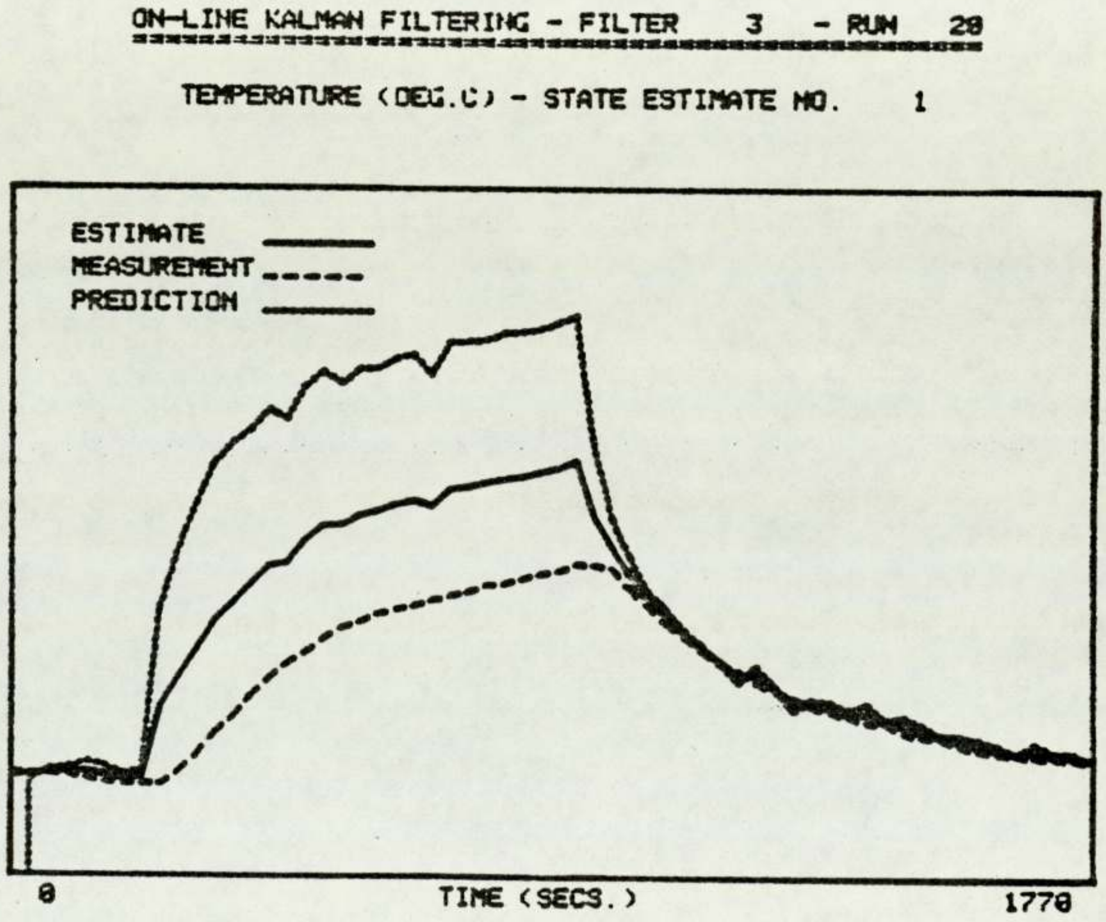
FIGURE K-1.8 - TEMPERATURE - EXIT CONDENSER TUBES



Q=0.1*I

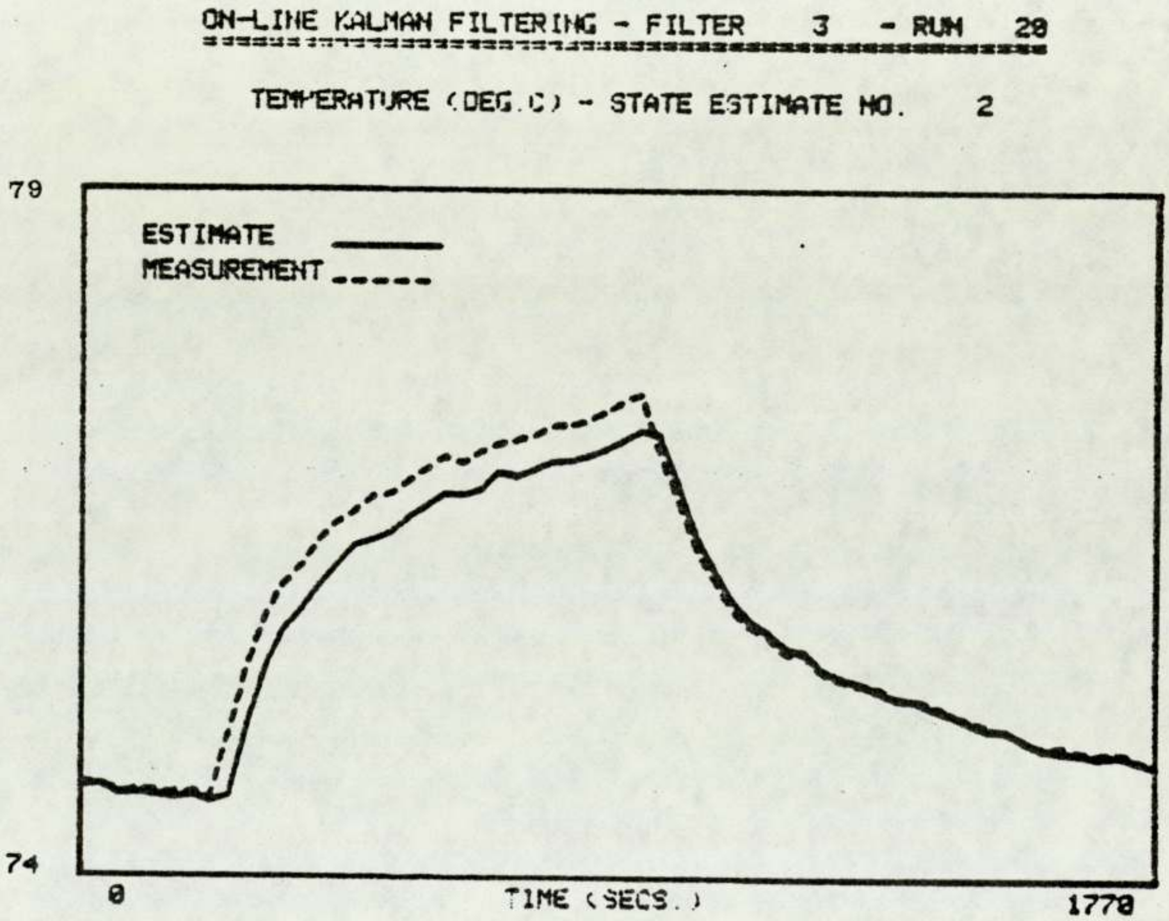
FIGURE K-1.9 - TEMPERATURE - EXIT SECOND EFFECT

