This algorithm was successfully tested with various values of Σ_o, the results are well documented in section 3.4.

As a point of interest the variable forgetting factor updating theory (equation (3.17)) is now further analysed under the influence of a incorrect value of the covariance matrix of the measurements. Suppose R is chosen incorrectly as kR, k a scalar, hence (3.17) becomes,

$$\lambda_{k+1} = 1 - \frac{1}{\Sigma_o \dim\{Y\}} \cdot e^T_{k+1} \left[kR + \psi^T_{k+1} P_k \psi_{k+1}\right]^{-1} e_{k+1} \qquad -(3.19a)$$

ie.

$$\lambda_{k+1} = 1 - \frac{1}{k\Sigma_o} \frac{1}{\dim\{Y\}} e^T_{k+1} \left[R + \psi^T_{k+1} \frac{P_k}{k} \psi_{k+1}\right]^{-1} e_{k+1} \qquad -(3.19b)$$

It is therefore concluded that the overall effect of choosing an incorrect value for R is to alter the effective memory length of the estimator as proved by (3.19b).

Whilst testing the algorithm on the simulation package, it was noted that since the noise level is low the process is of a near deterministic nature, hence very little dynamic information is available during steady-state operation. This would in fact give rise to problems if a constant forgetting factor was to be used, since old information is continually forgotten and little new information is acquired.

Experimental studies included in the next section show the basic operation of the algorithm. During steady-state operation the forgetting factor is close to unity and the estimator behaves like an infinite memory filter. When the system description is modified, resulting in a poor fit between the model and the actual process, and a smaller forgetting factor is calculated decreasing the effective memory length by placing more emphasis on the more recent data. After a while however, the estimator has retuned and the forgetting factor returns to its former value of unity.

Extensive tests were performed and the main results which were concluded are summarised below.

(i) Σ_0 small yielded a large confidence matrix and hence a very sensitive system. There is nevertheless a lower bound on Σ_0, as a very small value may again lead to a blowing-up of the P matrix.

(ii) Σ_0 large yielded a less sensitive estimator with slower adaptation characterisitics.

Apart from the required lower bound Σ_0 and hence λ, there does not appear to be any great sensitivity in the choice of Σ_0 for the algorithm as long as it is kept approximately consistent with the inverse of the system measurement noise. This last statement keeps consistency between Fortescue's SISO algorithm analysis and the MIMO algorithm developed for this project. Furthermore it was obtained that the size of the confidence matrix, for the numerous experiments varies around a constant value which is determined by the choice of R and Σ_0

One final point relevant to the estimator algorithm remains to be analysed, and that is the way in which the plant can initially be identified. Two methods were tested experimentally.

Firstly the recursive least-squares estimator was started whilst the plant simulation was under PID control. The estimator converged to adequate values after a tuning-in transient of the order of tens of iterations. Secondly the estimator was started by running the plant simulation under the following procedure. Set two demand levels, which should be different for better efficiency, input at full capacity until the outputs exceed their respective set points and then completely shut off the input until the output drops below the set point, and this procedure was repeated a number of times. The latter procedure yielded accurate parameters with a 'tuning-in' transient which was slightly less than that for the former procedure.

As can be concluded from analysing the results in section 3.4 the next factor, apart from Σ_0, which has a marked effect on the speed of adaptation of the estimator is the magnitude and frequency of the variations in the input signals. A universal random number generator was used to calculate these fluctuations. It is noted that when a system mode is not excited by the particular input then the corresponding parameter remains unaltered. The next time this mode is excited then the parameter is corrected and the relevant elements in the confidence matrix decreased until eventually convergence occurs.

The complete recursive least-squares multivariable estimator with weighting of past data can therefore be summarized by the following algorithm;

1. Formation of measurement vector ψ_{k+1} according to equations (3.7)

2. Error calculation:

$$e_{k+1} = Y_{k+1} - \psi_{k+1}^{T} \hat{\Theta}_k$$

3. Error covariance calculation:

$$\gamma_{k+1} = [R + \psi_{k+1}^{T} P_k \psi_{k+1}]^{-1}$$

4. Estimator-gain calculation:

$$\phi_k = P_k \psi_{k+1} \gamma_{k+1}$$

5. Estimates updating:

$$\hat{\Theta}_{k+1} = \hat{\Theta}_k + \phi_k e_{k+1}$$

6. Forgetting:

   - $\lambda_{k+1} = 1 - \frac{1}{2\Sigma_0} (e_{k+1}^{T} \gamma_{k+1} e_{k+1})$
   - Bound λ_{k+1}

7. Confidence matrix updating:

   - $P_{k+1} = \dfrac{[I - \phi_k \psi_{k+1}^{T}]P_k}{\lambda_{k+1}}$
   - Make P_{k+1} symmetric

The only difference between this and the original estimator given in (3.1) is step 6.

3.4. SIMULATION RESULTS

Preliminary tests on the recursive least-squares multivariable identification algorithm were performed via extensive simulation studies. Typical results are illustrated in the following graphs. A discussion related to these results is included in the next section.

Results of applying the estimator to the actual plant are included and discussed in Chapter 6.

A great advantage of the recursive least-squares algorithms is that P or in these results, the trace of P is available to give a guide to the variance of the parameter estimates. Hence the trace can be used to monitor the performance of the estimator.

GRAPH 3

IDENTIFICATION RESULTS

PARAMETERS :

Initial guesses $\alpha_o = -1$

$\beta_o = 1\ 000$

Memory length : $\Sigma_o = 750 * R$

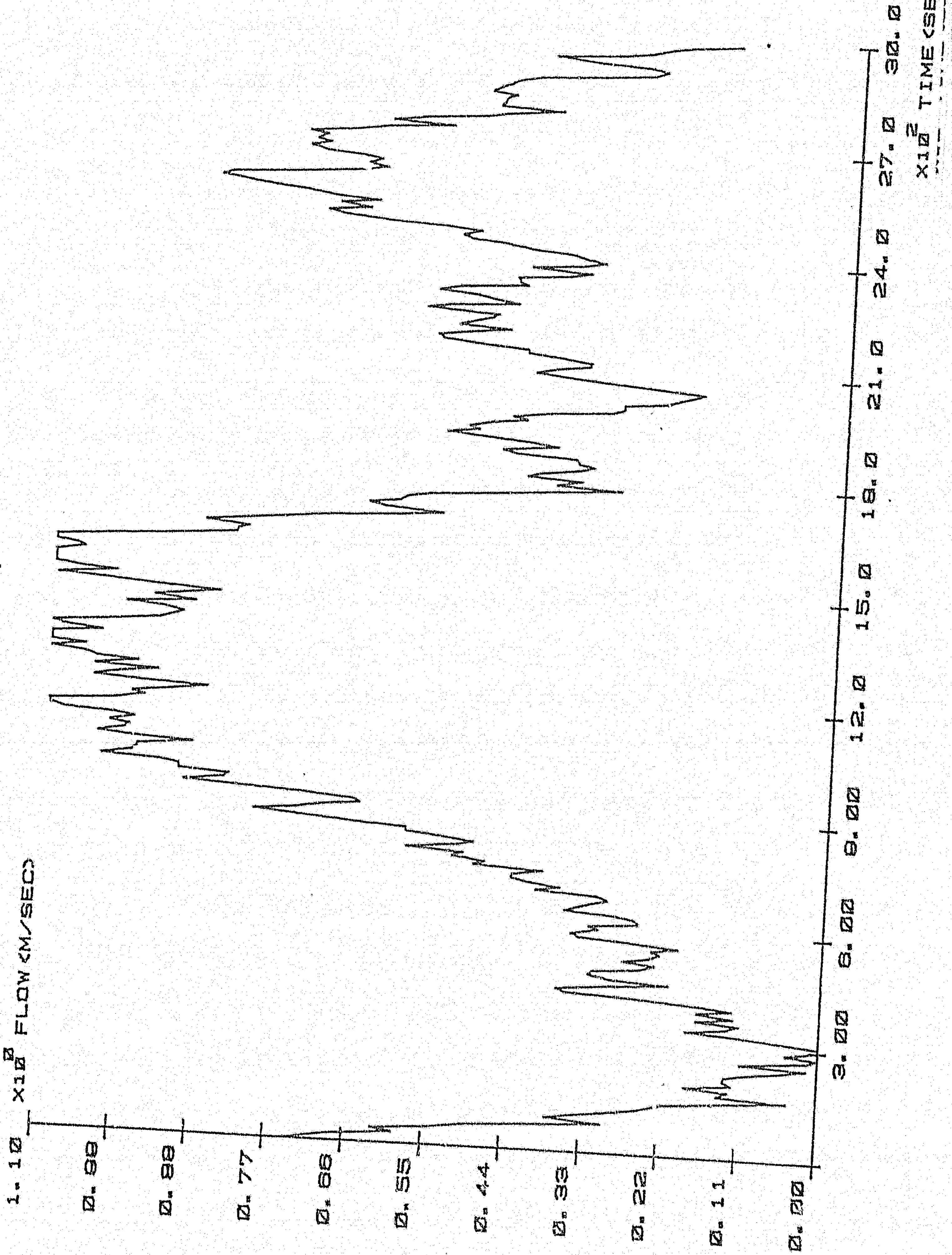
$R = 1E-8$

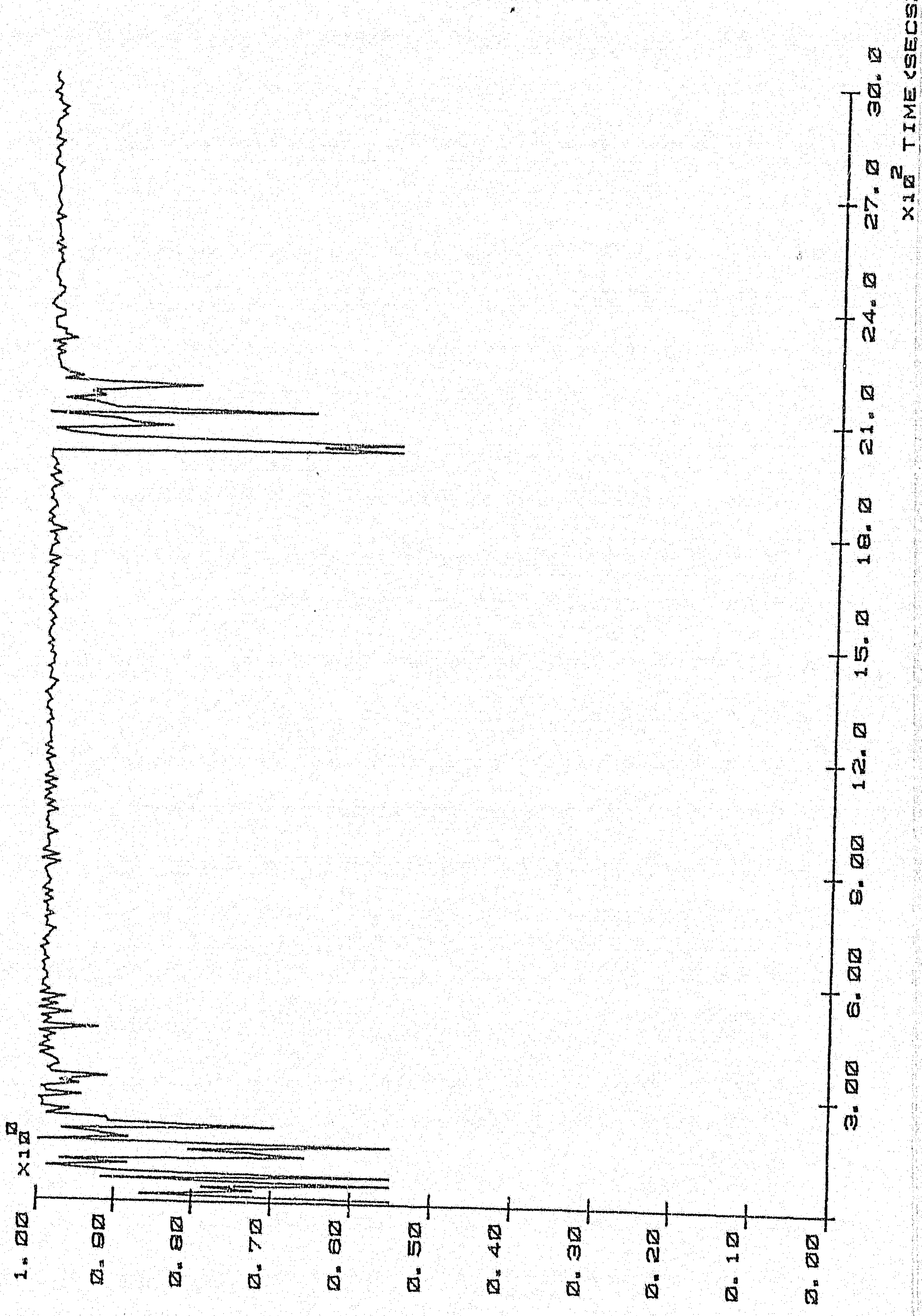
TEST CONDITIONS :

Plant changed from no-interaction ($K_{ij}(i \neq j) = 0$)

