Computer Programs for Fullscale Experiments

Jensen, Lars

1974

Document Version:
Publisher's PDF, also known as Version of record

Link to publication

Citation for published version (APA):
Jensen, L. (1974). *Computer Programs for Fullscale Experiments*. (Technical Reports TFRT-7042). Department of Automatic Control, Lund Institute of Technology (LTH).

Total number of authors:
1

LUND UNIVERSITY

PO Box 117
221 00 Lund
+46 46-222 00 00

COMPUTER PROGRAMS FOR FULLSCALE EXPERIMENTS

L.H. Jensen

This work has been supported by Grant D698 from the Swedish Council for Building Research to the Department of Building Science and the Division of Automatic Control, Lund Institute of Technology, Lund.

TABLE OF CONTENTS

1 INTRODUCTION

The purpose with this report is to describe some realtime programs for process communication, process control, process program control and datalogging and how they can be used. The programs have been used in experiments with control of room air temperature control of a normal office room (Jensen & Ljung R7322), lecture room (Ekström, Hänsel, Jensen and Ljung 1974a and 1974b), an enthalpy exchanger (Jensen & Hänsel 1974) and a hospital room (Jensen & Hänsel 1974).

The computer is a PDP-15 (Digital) with 32 k core memory and with a 256 k disc. The process interface is a coupler/controller (Hewlett Packard) and it is described in Jensen (1973).

All programs have been run in the realtime executive RSX-plus. The reader is supposed to be familiar with this executive. In this system all programs are referred to as tasks and this will also be the case in this report. The main purpose with the software was to make automatic experiments. The dynamics of the room is in the range of some minutes and more. Experiments can then last several hours. It has also been necessary to run the experiments during nights when the demand for computer time was low.

The tasks have been written rather straightforwardly. No effort has been made to make the software as fast as possible or as small as possible. A general view of how and when the different tasks are used are given in section 2.

A more detailed description is given in section 3 of the different tasks. The input command strings to different tasks are given. Also some examples are given of how the tasks can be used to execute a regulator and to change experiment conditions.

2 NORMAL USAGE OF TASKS

The RSX-plus realtime executive is a multiprogramming system. The core memory is divided into partitions in which different tasks execute according to priority. A task always execute to completion before it leaves its partition. There are four partitions of different sizes which the user can use. A task can be requested, run, synchronized or cancelled from the monitor console or from a task.

2.1 Defining regulators and experiment conditions

These are defined by reals and integers in a database on the disc. Several tasks work with this database.

First the database area is allocated on the disc by the task DEFDB, when requested. An old database can be fetched from dectape by the task DTDB. Modifications can then be done with the task CDB. The changes are checked by the task DISDB1 and DISDB2, which displays the database when requested. The database can also be printed on the lineprinter. This is done by using the tasks LISDB1 and LISDB2. The database can then be saved by the task DTDB so that it can be used in another experiment.

2.2 Preparation for datalogging

The user has to put in all necessary reals and integers in the IOCOM common area needed for the datalogging. The needed disc area is allocated by the task LOGG when requested.

2.3 Start of experiment

The coupler/controller input output task CCIO is run at every sample. The datalogging is started by setting bit 17 in LIWRD equal to one. The regulators are computed by running the task REGLE every sampling interval. The experiment conditions are changed by the task REGLA, which also has to be run every sampling interval.

2.4 Displaying of process variables

This is done by the tasks OLDIO and ...OLD. This can only take part when data is logged. The process variables can also be displayed as numbers by the task DISIO and DISDB1. The former task displays the IOCOM common area and thereby all the latest inputs and outputs from the process. The latter task displays the last five values of processvariables used in regulators. Also minimum, maximum, greatest change and mean values are displayed.

2.5 Stop of experiment

The datalogging can be stopped automatically when the number of samples specified is reached or by setting bit 17 in LIWRD equal to zero.

The experiment condition task REGLA and the regulator task REGLE are cancelled.

The task CCIO is also cancelled after the outputs have been changed to wanted values and thereby turn off the process.