CALANDRIA TUBES



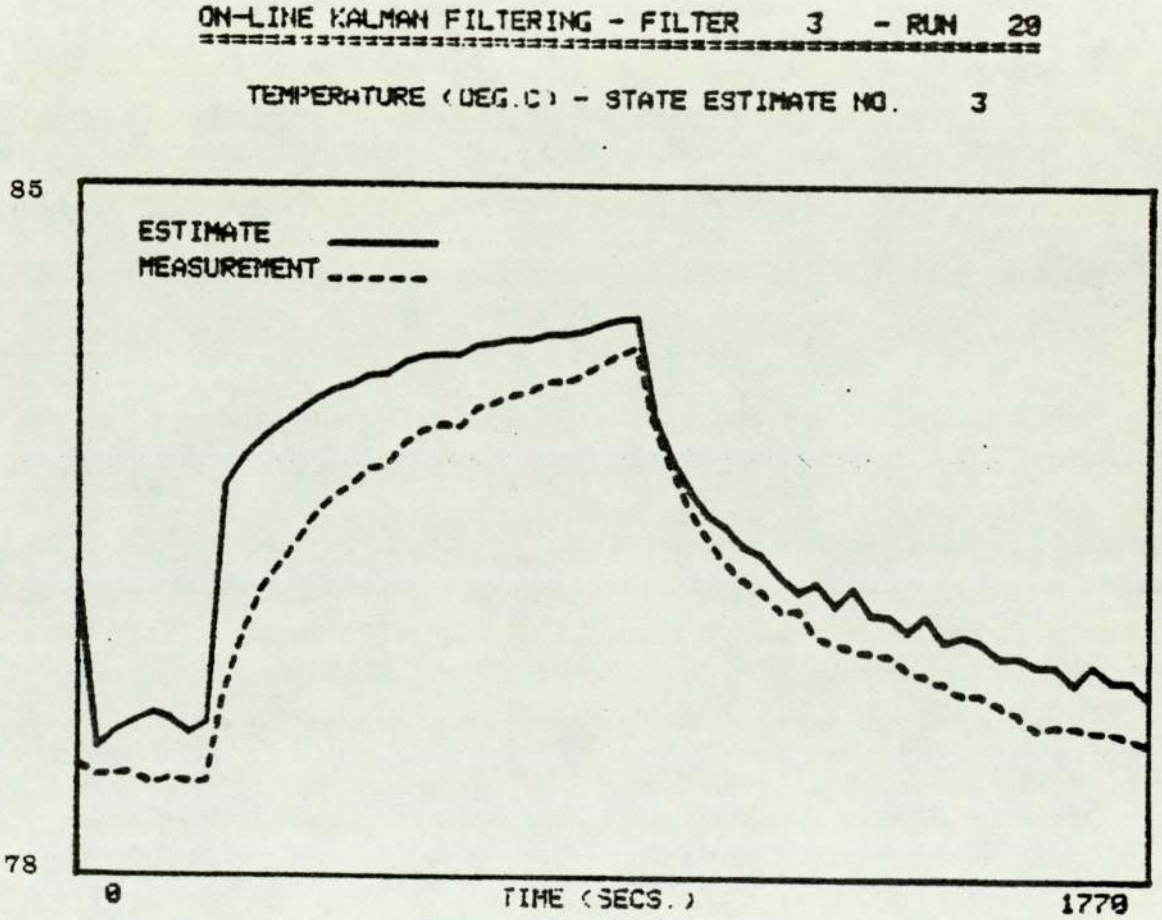
Q=0.1*I

FIGURE K-2.1 - TEMPERATURE - EXIT PREHEATER TUBES



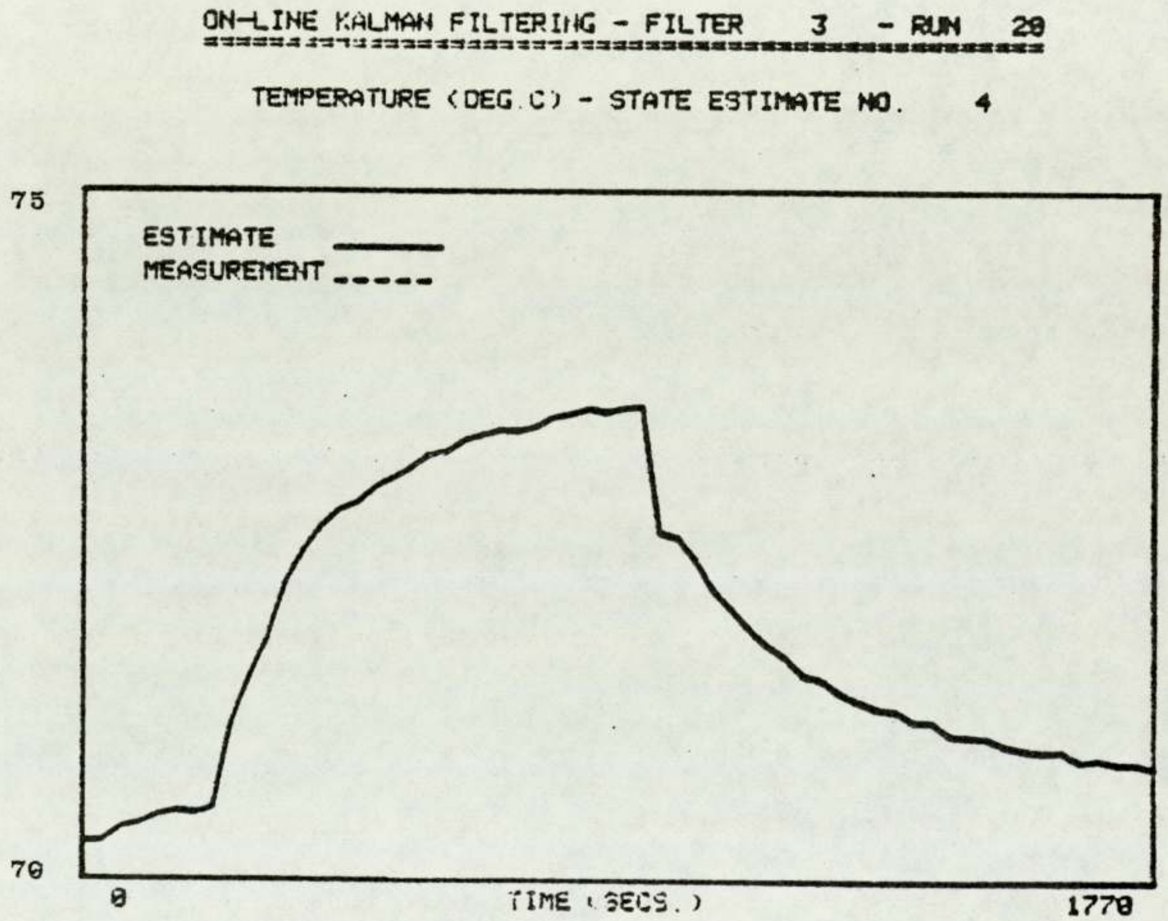
Q=0.1*I

FIGURE K-2.2 - TEMPERATURE - EXIT PREHEATER SHELL



Q=0.1*I

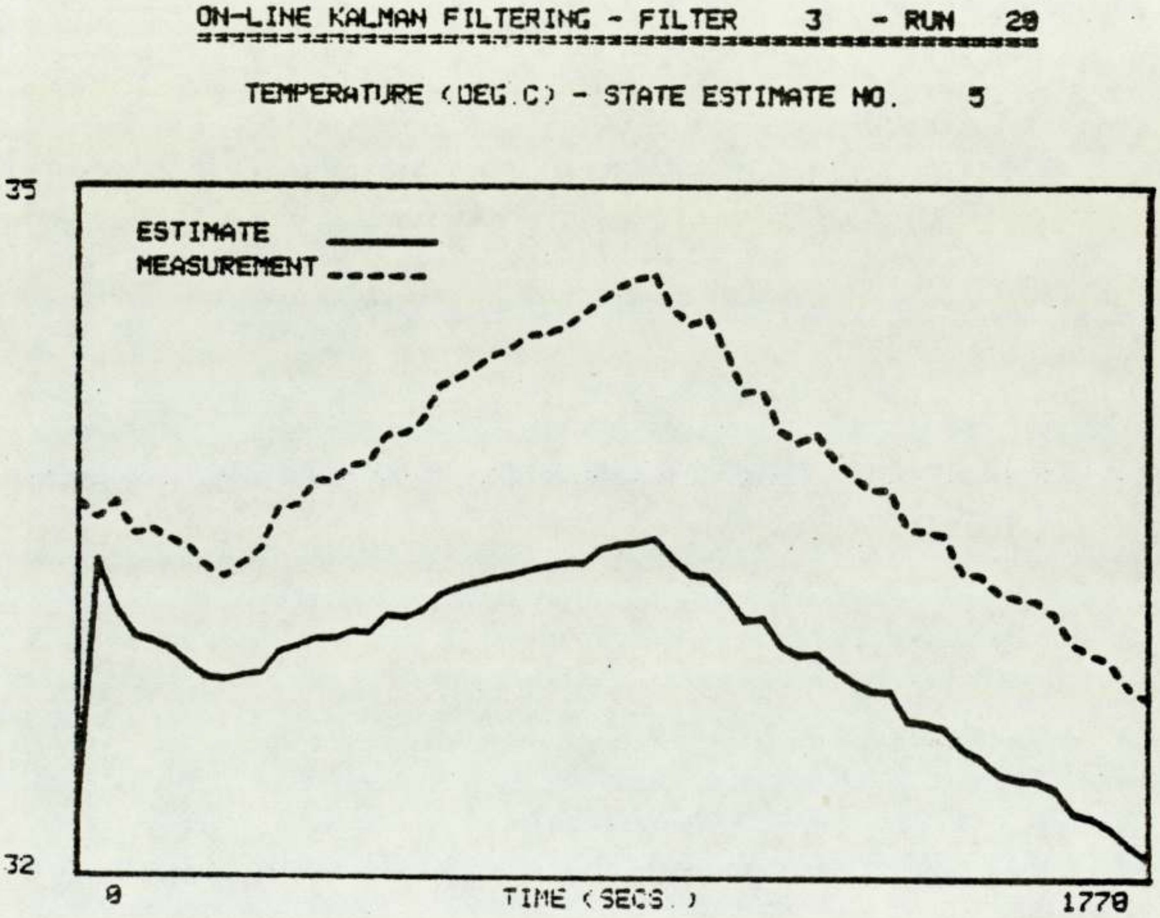