to swapped-inputs ($K_{ij}(i=j)=0$) at $t = 2\ 000$

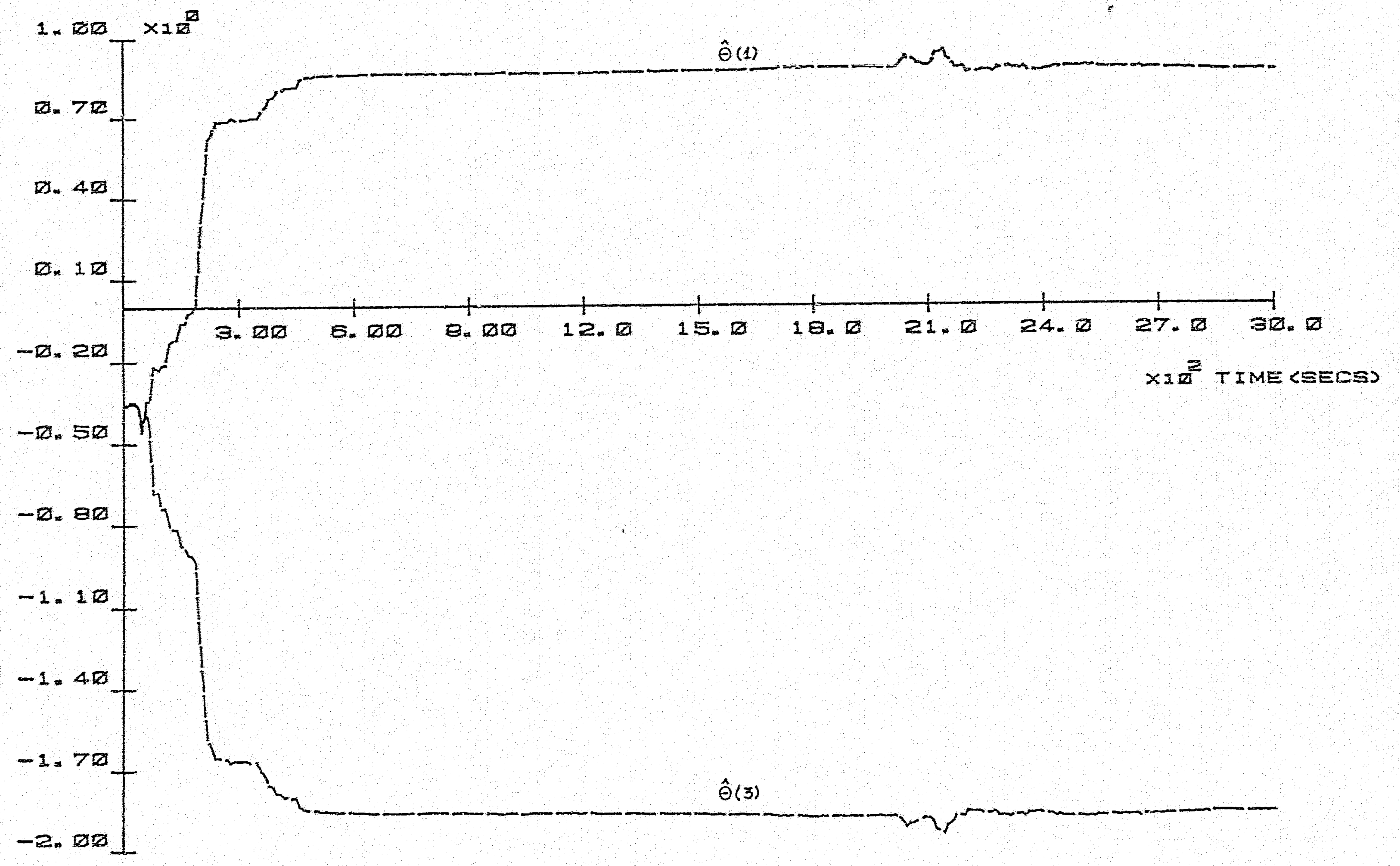
GRAPH 3.A INPUT-1



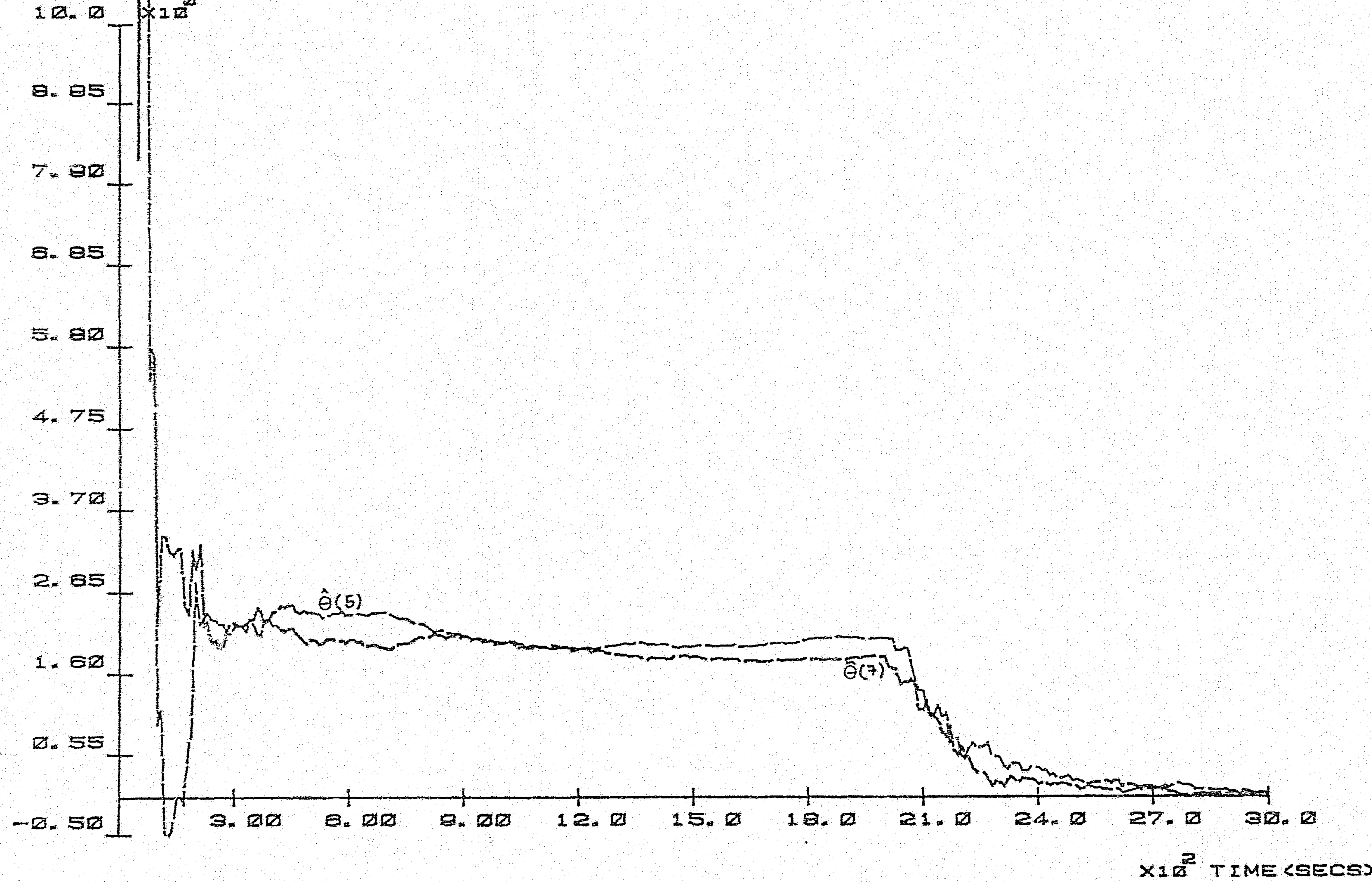


GRAPH 3.B FORGETING FACTOR ADAPTATION

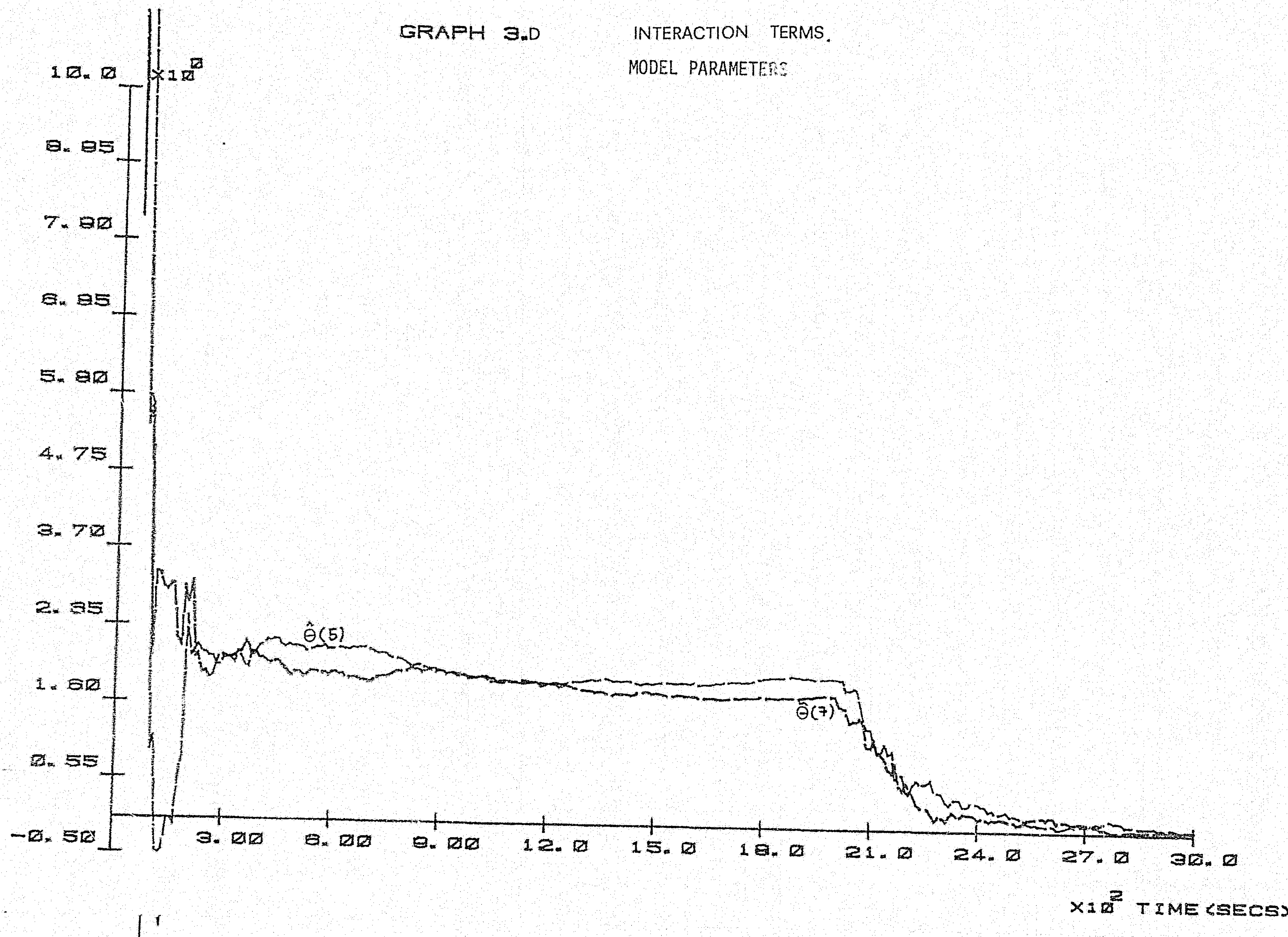
GRAPH 3.C . MODEL PARAMETERS (TIME-INVARIANT)



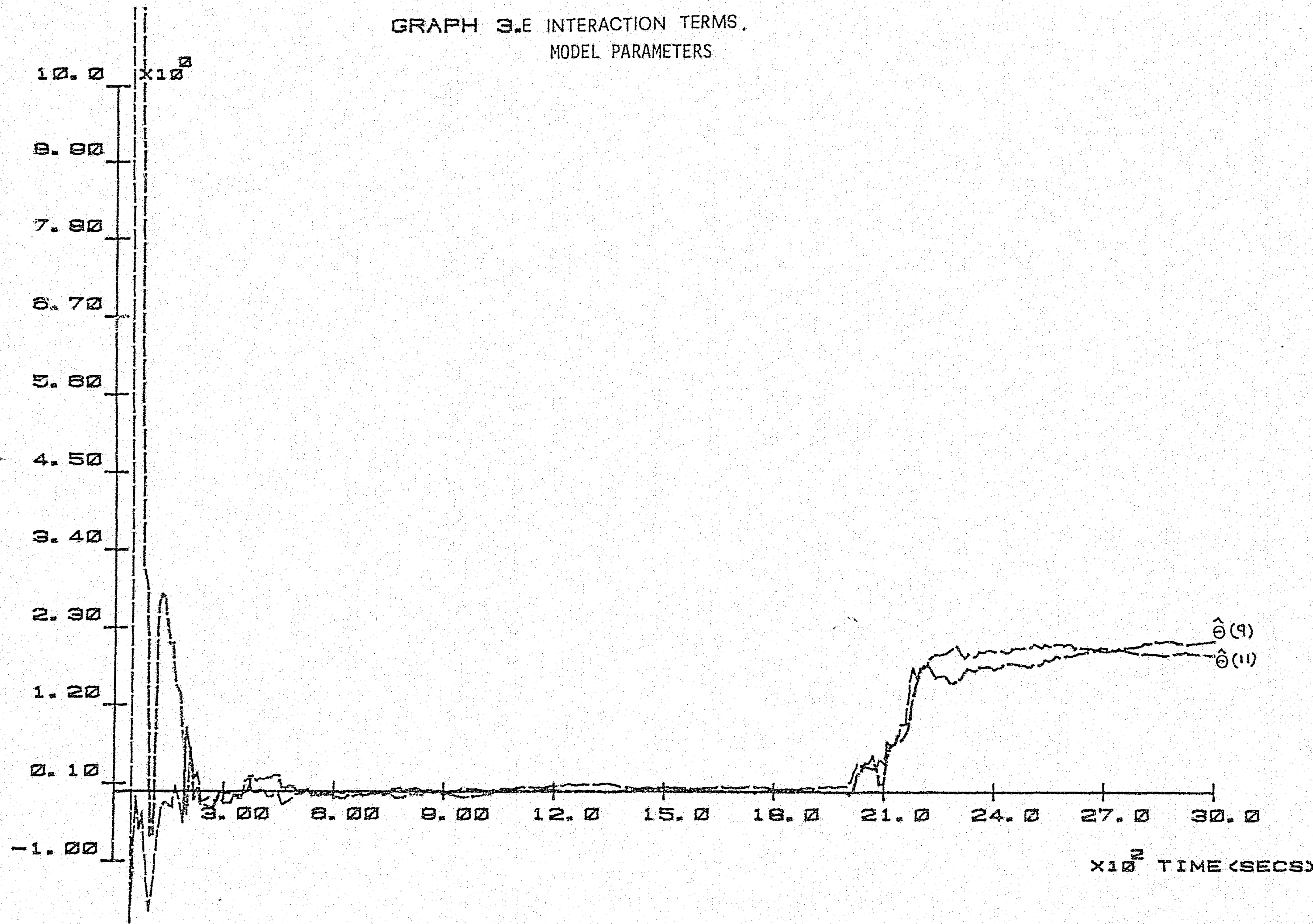
GRAPH 3.D INTERACTION TERMS.
MODEL PARAMETERS

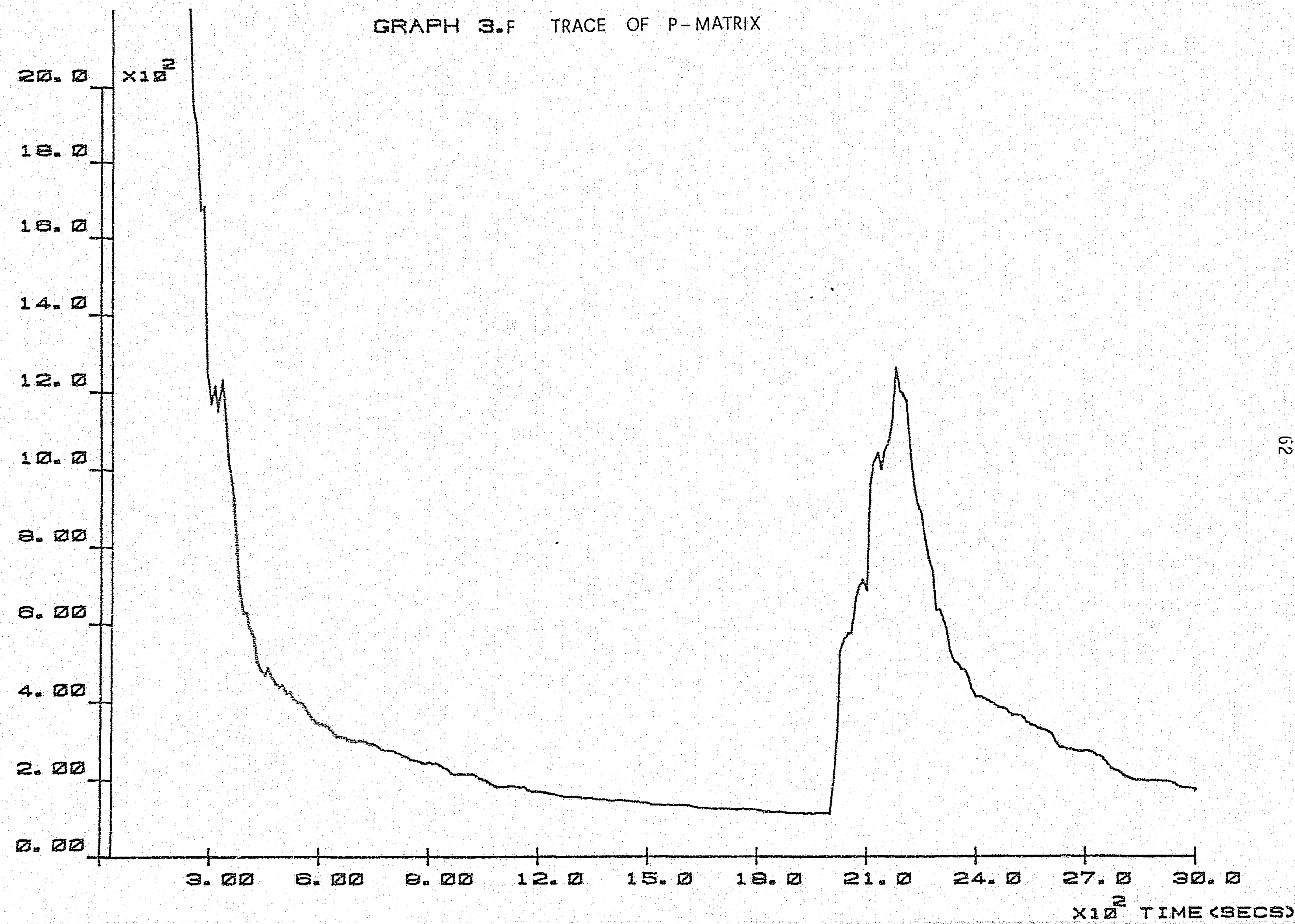


GRAPH 3.D INTERACTION TERMS.
MODEL PARAMETERS



GRAPH 3.E INTERACTION TERMS.
MODEL PARAMETERS





GRAPH 3.F TRACE OF P-MATRIX

GRAPH 3.G ESTIMATION ERROR

3.5. DISCUSSION

As explained in the previous two sections, there is a significant amount of tuning to be performed by the user of the identification algorithm. In particular, the following factors are directly decided upon by the user :

(i) The order of the system model :

The integers n_a and n_b equation (3.4) specify respectively the number of poles and zeros in the digital model of the unknown system. These must be selected according to the best available model of the system. These choices, as will be discussed in Chapter 4 will also affect the order of the controller to be implemented. It is important not to underestimate the system order which will lead to errc eous or divergent operation.

