MATHEMATICAL ALGORITHMS FOR OPTIMISATION OF LARGE SCALE SYSTEMS

BY

ZEINAB MOHAMED HENDAWY

A THESIS SUBMITTED FOR THE DEGREE OF DOCTOR OF PHILOSOPHY IN ENGINEERING

CONTROL ENGINEERING CENTRE
SCHOOL OF ELECTRICAL ENGINEERING
AND APPLIED PHYSICS

CITY UNIVERSITY, LONDON

OCTOBER 1989

CONTENTS

LIST OF TABLES

LIST OF FIGURES

ACKNOWLEDGEMENT

I wish to express my sincere gratitude and appreciation to Professor P. D. ROBERTS, my research supervisor for his invaluable advice, guidance, encourgement, understanding character and caring attitude throughout my research work.

I also wish to thank Dr. J. LIN of Xian Jai Tong University, of China for his advise and contribution in this research. Some of the *work* in this thesis have been suggested by him.

Thanks are also due to Dr. J. ELLIS, of the Control Engineering Centre of City University, Dr. P. Tatjewski and Dr. M. Brdyś of the Technical University of Warsaw for their interesting constructive discussions. In addition, I would like to express my gratitude to Dr. G. Pooler for his assistance using the Prime computer.

I would also like to thank my collegues in Control Engineering Centre and City University especially S. Filali, Zohir, Iman, Chaffi, Zari, Salim and Amal for their assistance in preparing and typing this thesis.

Special thanks are due to my husband for his enduring patience, encouragement and constant support.

I am also indebted to Mrs. J. RIVELLINI, the secretary of the Centre who assisted me during the work on this project.

Finally, I would like to thank the Egyptian Government for supporting this research.

Declaration

The author grants Powers of discretion to the University Librarian to allow this thesis to be copied in whole or part without further reference to her. This permission covers only single copies made for study purposes, subject to normal conditions of acknowledgement.

Publications

The following paper, based on the work described in this thesis is submitted to the European Conference for Mathematics in Industry (ECMI), 1988 and will be published soon :

Hendawy, Z. M. and Roberts, P. D. : " An On-Line Augmented Price Correction Technique for hierarchical control of interconnected industrial processes".

The following papers based on the work described in this thesis have been published:

Hendawy, Z. M. and Roberts, P. D. (1989) : " A Price Correction Mechanism using an Augmented Interaction Balance Method with Feedback of Large Scale Industrial Processes : Derivation and Applicability".
Advances in Modelling and Simulation, AMSE press, Vol. 16, No. 1, pp 53 - 64.

Lin, J. ; Hendawy, Z. M. and Roberts, P. D. (1988) : " Extension of Integrated System Optimisation and Parameter Estimation to Hierarchical Control of Steady State Systems with Output-Dependent Constraints ". Int. J. Control, Vol. 47, No. 2, pp 413 - 431.

Lin, J. ; Hendawy, Z. M. and Roberts, P. D. (1988) : " New Model-Based Double Loop Iterative Strategy for Integrated System Optimisation and Parameter Estimation of Large Scale Industrial Processes". Int. J. Control, Vol. 47 ,No. 3 , pp 753 - 773.

Lin, J. ; Hendawy, Z. M. and Roberts, P. D. (1988) : " A Modified Integrated Optimisition and Parameter Estimation Technique for Hierarchical Control of Steady State Systems". AMSE press, Vol. 12, No. 2, pp 43 - 52.

ABSTRACT

This research is concerned with the problem of optimisation of steady state large scale systems using mathematical models. Algorithms for on-line optimisation of interconnected industrial processes are investigated. The research is concerned with two different kinds of algorithms which are based on the structure of the model used and the way of incorporating the real process information in order to compensate for model-reality differences.

The first class of algorithms are developed from the price method with global feedback information which is mainly based on the normal Lagrangian function. Two existing algorithms are examined:

The double iterative price correction mechanism and the augmented interaction balance method. Both methods use a double iterative coordination strategy and global feedback measurements from the real process. They are based respectively on the normal and the augmented Lagrangian functions. Hence, the first algorithm can only be recommended for application to convex problems. An algorithm, namely the augmented price correction mechanism, has been developed to extend the applicability of the previous price correction mechanism to non-convex problems. It is also applicable to convex problems with the advantage of reducing the number of times that information is required from the real process.

The second class of algorithms is known as integrated system optimisation and parameter estimation (ISOPE) . The model used contains uncertain parameters and the algorithm solves the optimisation and parameter estimation tasks repeatedly until no furthur improvement is obtained. Developed ISOPE algorithms are involved in this research to cover the problems with output dependent constraints.

Simulation results show superiority of the double iterative algorithm over that of single loop method in considerably reducing the number of times that information is required from the real process and hence saving on-line computing time.

It is hoped that this work will provide useful information for implementing and furthur developing on-line steady state optimisation techniques.

LIST OF SYMBOLS

Symbol	Description	Chapter
a_*	Reality monotonic constant	8
c	control (set point) vector	2, 4 - 8
$\mathcal{CUY}$	feasible set of control, input and output mapping	4 - 8
E^n	n - dimensional Euclidean space	3
F (.)	input-output mappings	2, 4 - 8
f, f (x)	real valued function	3
g (.), G (.)	unequality constraint mapping	2, 4 - 8
H, H_i, H_{ij}	interconnection matrices	2, 4 - 8
Hk*(v)	reality interconnection input vector	4 - 8
h (x)	equality constraint mappings	3
I	identity matrix	4 - 8
IT, IS	number of total iterations and system iterations	4, 5
K (.)	control- output mappings	4 - 8
$\mathcal{L}$	normal lagrangian function	4 - 8
L_a , Λ	augmented lagrangian function	4, 5
Q, q	performance index	2, 4 - 8
$\mathcal{R}^a$	a- dimensional real space	4 - 8
s	shift variable vector	4, 5
Tol,Tol_1, Tol_2	tolerances of iterative loops	4, 5
u	interaction input	2, 4 - 8
v	control vector (applied to the process)	7, 8
w	input vector (applied to the process)	7, 8
y	output vector (interaction)	2,4 - 8
z	disturbance vector	2
Δ	imbalance	7, 8
α	model parameter vector	6 - 8
α_i	gain coefficient parameter	4
ϵ, δ	small positive increment	3
ϵ_1, ϵ_2	gain coefficient for updating the price vector λ and the shift vector	4 , 5
ϵ_p, ϵ_v, ϵ_ξ , ϵ_w	gain coefficient parameter	6 - 8

η, λ, μ, ξ	Lagrange multipliers	4 - 8
ρ	penality coefficient	3 - 5, 8
Ω	solution set	7, 8
$(.)^*$	denoting the reality of (.)	6 - 8
	the minimum of (.)	3
$(.)^\wedge$	optimal value of (.) obtained by the local decision unit	4 - 8
$(.)^0$	starting point of (.)	4 - 8
$(.)^t, (.)^{-1}$	the transpose and the inverse of (.)	4 - 8
$\in , \notin$	belongs to , does not belong to	4 - 8
$\forall$	for all	4 - 8
$< . , . >$	scalar product of . and .	4 - 8
$\triangleq$	equals by definition, denotes	4 - 8
$\|\cdot\|$	Eculidean norm	4 - 8

LIST OF ABBREVIATIONS

Abbreviation	Description
AIBMF	Augmented Interaction Balance Method with feedback
ALM	Approximate Linear Model Approach
CP	Coordinator problem
cpt	Computer processing time
DIPCM	Double Iterative Price Correction Mchanism
IBM	Interaction Balance Method
IBMF	Interaction Balance Method with feedback
inf	infimum
ISOPE	Integrated System Optimisation and Parameter Estimation
EOP	equivalent optimisation problem
K-T	Kuhn-Tucker
LP, LCP	local control problem
Lim	limit of
min	minimise
MTS	Modified Two Step Approach
NAG	Numerical Algorithm Group
NDIA	New Double Iterative Algorithm; chapter 8
ROP	real optimisation problem
SIA	Single Iterative Loop Algorithm; chapter 7
Sub.	Subsystem
sup	Supremum
V1	Version 1 of the Double Iterative Price Correction Mechanism (DIPCM), chapter 4, 5
V2	Version 2 of the DIPCM.

Chapter 1

Introduction

1.1 Computer control of industrial processes:

The field of computer control is growing rapidly and achieving wide spread industrial recognition and acceptance. Forecasts indicate that computing power is expected to continue to increase and be available at moderate cost. This is mainly due to continuing advances in microelectronics technology and computing.

Today, in general, computers are the first choice for control where they can be used to perform several tasks. Typically, these tasks include measurements, actuation, direct digital control, hierarchical functions, operator and management functions (Roberts, 1988a).

For large scale systems, complex and multivariable processes, the control objective is fundamentally to achieve economically superior process performance. An ideal controller would take into consideration all significant interactions among process variables, as well as their economic impact, and apply control action which will result in optimum process performance. In controlling an industrial process, it is common practice to split the control action into two parts: the follow-up or regulatory control and the supervisory or optimising control. The regulatory control is responsible for keeping the chosen process variables at their desired values in spite of fast disturbances acting upon the process. The supervisory control is responsible for determining and maintaining optimal values of the

set points.

The main concern of this thesis is to describe and investigate some techniques which can be implemented to solve the optimising control problem.

1.2 Steady-state optimisation and control:

Optimising control is difficult to accomplish because disturbances continually upset the process. Indeed, if it were not for disturbances, no control action would ever be required. Thus a process might be operating smoothly at optimum conditions but, when a disturbance occurs, some changes must be made in the process in order to counteract the disturbance.

Process optimisation problems may be solved by three ways. These are by utilising the system measurements directly or indirectly using a mathematical model of the process. The third is a combination of these two methods where the optimisation process is based on a mathematical model and also incorporates measurements from the process.

1.2.1 Direct process optimisation:

In direct process optimisation, a continual experimentation on the system is required in order to arrive empirically at the optimum operating point. Direct experimental optimisation could be practical for a process which involves only a small number of variables, and where the dynamic response is rapid with respect to the frequency of disturbances. However, for a multivariable complex process and a process with a slow dynamic response, the approach has two major disadvantages which severely limit its applicability. For a multivariable process, a large number of experiments are required and for each experiment the process must be allowed to reach the steady state before making another move. If the system response is slow, this method can be time consuming. It is not usually recommended for optimisation (Savas, 1965).

1.2.2 Mathematical model-based optimisation:

Optimisation using a mathematical model of the process has two major advantages over the direct methods. Firstly, the system need not be perturbed if a model is employed. Secondly, by simulation work on the computer, even processes with a large number of variables can be optimised rapidly.

However, the crucial problem in this method is the accuracy of the model used compared to the process. In order to guarantee the system optimum, the model representing the process has to satisfy certain sufficiency conditions (Durbeck, 1965; Foord, 1974; Ellis and Roberts, 1982). The process output derivatives with respect to the controller set points have to match exactly with their corresponding model derivatives. To satisfy these conditions, accurate models are required which are generally difficult to obtain. This is due to external uncertainties, like measurement noise, environmental conditions, etc. which can not be represented accurately in a model. Furthermore, if the structure of the model is assumed to be certain, so that it is a faithful representation of the system, such a model will be too complicated and this will tend to increase the computational complexity of the optimisation process. The simulation of such complex models is slower than real-time if standard solution techniques are used together with modest present-day computer technology (Singh et al., 1985; Allidina, Buro and Malinowski, 1985).

1.2.3 Optimisation using mathematical model and process measurement:

To overcome model-reality differences, it has long been recognised that measurements from the system must be incorporated in the optimisation process to accomodate model deficiencies. The measurements from the real system can be used in different ways. First, the measured feedback information may be used by the coordinator or by the local decision units. For example, Findeisen and co-workers (1978, 1980) give a variety of methods for optimisation of large scale systems. These methods are designed for hierarchical structures. They are based on a mathematical (fixed) model and use feedback measurements to improve the model based

solution in the form of an iterative procedure.

There are, in general, two principal methods of coordination used in hierarchical structures. The coordinator may set, for the local decision units, its desired values of the outputs and inputs (Direct coordination), or it may set prices on the outputs and inputs (Price coordination).

The obtained solution using the direct or the price coordination methods with feedback information is often better than the purely model based solution. However, they, in general, converge to suboptimal results.

Secondly, process measurements may also be used to overcome the model uncertainty by adopting a two step approach; optimisation and parameter estimation (for example, Youle and Duncanson, 1970; Durbeck, 1965; Foord, 1974). The parameter estimation problem can be solved by comparing output responses from the model with the corresponding responses from the real system. The estimated parameters are used to update the model which in turn is used in determining the optimum set point values. The two problems of optimisation and parameter estimation are solved repeatedly until, hopefully, the iterative process converges to the optimum. However, simply combining the optimisation and parameter estimation does not guarantee the optimum solution.

1.3 Scope of the research:

As was mentioned earlier, the mathematical model has to satisfy certain sufficiency conditions (Durbeck, 1965; Foord, 1974; Ellis and Roberts, 1982). This demand, in general, requires that the model structure is the same as that of the real process. For many industrial processes, their structures are highly uncertain and, hence, such a requirement is obviously unrealistic.

Roberts (1978) proposed the modified two step (MTS) approach which is an improved version of the standard two step approach (Roberts, 1977). It involves an iterative procedure for updating model parameters and modifying the model based optimisation problem. The optimisation

performance index is modified to incorporate an extra term which caters for any mismatch between model and process output derivatives with respect to controller set points. Therefore, by measuring the differences of these two sets of derivatives, the real optimal steady-state operating condition can be achieved in spite of model-reality differences. The method has proved to be successful in providing optimum solutions to many problems (Roberts, 1979; Roberts and Williams, 1981; Roberts and Lalui, 1982; Stevenson, Brdyś and Roberts, 1985; Ellis and Roberts, 1981, 1982, 1985).

Since the MTS approach couples the optimisation and parameter estimation together, it is sometimes called the integrated system optimisation and parameter estimation (ISOPE) technique. The original version of the MTS approach was first proposed (Roberts, 1978) for centralised systems and the process inequality constraints were assumed to be independent of process outputs. The often required large number of set point changes was also a problem.

The applicability of the modified two step approach was extended to large scale interconnected processes (Michalska, Ellis and Roberts, 1985; Brdyś and Roberts, 1986) by incorporating the price method (Findeisen et. al., 1980). A group of hierachical adaptive optimal algorithms were proposed (Brdyś and Roberts, 1986) which utilise whatever is available from the real process measurements. Some other proposed algorithms are: the single loop technique, system based double loop and the model based double loop techniques (Chen, Brdyś and Roberts, 1986; Brdyś, Abdullah and Roberts, 1986).

The model based double loop techniques have successfully reduced the number of set point changes compared with the single loop technique (Chen, Brdyś and Roberts, 1986).

In an attempt to extend the modified two step approach to encompass a more general class of problems, a certain technique has been introduced to deal with output-dependent inequality constraints (Chen, Brdyś and Roberts, 1986). However, the feasibility of the model-based optimisation solution of this algorithm is not ensured during the course of iteration and, also, the convergence conditions of the algorithm have not been

derived.

In this thesis, the research will be concentrated on two main areas:

Firstly: For the integrated system optimisation and parameter estimation technique, two algorithms will be developed to cater for output dependent inequality constraints and reduce the number of set point changes and consequently the on-line computing time.

Secondly: For the price coordination method with global feedback, a comprehensive study of the double iterative price correction mechanism (Shao and Roberts, 1983) and the augmented interaction balance method with feedback (Tatjewski, 1985) will be presented. A developed algorithm based on the augmented Lagrangian will be introduced. It extends the applicability of the double iterative price correction mechanism to non-convex problems.

Optimality and sufficient conditions for local convergence of the ISOPE algorithms will be derived. A comprehensive simulation study of all the algorithms will also be carried out.

1.4 Layout of the thesis:

This thesis consists of nine chapters which can be divided into four groups as follows:

The first group contains chapters 2 and 3, where they respectively provide a broad scientific and mathematical background on which the research reported in this thesis is based on.

Chapter 2 introduces the basic concepts of multilayer and multilevel structures used in hierarchical control systems. These concepts provide a background for developing hierarchical control used in designing complex control systems for large scale industrial processes.

In chapter 3, some basic mathematical definitions, relations, notation and theories are summarized.

The second group of the thesis contains chapters 4 and 5 where the main concern is devoted to three algorithms based on the price coordination method in an iterative structure using feedback information from the real process.

Chapter 4 presents an extensive simulation study of the double iterative price correction mechanism (Shao and Roberts, 1983) and the augmented interaction balance method (Tatjewski, 1985). Various aspects are studied including non-linearity and sensitivity.

Chapter 5 provides an algorithm which extends the applicability of the double iterative price correction mechanism to the non-convex problems. Discussion of the optimality conditions is presented and a simulation study to test the algorithm is also provided.

The third group of the thesis contains chapters 6, 7 and 8 where the integrated system optimisation and parameter estimation technique is their main concern. A brief summary of the centralised and the decentralised ISOPE techniques is presented in chapter 6.

Chapter 7 provides a new ISOPE algorithm which is an extension to the modified two step approach to include problems with inequality output dependent constraints. It has the advantages of ensuring the feasibility of the model based optimisation solution during the course of iteration.

Chapter 8 is devoted to another developed ISOPE technique. It has a double loop iterative strategy with self-adaptive properties where derivatives from the real process measurements are used to update the model. The optimality of the algorithm is examined and computer simulation is presented to demonstrate the behaviour of the algorithm.

The fourth and the final part of the thesis is represented by chapter 9 which summarizes the conclusions of this research and gives suggestions for further work.

Chapter 2

Control and Optimisation of large scale systems

2.1 The Complex System

Many of today's industrial systems are often too complicated to comprehend in their entirety. An example of a complex system is a large scale industrial process which can be considered as a collection of interconnected subsystems. The properties of the integrated systems are defined by understanding the properties of each individual subsystem together with understanding the interconnection between the subsystems, including the interconnections with the environment.

A complex system can be viewed as an arrangement of elements or subsystems in which outputs of one subsystem are connected with inputs of another subsystems, as shown in figure (2.1). Another structure can be used to describe most industrial processes by introducing an orderly input-output interconnection matrix H, as shown in figure (2.2). The matrix H represents the structure of the system. Each row of this matrix is associated with a single input of a subsystem. The elements in the rows are zeros except where a one shows the single output to which the given input is connected.

By exploiting the structure of a large scale system as an interconnected assembly of subsystems, it is possible to decompose the problem of controlling a complex system into interlinked subproblems of manageable size. Each subproblem can be solved independently of the other

subproblems, with some form of coordination procedure to account for the interconnections and to ensure that the overall system objectives and constraints are satisfied, as shown in figure (2.3). This is usually done via an iterative information exhange between the lower level and the coordinator level. Such an approach leads to decentralised and hierarchical control structures.

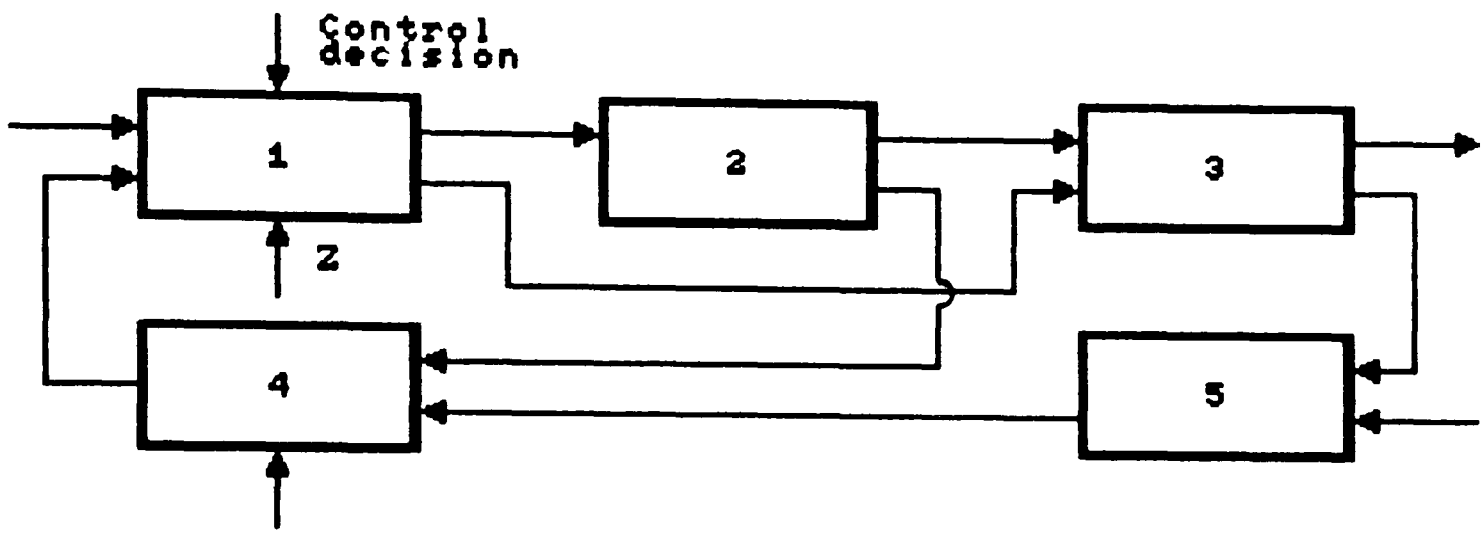


Figure (2.1) Complex system

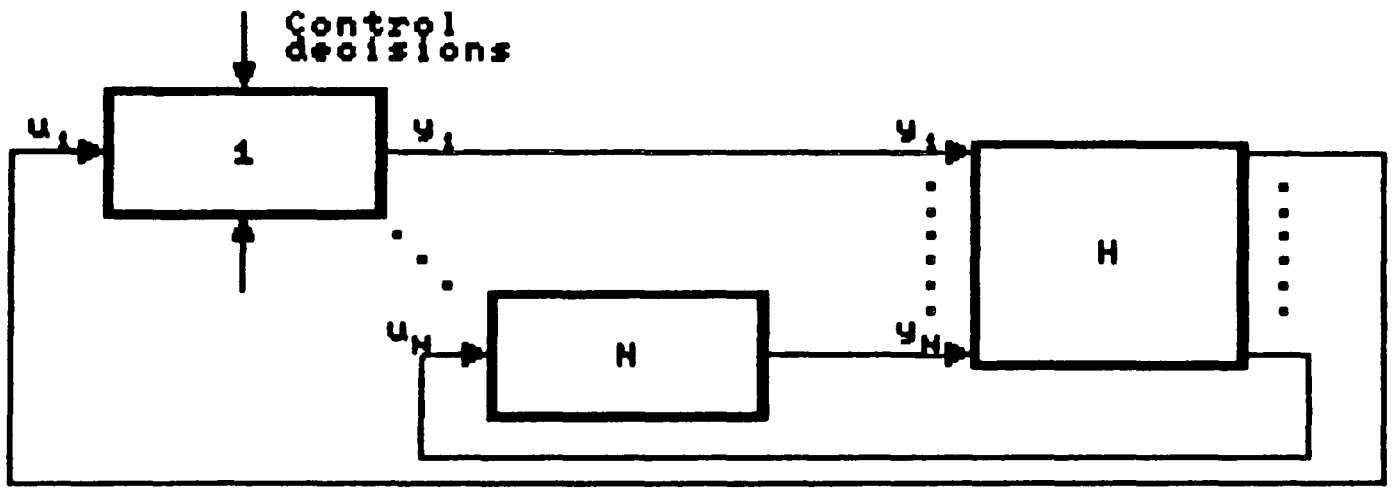


Figure (2.2) Complex system presented
with an ordering matrix H.

2.2 Control Structure of Large Scale Systems

Computers can be used in process control to solve the system optimisation problem in two different schemes; centralised or decentralised control. In centralised control, all the system information is available at a central location in which all the system variables are manipulated and control decisions are taken directly from that centre as shown in figure (2.4). In decentralised control, the system to be controlled is divided into individual subsystems and each subsystem has its own local decision unit to solve its control task as shown in figure (2.5). Such a system has benefits of improved control, reliability, flexibility and reduced cabling costs.

However, it is recognised that decentralised control may produce sub-optimum results (Maxwell, 1983) if each local decision unit has not taken into account the effect of the other subsystems. Such a conflict may occur because the optimum for the individual subsystems does not need to give rise to the global optimum of the overall system objectives. Therefore, the coordination unit has an important task of coordinating the infimal level units and preventing any conflicts between the individual decisions. In addition, the coodinator has to solve another problem (coordinator problem) due to the decomposition of the global problem into several subproblems.

A decentralised control system can also be hierarchical if the information subsets of some subsystems depend directly on the action of other subsystems at higher or lower levels.

A hierarchical control system consists of local decision units arranged in a priority structure, where at each level a number of units may operate in parallel giving rise to a pyramid structure as shown in figure (2.6).

The decision of implementing a specific control scheme will depend on some factors such as the size of the system, the cost and the achieved overall efficiency (Maxwell, 1982). However, the low cost of communication components have attracted control system engineers to decentralised control and forced them to rethink the way computers in process control (Maxwell, 1983) can be used.

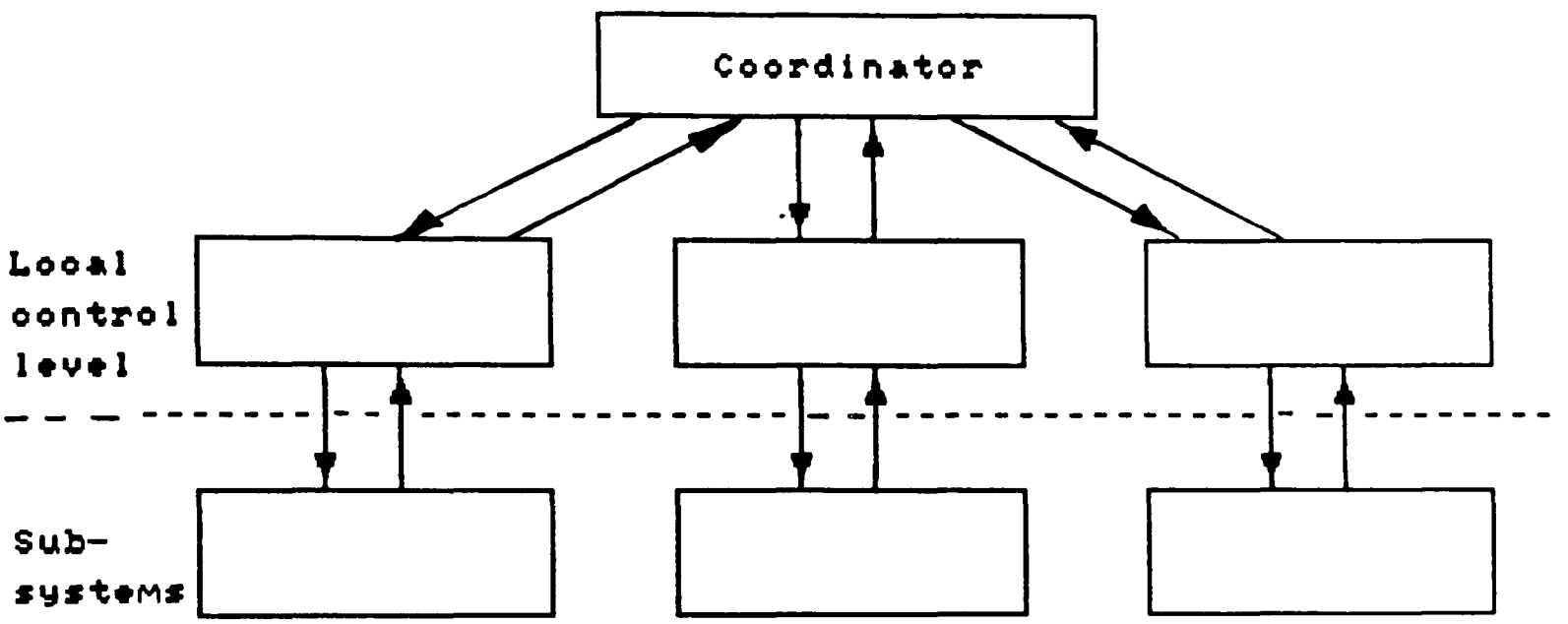


Figure (2.3) Decentralised control system with coordination.

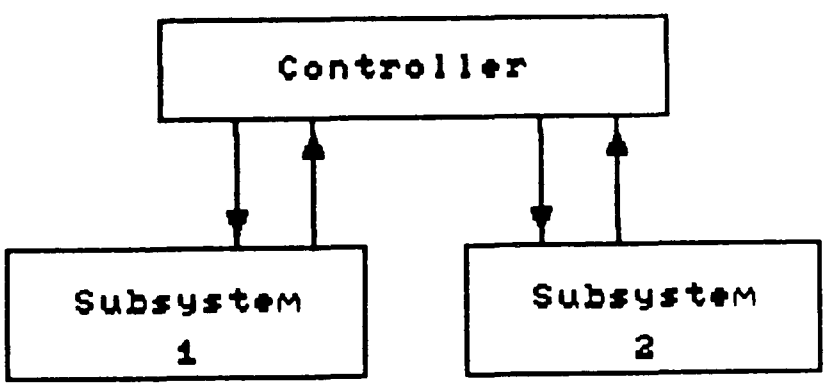


Figure (2.4) Control structure in centralised control.

