and hence can only be quantified in terms of a particular application. In this particular application a high trace is obtained with $P_0$, while a 'low' trace is obtained when the parameters have converged and the system is under the correct control (cf. section 3.4). In the tracking problem (changing system dynamics) considered traces which are too high will allow small errors to cause dramatic tuning transients, whilst traces that are too low will keep the parameters steady while the dymanics of the system change, until system errors are so large that retuning has to take place with a severe transient. The correct tracking response is one of small retuning transients at each small change in system dynamics. This is achieved by keeping the information content of the self-tuner balanced as explained in section 3.3.

(ii) The Sampling Interval

Although a discrete-time model is used, the underlying process is continuous time, and the choice of sampling interval is important and must be made with due regard to the natural time scales of the process response. Unlike the digital implementation of a 3-term controller, where a fast sample rate (at least up to the Nyguist frequency) produces a more accurate representation, self-tuning experiments indicate a sampling rate surprisingly low in relation to the physics of the system (typically 10% of the dominant plant time constant).

Small changes in sampling interval have minimal effect, large changes may be necessary if the initial guess was poor and this may be indicated by the tailoring polynomial which gives the best results. If good reponses are obtained with very slow closed-loop poles (eg.

$t_1 = 0.95$) then the sampling is probably too fast. Conversely, a tailoring polynomial with $t_1 = 0.3$ giving the best results would indicate the need for the faster sampling.

Control objectives, plant considerations and computational capabilities might fix the sampling interval, hence precluding on-line trimming. In the particualr experiments performed only the above ideas were verified, and therefore the sampling rate was fixed for each individual run. The selection of the 'best' sampling rate, in view of the literature available, remains one of the least clear decisions in self-tuning.

(iii) Tailoring polynomial ($T(z^{-1})$)

The tailoring polynomial affects the shape of the system step responses and also the form of the control signal, as was discussed in Chapter 4. Closed-loop pole positions can, therefore, be adjusted to 'shape' the step responses. However closed-loop zero positions are not specified, this is the reason why sometimes the step responses might look rather odd.

Changing $T(z^{-1})$ on-line requires re-tuning of the estimator, and hence this should only be performed when the trace of the P matrix is low, that is when the parameter estimates are nearly correct.

In (ii) above includes some results regarding the trade-off between the choice of $T(z^{-1})$ and sampling interval. It is also important to add that first-order tailoring polynomials are adequate, and increasing its order rarely helps. Most errors obtained regarding incorrect step responses are due to an incorrect system model order, and for these reasons on-line trimming of $T(z^{-1})$ was therefore avoided.

Care must be taken when specifying the closed-loop response, so that, as in conventional classical control strategies, the user must make sure that the request is achievable with acceptable control effort. This is the particular problem associated with the minimum variance law developed by Aström |3|.

The main aim of this chapter is to document the experiences encountered when running the self-tuner on the actual physical plant. This dictates how the self-tuner can cope with actuator non-linearities, sensor initial dead band and other inaccuracies in the modelling process. In section 6.4 this is discussed in detail, in particular, a comparison between the robustness of the self-tuner and the more conventional three-term control laws included. Obviously the latter strategies do not cope with a changing plant, but cope remarkably well with the non-linearities mentioned above.

The self-tuner could be attached to the plant and set into operation with some initial guesses of the various parameters, as is done in the simulations (cf. Chapter 4). However, it is more appropriate to use the following three-stage process;

(a) Close the loop with a conventional strategy and simply monitor input/output data.
(b) Start least-squares estimation and observe the trace of the P matrix or the number of samples.
(c) When this trace or the number of samples compares with some user-defined threshold the conventional strategy is replaced by the self-tuner.

The conventional strategy can be either a classical three-term law or the following which gave very good results. The output is full-on until the measurement exceeds the demanded set-points, and then full-off until the measurement is below the set-point. This being repeated a number of times. A further advantage of this strategy is that the cross-over-points in the response and set-point lines can be used to ascertain an appropriate sampling interval. Both these strategies were used in addition to the initial use of the self-tuner. The results are documented subsequently.

In conclusion, any self-tuner requires the user to have a adequate knowledge of the physical plant to be controlled, and of the identification and controller algorithms. The latter is, obviously, dependent upon the freedom of choice of parameters the particular suite gives the user. The more general the particular self-tuner, the more complex and the more knowledge the user is required to have. Since the main objective of this work was to develop and test a particular brand of self-tuner the user (see Chapter 5) is required to work at a fairly low level of abscraction. Appropriate comments concerning future recommendations on this topic are given in the next chapter.

## 6.3. EXPERIMENTAL RESULTS

In this section the experimental work in using the self-tuner algorithm described in Chapters 3 & 4 on-line with the flow rig described in Chapter 2 is presented.

It is important to note that the experience accumulated by running the simulator and doing off-line plant step tests was invaluable in providing the necessary insight into the correct choice of parameters.

Numerous experiments were performed in order to investigate the effect of the various identification and controller parameters and model order on the self-tuner behaviour. All the concepts discussed in sections 3.5 and 4.5 are still applicable. The next section describes in more detail the practical problems encountered in applying the self-tuner on-line with the plant.

## GRAPH 6

## EXPERIMENTAL RESULTS

PARAMETERS :

Initial guesses : $\alpha_o = -1$

$\beta_o = 1\ 000$

Memory length : $\Sigma_o = 600 * R$

$R = 1E-4$

(This choice of R has the effect of reducing the estimator sensitivity to measurement noise, however, it decreases the parameter adaptability capabilities).

Tailoring polynomial : $t_1^i$ = -0,85 for both loops

Initialisation stage : 0-1 000 secs

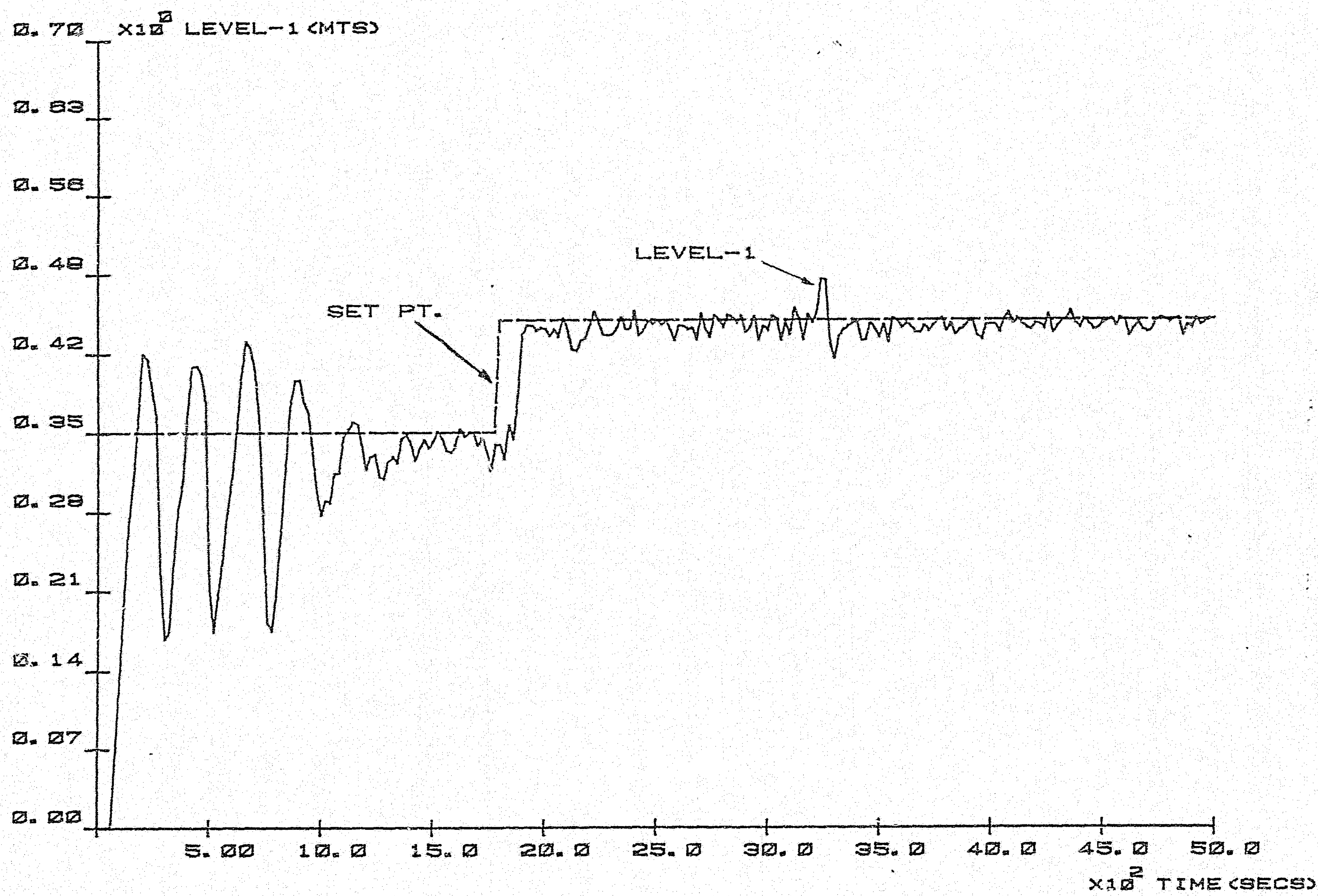
TEST CONDITIONS :