2.6 Save experiment

Now data can be saved from the disk to dectape by the task CONV. Output to the lineprinter of the experiment data is done by using the task PRINT. The logged data can also now be displayed by the tasks OLDIO and ...OLD. The allocated disc area is finally freed by requesting the task LOGG.

3 TASKS

Most of the tasks are requested, run or synchronized by the user with the monitor console routine. It types MCR> when it is ready to recieve a command string. In which partition and with what priority the tasks are executed are given in appendix 11.

3.1 Tasks for communication with coupler/controller

3.1.1 Task CCIO

This task performs the normal communication between the computer and the c/c, which is a process interface. The c/c is interfaced as a teletype to the computer.

There are ten analog inputs. These are measured by a digital voltmeter. The measurement ranges are ±10., ±1. and ±0.1 volt. The range can be fixed by the user or chosen automatically by the digital voltmeter. The measurements are given with four to five digits. The analog inputs are tested for nine different fault conditions. The error message is set in the reading as a great number which cannot be measured. These numbers are given in appendix 1. The multiplexer can be damaged if the analog inputs exceeds the interval (-15.,15.) volt.

There is no logical inputs. There are four analog outputs and the range is ±10. volts. Four logical outputs can also be controlled. These consists of relays which are either open or closed.

The task CCIO allows another high priority task to compute the regulator between the input from and the output to the process.

3.1.2 Task ...CCC

This task is a MCR routine, which can test, turn off and turn on the c/c. Allowed command stings are as follows:

MCR>CCC $a_1a_2a_3a_4$ MESSAGE

MCR>CCC COF

MCR>CCC CON

The four parameters a_1, a_2, a_3 and a_4 are integers between 0 and 9. They are interpretted as follows:

a_1	= 0	no input is expected from c/c
a_1	= 1	input is expected from c/c
a_2	= 0	input from c/c is not printed
a_2	= 1	input from c/c is printed
a_3		delay in tenths of a second between input from and output to c/c
a_4		number of times the message is sent

MESSAGE ASCII characters to be sent to c/c

The COF command turns off the c/c and the CON command turns on c/c.

3.1.3 Task ...CBC

This task is also a MCR routine, which changes variables in IOCOM common area. The allowed command strings are as follows:

```
MCR>CBC  IADD  REAL
MCR>CBC  IADD  INT
MCR>CBC  IADD  INT  BIT
```

The parameter IADD is a three digit integer, which is the adress in the IOCOM common area. With the first command is a real changed and with the second one an integer. The third command is used when INT th bit is changed to BIT. The routine changes variables only when the IOCOM common area is available. The old value is printed in octal form. Some of the addresses to variables in the IOCOM common area are given in appendix 2.

3.1.4 Task DISIO

This task displays the IOCOM common area. An example is shown in appendix 3.

3.2 Task for regulators

3.2.1 Task REGLE

A 360 word database is used to define the regulators. The task can be divided into two parts. The first part of the program deals with input of new process variables to the database from the IOCOM common area. These are checked for faults and statistics are computed. The second part computes control signals from data in the database and they are placed in the IOCOM common area.

Five process variables can be used in the database. The type and the position is determined by the two integer vectors IDT(1-5) and IDP(1-5). The standard for IDT is as follows:

IDT = 0 nothing
IDT = 1 analog input
IDT = 2 logical input
IDT = 3 analog output
IDT = 4 logical output

The process variables are transformed by multiplying a constant from the vector RIK(1-5) and adding a constant from the vector RIL(1-5). The transformed process variables are checked against minimum limits, maximum limits and greatest change. These values are found in the vectors TMIN(1-5), TMAX(1-5) and TDIF(1-5). At the same time some statistics are computed. These are such as minimum value, maximum value greatest change and the mean value. These are stored in the vectors RMIN(1-5), RMAX(1-5), RDIF(1-5) and RMED(1-5).

Old process variables are shifted down in the field RV(1-5, 2-5). The three PID values are also computed and stored in the field RV(6-10,1-3). The used setpoint values are found in the vector TBOR(1-5).