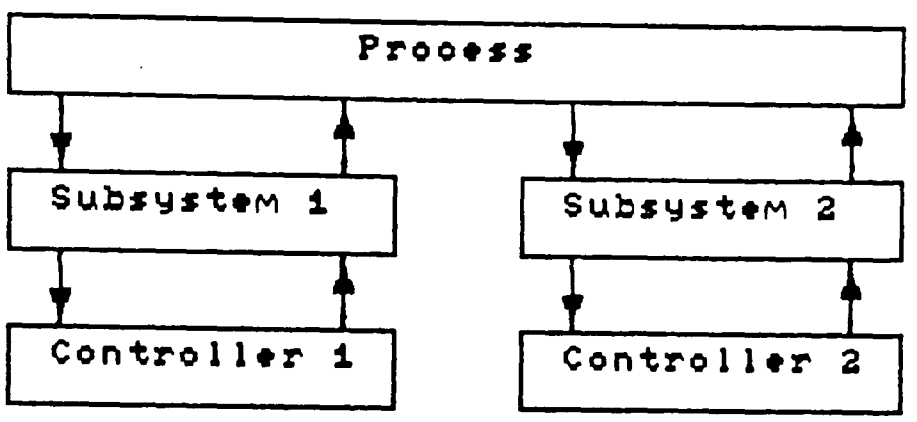


Figure (2.5) Decentralised control structure.

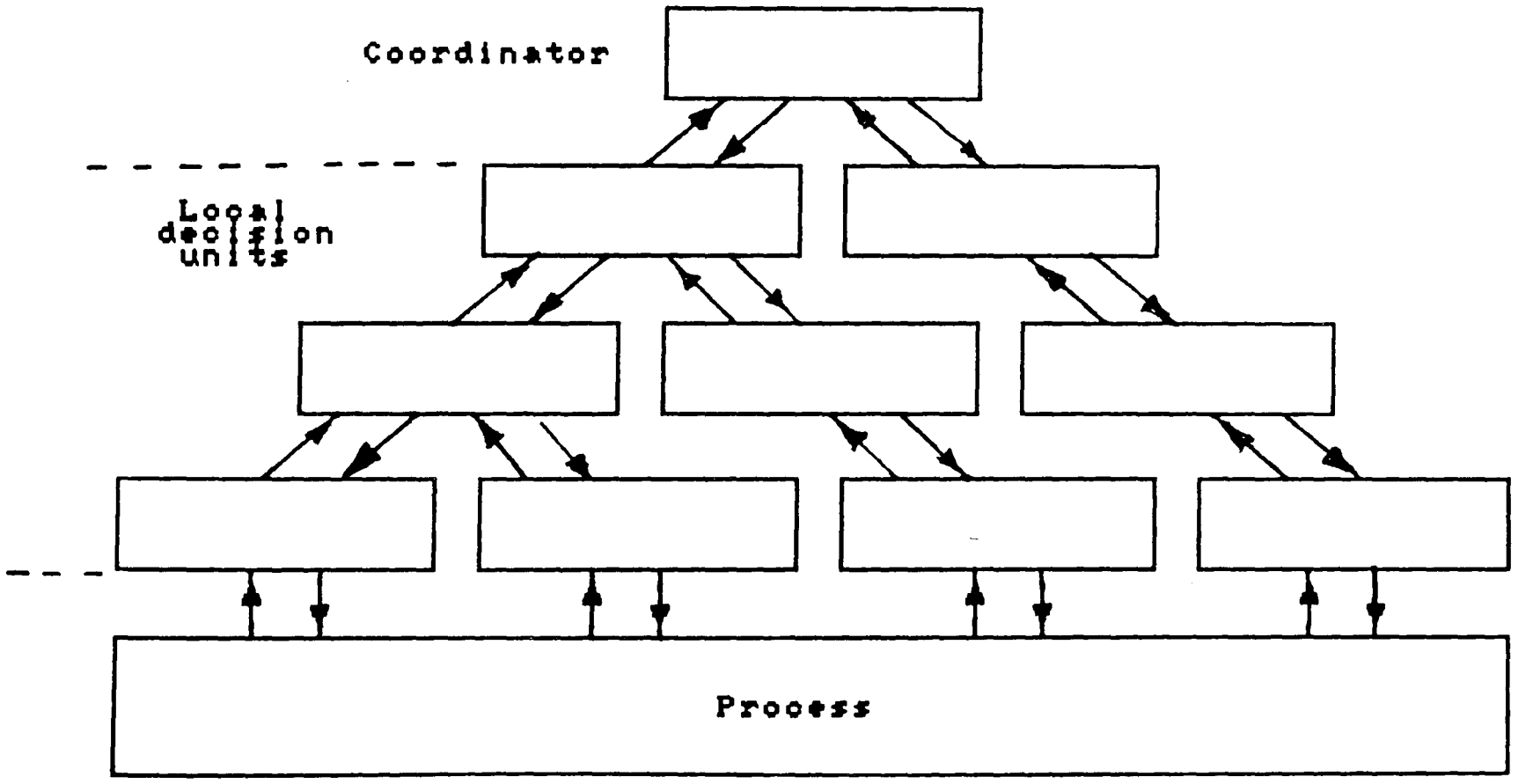


Figure (2.6) A hierarchical multi-level structure.

There are some reasons for favouring decentralised and hierarchical control of large scale systems (Roberts, 1988b) in comparison with a centralised sytem, such as :

1. The division of the decision makes it easier to handle a complex organisation;

2. The subsystems of a large organisation often are distant from one another and the transmission of information is both expensive and subject to distortion;

3. If one of the control units breaks down, the system will still survive;

4. The possibility that the system as a whole will be less sensitive to disturbance inputs if the local units can respond faster and more adequately than a more remote control decision unit; and

5. It is faster to perform more tasks in a given period of time if the jobs are performed in parallel.

2.3 Basic types of hierarchical structure.

In dealing with large scale systems, complexity is generally recognised where in the process industry many complex plants are difficult to comprehend as a whole. A natural way of viewing such a plant is to consider it as a collection of interconnected subprocess. Hence, it is possible to decompose the problem of controlling a complex industrial process into a number of interlinked smaller problems of managable size. The subproblems are solved more easily and independently of each other with a coordination procedure to account for the interconnections. This approach leads to integrated hierarchical control structures.

There are three basic types of hierarchical structures (Mesarovic, 1970), these are:

1. The stratified description;
2. The mutilevel description; and
3. The multilayer description.

It is important to note here that a system may belongs to more than one class of these description.

2.3.1 The stratified description

The stratum approach is concerned with describing the system by a family of models. Each model is concerned with the behaviour of the system as viewed from different level of abstraction. These levels are referred to as strata, where the lower levels are assigned more specialised description and details of the large scale system than the higher levels. The problem is separated into a number of smaller better defined subproblems and each of the subproblems are solved separately. It is necessary that the functioning on any level be as independent as possible of the functioning on other levels.

The characteristics of the stratified decomposition are :

i) Each stratum has a different task than others;

ii) The higher strata have priority over the lower ones;

iii) Each strata considers a different time horizon where the higher the level, the longer horizon they have and the less often the control action takes place. Figure (2.7) illustrates the stratified decomposition approach.

2.3.2 The multilevel description

The multilevel control description is one of the most general hierarchical structures. The system is divided into a family of interacting subsystems. Some of the subsystems are defined as decision making units which are arranged in a hierarchy having a pyramid structure and a principal characteristic is the existence of a supremal (top level) unit, as shown in figure (2.6). In general, the various decision units have conflicting goals but the units in a certain level can coordinate those in a lower level to them and be coordinated by a higher level one.

At an intermediate level each decision unit receives and transmits information from units superior to it in the hierarchy and also to those units inferior to it. The conflicts between decision units are resolved by the higher level units which play the role of coordinators and assure global solution in the end.

2.3.3 The multilayer description

This structure is a direct outcome of the complexities involved in a decision making process. The multilayer control hierarchy (Lefkowitz, 1966,1977) provides a systematic procedure for resolving the control task. In the control of a large scale system using the multilayer approach, the overall problem is naturally split into a set of subproblems which are identified by four functional aspects of the overall control problem. The layers are :

i) **Regulation** or **direct control** whose task is to maintain the chosen process variables at their desired values in spite of fast disturbances acting upon the process.

ii) **Supervisory** or **optimising control** whose task is to determine the optimal set points of the process according to a defined criterion. This layer often employs a mathematical model which is only valid for a given set of circumstances. As these change with time, the employed mathematical model contains uncertain parameters which may be updated according to the new state of the process. The values of the uncertain parameters may be prescribed by the next upper layer in the hierarchy (learning layer).

iii) **The learning** or **adaption layer** is concerned with adapting or updating the uncertain parameter values used in the mathematical model employed in the optimisation layer.

iv) **The management** or **self-organising layer** which has various tasks such as selecting the structure, functions and strategies for the lower layer so that an overall goal is achieved.

Figure (2.8) illustrates the above functional four layer hierarchy.

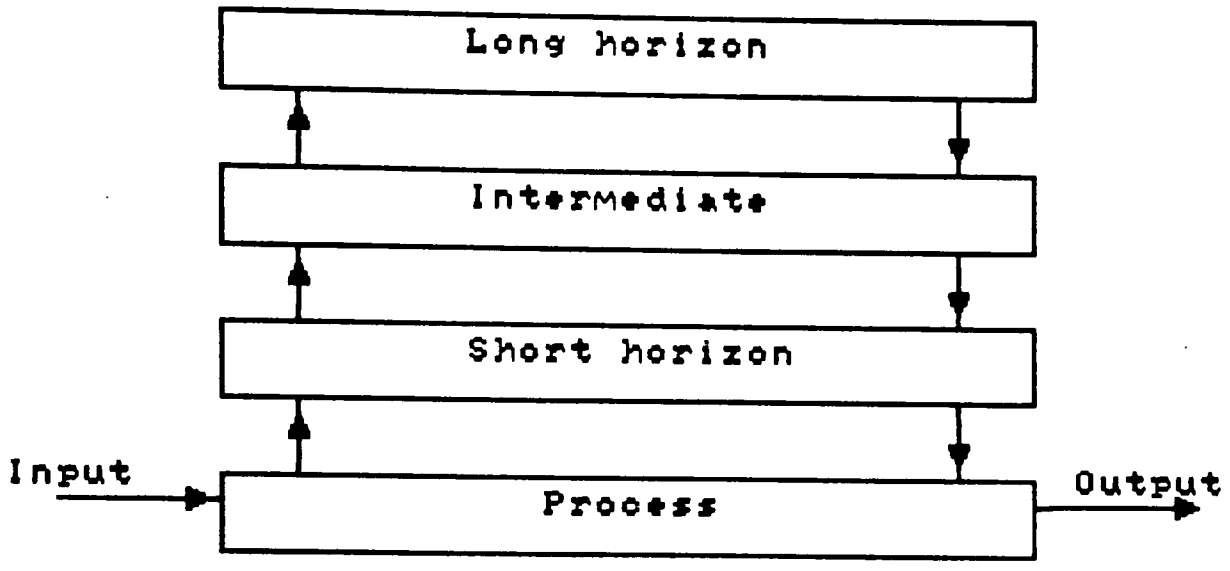


Figure (2.7) The stratified decomposition of control system.

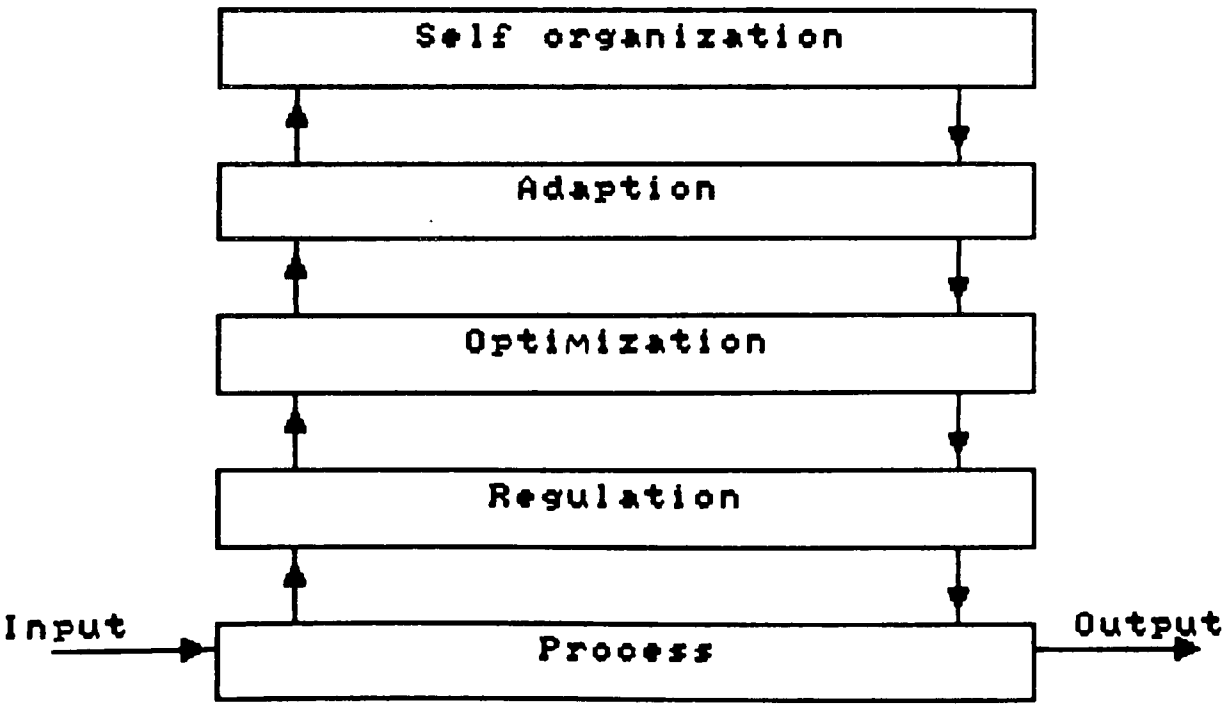


Figure (2.8) Functional-four layer hierarchy.

2.4 Steady state optimisation of large scale systems.

Industrial large scale process, which are regarded as decomposed into interconnected subprocesses, are usually considered to be controlled using a two layer hierarchical structure as shown in figure (2.9). The lower layer of the structure performs direct regulation control where the upper supervisory layer has the responsibility of determining the set points of the regulatory controllers to obtain optimum steady state performance. Each subsystem has its own local regulatory and optimising decision units, and coordination is performed at a supremal level to ensure that overall system objectives and constraints are satisfied. Figure (2.9) shows a two layer hierarchical structure for a process consisting of a two interconnected subsystems.

In the design stage of a large scale system, it is often required to determine the optimal dimensions of unit processes and their optimal operating conditions to satisfy specified overall goals of the process under certain conditions. A process might be operating smoothly at optimum conditions, but when a disturbance occurs, some changes must be made in the process in order to counteract the disturbance and determine the optimum operating set points of the process for the new conditions. This process is called the steady state optimisation problem.

As it was mentioned in chapter 1, there are three methods to solve the optimisation problem of the process in order to determine its optimum conditions. These are :

1. Direct process optimisation.
2. Purely mathematical model based optimisation.
3. Optimisation based on a mathematical model and process measurements.

The third method of optimisation which is based on using a mathematical model with process measurements to overcome the model-reality differences has proved to be the most efficient and reliable of all (Lowe and Hidden, 1971), as it discussed in chapter 1. For the first instance, a simple process model is used to avoid complications of the optimisation problem, then the

process model will be appropriately updated according to the measurements taken from the real system. The process of the optimisation and updating the model is repeated until the optimum operating points of the system is determined.

The optimising control process has three distinct but interrelated computational functions (Savas, 1965) required of the computer in this application : identification, optimisation and control, figure (2.10).

Identification: The identification step has the task of determining the present position or status of the process. It identifies the current values of all the model variables which are used in the optimisation step as a necessary starting point.

Optimisation: In this step, the optimisation calculation is performed in order to determine the optimal settings for the control variables.

Control: This step has the task of determining the strategy necessary to guide the process from the present operating condition to the desired optimum condition.

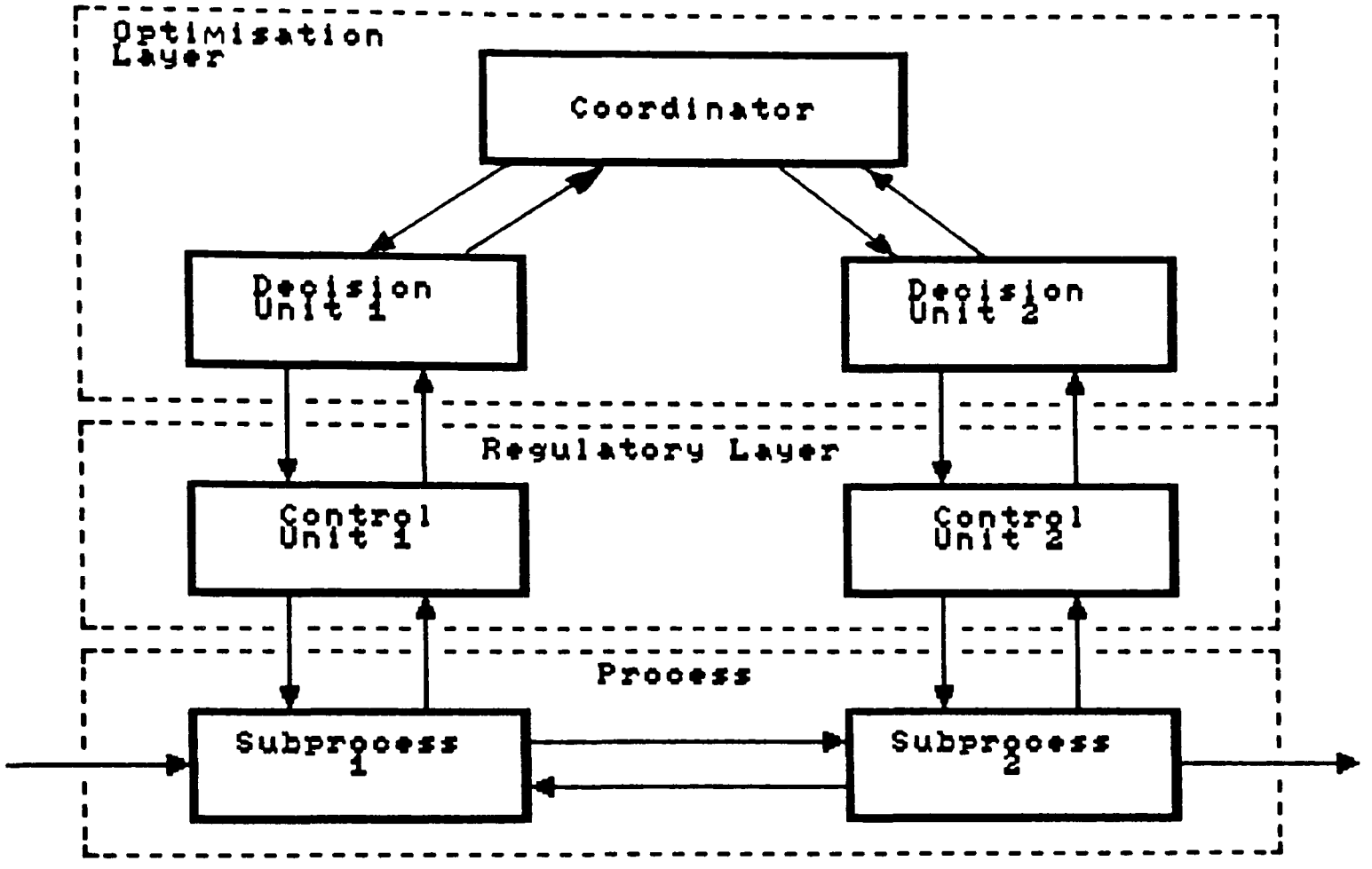


Figure (2.9) Two layer Hierarchical structure.

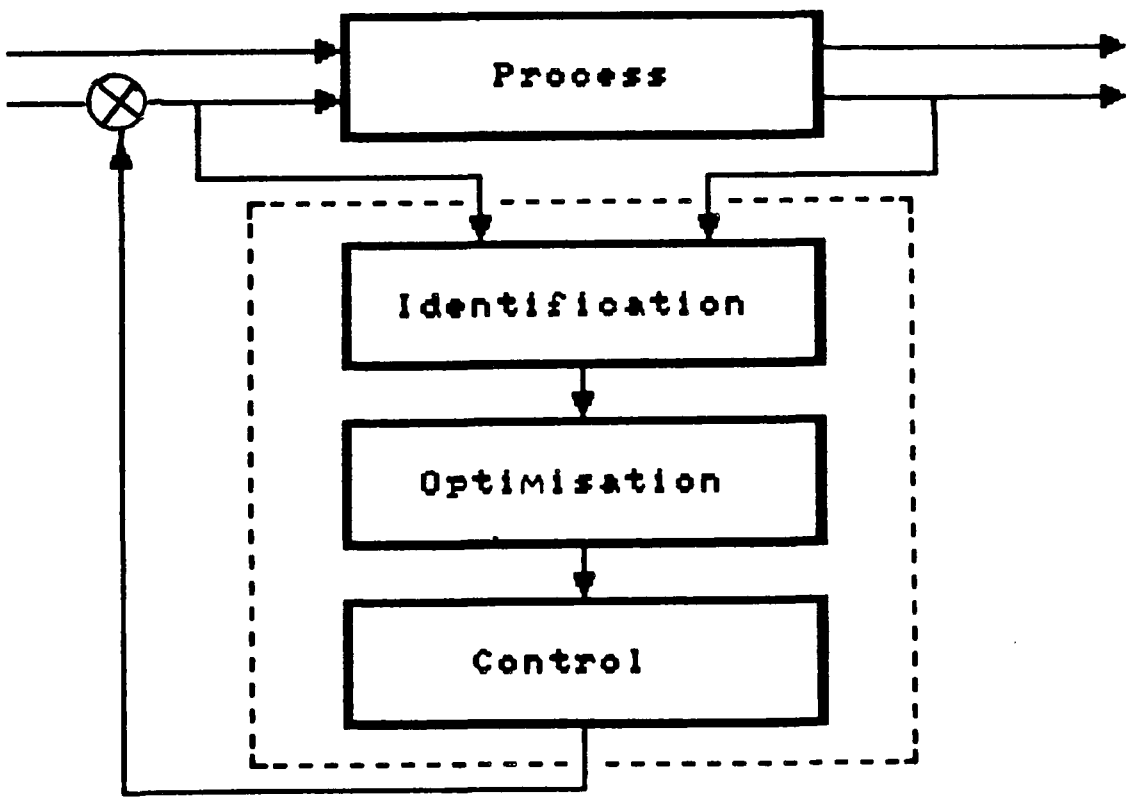


Figure (2.10) Computational function of optimising control.

2.5 Statement of the optimising control problem.

The steady-state performance of the controlled system is described in a decomposed way by the input-output mapping F_i^* as :

$$y_i^* = F_i^* (c_i, u_i^*, z_i^*) \qquad i \in 1 , N \qquad (2.1)$$

where N is the number of subprocesses and the variables c_i , u_i^*, z_i^* and y_i^* are respectively the controller set points, inputs, disturbance inputs and outputs vectors.

The subsystems are assumed to be interconnected as the structure shown in figure (2.2). The interconnection relation between the subprocess is defined by :

$$u_i^* = \sum_{j=1}^{N} H_{ji} \hat{y}_j \qquad i \in 1 , N \qquad (2.2)$$

where H is the interconnection matrix.
ji

The global preformance of the system Q(c,u,y) is assumed to be the summation of the separable indices $Q_i(c_i, u_i, y_i)$ of the subsystems (Findeisen et.al., 1980; Singh and Titli, 1978), ie.

$$Q (c , u^*, y^*) = \sum_{i=1}^{N} Q_i (c_i, u_i^*, y_i^*) \qquad (2.3)$$

In addition, there will be inequality constraints to be satisfied, represented by :

$$g_i^*(c_i, u_i^*, y_i^*, z_i^*) \leq 0 \qquad i \in 1 , N \qquad (2.4)$$

In solving the optimisation problem using a mathematical model the mapping F^* is not known exactly and, in general, approximate equations to reality are used. Thus, the model equations corresponding to equations (2.1) to (2.4) are written as :

$$y_i = F_i(c_i, u_i, z_i) \qquad (2.5)$$

$$u_i = \sum_{i=1}^{N} H_{ji}\, y_i \tag{2.6}$$

$$Q(c, u, y) = \sum_{i=1}^{N} (c_i, u_i, y_i) \tag{2.7}$$

$$g_i(c_i, u_i, y_i, z_i) \leq 0 \tag{2.8}$$

where $i \in 1, N$

Now, the model based overall optimisation problem may be considered in the form :

$$\min_{c,u,y} Q(c, u, y)$$

subject to :

$$y_i = F_i(c_i, u_i, z_i) \tag{2.9}$$

$$u_i = \sum_{j=1}^{N} H_{ji}\, y_i$$

$$g_i(c_i, u_i, y_i, z_i) \leq 0$$

and $i \in 1, N$

2.6 Summary.

In this chapter, a complex or large scale system was defined and control structures of such systems were summarized. Two structures were described; centralised and decentralised control. Experience shows that the decentralised control structure is favourable for flexibility, reliability, parallel processing and economic reasons. The basic types of hierarchical structures were examined. These are : the stratified, the multilevel and the multilayer descriptions, where a system may belong to more than one class of these descriptions.

The steady state optimisation problem is discussed and a statement of the problem is finally introduced. The discussion in this chapter gives a wider environment where the research of this thesis is within.

Chapter 3

Mathematical background

3.1 Introduction:

The purpose of this chapter is to summarize and review some basic definitions, notation and relations that are used in this thesis. It also contains a background of optimisation which is fundamental to the thesis content.

3.2 Topological concepts:

3.2.1 Continuous function:

A real valued function f defined on a subset of E^n is said to be continuous at x if $x_k \longrightarrow x$ implies $f(x_k) \longrightarrow f(x)$. Equivalently, f is continuous at x if given $\epsilon > 0$ there is a $\delta > 0$ such that $| y - x | < \delta$ implies $| f(y) - f(x) | < \epsilon$ (Luenberger, 1984).

3.2.2 Sets:

If x is a member of the set S, we write $x \in S$. The union of two sets S and T is $S \cup T$ and is the set consisting of the elements that belong to either S or T. The intersection of two sets S and T is denoted $S \cap T$ and is the set which contains the elements belonging to both S and T. If S is a subset of T, that is, if every member of S is also a member of T, we write $S \subset T$ or $T \supset S$.

Minimisation of a function f over the set S can be represented by either of these two ways:

$$\min_{x \in S} f(x) \qquad \text{or} \qquad \min \{ f(x) : x \in S \}$$

3.2.3 Bounded and closed sets:

A bounded set is a set which has both an upper bound (or supremum) and a lower bound (or infimum). Upper and lower bounds of a set S are denoted respectively by:

$$\sup_{x \in S} (x) \quad , \quad \inf_{x \in S} (x)$$

A set S is closed if every point that is arbitrarily close to the set S is a member of S.

3.2.4 Sequence and limit point:

A sequence of vectors $x_0, x_1, \ldots, x_k, \ldots$, denoted by $\{ x_k \}_k^{\infty} = 0$ (the index set is understood simply by $\{ x_k \}$), is said to converge to the limit x if:

$$| x_k - x | \longrightarrow 0 \qquad \text{as} \qquad k \longrightarrow \infty.$$

If $\{ x_k \}$ converges to x, we write $x_k \longrightarrow x$ or limit $x_k = x$.
Limit point: A point x is a limit point of the sequence $\{ x_k \}$ if there is a subsequence of $\{ x_k \}$ convergent to x. Thus x is a limit point of $\{ x_k \}$ if there is a subset M of the positive integers such that $\{ x_k \}_{k \in M}$ is convergent to x.

3.2.5 Compact set:

A set is compact if it is both closed and bounded. An important result, due to Weierstrass, is that: If S is a compact set and $\{ x_k \}$ is a sequence, each member of which belongs to S, then $\{ x_k \}$ has a limit in S (that is, there is a subsequence converging to a point in S).

3.2.6 Weierstrass theorem:

Weierstrass theorem is a theorem connected with a continuous function which states: A continuous function f defined on a compact set S has a minimum point in S; that is, there is an $x^* \in S$ such that for all $x \in S$, $f(x) \geq f(x^*)$.