FIGURE K-2.3 - TEMPERATURE - EXIT FIRST EFFECT TUBES



Q=0.1*I

FIGURE K-2.4 - TEMPERATURE - EXIT SECOND EFFECT

CALANDRIA SHELL



Q=0.1*I

FIGURE K-2.5 - TEMPERATURE - EXIT CONDENSER SHELL

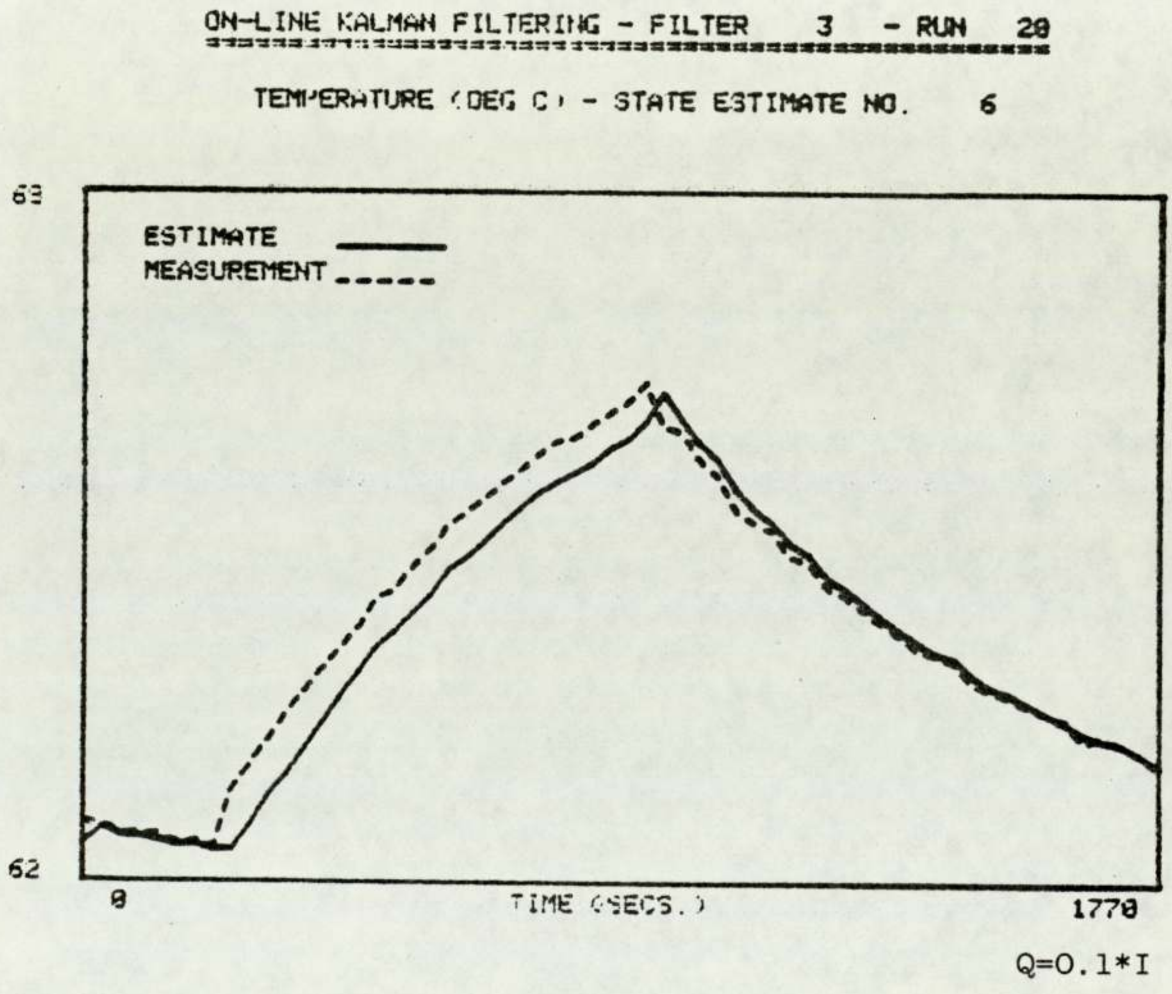
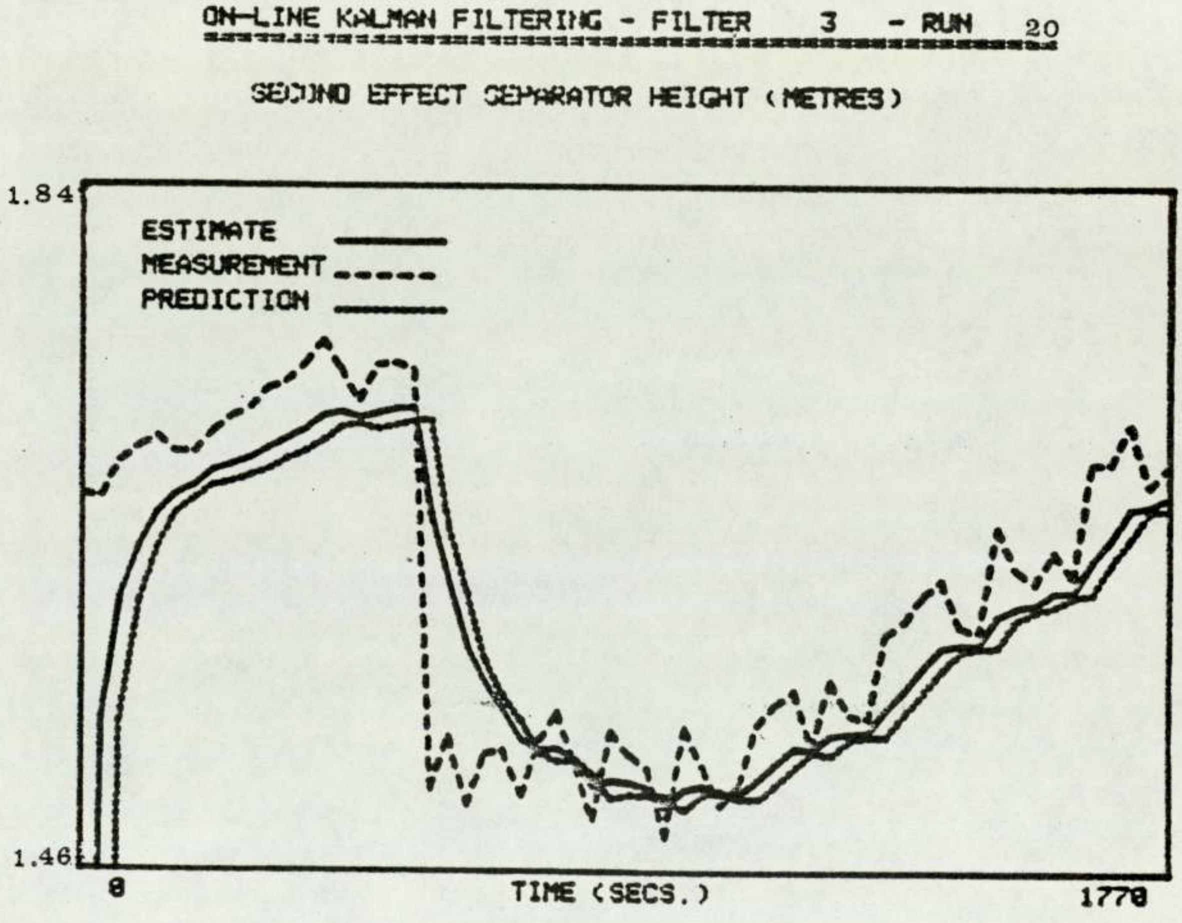