At 1 800 - Plant modified

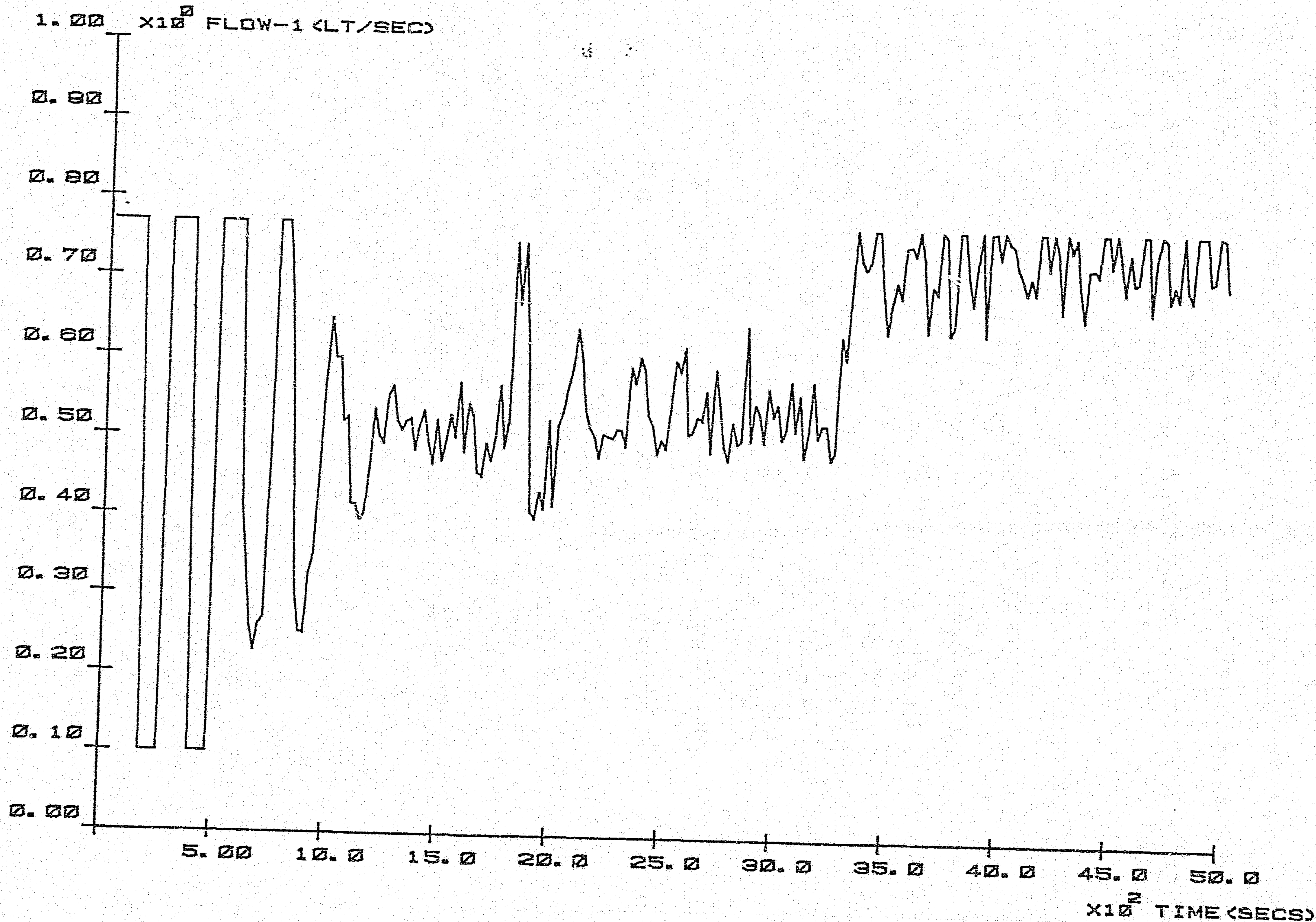
At 3 300 - Plant modified

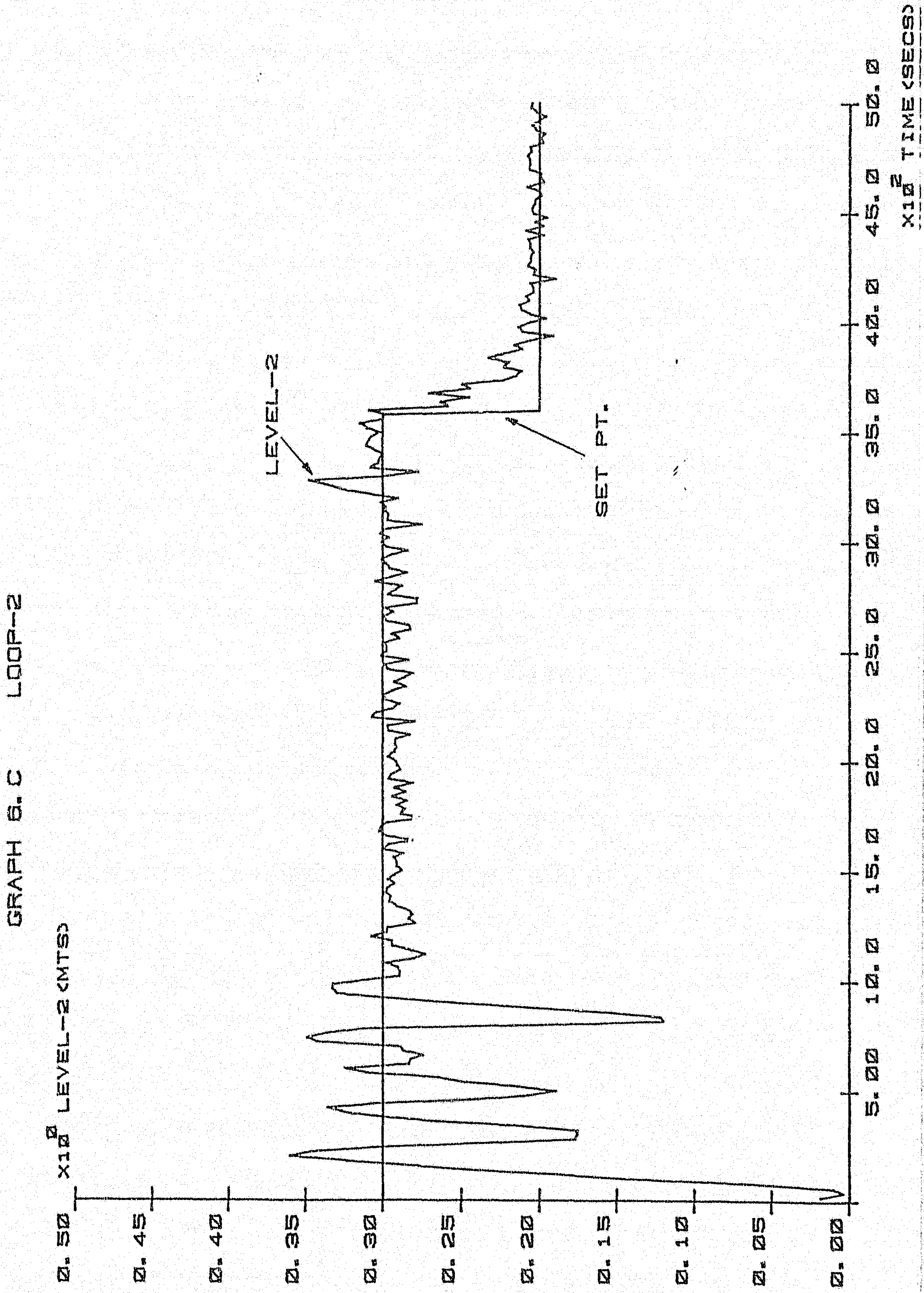
At 3 600 - Plant modified

GRAPH 6. A LOOP-1

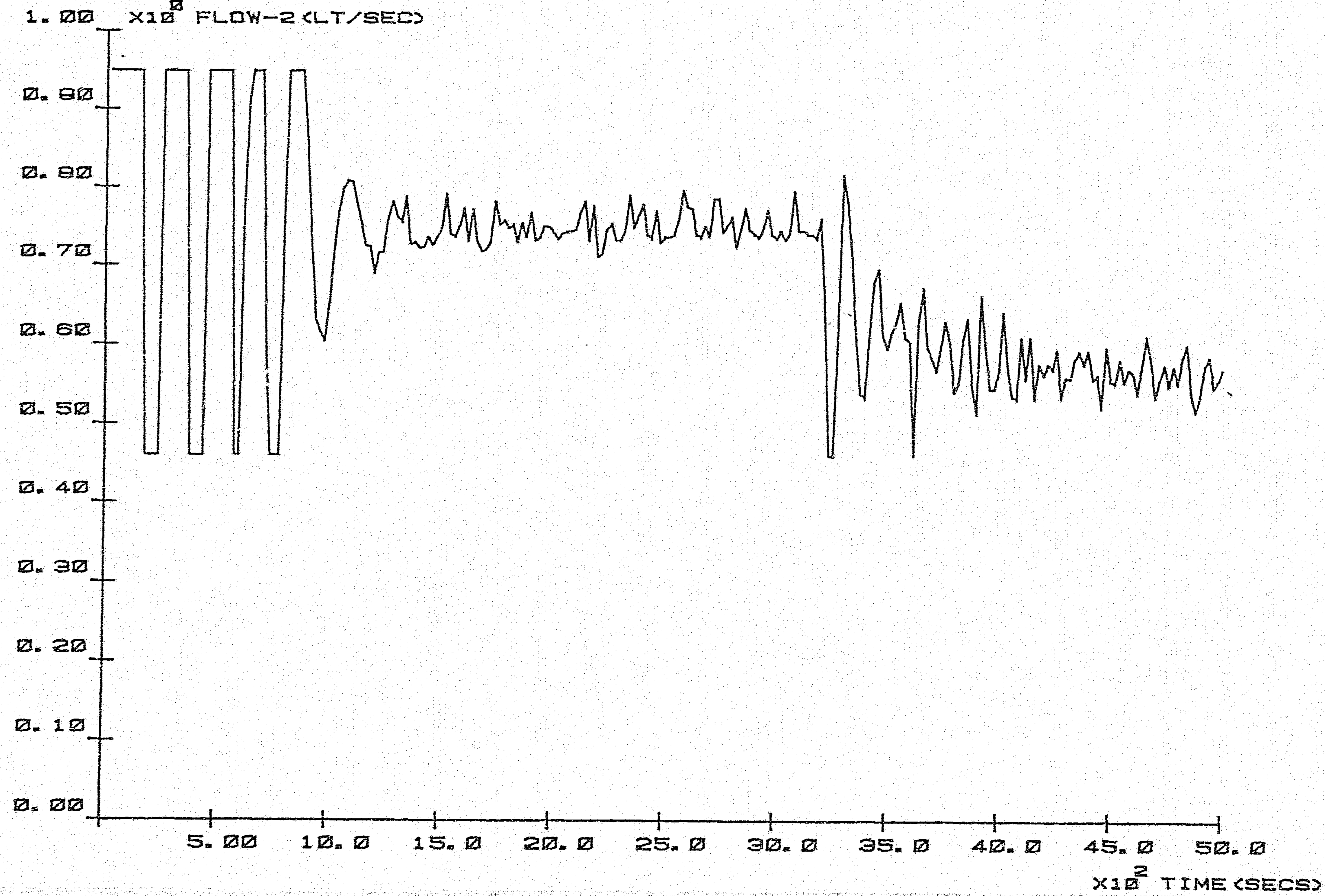


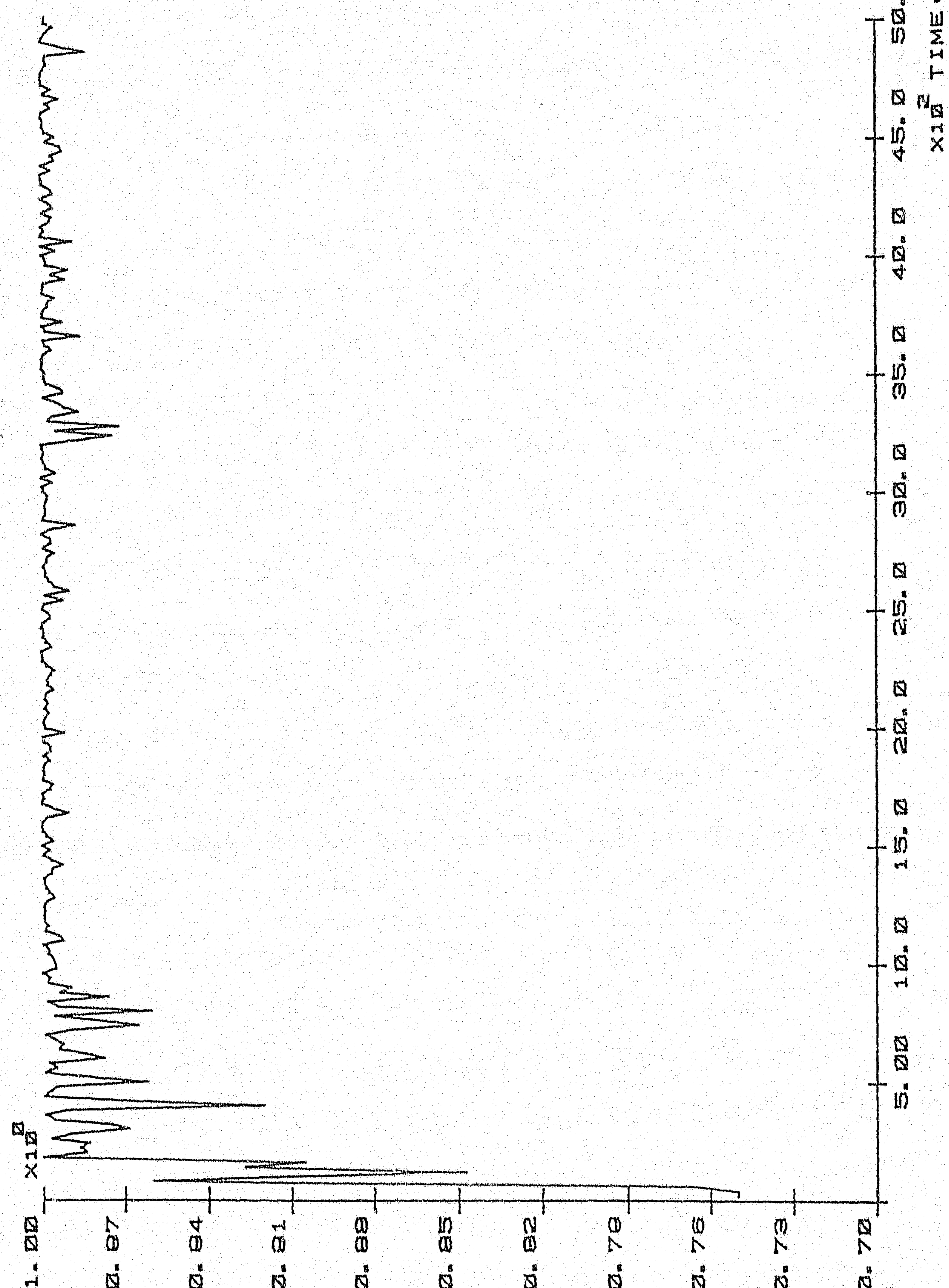
GRAPH 6.B FLOW-1




GRAPH 6.C LOOP-2
X10^0 LEVEL-2 (MTS)
0.50
0.45
0.40
0.35
0.30
0.25
0.20
0.15
0.10
0.05
0.00
LEVEL-2
SET PT.
5.00
10.0
15.0
20.0
25.0
30.0
35.0
40.0
45.0
50.0
X10^2 TIME (SECS)

GRAPH 6.D FLOW-2






GRAPH 6. E    F. F. -ADAPTATION

## 6.4. DISCUSSION

It might be argued that conventional PID laws will give acceptable results on processes such as the one under consideration, particularly if the methods outlined by Owens |17| & |29| or Porter |30| are used. However even then severe changes in interaction will not be accomodated. Satisfactory control will only be achieved if the controller parameters are periodically retuned by the operator. The self-tuner considered is a viable and stable alternative to periodic manual retuning. Self-tuning controllers are as robust (not sensitive to the unknown dynamics) as the ones obtained using the above strategies but provide better performance under varying dynamical situations.

Self-tuners are, therefore, intended to give superior control performance and hence their integrity is of great importance. A well designed self-tuner is one in which there is a robust control algorithm and diagnostic routines to ensure reliable estimation, such that it is relatively insensitive to incorrect parameter estimates.

With regards to controller integrity, the pole-assignment law used is thought to be the most reliable, trading robustness for optimality as was previously discussed (cf.Chapter 4). In the implemented self-tuner there is a simple classical back-up controller.

There are many causes of estimator failure, and since bad estimates lead to bad control, these must be dealt with. Section 3.5 discusses in detail the parameters involved from which conclusions about remedies on failure can be made. Some remedies modify the estimator while retaining estimation, others simply freeze the updating of parameters thus fixing the control law, in the worst case, as in the proposed scheme, a default controller is implemented and the tuning restarted from scratch. In the proposed system the short periods of poor control during the re-tuning transients can be tolerated as shown in the last section.

To yield its best performance, a self-tuner needs to be 'tuned' by the appropriate choice of parameters. Most of these parameters depend on the particular identification and control strategies employed which were already discussed in Chapters 3 and 4 respectively. Intuitive engineering judgements concerning sample rate, rate-limiting of set-point changes, closed-loop pole set locations are part of the choices left for the user. In particular experience has shown that the choice of initial estimates and of the covariance of the measurements have a marked effect on the final control configuration attained. The tailoring polynomial is also of great importance, it should be chosen with system capabilities in mind (ie. pumping and draining capacity).

On a whole the results show that the particular configuration attained, yields good performance in terms of the specifications included in Section 4.5. It must be stressed, however, that depending on the initial parameter estimates various (different) and still adequate control configurations may be achieved, and in this respect the self-tuner is very robust.

In the practical implementation of the self-tuner only the two following problems existed. Firstly the identification program had to be optimized to save computer calculation (delay) time, hence precluding the need to extend the $B(z^{-1})$ polynomial to take account of this. Secondly, the slightly oscillatory response is due to sensor noise which was, in fact, quite severe. Its effect was partially neutralised by taking the average of various readings.