3.3 Basic definitions (Luenberger, 1984) :

3.3.1 Mapping:

A correspondence f that associates with each point in a space A a point in a space B is said to be a mapping from A to B, and and symbolized by
$f : A \longrightarrow B$.
The mapping f may be either linear or non-linear.

3.3.2 Linear independence:

A set of vectors $a_1, a_2, \ldots, a_k$, is said to be linearly dependent if there are scalars $\lambda_1, \lambda_2, \ldots, \lambda_k$ not all zero, such that $\sum_{i=1}^{k} \lambda_i a_i = 0$. If no such set of scalars exists, the vectors are said to be linearly independent.

3.3.3 Symmetric matrix and non-singular matrix:

If A is an m×n matrix, then its transpose A^t is an n×m matrix. A symmetric matrix is a square matrix with $A = A^t$.
Non-singular matrix is a square matrix whose determinant is non-zero.

3.3.4 The Hessian matrix:

If f is a continuous function and has first and second derivatives, then f is denoted $f \in C^2$. The Hessian matrix of f at x is $\nabla^2 f(x)$. It is a symmetric n×n matrix of second partial derivatives, i.e the element in the i-th row and j-th column is $\frac{\partial^2 f}{\partial x_i \partial x_j}$,

$$\nabla^2 f(x) = \left[\frac{\partial^2 f(x)}{\partial x_i \partial x_j} \right]$$

3.3.5 The matrix definiteness:

A symmetric matrix A is said to be positive definite if the quadratic form x^tAx is positive for all non zero vectors x. Similarly, A is positive semi-definite, negative definite and negative semi-definite if $x^tAx \geq 0$, < 0 or ≤ 0 respectively for all x. The matrix A is indefinite if x^tAx is positive for some x and negative for others. It is easy to obtain a connection between definiteness and the eigenvalues of A (Luenberger, 1984). The matrix A is positive definite (or positive semi-definite) if and only if all eigenvalues of A are positive (or non-negative). Fortunately, it is not necessary to calculate the eigenvalues of A to determine whether or not A is positive or negative definite. By only investigating the principal minors of the matrix (Binmore, 1983), it has been shown that:

a - The matrix A is positive definite if all its principal minors are positive.

b - The matrix A is negative definite if all its principal minors of even order are positive and all its principal minors of odd order are negative.

3.3.6 Convexity:

A non-empty set $\mathcal{C}$ in n-dimensional Euclidean space E^n is said to be convex if the line segment joining any two points of the set also belongs to the set. In other words, if x_1 and x_2 are in $\mathcal{C}$, then $\lambda x_1 + (1-\lambda)x_2$ must also belong to $\mathcal{C}$ for $\lambda \in [0,1]$. This is illustrated in figure (3.1)

Convex and concave functions: A function f defined on a convex set $\mathcal{C}$ is said to be convex if, for every $x_1, x_2 \in \mathcal{C}$ and every λ, $0 \leq \lambda \leq 1$, there holds:

$$f(\lambda x_1 + (1 - \lambda)x_2) \leq \lambda f(x_1) + (1 - \lambda)f(x_2)$$

Figure (3.2) illustrates the definition of a convex function of a single variable (Reklaitis et. al., 1983).

If, for every λ, $0 < \lambda < 1$, and $x_1 \neq x_2$, there holds:

$$f(\lambda x_1 + (1 - \lambda)x_2) < \lambda f(x_1) + (1 - \lambda)f(x_2),$$

then f is said to be strictly convex.

A function g defined on a convex set C is said to be concave if the function f = -g is convex. The function g is strictly concave if -g is strictly convex.

Test of Convexity and concavity: A function f(x) is a convex function if the Hessian matrix of f is positive definite or positive semi-definite for all values of x. If the Hessian matrix of f is negative definite or negative semi-definite for all values of x, then the function is concave (Binmore, 1983)

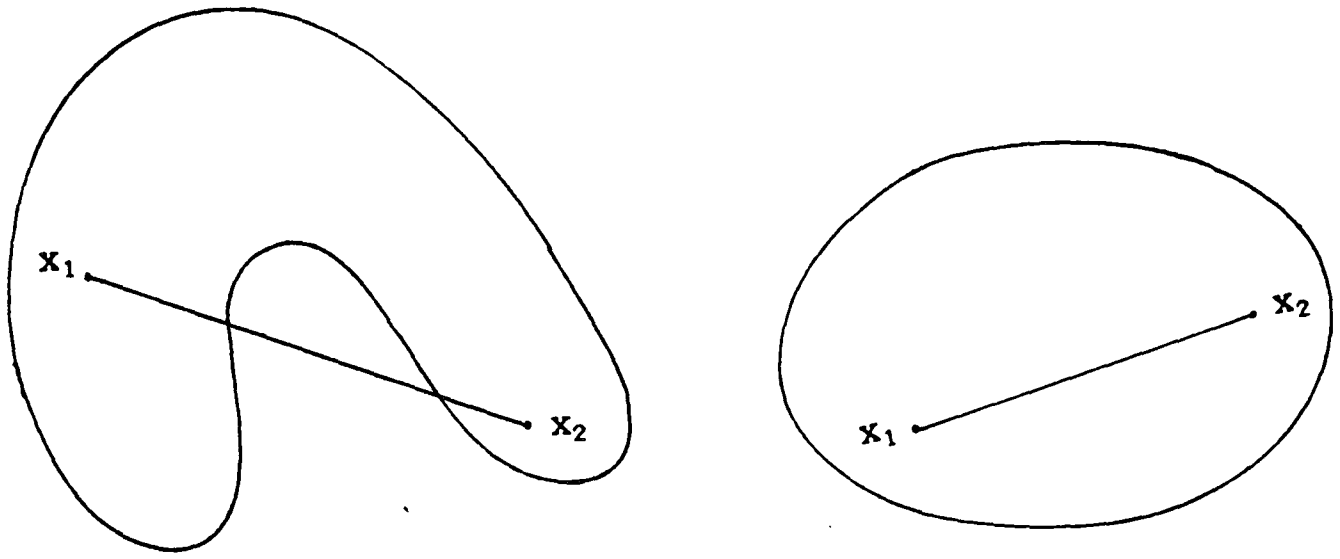


Figure (3. 1) Convex and Nonconvex Sets.

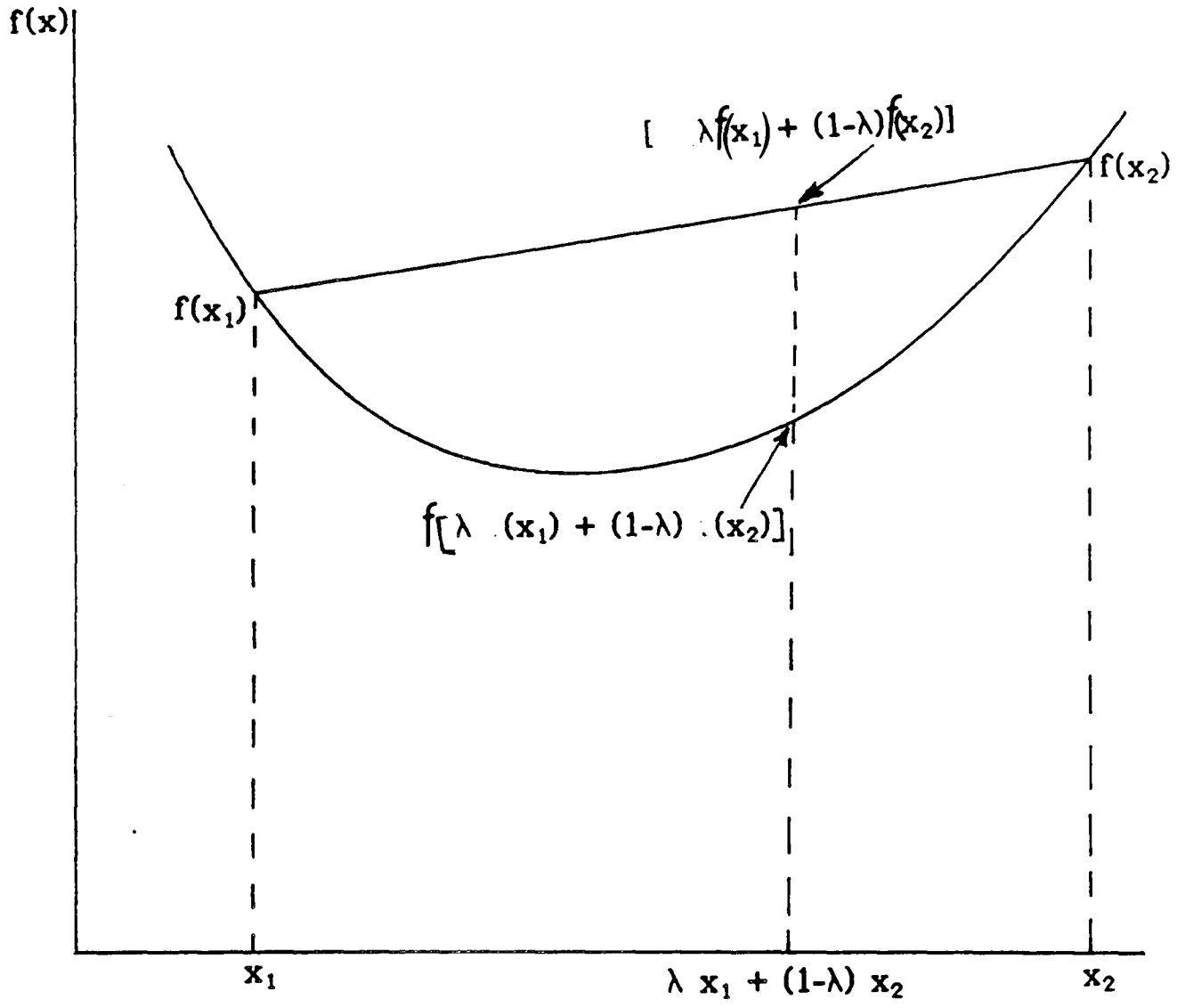


Figure (3. 2) Convex Functions.

3.3.7 Monotonic functions:

A function f(x) is monotonic (either increasing or decreasing) if for any two points x_1 and x_2, with $x_1 \leq x_2$, it follows that:

$$f(x_1) \leq f(x_2) \qquad \text{(monotonically increasing)}$$

$$f(x_1) \geq f(x_2) \qquad \text{(monotonically decreasing)}$$

Figure (3.3) shows a monotonically increasing and decreasing function.

3.4 Optimisation

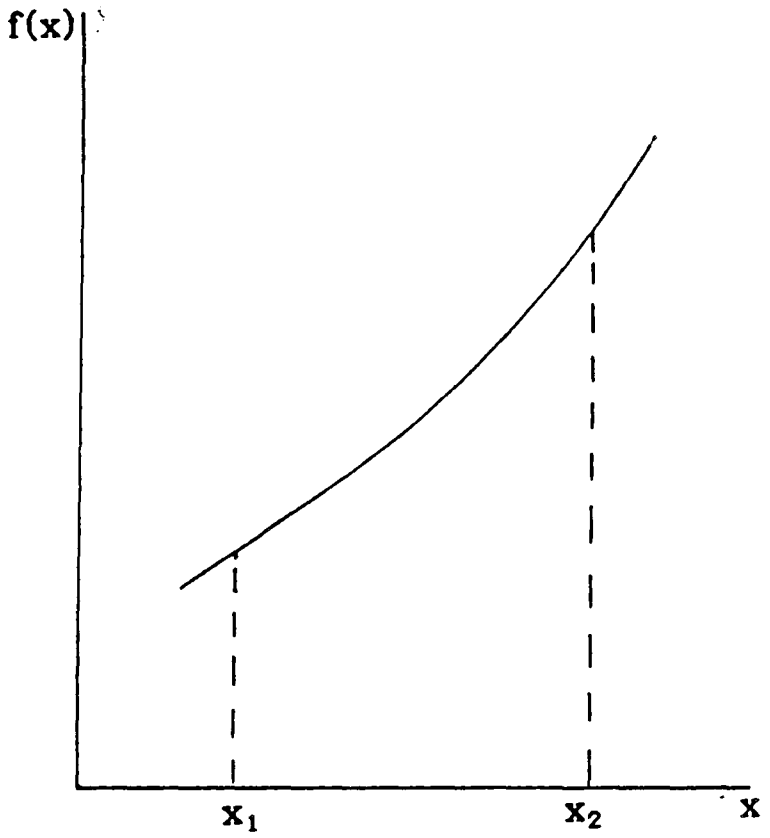
Optimisation can generally be defined as the act of obtaining the best result under given circumstances, or mathematically as the process of finding the conditions that give a maximum or minimum value of a function.

In considering optimisation problems, it is always desirable to know whether the determined solution is optimal or not. Thus, it is desirable to know the characteristics for an optimal solution. To answer this, some preliminaries are summarized in the next few subsections of this section.

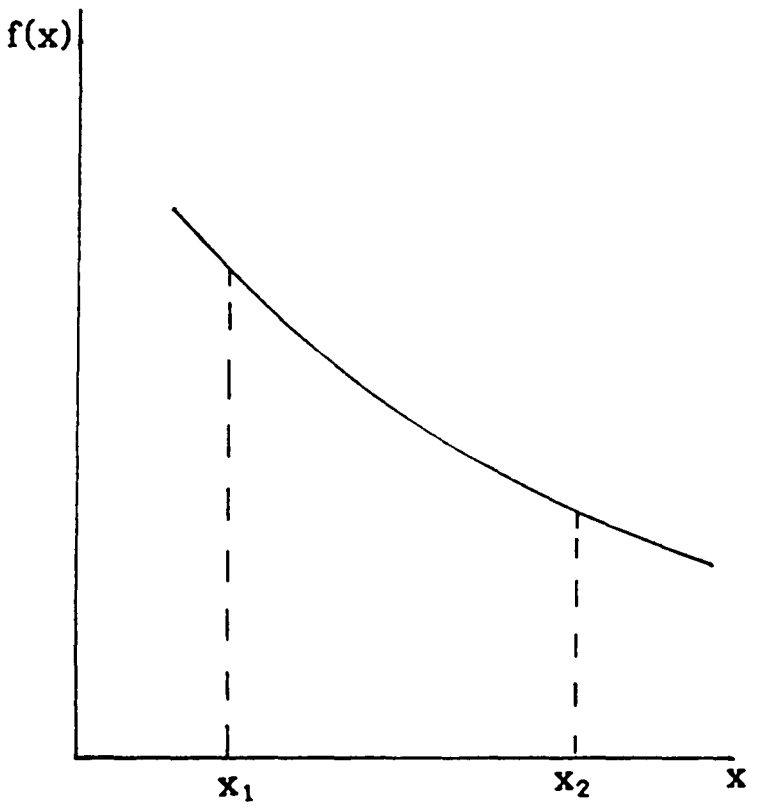
3.4.1 Stationary points:

For a function $f : \Re^n \longrightarrow \Re$, a stationary point is a point at which the rate of change of f in all directions is zero (Binmore, 1983).
For one variable case; $f : \Re \longrightarrow \Re$, a stationary point x^* for which $f'(x^*) = 0$; i.e. if f has a horizontal tangent line where $x = x^*$ (Figure 3.4).
A stationary point of well behaved functions fall into three classes; which are: local maxima, local minima or a saddle point.

In order to distinguish between these classes, we need the optimality conditions which are contained in the following theorem (Reklaitis, Ravindran and Ragsdell, 1983).



monotonically increasing function



monotonically decreasing function

Figure (3. 3) Monotonic Functions.

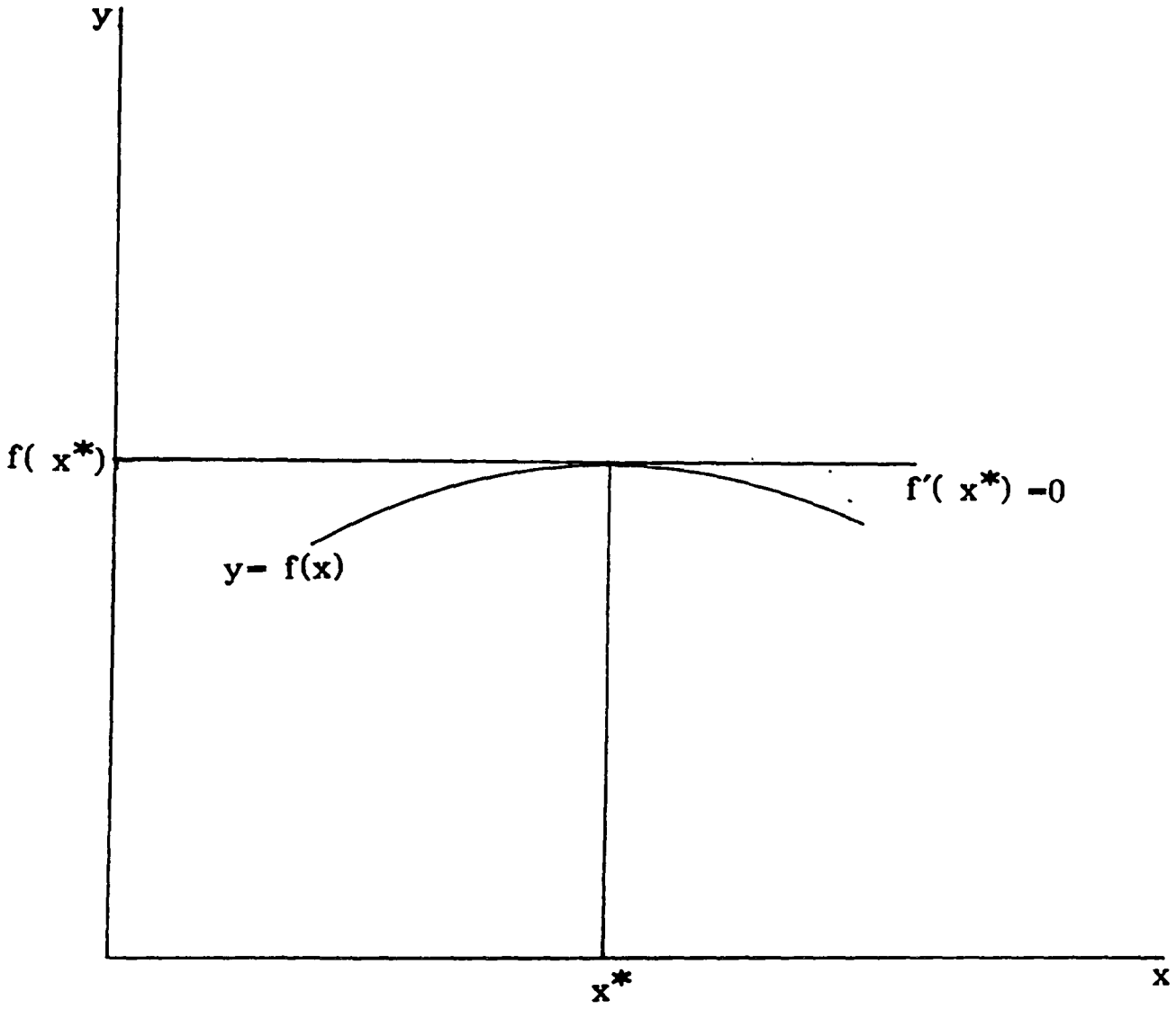


Figure (3. 4) Stationary Point.

Theorem 3.1:

Suppose at a point x^* the first derivative is zero and the first non-zero higher order derivative is denoted by n, then:

a- If n is odd, then x^* is a saddle point.

b- If n is even, then x^* is a local minimum.

Moreover,

1- If this derivative is positive, then the point is a local minimum.

2- If this derivative is negative, the point is a local maximum.

The proof of the theorem is derived directly from Taylor series expansion and is shown in Reklaitis et. al. (1983).

To generalise Theorem 3.1 for functions of several variables, the following results are summarized:

a- If $f'(x^*) = 0$ and $f''(x^*)$ is positive definite, then x^* is a local minimum.

b- If $f'(x^*) = 0$ and $f''(x^*)$ is negative definite, then x^* is a local maximum.

c- If $f'(x^*) = 0$ and determinant $f''(x^*) \neq 0$, but $f''(x^*)$ is neither positive definite nor negative definite, then x^* is a saddle point.

Theorem 3.2: Necessary conditions

For x^* to be a local minimum, it is necessary that:

$$f'(x^*) = 0$$

and

$f''(x^*)$ is positive semi-definite.

Theorem 3.3: Sufficient conditions

If:

$$f'(x^*) = 0$$

and

$f''(x^*)$ is positive definite,

then x^* is an isolated local minimum of f(x). However, if $f''(x) \geq 0 \ \forall x$, then f(x) is called a convex function and the local minimum is also a global one.

3.4.2 Structure of optimisation problems:

Most practical problems can be expressed as problems requiring the minimisation of a real valued function f(x) of an N-component vector argument $x = (x_1,..,x_n)$ whose values are restricted to satisfy a number of real valued equations $h_k(x) = 0$, a set of inequalities $g(x) \leq 0$ as:

$$\begin{aligned} &\text{minimise} && f(x) \\ &\text{subject to} && h(x) = 0 \\ & && g(x) \leq 0 \end{aligned} \qquad (3.1)$$

In solving such problem, all possible values of x that satisfy the subjected constraints are called feasible points and the feasible region is the set of all feasible solutions. The best feasible solution is called the optimal.

Optimisation problems are classified into different categories according to the properties of the objective and constraint functions (such as linear, non-linear, quadratic, etc.). The problem (3.1) is called a constrained optimisation problem. However, if there are no constraints, the problem is known as an unconstrained problem. There is a wide range of optimisation methods for all classes of optimisation problems (see for example; Fletcher, 1987; Bertsekas, 1982; Gill et. al., 1981). However, the augmented Lagrangian is one of the optimisation methods commonly used. It has been used by the Nag library Fortran routine in the simulation studies for Chapters 4, 5, 7 and 8 of this thesis and will be summarized later in this chapter.

3.4.3 Regularity condition:

It is said that the point x^* is a regular point of equation (3.1) if the gradient vectors of all active constraints at that point are linearly independent.

3.4.4 The Kuhn-Tucker conditions:

Kuhn and Tucker (1951) have developed the necessary and sufficient optimality conditions for the non-linear programming (NLP) problem. For the general NLP problem (3.1), assume that the functions f, g and h are differentiable. The Lagrangian function associated with the problem (3.1) is:

$$\mathcal{L}(x,\lambda,\mu) = f(x) + \lambda^t h(x) + \mu^t g(x) = 0 \qquad (3.2)$$

and assuming the existence of the multipliers λ and μ. Then the Kuhn-Tucker conditions are:

$$\nabla f(x) + \lambda^t \nabla h(x) + \mu^t \nabla g(x) = 0 \qquad (3.3)$$

$$g(x) \leq 0 \qquad (3.4)$$

$$h(x) = 0 \qquad (3.5)$$

$$\mu g(x) = 0 \quad , \quad \mu \geq 0 \qquad (3.6)$$

Conditions (3.3)-(3.6) give the necessary conditions of optimality.

Kuhn-Tucker necessary theorem:

Consider the NLP problem given by equation (3.1). Let f, g and h be differentiable functions and x^* be a feasible solution to NLP. Furthermore, let $g(x^*) = 0$, and $\nabla g(x^*)$ and $\nabla h(x^*)$ be linearly independent. If x^* is an optimal solution to NLP, then there exists a (μ^*, λ^*) such that (x^*, μ^*, λ^*) solves the Kuhn-Tucker (K-T) problem given by equations (3.3)-(3.6). The proof of the theorem is given in Bazaraa and Shetty(1979).

The conditions that $\nabla g(x^*)$ and $\nabla h(x^*)$ are linearly independent at the optimum is known as a constraint qualification. It essentially implies the regularity conditions on the feasible region there exists at least one feasible point x that is strictly inside the feasible region of the inequality constraints.

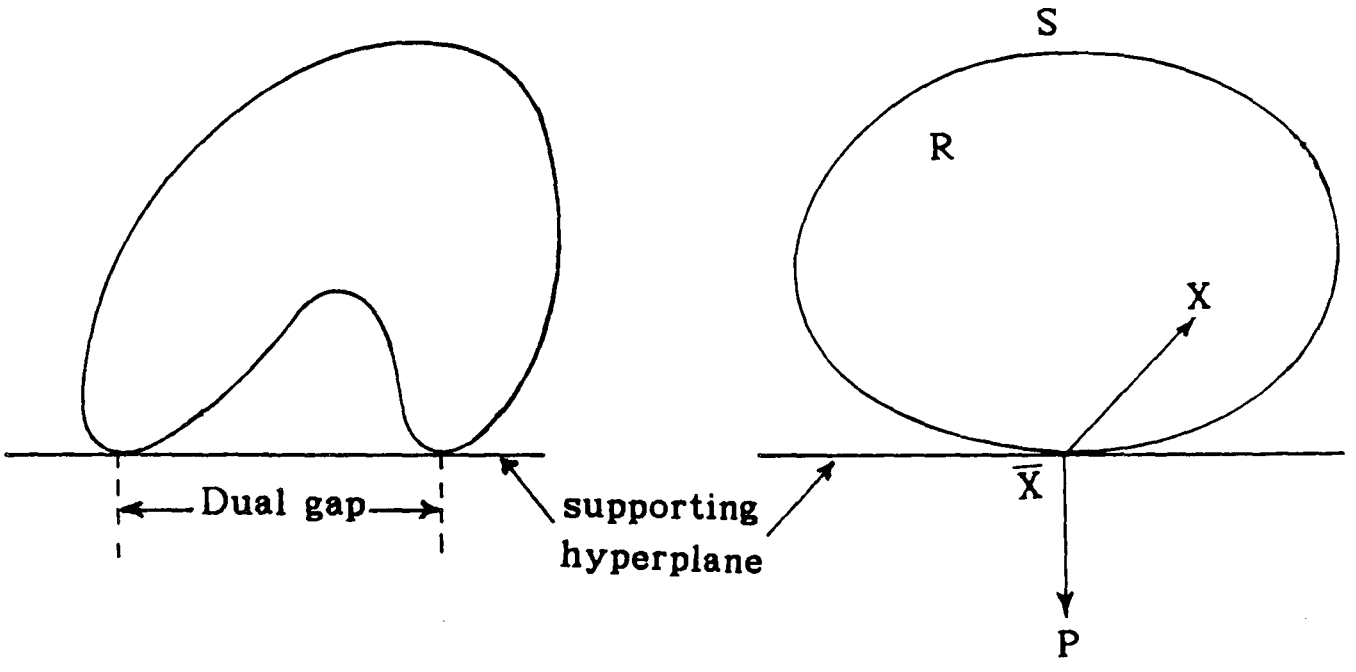
When the constraint qualification is not met at the optimum, there may not exist a solution to the K-T problem.

3.4.5 Dual gap:

For a convex set, it is known (Bazaraa and Shetty, 1979) that there exist a supporting hyperplane for every point (and therefore, for the solution point) of the set (Figure 3.5 a).

In case of a nonconvex set, the existence of supporting plane at each boundary point is not guaranteed and inaccessible regions such as that shown in figure (3.5 b) may be found. This region is known as dual gap. Such a dual gap arises if some choice of Lagrange multipliers λ produces at least two solutions (Stephanopoulos and Westerberg, 1975) to the Lagrangian problem; equation (3.1).

The normal Lagrangian methods can solve some nonconvex problems. However, if the solution point is inside a dual gap, such methods will fail to discover the solution and the methods based on the augmented Lagrangian are recommended for such problems.



b. Nonconvex Set. a. Convex Set

Figure (3. 5) Supporting hyperplane for convex and nonconvex problems.

3.4.6 The augmented Lagrangian method:

The augmented Lagrangian method is one of the most effective general classes of non-linear programming methods. It has proved (Findeisen et. al., 1980) to be more convenient, where it can be applied on convex and non-convex problems, and therefore more often used.
The augmented Lagrangian for the equality constrained problem:

$$\left.\begin{array}{ll} \text{minimise} & f(x) \\ \\ \text{s.t:} & h(x) = 0 \end{array}\right\} \qquad (3.7)$$

is the function $\mathcal{L}_a(x,\lambda,\rho)$; such that (Hestenes, 1969; Powell, 1969):

$$\mathcal{L}_a(x,\lambda,\rho) = f(x) + \lambda^t h(x) + \frac{1}{2}\rho \|h(x)\|^2 \qquad (3.8)$$

where ρ is a positive constant, and the quadratic term is to convexify the problem.

For a large class of non-convex problems, normal Lagrangian do not yields suitable saddle points which correspond to the optimal solution, but because of the convexification, the augmented Lagrangian yield saddle points for these non-convex problems if ρ is chosen appropriately.

