Proceedings of 25th Conference
on Decision and Control
Athens, Greece • December 1986

TA5 - 9:45

HIERARCHICAL INTEGRATED SYSTEM OPTIMISATION AND PARAMETER ESTIMATION USING A MICROCOMPUTER BASED SYSTEM

S Chen, D S Wadhwani and P D Roberts

Control Engineering Centre, The City University
London

Abstract

The advent of inexpensive and more powerful microcomputers has facilitated the implementation of on-line optimization schemes to industrial processes. Most optimization algorithms require a process model and their performance clearly depends on how accurate the model is to reality. Unfortunately, for many industrial processes an accurate model may be very difficult, time consuming or impractical to obtain. In this paper a hierarchical integrated system optimis ation and parameter estimation technique developed by combining the interaction balance method[6] and the modified two-step approach[1,8] has been investigated using a microcomputer based system. Real-time simulation results obtained show that two versions of this technique (a single iterative algorithm and a double iterative algorithm) are capable of achieving the real process steady state optimal operating condition in spite of the model used to solve the optimisation problem not being a faithful representation of reality. The results also demonstrate that the double iterative version has an important advantage under real time operation. It significantly reduces the time for determining the real optimum operating condition.

1. Introduction

During the last two decades considerable advances have been made in the field of process control. There is an increasing interest in maintaining an industrial process at its optimal operating condition efficiently. This may be due to the increasing limitation on natural resources and rising costs of energy and raw materials. Furthermore, the improvement in microcomputer technology has a great impact on this issue.

When considering hierarchical steady-state optimal control of an interconnected system, a fundamental difficulty appears in that model-reality differences inevitably exist. Hence, the purely model based interaction prediction method and the interaction balance method often fail to attain the real system optimal operating condition and may even violate the system constraints. Using feedback information to improve these two methods has been thoroughly investigated[6]. However, in general, the resulted techniques can only achieve a suboptimal solution.

In previous work[1,3,7] a hierarchical integrated system optimization and parameter estimation (ISOPE) technique has been suggested. This hierarchical ISOPE technique shares many common advantages (such as simplifying the overall optimization task, enabling parallel calculation and being suitable for implementation using the distributed computer system) with other existing hierarchical optimization techniques mentioned above. Moreover, in an efficient manner and based only on an approximate model, it attains the optimal steady state operating condition rather than a suboptimal one. This benefit is obtained at the cost of demanding more information from the real process in the form of measurement derivatives with respect to the controller set points. A further advantage occurs, however, in that due to its adaptive nature, the technique enables simpler models to be used, and consequently has a great advantage in the situation where process knowledge is highly uncertain. For example, if any disturbances occur at any stage during the operation (for instance, changes in the raw material composition) the technique will adapt and proceed towards a new optimum.

In this paper, this hierarchical ISOPE technique is implemented using a microcomputer based system, where the aim is to investigate the behaviour of the technique in a real-time environment and to study methods for dealing with some problems associated with on-line implementation. The efficiency of the two versions (a single iterative algorithm and a double iterative algorithm) of the technique is also compared. The paper is an extension of existing work[4].

2. Hierarchical ISOPE technique

It is assumed that the real system including its follow-up controllers is described in a decomposed way by a set of subsystem input-output mappings

$F_i^*: \mathcal{C}_i \times \mathcal{U}_i \to \mathcal{Y}_i$, i=1,...,N,

as follows

$y_i = F_i^*(c_i, u_i)$, i=1,...,N,

where N denotes the number of subsystems, $\mathcal{C}_i$, $\mathcal{U}_i$ and $\mathcal{Y}_i$ are finite dimensional speces, c_i, u_i and y_i are the ith subsystem's control, interaction input and interaction output vectors, respectively. The subsystems are interconnected through the coupling equations

$$u_i = H_i y = \sum_{j=1}^{N} H_{ij} y_j, \quad i=1,\dots,N,$$

where H_i and H_{ij} are interconnection matrices. The local constraint set takes the form

$$(c_i,u_i,y_i) \in CUY_i \triangleq \{(c_i,u_i,y_i) \in \mathcal{C}_i \times \mathcal{U}_i \times \mathcal{Y}_i : G_i(c_i,u_i,y_i) \leqslant 0\}, i=1,\dots,N,$$

where G_i is the ith subsystem constraint function vector. The system as a whole can be written as

$$y = F^*(c,u), \tag{1}$$

$$u = Hy, \tag{2}$$

$$(c,u,y) \in CUY \triangleq \{(c,u,y) \in \mathcal{C}\times\mathcal{U}\times\mathcal{Y}: G(c,u,y) \leqslant 0\}. \tag{3}$$