FIGURE K-2.6 - TEMPERATURE - EXIT SECOND EFFECT CALANDRIA TUBES



Q=0.1*I

FIGURE K-2.7

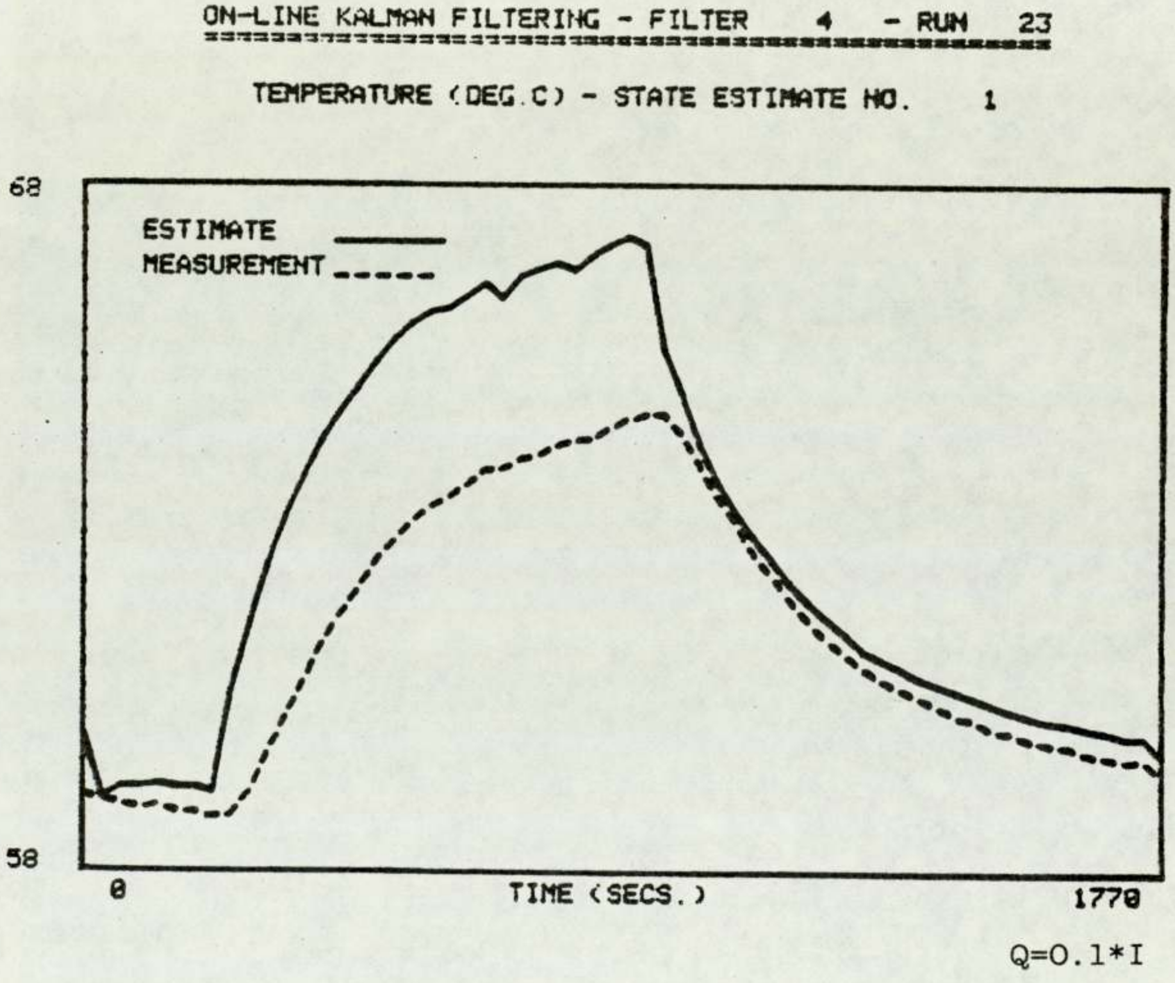
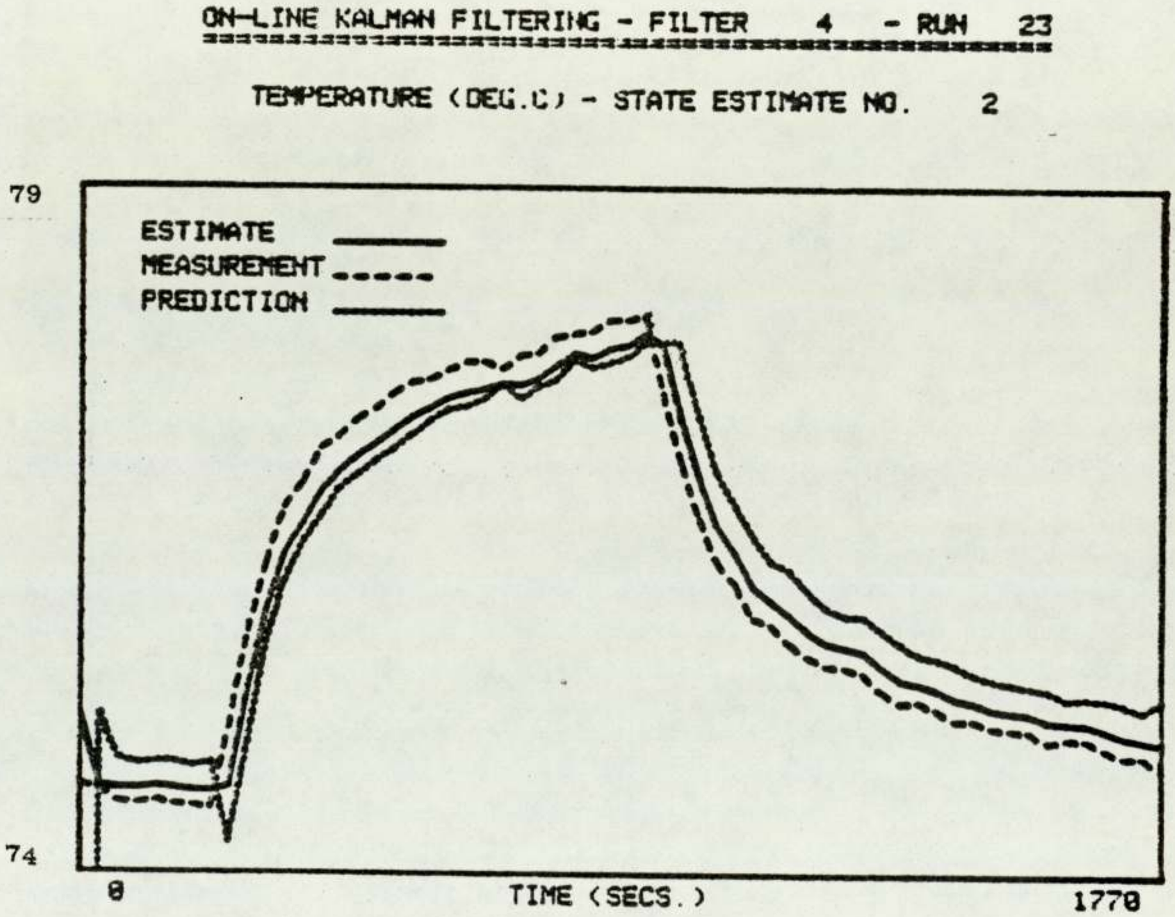
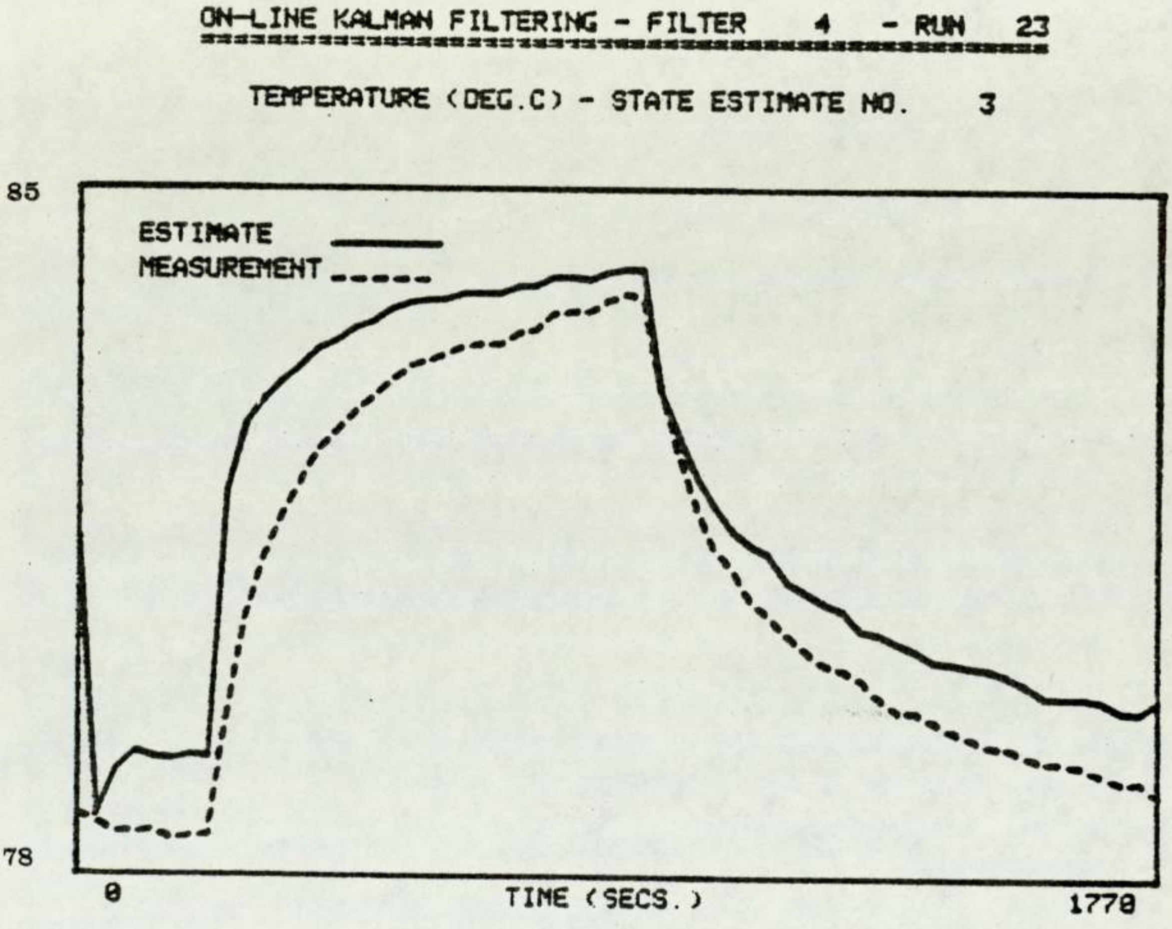