The augmented Lagrangian function can be used sequentially to give a way of solving the optimisation problem (3.7). There exists a value of λ for which x^* minimises $\mathcal{L}_a(x, \lambda, \rho)$, and this is in fact λ^*, the Lagrange multiplier vector associated with the solution x^*. This result (Fletcher, 1987) is usually independent of ρ (provided ρ is sufficiently large). So if the minimiser of the augmented Lagrangian function (3.8) is denoted by $x(\lambda)$ and ρ is fixed during the algorithm, then the basic implementation of the approach is as follows:

a- Determine a sequence $\{\lambda^{(k)}\} \longrightarrow \lambda^*$.
b- For each $\lambda^{(k)}$, minimise the augmented Lagrangian function (3.8) to find a local minimiser $x(\lambda^{(k)})$.
c- The process is terminated when $\rho h(x(\lambda^{(k)}))$ is sufficiently small (Fletcher,

1987).

A standard method to update λ^k (Luenberger, 1984) is:

$$\lambda^{k+1} = \lambda^k + \rho h(x^k) \tag{3.9}$$

The augmented Lagrangian method can also be applied for a more general non-linear programming problem involving both equality and inequality constraints, as in problem (3.1). The augmented Lagrangian function associated with the NLP problem (3.1) is as follows (Bertsekas, 1982; Luenberger, 1984):

$$\mathcal{L}_a(x, \lambda, \mu, \rho) = f(x) + \lambda^t h(x) + \frac{1}{2}\rho\|h(x)\|^2$$

$$+\frac{1}{2\rho}\sum_{j=1}^{r}([\max\{0,\mu_j+\rho g_j(x)\}]^2-\mu_j^2) \tag{3.10}$$

Typical advantages cited in favour of the augmented Lagrangian approach are its robustness and its ease in programming, and its capability of application to a broad class of problems (Rockafellar, 1974).

3.5 Summary:

A summary of some basic definitions, notation and relations that are used in this thesis has been given in this chapter. Structure of the optimisation problems and the conditions of optimality are included. The augmented Lagrangian method is also described as an example of the non-linear optimisation method. It has been used by the NAG library routines in most of the simulation studies in the next chapters of this thesis.

Chapter 4

Algorithms for steady state large scale systems

4.1 Introduction:

This chapter is generally concerned with hierarchical control algorithms but specifically the aim is to make a comparison between the double iterative price correction mechanism DIPCM (Shao and Roberts, 1983) and the augmented interaction balance method with feedback AIBMF (Tàtjewski, 1985).

Hierarchical control algorithms are characterised by the use of local decision units and a supremal coordinating unit. Each subprocess of the industrial plant has its own optimising decision unit which computes its own set point values to maximise its local performance subject to local model equations, constraints and estimates of disturbances.

A coordinator is often required at a second level within the supervisory layer, to ensure that overall system objectives, interconnections and constraints are satisfied. Coordination is performed at a higher level unit which can intervene in local decision problems (Findeisen et. al., 1980).

There are two principal methods of coordination; firstly by the interaction prediction method, and secondly by the interaction balance method.

In this chapter, the involved algorithms (DIPCM and AIBMF) are both based on the price coordination method. Therefore, the basic version of the price method is introduced, then a brief summary for each algorithm is given.

Finally, a comparative study between the double iterative price correction mechanism and the augmented interaction balance method with global feedback is provided.

4.2 Interaction balance method:

The interaction balance method or price coordination is based on equating supply and demand, that is, equating the corresponding outputs and inputs (Findeisen et. al., 1980). In terms of the system description, the aim of the coordinator is to adjust parameters within the modification in order to provide satisfaction of the interconnection equation, i.e.

$$u = H\,y \tag{4.1}$$

where u is the inputs vector, y is the outputs vector and H is the interconnection matrix of the system.

In price methods, the interconnections between subprocess models are all removed by "cutting" the links between subsystems and introducing corresponding additional equality constraints. It is only in the obtained final solution that the interconnection relations between the subsystems (equation 4.1) are required to be fulfilled and this condition is ensured by the coordinator.

An efficient implementation of the interaction balance method is the closed loop method with global feedback since open loop methods rely on accurate mathematical models of the steady state behaviour of the industrial processes. But, in practice, model-reality differences often exist; therefore, closed loop methods are preferred in order to overcome these difficulties.

Also the interaction balance method with local feedback suffers from stability problems (Bakalis, 1986; Findeisen et. al., 1980).

In the next section, the interaction balance method with global feedback is summarized, since the double iterative price connection mechanism (Shao and Roberts, 1983) and the augmented interaction balance method with

global feedback (Tatjewski, 1985) are both based on the basic version of the interaction balance method with global feedback (Findeisen et. al., 1980).

4.3 Interaction balance method with global feedback:

The tasks of local decision units which are based on mathematical models are to find the set points c_i and the model interconnection inputs u_i. The solutions of the local decision units c_i and u_i are transmitted respectively to the local real subprocesses and the coordinator.

The coordinator determines the modifiers λ such that $\hat{u}(\lambda) = u^*(\lambda)$, where u^* are the real process interconnections ($u^* = H\ y^*$) which are produced as a result of applying the set points c_i to the real process and are measured and transmitted to the coordinator.

The task of each local decision units is:
For a given value of coordinator variable λ find $c_i(\lambda)$ and $u_i(\lambda)$ such that:

$$\left.\begin{aligned} \min_{c_i,u_i}\ & Q_i(c_i\ ,\ u_i) + \lambda^t_i\ u_i - \sum_{j=1}^{N} (\lambda^t_i\ H_{ji}\ y_i) \\ \text{subject to:}\ & y_i = F_i(c_i\ ,\ u_i) \\ & g_i(c_i\ ,\ u_i) \le 0 \end{aligned}\right\} LP_i \qquad (4.2)$$

The task of the coordinator is:
For a given value of $\hat{c}_i$, $\hat{u}_i$, find λ such that:

$$\min_{\lambda} \sum_{i=1}^{N} \left((u^*_i - HF(\hat{c}_i\ ,\ \hat{u}_i) \right)^2 \qquad (4.3)$$

The structure of the interaction method with global feedback is shown in figure (4.1).

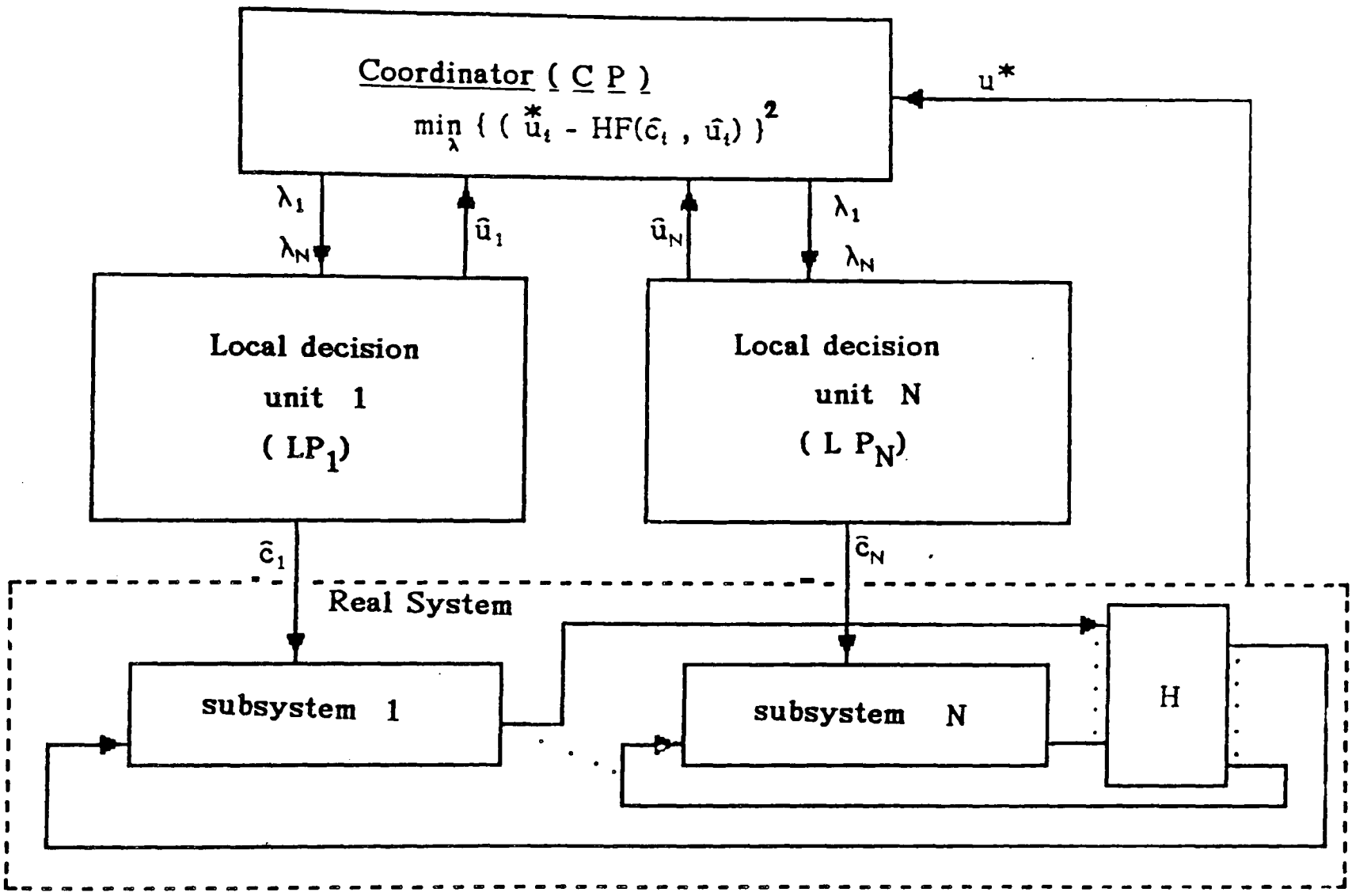


Figure (4. 1) Closed loop Price method with global feedback.

One of the biggest disadvantages of the interaction balance method with feedback (IBMF) is the large number of set point changes which may be required to achieve a converged solution. Therefore, the real system may need to be disturbed many times, requiring a large amount of on-line computing time.

Shao and Roberts (1983) have proposed another approach which modifies the IBMF method to reduce the number of set points changes required and hence, less disturbance of the real system is caused . It is summarized in the following section.

4.4 Double iterative price correction mechanism:

The proposed double iterative price correction mechanism (DIPCM) with global feedback by Shao and Roberts (1983) was designed to reduce the number of times that information is required from the real system. The idea of the basic version of the IBMF method is for every iteration, that is for every change of λ, it was necessary to apply the control set, obtained by the local decision units, to the real system and wait for a new steady state in order to obtain the required information. In the double iterative price correction mechanism, the coordinator's task has been modified so that the coordinator has to find two variables, s and $\lambda(s)$, where $\lambda(s)$ is the price vector and s is the shift vector. The two variables are adjusted by the coordinator using a double iterative process, where the price vector, λ, is adjusted in the inner loop using model information only.

The shift vector s, is adjusted in the outer loop where here only real system information is required.

Two versions were introduced by Shao and Roberts, 1983, which are based on shifting techniques and will be described in the following subsection.

4.4.1 By shifting the interaction balance condition (Version 1):

This algorithm is based on shifting the interaction balance condition of the overall problem using an auxiliary parameter vector s, where $s \in \mathcal{U}$ and s is provided by the coordinator. It is expected that at convergence, at some value of $\hat{s}$, the model interaction inputs $\hat{u}$ will be equal to the real interaction inputs $u^* = HK^*(\hat{c})$, within a desired tolerance. So, the interaction balance condition of the overall problem is shifted by s as:

$$u = H\,F(c,u) + s \tag{4.4}$$

Then, applying the normal Lagrangian technique to the problem, we obtain:

$$\mathcal{L}(c,u,y,\lambda) = Q(c,u,y) + \langle \lambda\ ,\ u\text{-}HF(c,u)\text{-}s \rangle$$

where Q is the performance index of the problem, $\lambda \in \Lambda \subset \mathcal{U}$.

The task of local decision units and coordinator are:

For a given value of λ, find $\hat{c}_i(\lambda)$, $\hat{u}_i(\lambda)$, such that:

$$LP_i\begin{cases} (\hat{c}_i(\lambda)\ ,\ \hat{u}_i(\lambda)) = \arg\min \mathcal{L}_i(c_i,u_i,\lambda) \\ \text{where: } \mathcal{L}_i(c_i,u_i,\lambda) = Q_i(c_i,u_i,F_i(c_i,u_i)) + \langle \lambda_i,u_i \rangle - \sum_{j=1}^{N} \langle \lambda_{ji},\ H_{ji}F_i(c_i,u_i) \rangle \\ \text{s.t: } g_i(c_i\ ,\ u_i) \geq 0 \end{cases} \tag{4.5}$$

The task of the coordinator is:

Find $\hat{\lambda}(\hat{s})$ and $\hat{s}$, such that:

$$CP\begin{cases} \hat{u}(\hat{\lambda}(\hat{s})) - HF(c(\hat{\lambda}(\hat{s}))\ ,\ \hat{u}(\hat{\lambda}(\hat{s}))) - \hat{s} = 0 \\ \\ \hat{u}(\hat{\lambda}(\hat{s})) - HK^*(\hat{c}(\hat{\lambda}(\hat{s}))) = 0 \end{cases} \tag{4.6}$$

This task is performed using a double iterative mechanism. The implementation aspects of the algorithm will summarize the process, and the structure of version 1 is shown in figure (4.2)

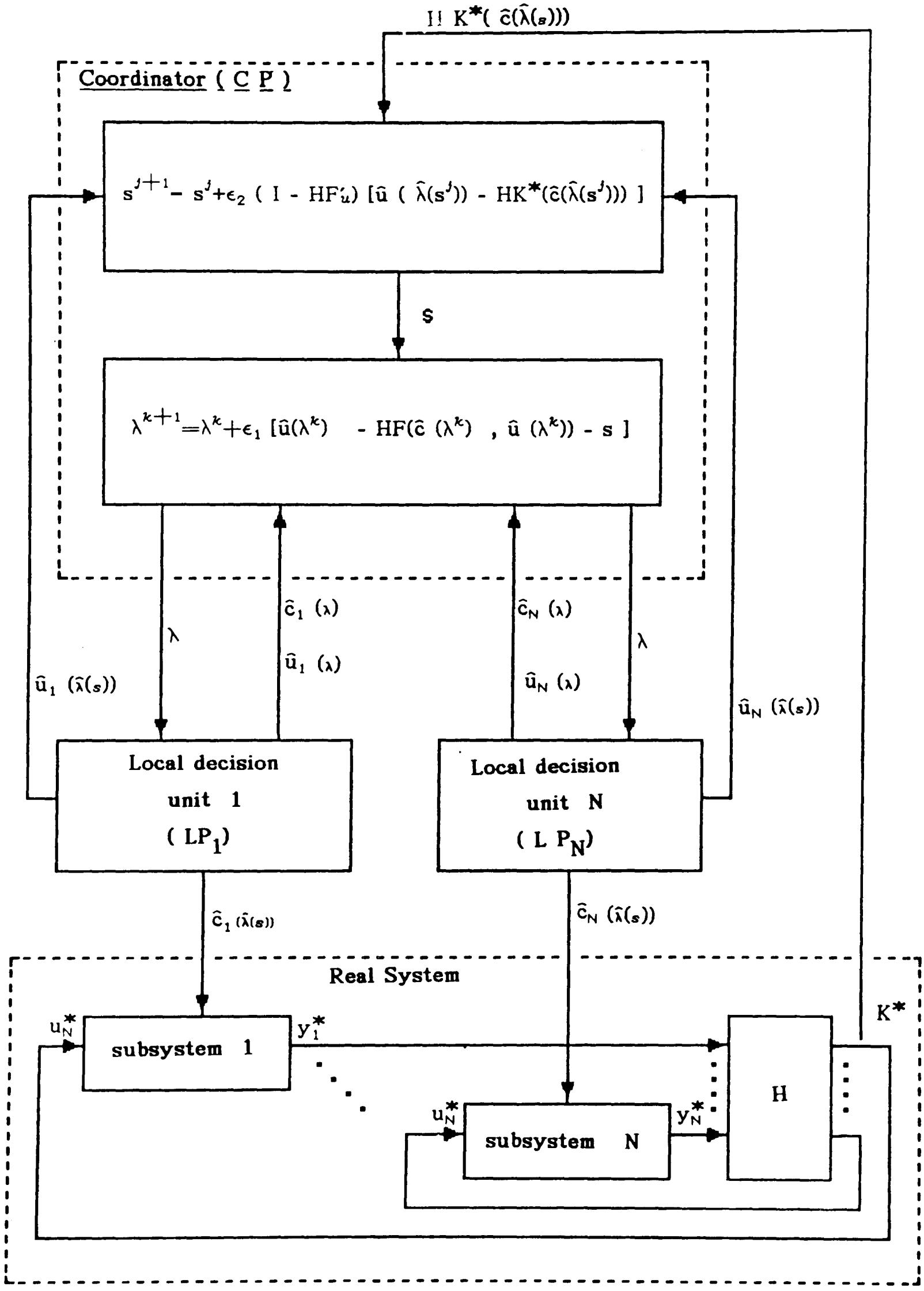


Figure (4. 2) Information Structure of Version 1 Algorithm.

4.4.2 Implementation aspects of version 1:

1 - With a given fixed values of s (initially usually zero) and starting points of λ^0, c_i^0 and u_i^0, a solution of the local decision units problems, LP_i, $(\hat{c}_i(\lambda^k), \hat{u}_i(\lambda^k))$ are obtained. It is worth mentioning here that the starting point of the algorithm $(\lambda^0, c_i^0, u_i^0)$ can not be any value but it has to be the solution of the open loop model based optimisation problem (Findeisen et. al., 1980).

2 - The price vector λ is adjusted in an interior iterative procedure on the basis of imbalance using the LP_i solutions as follows:

$$\lambda^{k+1} = \lambda^k + \epsilon_1[\hat{u}(\lambda^k) - HF(\hat{c}(\lambda^k), \hat{u}(\lambda^k)) - s] \tag{4.7}$$

where ϵ_1 is a step length coefficient, $0 \leq \epsilon_1 \leq 1$.

3 - The procedure is continued between the coordinator, to update λ according to equation (4.7), and the local decision units to obtain solutions of their problems based purely on a model basis until the shifted balance condition (4.8) is achieved within a desired tolerance.

$$\hat{u}(\lambda^k) - HF(\hat{c}(\lambda^k), \hat{u}(\lambda^k)) - s = 0 \tag{4.8}$$

4 - The control set $\hat{c}(\hat{\lambda}(\hat{s}))$ is applied, as controller set points, to the real system. After the system reaches steady state conditions then the real steady state interconnection inputs $u^* = HK^*(\hat{c}(\hat{\lambda}(\hat{s})))$ are measured and transmitted to the coordinator.

5 - The coordinator uses this feedback information to update the values of the shift vector s within an outer iterative procedure according to the equation (Shao and Roberts, 1983):

$$s^{j+1} = s^j - \epsilon_2[I - HF'_u] (\hat{u}(s^j) - HK^*(\hat{c}(s^j))) \tag{4.9}$$

6 - The new values of s are used to modify the values of λ and then the coordinator will send these updated values of λ to the local decision units which solve their problems again and then the iterative procedure restarts from (2) until the coordinator achieves the following condition in the outer loop:

$$\hat{u}(\hat{\lambda}(\hat{s})) - HK^{*}(\hat{c}(\hat{\lambda}(\hat{s}))) = 0 \qquad (4.10)$$

When condition (4.10) is achieved within a desired tolerance, the overall process is terminated.

4.4.3 By shifting the model outputs (Version 2):

This algorithm is based on application of the normal Lagrangian as in version 1 but with the vector $s \in \mathcal{Y}$ to shift the model outputs from

$y = F(c,u)$ to:

$$y^{s} = F(c,u) + s \qquad (4.11)$$

Hence, the interaction input vector u is accordingly shifted from $u = Hy$ to:

$$u^{s} = u + Hs \qquad (4.12)$$

The performance index and the constraints are modified according to equations (4.11) and (4.12) as follows:

$$Q = \sum_{i=1}^{N} (Q_i(c_i, u_i^{s}, y_i^{s}) = \sum_{i=1}^{N} Q_i(c_i, u_i+H_is, y_i+s_i) \qquad (4.13)$$

$$g_i(c_i, u_i+H_is, y_i+s_i) \geq 0 \qquad (4.14)$$

In version 2, it is expected at some value of s that the real outputs y^{*} will match the shifted outputs of equation (4.11).
The tasks of the local decision units (LP_i) and the coordinator (CP) respectively are:

For fixed values of λ and s specified by the coordinator, find $\hat{c}_i(\lambda,s)$ and $\hat{u}_i(\lambda,s)$ such that:

$(\hat{c}_i(\lambda,s)\ ,\ \hat{u}_i(\lambda,s)) = \arg\min \mathcal{L}_i(c_i,u_i,\lambda)$, where:

$$LP_i\Big\{ \mathcal{L}_i(c_i,u_i,\lambda) = \sum_{i=1}^{N} Q_i(c_i,\ u_i+H_i s,\ F_i(c_i,u_i)+s_i) + \langle\lambda_i,u_i\rangle$$

$$- \sum_{j=1}^{N} \langle\lambda_{ji}\ ,\ H_{ji}\ F_i(c_i,u_i)\rangle \qquad (4.15)$$

subject to: $g_i(c_i,\ u_i+H_i s,\ F_i(c_i,u_i)+s_i) \geq 0$

The coordinator problem is:

$$CP\Big\{ \begin{array}{l} \text{Find } \hat{s} \text{ and } \hat{\lambda}(\hat{s}), \text{ such that} \\ K^*(\hat{c}(\hat{\lambda}(\hat{s}))) - F(\hat{c}(\hat{\lambda}(\hat{s})),\ \hat{u}(\hat{\lambda}(\hat{s}))) - \hat{s} = 0 \qquad (4.16) \\ \hat{u}(\hat{\lambda}(\hat{s})) - HF(\hat{c}(\hat{\lambda}(\hat{s})),\ \hat{u}(\hat{\lambda}(\hat{s}))) = 0 \qquad (4.17) \end{array}$$

As in version 1, the coordinator task is performed by a double iterative process, where the inner loop is involved with obtaining λ using model information only while, in the outer loop, information from the real process is only required to be used in updating s. The structure of version 2 is shown in figure (4.3).

4.4.4 Implementation aspects of version 2:

1 - The first step here is exactly as in the case of version 1.

2 - The price vector λ is updated within the inner iterative loop according (Shao and Roberts, 1983) to equation (4.18):

$$\lambda^{k+1} = \lambda^k + \epsilon_1[\hat{u}(\lambda^k,s) - HF(\hat{c}(\lambda^k,s),\ \hat{u}(\lambda^k,s))] \qquad (4.18)$$

where ϵ_1 is the step length and $0 \leq \epsilon_1 \leq 1$.

The updated values of λ are applied to the local decision units to solve their problems obtaining solutions of $(\hat{c},\ \hat{u})$ which in turn will be used to adjust the values of λ. This process is continued until condition (4.17) is satisfied within a desired tolerance.

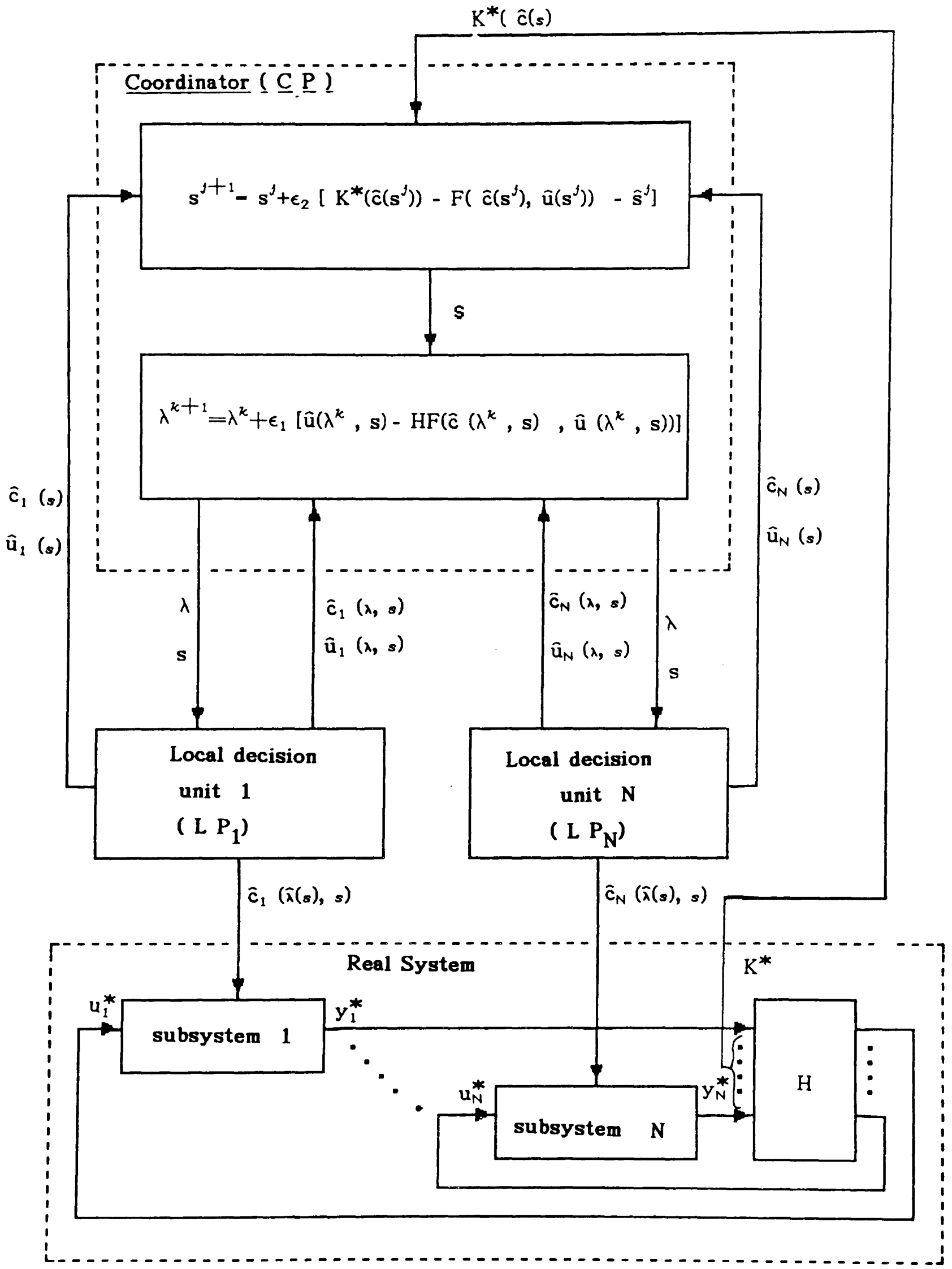


Figure (4. 3) Information Structure of Version 2 Algorithm.

3 - The local decision units solutions $\hat{c}(\hat{\lambda}(s))$ are applied to the real system as controller set points. When the real system reaches the steady state condition the real outputs y^* are measured and transmitted to the coordinator.

4 - The measured outputs are used by the coordinator to obtain a new value of the shift vector s according to equation (4.19):

$$s^{j+1} = s^j + \epsilon_2[K^*(\hat{c}(s^j)) - F(\hat{c}(s^j), \hat{u}(s^j)) - s^j] \qquad (4.19)$$

5 - The new values of s are sent to the local decision units to calculate $(\hat{c})$ and $(\hat{u})$ which in turn will be used to calculate new values of λ. Then, the process restarted from step (2) and continued until (4.16) is achieved within a desired tolerance, whence, the iterative procedure is terminated.

4.5 The augmented interaction balance method with feedback (AIBMF)

The augmented interaction balance method with feedback was developed by Tatjewski (1985) to extend the applicability of the interaction balance method with feedback (Findeisen et. al., 1980) to cover the case of problems with duality gaps. This algorithm is based on the application of the augmented Lagrangian; where an augmented term is introduced to convexify non-convex problems. The structure of the algorithm is iterative with double loops. The inner loop involves model based information only while feedback information from the real system is only required in the outer loop to update the price vector λ.