Next, it is required that for each $c \in \mathcal{C}$, the equation $y = F^*(c,Hy)$ has a unique solution

$$y = K^*(c), \tag{4}$$

where $K^*: \mathcal{C} \to \mathcal{Y}$ and $K^*(c) = ((K_1^*(c))^T \dots (K_N^*(c))^T)^T$.

In general the above real system relations are not known exactly and, consequently, we have only their approximate models

$F_i: \mathcal{C}_i \times \mathcal{U}_i \times \mathcal{A}_i \rightarrow \mathcal{Y}_i$,

$y_i = F_i(c_i, u_i, \alpha_i)$, $i=1,\ldots,N$,

where $\mathcal{A}_i$ is a finite dimensional space and α_i is the ith subsystem model parameter vector. As before, the global model equations can be written as

$$y = F(c,u,\alpha). \tag{5}$$

The interconnection matrix and the local constraint set are assumed to be known exactly.

Finally, a known local performance index, associated with each subsystem

$Q_i: \mathcal{C}_i \times \mathcal{U}_i \times \mathcal{Y}_i \rightarrow R$, $i=1,\ldots,N$,

is required to be minimized. The overall performance function

$Q: \mathcal{C} \times \mathcal{U} \times \mathcal{Y} \rightarrow R$,

is assumed to be

$$Q(c,u,y) = \sum_{i=1}^{N} Q_i(c_i,u_i,y_i). \tag{6}$$

The task of determining the optimal controller set points c for a real system can be defined as the following real steady state optimizing control problem[2]:

$$\text{(EOCP)} \quad \begin{aligned} &\min_{c,v,u,\alpha} q(c,u,\alpha), \\ &\text{s.t. } F(v, HK^*(v), \alpha) = K^*(v), \\ &\quad u = HF(c,u,\alpha), \\ &\quad g(c,u,\alpha) \leqslant 0, \\ &\quad v = c, \end{aligned} \tag{7}$$

where $q(c,u,\alpha) = Q(c,u,F(c,u,\alpha))$ and $g(c,u,\alpha) = G(c,u,F(c,u,\alpha))$.

The modifier formula can be derived[2]

$$\lambda = \lambda(c,u,v,\alpha,p,\xi) = \left[\frac{d^T F(v,HK^*(v),\alpha)}{dv} - \frac{d^T K^*(v)}{dv}\right] \left[\frac{d^T F(v,HK^*(v),\alpha)}{d\alpha}\right]^{-1} \left[\frac{d^T q(c,u,\alpha)}{d\alpha} + \frac{d^T g(c,u,\alpha)}{d\alpha}\xi - \frac{d^T F(c,u,\alpha)}{d\alpha} H^T p\right]. \tag{8}$$

The ith local parameter estimation problem is defined as: determining the value of the ith parameter vector so that

$$F_i(v_i, H_i K^*(v), \alpha_i) = K_i^*(v). \quad i=1,\ldots,N. \tag{9}$$

The ith local model based optimisation problem can be defined as:

$$\min_{c_i u_i} L_i(c_i,u_i,\alpha_i,\lambda_i,p), \qquad i=1,\ldots N,$$

$$\text{s.t. } g_i(c_i,u_i,\alpha_i) \leqslant 0,$$

$$\text{where } L_i(c_i,u_i,\alpha_i,\lambda_i,p) = q_i(c_i,u_i,\alpha_i) - \lambda_i^T c_i + p_i^T u_i - \sum_{j=1}^{N} p_j^T H_{ji} F(c_i,u_i,\alpha_i). \tag{10}$$

The following two iterative algorithms can be used to find a solution of the overall problem.