Five regulators can be computed. This is done by adding up three scalar products of process variable vectors from the field RV and regulator coefficient vectors from the field CV. The process variable vectors can either be a time vector of a process variable (the five first rows in the field RV), or a PID vector (the five last rows in the field RV). Corresponding regulator coefficient vectors exist and they are found in the field CV.

The five sums of the three scalar products are determined by the fields ID(1-5,1-3) and IC(1-5,1-3). The value of ID determines which one of the ten rows in the process variable field RV that should be used. The value of IC is used in the same way with respect to the rows in the regulator coefficient field CV.

The outputs are then transformed by multiplying a constant from the vector ROK(1-5) and by adding a constant from the vector ROL(1-5). The vectors IRT(1-5) and IRP(1-5) determines the type and the position of the control signal. The standard for IRT is the same as earlier mentioned for the process variable type IDT.

The task waits until the database is available. The database is reserved when it is used.

In appendix 5 a simple PID controller is defined.

3.2.2 Task CDB

This task is used to change parameters in the database used by the tasks REGLE and REGLA. The command string is as follows:

```
MCR > REQ   CDB
 >IADD   V1  V2...
```

The integer IADD defines the start address where the values V1 V2... are to be placed in the database. Five reals or ten integers can be changed at a time. Where the different parameters are situated are given in appendix 4 and 7. The task reserves the database only when the disktransfer is going to be made.

3.2.3 Task DEFDB

This task allocates disk area for the database used by the task REGLE and REGLA.

3.2.4 Task DTDB

This task transfers a database between disc and dectape in both directions. The command string is as follows:

```
MCR > REQ   DTDB
> SAVE/GET  FILNM  EXT
```

The extension should be DB1, DB2 or DB3 in order to transfer the first 360, 720 or 1080 words of the database.

3.2.5 Task DISDB1

This task displays the database used by the task REGLE. It shows regulator parameters, timevectors, PID values, statistic values and transformation constants. Only the used variable are displayed .

3.2.6 Task LISDB1

This task prints the database on the lineprinter. An example of a printout is given in appendix 6.

3.3 Tasks for experiment control

3.3.1 Task REGLA

This task can change all real numbers in the database part used by the regulator task REGLE. The task REGLA has a database of 360 words to store the parameters to be changed, where and when the changes shall be made. Fifteen different parameters can be changed independently to seven different values. These are stored in the field PUFF(1-15,1-7).

Where and when a change is made is determined by the corresponding integer field IPUFF(1-15,1-9).

Where a change is made is given by the eight column of the field IPUFF. The integer value IPUFF(I,8) is the address in the database where the Ith parameter, that is changed, is situated.

When a change of the Ith parameter is to be made is determined by the integers IPUFF(I,1-7) and IPUFF(I,9). The last integer is a counter, which is updated at every run of the task (REGLA). The seven integers IPUFF(I,1-7) are used as timers. The parameter value PUFF(I,J) is used if the timer IPUFF(I,J) is the smallest timer that is greater or equal to the counter IPUFF(I,9). If the counter IPUFF(I,9) is greater than any timer value IPUFF(I,J) then the counter IPUFF(I,9) is zeroed. This means that the changes become periodical.

An example of different changes of parameters in the earlier mentioned PID regulator is given in appendix 8.

Also this task checks if the database is available before using it.

3.3.2 Task DISDB2

This task displays the part of the database used by the task REGLA. Only the used parameters are displayed.

3.3.3 Task LISDB2

This task prints the database on the line printer. An example is given in appendix 9.

3.4 Tasks for datalogging

3.4.1 Task LOGG

This task allocates or deallocates diskarea according to the numbers in the IOCOM common area. The task asks the user if it shall allocate or deallocate. This is done as follows:

```
MCR>REQ   LOGG
 >START  OR  STOP?
```

3.4.2 Task LOGGA

This task is requested by the coupler controller input output task CCIO after every sampling interval if the 17th bit in LIWRD is equal to one. When the specified number of samples NUSAM is reached, then the logging is interrupted. The line LOGGA STOP NASA = NUSAM is printed. If instead bit 17 is set equal to zero then the logging is interrupted. The line LOGGA STOP BIT 17 = 0 is printed.