The algorithm of AIBMF is based on the augmented Lagrangian which is distinguished from the normal Lagrangian by adding an extra term as:

$$\mathcal{L}_a(c,u,\lambda,\rho) \triangleq Q(c,u) + \lambda^t (u - HF(c,u)) + 0.5\rho\|u - HF(c,u)\|^2 \qquad (4.20)$$

Equation (4.20) can be expressed in terms of individual subsystems as follows:

$$\mathcal{L}_a(c,u,\lambda,\rho) \triangleq \sum_{i=1}^{N} Q_i(c_i,\ u_i) + \lambda_i{}^t u_i - \sum_{j=1}^{N} \lambda_j{}^t\ H_{ji}\ F_i(c_i,u_i) +$$

$$+ 0.5\rho\ (\|u_i\|^2 + \|F_i(c_i,u_i)\|^2) - \rho u^t\ HF(c,u) \qquad (4.21)$$

The last term in equation (4.21) is not separable. A suitable linearization (Stephanopoulos and Westerberg, 1975) around some point $(u^k,\ HF(c^k,\ u^k))$ can be used to overcome this problem as in equation (4.22):

$$u^t HF(c,u) \cong -u^{kt} HF(c^k,\ u^k) + u^t HF(c^k,\ u^k) + u^{kt} HF(c,u) \qquad (4.22)$$

Using this result in equation (4.21), we obtain:

$$\Lambda_a(c,u,\rho,c^k,u^k) = \sum_{i=1}^{N} Q_i(c_i,\ u_i) + \lambda_i{}^t u_i - \sum_{j=1}^{N} \lambda_j{}^t H_{ji} F_i(c_i,u_i)$$

$$+ 0.5\rho\ [\|u_i\|^2 + \|F_i(c_i,u_i)\|^2 - 2u_i{}^t\ H_i F(c^k,u^k)$$

$$- 2\ \sum_{j=1}^{N} \lambda_j{}^{kt}\ H_{ji} F_i(c_i,u_i))] + \rho u^{kt}\ HF(c^k,u^k)$$

$$\triangleq \sum_{i=1}^{N} \Lambda_{ai}(c_i,u_i,\lambda,\rho,c^k,u^k) + \rho u^{kt}\ HF(c^k,u^k) \qquad (4.23)$$

In this approximate separable form; $\Lambda_{ai}(c_i,u_i,\lambda,\rho,c^k,u^k)$ can be used in formulating the optimisation problems which are solved in the local decision units.

Implementation aspects of AIBMF:

1 - For a given value of λ, solve the local decision unit tasks as expressed as minimisation problems of equation (4.23). The starting points of the algorithm for the first iteration $(\lambda^0,\ c_i{}^0,\ u_i{}^0)$ are those obtained from solving the open loop model based optimisation problem.

2 - Compare the obtained solutions $(\hat{c}_i(\lambda),\ \hat{u}_i(\lambda))$ with $(c^k,\ u^k)$:

$$(\hat{c}_i(\lambda),\ \hat{u}_i(\lambda)) - (c^k,\ u^k) = 0 \qquad (4.24)$$

If equation (4.24) is achieved within a desired tolerance, then go to step 3 otherwise reset:

$$\left.\begin{aligned} c^{k+1} &= \hat{c} \\ u^{k+1} &= \hat{u} \end{aligned}\right\} \qquad (4.25)$$

for subsequent solving of local decision unit problems until equation (4.24) is achieved. This is the inner loop procedure of the algorithm.

3 - The controls $\hat{c}_i(\lambda)$ are applied to the real system as controller set points. When the system reaches a steady state condition the interactions $u^*(\lambda)$ are measured and transmitted as feedback information to the coordinator.

4 - The coordinator uses the feedback information to improve the values of λ according to (Tatjewski, 1985) equation (4.26) in an outer iterative loop:

$$\lambda^{j+1} = \lambda^j + \alpha_i\rho[\hat{u}(\lambda^j) - u^*(\lambda^j) - HF(\hat{c}(\lambda^j), \hat{u}(\lambda^j)) + HF(\hat{c}(\lambda^j), u^*(\lambda^j))] \qquad (4.26)$$

where $\alpha_i \in (0,2)$ and ρ is a penalty coefficient.

5 - The coordinator sends the updated values of λ to the local decision units where they can be used in calculating new values of $\hat{c}(\lambda)$, $\hat{u}(\lambda)$.

6 - The obtained values of interaction inputs $\hat{u}(\lambda)$ are compared with the real measurements $u^*(\lambda)$. If:

$$\hat{u}(\lambda) - u^*(\lambda) = 0 \qquad (4.27)$$

is fulfilled within a desired tolerance the overall process is terminated otherwise go back to step (1) and continue until condition (4.27) is achieved.

The structure of the augmented interaction balance method with feedback is shown in figure (4.4).

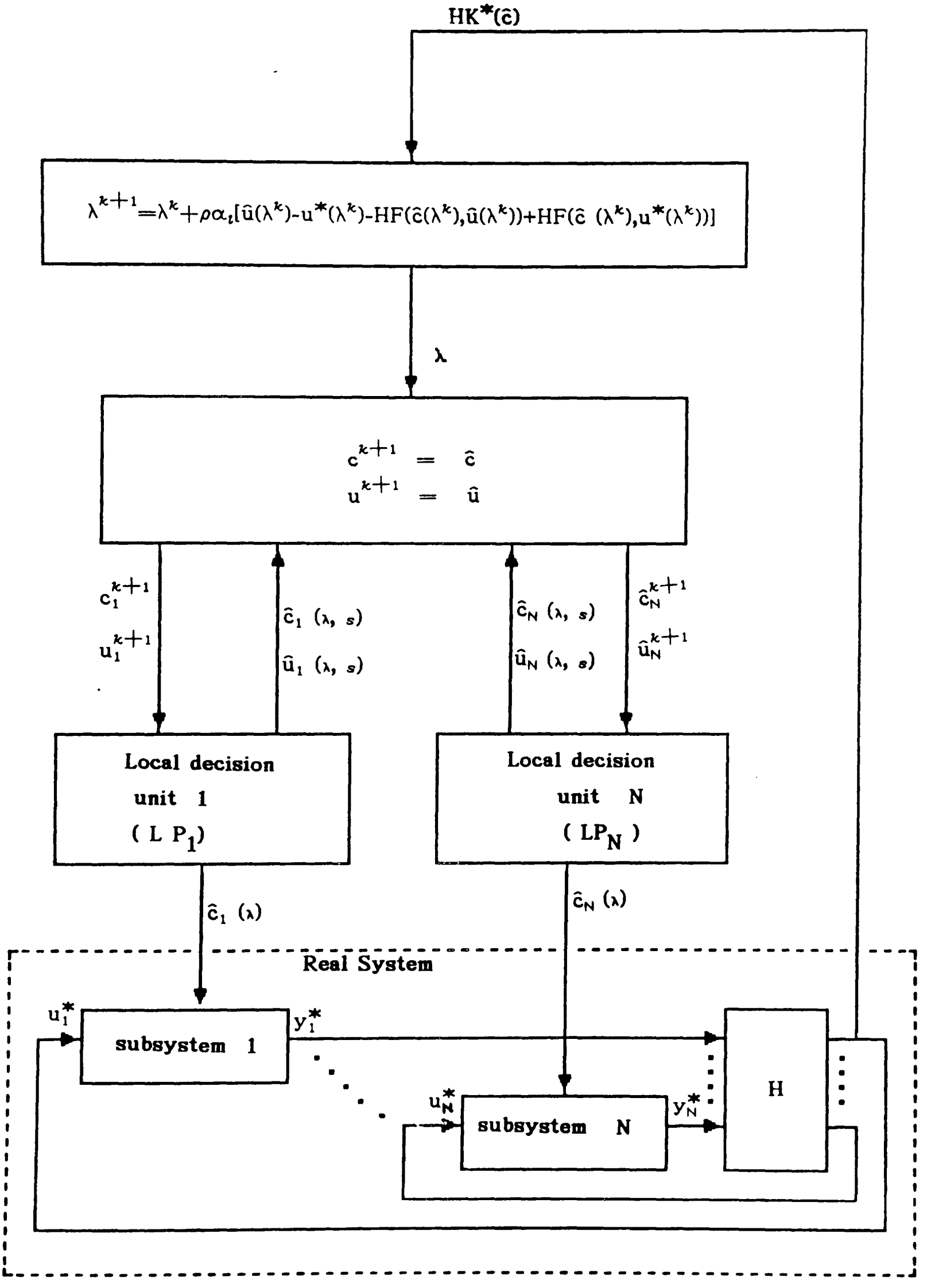


Figure (4.4) Information Structure of the Augmented IBMF.

4.6 Simulation study:

Simulation examples:

Example 1

For the first example consider a simple system composed of two interconnected subsystems, with interconnection constraints:

$$u_1 = y_2 \qquad \text{and}$$

$$u_2 = y_1$$

i.e. with interconnection matrix H of the form:

$$H = \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix}$$

Each subsystem has one control variables c_i, input u_i and output y_i, i.e.

$$c_i \in \mathcal{R}^1 \quad , \quad u_i \in \mathcal{R}^1 \quad , \quad y_i \in \mathcal{R}^1 \quad \text{and } i = 1,2.$$

The subsystem's real and model equations are:

$$F_1^*(c_1, u_1) = 2.1c_1 + u_1$$

$$F_2^*(c_2, u_2) = 0.6c_2 + 0.55u_2$$

$$F_1(c_1, u_1) = 2c_1 + u_1$$

$$F_2(c_2, u_2) = 0.5c_2 + 0.5u_2$$

The performance indices are:

$$Q_1(c_1, u_1) = 32c_1^2 - 16c_1 + (2c_1 + u_1 - 1)^2$$

$$Q_2(c_2, u_2) = 10c_2^2 + 4c_2u_2 + 0.5\ (2c_2 + 2u_2 - 4)$$

The local constraints are:

$$2c_1 + u_1 \leq 1$$

Example 2

This example contains two subsystems where the performance indices and the constraints involve system outputs.
The reality and model equations are:

$$y_{11}^* = F_{11}^*(c_1, u_1) = 1.4c_{11} - 0.6c_{12} + 1.8u_{11}$$

$$y_{21}^* = F_{21}^*(c_2, u_2) = 1.3c_{21} - 1.1c_{22} + 1.1u_{21}$$

$$y_{22}^* = F_{22}^*(c_2, u_2) = 2.3c_{22} - 0.7c_{23} - 1.1u_{21}$$

$$y_{11} = F_{11}(c_1, u_1) = c_{11} - c_{12} + 2u_{11}$$

$$y_{21} = F_{21}(c_2, u_2) = c_{21} - c_{22} + u_{21}$$

$$y_{22} = F_{22}(c_2, u_2) = 2c_{22} - c_{23} - u_{31}$$

The performance indices are:

$$Q_1(c_1, u_1, y_1) = (y_{11} - 1)^2 + c_{11}^2 + c_{21}^2$$

$$Q_2(c_2, u_2, y_2) = 2\ (y_{21} - 2)^2 + (y_{22} - 3)^2 + c_{21}^2 + c_{22}^2 + c_{23}^2$$

The interconnection equation is:

$$\begin{bmatrix} u_{11} \\ u_{21} \end{bmatrix} = \begin{bmatrix} 0 & 1 & 0 \\ 1 & 0 & 0 \end{bmatrix} \begin{bmatrix} y_{11} \\ y_{21} \\ y_{22} \end{bmatrix}$$

and the system constraints are

$$\mathcal{CUY}_1 \triangleq \{(c_1,u_1,y_1) \in \mathcal{R}^4,\ |c_{11}| \leq 1,\ |c_{12}| \leq 1,\ y_{11} \geq 0 \text{ and}$$

$$(0.8 - c_{12} - 0.6u_{11}) \geq 0\}$$

$$\mathcal{CUY}_2 \triangleq \{(c_2,u_2,y_2) \in \mathcal{R}^6,\ |c_{21}| \leq 1,\ |c_{22}| \leq 1,\ |c_{23}| \leq 1,\ y_{21} \geq 0,\ y_{22} \geq 0$$

$$\text{and } (2.04 + 1.05u_{21}^2 - c_{21}^2 - c_{22}^2 - c_{23}^2) \geq 0\}$$

where:

$c_1 = (c_{11}, c_{12})$ $\quad u_1 = u_{11}$ $\quad y_1 = y_{11}$

$c_2 = (c_{21}, c_{22}, c_{23})$ $\quad u_2 = u_{21}$ $\quad y_2 = (y_{21}, y_{22})$

Example 3

The third example involves three interconnected subsystems, where the system constraints do not involve the outputs but the system equations contain non-linear terms.

The subsystems real and model equations are:

$$y_{11}^* = F_{11}^*(c_1, u_1) = 1.3c_{11} - c_{12} + 2u_{11} + 0.15u_{11}c_{11}$$

$$y_{21}^* = F_{21}^*(c_2, u_2) = c_{21} - c_{22} + 1.2u_{21} - 3u_{22} + 0.1c_{22}^2$$

$$y_{22}^* = F_{22}^*(c_2, u_2) = 2\ c_{22} - 1.25c_{23} - u_{21} + u_{22} + 0.25c_{22}c_{23} + 0.1$$

$$y_{31}^* = F_{31}^*(c_3, u_3) = 0.8c_{31} + 2.5c_{32} - 4.2u_{31}$$

$$y_{11} = F_{11}(c_1, u_1) = c_{11} - c_{12} + 2u_{11}$$

$$y_{21} = F_{21}(c_2, u_2) = c_{21} - c_{22} + u_{21} - 3u_{22}$$

$$y_{22} = F_{22}(c_2, u_2) = 2c_{22} - c_{23} - u_{21} + u_{22}$$

$$y_{31} = F_{31}(c_3, u_3) = c_{31} + 2.5c_{32} - 4u_{31}$$

where:

$$c_1 = (c_{11}, c_{12}) \qquad u_1 = u_{11} \qquad y_1 = y_{11}$$

$$c_2 = (c_{21}, c_{22}, c_{23}) \qquad u_2 = (u_{21}, u_{22}) \qquad y_2 = (y_{21}, y_{22})$$

$$c_3 = (c_{31}, c_{32}) \qquad u_3 = u_{31} \qquad y_3 = y_{31}$$

The subsystem performance indices are:

$$Q_1(c_1, u_1) = (u_{11} - 1)^4 + 5\,(c_{11} + c_{12} - 2)^2$$

$$Q_2(c_2, u_2) = 2\,(c_{21} - 2)^2 + c_{22}^2 + 3\,c_{23} + 4u_{21}^2 + u_{22}^2$$

$$Q_3(c_3, u_3) = (c_{31} + 1)^2 + (u_{31} - 1)^2 + 2.5c_{32}^2$$

The system constraints are:

$$\mathcal{CU}_1 \triangleq \{(c_1,u_1) \in \mathcal{R}^3,\ c_{11}^2 + c_{12}^2 \leq 1,\ 0 \leq u_{11} \leq 0.5\}$$

$$\mathcal{CU}_2 \triangleq \{(c_2,u_2) \in \mathcal{R}^5,\ 0.5c_{21} + c_{22} + 2c_{23} \leq 1,$$

$$4c_{21}^2 + 2c_{21}u_{21} + 0.4u_{21} + c_{21}c_{23} + 0.5c_{23}^2 + u_{21} \leq 4\}$$

$$\mathcal{CU}_3 \triangleq \{(c_3,u_3) \in \mathcal{R}^3,\ c_{31} + u_{31} + 0.5 \geq 0,\ \ 0 \leq c_{32} \leq 1\}$$

Finally, the coupling equation is:

$$\begin{bmatrix} u_{11} \\ u_{21} \\ u_{22} \\ u_{31} \end{bmatrix} = \begin{bmatrix} 0 & 1 & 0 & 0 \\ 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} y_{11} \\ y_{21} \\ y_{22} \\ y_{31} \end{bmatrix}$$

Example 4:

This example consists of two subsystems with interconnection constraints $u_1 = y_2$, $u_2 = y_1$, i.e., with interconnection matrix:

$$H = \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix}$$

Each subsystem has one control c_i, input u_i and output y_i, i.e.

$$c_i \in \mathcal{R}^1 \quad , \quad u_i \in \mathcal{R}^1 \quad , \quad y_i \in \mathcal{R}^1 \quad , \quad i = 1,2.$$

The subsystem output mappings are:

$$F_1^*(c_1, u_1) = 2.1c_1 + u_1$$

$$F_2^*(c_2, u_2) = 0.6\ c_2 + 0.55u_2$$

and their models are:

$$F_1(c_1, u_1) = 2c_1 + u_1$$

$$F_2(c_2, u_2) = 0.5c_2 + 0.5u_2$$

The performance indices are:

$$Q_1(c_1, u_1, y_1) = 32c_1^2 - 16c_1 + (y_1 - 1)^2$$

$$Q_2(c_2, u_2, y_2) = 10c_2^2 + 4c_2u_2 - 8y_2^2$$

The local constraint sets are:

$$\mathcal{CU}_1 = \{c_1, u_1 : 2c_1 + u_1 \leq 2.25\}$$

$$\mathcal{CU}_2 = \mathcal{R}^2$$

4.7 A comparison between algorithms

In this section, a comparative study between the double iterative price correction mechanisms (Shao and Roberts, 1983) and the interaction balance method with feedback (Tatjewski, 1985) is discussed. In comparing the algorithms, some important factors must be considered. The study presented in this section includes the following factors:

1 - Generality, reliability, and computational effort.
2 - Study of non-linearity.
3 - Sensitivity to parameters.
4 - Convergence and precision.

4.7.1 Generality, reliability and computational effort:

The generality and reliability of an algorithm refers to the variety of problems that the algorithm can solve with reasonable accuracy and the restrictiveness of the assumptions required by the algorithm.

The chosen simulation examples were selected to cover a wide range of various classes and sizes of problems. The first example is taken from the work by Tatjewski, 1985. It is a simple example, contains two subsystems and each subsystem has only one set point (see fig. 4.5). The first subsystem is subjected to only one linear inequality constraint and the performance index is a function of a second order.

The second example is taken from Shao and Roberts, 1983. It also consists of two subsystems. The first subsystem has two set points while the second has three set points (see fig 4.6). The performance index is also a function of the second order and the system is subjected to fixed bounds and non-linear inequality constraints. In addition, the system constraints involve outputs.

The third example is also taken from Shao and Roberts, 1983. It is a bigger and more labourious problem. It contains three subsystems with seven set points (see fig 4.7). The performance index is a function of the fourth

order and the system is subjected to fixed bounds and non-linear inequality constraints. The system equations involve non-linear terms.

The fourth example is taken from Tatjewski, 1985. It is a non-convex problem with a duality gap. The construction of the system is the same as that in example 1; figure (4.5).

In the simulation study of all the examples, the starting values of λ, c, u were obtained by solving the problem using the open loop model based optimisation technique (Findeisen et. al, 1980). This was carried out for both considered algorithms.

A selected representative set of simulation results for the double iterative price correction mechanisms (version 1 and version 2) and the augmented IBMF are presented for the simulation examples 1, 2, 3 and 4 in Tables (4.1) to (4.4) respectively.

The first two columns show the tolerances of the inner loop (Tol_1) and outer loop (Tol_2) respectively. The second two columns give the gains (ϵ_1, ϵ_2) for updating λ and s respectively, while in the case of AIBMF, they show the penalty coefficient (ρ) and the gain (α_t) for the outer loop to update λ. The following three columns show the final value of the performance index, the number of system iterations and the total number of optimising iterations. The last two columns represent the suboptimality figures and the computer processing time (cpt) respectively. The suboptimalities are calculated using the expression (Shao and Roberts,1983):

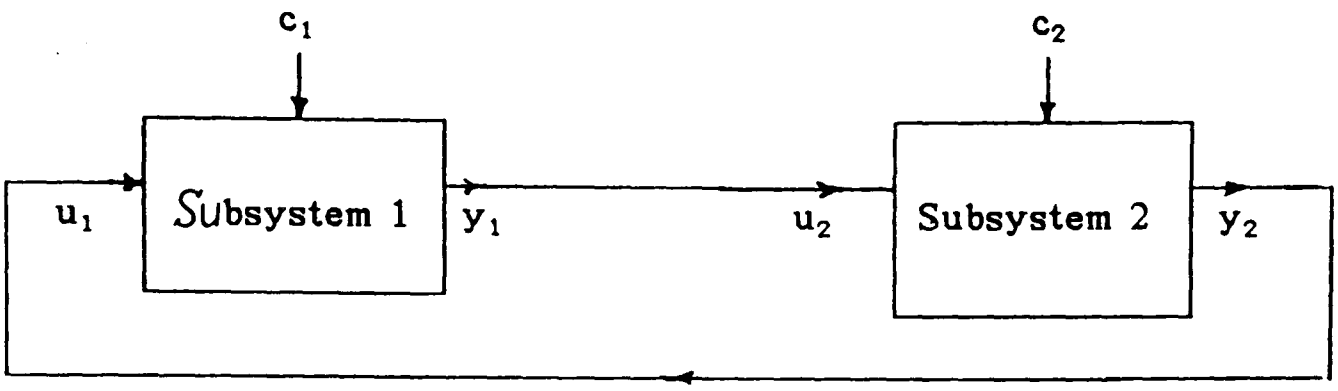


Figure (4. 5) The Structure of the System of Examples 1 and 4.

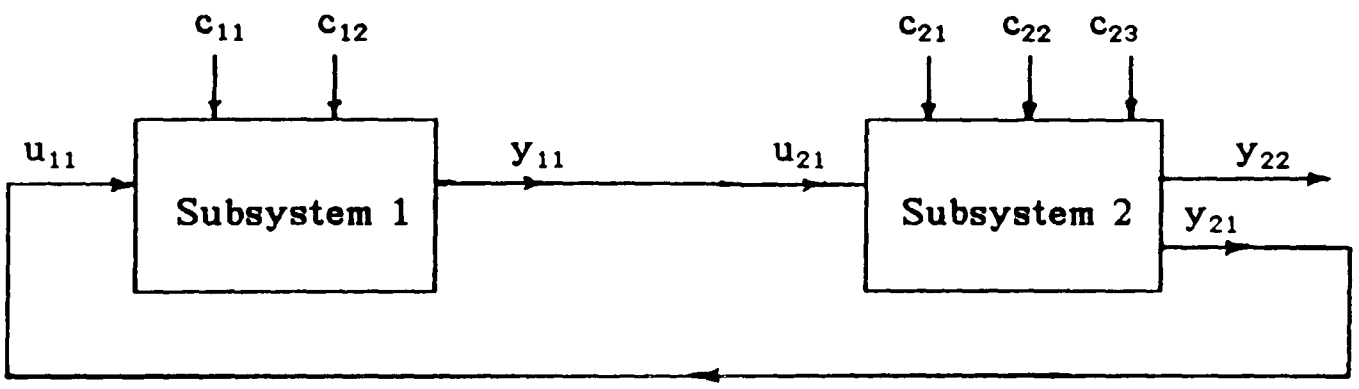


Figure (4. 6) The Structure of the System of Example 2.

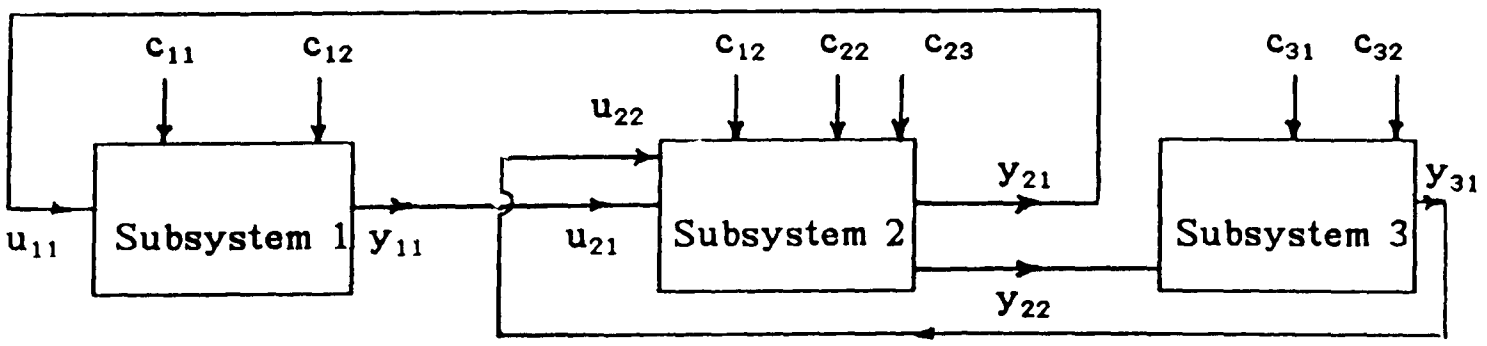


Figure (4. 7) The Structure of the System of Example 3.

ALG.	Tol_1	Tol_2	ϵ_1 ρ	ϵ_2 α	Q	IS	IT	Sub.%	CPt m	CPt s
V1	10^{-4}	10^{-4}	0.95	0.95	-0.06833	4	355	0.778	2	12
	$5x10^{-5}$	$5x10^{-5}$	0.95	0.95	-0.06824	5	671	0.905	3	54
V2	10^{-4}	10^{-4}	0.90	0.90	-0.06845	6	525	0.607	4	39
	$5x10^{-5}$	$5x10^{-5}$	0.95	0.95	-0.06812	5	818	1.082	8	47
AIBMF	10^{-4}	10^{-4}	30	1.0	-0.06852	10	40	0.501	-	26
	$5x10^{-5}$	$5x10^{-5}$	30	1.0	-0.06827	11	41	0.861	-	27

Table (4.1) Comparison of Algorithms for Example 1.

ALG.	Tol_1	Tol_2	ϵ_1 ρ	ϵ_2 α	Q	IS	IT	Sub.%	CPt m	CPt s
V1	10^{-4}	10^{-4}	0.95	0.95	5.98463	6	115	0.988	1	52
	$5x10^{-5}$	$5x10^{-5}$	0.90	0.60	5.98459	11	137	0.987	2	19
V2	10^{-4}	10^{-4}	0.95	0.95	6.04331	6	79	1.978	1	18
	$5x10^{-5}$	$5x10^{-5}$	0.90	0.60	6.04333	13	151	1.978	2	29
AIBMF	10^{-4}	10^{-4}	4	1.0	5.98457	18	63	0.987	1	08
	$5x10^{-5}$	$5x10^{-5}$	4	1.0	5.98457	19	72	0.987	1	15

Table (4.2) Comparison of Algorithms for Example 2.

ALG.	Tol_1	Tol_2	ϵ_1 ρ	ϵ_2 α	Q	IS	IT	Sub. %	CPt m	CPt s
V1	10^{-4}	10^{-4}	0.2	0.6	6.33583	8	209	0.1475	12	3
	$5*10^{-5}$	$5*10^{-5}$	0.2	0.6	6.33583	9	278	0.1475	16	13
V2	10^{-4}	10^{-4}	0.2	0.6	6.33578	8	216	0.1467	15	34
	$5*10^{-5}$	$5*10^{-5}$	0.2	0.6	6.33581	9	265	0.1471	18	46
AIPMF	10^{-4}	10^{-4}	10	1.0	6.33577	4	51	0.1465	6	38
	$5*10^{-5}$	$5*10^{-5}$	10	1.0	6.33576	5	67	0.1464	8	3

Table (4.3) Comparison of Algorithms for Example 3.