2.1. Single iterative algorithm (SIA)

The ith local control problem which consists of the ith local parameter estimation problem and the ith local optimisation problem is defined:

LCP_i, $i=1,\ldots,N$:

(i) For given v_i and measurements $K_i^*(v)$ find the model parameters α_i which satisfy

$$F_i(v_i, H_i K^*(v), \alpha_i) - K_i^*(v) = 0,$$

(ii) For given α_i, p and λ_i find the controller set points $\hat{c}_i$ and interconnection inputs $\hat{u}_i$ such that

$$(\hat{c}_i, \hat{u}_i) = \arg\min_{c_i,u_i} L_i(c_i,u_i,\alpha_i,\lambda_i,p),$$

$$\text{s.t. } g_i(c_i,u_i,\alpha_i) \leqslant 0.$$

The following iterative strategies for updating v, ξ and p are proposed

$$v_i^{k+1} = \Psi_{v_i}(\hat{c}_i^k, v_i^k) = v_i^k + K_{v_i}(\hat{c}_i^k - v_i^k), \quad i=1,\ldots,N, \tag{11}$$

$$\xi_i^{k+1} = \Psi_{\xi_i}(\hat{\xi}_i^k, \xi_i^k) = \xi_i^k + K_{\xi_i}(\hat{\xi}_i^k - \xi_i^k), \quad i=1,\ldots,N, \tag{12}$$

$$p^{k+1} = \Psi_p(p^k, \Delta\hat{u}^k) = p^k + K_p(\hat{u}^k - HF(\hat{c}^k, \hat{u}^k, \alpha^k)), \tag{13}$$

where K_{v_i}, K_{ξ_i} and K_p are diagonal gain matrices. The kth iteration is described as follows:

(a) The local control units update v_i, $i=1,\ldots,N$, then apply v_i^k to the real subsystems and obtain the corresponding steady-state measurements of local outputs $K_i^*(v^k)$ and interconnection inputs $H_i K^*(v^k)$. The units then determine α_i^k by solving

$F_i(v_i^k, H_i K^*(v^k), \alpha_i) = K_i^*(v^k)$, $i=1,\ldots,N$,

and finally perform additional perturbations about v_i^k and take the corresponding real measurements. The values of α_i^k, v_i^k and all real measurements are sent to the coordinator for computing finite difference approximations of the derivatives.

(b) The coordinator computes ξ^k and p^k, calculates modifiers according to (8)

$\lambda^k = \lambda(\hat{c}^{k-1}, \hat{u}^{k-1}, v^k, p^k, \alpha^k, \xi^k)$,

and sends λ_i^k and p^k to the ith local unit.

(c) The local units perform their own optimisation task

$$(\hat{c}_i^k, \hat{u}_i^k) = \arg\min_{c_i,u_i} L_i(c_i,u_i,\alpha_i^k,\lambda_i^k,p^k)$$

$$\text{s.t. } g_i(c_i,u_i,\alpha_i^k) \leqslant 0, \quad i=1,\ldots,N.$$

The values of c_i^k, u_i^k and associated Lagrange multiplier λ_i^k are then sent to the coordinator.

The overall convergence is achieved when successive solutions of v and ξ are unchanged and interaction balance is satisfied. In practice, the overall process may be terminated when every element of v, $\hat{u}$ and ξ satisfies

$$|v_{ij}^k - \hat{c}_{ij}^k| < \beta_v, \tag{14}$$

$$|\xi_{il}^k - \xi_{il}^k| < \beta_\xi, \tag{15}$$

$$|\hat{u}_{is}^k - (H_i F(c^k, u^k, \alpha^k))_s| < \beta_p, \tag{16}$$

where β_v, β_ξ and β_p are some desired tolerances. $(H_i F)_s$ denotes the sth element of $H_i F$. A special situation occurs when the system constraints do not depend on the outputs. In this case the iterative strategy (12) and the criterion (15) are not needed.

Calculating real measurement derivatives is time consuming. For example, considering the following simple finite difference formula:

$$\frac{\partial K_j^*(v)}{\partial v_i} \approx \frac{K_j^*(v + e_i\sigma_i) - K_j^*(v)}{\sigma_i}, \quad j=1,\ldots,m, \quad i=1,\ldots,n. \tag{17}$$

where e_i is a unit vector with unity ith element and others all zero, σ_i is a small perturbation on the ith element of v,m is the number of outputs and n is the number of controller set points. Let T_m be the plant settling time period required to wait for the process to reach a new steady state after the set points have been altered. The time required for step (a) will approximately be $(n + 1)T_m$. Therefore, advantage would be gained if it is possible to design an alternative strategy which updates α and λ less frequently. In the next subsection it will be shown that this aim can be achieved by employing a double iterative mechanism.