(ii) The initial estimate of the system :

The algorithm needs some apriori information about the sort of system it has to identify. This point has already been discussed in Section 3.2 and 3.3.

(iii) The initial confidence matrix :

As previously mentioned, at the first iteration of the algorithm an initial Po matrix is required. In the same way that the size of Po determines the degree of apriori knowledge about the system parameters, it also determines the way in which the searching for the correct parameters is conducted. A large value of P_0 gives a rapid search with large initial parameter fluctuations. A small P_0, however, gives a cautious search, which may in fact not be able to converge.

(iv) The forgetting factor:

The identification algorithm should be allowed to 'forget' poor information supplied to it as for example, during the initial stage. In section 3.3 a multivariable least-squares variable forgetting factor algorithm is developed, exploring the concept of information balance of the estimator.

(v) The number of integer time delays:

This has been omitted from the development in this Chapter, and is assumed to be zero. Moreover n_b must be increased to n_b+k_d to accommodate the delay. Obviously one of the tests to the identifier is to analyse how it copes with the plant delays, which are unknown in any case.

(vi) The information content of the estimator

The larger this value is (ie. Σ_0) the less pronounced the effect on the forgetting factor and hence the less sensitive the estimator is, resulting in slower adaptation. On the other hand the smaller Σ_0 the larger the sensitivity and the faster the adaptation. However, as explained in Section 3.3 there is a limit, under the control of the user to the value of Σ_0 to prevent instability and divergence of the confidence matrix.

(vii) Estimator dead zones:

Any parameter can be fixed by making the corresponding entry in P_0 zero. What can be done is to constantly monitor the trace of P_0 and if it is below a prespecified value the parameter vactor is frozen, but estimation continues in the same way to monitor the trace. The advantage of this is that, as will be seen later when the controller algorithm is formulated, the noise sensitivity of the controller is substantially reduced. Noise in this case incorporated the characteristics of transducers and actuators.

There are many causes of estimator failure, and it is important that they are dealt with by appropriate software, because bad parameters naturally lead to bad control. Some remedies |9| modify the estimator while retaining estimation (as the random walk method described in section 3.2), some simply freeze the updating of parameters so that the control law is fixed (as mentioned in (vii) above), whilst in the worst cases a default control is brought in and the tuning has to be restarted ie. the philosophy of graceful degradation. Some of the causes of, and solutions to failures are analysed subsequently.

(i) A signal does not persistently excite all modes of the system, implying that the confidence matrix will grow. If the signal under consideration is due to control saturation or transducer failure then the overall estimation should be stopped. The problem is to identify this condition.

(ii) The signals although exciting are not perturbing the plant. For example if there is a dead-zone in the actuator or transducer, then the estimated model between U and measured Y does not reflect the true process dynamics. The literature in this and the above (i) is minimal, one possible solution is to freeze the estimation when say the measurement is close to the setpoint. Another way is to completely ignore these problems (policy applied in this project) and then to see how the self-tuner handles them. It should be able to cope well since estimation theory indicates that the larger and more frequent the excitation, the more accurate the estimates, hence a self-tuner will 'learn' the most when it has made a mistake. This point is further analysed in a subsequent chapter.

(iii) The process is strongly non-linear, or subject to load disturbances. This should be seen by the estimator's ability to track the time-varying parameters, which is closely related to the variable forgetting factor method implemented.

Tests can be made on the P matrix and the model's prediction error. The relative size of the diagonal terms of P is indicative of how individual parameters are significant, and also the trace of P is used as an overall guide to estimation quality. Estimation integrity is of great importance if superior control performance is to be achieved and this is a topic requiring research beyond the scope of this dissertation. On a whole the estimator, as the results in section 3.4 illustrate, copes well with time varying parameters. The real test comes in Chapter 6 however, when estimator limitations and their significance in the presence of actuator and sensor non-linear maps are analysed.

CHAPTER 4

SYSTEM CONTROL

4.1. INTRODUCTION

The original idea of self-tuning control was introduced by Kalman in about 1958. Since then, many schemes using some kind of optimal design criterion have been developed. Aström |3|, for example, employs a minimum variance strategy or one of its suboptimal derivatives, minimizing the mean-square variability of the output variable. This was subsequently generalized by Clarke |8|. He employed a linear quadratic control law with an input weighting which allows a trade-off between the regulation and servo-properties of the closed loop system.

The self-tuning technique adopted in this project is a departure from the optimal-control methods mentioned above. In particular, it is due to Wellstead |20| and has its roots in classical control methods. The design rule allows the closed-loop pole and zero positions to be assigned, within certain limitations, to prespecified locations defining the required transient response. This approach has the same self-tuning property as do optimal regulators |22| but is more robust and has the intuitive appeal that control engineers can easily relate closed-loop pole positions to transient performance. Indeed it was this need to relate the performance of self-tuning systems to pratical limitations that in part stimulated this work.

Moreover, pole-assignment self-tuners have important practical advantages which go beyond the mere added insight. These will now be briefly analyzed by considering certain properties of minimum variance laws and their undesirable behaviour in the self-adaptive mode.

(i) Control limits

Minimum variance type control |3| & |8| usually leads to a rapidly varying control action, which can damage or impair the system input actuators. Pole-assignment, on the other hand, produces a much smoother control action, which is more acceptable in direct-digital control. Wellstead |22| illustrates this point with an example.

(ii) Non-minimum phase behaviour

Systems which may be considered as prime candidates to self-tuning often exhibit the following properties :

(a) non-minimum phase behaviour,

(b) unknown, possibly variable, time delay.

In such cases the positions assigned to the closed-loop zeros by optimal laws result in unsatisfactory performance. Optimal self-tuners attempt to cancel system dynamics, and this is a much less robust approach than modifying them as is done via pole-assignment strategies. Of course, optimal laws can also handle such systems but only by the introduction of appropriate modifications as explained by Clarke |8|.

Pole-assignment self-tuners are in many situations preferable to the optimal methods, mainly because of their added robustness and practicality at the expense of optimality.

In this chapter a survey of self-tuning control laws is included in the next section. Section 4.3 develops the main theory of the multivariable error-actuated pole-assignment controller. The last two sections include results on the simulation and comments.

4.2. DIGITAL CONTROL LAWS - A SURVEY

Consider a system which can be represented by the digital model outlined in Chapter 2,

$$Y_k = -A(z^{-1})Y_k + z^{-k}B(z^{-1})\ U_k + n_k \qquad -(4.1a)$$

$$n_k = C(z^{-1})\ e_k \qquad -(4.1b)$$

One of the purposes of applying feedback control to the system described by (4.1) will be to minimise the influence of n_k on the system output Y_k, ie. regulator problem. Further, it will usually be necessary to ensure that Y_k comes into steady state correspondence with a demanded reference input Y_r. Reference signal tracking, as is analysed later can be achieved by incorporating digitial integrators into the loop. From a practical viewpoint this relates to the well established technique of changing the system type number to ensure zero steady state error. In the following discussion only the regulator problem is considered.

The regulator is defined to be,

$$U_k = \frac{G(z^{-1})}{1+F(z^{-1})} Y_k \qquad -(4.2)$$

This is illustrated in Fig. 4.1. The regulator coeffiefficients (g_o.......g_{ng}; f ......f_{nf}) are selected according to the type of control required. At this stage one can discern two basic approaches, namely, optimal control via minimum variance regulation, and classical-type control via pole placement. The minimum variance related

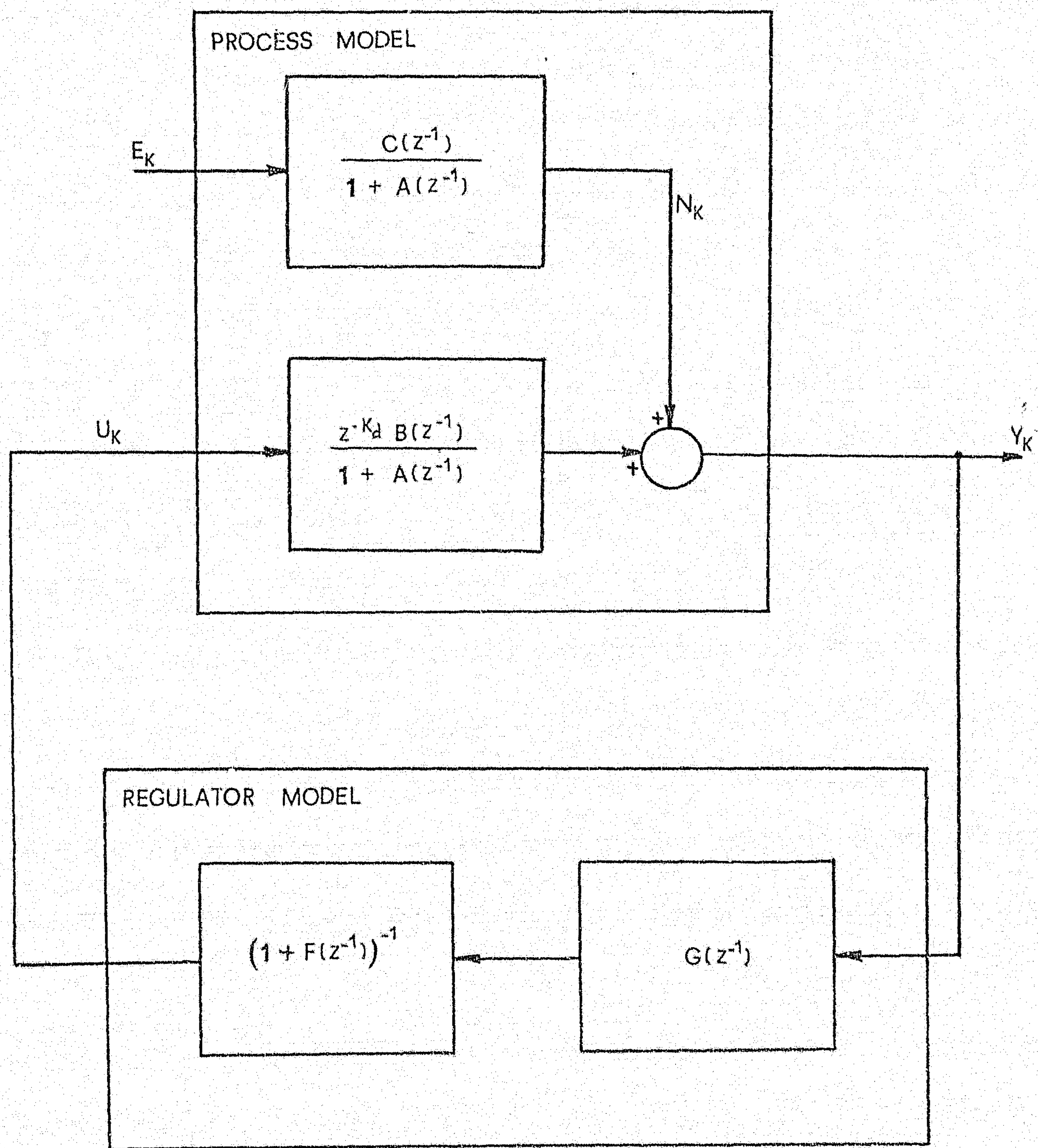


Fig. 4.1. : Regulator Scheme

strategies are described subsequently in a form which has been roughly transcribed from the original discrete-time single-input/single-output formulations due to Aström |3| and Clarke |8|.

4.2.1 Minimum Variance Control (Astrom 1970)

The aim of this regulation strategy is to force the output sequence Y_k to be a moving average of order K_d. The aim is to reduce the effect of n_k on the output Y_k. With a knowledge of $C(z^{-1})$ above the minimum variance regulator can be obtained by the solution of the following idenity,

$$z^{-K_d}C(z^{-1}) = (1+A(z^{-1}))\,(1+F(z^{-1})) + G(z^{-1}) \qquad -(4.3)$$

However by exploiting the self-tuning principles |3| and |21| it is possible to avoid this step by assuming $C(z^{-1})$ to be unity (ie white noise). If this assumption is made, and the system model of (4.1) is multiplied through by the $K_d{}^{th}$ order polynomial $(1+P(z^{-1}))$ which sets the first K_d terms of $A(z^{-1})$ to zero, one obtains the modified system model as,

$$Y_k = z^{-K_d}|-\tilde{A}(z^{-1})Y_k + (1+P(z^{-1}))B(z^{-1})U_k| + (1+P(z^{-1})e_k \qquad -(4.4a)$$

where

$$(1+P(z^{-1}))(1+A(z^{-1})) = 1 + z^{-K_d}\tilde{A}(z^{-1}) \qquad -(4.4b)$$

Hence there is, at the very least, a delay of K_d sample intervals in the feedback. So it is not possible to influence the disturbance for K_d time steps into the past. However the last term on the right of equation (4.4.a) above only involves K_d steps into the past.

This reasoning leads to the conclusion that the best one can do is to make,

$$y_k = (1+P(z^{-1}))e_k \qquad -(4.5)$$

This is the output under minimum variance control and can be obtained by selecting the regulator parameters so as to set the square bracketted term in (4.4a) to zero. Thus,

$$\frac{G(z^{-1})}{1+F(z^{-1})} = \frac{\tilde{A}(z^{-1})}{(1+P(z^{-1})B(z^{-1}))} \qquad -(4.6)$$

The following conditions are necessary:

(i) Polynomial $B(z^{-1})$ has all zeros inside the unit circle.

∴ Hence system (4.1) must be minimum phase.

(ii) Polynomial $C(z^{-1})$ has all zeros inside the unit circle.

These conditions are discussed at length in |25|.

4.2.2. Generalised (Detuned) Minimum Variance Control (Clarke, 1975)

The minimum variance control rule just described does not try to penalise excessive control action, it, therefore, results very frequently in large central signal fluctuations. These excursions can be reduced in two ways,

(a) by introducing slow poles in the closed-loop system - de-tuned minimum variance

(b) by weighting the mean square value of the control signal - generalised (Lambda) minimum variance

This reasoning leads to the conclusion that the best one can do is to make,

$$Y_k = (1+P(z^{-1}))e_k \qquad -(4.5)$$

This is the output under minimum variance control and can be obtained by selecting the regulator parameters so as to set the square bracketted term in (4.4a) to zero. Thus,

$$\frac{G(z^{-1})}{1+F(z^{-1})} = \frac{\tilde{A}(z^{-1})}{(1+P(z^{-1})B(z^{-1}))} \qquad -(4.6)$$

The following conditions are necessary:

(i) Polynomial $B(z^{-1})$ has all zeros inside the unit circle.
$\therefore$ Hence system (4.1) must be minimum phase.

(ii) Polynomial $C(z^{-1})$ has all zeros inside the unit circle.
These conditions are discussed at length in |25|.

4.2.2. Generalised (Detuned) Minimum Variance Control (Clarke, 1975)

The minimum variance control rule just described does not try to penalise excessive control action, it, therefore, results very frequently in large central signal fluctuations. These excursions can be reduced in two ways,

(a) by introducing slow poles in the closed-loop system - de-tuned minimum variance

(b) by weighting the mean square value of the control signal - generalised (Lambda) minimum variance

The former is easily achieved by making the response equal to,

$$Y_k = \frac{(1+P(z^{-1})}{1+z^{-k_d}\tilde{S}(z^{-1})} \; e_k \qquad -(4.7a)$$

where the roots of $1+z^{-k_d}\tilde{S}(z^{-1}) = \emptyset$ determine the extra moderating poles. To obtain this response the control should be calculated from,

$$\frac{G(z^{-1})}{1+F(z^{-1})} = \frac{\tilde{A}(z^{-1}) - \tilde{S}(z^{-1})}{\tilde{B}(z^{-1})} \qquad -(4.7b)$$

where

$$\tilde{B}(z^{-1}) = (1+P(z^{-1})\; B(z^{-1})) \qquad -(4.7c)$$

The alternative scheme whereby the control excursions are reduced, weights the mean square value of the control U_k by introducing a pseudo-output which is a linear function of past values of the system's input and output, thus,

$$Z_k = P(z^{-1})Y_k + Q(z^{-1})\; U_{k-k_d} \qquad -(4.8)$$

This allows the control excursions to be accounted for by the polynomial $Q(z^{-1})$. Moreover the polynomial $P(z^{-1})$ permits a form of detuning of the type described by equations (4.7).