It must also be noted that the variable forgetting factor extension to the normal recursive least-squares algorithm affected dramatically the rate of convergence of the parameters. Several experiments were performed with a constant forgetting factor and the convergence was totally inadequate, making it impossible to track time varying dynamics with reasonable 'tuning' transients.

As a final remark attention is drawn to the fact that the limited capacity of the top tanks in the coupled tanks system (cf. Fig. 2.1) warranted an upper bound on the maximum flow rate to prevent overflows of these tanks. This, obviously also degraded somewhat response times.

# CHAPTER 7

# CONCLUSION

Existing self-adaptive algorithms operate well on processes with constant but unknown parameters and were designed to solve the regulator problem |3|. Most industrial processes, in particular in the chemical/mineral industry, exhibit time varying and non-linear dynamics. The main objective of this research was to extend one of the existing multivariable algorithms to accomodate the above and to incorporate set-point tracking capabilities. This dissertation documents the theoretical developments and the significance, and limitations of the experience acquired in applying the algorithm to a pyhsical system.

The recursive least-squares estimator is able to adjust its parameter estimates in order to follow the system time varying dynamics. This is achieved by preventing the estimator from converging by introducing a forgetting factor. This forgetting factor is calculated 'on-line' to ensure a balance of information in the estimator. Hence at each sample the amount of forgetting is equal to the amount of new information in the latest measurement, estimation is therefore always based on the same amount of information.

The self-adaptive error-actuated controller uses a classical pole-assignment technique which is non-optimal but robust. The control objective is to move the closed-loop pole set to user-prespecified positions, which define the required response. This method is therefore intuitivelly more appealing and more generally applied than the optimal laws, however the complexity of the required computational effort is greater.

The algorithm was extensively tested in an actual physical system and also on a computer simulation. The various results are documented at the end of each Chapter. The following table, nevertheless summarizes the most typical malfunctions and the appropriate actions to nulify them.

| SYMPTOM | DIAGNOSIS | ACTION |
|---|---|---|
| a) Traces levelling off to a high level on system without noise | Too little information entering the estimator | Inject set-point changes and/or disturbances |
| b) Traces levelling off to a high value on system with noise | Self-tuner cannot tune. System order probably incorrect | 1. Change controller order<br>2. Change sampling interval |
| c) Traces low but control unsatisfactory (self-tuner accommodates this automatically!) | Self-tuner has converged to a suboptimal configuration | 1. Inject set-point changes and/or disturbances<br>2. Use random walks<br>3. Decrease forgetting factor |
| d) Unsatisfactory control but correct parameters and traces | | Change (decrease) $T(Z^{-i})$ |

The implemented self-tuner deals adequatelly with the plant non-linearities ie. operating-point and actuator/sensor non-linearities. The former is dealt with as time-varying parameters, since a change in operating conditions induces a corresponding variation in the locally linearised transfer function. Actuator non-linearities require supervisory software (ie. rate limits) to protect the basic algorithm.