2.2. Double iterative algorithm (DIA)

This algorithm is formed by separating the function of improving the modifier λ and local optimization. The latter gives rise to an interior procedure during which information is interchanged only between local optimization units and the coordinator, while the former, which requires real system information feedback, is solved at less frequent intervals in an outer iterative loop. The overall iterative procedure is described as follows:

With α and λ fixed, the interior procedure is the entirely model-based problem

$$\text{ILOP}_i: \; i=1,\ldots,N \quad \begin{cases} \text{Find } \hat{c}_i \text{ and } \hat{u}_i \text{ such that} \\ (\hat{c}_i,\hat{u}_i) = \arg\min_{c_i,u_i} L_i(c_i,u_i,\alpha_i,\lambda_i,p), \\ \text{s.t. } g_i(c_i,u_i,\alpha_i) \leqslant 0, \end{cases}$$

$$\text{ICP}: \quad \begin{cases} \text{Find } \hat{p} \text{ such that} \\ u - HF(c,u,\ ) = 0. \end{cases}$$

This problem is solved iteratively using an open-loop interaction balance method. When interaction balance has been achieved, α and λ are adjusted by the outer iterative loop and then the interior procedure restarts with the new α and λ. The outer loop iterates α and λ similarly to the single iterative algorithm as described in steps (a) and (b) except that $\hat{c}^{k-1}, \hat{u}^{k-1}$ and p^k are the solution and corresponding price vector of the interior procedure. the overall process is terminated when v and ξ determined in the outer procedure remain sufficiently unchanged between successive iterations.

Both algorithms operate in a decentralized manner at the local level, and there is no need for communication among the local units. They can easily be implemented using hierarchical computer networks such as the one to be discussed later, with a central computer (supremal level) serving as the coordinator and local computers (infimal level) serving as the local control units. Parallel local optimization and parameter estimation calculations are achieved.

Two types of information interchange which occur during the iterative procedure are defined. The first type occurs between the local control units and the coordinator after each model-based optimization iteration, which we will denote as off-line information interchange. The second type occurs between the real subsystems and the coordinator (through the local control units) after each time that the controller set points have been changed, which we will refer to as on-line information interchange. It is assumed that the total number of information exchanges is roughly equal to the sum of the off-line and on-line information interchanges.

3. Distributed hierarchical computer system

A popular distributed network structure which is often used for small networks is the star. In a star network, the local computers are connected by individual communication links to the main (host) computer. This modularly organized structure ensures that it is easy to handle expansion of the network. In addition, the operating speed of the links may be slow and the hardware required to support them can be very simple. This structure is efficient provided that there is little need for communication between local computers (since all such communication must be routed via the host). The major disadvantage of the structure, however, is that the system may breakdown if the host computer fails and, hence, a standby computer is often required. In spite of this reliability problem, the star network is often used in control systems because the structure reflects their hierarchical nature.

One of the main areas of research in the Control Engineering Centre of The City University is concerned with the application of hierarchical control techniques to industrial processes. Much of this research is conducted in the computer control laboratory using a distributed two-level computer network (Fig.1.) which operates several pilot-scale processes simultaneously. The network has a typical star structure. Previously, several Intel 8085 based I-MIC microcomputers occupied the current positions of Acorn BBC model B microcomputers. In some applications, the speed of the communication links between the I-MICs and the host computer proved to be insufficient. For example, in an application of the algorithms given in Section 2 it was found that considerable amount of time was spent in data communication[4]. In order to improve the efficiency of the network and to accommodate new research activities, these I-MICs have gradually been replaced by BBCs with a more efficient inter-communication system.

The host computer, a LSI11/23 which is used to coordinate and monitor the computers at the infimal level, has a full complement of 256Kbyte of RAM and runs the TSX-PLUS time-shared operating system. Programmes and data are stored on twin 20Mbyte Winchester discs and a 1.2Mbyte floppy disc. Other peripherals available at the supremal level include a Tektronix graphics terminal, a hard copy unit, a 180 cps line printer and an Intecolor 8001G graphics terminal. The LSI11/23 is programmed mainly in FORTRAN IV because the real-time support routines are written in FORTRAN or are FORTRAN callable.