ALG.	Tol_1	Tol_2	ϵ_1 ρ	ϵ_2 α	Q	IS	IT	Sub. %	CPt m	CPt s
V1	10^{-4}	10^{-4}	0.5	0.5	NOT WORKING					
V2	10^{-4}	10^{-4}	0.5	0.5	NOT WORKING					
AIPMF	10^{-4}	10^{-4}	16	1.0	-9.54912	12	34	0.0108	-	26
	10^{-4}	10^{-4}	12	1.0	-9.54893	11	38	0.0128	-	22
	$5x10^{-5}$	$5x10^{-5}$	16	1.0	-9.54881	13	36	0.0140	-	27

Table (4.4) Comparison of Algorithms for Example 4.

$$\text{Suboptimality} = \frac{Q-Q^*}{Q} \times 100\% \tag{4.28}$$

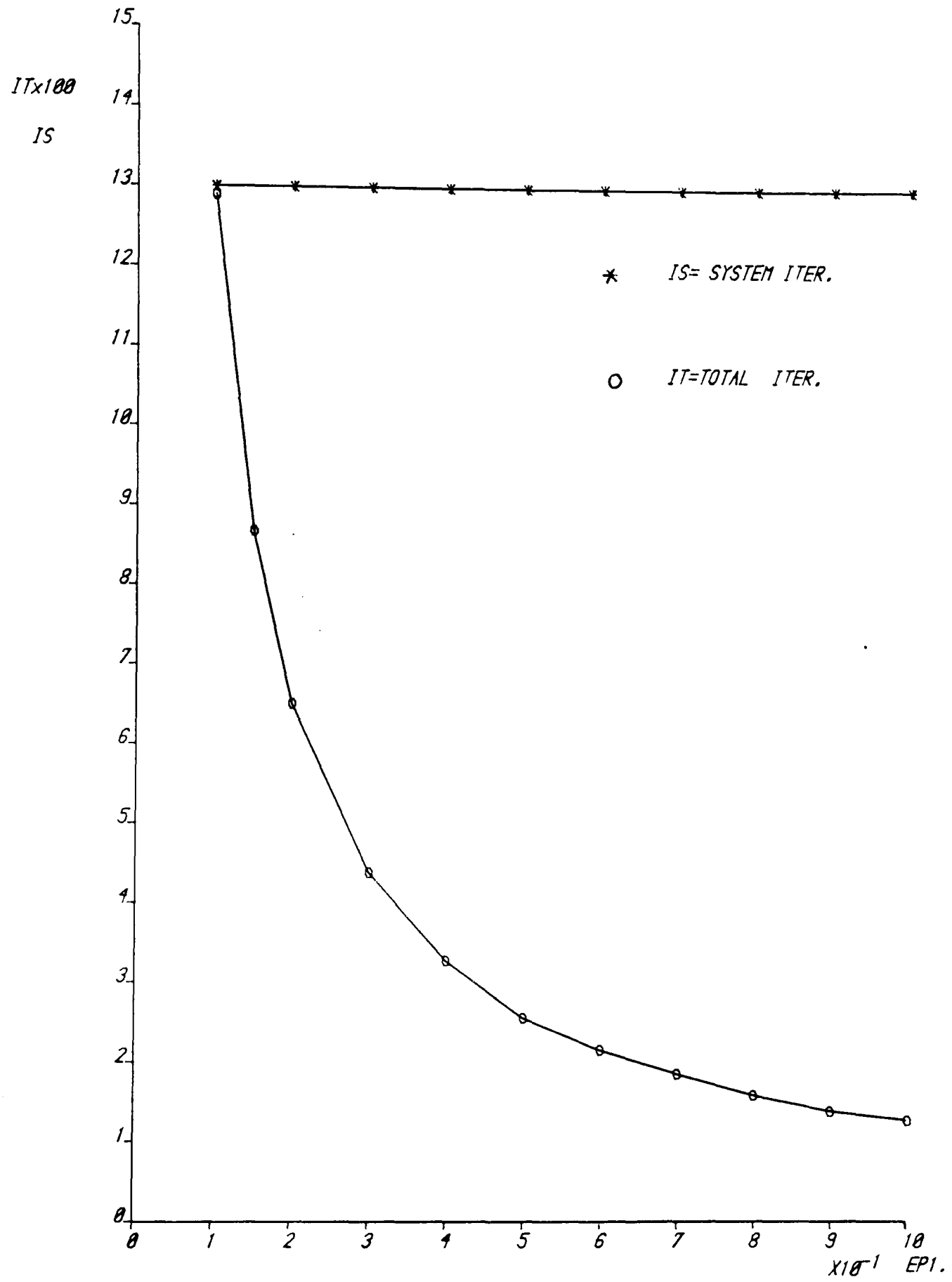
where Q is the obtained final value of performance index, and Q^* is the real optimum performance that would have been obtained if the model were perfect.

The computational effort of the algorithms were measured by the number of system iterations (IS), number of optimising iterations (IT) and computer processing time (cpt) which are shown for the algorithms and the simulation examples in tables (4.1) to (4.4)

4.7.2 Sensitivity to parameters:

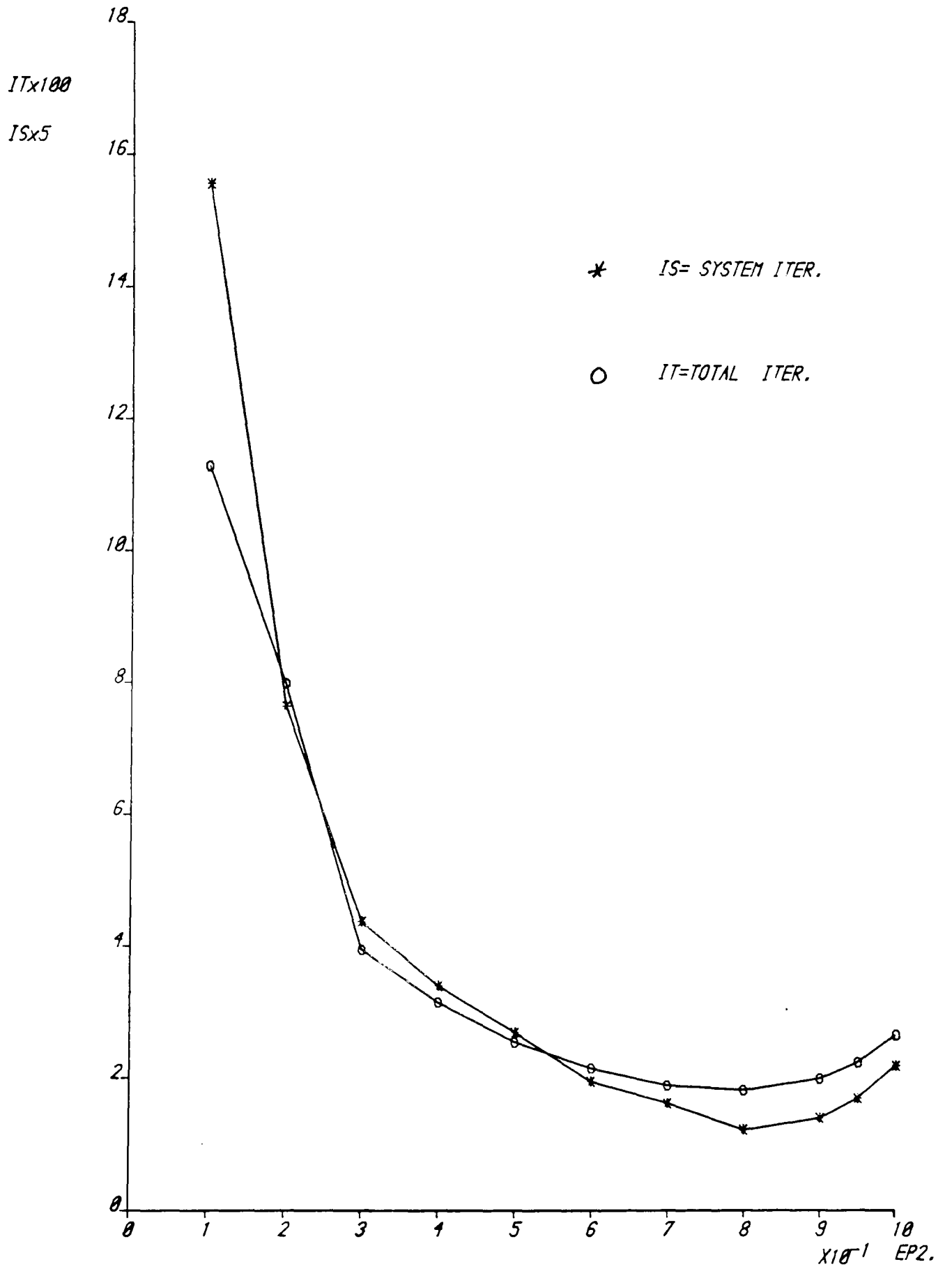
The sensitivity of the double iterative price correction mechanisms, both versions (V1 and V2), were studied using example 1. The results are shown in figures (4.8) to (4.11). Figures (4.8) and (4.9) show respectively the effect of the gain coefficients (ϵ_1) for updating the price vector λ and the gain coefficient (ϵ_2) for updating the shift vector s on the number of system iterations (IS) and total iterations (IT). These results are obtained for version 1 and a similar set of results are shown in figures (4.10) and (4.11) for version 2. The tolerance of the double loops were both satisfied to 10^{-4} and in studying the effect of a gain, the other gain value was kept constant to an average value of 0.5.

In the augmented interaction balance method with feedback the contained parameters are the gain coefficient for updating λ ; α_t and the penalty factor ρ. A study of the effect of α_t on the number of system iterations and total iterations are presented respectively in figures (4.12) and (4.13) for penalty coefficient $\rho = 10$ and using example 1. Figure (4.14) shows the effect of varying the value of the penalty coefficient on convergence for example 1. During the study of penalty coefficient effects, the gain α_t was kept constant to a value of 1.0 and the tolerance for terminating the iterative loops of the algorithm were both 10^{-4}.



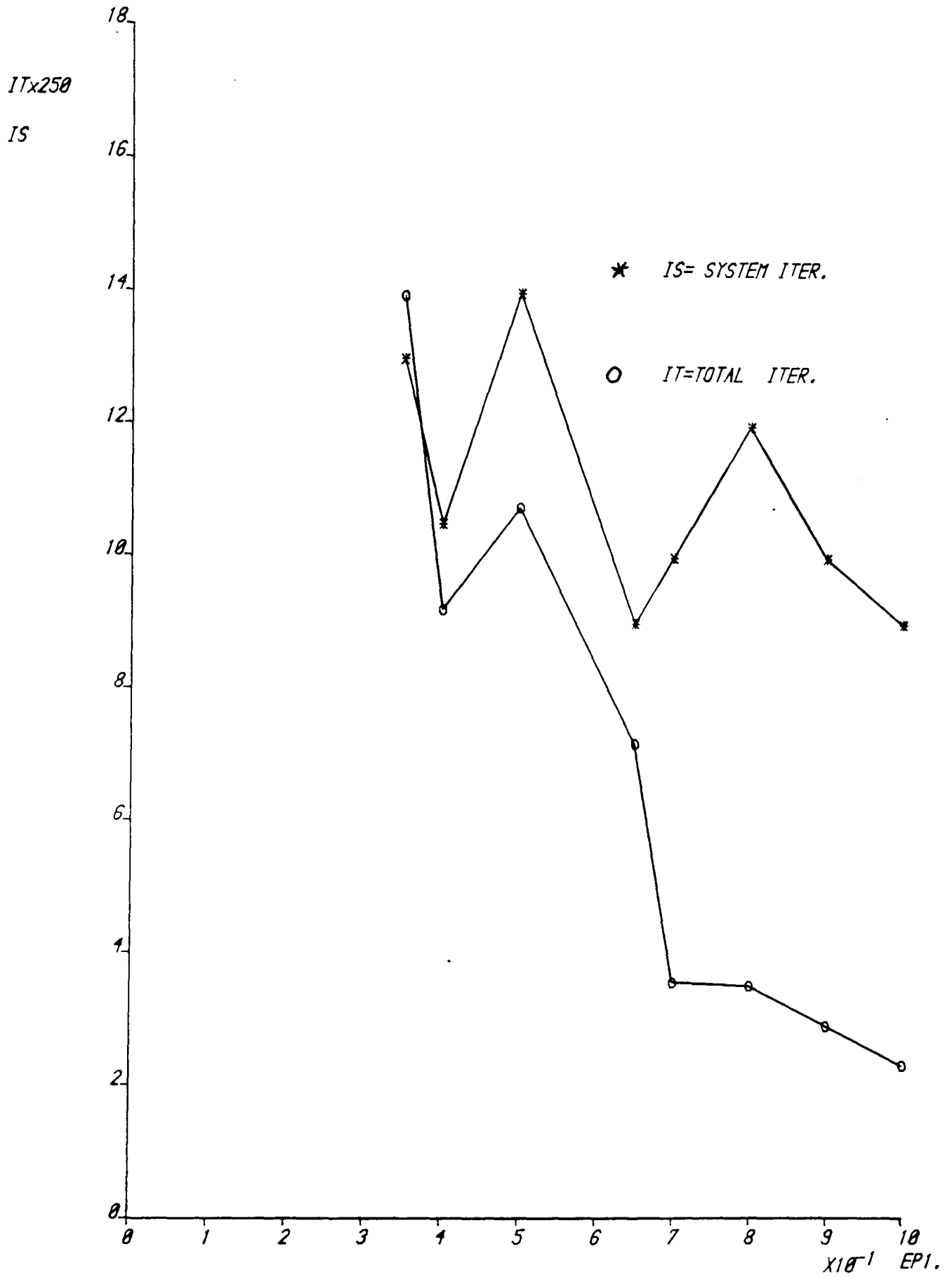
FIGURE(4.8) SENSITIVITY OF DIPCM(V.1),EXAMPLE 1

EFFECT OF GAIN(EP1) ON IT/IS ,EP2=0.5.



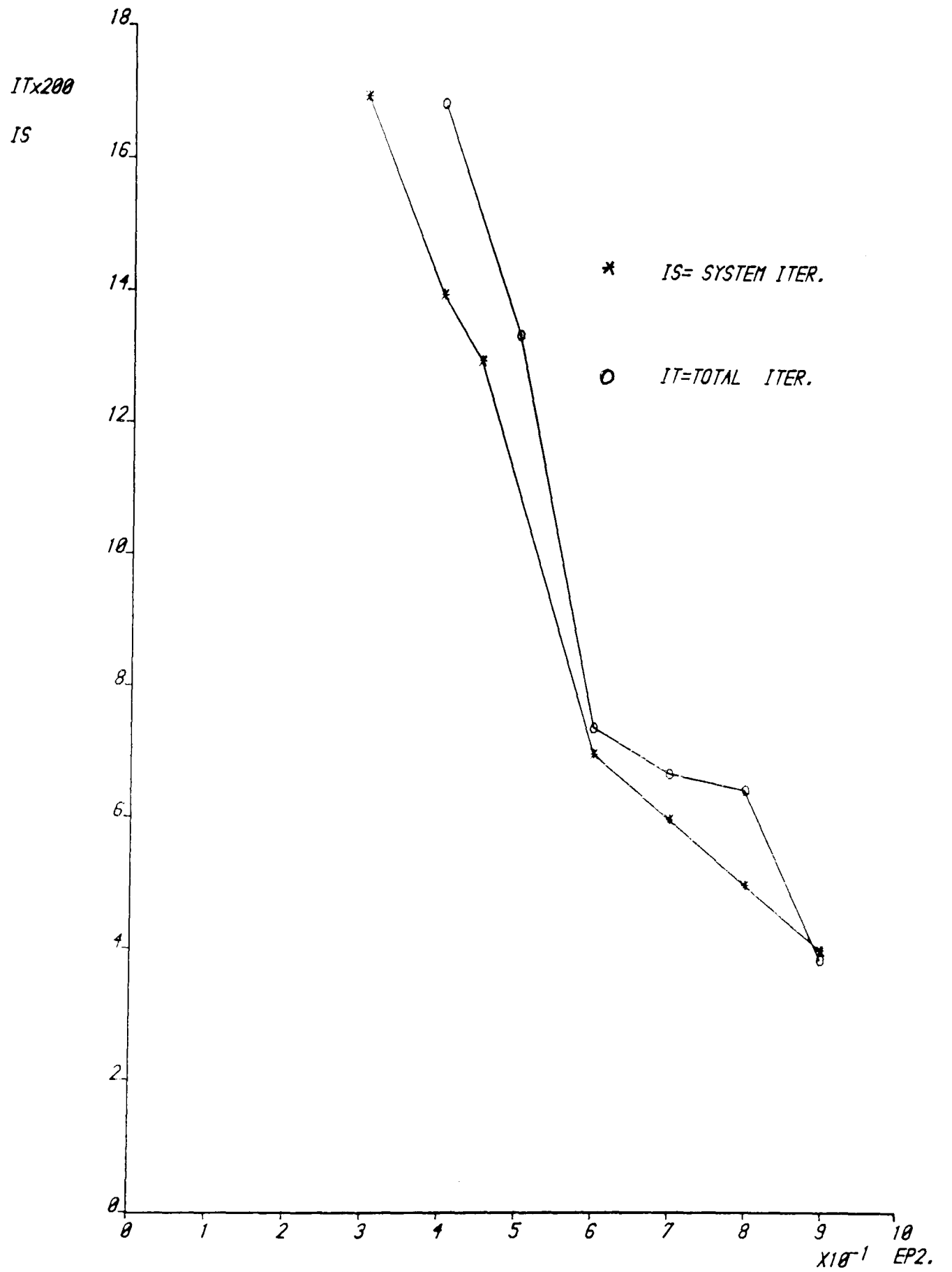
FIGURE(4.9) SENSITIVITY OF DIPCM(V.1),EXAMPLE 1

EFFECT OF GAIN(EP2) ON IT/IS ,EP1=0.5.



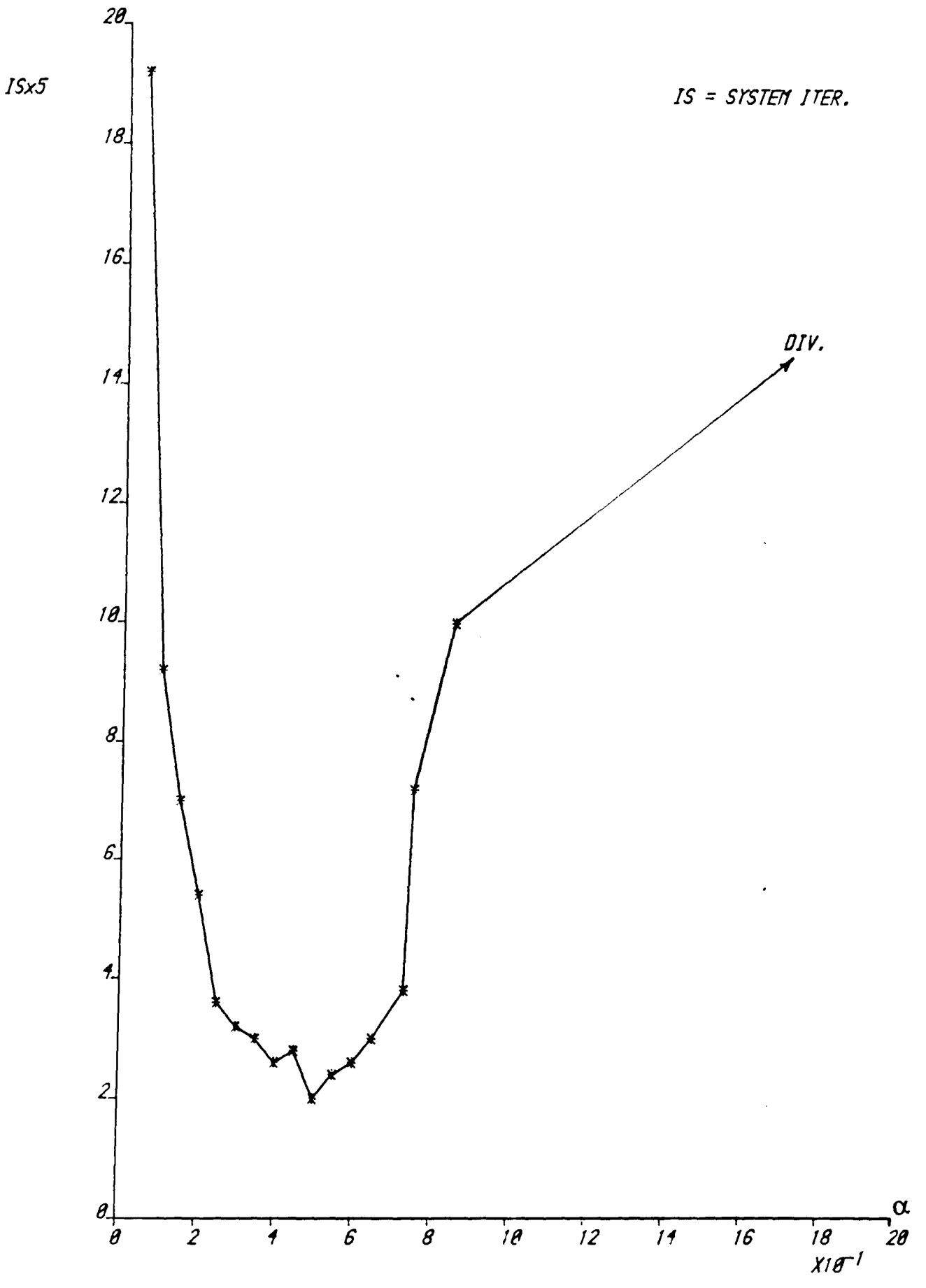
FIGURE(4.10) SENSITIVITY OF DIPCM(V.2),EX. 1

EFFECT OF GAIN(EP1) ON IT/IS ,EP2=0.5.

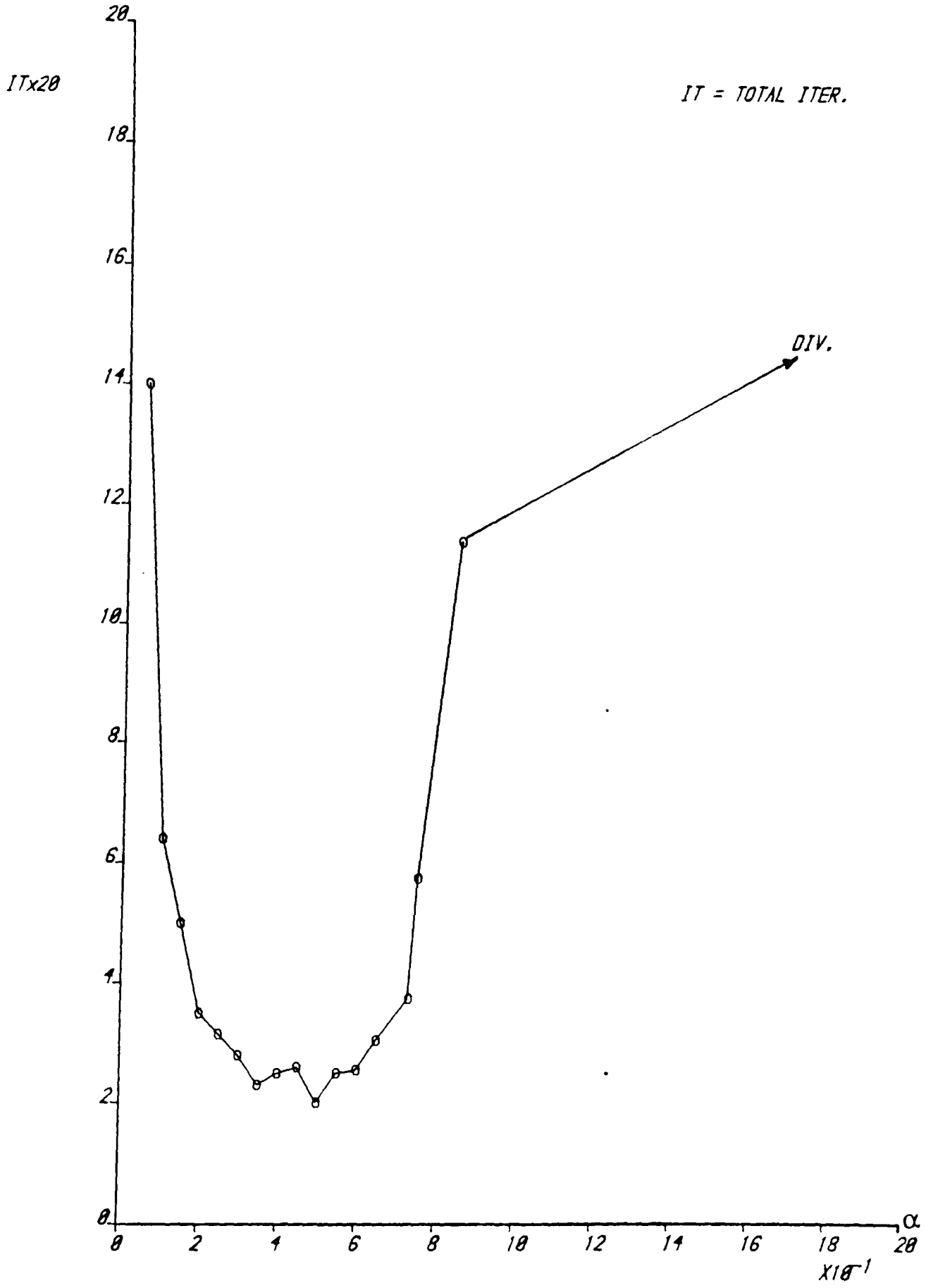


FIGURE(4.11) SENSITIVITY OF DIPCM(V.2),EX. 1

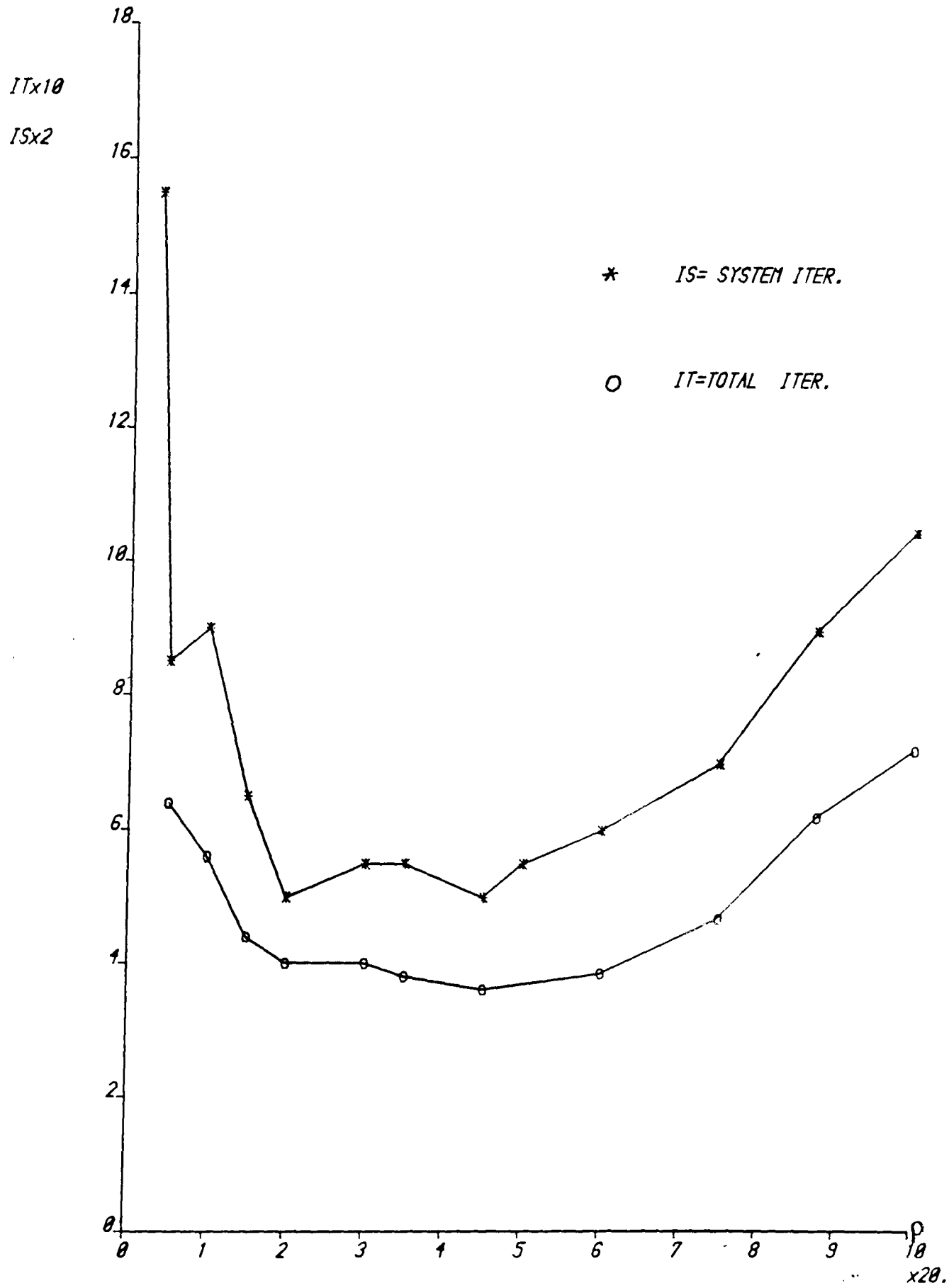
EFFECT OF GAIN(EP2) ON IT/IS ,EP1=0.5.



FIGURE(4.12) EFFECT OF α ON IS FOR ρ=10, EX. 1



FIGURE(4.13) EFFECT OF α ON IT FOR ρ=10, EX. 1



FIGURE(4.14) SENSITIVITY OF THE AIBMF, EX. 1

EFFECT OF ρ ON IT/IS , α=1.0.