The developed algorithm in addition to being an adaptive control method, can be regarded as a powerful on-line controller design tool. The fact that it provides a direct link between identification results and digital controller design is its main advantage. It is evident from this dissertation that a substantial amount of adjustment before an experiment is required by the self-tuner algorithm.

This research has indicated a number of associated areas where further work is required,

- Expansion of the suite to include other (optimal) control laws

- The behaviour of the algorithm in the presence of a time (transport) delay. Theory indicates that this can be accomodated by extending the model order |21| & |22|.

- Convergence properties of the algorithm. The self-tuner discussed was found to converge in a variety of cases. Neverthel it is felt that a more analytical approach is required especially involving the effect of the initial 'self-tuner parameter' choices upon the rate of convergence and the final achievable controller configuration.

- Although the structure of the suite is modular, enabling easy modifications, these have to be done to the particular subrontines involved. The suite could be made more user-friendly, especially as far as the simulator is concerned.

## REFERENCES

## PAPERS

|1| ALBERT, A. and R.W. SITTLER (1966)

'A method for computing least squares estimates that keep up with the data'

SIAM J. CONTROL, Vol 3.
Pgs 384-417 (6 refs)

|2| ASTROM, K.J. (1981)

'Theory and applications of Adaptive Control'

IFAC 8TH WORLD CONGRESS - KYOTO, JAPAN
Pgs 28-39 (50)

|3| ASTRÖM, K.J. and B. WITTENMARK (1973)

'On self-tuning Regulators'

AUTOMATICA, Vol 9
Pp. 185-199 (10)

|4| ASTROM, K.J. and P. EYKHOFF (1971)

'System Identification - A Survey'

AUTOMATICA, Vol 7
Pp. 123-162 (230)

|5| BORRISON, U. and R. SYDING (1976)

'Self-Tuning control of an ore-crusher'

AUTOMATICA, Vol 12
Pp. 1-7

|6| BORRISON, U (1979)

'Self-Tuning regulators for a class of Multivariable systems'

AUTOMATICA, Vol. 15
Pgs 209-215 (14)

|7| CEGRELL, T. and T. HEDQVIST (1975)

'Successful adaptive control of paper machines

AUTOMATICA, Vol 11
Pgs 53-59

|8| CLARKE, D.W. and P.J. GAWTHROP (1975)

'Self-Tuning Controller'

PROC. IEE, Vol 122 No. 9
Pp. 929-934 (12)

|9| CLARKE, D.W. (1981)

'Introduction to Self-Tuning Controllers'
Self-Tuning and Adaptive Control : Theory and Applications

IEE CONTROL ENG. SERIES 15
PETER PEREGRINUS
Pp. 36-71 (30)

|10| DUMONT, G.A. and P.R. BELANGER (1978)

'Self-Tuning Control of a Titanium-dioxide kiln!

IEEE TRANS. AUT. CONTROL, Vol. AC-23. No. 4
Pgs 588-592

|11| FORTESCUE, T.R.: L.S. KERSHENBAUM and B.E.YDSTIE (1981)

'Implementation of Self-Tuning Regulators with variable forgetting factors'

AUTOMATICA, Vol.17, No. 6
Pp. 831-835 (15)

|12| ISERMANN, R. (1982)

'Parameter adaptive control algorithms - a tutorial'

AUTOMATICA, Vol. 18, No. 5
Pp. 513-528 (55)

|13| ISERMANN, R.; H. KURZ and R. SCHUMANN (1980)

'Experimental Comparison and application of various parameter adaptive Control algorithms'

AUTOMATICS, Vol 16,
Pp. 117-133 (28)

|14| KEVICZY, L: J. HETTHESSY: M. HILGER and J. KOLOSTORI (1978)

'Self-Tuning adaptive control of cement raw material blending'.

AUTOMATICA, Vol. 14
Pp. 525-532.

|15| MACFARLANE, A.G.J. and J.J. BELLETRUTTI (1971)

'Characteristic Loci Techniques in Multivariable control system design'.

PROC IEE, Vol. 118, No. 9
Pp. 1291-1297 (12)

|16| MILLER, J.A.: A.M. LOPEZ; C.M. SMITH and P.W. MURRILL (1967)

'A Comparison of controller tuning techniques'.

CONTROL ENG - DECEMBER
Pp. 72-75 (4)

|17| OWENS, D.H. and A. CHOTAI (1982)

'High performance controllers for unknown multivariable systems.'

AUTOMATICA, Vol. 18, No 5
Pp. 583-587 (23)

|18| ROSENBROCK, H.H. (1969)

'Design of Multivariable control systems using the INA.'

PROC. IEE, Vol 116, No. 11
Pp. 1929-1936 (9)

|19| SASTRY, V.A.: D.E. SEBORG and R.K. WOOD (1977)

'A self-tuning regulator applied to a binary distillation column.'

AUTOMATICA Vol 13, No 4
Pp. 417-424

|20| WELLSTEAD, P.E. and D.L. PRAGER (1980)

'Multivariable Pole-Assignment Self-tuning regulators'

Proc. IEE, Vol. 128 (Pt.D), No 1
Pp. 9-18 (20)

|21| WELLSTEAD, P.E., D.L. PRAGER and P.ZANKER (1979)

'Pole-Assignment Self-Tuning Regulator'

PROC. IEE, Vol. 26, No 8
Pp. 781-787 (19)

|22| WELLSTEAD, P.E., J.M. EDMUNDS, D.L. PRAGER and P. ZANKER (1979)

'Self-Tuning Pole/Zero Assignment Regulators'

INT. J. CONTROL, Vol 30, No. 1
Pp. 1-26 (8)

|23| WELLSTEAD, P.E. and P. ZANKER (1979)

'Servo Self-Tuners'

INT. J. CONTROL, Vol. 30, No. 1
Pp. 27-36.

|24| WITTENMARK, B (1975)

'Stochastic Adaptive Control Methods : A Survey'

INT. J. CONTROL Vol. 21, No. 5.
Pp. 705-730 (85)

BOOKS

|25| ASTROM, K.J. (1970)
'Introduction to Sochastic Control Theory'
MATHEMATICS IN SCIENCE AND ENGINEERING - Vol. 70, ACADEMIC PRESS.

|26| DAVIES, W.D.T. (1970).
'System Identification for Self-Adaptive Control.'
J. WILEY.

|27| EYKHOFF, P. (1974).
'System Identification : Parameter and State Estimation.'
J. WILEY.

|28| KUO, B.C. (1980)
'Digital Control Systems.'
HOLT, RINEHART and WINSTON, INC.

|29| OWENS, D.H. (1977).
'Feedback and Multivariable Systems'
IFEE CONTROL ENGINEERING SERIES 7
PETER PEREGRINUS.

|30| PORTER, B (1981)
'Design of High Performance Tracking Systems'
LECTURE NOTES - UNIVERSITY OF SALFORD.

|31| WOLBERG, J.R. (1967)
'Prediction Analysis'
D. VAN NOSTRAND.

|32| USER GUIDE - REFERENCE MANUAL (1981)
'Advanced Continuous Simulation Language (ACSL)
Mitchell and Gunther Assoc. Inc.
Mass. USA.