The infimal level of the network contains two I-MICs, a DEC LSI11/02 minicomputer and four BBCs. The I-MIC has 16Kbyte of RAM and its software is mostly written in CONTOL BASIC with a few routines, such as the link communication routines, written in Intel 8085 machine code in situations where the execution speed of the high level interpretive language is insufficient. Each BBC has 32Kbyte of RAM and a floppy disc system, and is programmed in BBC BASIC with the exception of the link routines which are written in 6502 assembly language. The LSI11/02 is programmed entirely in FORTRAN IV.

At present an I-MIC controls a hot and cold water mixing rig, a BBC and an I-MIC control a pilot-scale freon vaporiser which is viewed as an interconnected two-subsystem plant. A BBC is used in the control of a TQ 2000 robot manipulator. The rest of the BBCs control an analogue computer simulating interconnected plants. The LSI11/02 controls a pilot-scale travelling load furnace.

The network is used as a development tool for designing and testing control algorithms in a real-time situation. Three types of data communication links exist. The communication between the LSI11/02 and the LSI11/23 takes place over a 16 bit optically isolated parallel link. The data transfer rate for the parallel link is limited primarily by the response speed of the time-shared operating system to interrupt requests. The communication between an I-MIC and the LSI11/23 takes place over a 20mAcurrent loop serial line at 1200bd, whilst the communication between a BBC and the LSI11/23 takes place via RS423 (RS232) interfaces at 9600bd, i.e., eight times faster than that of the I-MIC and the LSI11/23 link. The configurations, communication protocol and supporting software for these three types of links have been developed[5,10].

Part of the network used to implement the algorithms given in Section 2 includes the LSI11/23, which serves as the coordinator, twoBBCs which are used to perform local parameter estimation and optimization tasks, and an EAL Pace general purpose analogue computer, which simulates an interconnected two-subsystem dynamic process. Measurements are taken using the BBC analogue port. Although this 4 channel 12-bit A to D convertor is not a high performance device we are only concerned with steady state measurements and, provided that a simple averaging procedure is performed to reduce the thermo-effect of the A/D port, this analogue port is adequate for this task. Control signals applied to the plant are generated through a BBC user port to a 3 channel 8-bit D to A convertor using software control.

4. Some on-line implementation aspects

Using the distributed computer system, parallel computation can be performed at the local level once coordination variables have been received from the supremal level. However, synchronisation and inter-process communication problems arise at the local units because the iteration in each unit is finished within a different time interval.

Because steady state measuements are needed in order to estimate parameters and, particularly, to compute finite difference approximations of the derivatives with respect to the controller set points, bad quality measurements will affect the iterative procedure. For this reason synchronisation is recommended to enable controls to be sent to the real process simultaneously, such as elapsed time or semaphore techniques[9].

After controls have been applied to the real subprocess, all local control units should wait a sufficient time until the plant has settled down, at which point steady state measurements are taken. It is difficult to reduce this on-line waiting time interval without risking the quality of the steady state measurements. However, if the number of controller set point changes can be reduced the time for determining the optimal solution may be less. The double iterative algorithm is aimed at achieving this purpose.

The requirement to evaluate real process derivatives represents a significant drawback of the technique. In reality, the process measurements are often contaminated with noise. The errors in measurements will be further amplified by the use of finite difference approximations to determine the derivatives. In the situation where the noise level becomes intolerable filter techniques should be applied to reduce the influence of noise.

5. Simulation example and results

An interconnected two-subsystem plant has been simulated using the TR48 analogue computer. Details of the plant, its steady state model and performance indices are given elsewhere[9,11].

It is observed that measurement noise is near 1% on average about normal operating conditions. Sources of noise include the low quality of the A/D convertors, the long age of the analogue computer and some other unknown factors. However, such a noise level is small in a typical industrial environment and should not cause difficulties to the algorithms if they are to have any practical usefulness. When the process settling time period is set at 20 seconds, an item of on-line information interchange, beginning when the local control units apply the controls to the real process until the start flag of the next action is received, will take about 21.5 seconds. It is observed that it takes approximately 1.5 seconds for the system to complete a typical off-line information interchange (each local unit performs its optimization calculation, sends its results to the coordinator and then waits until it is informed by the coordinator that all other units have completed their tasks). A notable improvement in data communication speed has been achieved (The previous I-MIC - LSI11/23 system took nearly 10 seconds to complete the same item of off-line information interchange).