3.4.3 Task CONV

This task transfers data between disc and dectape in both directions. The command string is as follows:

```
MCR> REQ  CONV
 >SAVE/GET  FILNM  EXT
```

3.4.4 Task PRINT

Logged data on the disk is printed on the line printer by this task when requested.

3.5 Tasks for online plotting

A good way to see how the process behaves is to display the process inputs and outputs as curves in time. This can be done by using the tasks OLDIO and ...OLD. Only the variables that are logged can be displayed.

3.5.1 Task OLDIO

This task can display one to four different logged process variables at a time. Each curve has its own y- and x- axis. The number of samples per picture and the y scale minimas and maximas are controlled by the user with the task ... OLD.

The process variables can be displayed online if the task is synchronized with the data logging. The first picture has to be inited by the task ...OLD. An example is given in appendix 10.

3.5.2 Task ...OLD

This is a MCR routine which is easy to use. Three different command strings can be given. The variable that is to be displayed and its y scale is determined by the command string.

```
MCR>OLD  IT  IC  IV  MIN  MAX
```

Where

IT = 4	replot all pictures
IT = 5	replot last picture
IC = 1-4	curve number from the bottom
IV	variable position in the amount that is written on the disc every sample
MIN	y scale minima
MAX	y scale maxima

After variables and its y scales are determined then the plot is initilized by command string:

```
MCR>OLD  IT  NC  NP
```

Where

IT = 0	plot all pictures
IT = 1	plot last picture
IT = 2	as IT = 0 but go into online mode
IT = 3	as IT = 1 but go into online mode
NC = 1-4	number of curves to be plotted
NP	number of samples per picture

The y scales and x scales can be changed when the online plotting is running. To change the time scale a command string is used as follows:

MCR>OLD IT NP

Where

IT = 6	replot all pictures
IT = 7	replot only last picture
NP	new number of samples per picture

The 31 last words in the IOCOM common area are used to store information about the plotting. An example of a plot is given in appendix 10.

The task DISIO, DISDB1 and DISDB2 can also use the display when online plotting is done. The task OLDIO will automatically replot the display picture when it is requested at the next sampling.

4 REFERENCES

Ekström, L., Hänsel, R., Jensen, L.H., Ljung, L., 1974 a, Different dynamic models in a airconditioned building (Department of Building Science and Division of Automatic Control, Lund Institute of Technology) Work-report 1974:1.

Ekström, L., Hänsel, R., Jensen, L.H., Ljung, L., 1974 b, Experiments with computer control of an airconditioning plant (Department of Building Science and Division of Automatic Control, Lund Institute of Technology). Work-report 1974:2.

Jensen, L.H. & Ljung, L., 1973, Experiments with computer control of room air temperature (Division of Automatic Control, Lund Institute of Technology) Report 7322, July 1973. Lund.

Jensen, L.H., 1973, Ett coupler/controllersystem (Institutionerna för byggnadskonstruktionslära och reglerteknik, Lunds Tekniska Högskola) Arbetsrapport 1973:4 (in Swedish). Lund.

Jensen, L.H. & Hänsel, R., Computer control of an enthalpy exchanger (Department of Building Science and Division of Automatic Control, Lund Institute of Technology) (to be published).

Jensen, L.H. & Hänsel, R., Computer control of room air temperature by onoff control of a postheater (Department of Building Science and Division of Automatic Control, Lund Institute of Technology).
(to be published).

5 APPENDIX

Appendix 1

Analog input error codes.

Error value in FLAI	Checking order	Fault
126.0	5	overload plus
126.5	6	overload minus
127.0	4	range digit is not 0,1 or 2
127.5	3	channelnumber is not equal knr
128.0	1	wrong character found
128.5	7	overload digit is >3 or <0
129.0	2	overrangedigit is not 0 or 1
129.5	8	overloaddigit and sign contradiction
130.0	9	iamp + range +5 > 0 and over flow

As soon as a fault is discovered then the corresponding error code is set and further checking is stopped. The checking starts with the lowest checking number.