# APPENDIX 1

## DISCRETE-TIME MODELS

Considering the following second-order Laplace domain transfer function, two possible z-domain equivalents arise:

$$G(s) = \frac{K}{(s+a)(s+b)} \qquad -(A.1.1)$$

(1) Modelling the sample + hold as a first order function plus a delay,

$$G(z^{-1}) = \frac{\beta_1 z^{-1} + \beta_2 z^{-2}}{1 + \alpha_1 z^{-1} + \alpha_2 z^{-2}} \qquad -(A.1.2)$$

where

$$\alpha_1 = -(e^{-aT_s} + e^{-bT_s})$$

$$\alpha_2 = e^{-(a+b)T_s}$$

$$\beta_1 = \frac{K}{2ab-a^2-b^2}(2-e^{-aT_s}-e^{-bT_s}+\frac{a}{b}(e^{-bT_s}+1)+\frac{b}{a}(e^{-aT_s}-1))$$

$$\beta_2 = \frac{K}{2ab-a^2-b^2}(2e^{-(a+b)T_s}-e^{-aT_s}-e^{-bT_s}+\frac{a}{b}(e^{-aT_s}(1-e^{-bT_s}))+\frac{b}{a}(e^{-bT_s}(1-e^{-aT_s})))$$

(2) Modelling the sample + hold as a pure delay of half the sampling internal,

$$G(z^{-1}) = \frac{\beta_1 z^{-1}}{1+\alpha_1 z^{-1} + \alpha_2 z^{-2}} \qquad -(A.1.3)$$

where

$\alpha_1$ and $\alpha_2$ are as for (1)