The derivatives required to compute λ were approximated using (17). Suitable values of the gain matrices K_v, K_p were found by experiment as $K_v=0.4I$ and $K_p=\text{diag}\{0.8, 0.9\}$, where I is an identity matrix. Desired tolerances were chosen as $\beta_v=\beta_p=0.01$ according to measurement accuracy. Starting from zero initial controls, typical results obtained using the technique discussed in Section 2 are shown in TABLE 1 where the results are compared with the real optimum values. To compare the efficiency of the single iterative algorithm and the double iterative algorithm, the time required to determine the real optimum solution and the communication requirement of both versions are listed in TABLE 2.

In practice the technique can be applied such that after the optimal operating condition has been found, the plant is maintained at this condition and the control system is switched into a 'supervising mode', ready to track the new optimum if the process conditions are changed. Fig.2. demonstrates this adaptability, where at time instant 0, a disturbance occurred in real subsystem 1. Because of the interconnections all real measurements were affected. A sudden deterioration in the performance was reported to the host computer who restarted the procedure. It can be seen from Fig.2. that, in this case, the DIA again took only half of the time required by the SIA to converge to the new optimum.

To study the behaviour of the algorithms under a stochastic situation additional random noise sources, ranging from -0.05 to 0.05, generated using the BBC function RND were added to all real measurements. The given noise was about 5% of the normal operating conditions. The minimum perturbation on controller set points was set as 0.06. using (17) to calculate the derivatives under these noise conditions gives a maximum error of approximately 1.7, which was about 150% of the average real derivative. To enable the application of the DIA, inner loop tolerance was increased to 0.06 and a maximum limitation of 20 iterations was imposed on the inner loop. Fig.3. and Fig.4. illustrate the serious effect of the additional noise on real system performance during the course of the iterations and the improvement achieved by the use of a simple low-pass technique. For the example tested, the DIA seemed to yield more robust results.

6. Conclusions

The practical implementation of an ISOPE technique for hierarchical control of steady state systems has been studied using a microcomputer based system. Real-time simulation results demonstrate that both versions of the technique can be used to overcome model-reality differences and to track slowly changing optimal operating conditions in on-line application. The aim of the DIA is to reduce on-line information interchange and hence, to reduce the time required for determining the optimum steady state control, even at the cost of increasing off-line information exchange. This feature is well demonstrated by the results shown in TABLE 2.

The DIA is particularly desirable for processes which have long settling time periods. While the real process settling time period is often beyond our control using more powerful computers at the local level and increasing the efficiency of the communication link will reduce the time required for off-line information interchange, which also favours the DIA.

Although in this small example, the total number of information interchanges (defined as the sum of off-line and on-line information interchanges) needed by the DIA is slightly greater than that required by the SIA, a previous off-line simulation study[3] has suggested that, for many examples, the DIA does not increase the total number of information exchanges and, in larger examples, may even reduce it significantly.

As with the majority of decentralised methods, to guarantee good performance, the hierarchical ISOPE technique requires synchronised operation. Because of the complexity of the method (mainly calculating real output derivatives) it is difficult to choose a suitable asynchronised operating scheme for the technique. If the measurements are taken long before the process entirely reaches its steady state, the performances of the technique will deteriorate.

The requirement to measure first-order derivatives of real outputs imposes an important practical limitation to the technique. To enable its application to stochastic processes, further research is needed to investigate the possibility of incorporating filter theory with the hierarchical ISOPE technique.

References

[1] Brdyś M, S Chen, and P D Roberts (1984). An extension to the modified two step algorithm for steady state system optimization and parameter estimation. Research Memorandum No.CEC/MB/SC/PDR-14, Control Eng. Centre, The City University, London. (Accepted by the Int. J. of Sys. Sci.).

[2] Brdyś M, and P D Roberts (1984). Optimal structures for steady-state adaptive optimising control of large scale industrial processes. Research Memorandum No.CEC/MB-PDR/1, Control Eng. Centre, The City University, London.

[3] S Chen, Brdyś M and P D Roberts (1985). An integrated system optimisation and parameter estimation technique for hierarchical control of steady state systems. Research Memorandum No.CEC/SC-MB-PDR/19, Control Eng. Centre, The City University, London. (Accepted by Int. J of Sys. Sci.).