4.7.3 Study of non-linearity:

Example 3 has been modified to include a varying non-linearity term in the reality equation of the first subsystem. So, the system equations become:

$$y_{11}^{*} = F_{11}^{*}(c_1, u_1) = 1.3c_{11} - c_{12} + 2u_{11} + \beta u_{11}c_{11}$$

$$y_{21}^{*} = F_{21}^{*}(c_2, u_2) = c_{21} - c_{22} + 1.2u_{21} - 3u_{22} + 0.1c_{22}^{2}$$

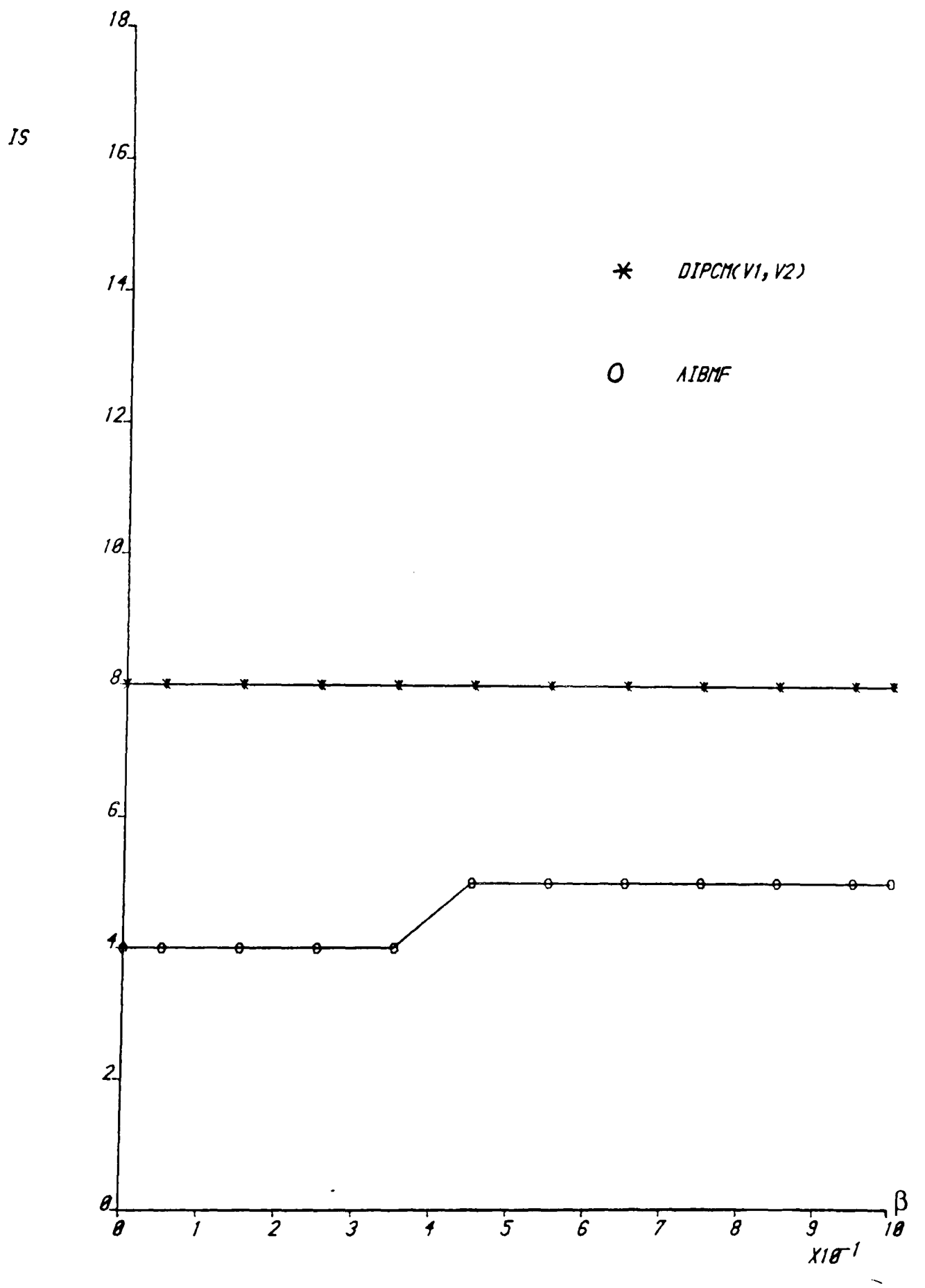
$$y_{22}^{*} = F_{22}^{*}(c_2, u_2) = 2c_{22} - 1.25c_{23} - u_{21} + u_{22} + 0.25c_{22}c_{23} + 0.1$$

$$y_{31}^{*} = F_{31}^{*}(c_3, u_3) = 0.8c_{31} + 2.5c_{32} - 4.2u_{31}$$

where $\beta \geq 0$ is a parameter scaling non-linear term in y_{11}^{*} with a nominal value of β=0.15.

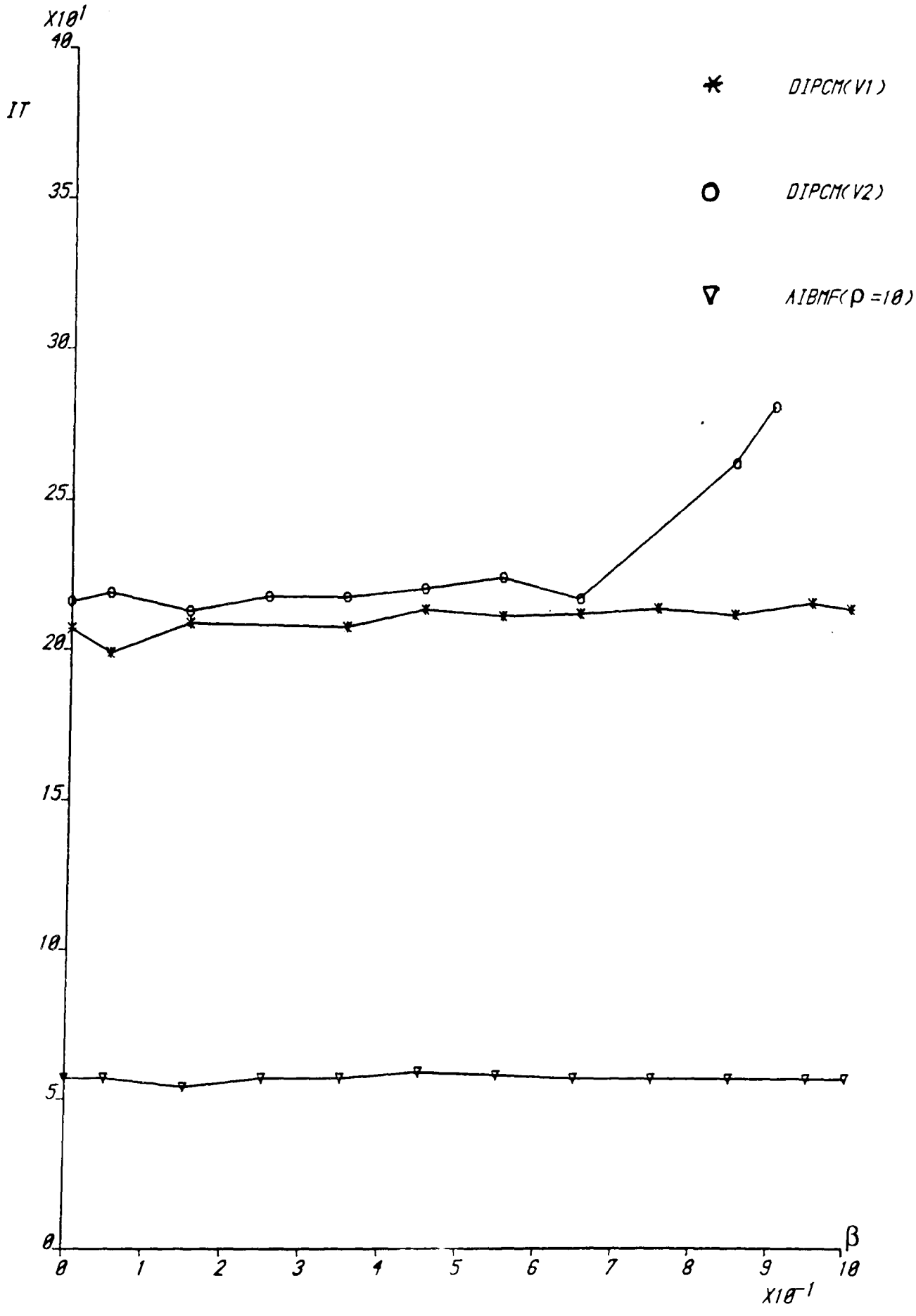
The model equations, performance indices, sujected constraints and coupling equations are the same as in example 3.

Results for $\beta \in [0, 1]$ are shown in figures (4.15) and (4.16) for the double iterative price correction mechanisms (V1 and V2) and the augmented interaction balance method with feedback (AIBMF). The shown results were obtained using a penalty coefficient ρ = 10 and an average gain value α_t = 1.0 for the AIMBF and for versions V1, V2 the gain values for the inner and outer loop respectively are 0.2 and 0.6. The tolerances for the iterative loops for all the results of this study were satisfied to a value of 10^{-4}.



FIGURE(4.15) EFFECT OF NONLINEARITY ON IS

FOR THE DIPCM(V1, V2) AND THE AIBMF(ρ=10), EX. 3



FIGURE(4.16) EFFECT OF NONLINEARITY ON TOTAL ITER. FOR DIPCM(V1, V2) AND THE AIBMF, (EX.3)

4.7.4 Convergence and precision:

The convergence and the quality of the points produced by both algorithms are measured from the previous simulation study of the examples.

The quality of the final converged performance function is measured by comparing it with the real optimum value and the suboptimality for each solution is shown in tables (4.1), (4.2), (4.3) and (4.4) for examples 1, 2, 3 and 4 respectively.

The convergence of the performance index of example 3 is shown in figure (4.17) for the algorithms (V1, V2 and AIBMF) while figures (4.18) and (4.19) show the convergence of the set point changes (c_{11}, c_{12}) for the first subsystem of the same example. For V1 and V2, the gain of the inner and outer loops were respectively 0.2 and 0.6. The penalty coefficient value for the AIBMF was 10 and the tolerance for all algorithms for both loops were achieved to 10^{-4}.

4.8 Discussion of results:

The results obtained from the simulation study using four examples are presented in tables (4.1) to (4.4). These indicate that the DIPCM and the AIBMF are both applicable to a wide range and different sizes of problems.

However, the DIPCM are not applicable to non-convex problems which have a duality gap since they are based on normal Lagrangians, as it is evident from table (4.4).

Both algorithms produce suboptimal results, but the optimality of the DIPCM (V2) in most cases is worse than the other two algorithms. The AIBMF produces results closer to the optimium. The total number of iterations of the AIBMF is much less than those of V1 and V2. Therefore, its computer processing time is also often very much less than V1 and V2,

but the number of system set point changes (IS) of the AIMBF is higher than V1 and V2 for examples 1 and 2.

Increasing the value of the price gain ϵ_1 will help to decrease the total number of iterations (IT) of V1 as it is shown in figure (4.8) for example 1, while the system set point changes are not affected by varying ϵ_1. However, for this particular example, IS and IT of the algorithm V1 both vary with the gain ϵ_2, where by increasing ϵ_2 decreases IT and IS, until $\epsilon_2 \leq 0.8$, then increasing $0.8 < \epsilon_2 \leq 1.0$ gradually increases IS and IT as shown in figure (4.9).

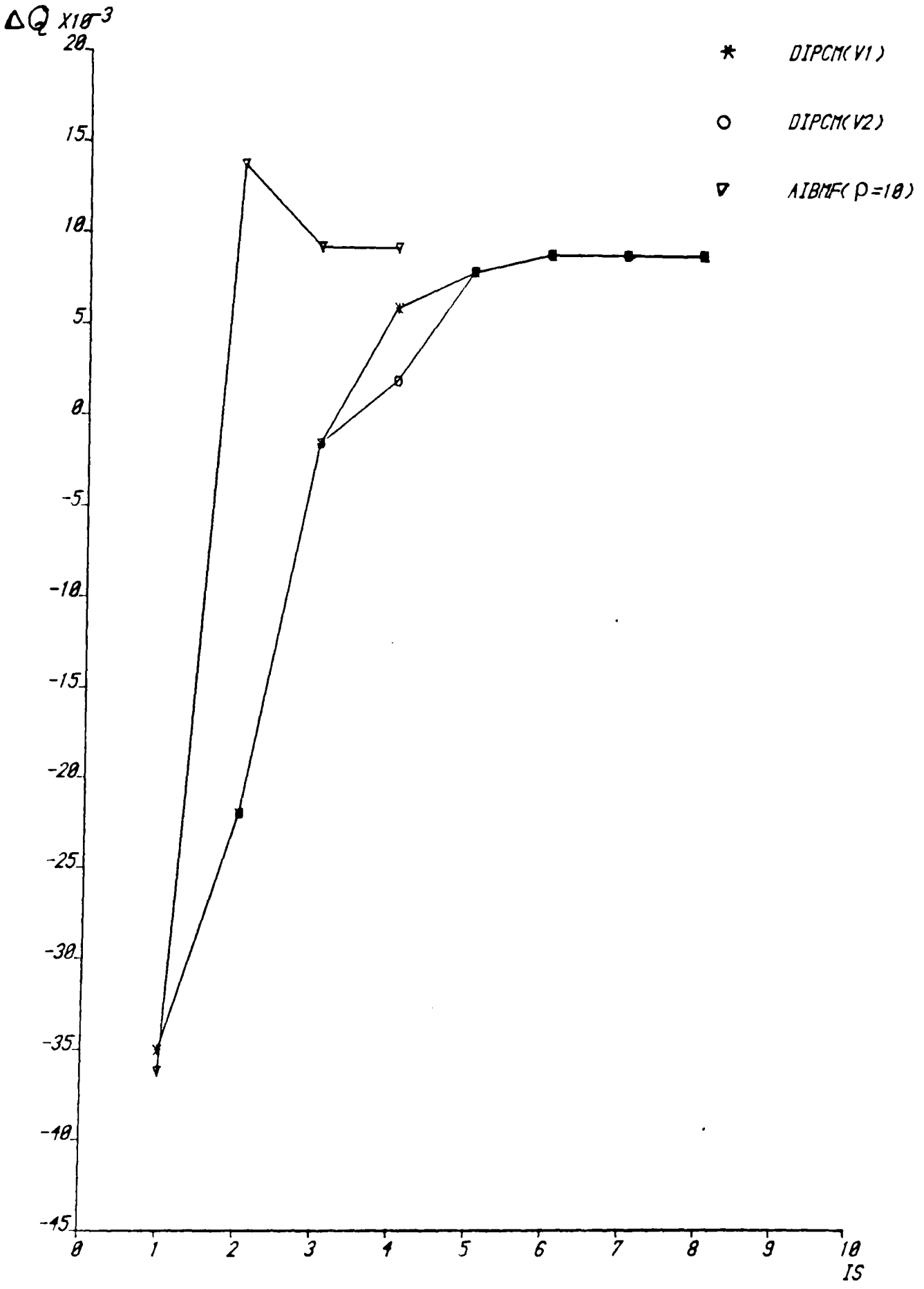
Version V2 is sensitive to ϵ_1 and ϵ_2 as it is evident from figures (4.10) and (4.11).

A study of the effect of varying the gain α_t (which is required to update the price vector λ of the AIBMF, where $0 \leq \alpha_t \leq 2$) is shown in figures (4.12) and (4.13) for $\rho = 10$ and using example 1. It can be noticed that the number of set point changes sharply decreases for increasing α_t within the range $0 \leq \alpha_t \leq 0.6$, then it gradually decreases for $0.6 \leq \alpha_t \leq 1.0$. Any increase in α_t will increase IS gradually in the beginning then sharply until $\alpha_t = 1.7$. For $\alpha_t > 1.7$, the algorithm fails to converge for this example.

The effect of varying α_t on IT is exactly the same as on IS as is seen in figures (4.12) and (4.13).

A study of the effect of the penalty coefficient ρ on the convergence of the AIBMF is shown in figure (4.14) for $0 \leq \rho \leq 200$ for example 1. The performance of the algorithm improves by increasing ρ until $\rho \leq 100$ then IT and IS increase for every increase of ρ, for this example.

A test of the DIPCM (V1, V2) and the AIBMF, with respect to non-linearity can be seen in figures (4.15) and (4.16). The DIPCM and the AIBMF algorithms show a good performance throughout the range $0 \leq \beta \leq 1.0$, where for the DIPCM the converged solutions were always obtained after 8 set point changes and 4 to 5 set point changes for the AIBMF. There is an exception of V2 where IT deteriorates for $\beta > 0.65$.



FIGURE(4.17) CONVERGENCE OF ALGORITHMS FOR EXAMPLE 3.

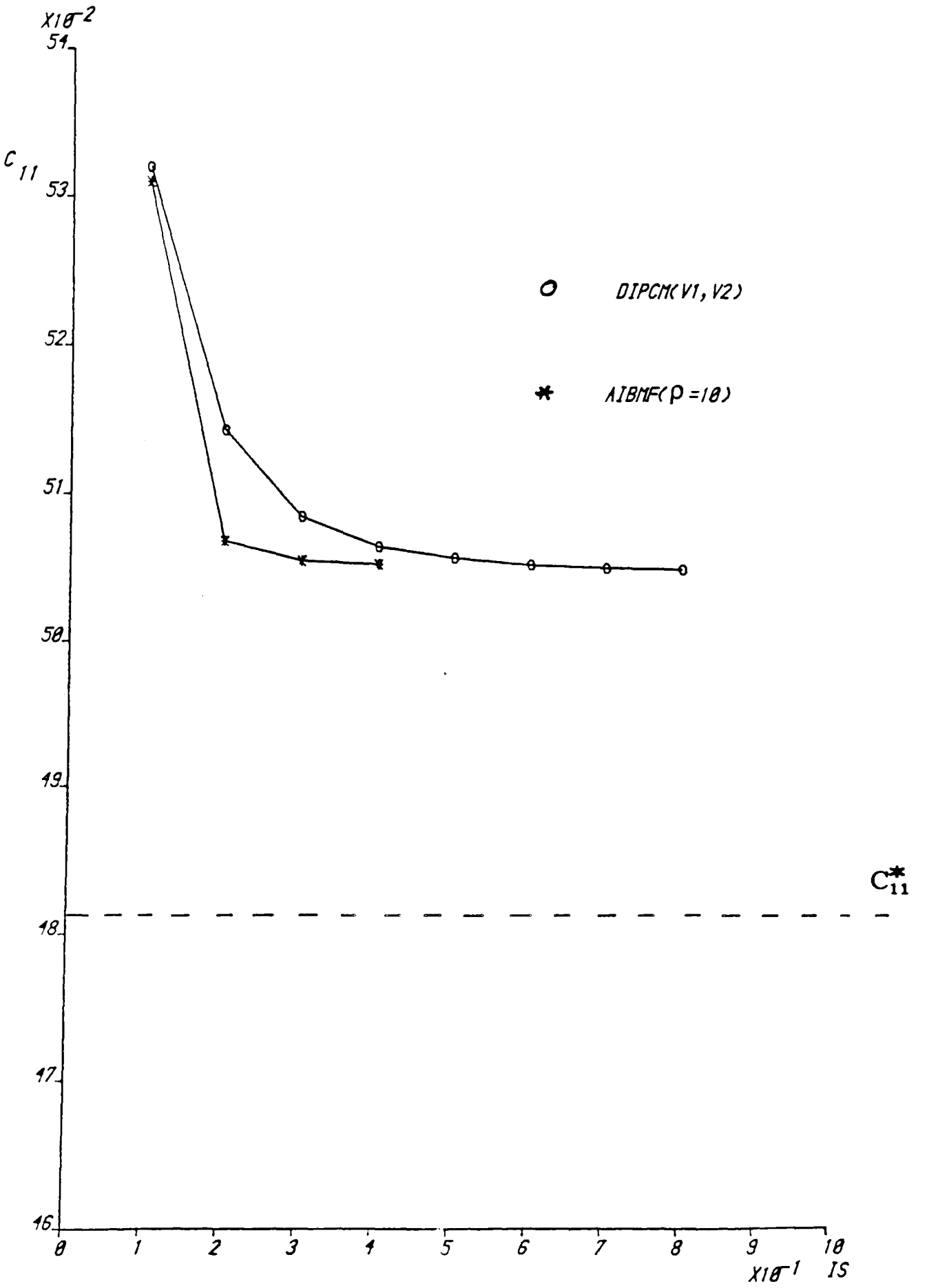
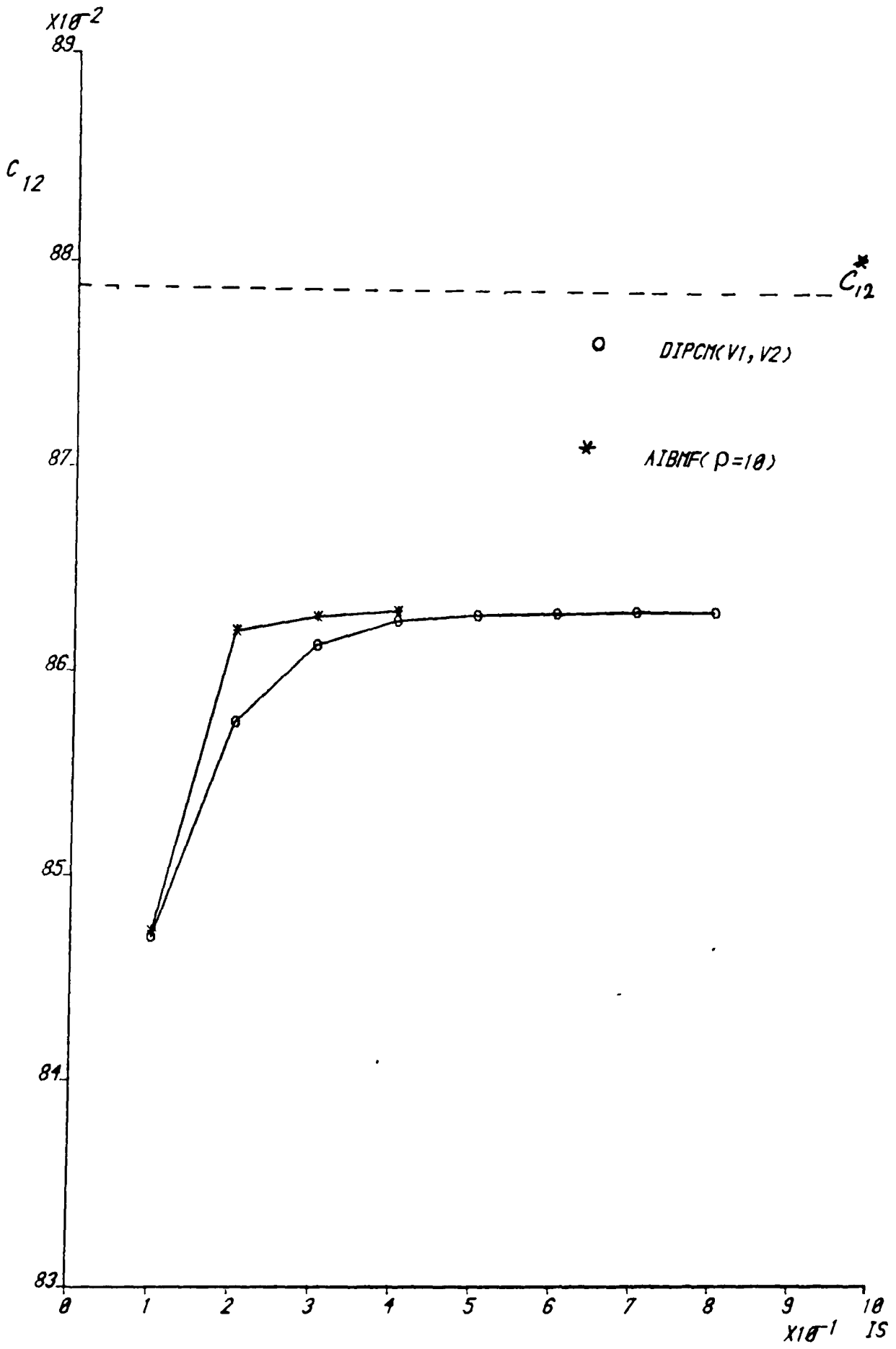


Figure (4.18) Convergence of set point C_{11} , Ex. 3



FIGURE(4.19) CONVERGENCE OF SET POINT C_{12}, EX. 3

The convergence of the DIPCM and the AIBMF for example 3 is shown in figure (4.17) where figures (4.18) and (4.19) present the convergence of the set point changes c_{11} and c_{12} of the first subsystem of this problem. It can be noticed that the AIBMF initially converges sharply towards the optimum for this example, and therefore the number of set point changes of the AIBMF is consequently smaller than V1 and V2.

4.9 Summary and conclusions:

Both algorithms, the AIBMF and the DIPCM are applicable on a wide range and different sizes of problems with the exception of the DIPCM in case of non-convex problems. An extension of the DIPCM to make it applicable to non-convex problems is described in chapter 5.

The DIPCM and the AIBMF both produce suboptimal results but the suboptimality of V2 is often higher than V1 and the AIBMF.

The solution of V1 of the DIPCM often converges within less set point changes than V2 and the AIBMF with the exception of example 3, where the AIBMF was superior. But the total number of iterations IT of the AIMBF are always less than those of the DIPCM. This is due to fast convergence of the inner loop of the algorithm (Tatjewski, 1985). Therefore, the computing processing time of the AIBMF are always smaller than those of the DIPCM.

Both algorithms have two parmeters to be selected; (ϵ_1, ϵ_2) for the DIPCM and (ρ, α_t) for the AIBMF.The best values of these parameters are obtained by trial and errors, where high values of (ϵ_1, ϵ_2, α_t) may cause divergence and too small values will considerably slow the solution unnecessarily, but V2 is more sensitive to these parameters than V1 and AIBMF.

Finally, both algorithms can cope with system non-linearity with the exception of an observed deterioration of V2 which, for the particular example investigated exhibited an increase in the total number of iterations for $\beta > 0.65$.

Chapter 5

An on-line Augmented Price Correction algorithm

5.1 Introduction

In previous chapters it was mentioned that it is always difficult to formulate a mathematical model of a system with no differences between the mathematical model and reality. To cope with this problem, techniques with feedback information from the real system have been introduced by Findeisen et. al. (1980). Another problem which occurs when using these methods is that a large number of set point changes may be required in order to converge to the final solution, where for every iteration it is necessary to apply the control set points to the real system and wait for a new steady state in order to obtain the required information. Hence, a large amount of on-line computing time may be needed. Some research work has been done to reduce the number of times that information is required from the real process by Shao and Roberts (1983).

However, there are some application limitations on the double iterative technique by Shao and Roberts (1983). For instance, the technique can not be safely applied if the problem is non-convex and has a duality gap.

In this chapter, an alternative approach is introduced which was proposed by Hendawy and Roberts (1989). The new algorithm is designed to extend the applicability of the previous price correction mechanism to cover the case of non-convex problems, and to improve the effectiveness of the iterative correction process. The structure of this algorithm is of a

hierarchical nature with three loops. The real process measurements are required only within the outer loop while the other two inner loops involve model based computations only. This reduces on-line control correction time and makes application of the algorithm also useful for convex problems which have no duality gaps.

5.2 The Augmented Price Correction Mechanism

5.2.1 Description of the algorithm

The algorithm is originally based on two main principals:

i) Shifting the model outputs of the problem

The model output is shifted from

$$y = F(c , u) \tag{5.1}$$

to

$$y^s = F(c , u) + s = y + s \tag{5.2}$$

where $s \in \mathcal{Y}$ is a fixed vector provided by the coordinator . It is expected at some value of s, the model output $\hat{y}$ will be equal to the real output y^*.