In |8| Clarke derives a self-tuning controller which minimises a cost function containing terms like in (4.8), however, he also includes the set-point, thus ensuring set-point following capabilities.

Hence it can be summarised that a control design for minimizing the output variance is often inappropriate as it ignores the control effort required; moreover it is highly sensitive to the positions of the process zeros in discrete-time, which, as seen in Appendix 1 bear little relationship with their continuous-time counterparts. A solution to this problem is now discussed in the contenxt of pole-placement control.

4.2.3. Pole-Shifting Control : (Wellstead, 1976 and Truxal, 1955)

A severe problem with the controller design methods briefly discussed in the previous paragraphs is that to achieve performance they cancel system dynamics and are, therefore, highly sensitive to the positions of system zeros,. This is clearly illustrated by examining equations (4.6) and(4.7), where the roots of $B(z^{-1})$ are present among the poles of the regulator.

From the discussion on digital models for continuous time systems the point was made (and Appendix 1 will illustrate it for some value of sampling interval) that $B(z^{-1})$ will often have zeros outside the unit circle. These non-minimum phase zeros, which can also be caused by frational time delays |21| will evidently give rise to unstable poles in the regulator such that the closed-loop response can itself be unstable when under minimum variance control.

By the same token the number of integer time delays, k_d, may be unknown or even variable. If this is true the minimum variance strategies may experience problems if k_d in the model has been underestimated.

In other words these regulators not only ignore the control effort required, as mentioned, but also neglect the 'practical controlability' of the process.

The difficulties outlined in the previous two paragraphs can be avoided by the pole-shifting design method explained subsequently. The idea, in this method, is to concentrate entirely in the closed-loop poles, leaving the zeros alone.

Combining the model and regulator equations (4.1) and (4.2), the closed-loop characteristic equation is obtained,

$$|(1+A(z^{-1}))(1+F(z^{-1})-B(z^{-1})G(z^{-1})|Y_k = (1+T(z^{-1})e_k \qquad -(4.9)$$

where because it is assumed that k_d is unknown, the time delay has been absorbed into $B(z^{-1})$, ie. some of the leading coefficients of $B(z^{-1})$ may be zero.

Suppose that the desired closed-loop response has poles specified by the polynomial $1+T(z^{-1})$ such that the closed-loop system has the following response,

$$Y_k = \frac{(1+F(z^{-1})}{(1+T(z^{-1})}e_k \qquad -(4.10)$$

By inspection of the last two equations the following identity arises,

$$(1+A(z^{-1}))(1+F(z^{-1}))-B(z^{-1})G(z^{-1}) = 1+T(z^{-1}) \qquad -(4.11)$$

By equating coefficients in (4.11) a set of linear equations which can be solved for the unknown controller parameters, is obtained. In the next section a multivariable, set-point (ie. error actuated) version of this is formulated.

The pole-placement design strategy although non-optional, can be used safely on non-minimum phase systems or on systems with unknown time delays. In addition, it produces much gentler control action.

One possible difficulty is the numerical sensitivity with the resolution of (4.11), especially when that is an over-parameterization of the model. For example if $|1+A(z^{-1})|$ and $B(z^{-1})$ share a common root then the matrix used to solve for the controller coefficients is singular. As will be shown in the remainder of this chapter simulation experience has been good.

The method can be extended to the servo case (Wellstead |23|) by augmenting (4.11) to include two extra signals, one corresponding to a pre-compensated set-point and the other a feedforward signal which is added directly to the regulating control U_k. Further analysis is beyond the scope of this project.

4.2.4. The self-tuning property:

A self-tuner is a threefold process, measurement, identification and control, which continues at each step in time as suggested by Fig. 4.2 (for SISO regulator problem).

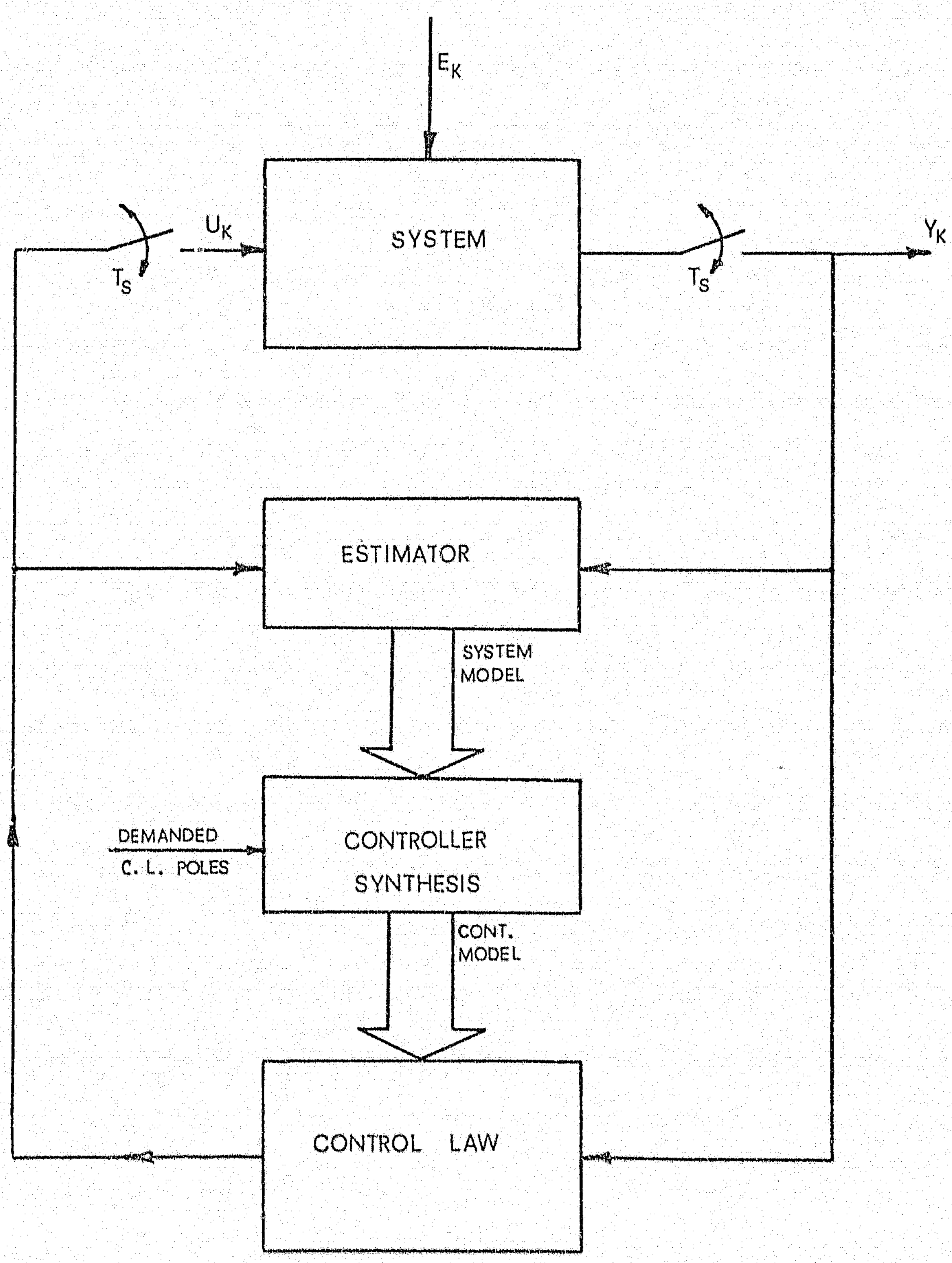


Fig. 4.2. Schematic of Self-tuning Regulator

As was previously mentioned, the self-tuner algorithm assumes that the system noise dynamics are white, such that if $C(z^{-1}) \neq 1$ then one would expect the closed-loop system to be incorrectly configured if and when it converges. However, Aström |3| has shown that for minimum variance type controllers the combined identifier/regulator can converge to the desired optimal regulator despite the incorrect assumption concerning the noise dynamics. Similarly Wellstead |21| and |22| shows that the above assumption/conversion also holds true for detuned variance and pole-assignment regulators.

The most relevant conditions for the success of this property (relevant in context to this work) are concerned with the order of the controller polynomials, hence,

$$n_F = n_\beta - 1 \qquad -(4.12a)$$

$$n_G = n_\alpha - 1 \qquad -(4.12b)$$

also,

$$n_T \geq n_A + n_B + k_d - 1 - n_C \qquad -(4.12c)$$

where all the variables in the right-hand side of the (4.12) were explained in the previous two chapters.

4.3 MULTIVARIABLE POLE ASSIGNMENT CONTROLLER

The system to be controled was described in Chapter 2 and is assumed to be controllable and observable. Its input/output matrix transfer function is deduced from equations (2.26) and (2.28) to be,

$$(I+\alpha(z^{-1}))Y_k = \beta(z^{-1})U_k + C_k + e_k \qquad -(4.13)$$

where Y_k, U_k and C_k are vectors of output, input and steady state values.

The control law is a multivariable version of (4.2). Since the output variables are to be controlled to a given set point, a digital integrator must be incorporated in the loop to ensure zero steady-state error. Also at steady-state the outputs are stationary and, in the context of the actual system, the input flows equal the output flows. All these conditions are met by the following control law,

$$U_k = U_o + \frac{G(z^{-1})}{(I+F(z^{-1}))}(Y_k - Y_{sp}) \qquad -(4.14)$$

where U_o and Y_{sp} are vectors of outflow steady state values and demanded set points respectively.

Further from equation (2.28) it is realized that,

$$C_{dc} = (I+\alpha(z^{-1}))Y_o - \beta(z^{-1}))U_o \qquad -(4.15)$$

Substituting equation (4.14) into (4.13) the closed-loop system becomes,

$$(I+\alpha(z^{-1}))Y_k = \beta(z^{-1})(U_o+G(z^{-1})(I+F(z^{-1})^{-1}(Y_k-Y_{sp}))+C_k+e_k \qquad -(4.16)$$

which upon substitution from (4.15) yields,

$$(I+\alpha(z^{-1}))(Y_k-Y_o) = \beta(z^{-1})G(z^{-1})(I+F(z^{-1})^{-1}(Y_k-Y_{sp})+e_k \qquad -(4.17)$$

Hence if one lets $Y_o = Y_{sp}$, which is reasonable since at steady state there should be zero steady state error, this last equality can be modified to yield,

$$|I+\alpha(z^{-1}) - \beta(z^{-1})G(z^{-1})(I+F(z^{-1}))^{-1}|(Y_k-Y_{sp}) = e_k \qquad -(4.18)$$

As was explained in the previous section, the closed-loop transfer function of the system under pole-assignment is given by the following extension of (4.10),

$$(Y_k-Y_{sp}) = (I+F(z^{-1}))(I+T(z^{-1})^{-1}e_k \qquad -(4.19)$$

Equating the latter two equations yields the pole-assignment equation,

$$(I+\alpha(z^{-1})(I+F(z^{-1}) - \beta(z^{-1})G(z^{-1}) = I+T(z^{-1}) \qquad -(4.20)$$

The self-tuning property in the context of the multivariable case just described is proved by Wellstead |20|. The method is basically a means of designing 'on-line' a controller without explicitly determining the matrix polynomial defining the noise characteristics ie. $C(z^{-1})$. The self-tuning procedure can thus be briefly described as,

(i) At each iteration estimate the plant parameter polynomials $\alpha(z^{-1})$ and $\beta(z^{-1})$ as in Chapter 3 using recursive least-squares, so the ξ_k becomes the sequence of fitting errors.

(ii) Solve the pole-shifting equation (4.20) such that control can be calculated via equation (4.14).

The self-tuning principle |20| then states that, if the system converges, it converges to the closed-loop system that would have been obtained had the parameters $A(z^{-1})$, $B(z^{-1})$ and $C(z^{-1})$ been known apriori. The output converges to (4.19). Furthermore the residual sequence ξ_k will converge to the true system driving noise e_k.

The algorithm presented here is intended to solve the set-point tracking problem and is depicted in block diagram form in Fig. 4.3. Simulation studies have shown (see Section 4.4) that the algorithm exhibits good convergence properties, however further theoretical investigation is required in this area.

The work presented so far in this chapter has not considered the problem of loop-interaction mentioned in Section 2.4. Analysing the controller parameters, the multivariable self-tuner presented here, does not, in general, lead to a decoupled or diagonally dominant controller as in Rosenbrock |18|. The following observations are, nevertheless worth mentioning.

(i) The degree of inter-loop coupling is dependent upon the polynomial matrices $F(z^{-1})$ (ie. closed-loop zeros) and $T(z^{-1})$ (ie. closed-loop poles) as in equation (4.19). However since the latter can be chosen, without loss of generality to be diagonal, it is $F(z^{-1})$ that gives the crosscoupling terms.

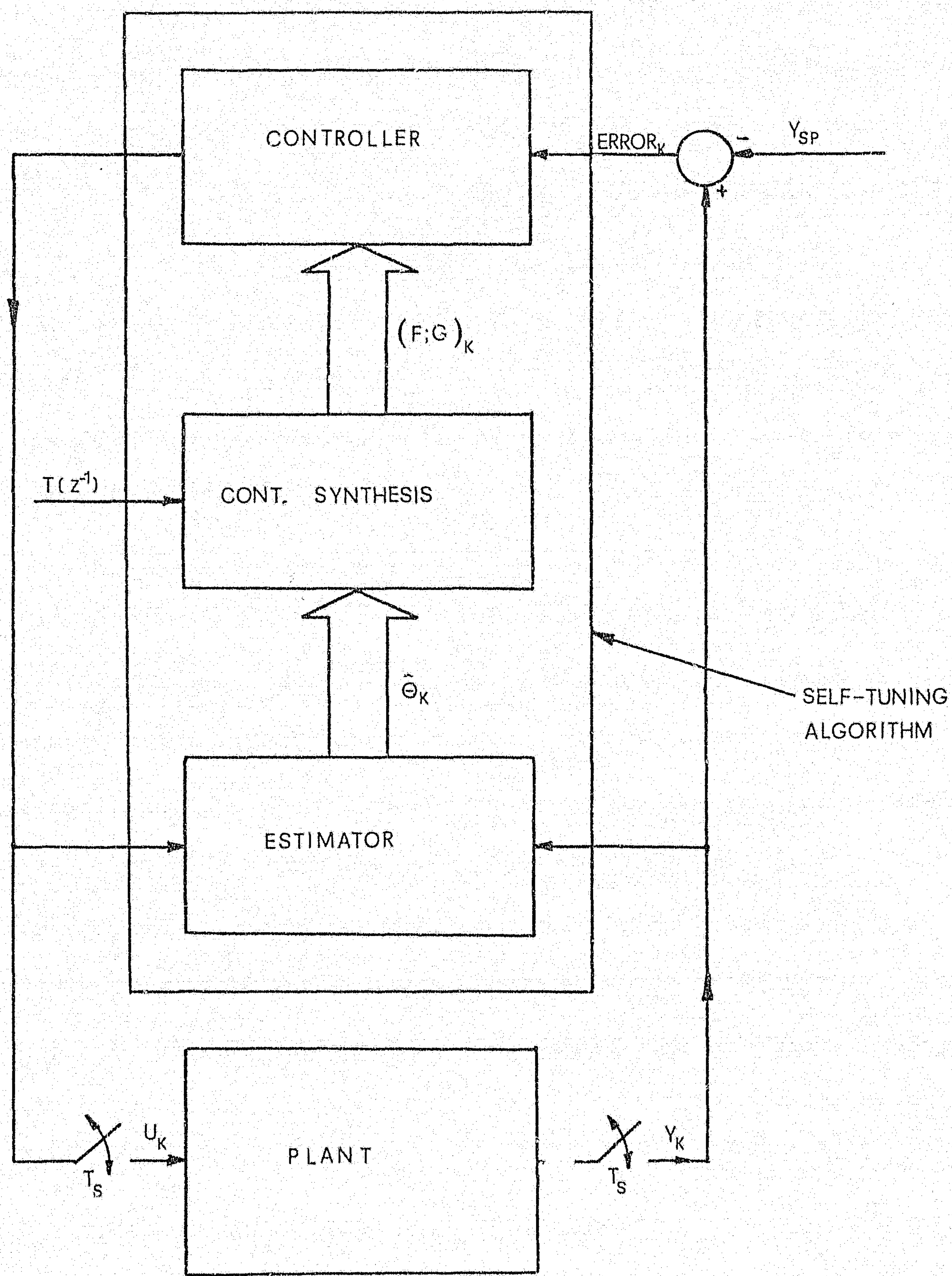


Fig. 4.3 : Multivariable self-tuning controller scheme

(ii) In the particular case where the polynomial matrix $B(z^{-1})$ is first-order (as in the first simulation examples . Section |4.4|) the $(I+F(z^{-1}))$ matrix reduces to the unity matrix (by (4.12a)) and hence complete inter-loop decoupling is achieved.