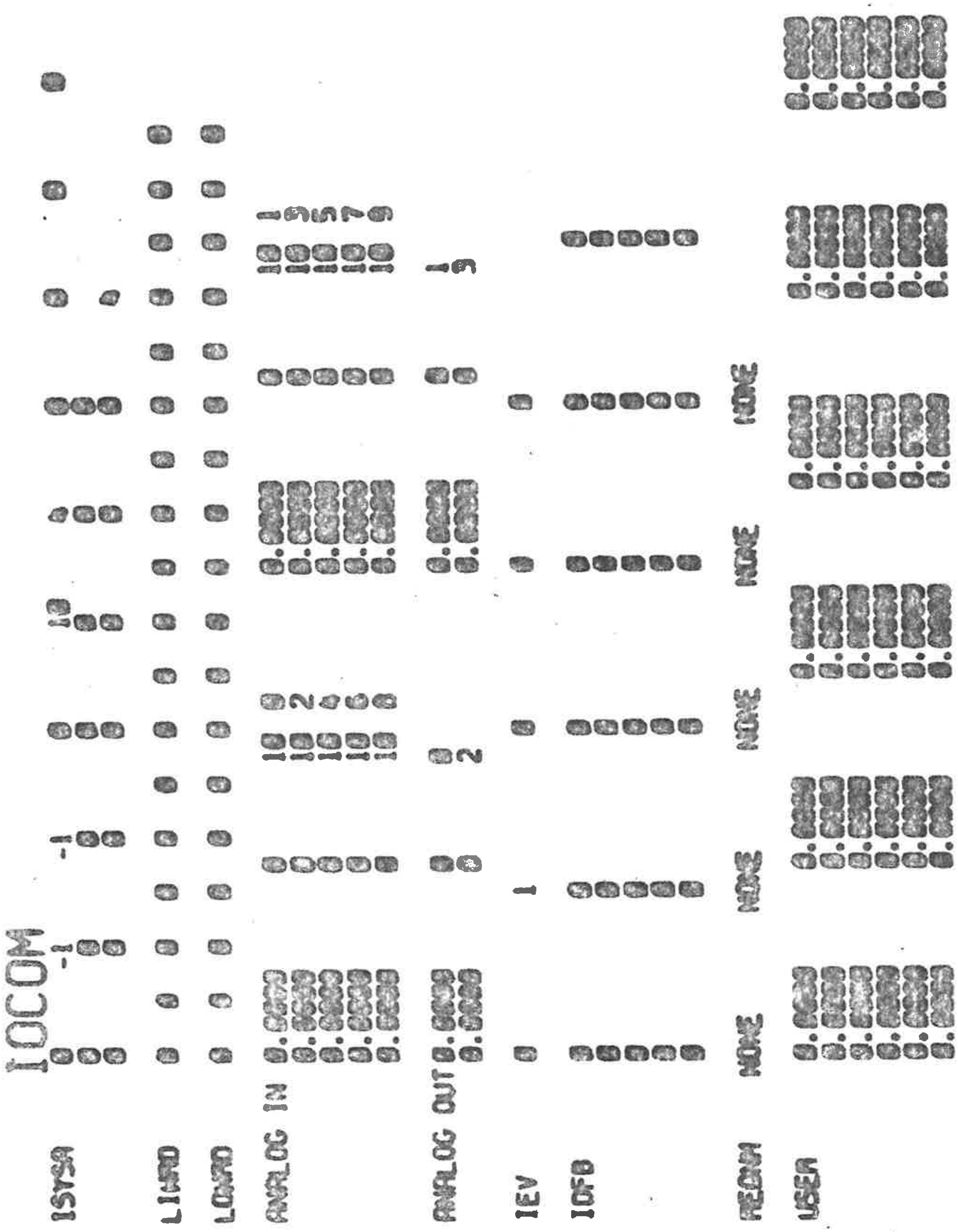
Appendix 2

Location for variables in IOCOM common area.