[4] S Chen, P D Roberts and D S Wadhwani (1985). Integrated system optimisation and parameter estimation technique using a distributed hierarchical computer system. Presented at the 1st European Workshop on Parallel Processing Techniques for Simulation, UMIST, Manchester, England, Oct.28-29, 1985.

[5] S Chen, D S Wadhwani, and P D Roberts (1985). Asynchronous serial data communication between a BBC microcomputer and a DEC LSI11/23 minicomputer. Research Memorandum No.CEC/SC/DSW/PDR-26, Control Eng. Centre, The City University, London.

[6] Findeisen W, F N Bailey, M Brdyś, K Malinowski, P Tatjewski, and A Wosniak (1980). Control and Coordination in Hierarchical Systems. Wiley, London.

[7] Michalska M, J E Ellis, and P D Roberts (1984). Joint coordination method for the optimising control of large scale systems. Research Memorandum No.CEC/HM-JEE -PDR/2, Control Eng. Centre The City University, London.

[8] P D Roberts (1979). An algorithm for steady-state system optimization and parameter estimation. Int. J. of Sys. Sci., Vol.10, No.7, pp.719-734.

[9]P D Roberts, C W Li, I A Stevenson, and D S Wadhwani (1984). On-line distributed hierarchical control and optimization of large scale processes using a micro-computer based system. Preprints of the 1st European Workshop on the 'Real Time Control of Large Scale System', University of Patras, Greece, pp.432-441.

[10] I A Stevenson (1985) On-line Adaptive Computer Control of Travelling Load Furnace. PhD Thesis, Control Eng. Centre, The City University, London.

[11] S Chen (1986). Integrated System Optimisation and Parameter Estimation Methods for On-line Control of Industrial Processes. PhD Thesis, Control Eng. Centre, The City University, London.

TABLE 1 Comparison of results

	c_{11}	c_{12}	c_{21}	c_{22}	c_{23}	$Q(c,y^*)$
real optimum	-0.72	0.12	0.90	1.00	-0.83	5.93
SIA	-0.72	0.12	0.89	1.00	-0.85	5.93
DIA	-0.71	0.12	0.90	1.00	-0.83	5.93

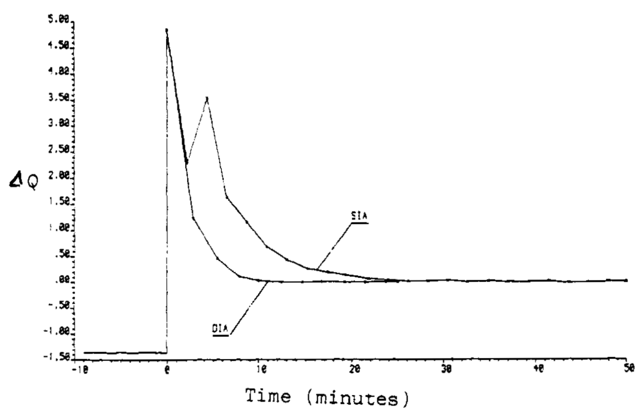

Fig. 2. Adaptability of SIA and DIA
(ΔQ = real performance - 7.3
Plant settling time period = 20 seconds)

TABLE 2 Comparison of efficiency (Plant settling time period = 20 seconds)

algorithm	iterations of modifiers	optimisation iterations	set point changes	total information interchanges	computation time (minutes)
SIA	23	24	138	162	50.4
DIA	11	113	66	179	27.2