(iii) Integral action is included in all controller loops to achieve zero steady-state errors. Hence it can be concluded that loop crosscoupling is only transient in nature because integral action effectively 'takes out' the errors. This is particularly important when case (ii) above is not true. It can therefore be said that under any circumstances the loops are decoupled in the sense that their steady-state errors are independent and zero.

A better solution format for the pole-shifting equation (4.20) for the two cases when the polynomial matrix $B(z^{-1})$ is first or second order (ie. $F(z^{-1})$ is null or first-order respectively) is now presented.

Considering first the case when $B(z^{-1})$ and hence $\beta(z^{-1})$ is first order, therefore equations (4.12) yield,

$$\begin{aligned} \beta(z^{-1}) &= \beta_1 z^{-1} \\ G(z^{-1}) &= g_0 + g_1 z^{-1} \\ \alpha(z^{-1}) &= \alpha_1 z^{-1} + \alpha_2 z^{-2} \\ T(z^{-1}) &= t_1 z^{-1} \end{aligned} \qquad -(4.21)$$

Re-arranging (4.20) and writing the resultant equation in element form, one obtains,

$$\begin{bmatrix} \beta_1 & \beta_2 \\ \beta_3 & \beta_4 \end{bmatrix} z^{-1} \left(\begin{bmatrix} g_{11} & g_{12} \\ g_{13} & g_{14} \end{bmatrix} z^{-1} + \begin{bmatrix} g_{01} & g_{02} \\ g_{03} & g_{04} \end{bmatrix} \right) = \left(\begin{bmatrix} \alpha_1 & \emptyset \\ \emptyset & \alpha_2 \end{bmatrix} - \begin{bmatrix} t_1 & \emptyset \\ \emptyset & t_2 \end{bmatrix} \right) z^{-1} + \begin{bmatrix} \alpha_3 & \emptyset \\ \emptyset & \alpha_4 \end{bmatrix} z^{-1} \qquad -(4.22)$$

which can easily be solved as,

$$\begin{bmatrix} g_{11} & g_{12} \\ g_{13} & g_{14} \end{bmatrix} = \begin{bmatrix} \beta_1 & \beta_2 \\ \beta_3 & \beta_4 \end{bmatrix}^{-1} \begin{bmatrix} \alpha_1 & \emptyset \\ \emptyset & \alpha_4 \end{bmatrix} \qquad -(4.23a)$$

and,

$$\begin{bmatrix} g_{01} & g_{02} \\ g_{03} & g_{04} \end{bmatrix} = \begin{bmatrix} \beta_1 & \beta_2 \\ \beta_3 & \beta_4 \end{bmatrix}^{-1} \begin{bmatrix} \alpha_1 - t_1 & \emptyset \\ \emptyset & \alpha_2 - t_2 \end{bmatrix} \qquad -(4.23b)$$

Control is then applied via (4.14) at each sample interval.

In the case when $\beta(z^{-1}) = \beta_1 z^{-1} + \beta_2 z^{-2}$ then the controller polynomial matrix $F(z^{-1})$ becomes $f_1 z^{-1}$, and a similar analysis of equation (4.20) yields the following equating like powers of z^{-1} in either side of the equation,

$$\begin{aligned} &z^{-1} \text{ powers :} && \beta_1 g_0 - f_1 = \alpha_1 - t_1 \\ &z^{-2} \text{ powers :} && \beta_1 g_1 + \beta_2 g_0 - \alpha_1 f_1 = \alpha_2 - t_2 \qquad -(4.24) \\ &z^{-3} \text{ powers :} && \beta_2 g_1 - \alpha_2 f_1 = \emptyset \end{aligned}$$

which can be written in a more convenient form as,

$$\begin{bmatrix} \beta_1 & \beta_3 & \emptyset & \emptyset & -1 & \emptyset \\ \beta_7 & \beta_5 & \emptyset & \emptyset & \emptyset & -1 \\ \beta_2 & \beta_4 & \beta_1 & \beta_3 & -\alpha_1 & \emptyset \\ \beta_8 & \beta_6 & \beta_7 & \beta_5 & \emptyset & -\alpha_2 \\ \emptyset & \emptyset & \beta_2 & \beta_4 & -\alpha_3 & \emptyset \\ \emptyset & \emptyset & \beta_8 & \beta_6 & \emptyset & -\alpha_4 \end{bmatrix} \begin{bmatrix} g_{01} & g_{02} \\ g_{03} & g_{04} \\ g_{11} & g_{12} \\ g_{13} & g_{14} \\ f_{11} & f_{12} \\ f_{13} & f_{14} \end{bmatrix} \doteq \begin{bmatrix} \alpha_1 - t_1 & \emptyset \\ \emptyset & \alpha_2 - t_2 \\ \alpha_3 - t_3 & \emptyset \\ \emptyset & \alpha_4 - t_4 \\ \emptyset & \emptyset \\ \emptyset & \emptyset \end{bmatrix} \qquad -(4.25)$$

This is easily solved if the 6x6 matrix is non-singular. Singularity was never encountered during the extensive simulation tests performed. The control law is also governed by (4.14) but now is slightly more complicated as shown below,

$$\begin{bmatrix} U^1{}_k \\ U^2{}_k \end{bmatrix} = \begin{bmatrix} f_{11} & f_{12} \\ f_{13} & f_{14} \end{bmatrix} \begin{bmatrix} U^1{}_{o_{k-1}} \\ U^2{}_{o_{k-1}} \end{bmatrix} - \begin{bmatrix} f_{11} & f_{12} \\ f_{13} & f_{14} \end{bmatrix} \begin{bmatrix} U^1{}_{k-1} \\ U^2{}_{k-1} \end{bmatrix} + g_o\,\Delta Y_k + g_1\,\Delta Y_{k-1} + U_{o_k} \qquad -(4.26a)$$

where,

$$\Delta Y_k = Y_k{}^i - Y_{sp}^i$$

and

$$\Delta Y_{k-1} = Y^i_{k-1} - Y^i_{sp} \quad , \qquad i = 1,2 \qquad -(4.26b)$$

As in fixed term control, adaptive controllers must also take in consideration the limitations of the systems they control, the system non-linearties can cause them to detune. Control saturation is the most usual factor. In the prepared scheme software control limits fall within the actuator limits, (ie. maximum pumping capacity) and are reflected back into the estimator.

Furthermore as the original algorithm developed by Wellstead |20| was to solve the regulator problem, the proposed algorithm modification causes the self-tuner to determine whenever there is a step demand change. Hence to avoid too much self-tuner detuning and to ensure that such demanded changes were within the system's capabilities, the level reference inputs (Y_{sp}^{i}, i=1,2) were rate limited.

The overall controller algorithm can now be summarized as follows,

At each simple interval,

(i) Calculate difference between measured and demanded outputs (ie. $Y_k^{i} - Y_{sp}^{i}$, i = 1,2) and if necessary, rate limit any change in Y_{sp}.

(ii) Use estimator parameter vector ($\hat{\Theta}_k$) to calculate the control signal (U_k) via (4.14)

(iii) Limit U_k and apply.

4.4. SIMULATION RESULTS

Extensive simulation tests were performed on controller order, control objective (ie. the polynomial $T(z^{-1})$), set point rate limits and control actuator bounds.

A detailed discussion of the most important facets of the results is included in the next section.

GRAPH 4

SIMULATION/CONTROLLER RESULTS

PARAMETERS :

Initial guesses : $\alpha_o = -1$

$\beta_o = 1\ 000$

Memory length : $\Sigma_o = 600 * R$

$R = 1E-8$

Tailoring polynomial : $t_1 = -0{,}775$ for both loops.

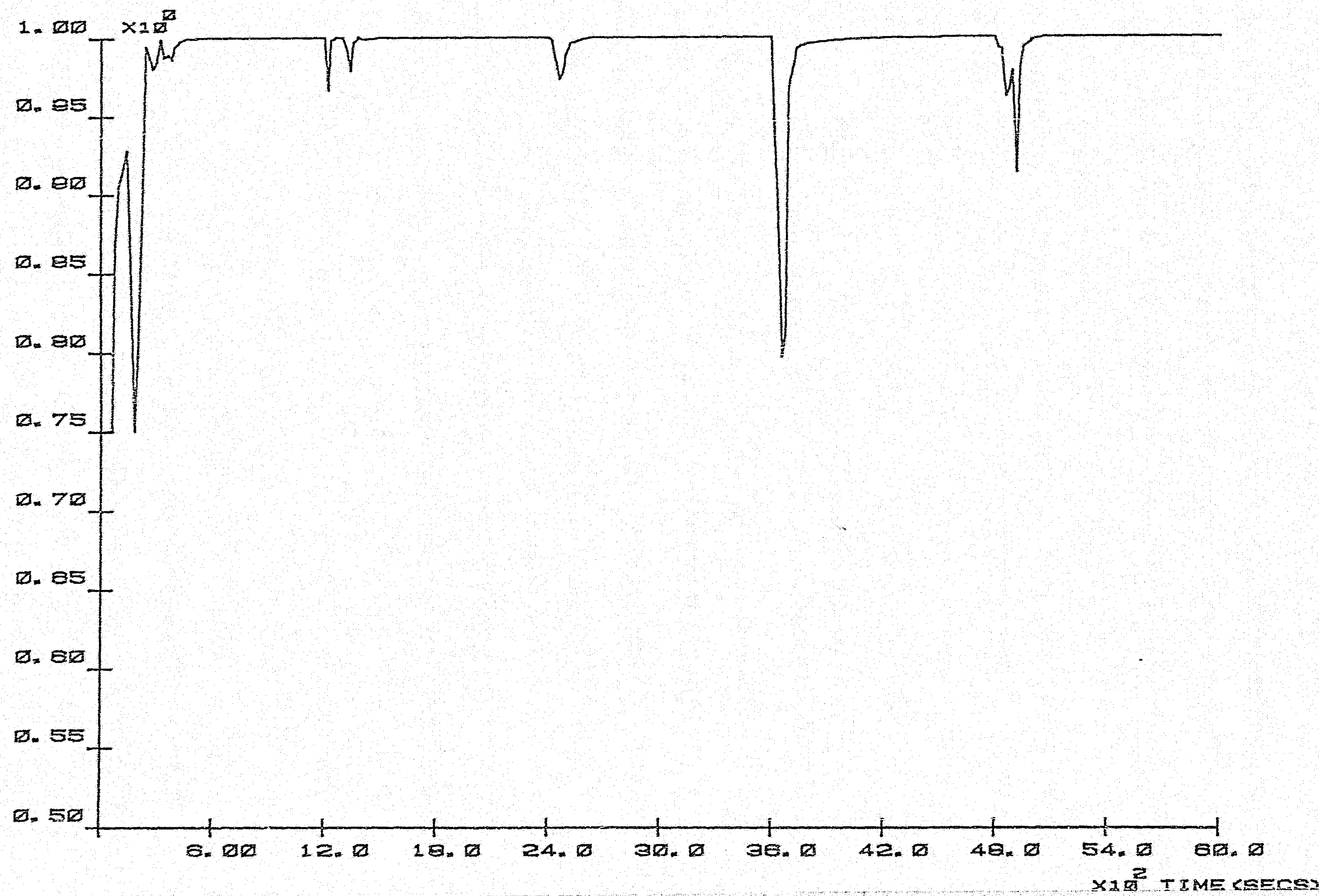
TEST CONDITIONS :

Initialisation stage : 0-300 secs

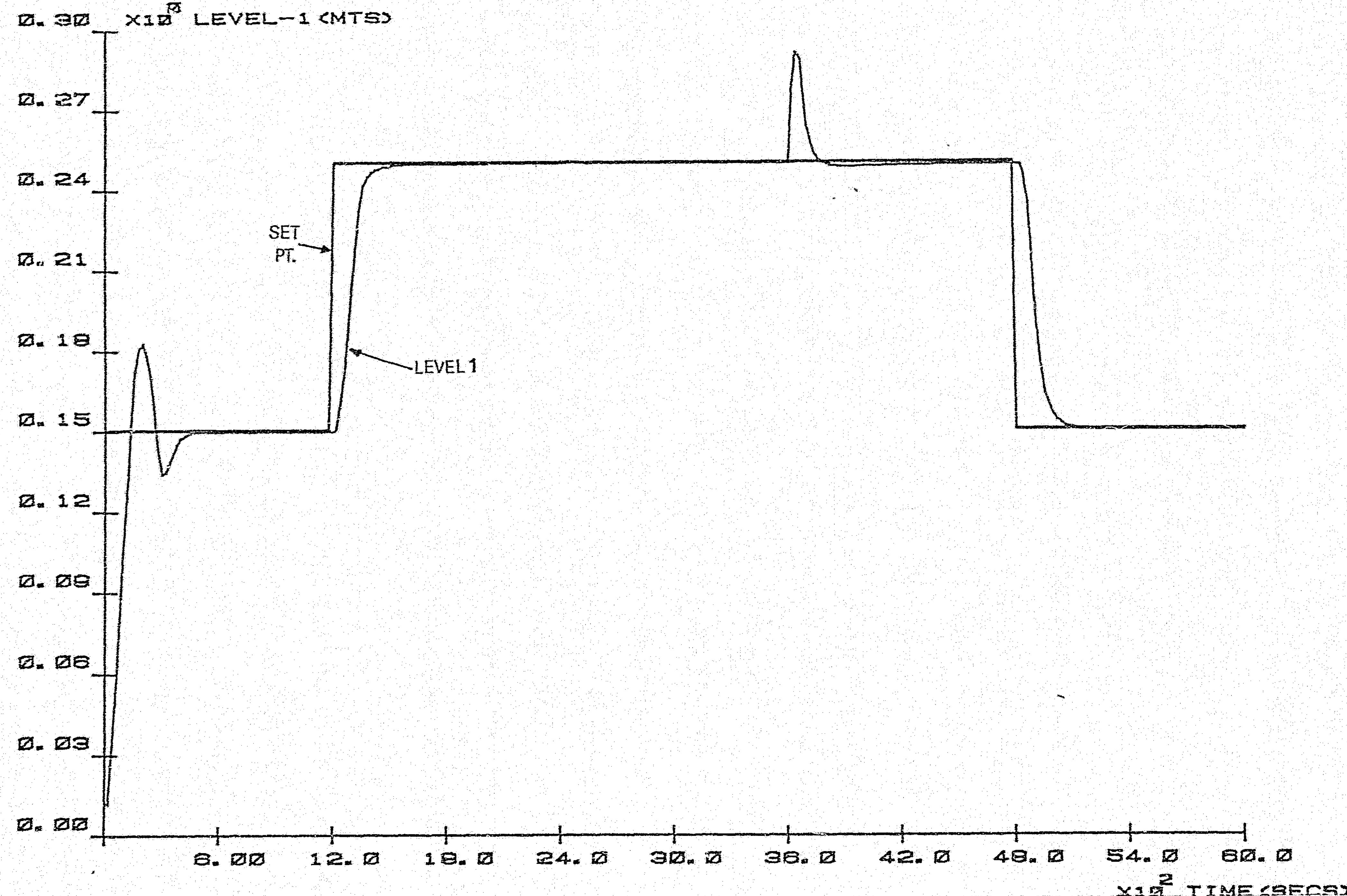
At 1 200 - Plant modified. (increase interaction)

At 4 800 - Plant modified. (increase interaction)