Parameter or vector	Address
NVAI	004
NVAO	005
NWRAI	006
NWRAO	007
NWRUC	008
NWRSL	009
NUSAM	010
NASA	011
LIWRD	025
LOWRD	026
IDFB	091 - 120
IEV	131 - 135
FLAI	136 - 167
FLAO	168 - 183
USER	184 - 255

Appendix 3

Display picture of IOCOM common area generated by the task DISIO



Appendix 4

Location of variables in database used by the task REGLE

Vector or field	Address	Variable type
RMIN	0 - 8	minimum value
RMAX	10 - 18	maximum value
RDIF	20 - 28	greatest change value
RMED	30 - 38	mean value
TMIN	40 - 48	test minimum value
TMAX	50 - 58	test maximum value
TDIF	60 - 68	test greatest change value
TBOR	70 - 78	set point value
RV	80 - 178	time vectors
RIK	150 - 158	input gain
RIL	170 - 178	input bais
PVA	090 - 098	P value
IVA	110 - 118	I value
DVA	130 - 138	D value
CV	180 - 278	coefficient vectors
ROK	250 - 258	output gain
ROL	270 - 278	output bais
PCO	190 - 198	P coefficient
ICO	210 - 218	I coefficient
DCO	230 - 238	D coefficient
IRT	300 - 304	regulated variable
IRP	305 - 309	position
IDT	310 - 314	database variable
IDP	315 - 319	position
ID	320 - 334	database pointer
IC	335 - 349	coefficient pointer

Appendix 5

The controlled variable is assumed to be analog input seven and it has got an offset of one unit which has to be removed. The analog input is placed as second process variable then

```
IDT(2) = 1
IDP(2) = 7
RIK(2) = 1.
RIL(2) = -1.
```

The analog input should be in the interval (-10.,10.). The greatest change between two readings is 5. and the setpoint is 0. then

```
TMIN(2) = -10.
TMAX(2) = 10.
TDIF(2) = 5.
TBOR(2) = 0.
```

The PID parameters are K_P=-1., K_I=-0.5 and K_D=-0.4 then

```
CV(6,1) = -1.
CV(6,2) = -0.5
CV(6,3) = -0.4
```

The first regulator shall control the analog input seven with the analog output five. The output should be offset plus ten units then

```
ID(1,1) = 7
IC(1,1) = 6
IRT(1)  = 3
IRP(1)  = 5
ROK(1)  = 1.
ROL(1)  = 10.
```

The command strings to set up the PID regulator are as follows:

MCR >REQ	CDB
>311	1
>316	7
>152	1.0
>172	-1.0
>042	-10.
>052	10.
>062	5.
>072	0.
>190	-1.
>210	-0.5
>230	-0.3
>320	7
>335	6
>250	1.
>270	10.
>300	3
>305	5

Appendix 6

Printout of the first data base part by the task LISDB1.

```
    RMIN          RMAX          RDIF          RMED
     0.0000        0.0000        0.0000        0.0000
     0.0000        0.0000        0.0000        0.0000
     0.0000        0.0000        0.0000        0.0000
     0.0000        0.0000        0.0000        0.0000
     0.0000        0.0000        0.0000        0.0000
    TMIN          TMAX          TDIF          TBOR
     0.0000        0.0000        0.0000        0.0000
   -10.0000       10.0000        5.0000        0.0000
     0.0000        0.0000        0.0000        0.0000
     0.0000        0.0000        0.0000        0.0000
     0.0000        0.0000        0.0000        0.0000

INPUT GAIN
      1.00000      1.00000      1.00000      1.00000      1.00000
INPUT BIAS
      0.00000     -1.00000      0.00000      0.00000      0.00000
OUTPUT GAIN
      1.00000      1.00000      1.00000      1.00000      1.00000
OUTPUT BIAS
     10.00000      0.00000      0.00000      0.00000      0.00000
TIME-VECTORS
      0.00000      0.00000      0.00000      0.00000      0.00000
      0.00000      0.00000      0.00000      0.00000      0.00000
      0.00000      0.00000      0.00000      0.00000      0.00000
      0.00000      0.00000      0.00000      0.00000      0.00000
      0.00000      0.00000      0.00000      0.00000      0.00000
PID-VALUES
      0.00000      0.00000      0.00000      0.00000      0.00000
      0.00000      0.00000      0.00000      0.00000      0.00000
      0.00000      0.00000      0.00000      0.00000      0.00000
TIME-COEFFICIENTS
      0.00000      0.00000      0.00000      0.00000      0.00000
      0.00000      0.00000      0.00000      0.00000      0.00000
      0.00000      0.00000      0.00000      0.00000      0.00000
      0.00000      0.00000      0.00000      0.00000      0.00000
      0.00000      0.00000      0.00000      0.00000      0.00000
PID-COEFFICIENTS
     -1.00000      0.00000      0.00000      0.00000      0.00000
     -0.50000      0.00000      0.00000      0.00000      0.00000
     -0.30000      0.00000      0.00000      0.00000      0.00000

TYPE DATABASE VARIABLE
        0        1        0        0        0
POSITION DATABASE VARIABLE
        0        7        0        0        0
TYPE REGULATED VARIABLE
        4        0        0        0        0
POSITION REGULATED VARIABLE
        5        0        0        0        0
POSITION REGULATOR-COEFFICIENTS
        6        0        0        0        0
        0        0        0        0        0
        0        0        0        0        0
POSITION DATABASE-VECTORS
        7        0        0        0        0
        0        0        0        0        0
        0        0        0        0        0
```