FIGURE K-3.1 - TEMPERATURE - EXIT PREHEATER TUBES



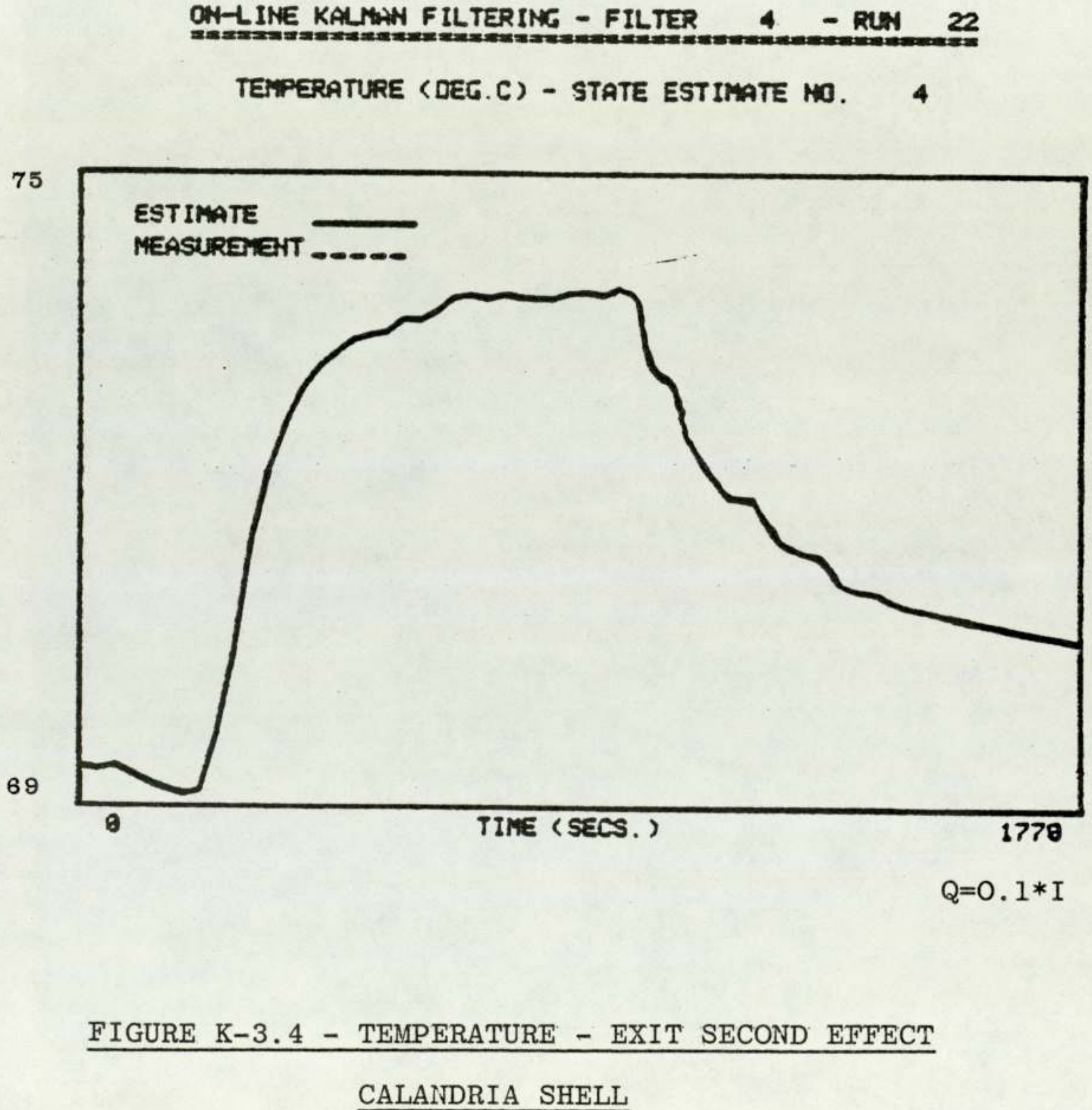
Q=0.1*I

FIGURE K-3.2 - TEMPERATURE - EXIT PREHEATER SHELL



Q=0.1*I

FIGURE K-3.3 - TEMPERATURE - EXIT FIRST EFFECT TUBES



Q=0.1*I

FIGURE K-3.4 - TEMPERATURE - EXIT SECOND EFFECT

CALANDRIA SHELL

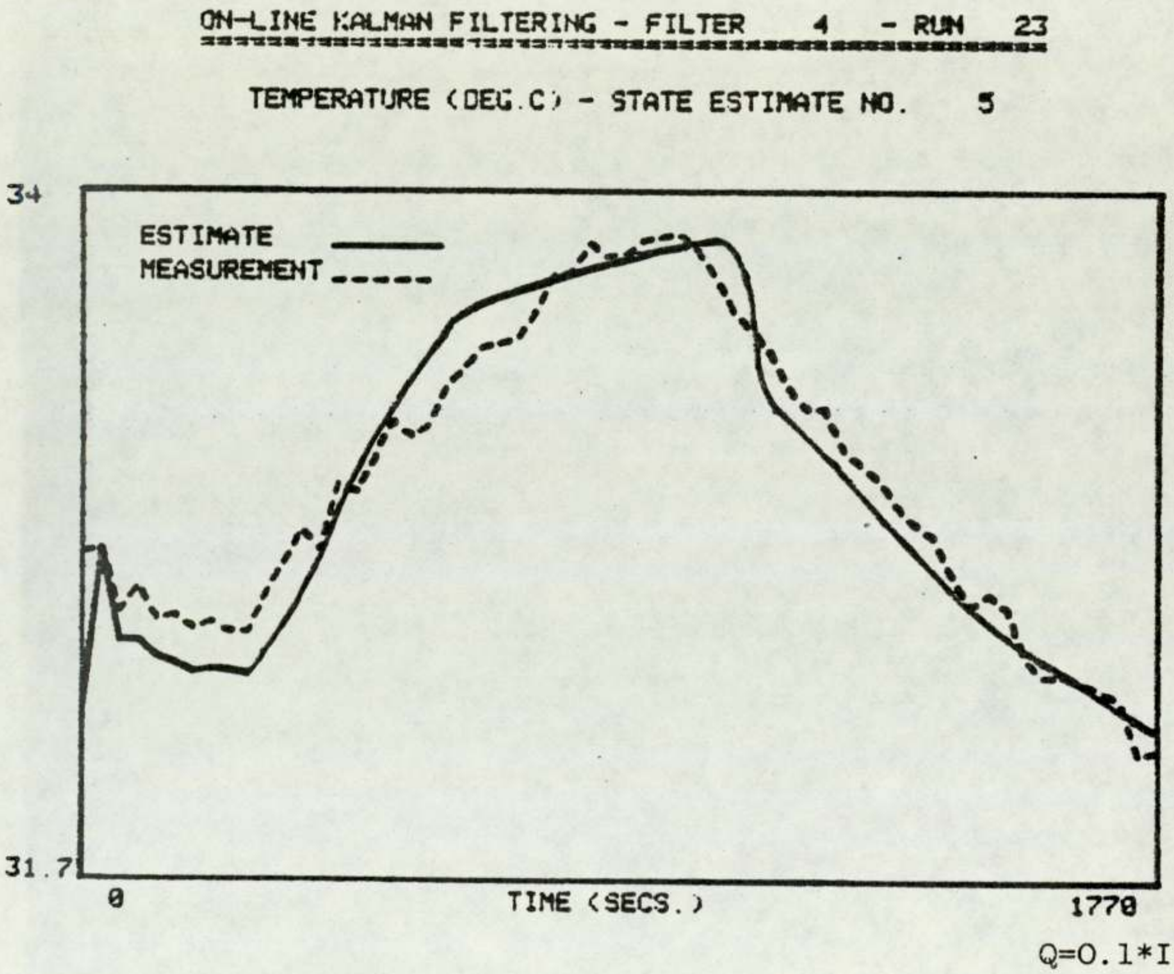
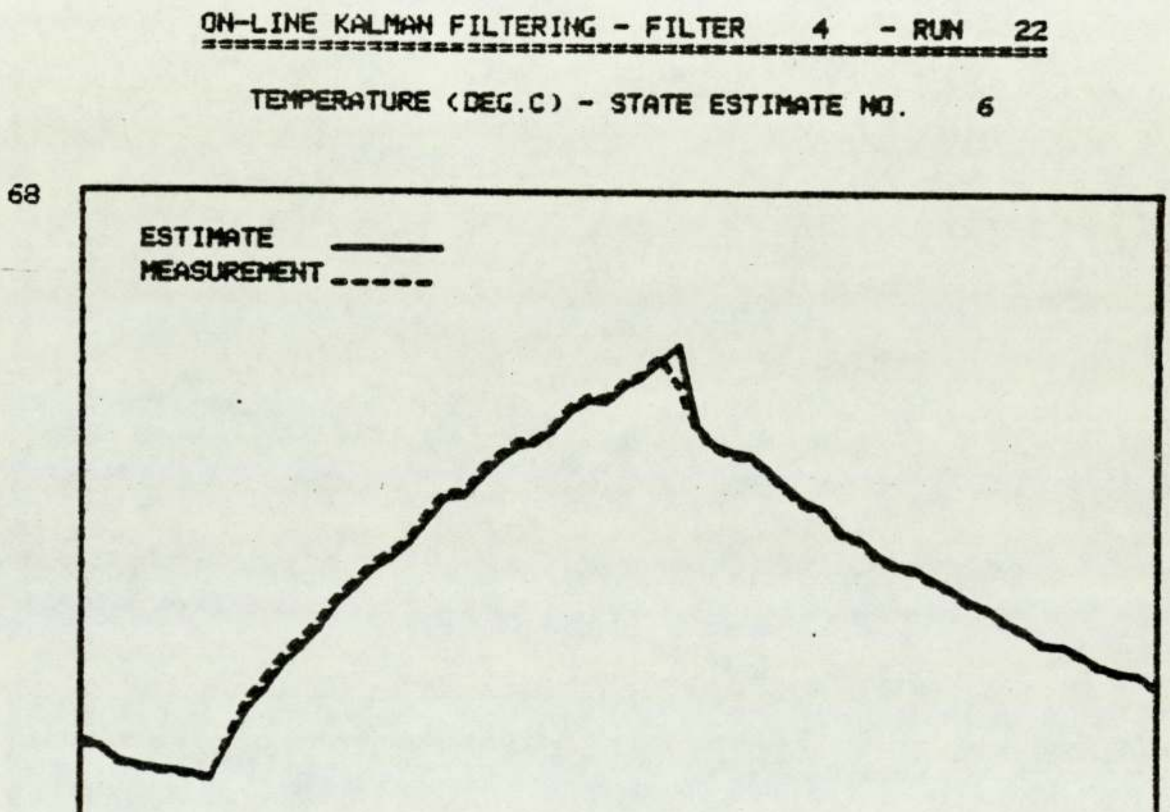
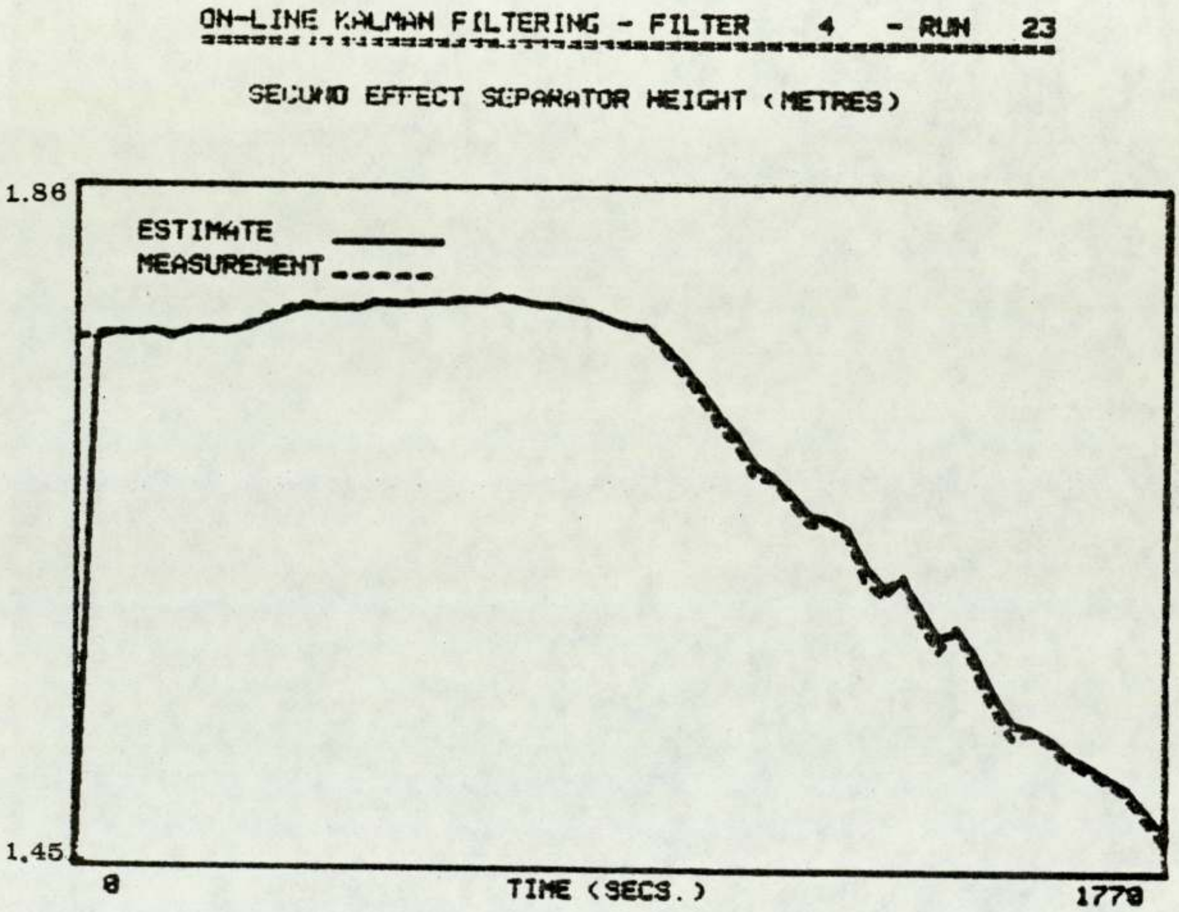


FIGURE K-3.5 - TEMPERATURE - EXIT CONDENSER TUBES



Q=0.1*I

FIGURE K-3.6 - TEMPERATURE - EXIT SECOND EFFECT CALANDRIA TUBES



Q=0.1*I

FIGURE K-3.7

BIBLIOGRAPHY

REFERENCES FOR CHAPTER 1

1.1 Kalman, R.E., J. Basic. Eng., Trans. ASME, 82D, 33, 1960.

REFERENCES FOR CHAPTER 2

2.1 Legendre, A.M., "Nouvelles Methods pour la Determination des Orbites des Cometes.", Paris, 1806.
2.2 Gauss, K.F., "Theory of the motion of the Heavenly Bodies Moving about the Sun in Conic Sections.", Dover, New York, 1962.
2.3 Deutsch, R., "Estimation Theory.", Prentice-Hall, New Jersey, 1959.
2.4 Pearson, K., "Contributions to the Mathematical Theory of Evolution.", Philos. Trans. Royal. Soc., London, 185, 71, 1894.
2.5 Fisher, R.A., "Theory of Statistical Estimation.", Proc. Camb. Philos. Soc., 22, 760, 1925.
2.6 Swerling, P., J. Astronomical Science, 6, 46, 1959.