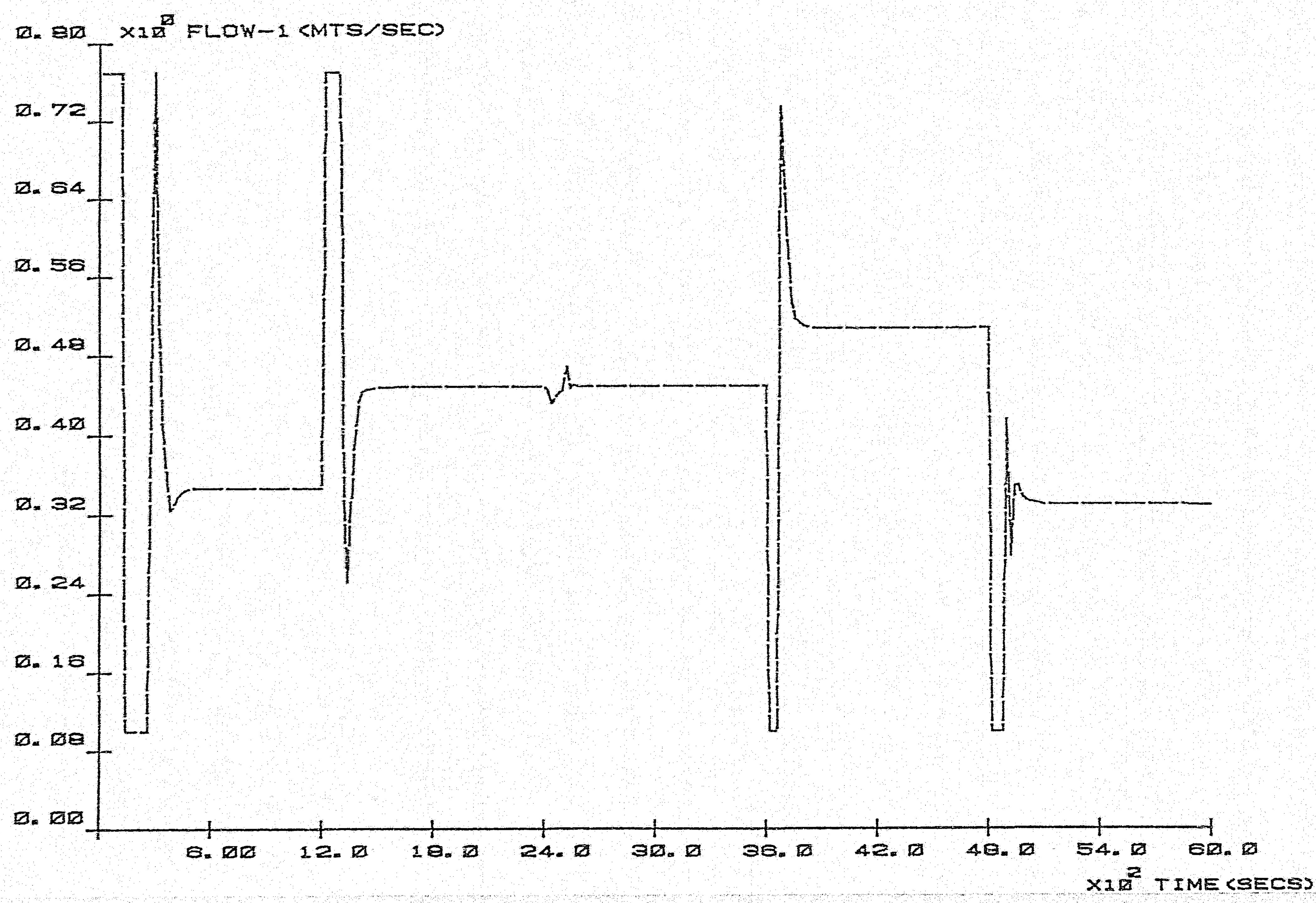
GRAPH 4.A F.F-ADAPTATION

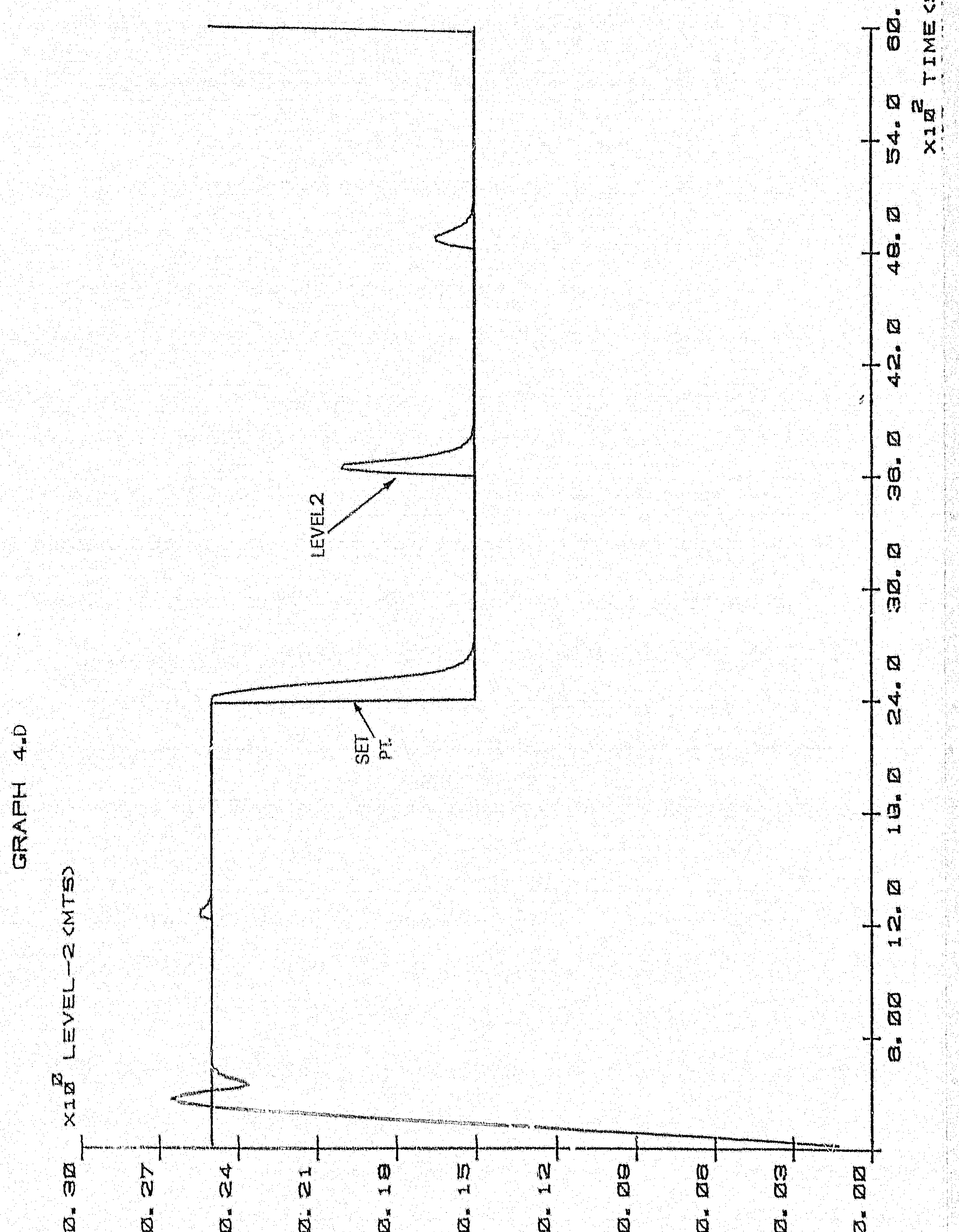


GRAPH 4.8h. LOOP 1

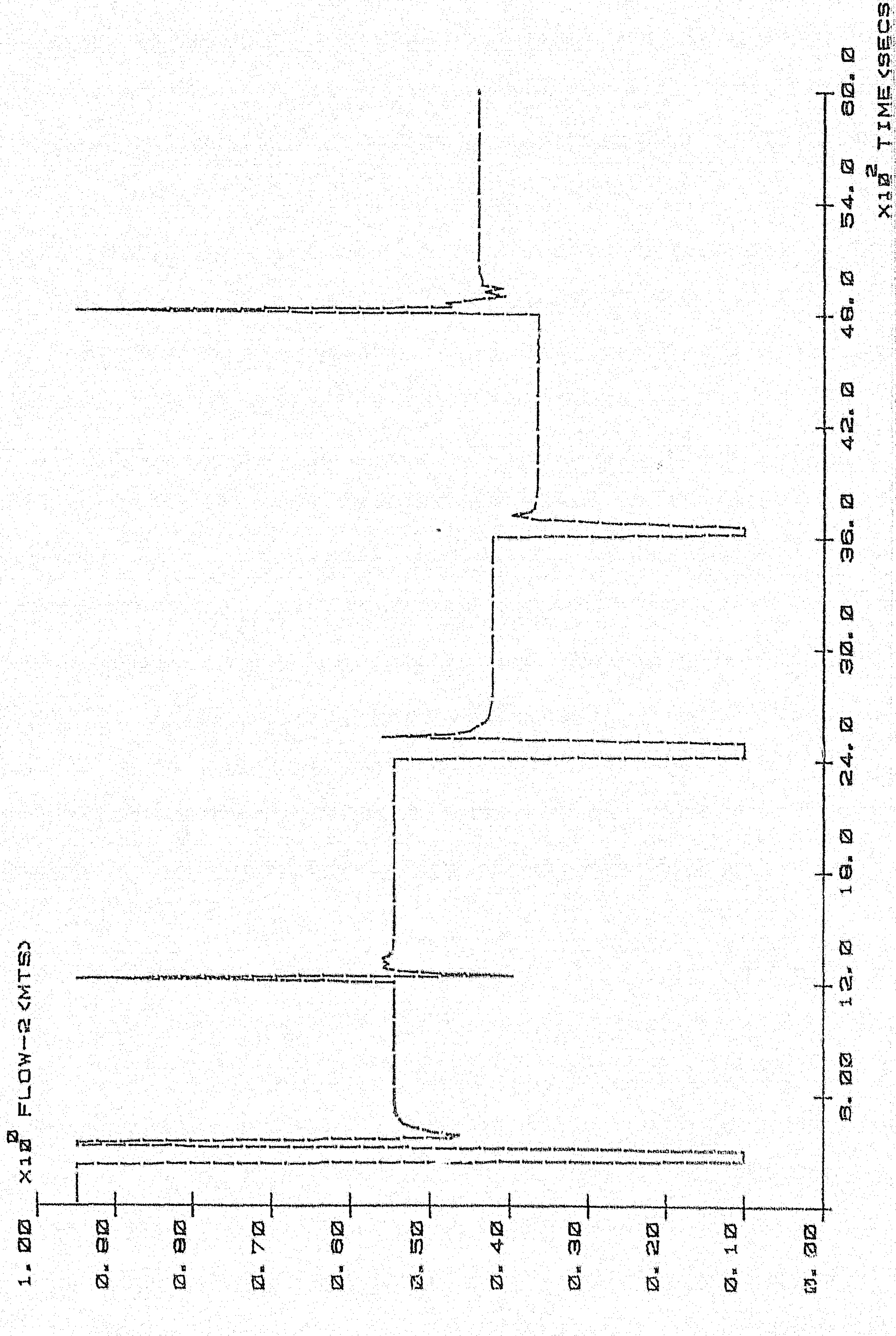


GRAPH 4.C' FLOW1

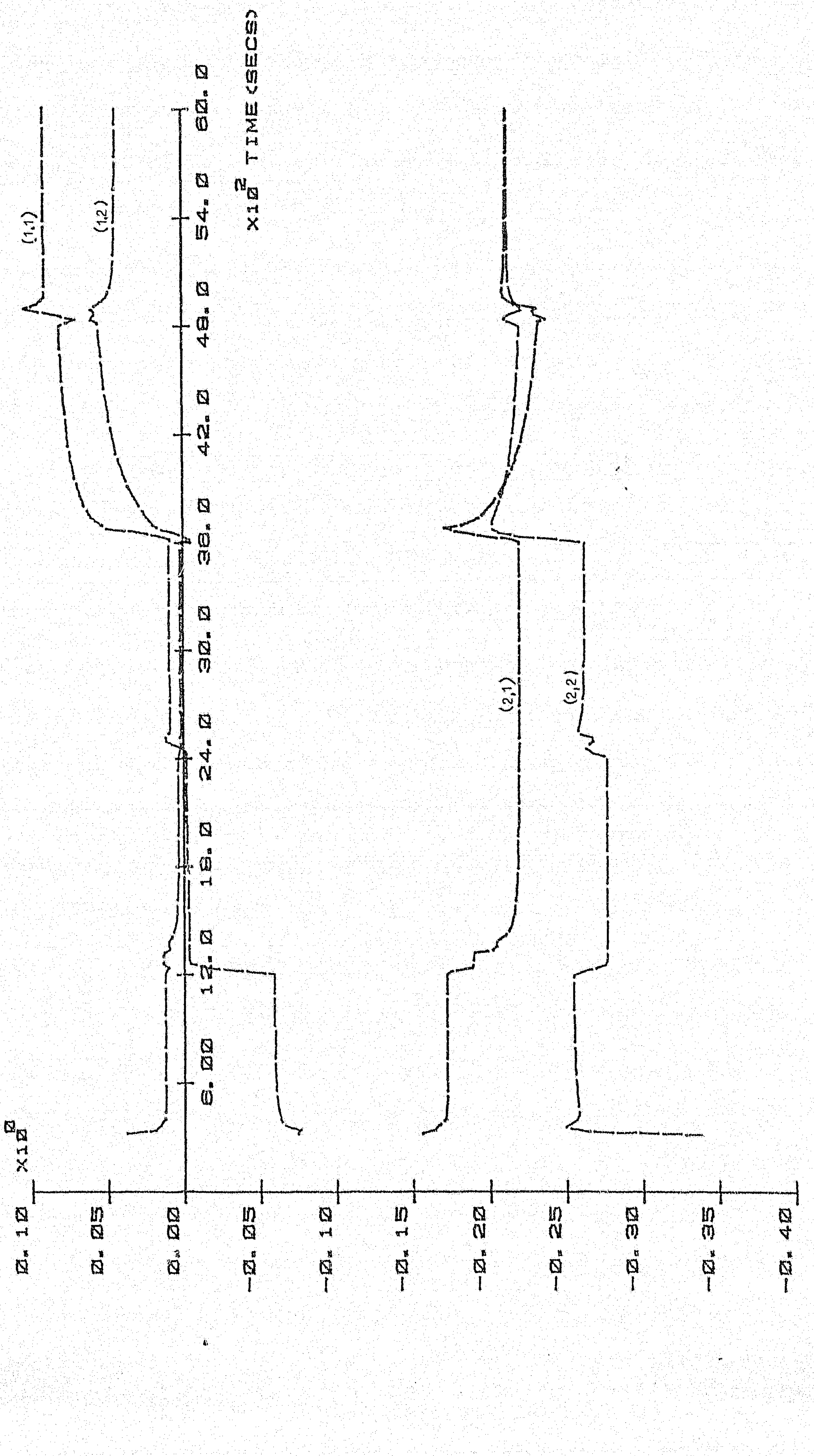



GRAPH 4.D
X10^0 LEVEL-2 (MTS)
0.30
0.27
0.24
0.21
0.18
0.15
0.12
0.09
0.06
0.03
0.00
LEVEL 2
SET PT.
6.00
12.0
18.0
24.0
30.0
36.0
42.0
48.0
54.0
60.0
X10^2 TIME (SECS)