Appendix 7

Location of variables in database used by the task REGLA

Vector or field	Address	Variable type
PUFF	360 - 568	parameters
IPUFF	570 - 674	timers
IPUFF	675 - 689	pointer
IPUFF	690 - 704	counter

Appendix 8

The parameters in the earlier mentioned PID regulator is going to be tested for two different values of each regulator parameter for 100 samples. This gives eight different regulators that are to be tested. The regulator parameters are

K_P = -2. and -1.
K_I = -0.6 and -0.3
K_D = -0.4 and -0.2

Then the following variables have to be set in the database part used by the task REGLA:

The parameters

PUFF(1,1) = -2.
PUFF(1,2) = -1.
PUFF(2,1) = -0.6
PUFF(2,2) = -0.3
PUFF(3,1) = -0.4
PUFF(3,2) = -0.2

The timers

IPUFF(1,1) = 400
IPUFF(1,2) = 800
IPUFF(2,1) = 200
IPUFF(2,2) = 400
IPUFF(3,1) = 100
IPUFF(3,2) = 200

The pointers

```
IPUFF(1,8) = 96
IPUFF(2,8) = 106
IPUFF(3,8) = 116
```

The pointers have to be computed as the index to a real array. The address to the P parameter is 190 when the task CDB is used. The first address for a variable is 0 when CDB is used and 1 when REGLA is used. The addresses in appendix 4 can be converted to pointers by dividing by two and adding one.

The command strings to set up the experiment conditions to test eight different PID regulators are given below.

```
MCR > REQ  CDB

>360   -2.   -0.6   -0.4
>390   -1.   -0.3   -0.2
>570   400   200    100
>585   800   400    200
>675   96    106    116
```

Appendix 9

Printout of the second data base part by the task LISDB2.

PUFF(15,7)