and,

$$\beta_1 = \frac{K}{b-a}(e^{-aT_s}-e^{-bT_s})$$

# APPENDIX 2

## SOFTWARE EXPLANATORY LISTING

```
C
C
C
C ***
      PROGRAM ADAPT
C ***
C
C     A.M.P.Ferreira
C     NOV-1983
C
C
C
C                  * * * * * * *
C                  * MAIN TASK *
C                  * * * * * * *
C
C     PURPOSE: DESIGN OF A SOFTWARE PACKAGE TO IMPLEMENT THE FOLLOWING
C              FIVE MODES OF OPERATION.
C                  1- SELF TUNER CONTROL
C                  2- CLASSICAL PID CONTROL
C                  3- MANUAL CONTROL
C                  4- NON-LINEAR T-DOMAIN SIMULATOR
C                  5- RECURSIVE LEAST-SQUARES ESTIMATOR
C
C     PROCEDURE: THE FOLLOWING TEST PACKAGE IS DIVIDED INTO FOUR
C                MODULES, WHOSE FUNCTIONAL AND DATA TRANSFER XISTICS
C                ARE EXPLAINED SUBSQUENTLY.
C                     MODULE 1: USER INITIALIZATION ROUTINE
C                     MODULE 2: SIMULATION ROUTINE
C                     MODULE 3: ESTIMATOR ROUTINE
C                     MODULE 4: CONTROLLER SYNTHESIS ROUTINE
C
C     PARAMETER DESCRIPTION:
C                           ALFI : Value of alfas' initial estimates
C                           AM   : X-Sectional area of mixing tanks
C                           AT   :       "          "  of top tanks
C                           BETI : Value of betas' initial estimates
C                           COV1 : Initial variance of alfa estimates
```

```
C                         COV2 :    "         "    of beta     "
C                         FF0  : Value of the memory length
C                         FF   : Memory length factor
C                         H    : Integration (t-incremental) step
C                         IEND : Termination flag
C                         INT  : Interval for data output
C                         ISEED: Initial seed for random generator
C                         ITN  : Initialization flag
C                         LL   : Lower bound on minimum level
C                         Q''H : Upper bound on input '' flow rate
C                         Q''L : Lower   "   on   "   ''   "    "
C                         Q1   : Flow(1) at time t
C                         Q2   :   " (2) at   "  t
C                         Q3   :   " (1) at   "  t-1
C                         Q4   :   " (2) at   "  t-1
C                         Q5   :   " (1) at   "  t-2
C                         Q6   :   " (2) at   "  t-2
C                         R    : Expected variance in level measurmt
C                         SP   : Required set-points
C                         T    : Time
C                         THTA : Vector of estimated parameters
C                         TK   : Vector of tanks' discharge coeffs.
C                         TLP  : Tailoring polynomial
C                         TRAC : Trac of confidence matrix
C                         TSP  : Sampling time interval
C                         TSP1 : Time interval for changes in input1
C                         TSP2 :   "      "       "      "   in input2
C                         UL   : Upper bound on state variables
C                         XQ'' : Random flow [0,1] for input ''
C                         YQ'' :   "      "  [-1,1] "    "    ''
C                         X(1) : Level of top-tank 1
C                         X(2) :   "   of  "    "  2
C                         X(3) :   "   of  mixing-tank 1 at time t
C                         X(4) :   "   of     "     "  2 at   "  t
C                         X(5) :   "   of     "     "  1 at   "  t-1
C                         X(6) :   "   of     "     "  2 at   "  t-1
C                         X(7) :   "   of     "     "  1 at   "  t-2
C                         X(8) :   "   of     "     "  2 at   "  t-2
```

```
C
C     REMARKS:
C             SUBROUTINES USED:
C                              1. BND   : Saturation routine
C                              2. DIAGN : Diagnostic routine
C                              3. ESTMT : Recursive plant estimator
C                              4. INPT  : Input routine
C                              5. INTG  : T-domain plant simulator
C                              6. NXTM  : Next-time calculation routine
C                              7. OUTPT : Output routine
C                              8. PLASS : Pole-assignment controller
```

```
C
C
C              * * * * * * * * * * * *
C              *      MODULE 1       *
C              * USER INITIALIZATION *
C              * * * * * * * * * * * *
C
C
C    PURPOSE: USER INITIALIZATION BLOCK
C
C    PROCEDURE: USER PROVIDES ALL NECESSARY PARAMETERS AT THE CONSOLE
C
C    PARAMETER DESCRIPTION:
C                           INTERNAL PARAMETERS: NONE
C
C                           EXTERNAL PARAMETERS:
C                           IGEN : Random number generator flag
C                           INFLG: Initialisation flag
C                                  0 := Initialise with PID control
C                                  1 :=      "        "  SELF-TUNER
C                                  2 :=      "        "  other method
C                           ISIM : Simulation flag
C                                  0 := On-line with plant
C                                  1 := On-line with simulator
C                           ITUNE: Controller flag
C                                  0 := PID control
C                                  1 := SELF-TUNER control
C
C
C ***
      SUBROUTINE DIAGN
C ***
C
C    PURPOSE: MONITOR THE LEVELS TO ENSURE THAT ANY FAULTY OPERATION
C             IS CLEARED BY REMOVING IN-FLOWS AND REVERTING TO MANUAL
C             MODE OPERATION.
C
C    PARAMETER DESCRIPTION:
```

```
C                               INTERNAL PARAMETERS: NONE
C
C                               EXTERNAL PARAMETERS:
C                                                  IEND
C                                                  Q1
C                                                  Q2
C                                                  X
C
C
C ***
      SUBROUTINE INPT
C ***
C
C
C     PURPOSE: SAMPLE PLANT AND INPUT LEVELS
C
C     PROCEDURE: HP2240 MEASUREMENT & CONTROL PROCESSOR APPLICATION
C                SOFTWARE IS USED.
C                REF: HP2240A
C                     1- USERS MANUAL
C                     2- INSTALLATION MANUAL
C                     Library Index Number: 2240A.005.02240-93001
C                     1978
C
C     PARAMETER DESCRIPTION:
C                               INTERNAL PARAMETERS: NONE
C
C                               EXTERNAL PARAMETERS:
C                                                 X
C
C
C ***
      SUBROUTINE NXTM
C ***
C
C     PURPOSE: CALCULATION OF THE TIME TO SAMPLE THE PLANT
C
C     PROCEDURE: USE OF HP1000 EXECUTIVE (REAL-TIME) FACILITIES.
```

```
C
C     PARAMETER DESCRIPTION:
C                                 INTERNAL PARAMETERS: NONE
C
C                                 EXTERNAL PARAMETERS:
C                                                               TSP
C
C
C ***
      SUBROUTINE OUTPT
C ***
C
C     PURPOSE: OUTPUT REQUIRED FLOW-RATES TO THE PLANT.
C
C     PROCEDURE: USE OF HP2240 SOFTWARE FACILITIES AS FOR INPT ROUTINE.
C
C     PARAMETER DESCRIPTION:
C                                 INTERNAL PARAMETERS: NONE
C
C                                 EXTERNAL PARAMETERS:
C                                                               Q1
C                                                               Q2
```

```
C
C
C              * * * * * * * * * * * *
C              *      MODULE 2        *