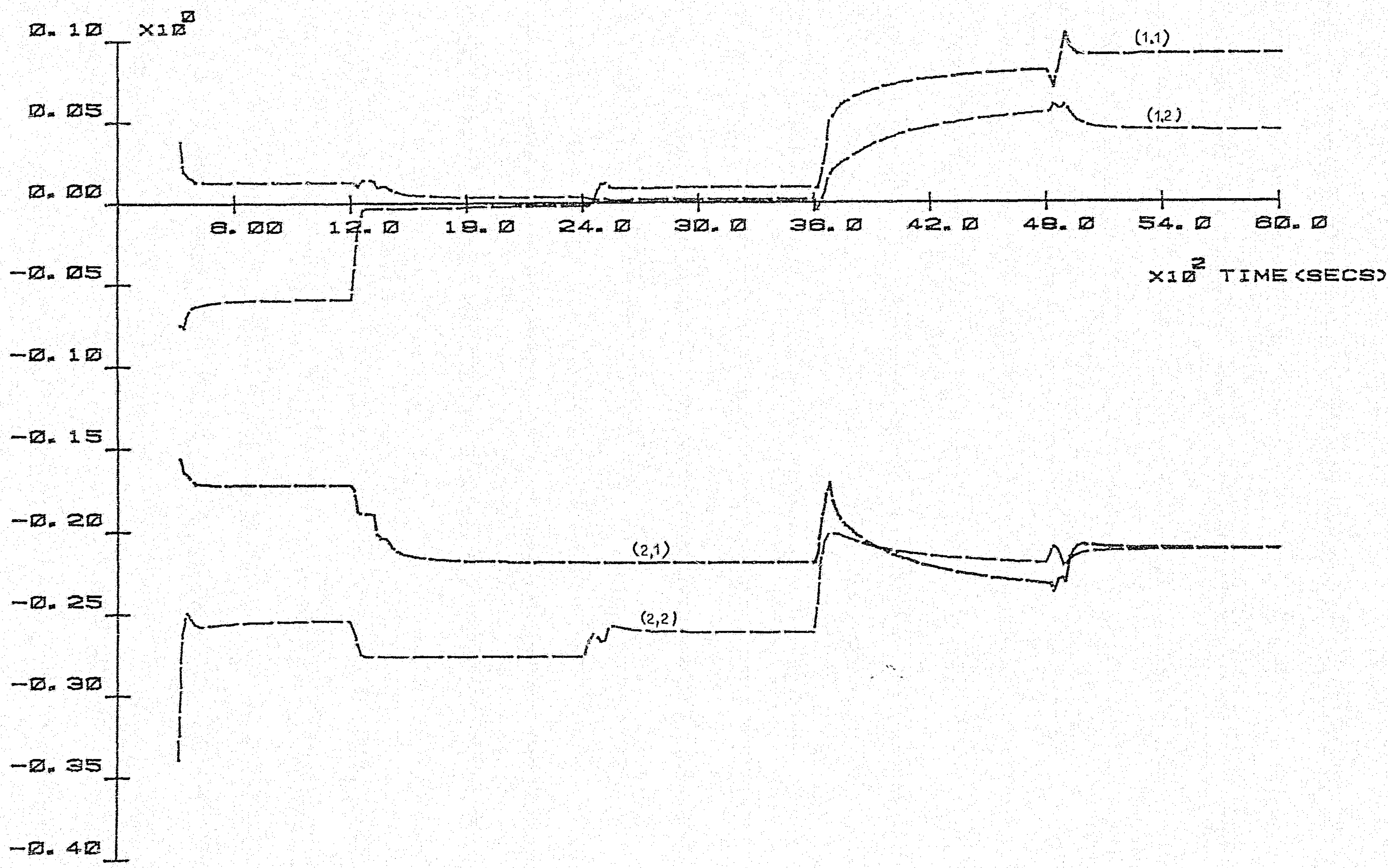
GRAPH 4.E ; FLOW 2



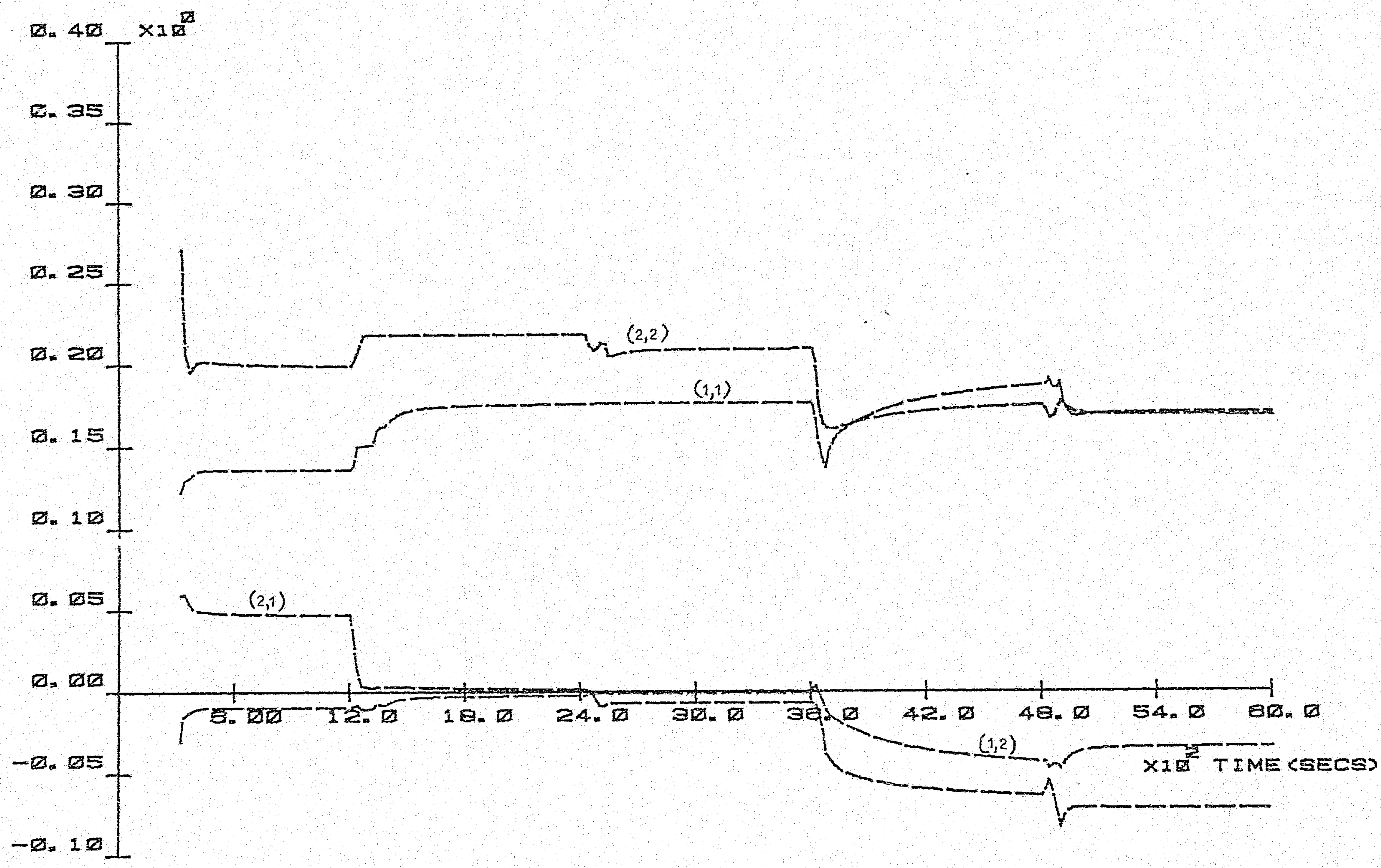
GRAPH 4.F $\cdot G_0$-MATRIX



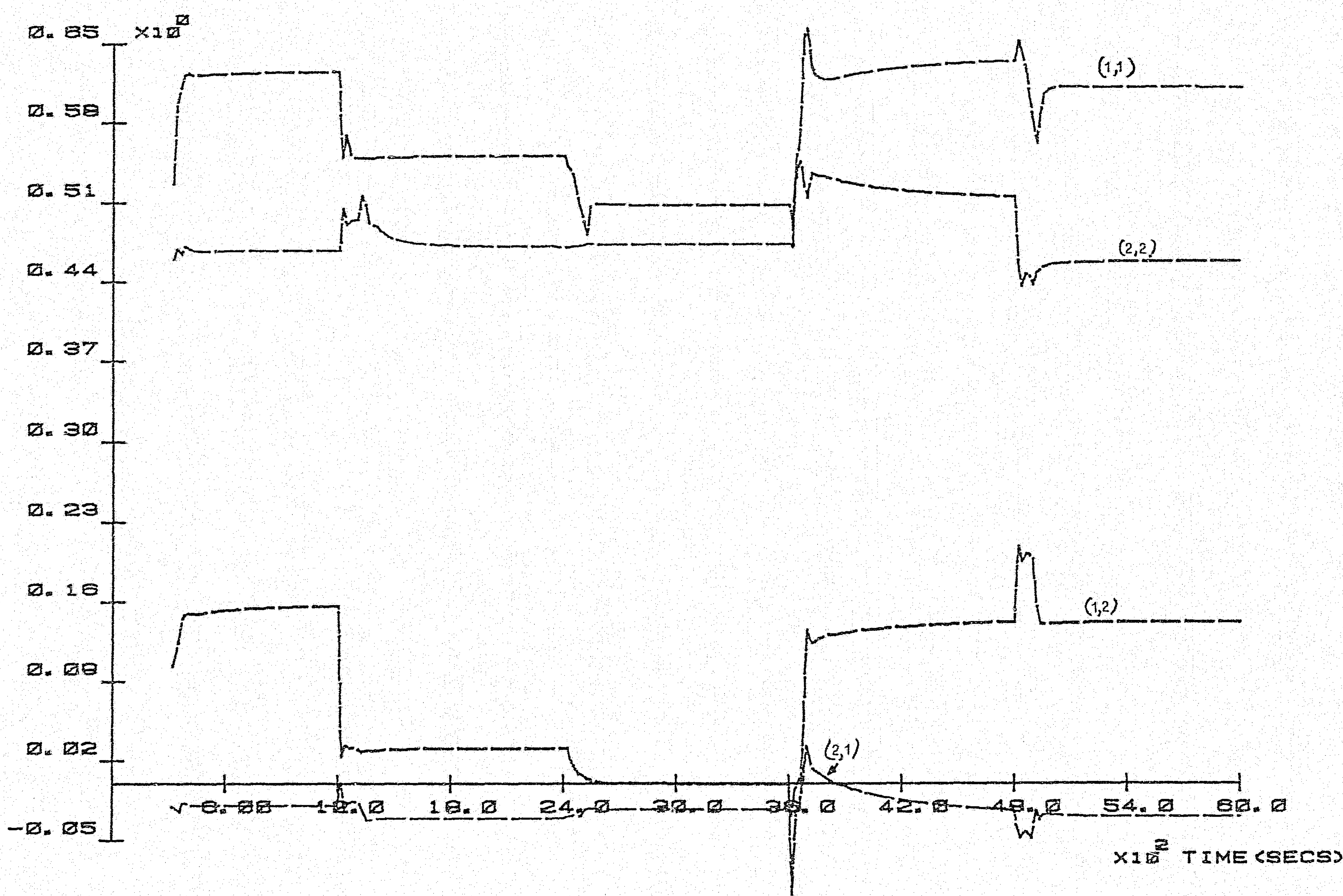
GRAPH 4.F $\cdot G_0$ ~ MATRIX



GRAPH 4.G G_1 - MATRIX



GRAPH 4.H F_1 - MATRIX



4.5. DISCUSSION

As was discussed the type of controller and its details are determined largely by the physics of the system to be controlled. Understanding of the system and the control objectives are requisites. A summary of the various concepts and factors involved is presented in the remainder of this section. Their significance and limitations are also included.

Throughout the test runs (presented in the preceeding section) the same basic pattern was used, the control laws can be evaluated, in terms of performance, by applying the following criteria.

(i) good disturbance rejection,
(ii) satisfactory transient response and small loop interaction,
(iii) absence of violent control excursions,
(iv) small tuning-in and re-tuning transient, and
(v) rapid convergence of the parameters

The former three criteria are those associated with conventional multi-variable control laws, but the additional points are required for adaptive controller evaluation.

It becomes clear from the foregoing analysis that self-tuning error actuated controllers require explicit management to protect them from certain operating conditions. Supervisory control is therefore essential to monitor these conditions.

The choice of self-tuning design method is, to a degree, a subjective one based upon individual preferences for optimal or classical control objectives. Broadly speaking, however, the classical pole-assignment methods are more robust and work in a wide variety of difficult situations, including when the unknown system is non-minimum phase and has variable time-delays. Optimal methods, on the other hand, are firmly linked to well specified cost-functions and as such give closely controlled optimal performance.

Unfortunately the amount of computation required for the pole-shifting method is a good deal more than for the minimum variance laws. When the computational delay in working out the 'next' control value, becomes significant in comparison with the sampling period, the $\beta(z^{-1})$ polynomial should be extended accordingly.

In the literature much emphasis is placed upon the convergence of the algorithm to the theoretically correct control law |22| & |3|. In practice, however, the theoretical assumptions on linearity and finite system order are invalid, in most cases. The result is that the self-tuner can converge to any one of a number of acceptable closed-loop configurations. In this respect they are also robust since, even though the theoretical convergence point may not be achieved, one of a range of acceptable configurations may be.

The tailoring polynomial $(T(z^{-1}))$ introduced to detune the minimum variance self-tuner, in order to 'damp out' violent or oscillatory control actions giving rise to the pole-shifter law also has an obvious effect on performance. A compromise had to be found between heavy detuning (large t_1) which left noticiable lags in the responses and light detuning (small t_1) which allowed excessive control activity. These poles determine the speed of the closed-loop performance and should be selected with this in mind. In practice the closed-loop pole set was selected to give initially rather modest performance, the set is then varied (tuned) until satisfactory response is obtained. Moreover slow closed-loop poles cause smoother, slower control actions and have similar effects on step responses. Closed-loop zero positions cannot, however, be specified and hence step responses can sometimes look rather odd. First-order tailoring polynomials were adequate, increasing their order rarely seemed to help, if no satisfactory step response or control shape can be found then the system model order is probably wrong.

The last point must be addressed to the form of non-linearities which the self-tuner is able to handle. It was seen that by including rate-limiting supervisory software the self-tuner was able to handle well operating-point non-linearities. In fact these were accomodated as time-varying parameters. Actuator non-linearities require limits on the control signal and will be analyzed in the next chapter. Rate limiting in this chapter refers to the reference signals. Obviously, since the actuator dynamics are so much faster, rate-limiting of the control signal was not necessary. In the context of this work rate-limiting of set-point changes are useful in two ways :

(a) Tuning, and retuning transients are limited,

(b) step responses are smoothed.

CHAPTER 5

SOFTWARE ENGINEERING OF SYSTEM

CHAPTER 5

SOFTWARE ENGINEERING OF SYSTEM

5.1. INTRODUCTION

A practical experiment was designed to verify the utility of self-tuning and the robustness of a typical algorithm to the assumptions made about the process. This Chapter describes the design (hardware and software) of such an experiment. The overall system configuration is depicted in Fig. 5.1.

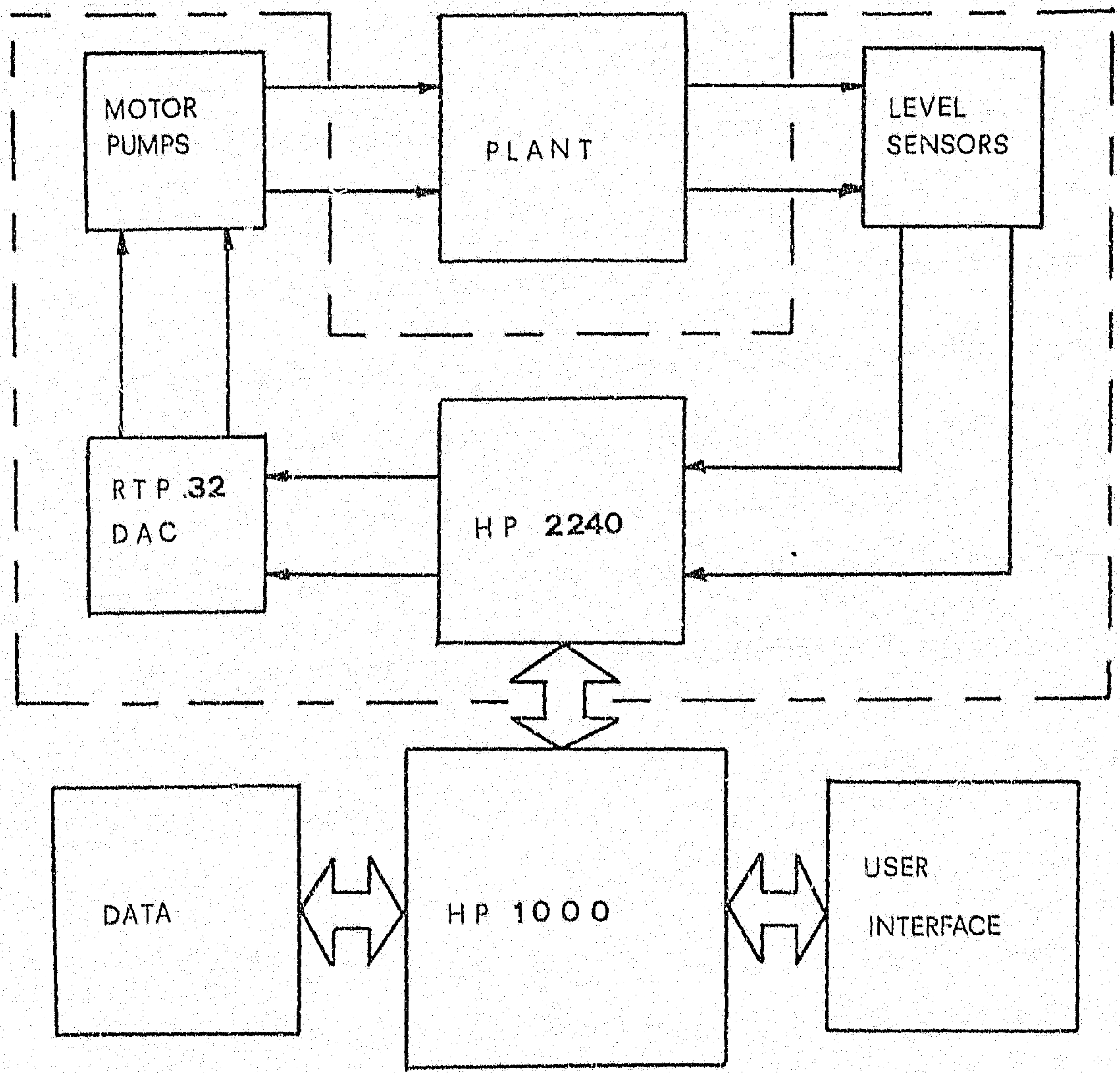


Fig. 5.1 : Block diagram of system configuration

It is important to recall at this stage that the key use of the self-tuning technique is to automatically design and implement high order digital controllers. In terms of the subsequently described software suite an unknown system is first connected to the computer either via the simulation package or the input/output measurement and control processor. The computer is then put through a cycle of simultaneous identification and control (Chapters 3 & 4) at the end of which the system is under direct digital control.