2.7 Wiener, N., "The Extrapolation, Interpolation and Smoothing of Stationary Time Series.", J. Wiley and Sons, New York, 1949.
2.8 Kolmogorov, A.N., "Interpolation and Extrapolation of Stationary Random Sequences", Translated by W. Doyle and J. Selin, RM-3090-PR, The Rand Comp., U.S.A., 1962.
2.9 Astrom, K.J., "System Identification, a Survey", IFAC Symp., Prague, 1970.
2.10 Nieman, R.E., Fisher, D.G. and Seborg, D.E., Int. J. Cont., 18 2, 209, 1971.
2.11 Sienfeld, J.H., Ind. Eng. Chem., 62 132, 1970.
2.12 Cuenod, M. and Sage, A., Automatica, 4, 235, 1968.
2.13 See reference 1.1.
2.14 Kalman, R.E. and Bucy, R.S., J. Basic. Eng., Trans. ASME, 83D, 33, 1961.
2.15 Kalman, R.E., Proc. 1st Symp. Appl. Random Function Theory and Probability, (J.L. Boganoff and F. Kozin, eds), J. Wiley, New York, 1963.
2.16 Sorenson, H.W., "Advances in Control Systems.", (C.T. Leondres, ed), 3, 219, 1967.
2.17 Coggan, G.C., and Noton, A.R.H., Trans. Inst. Chem. Engnrs., 48, T245, 1970.
2.18 Storey, B., M.Sc. Thesis, University of Aston in Birmingham, 1973.
2.19 Aoki, M., "Optimisation of Stochastic Systems", Academic Press, New York, 1967.
2.20 Cox, H., Proc. 1964 Joint Automatic Control Conf., Stanford, U.S.A., p376.
2.21 Kushner, H.J., SIAM J. Control, 2, 106, 1964.