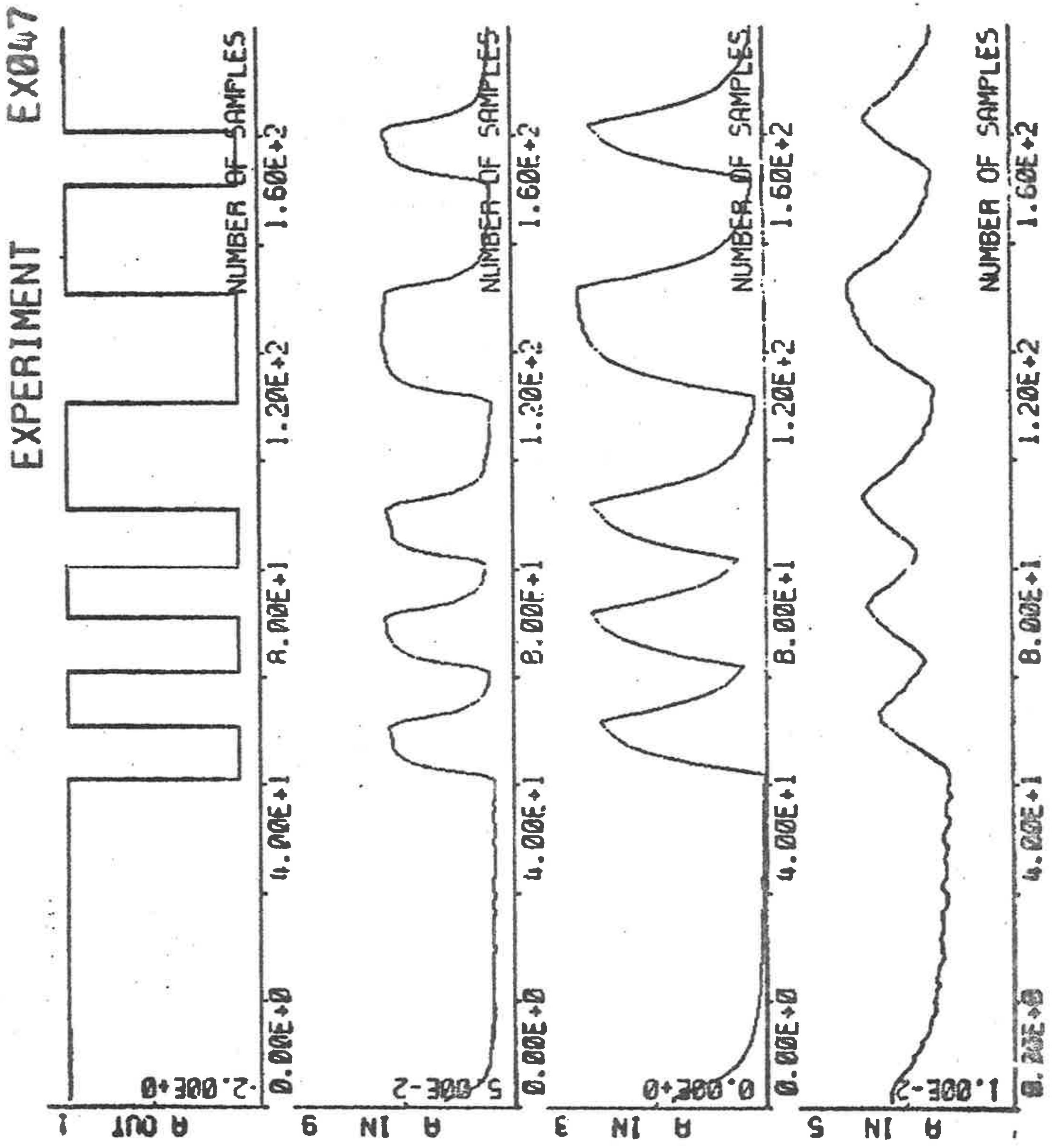
-2.0000	-1.0000	0.0000	0.0000	0.0000	0.0000	0.0000
-0.6000	-0.3000	0.0000	0.0000	0.0000	0.0000	0.0000
-0.4000	-0.2000	0.0000	0.0000	0.0000	0.0000	0.0000
0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000

IPUFF(15,10)

400	800	0	0	0	0	0	96	0	0
200	400	0	0	0	0	0	106	0	0
100	200	0	0	0	0	0	116	0	0
0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0

Appendix 10

A picture of displayed process variables made by the task OLDIO.



Appendix 11

Task	Partition	Priority
...CBC	MCR	100
...CCC	MCR	100
...OLD	MCR	100
CCIO	RTIO	225
CDB	PUSER	400
CONV	PUSER	500
DEFDB	PIPUS	200
DISDB1	PUSER	500
DISDB2	PUSER	500
DISIO	PUSER	500
DTDB	PUSER	300
LISDB1	PUSER	500
LISDB2	PUSER	500
LOGG	PUSER	400
LOGGA	RTIO	275
OLDIO	PUSER	500
PRINT	PUSER	500
REGLA	PIPUS	150
REGLE	PIPUS	300