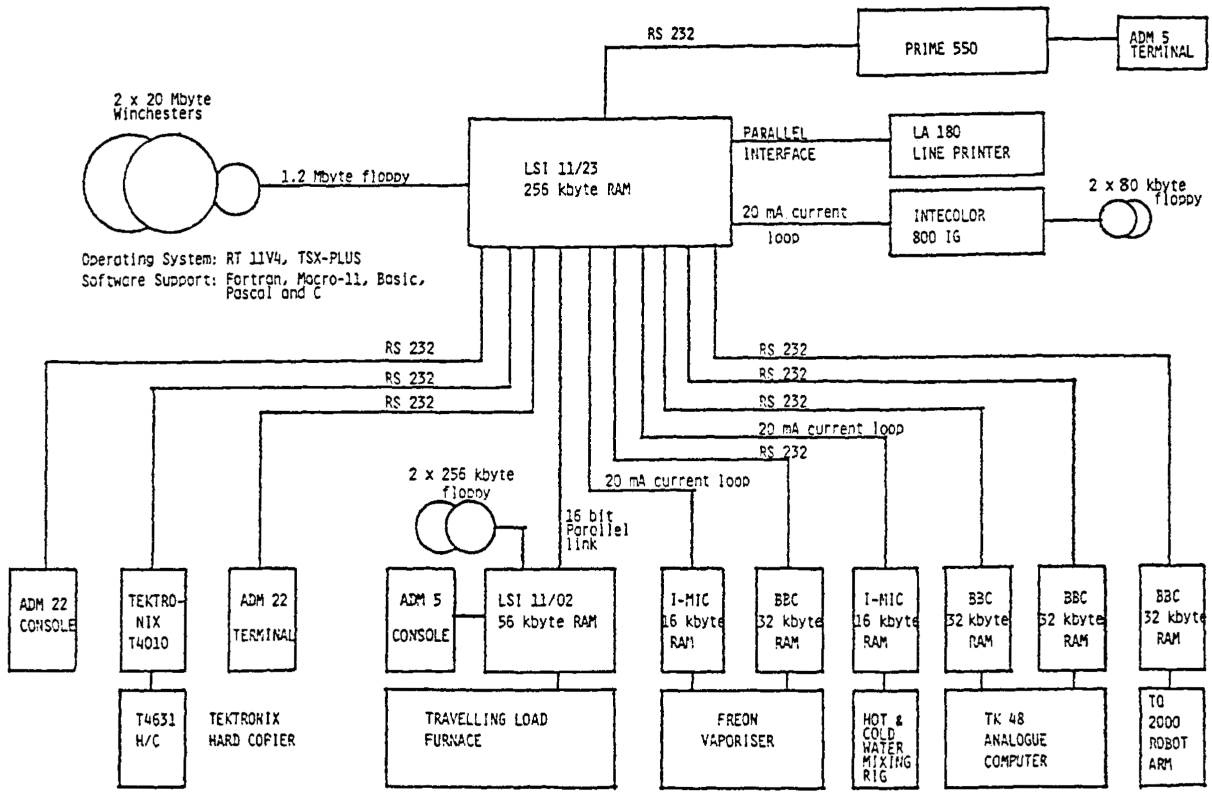

Fig. 1. Distributed computer network

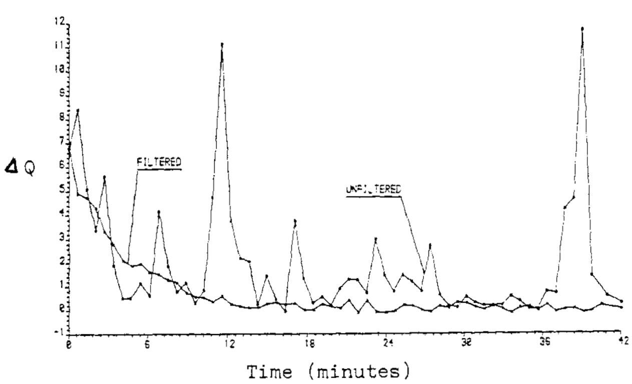

Fig. 3. Performance of SIA in stochastic situation
(ΔQ = real performance - 5.93
Plant settling time period = 5 seconds)

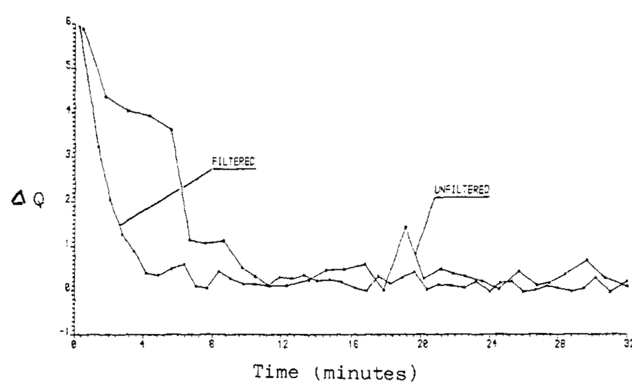

Fig. 4. Performance of DIA in stochastic situation
(ΔQ = real performance - 5.93
Plant settling time period = 5 seconds