2.22 Friedland, B. and Bernstein, I., J. Franklin Inst., 281, 455, 1966.
2.23 Jazwinski, A.H., "Stochastic Processes and Filtering Theory", Academic Press, New York, 1970.
2.24 Alspach, D.L., Proc. 14th Midwest Symposium on Circuit Theory, 1971, Denver, U.S.A., p.8.1-1.
2.25 Bucy, R.S. and Senne, K.D., "Realisation of Optimum Discrete-Time, Non-linear Estimators.", Proc. of the Symposium on Non-Linear Estimation Theory and its Applications, San Diego, U.S.A., 1970.
2.26 Alspach, D.L. and Sorenson, H.W., "Approximation of Density Functions by a Sum of Gaussians for Non-Linear Bayesian Estimation.", Proc. Symposium on Non-Linear Estimation Theory and its Applications, San Diego, U.S.A., 1970.
2.27 Kushner, H.J., SIAM Control, 2, 1964, 106.
2.28 Athans, M., Wishner, R.P. and Bertolini, A., IEEE Trans. Automatic Control, AC13, 5, 504, 1968.
2.29 Tse, E., Proc. 3rd Symposium on Non-Linear Estimation Theory, San Diego, U.S.A., 1972.
2.30 Wishner, R.P., Tabaczynki, J.A. and Athans, M., Automatica, 5, 487, 1969.
2.31 Denham, W.F. and Pines, S., AIAA J., 4, 1071, 1966.
2.32 Leung, V.P. and Padmanabhan, L., The Chem. Engineering J., 5, 487, 1969.
2.33 Jazwinski, A.H., "Non-Linear Filtering - Numerical Experiments" NASA - Goddard Astrodynamics Conf., Maryland, U.S.A., April 1966.
2.34 Wishner, R.P., Lawson, R.F. and Athans, M., Proc. Symp. Non-Linear Estimation Theory and Applications, San Diego, U.S.A., 1970.
2.35 Schwartz, L. and Stear, E.B., IEEE Trans. Auto. Control, AC-13, 83, 1968.
2.36 Smith, G.L., Schmidt, S.F. and McGee, L.A., NASA-TR R-35, 1962.
2.37 Smith, G.L., SIAM J. Control, 2, 19, 1964.
2.38 Scmidt, S.F., Proc. 1962 Joint Automatic Control Conf., U.S.A., 1962.
2.39 Ford-Levine, C., Proc. 6th Hawaii Int. Conf. on Systems Sci., Honolulu, Hawaii, Jan 1973.
2.40 Szeto, M.W., Transp. Sci., 6, 419, 1972.
2.41 Wheelwright, S.W., Operations Res. Quarterly, 24, 55, 1973.
2.42 Noton, A.R.M. and Choquette, P., Proc. IFAC/IFIP Symp., Toronto, 1968.
2.43 Choquette, P., Noton, A.R.M. and Watson, C.A.G., Proc. IEEE, 58, 1, 10, 1970.
2.44 Schmidt, S.F., "Computational Techniques in Kalman Filtering", in "Theory and Applications of Kalman Filtering", ed. C.T. Leondres, NATO-AGARD, 1970.

2.45 Coggan, G.C. and Wilson, J.A., Symp. on On-line Computer Methods Relevant to Chemical Engineering, Inst. Chem. Engrs. (London), Nottingham, England, 1971.
2.46 Coggan, G.C. and Wilson, J.A., Computer J., 14, 61, 1971.
2.47 Seinfeld, J.H., Chem. Eng. Sci., 24, 75, 1969.
2.48 Gavalas, G.R. and Seinfeld, J.H., Chem. Eng. Sci., 24, 625, 1969.
2.49 Seinfeld, J.H., Gavalas, G.R. and Hwang, M., Ind. Eng. Chem. Fund., 8, 257, 1969.
2.50 Yu, T.K. and Seinfeld, J.H., A.I.Ch.E. J., 19, 389, 1973.
2.51 Ajinkya, M.B., Ray, W.H., Yu, T.K., Seinfeld, J.H., Nat. J. Syst. Sci, 6, 313, 1975.
2.52 Goldman, S.F. and Sargent, R.W.H., Chem. Eng. Sci., 26, 1542, 1971.
2.53 Goldman, S.F., Ph.D. Thesis, University of London, 1969.
2.54 Joffe, B.L. and Sargent, R.W.H., Trans. Inst. Chem. Eng., 50 T270, 1972.
2.56 Fortescue, T.R. and Kershenbaum, L.S., "On-line Estimation and Control of a CO_2 Absorption Plant", The Institution of Chemical Engrs., 1st Annual Research Meeting, Imperial College, London, 1974.
2.56a Wells, C.H., A.I.Ch.E.J., 17, 966, 1971.
2.57 Mehra, R.K. and Wells, C.H., Preprints Joint Automatic Control Conference, Atlanta, Georgia, June 1970.
2.58 Coleby, J.M., Ph.D. Thesis, University of Aston in Birmingham, 1974.
2.59 Wells, C.H., and Wismer, D.A., Advances in Control Systems, 8, 217, 1971.
2.60 Soliman, M.A., Ray, W.H. and Szekely, J., Int. J. Control, 20, 4, 641, 1970.
2.61 Lynch, R.B. and Ramirez, W.F., A.I.Ch.E.J., 21, 4, 799, 1975.
2.62 Fisher, D.G. and Seborg, D.E., "Multivariable Computer Control - A Case Study", North-Holland/American-Elsevier, 1976.
2.63 Hamilton, J.C., Seborg, D.E. and Fisher, D.G., A.I.Ch.E.J., 19, 5, 901, 1973.
2.64 Seborg, D.E., Fisher, D.G. and Hamilton, J.C., Proc. 4th IFAC/IFIP Conference on Digital Computer Applications to Process Control, Zurich, March 1974.
2.65 Payne, S.G., Ph.D. Thesis, University of Aston in Birmingham, 1974.
2.66 Singer, R.A. and Sea, R.G., IEEE Trans. Auto. Control, AC-16, 254, 1971.

2.67 Chaw Bing Chang, Proc. 6th Hawaii Int. Conf. on Systems Sci., Honolulu, Hawaii, January 1973.
2.68 Krepyshev, G.P. and Pollack, Ya.A., Int. Process Lett., 1, 5, 211, 1972.
2.69 Mehra, R.K., IEEE Trans. Auto. Control, AC-16, 12, 1971.
2.70 Graupe, D. and Krause, D.J., Proc. 14th Midwest Symp. on Circuit Theory, Denver, U.S.A., 1971.
2.71 Schlee, F.H., Standish, C.J. and Toola, N.F., AIAA, 5, 1114, 1967.
2.72 Anderson, B.D.O., Int. Sciences, 5, 217, 1973.
2.73 Lee, R.C.K., Aerospace Comp. Tech. Memo., ATM-65 (5540-10) -18, 1965.
2.74 Friedland, B., IEEE Trans. Auto. Control, AC-14, 359, 1969.
2.75 Lin, J.L. and Sage, A.P., IEEE Trans. Syst. Man. Cybern., SMC-1, 314, 1971.
2.76 Jazwinski, A.H., Proc. IFAC Symp. on Multivariable Control Systems, Dusseldorf, Germany, 2, 1, 1968.
2.77 Jazwinski, A.H., Automatica, 5, 475, 1965.
2.78 Sriyananda, H., Int. J. Control, 16, 1101, 1972.
2.79 Jazwinski, A.H., Automatica, 10, 203, 1974.
2.80 Kilbride-Newman, R.F., Ph.D. Thesis, University of Aston in Birmingham, 1974.
2.81 Eckert, E.R.G., Sparrow, E.M., Goldstein, R.J. et al., Int. J. Heat and Mass Transfer, 17, 1287, 1974.
2.82 Kern, D.Q., "Process Heat Transfer", McGraw-Hill, New York, 1950.
2.83 McAdams, W.H., "Heat Transmission", McGraw-Hill, New York, 1954.
2.84 Kreith, F., "Principles of Heat Transfer", International Text Book Co., London, 1972.
2.85 British Patent 24, 024, 1899.
2.86 Brooks, C.H. and Badger, W.L., Trans. A.I.Ch.E., 83, 392, 1937.
2.87 Stroebe, G.W., Baker, E.M. and Badger, W.L., Trans. A.I.Ch.E., 35, 17, 1939.
2.88 Gupta, A.S. and Holland, F.A., Canadian J. Chem. Eng., 44, 77, 1966.
2.89 Dukler, A.E., Hwang, C.J. and Lee, H.M., Desalination, 6, 25, 1969.
2.90 Beveridge, G.S.G., Stewart, G. and Rajeswaran, T., I.Chem.Eng., Symp. Series No.35.
2.91 Heidemann, R.A., Huckaba, C.E., Eisen, F.S., Weisman, L.I., and Gallatig, G.M., Canadian J. Chem. Eng., 49 147, 1971.
2.92 Gallatig, G.M., M.Sc. Thesis, Drexel Institute of Technology, Philadelphia, U.S.A., 1967.