C              * T-DOMAIN SIMULATION *
C              * * * * * * * * * * * *
C
C
C ***
      SUBROUTINE INTG
C ***
C
C
C     PURPOSE: NON-LINEAR TIME-DOMAIN PLANT SIMULATION IMPLEMENTATION
C
C     PROCEDURE: FLOW IS INTEGRATED OVER ONE SAMPLE INTERVAL.
C                A 4TH ORDER RUNGE-KUTTA ROUTINE PERFORMS THIS
C                INTEGRATION IN 'NSTP' INTEGRATION STEPS.
C
C     PARAMETER DESCRIPTION:
C                           EXTERNAL PARAMETERS:
C                                                H
C                                                INT
C                                                ITN
C                                                T
C                                                TSP
C                                                X
C                           INTERNAL PARAMETERS:
C                                                D   : Vector of states
C                                                      time-derivatives
C                                                NSTP: No. intgr. steps
C                                                RES : Measmt. resolution
C                                                STOR: Storage registers
C
C     REMARKS:
C             SUBROUTINES USED:
C                              1. RGKT
C
```

```
C
C ***
      SUBROUTINE RGKT
C ***
C
C     PURPOSE: IMPLEMENTATION OF A 4TH ORDER RUNGE-KUTTA ALGORITHM
C              TO INTEGRATE OVER ONE COMMUNICATION INTERVAL.
C
C     PROCEDURE:
C                THIS SUBROUTINE IS ENTERED FOUR TIMES FOR EACH
C                INTEGRATION STEP , AT THE SECOND & FOURTH ENTRANCES
C                "T" BEING INCREMENTED BY "H/2".
C                AT A PRE-SPECIFIED FREQUENCY THE OUTPUTS ARE
C                CONTAMINATED WITH NOISE WITH A VARIANCE OF THE SAME
C                ORDER AS THAT PREDICTED FOR THE LEVEL MEASUREMENT.
C                THEORY REF: "BASIC NUMERICAL METHODS"
C                             T.J.YPMA : PP.269-235
C
C     PARAMETER DESCRIPTION:
C                           INTERNAL PARAMETERS:
C                                               K  : Vector Runge-Kutta
C                                                    4-point functions
C                                               XS : Vector of states
C                                                    at previous step
C                           EXTERNAL PARAMETERS:
C                                               H
C                                               T
C                                               X
C
C     REMARKS:
C              SUBROUTINES USED:
C                               1. SDER
C
C
C ***
      SUBROUTINE SDER
C ***
C
```

```
C     PURPOSE: IMPLEMENTATION OF PLANT DYNAMIC MODEL.
C              TIME-DERIVATIVES OF THE STATES ARE CALCULATED.
C
C     PROCEDURE: THE PLANT STATE EQUATIONS ARE USED TO CALCULATE
C                THE TIME-DERIVATIVES OF THE STATES AND THE RESULT IS
C                STORED IN D-VECTOR. THE INTEGRATION ROUTINES ARE THEN
C                USED TO CALCULATE THE CORRESPONDING STATE VALUES.
C
C     PARAMETER DESCRIPTION:
C                          INTERNAL PARAMETERS:
C                                          U : Flow-Level non-linear
C                                                  transfer function
C                          EXTERNAL PARAMETERS:
C                                          D
C                                          STOR
C                                          T
C                                          X
C
C     REMARKS:
C              SUBROUTINE USED:
C                          1. BND
C
C
C ***
      FUNCTION BND(X,UL,LL)
C ***
C
C     PURPOSE: SIMULATION OF A SATURATION NON LINEARITY
C
C     PARAMETER DESCRIPTION:
C                          INTERNAL PARAMETERS: NONE
C                          EXTERNAL PARAMETERS:
C                                          LL : Lower limit
C                                          UL : Upper limit
C                                          X  : Input/Output value
```

```
C
C
C                    * * * * * * * * * * * * * *
C                    *        MODULE 3         *
C                    * LEAST-SQUARES ESTIMATOR *
C                    * * * * * * * * * * * * * *
C
C
C ***
      SUBROUTINE ESTMT
C ***
C
C     PURPOSE: MULTIVARIABLE LEAST-SQUARES IDENTIFICATION ROUTINE
C
C     PROCEDURE: A RECURSIVE LEAST-SQUARES ALGORITHM IS USED TO IDENTIFY
C                THE PARAMETERS OF A SUITABLE PLANT MODEL. USE IS MADE
C                OF A VARIABLE FORGETTING FACTOR TO WEIGHT FRESH DATA
C                ACCORDING TO ITS INFORMATION CONTENT.
C                THEORY REF: " SYSTEM IDENTIFICATION - A SURVEY"
C                              K.J.ASTROM & P.EYKHOFF
C                              AUTOMATICA - VOL.7 - 1971
C
C     PARAMETER DESCRIPTION:
C                           INTERNAL PARAMETERS:
C                                                 C1  : Work array
C                                                 C2  :  "     "
C                                                 C3  : Estimator levels
C                                                 C4  : Error variance
C                                                 C6  : Correction factor
C                                                       for estimates
C                                                 C7  : Work array
C                                                 C8  :  "     "
C                                                 C10 : Correction factor
C                                                       for covariance
C                                                 FFA : Loop1 e**2/varnce
C                                                 FFB : Loop2  "  "    "
C                                                 FN  : Asymptotic memory
C                                                       length
```

```
C                                          FF1 : (ffa+ffb)=dim(out)

C                                          GAMA: Gamma matrix
C                                          PHY : Estimator gain
C                                          TERR: Error transpose
C                             EXTERNAL PARAMETERS:
C                                          ALFI
C                                          BETI
C                                          COV1
C                                          COV2
C                                          COV : Covariance matrix
C                                          ERR : Error in estimates
C                                          FFI : Forgetting factor
C                                          PSY : Measurement vector
C                                          R
C                                          THTA: Vector estimates
C                                          TPSY: Transpose of PSY
C                                          TRAC: Trace COV matrix
C
C     REMARKS:
C          SUBROUTINES USED:
C                             1. BND
C                             2. FRSH : Formation of measuremt vector
C
C ***
      SUBROUTINE FRSH
C ***
C
C     PURPOSE: FORMATION OF THE MEASUREMENT FROM IN/OUT DATA
C
C     PROCEDURE: PRESENT & LAST TWO SAMPLES OF DATA ARE ARRANGED
C                ACCORDING TO PRESPECIFIED PATTERN. THIS PATTERN
C                DEPENDS ON THE POSITION OF THE CORRESPONDING
C                ELEMENTS IN THE 'VECTOR OF ESTIMATES'.
C
C     PARAMETER DESCRIPTION:
C                             INTERNAL PARAMETERS: NONE
C
```