The suite gives the user some freedom in setting up various parameters in the initialisation stage. However this suite can be easily extended to include various types of estimation and control algorithms |12| & |13|. The modular structure of the software facilitates such an extension.

The remainder of this Chapter is devoted to explaining the software suite, whose various modules are described in more detail in Appendix 2. With respect to the basic hardware required, the large dotted block in Fig. 5.1 indicates those parts of the system which required hardware interfacing, the details of which are beyond the scope of this report.

It is important to recall at this stage that the key use of the self-tuning technique is to automatically design and implement high order digital controllers. In terms of the subsequently described software suite an unknown system is first connected to the computer either via the simulation package or the input/output measurement and control processor. The computer is then put through a cycle of simultaneous identification and control (Chapters 3 & 4) at the end of which the system is under direct digital control.

The suite gives the user some freedom in setting up various parameters in the initialisation stage. However this suite can be easily extended to include various types of estimation and control algorithms |12| & |13|. The modular structure of the software facilitates such an extension.

The remainder of this Chapter is devoted to explaining the software suite, whose various modules are described in more detail in Appendix 2. With respect to the basic hardware required, the large dotted block in Fig. 5.1 indicates those parts of the system which required hardware interfacing, the details of which are beyond the scope of this report.

5.2. STRUCTURE OF THE SUITE

The structure of the following software suite brings together many software rehability concepts and has yielded a software system which is not only simple to test, but also highly maintainable and extendable. The overall structure is shown in block diagram format in Fig. 5.2.

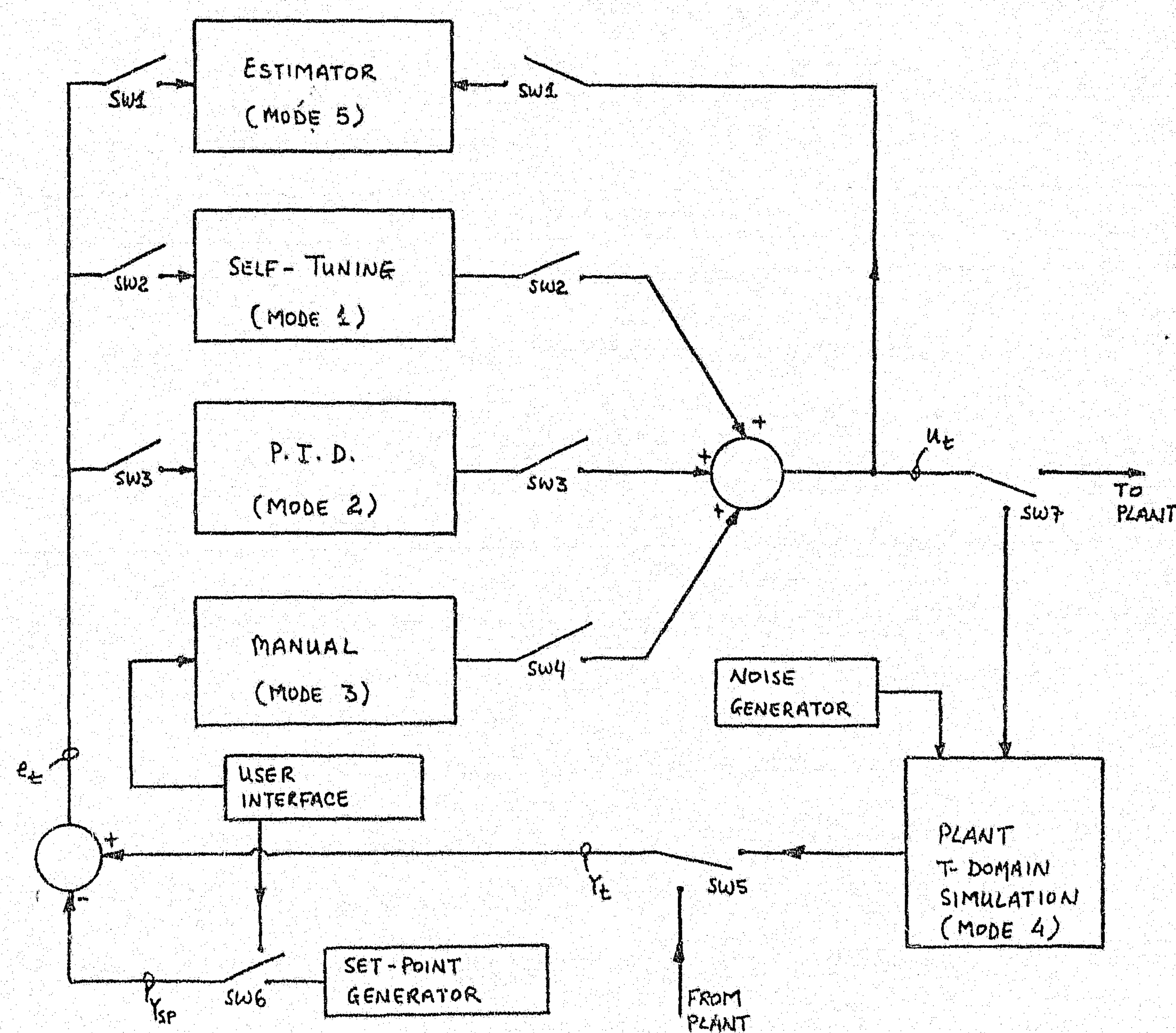


Fig. 5.2. : Block diagram of software structure

As can be seen the structure is highly modular and flexible facilitating future system enhancements. Further since the entire computing power was solely dedicated to the control of the plant, and the sampling interval required imposed no constraints, the software is run using a batch technique.

In the user initialization block the switches (SW1 - SW6) in Fig. 5.2 are set-up. According to these various modes identifying flags are set. Each individual routine then conditionally branches to the relevant section. In this way the batch structure is maintained throughout all modes. Fig. 5.3 depicts in block diagram form the structure of the 'flow-rig' control algorithm.

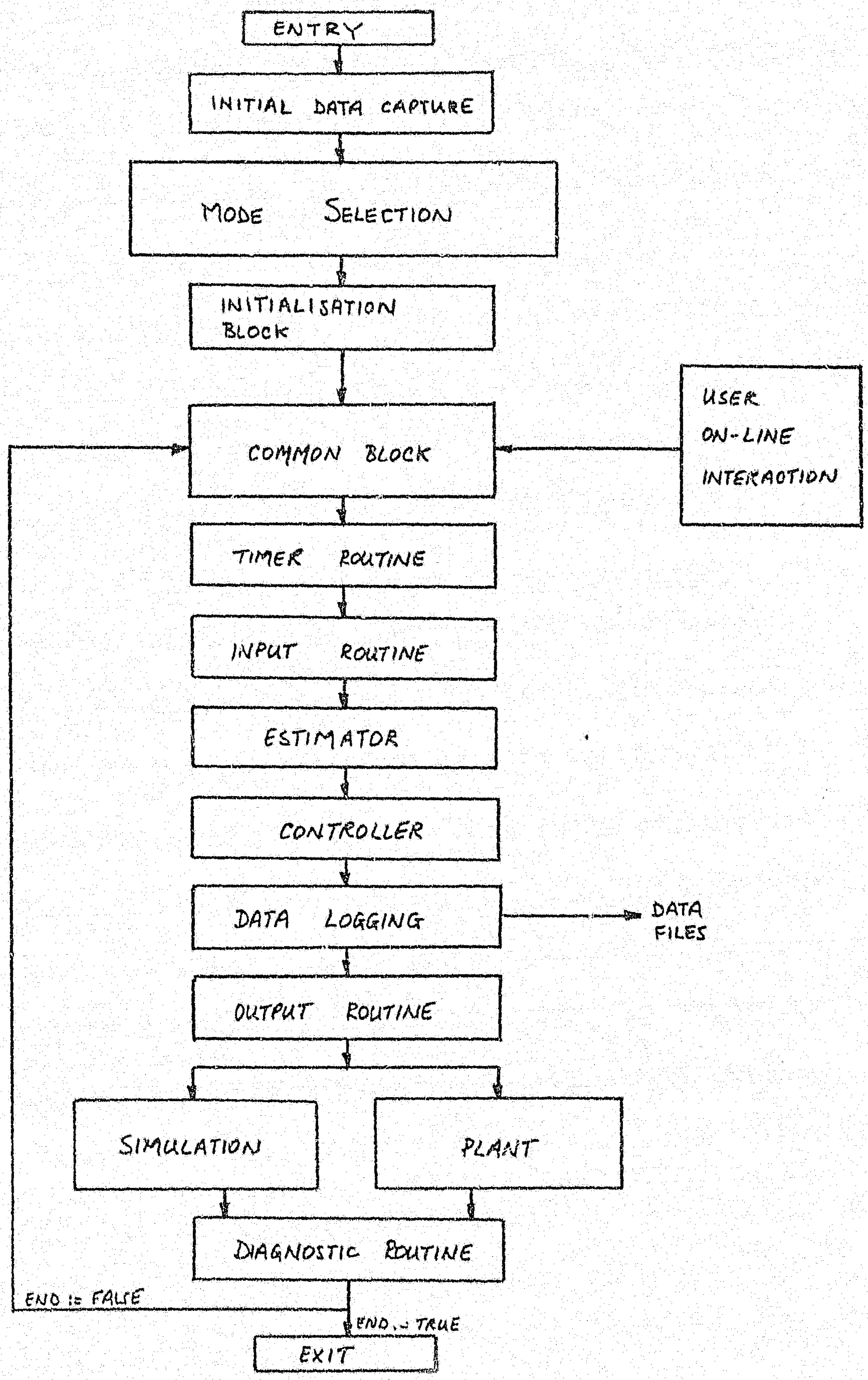


Fig. 5.3. : Block Diagram Structure of Flow-Rig Control Algorithm

A detailed listing of the user-initialization block is included in Appendix 3. Depending on the mode switches there are six possible operations of the suite, as follows :

(i) <u>MODE 1</u>

Self-tuner : Pole assignment controller and least-squares identification.

(ii) <u>MODE 2</u>

Classical SISO Control : Proportional-Integral-Derivative Controller tuned using ISE criterion as in |16|.

(iii) <u>MODE 3</u>

Manual Control : through user interactive dialog.

(iv) <u>MODE 4</u>

Non-linear time-domain plant simulation.

(v) <u>MODE 5</u>

Least-squares Estimator.

All these modes can be run either on the actual plant or on the digital plant simulator by appropriate setting of the simulation flag during initialization.

Run Time interaction allows the user to modify setpoints and/or the process variables. This is performed by a small program which 'shares' the same memory block as the main flow-rig program. Another facility is that any mode can be selected on-line.

The diagnostic software block is implemented to improve the integrity of the self-tuner. There is an error magnitude bound, which when exceeded makes the self-tuner revert to Mode 2 operation or Mode 3 and automatic shut-off.

There exist two system models which can be selected as per equations (2.20) and (2.21) by setting the appropriate flag during initialization.

Appendix 4 includes the format of the data files for the data logging block, whilst Appendix 5 contains the user-on-line interactive dialogue.

5.3. DISCUSSION

The software format used was chosen for its simplicity and hence understandability. It is recognized that other structures were possible, however, the solution presented is far less complex and can be easily implemented on other minicomputer systems. Solutions making use of the HP 1000 Real-Time Executive (eg. scheduling programming and mail box interprogram communication) are far more complex and the parallelism gained would only be of importance if more than one process was to be controlled thus imposing constraints on sample interval.

Software design is essentially a human problem-solving process. Hence solutions should focus on techniques that reduce the complexity of the problem and its solution to levels that can be easily comprehended and resolved by users other than the designer.

The modular structure employed has its main advantages in testability and flexibility. The proposed solution can be easily modified to include other control laws, other identification algorithms and even other plants. The user is only required to modify certain self-contained sections of the particular modules.

Much more could be written about software engineering system considerations, this is however a subject on its own. The structure developed was basically decided upon its user-friendly attributes.

CHAPTER 6

SELF-TUNING AS COMBINED IDENTIFICATION AND CONTROL

6.1. INTRODUCTION

In this chapter the practical aspects of self-tuning are considered in the context of a two-input/two-output physical system described in Chapter 2. This process embodies severe operating point dependent and actuator non-linearities. The performance of the self-tuner is then judged not only according to the performance criteria listed in Section 4.5 but also regarding its ability to handle the above mentioned non-linearities.

Another feature of process systems of this nature is that they are non-minimum phase in the discrete time domain, as can be proved for certain sampling times (Appendix 1), hence conventional optimal self-tuners will have difficulty in dealing with them. The robustness, in this respect, of the pole-assignment self-tuner used will therefore also be investigated.

The component parts of the self-tuner discussed in Chapters 2, 3 & 4 are now joined together in a single adaptive control algorithm, thus (see Fig. 4.3),

(i) at each sample interval the inputs U_K and outputs Y_K to the plant are sampled;

(ii) this new data is used by the recursive least-squares algorithm to update the estimates of the model parameters $\hat{\theta}_K$;

(iii) these corrected estimates are used to determine the improved controller parameters. The type of control being determined by the pole-assignment law.

These three steps are executed in sequence indefinitely under the conditions explained in the previous chapter.

If the operating point is changed or the plant is modified, then the variable forgetting factor method helps the self-tuner to re-tune to the new configuration by increasing its sensitivity. If the process was not time-varying then once the estimates of the system parameters, and hence the control, have converged to steady values, the whole self-tuner can be 'frozen'. The controller parameters are then correct and hence the identification section can be suspended. This is very seldom the case.

The self-tuner can therefore be seen as a powerful on-line controller design tool. The algorithm being capable to tuning a large number of parameters to meet the control objective. This control objective must, however, be defined by the user requiring some apriori knowledge about the system and engineering judgement.

The subsequent sections provide some information concerning the experience obtained in running the self-tuner on-line with the physical process. In particular factors regarding the tuning of the algorithm both initially and on-line are analysed in detail.

As a final remark it is noted that the implementation of the self-tuner tested extensivelly on a plant simulation (Chapter 4) on the actual process will dictate whether or not it is robust enough to cope with actuator non-linearities and sensor disturbances. It has already been seen (Chapter 4) that operating point non-linearities are coped with as time-varying parameters.

6.2. SELF-TUNER CHARACTERISTICS : ANALYSIS

The self-tuner incorporates a number of parameters defining its objectives in terms of noise rejection, set-point following, control signal behaviour (discussed in section 4.5) and parameter tracking capabilities (discussed in section 3.5). A number of these parameters can be usefully adjusted on-line either automatically or by the user, thus tuning the controller rather than the control law to meet the required objectives. Such parameters are briefly analysed in the following.

(i) The state of the self-tuner :

The state of the self-tuner is defined by its current parameter estimate vector $\hat{\theta}_k$ and its confidence matrix P_k (Chapter 3). This defines the tracking ability of the self-tuner. The confidence matrix is a measure of how much information the estimator has received in the past. Parameter estimates are updated at each step by an amount related to the current errors and measurement (inputs/outputs) vector times the confidence matrix (see equation 3.1) Hence the larger the entries in the latter the more attention will be paid to current I/O data.

The variable forgetting factor concept developed in section 3.3 is a means of automatically sizeing the confidence matrix. In this way the forgetting factor is varied in order to encourage initial tuning-in and, later, inhibit excessive sensitivity to new incoming data.

An adequate measure of the size of the confidence matrix is its trace. Numerical values of the trace are dependent upon scaling of I/O information