2.93 Buckley, P.S., "Techniques of Process Control", J. Wiley, New York, 1964.
2.94 Lees, S. and Haugen, J.O., Ind. Eng. Chem., 48, N6, 1064, 1956.
2.95 Stemole, F.J. and Larson, M.A., A.I.Ch.E. J. 10, 5, 688, 1964.
2.96 Privett, W.J. and Ferrell, J.K., Chem. Eng. Progress Symp. Series, 62 (64), 200, 1966.
2.97 Stainthorp, F.P. and Azon, A.G., Chem. Eng. Sci., 20, 107, 1965.
2.98 Korol'kov, B.P., Heat Transfer - Soviet Research, 2, 3, 74, May 1970.
2.99 Finlay, I.C. and Smith, J., J. Mech. Eng. Sci., 9, 3, 211, 1967.
2.100 Koppel, L.B., Ind. Eng. Chem. Fundamentals, 5, 3, 403, 1966.
2.101 Myers, G.E., Mitchell, J.W. and Norman, R.F., Trans ASME-J. of Heat Transfer, 89, 1, 75, 1967.
2.102 Dorri, M.K., Heat Transfer - Soviet Research, 1. 2, 186, 1969.
2.103 Schoenberg, A.A., "Mathematical Models with Experimental Verification for the Dynamic Behaviour of a Single-Tube Condenser", NASA-TN-D-3453, June 1966.
2.104 Momen, A., Ph.D. Thesis, University of Aston in Birmingham, to be published.
2.105 Johnson, D.E., I.S.A. J., 7, 46, 1960.
2.106 Nisenfeld, A.E. and Hoyle, D.L., Instrum. Tech., 17, 49, 1970.
2.107 Anderson, J.A., Glasson, L.W.A. and Lees, F.P., Trans. Soc. Inst. Tech. 13, 21, 1961.
2.108 Day, R.L., "Plant and Process Dynamics", Butterworths, 1957.
2.109 Zavorka, J., Sutek, L., Aguado, A. and Delgado, E., "Identification in Automatic Control Systems", IFAC. Symp., Prague, 1967.
2.110 Burdett, J.W. and Holland, C.D., A.I.Ch.E.J., 17, 5, 1080, 1971.
2.111 Ritter, R.A. and Andre, H., Canadian J. Chem. Eng., 48, 696, 1970.
2.112 Newell, R.B. and Fisher, D.G., Ind. Eng. Chem. Process Des. Dev., 11, 2, 213, 1972.
2.113 Wilson, R.G., Fisher, D.G. and Seborg, D.E., Int. J. Cont., 16, 3, 549, 1972.
2.114 Niemi, A. and Koistinen, R., Ppaeri Ja Puu, 54, 297, 1972.

REFERENCES FOR CHAPTER 3

3.1 See reference 2.63.
3.2 See reference 2.65
3.3 See reference 2.58
3.4 See reference 2.80
3.5 See reference 2.44
3.6 See reference 2.77
3.7 See reference 2.13
3.8 See reference 2.16
3.9 See reference 2.17
3.10 Koenig, H.E., Tokad, Y. and Kesaven, H.K., "Analysis of Discrete Physical Systems", McGraw-Hill, New Yor, 1967.
3.11 Bellman, R., "Introduction to Matrix Analysis", 2nd Edition, McGraw-Hill, New York, 1970.

REFERENCES FOR CHAPTER 4

4.1 See reference 2.17.
4.2 Lapidus, L., "Digital Computation for Chemical Engineers", McGraw-Hill, New York, 1962.
4.3 Wilkinson, J.H., "The Algebraic Eigenvalue Problem", Oxford University Press, London, 1965.
4.4 Rutishauser, H., Appl. Math. Ser. Nat. Bur. Stand., 49, 47, 1958.
4.5 Carnahan, B., Luther, H.A. and Wilkes, J.O., "Applied Numerical Methods", J. Wiley and Son, New York, 1969.
4.6 McCracken, D.D., "A Guide to FORTRAN IV Programming", John Wiley and Son, New York, 1965.
4.7 Gay, B. and Payne, S.G., Computer Journal, 16, 118, 1973.
4.8 Computer Aided Design Centre, "GINO-F User Manual",
4.9 Goldman, S.F., Ph.D. Thesis, University of London, 1969.
4.10 Jacobs, O.L.R., "Introduction to Control Theory", Oxford University Press, London, 1974.

REFERENCES FOR CHAPTER 5

5.1 Kestner Evaporator and Engineering Co. Ltd., "Laboratory Size Double Effect Evaporator, Document R-299.
5.2 Honeywell Information Systems Ltd., Brentford, England, "H316 and HADIOS - General Information".
5.3 Honeywell Information Systems Ltd., Brentford, England, 1970 "H316 and DDP 516 - Operators Guide", Honeywell Document No 42400343402A.

REFERENCES FOR CHAPTER 6

6.1 See reference 4.7.
6.2 Monro, D.M., "Interactive Computing with BASIC", Edward Arnold, London, 1974.
6.3 Honeywell Information Systems Ltd., Hemel Hempstead, England, "Series 16 Software - BASIC Language", 1971, Document No 41286103128A.
6.4 American Standards Association, "FORTRAN-Proposed Standard", 1969.
6.5 See reference 4.6.
6.6 Honeywell Information Systems Ltd., Hemel Hempstead, England, "FORTRAN IV Manual for DDP Computers", 1967, Document No 130071364A.
6.7 Honeywell Information Systems Ltd., Hemel Hempstead, England, "Honeywell Series 16 Assembly Languages - DAP-16 K DAP-16MOD2", 1971, Document No 41286384-000-01.
6.8 Honeywell Information Systems Ltd., Hemel Hempstead, England, "Honeywell Series 16 Software - OP-16 User's Guide", 1972, Document No 41286103020C.
6.9 Tektronix Information Display Products, Beaverton, U.S.A., "PLOT-10 Terminal Control System - 4010 User's Manual", 1972, Document No 062-1474-00.
6.10 University of Aston in Birmingham, Department of Chemical Engineering, "H316 Computer - Extended BASIC Graphics", 1974.
6.11 Honeywell Information Systems Ltd., Hemel Hempstead, England, "Honeywell Series 16 Software - OP-16 UTILITY PROGRAMS", 1972, Document No 41286103055B.

REFERENCES FOR CHAPTER 7

7.1 Hewlett Packard 2000 Access Computer, Probability and Statistics (400), Library Program 36178.
7.2 British Driver Harris, Data Sheet 2.

7.3 British Standard 1042: Part 1: 1964.
7.4 Richards, J.M., J. Phys. D : Appl. Phys., 4, L15, 1971.
7.5 Mayhew, Y.R. and Rogers, G.F.C., "Thermodynamic and Transport Properties of Fluids", Basil Blackwell, Oxford, 1970.
7.6 See reference 2.83.
7.7 British Patent, 2024,1899

7.8 See reference 2.92.
7.9 See reference 2.93.
7.10 Davies, O.L. and Goldsmith, P.L., "Statistical Methods in Research and Production", Oliver and Boyd, 1972.
7.11 See reference 2.58.

REFERENCES FOR CHAPTER 8

8.1 See reference 2.65.
8.2 See reference 2.58.
8.3 See reference 2.63.
8.4 See reference 2.115.
8.5 Andre, H., Ph.D. Thesis, University of Alberta, 1966.
8.6 See reference 2.110.
8.7 Huddle, J.R. and Wismer, D.A., IEEE TRANS. AUTO CONTROL, AC-13, 421, 1968.
8.8 See reference 7.1.
8.9 See reference 7.5.

REFERENCES FOR CHAPTER 9

9.1 See reference 4.4.
9.2 See reference 4.5.
9.3 See reference 2.58.
9.4 See reference 2.65
9.5 See reference 2.52.
9.6 See reference 2.53
9.7 See reference 2.67.
9.8 See reference 2.68.
9.9 See reference 2.71.
9.10 See reference 2.23.
9.11 See reference 2.56.
9.12 See reference 2.80.

REFERENCES FOR CHAPTER 10

10.1 See reference 2.80.
10.2 See reference 2.105.

REFERENCES FOR APPENDIX C

C.1 Momen A., Private communication.

REFERENCES FOR APPENDIX E

E.1 See reference 6.8.
E.2 See reference 6.11.

REFERENCES FOR APPENDIX G

G.1 See reference 5.3.