C EXTERNAL PARAMETERS: ALL

```
C
C
C                 * * * * * * * * * * * * * * * *
C                 *          MODULE 4          *
C                 * POLE-ASSIGNMENT  CONTROLLER *
C                 * * * * * * * * * * * * * * * *
C
C
C ***
      SUBROUTINE PLASS
C ***
C
C
C     PURPOSE: IMPLEMENTATION OF A 'CLASSICAL' POLE-ASSIGNMENT
C              CONTROLLER SYNTHESIS ROUTINE.
C
C     PROCEDURE: THE THEORETICAL CONCEPTS DISCUSSED IN CHAPTER 3 ARE
C                IMPLEMENTED TO OBTAIN AN ON-LINE ERROR ACTUATED
C                CONTROLLER.
C
C     PARAMETER DESCRIPTION:
C                           EXTERNAL PARAMETERS:
C                                               INT
C                                               Q''H
C                                               Q''L
C                                               Q1
C                                               Q2
C                                               Q3
C                                               Q4
C                                               SP
C                                               T
C                                               THTA
C                                               TLP
C                                               X
C
C                           INTERNAL PARAMETERS:
C                                               F1  : Controller matrix
C                                               GO  :      "        "
```

```
C                                                    G1   :      "        "
C                                                    All other matrices are
C                                                    work arrays.
C
C     REMARKS:
C              SUBROUTINES USED:
C                                   1. XINV : Matrix inversion
C                                   2. XMLT :     "   multiplication
C
C
C
C ***
      SUBROUTINE XINV(A,N)
C ***
C
C     PURPOSE: MATRIX INVERSION
C
C     PROCEDURE: THE GAUSS-JORDAN METHOD WITH COMPLETE PIVOTING IS USED.
C
C     PARAMETER DESCRIPTION:
C                              INTERNAL PARAMETERS:
C                                                    B   : Work array
C
C                              EXTERNAL PARAMETERS:
C                                                    A   : Result
C                                                    N   : Matrix order
C
C     REMARKS:
C             THEORY REF: "FORTRAN WITH ENGINEERING APPLICATIONS"
C                          D.D. McCRACKEN
C                          J. WILEY(1967)
C
C
C ***
      SUBROUTINE XMLT(A,B,C,L,M,N)
C ***
C
C     PURPOSE: MATRIX MULTIPLICATION
```

```
C
C     PARAMETER DESCRIPTION:
C                         INTERNAL PARAMETERS: NONE
C
C                         EXTERNAL PARAMETERS:
C                                             A : Left-hand matrix
C                                             B : Right- "      "
C                                             C : Result
C                                             L : No. rows of A
C                                             M : No. colmns of A
C                                             N : No. colmns of B
C
```

# APPENDIX 3

## USER INITIAL DIALOGUE

ENTER VALUE OF SIMULATION FLAG:

ENTER VALUE OF SELF-TUNNING FLAG:

ENTER PLANT INITIALISATION FLAG:

ENTER VALUE OF INITIAL FLOW RATE FOR BOTH MOTOR PUMPS:

SAMPLING INTERVAL,H & LU:

ENTER VALUE OF INITIAL ESTIMATES ,UNCERTAINTY & MEMORY LENGTH:

COV. OF LEVELS & MEMORY LENGTH FACTOR:

ENTER INTERVAL FOR DATA OUTPUT:

ENTER REQUIRED SET POINTS:

ENTER INITIAL TUNNING-IN TIME & TIME OFFSET:

ENTER TAYLORING POLYNOMIAL FOR LOOP(1):

ENTER TAYLORING POLYNOMIAL FOR LOOP(2):

ENTER FLOW BOUNDS FOR LOOP(1):

ENTER FLOW BOUNDS FOR LOOP(2):

# APPENDIX 4

## FORMAT OF DATA PRINTOUTS

```
C
C     DATA STORAGE EXPLANATORY LISTING
C
C
C     LOADR COMMAND TO PERFORM THE FOLLOWING
C
      $FILES(12,12)
C
C     OPENING OF REQUIRED UNITS
C
      OPEN(FILE='DAT1',UNIT=17,STATUS='NEW')
      OPEN(FILE='DAT2',UNIT=18,STATUS='NEW')
      OPEN(FILE='DAT10',UNIT=19,STATUS='NEW')
      OPEN(FILE='DAT11',UNIT=20,STATUS='NEW')
      OPEN(FILE='DAT30',UNIT=21,STATUS='NEW')
      OPEN(FILE='DAT20',UNIT=23,STATUS='NEW')
      OPEN(FILE='DAT21',UNIT=24,STATUS='NEW')
      OPEN(FILE='DAT12',UNIT=25,STATUS='NEW')
      OPEN(FILE='DAT22',UNIT=26,STATUS='NEW')
      OPEN(FILE='DAT50',UNIT=27,STATUS='NEW')
      OPEN(FILE='DAT51',UNIT=28,STATUS='NEW')
      OPEN(FILE='DAT52',UNIT=29,STATUS='NEW')
C
C     STORAGE OF IN/OUT DATA
C
      WRITE(17,5000)T,X(3),SP(1),Q1
      WRITE(18,5000)T,X(4),SP(2),Q2
5000  FORMAT(2X,I6,2X,G13.6,2X,G13.6,2X,G13.6)
C
C     STORAGE OF ESTIMATOR ROUTINE DATA
C
      WRITE(19,5000)T,FFI,THTA(1,1),THTA(3,1)
      WRITE(20,5005)T,THTA(5,1),THTA(7,1),THTA(9,1),THTA(11,1)
      WRITE(21,5000)T,TRACE,GAMA(1,1),GAMA(2,2)
5005  FORMAT(2X,I6,2X,G13.6,2X,G13.6,2X,G13.6,2X,G13.6)
      WRITE(23,5000)T,FFI,THTA(2,1),THTA(4,1)
      WRITE(24,5005)T,THTA(6,1),THTA(8,1),THTA(10,1),THTA(12,1)
```

```
      WRITE(25,5010)T,ERR(1,1),THTA(9,1)
      WRITE(26,5010)T,ERR(2,1),THTA(10,1)
5010  FORMAT(2X,I6,2X,G12.6,2X,G12.6,2X,G12.6)
C
C     STORAGE OF CONTROLLER ROUTINE DATA
C
      WRITE(27,4000)T,G0(1,1),G0(2,1),G0(1,2),G0(2,2)
      WRITE(28,4000)T,G1(1,1),G1(2,1),G1(1,2),G1(2,2)
      WRITE(29,4000)T,F1(1,1),F1(2,1),F1(1,2),F1(2,2)
4000  FORMAT(2X,I6,2X,G13.6,2X,G13.6,2X,G13.6,2X,G13.6)
C
C     CLOSING OF THE UNITS (only performed at end of run)
C                          (ie. when IEND=1             )
C
      CLOSE(17)
      CLOSE(18)
      CLOSE(19)
      CLOSE(20)
      CLOSE(21)
      CLOSE(22)
      CLOSE(24)
      CLOSE(25)
      CLOSE(26)
      CLOSE(27)
      CLOSE(28)
      CLOSE(29)
```

# APPENDIX 5

## ON-LINE USER INTERACTIVE DIALOGUE

```
C
C ***
      PROGRAM SETPT
C ***

C
C     PURPOSE: ON-LINE SET-POINT & OPERATIONAL MODE MODIFICATION.
C
C     PROCEDURE: SYSTEM COMMON BLOCK AREA IS UPDATED ON CORRECT
C                COMPLETION OF THE REQUIRED UPDATING.
C
C     PARAMETER DESCRIPTION:
C                                   INTERNAL PARAMETERS: NONE
C
C                                   EXTERNAL PARAMETERS:
C                                                        IEND
C                                                        IGEN
C                                                        INFLG
C                                                        ISIM
C                                                        ITUNE
C                                                        SP
C
ENTER NEW SET POINTS IN ORDER:
   0.00       0.00

ENTER TERMINATION FLAG:

ENTER NEW MODE OF OPERATION (0=NO CHANGE):

IS DATA CORRECT (0=NO;1=YES;2=NO CHANGE):
```