The interconnection input vector is also shifted from

$$u = H\, y \tag{5.3}$$

to

$$u^s = H\, y^s = H\,(y + s) = u + H\, s \tag{5.4}$$

The performance index and feasible set within the local problems are also modified as follows:

$$Q = \sum_{i=i}^{N} Q_i(c_i , u_i^s, y_i^s) = \sum_{i=1}^{N} Q_i\,(c_i , u_i + H_i s , y_i + s_i\,) \tag{5.5}$$

$$C\ U_i = \left\{ (c_i , u_i\,) \in \mathcal{C}_i \times \mathcal{U}_i \right\} \tag{5.6}$$

$$G_j(c_i , u_i + H_i s , F_i(c_i , u_i) + s_i) \geq 0 , \qquad j \in J_i \tag{5.7}$$

and

$$u_i = H F_i (c_i , u_i) \tag{5.8}$$

where $i \in 1 , N$

ii) Applying the augmented lagrangian technique

The augmented lagrangian of the overall problem is:

$$\mathcal{L}_a (c, u, y, \lambda, \rho) = Q (c , u , y) + \lambda^t (u - HF (c , u))$$

$$+ 0.5 \rho \| u - HF (c , u) \|^2 \tag{5.9}$$

By shifting the model outputs, equation (5.9) can be written as follows:

$$\mathcal{L}_{as} (c , u , y , \lambda , \rho , s) = Q (c , u + Hs, y+s) + \lambda^t (u - HF (c , u))$$

$$+ 0.5 \rho \| u - HF (c , u) \|^2 \tag{5.10}$$

The shifted augmented Lagrangian in equation (5.10) can be written in the form of a hierarchical decomposable structure as:

$$\mathcal{L}_{as} (c, u, y , \lambda, \rho, s) = \sum_{i=1}^{N} [Q_i (c_i , u_i + H_i s, F_i(c_i , u_i) + s_i)$$

$$+ \lambda_i^t \; u_i \; - \sum_{j=1}^{N} \lambda_j^t \; H_{ji} F_i(c_i , u_i)$$

$$+ 0.5 \rho (\| u_i \|^2 + \| F_i(c_i, u_i) \|^2) - \rho u^t H F(c , u) \tag{5.11}$$

The augmented Lagrangian in this form can not be applied directly because the last term in equation (5.11) is not separable.
Using a suitable linearisation technique as proposed by Stephanopoulos and Westerberg (1975), this term can be linearized around some point $(u^k , HF (c^k , u^k))$ as :

$$u^t\ HF\ (\ c\ ,\ u\ ) = -\ u^{k^t}\ (\ HF\ (\ c\ ,\ u\ )\ ) + u^t\ HF\ (\ c^k\ ,\ u^k\ )$$

$$+\ u^{k^t}\ HF\ (\ c\ ,\ u\ ) \qquad (5.12)$$

Substituting from equation (5.12) into equation (5.11), the augmented Lagrangian can be transformed to the following separable form:

$$\Lambda_{as}\ (c,\ u,\ \lambda,\ \rho,\ s,\ c^k,\ u^k) = \sum_{i=1}^{N}\ [\ Q_i\ (\ c_i\ ,\ u_i + H_i s\ ,\ F_i(\ c_i\ ,\ u_i\ ) + s_i\ )$$

$$+\ \lambda_i^t\ u_i - \sum_{j=1}^{N} \lambda_j^t\ H_{ji}\ F_i(\ c_i\ ,\ u_i\ ) + 0.5\ \rho\ (\|\ u_i\ \|^2 + \|\ F_i\ (c_i\ ,\ u_i\ )\ \|^2$$

$$-2\ u_i^t\ H\ F\ (c^k,\ u^k\ ) - 2\sum_{j=1}^{N} u_j^{k^t}\ H_{ji}\ F_i(\ c_i\ ,\ u_i\ ))] + \rho\ u^{k^t}\ HF\ (c^k\ ,\ u^k\ )$$

$$= \sum_{i=1}^{N}\ \Lambda_{as}\ \ (c_i\ ,\ u_i\ ,\ \lambda\ ,\ \rho\ ,\ s\ ,\ c^k\ ,\ u^k) + \rho\ u^{k^t}\ HF(\ c^k\ ,\ u^k\ ) \qquad (5.13)$$

5.2.2 Implementation aspects of the algorithm

The computer flow chart of the augmented price correction mechanism with feedback is illustrated in Figure (5.1) while the proposed structure of the algorithm is shown in Figure (5.2). It can be described as follows:

For given initial values of λ , c^0 , u^0 which are obtained by solving the problem using the open loop model based optimisation caculation (Findeisen et. al., 1980) the on-line implementation involves performing repeatedly the following procedure:

1. For a given value of s, solve problem (5.14) using model based calculations to obtain solutions ($\hat{c}_i\ (\ \lambda\ ,\ s\ )$, $\hat{u}_i\ (\lambda\ ,\ s\ )$).

$$\min_{(c_i\ ,\ u_i)\ \in\ CU_i}\ \ \Lambda_{asi}\ \ (c_i\ ,\ u_i\ ,\ \lambda\ ,\ \rho\ ,\ s\ ,\ c^k\ ,\ u^k) \quad ,\ i=1,\ \ldots\ ,\ N \qquad (5.14)$$

2. Repeat solving problem (5.14) until condition (5.16) is achieved, otherwise

reset

$$(c^{k+1} , u^{k+1}) = (\hat{c} (\lambda , s) , \hat{u}(\lambda , s)) \tag{5.15}$$

for the following solution of problem (5.14) until condition (5.16) is satisfied within a desired tolerance.

$$(\hat{c} (\lambda , s) , \hat{u}(\lambda , s) - (c^{k} , u^{k})) = 0 \tag{5.16}$$

3. Compare the obtained interaction input vector $\hat{u}(\lambda,s)$ with the overall balance condition $HF(\hat{c} (\lambda , s) , \hat{u}(\lambda , s))$, using

$$\hat{u}(\lambda , s) - HF(\hat{c} (\lambda , s) , \hat{u}(\lambda , s)) = 0 \tag{5.17}$$

If condition (5.17) is not fulfilled within a reasonable tolerance, then the coordinator improves the values of λ according to:

$$\lambda^{n+1} = \lambda^{n} + \rho \, \epsilon_1 \, [\, \hat{u}(\lambda^{n}, s) - HF(\hat{c} (\lambda^{n} , s) , \hat{u}(\lambda^{n}, s) \,] \tag{5.18}$$

where ρ is a penalty coefficient factor, ϵ_1 is a step length coefficient and

$$0 \leq \epsilon_1 \leq 1 .$$

The updated values of λ will be used, and the process is repeated from the beginning until condition (5.19) is satisfied within a desired tolerance.

4. Control $\hat{c} (\lambda(s))$ is applied to the real system as controller set points. After the transient process in the system is terminated and the system reaches a steady state condition, an outer iterative procedure is employed to adjust $\hat{s}$ from knowledge of the difference between real and shifted model outputs. The real steady state outputs $y^* = K^* (\hat{c}(s))$ are measured and transmitted to the coordinator.

5. The coordinator improves the values of " s " according to equation (5.19) (Shao and Roberts, 1983) and then sends it to the local decision units to start from the beginning, as long as condition (5.20) is not satisfied.

$$s^{j+1} = s^j + \epsilon_2 [\ K^*(\hat{c}(s^j)) - F(\hat{c}(s^j), \hat{u}(s^j)) - s^j\ ] \tag{5.19}$$

where ϵ_2 is a step length coefficient , $0 \leq \epsilon_2 \leq 1$

$$K^*(\ \hat{c}(\hat{\lambda}(\hat{s}))) - F(\hat{c}(\hat{\lambda}(\hat{s})), \hat{u}(\hat{\lambda}(\hat{s}))) - \hat{s} = 0 \tag{5.20}$$

The iterations are continued until condition (5.20) is fulfilled within a desired tolerance.

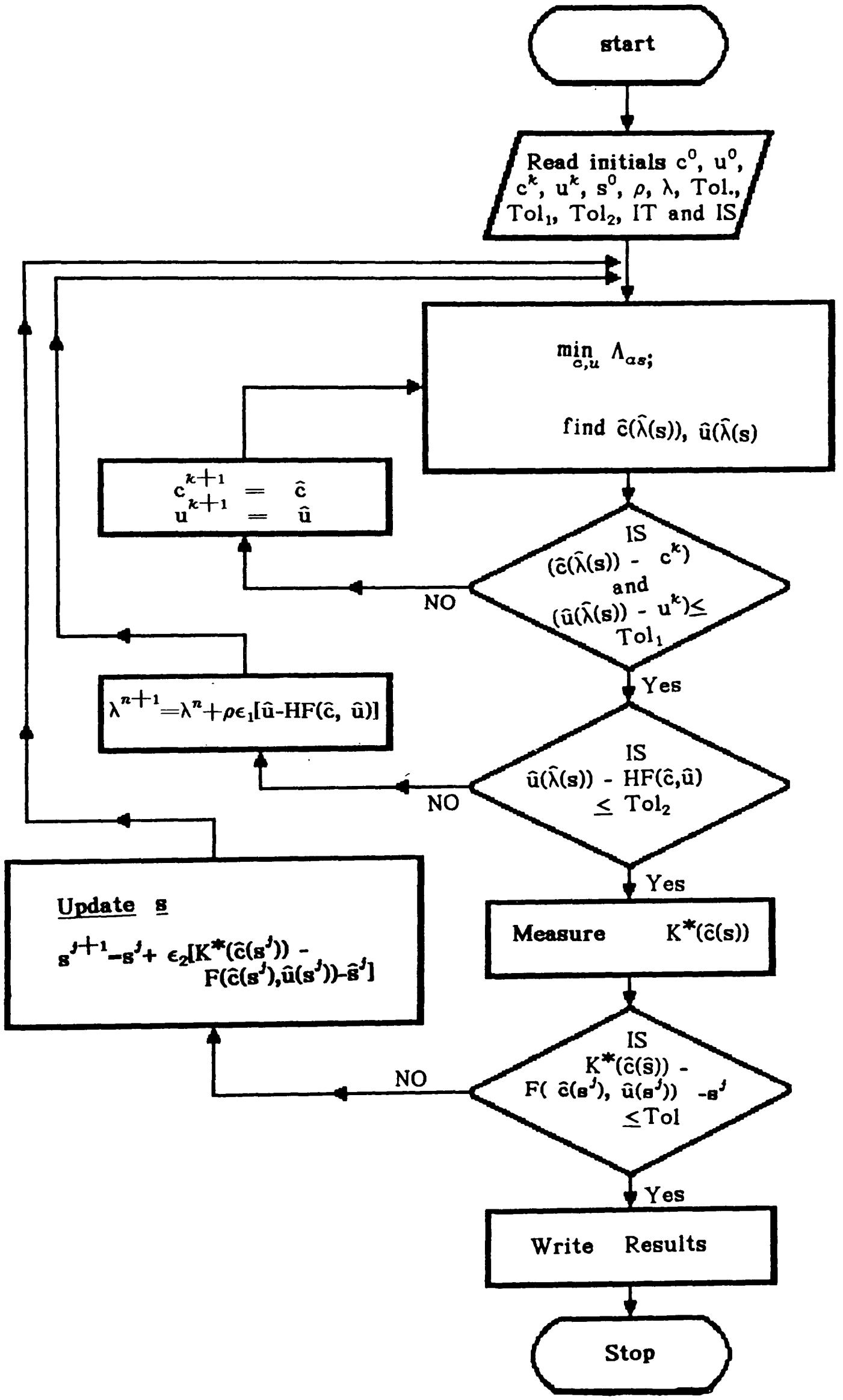


Figure (5.1) Flow chart of the Augmented Price Correction Algorithm.

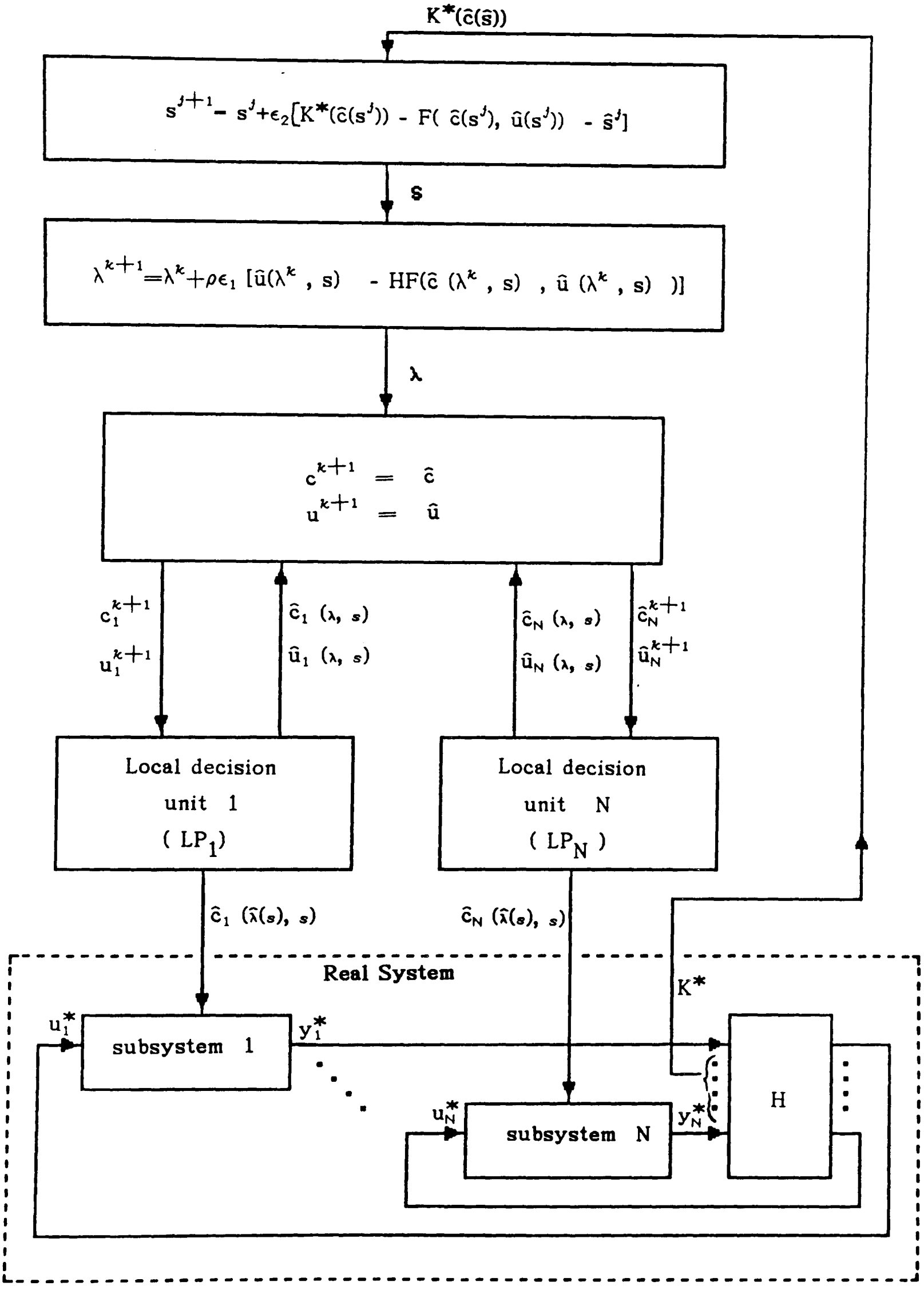


Figure (5. 2) Structure of Algorithm.

5.3 Coordination strategy of the algorithm

The upper level or coordinator has two tasks:

a) adjusting the multipliers λ

b) adjusting the shifting vector s.

1. To adjust the multiplier λ, the method of successive approximation is used (Luenberger 1984, Hestenes 1969) as an iterative process, according to the formula,

$$\lambda_{n+1} = \lambda_n + \rho_n h(x_n) \tag{5.21}$$

where $0 < \rho_n \leq \rho$ and ρ is a penalty coefficient.

Various rules can be given (Hestenes 1969) for selecting ρ_n. For example, we can choose $\rho_n = \gamma \rho$, where γ is a positive constant normally assuming a value ≤ 1. However, a simulation study is given later in this chapter, to show the effect of varing γ on the convergence of the algorithm.

The initial values of λ for the first iteration are obtained from the open loop model based solution of the problem (Findeisen et. al., 1980). The process for updating λ is repeated until the difference between λ_n and λ_{n+1} satisfies the required tolerance.

2. For adjusting the shift vector s, a simple Newton-type algorithm is used in the same manner described by Shao and Roberts (1983) (see for example Luenberger, 1984). The updating formula for s has the form

$$s^{j+1} = s^j - \epsilon_2 [R^{*'} (s^j)]^{-1} R^*(s^j) \tag{5.22}$$

where

$$R^*(s) = K^*(\hat{c} (s)) - F (\hat{c} (s) , \hat{u} (s)) - s \tag{5.23}$$

$$R^*(s) = F^*(\hat{c}(s),\ u^*(s)) - F(\hat{c}(s),\ \hat{u}(s)) - s \tag{5.24}$$

Then using the Taylor series expansion (Luenberger, 1984) we obtain

$$F^*(\hat{c}(s),\ u^*(s)) = F^*(\hat{c}(s),\ \hat{u}(s)) + F_u^{*'}[\,u^* - \hat{u}\,] \tag{5.25}$$

Substituting from (5.25) into (5.24), we obtain:

$$R^*(s) = F^*(\hat{c}(s), \hat{u}(s)) - F_u^{*'}[u^* - \hat{u}] - F(\hat{c}(s), \hat{u}(s)) - s \tag{5.26}$$

Since the real system mapping F^* is not known exactly, it is necessary to make the further approximation (Shao and Roberts, 1983) ;

$$F^* \simeq F$$
and
$$F^{*'} \simeq F' \tag{5.27}$$

substituting from (5.27) into (5.26), we obtain:

$$R^*(s) = F_u'\,[\,u^* - \hat{u}\,] - s \tag{5.28}$$

substituting by:

$$u^* = HK^*(\hat{c}(s))$$

$$\hat{u} = HF(\hat{c}(s),\ \hat{u}(s))$$

into equation (5.28), we obtain

$$R^*(s) = F_u'\,[\,HK^*(\hat{c}(s)) - H\,F(\hat{c}(s),\ \hat{u}(s))\,] - s \tag{5.29}$$

Equation (5.29) can be written as:

$$R^*(s) = F_u'\,[\,\{\,HK^*(\hat{c}(s)) - H\,F(\hat{c}(s),\ \hat{u}(s)) - Hs\,\} + Hs\,] - s \tag{5.30}$$

From equation (5.23), equation (5.30) can be written as:

$$R^*(s) = F'_u [HR^*(s) + Hs] - s \tag{5.31}$$

Hence:

$$R^*(s) = F'_u \; HR^*(s) + F'_u \; Hs \; - s$$

$$R^*(s) - F'_u \; HR^*(s) = F'_u \; Hs \; - s$$

$$[I - F'_u \; H] \; R^*(s) = - [I - F'_u \; H] \; s$$

and if $[I - F'_u \; H]$ is non-singular, we obtain:

$$R^*(s) = - s \tag{5.32}$$

Then,

$$R^{*'}(s) = - I \tag{5.33}$$

Hence, substituting equations (5.32) and (5.33) into (5.22), the updating formula for s can be written as:

$$s^{j+1} = s^j + \epsilon_2 [K^* (\hat{c}(s^j)) - F (\hat{c} (s^j) , \; \hat{u} (s^j)) \; - s^j] \tag{5.34}$$

where ϵ_2 is a step length coefficient.

5.4 Optimality conditions

A system augmented Lagrangian for the system optimisation problem can be formulated as follows:

$$\mathcal{L}^*(c, u, \lambda, \mu, \rho) = Q^*(c, u^*, F^*(c, u^*)) + \lambda^t(u^* - H F^*(c, u^*))$$

$$+ \mu^{*t} G^*(c, u^*, F^*(c, u^*))$$

$$+ 0.5\rho \| u^* - H F^*(c, u^*) \|^2 \qquad (5.35)$$

where λ, μ^* are Lagrange multiplier vectors and ρ is a penalty coefficient. The Kuhn-Tucker necessary optimality conditions (for example, Luenberger 1973) take the form:

$$\nabla_c \mathcal{L}^* = \frac{\partial^t Q^*}{\partial c} + \frac{\partial^t F^*}{\partial c}\frac{\partial^t Q^*}{\partial F^*} - \lambda^t H \frac{\partial F^*}{\partial c} + \mu^{*t}\left\{\frac{\partial^t G^*}{\partial c} + \frac{\partial^t F^*}{\partial c}\frac{\partial G^*}{\partial F^*}\right\}$$

$$- \rho H \frac{\partial^t F^*}{\partial c}(\overset{*}{u} - H \overset{*}{F}(c, \overset{*}{u})) = 0 \qquad (5.36)$$

$$\nabla_{u^*} \mathcal{L}^* = \frac{\partial^t Q^*}{\partial u^*} + \frac{\partial^t F^*}{\partial u^*}\frac{\partial^t Q^*}{\partial F^*} + \lambda^t - \lambda^t H \frac{\partial F^*}{\partial u^*} + \mu^{*t}\left\{\frac{\partial^t G^*}{\partial u^*} + \frac{\partial^t F^*}{\partial u^*}\frac{\partial G^*}{\partial F^*}\right\}$$

$$+ \rho(I - H \frac{\partial F^*}{\partial u})^t (\overset{*}{u} - H F^*(c, \overset{*}{u})) = 0 \qquad (5.37)$$

$$\nabla_\lambda \mathcal{L}^* = (u^* - HF^*(c, \overset{*}{u})) = 0 \qquad (5.38)$$

$$\nabla_{\mu^*} \mathcal{L}^* = G^*(c, u^*, F^*(c, u^*)) = 0 \qquad (5.39)$$

$$\nabla_\rho \mathcal{L}^* = 0.5 \| \overset{*}{u} - H F^*(c, u) \|^2 = 0 \qquad (5.40)$$

and,

$$\mu^*_j = 0 \quad \text{if} \quad G^*_j < 0$$

$$\mu^*_j > 0 \quad \text{if} \quad G^*_j = 0 \qquad (5.41)$$

The model augmented Lagrangian analogous to equation (5.35) can be formed as follows:

$$\mathcal{L}(c\ ,\ u\ ,\ \lambda\ ,\ \mu\ ,\ \rho) = Q\ (\ c\ ,\ u + Hs\ ,\ F(\ c\ ,\ u\ ) + s\ )$$

$$+\lambda^t\ (u - H\ F(c\ ,u)) + \mu^t G\ (c,\ u + Hs,\ F(c\ ,u\ ) + s)$$

$$+ 0.5\ \rho\ \|\ u\ - H\ F(\ c\ ,\ u)\ \|^2 \qquad (5.42)$$

The Kuhn-Tucker necessary optimality conditions are:

$$\nabla_c\ L = \frac{\partial^t Q}{\partial c} + \frac{\partial^t F}{\partial c}\ \frac{\partial^t Q}{\partial F}\ - \lambda^t H\ \frac{\partial\ F}{\partial c} + \mu^t \left\{ \frac{\partial^t G}{\partial c} + \frac{\partial^t F}{\partial c}\ \frac{\partial G}{\partial F} \right\}$$

$$- \rho\ H\ \frac{\partial^t F}{\partial c}\ (\ u - HF(c\ ,\ u) = 0 \qquad (5.43)$$

$$\nabla_u\ L = \frac{\partial^t Q}{\partial u} + \frac{\partial^t F}{\partial u}\ \frac{\partial^t Q}{\partial F} + \lambda^t - \lambda^t H\ \frac{\partial\ F}{\partial u} + \mu^t \left\{ \frac{\partial^t G}{\partial u} + \frac{\partial^t F}{\partial u}\ \frac{\partial G}{\partial F} \right\}$$

$$+ \rho\ (\ I - H\ \frac{\partial F}{\partial u}\ )^t\ (\ u - HF(c\ ,\ u) = 0 \qquad (5.44)$$

$$\nabla_\lambda L = \ u - HF\ (c\ ,\ u\ ) = 0 \qquad (5.45)$$

$$\nabla_\mu\ L = \ G\ (\ c\ ,\ \ u\ ,\ \ F\ (c\ ,\ u) + s) = 0 \qquad (5.46)$$

$$\nabla_\rho\ L = \ 0.5\ \|\ u\ - H\ F(\ c\ ,\ u)\ \|^2 = 0 \qquad (5.47)$$

$$\nabla_s\ L = \ \frac{\partial^t Q}{\partial s}\ + \mu^t\ \frac{\partial G}{\partial s} = 0 \qquad (5.48)$$

and

$$\mu_j = 0 \qquad \text{if} \quad G_j < 0$$

$$\mu_j > 0 \qquad \text{if} \quad G_j = 0 \qquad (5.49)$$

Comparing between the model equations (5.43)—(5.49) and the system equations (5.36)—(5.41) , we can derive the conditions to ensure that the model will produce optimum results if they are achieved, i.e. if the following conditions apply:

$$\left.\begin{aligned} F &= F^* \\ u &= u^* \\ s &= 0 \end{aligned}\right\} \qquad (5.50)$$

If conditions (5.50) apply, there will be agreement between the model and system constraints and it ensures that the respective multipliers are in agreement. That is:

$$\mu = \mu^*$$

And if there is derivative matching, there will be full agreement between the model and system optimality conditions:

$$\text{i)} \quad \frac{\partial F}{\partial c} = \frac{\partial F^*}{\partial c}$$

$$\text{ii)} \quad \frac{\partial F}{\partial u} = \frac{\partial F^*}{\partial u^*} \qquad (5.51)$$

$$\text{iii)} \quad F = F^*$$

$$\text{iv)} \quad u = u^*$$

$$\text{v)} \quad s = 0$$

Thus, if the sufficient conditions of equation (5.51) are fulfilled, the results of the model based optimisation problem will provide the system optimum. With looking at the equation (5.51), these conditions can be satisfied, only by using a perfect model. However, these conditions are difficult to satisfy for the following reasons:

1- Since the vector s is mainly required in this algorithm to shift the model outputs in order to match the system outputs, therefore, the last condition s = 0, can only be achieved within a desired tolerance.

2- It is often impossible to obtain a model which is a perfect representation of the system. Therefore, this algorithm will remain suboptimal since the conditions of optimality can not be achieved.

5.5 Simulation study

5.5.1 Simulation examples

Example 1:

This example is a convex problem consisting of two interconnected subsystems, with interconnection equations:

$$u_1 = y_2$$
$$u_2 = y_1$$

and an interconnection matrix H of the form:

$$H = \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix}$$

The model and reality equations of the system are respectively:

$$y_1 = 2\ c_1 + u_1$$

$$y_2 = 0.5\ c_2 + 0.5\ u_2$$

and

$$y_1^* = 2.1\ c_1 + u_1 + \beta\ c_1 u_1$$

$$y_2^* = 0.6\ c_2 + 0.55\ u_2$$

where $\beta = [\ 0\ ,\ 1]$

The performance index is:

$$Q = 32\ c_1^2 - 16\ c_1 + (2c_1 + u_1 - 1) + 10\ c_2^2 + 4\ c_2\ u_2 + 0.5(\ 2\ c_2 + 2u_2 - 4)^2$$

subject to the inequality constraints:

$$2\ c_1 + u_1 \leq 1$$

Example 2:

This example consists of two interconnected subsystems. The interconnection matrix, model and reality equations of the system are the same as for examle 1. The performance index of the system and subjected constraints are modified for the problem to be non-convex, as follows:

$$Q = 32\ c_1^2 - 16\ c_1 + (2c_1 + u_1 - 1) + 10\ c_2^2 + 4\ c_2\ u_2 - 0.5(\ 2\ c_2 + 2u_2)^2$$

subject to the inequality constraints:

$$2\ c_1 + u_1 \leq 2.25$$

Example 3:

This example consists of two subsystems where the constriants involve system outputs. The model and reality equations and performance indices are:

$$y_{11} = F_{11}(c_1\ ,\ u_1) = c_{11} - c_{12} + 2\ u_{11}$$

$$y_{21} = F_{21}(c_2\ ,\ u_2) = c_{21} - c_{22} + u_{21}$$

$$y_{22} = F_{22}(c_2\ ,\ u_2) = 2\ c_{22} - c_{23} + u_{21}$$

$$y_{11}^* = F_{11}^*(c_1\ ,\ u_1) = 1.4\ c_{11} - 0.6\ c_{12} + 1.8\ u_{11}$$

$$y_{21}^* = F_{21}^*(c_2\ ,\ u_2) = 1.3\ c_{21} - 1.1\ c_{22} + 1.1\ u_{21}$$

$$y_{22}^* = F_{22}^*(c_2\ ,\ u_2) = 2.3\ c_{22} - 0.7\ c_{23} - 1.1\ u_{21}$$

$$Q_1 (c_1 , u_1 , y_1) = (y_{11} - 1)^2 + c_{11}^2 + c_{12}^2$$

$$Q_2 (c_2 , u_2 , y_2) = 2 (y_{21} -2)^2 + (y_{22} -3)^2 + c_{21}^2 + c_{22}^2 + c_{23}^2$$

The system constraints are:

$$\mathcal{CUY}_1 \triangleq \{ (c_1 , u_1 , y_1) \in \Re^4 \mid | c_{11} | \leq 1 , \ | c_{12} | \leq 1 , \ y_{11} \geq 0,$$

$$((0.8 - c_{12} - 0.6 u_{11}) \geq 0 \}$$

$$\mathcal{CUY}_2 \triangleq \{ (c_2 , u_2 , y_2) \in \Re^6 \mid |c_{21}| \leq 1, \quad |c_{22}| \leq 1 , |c_{23}| \leq 1,$$

$$y_{21} \geq 0, \quad y_{22} \geq 0 , \quad (2.04 + 1.05 u_{21}^2 - c_{21}^2 - c_{22}^2 - c_{23}^2) \geq 0 \}$$

where:

$c_1 = (c_{11} , c_{12})$ $\qquad$ $c_2 = (c_{21}, c_{22}, c_{23})$

$u_1 = u_{11}$ $\qquad$ $u_2 = u_{21}$

$y_1 = (y_{11})$ $\qquad$ $y_2 = (y_{21}, y_{22})$

Finally, the coupling equation is:

$$\begin{bmatrix} u_{11} \\ u_{21} \end{bmatrix} = \begin{bmatrix} 0 & 1 & 0 \\ 1 & 0 & 0 \end{bmatrix} \begin{bmatrix} y_{11} \\ y_{21} \\ y_{22} \end{bmatrix}$$

Example 4:

This example involves three interconnected subsystems. The real and model equations of the system are:

$$y_{11}^{*} = F_{11}^{*}(c_1, u_1) = 1.3c_{11} - c_{12} + 2\,u_{11} + 0.15\,c_{11}\,u_{11}$$

$$y_{21}^{*} = F_{21}^{*}(c_2, u_2) = c_{21} - c_{22} + 1.2\,u_{21} - 3\,u_{22} + 0.1\,c_{22}^{2}$$

$$y_{22}^{*} = F_{22}^{*}(c_2, u_2) = 2\,c_{22} - 1.25\,c_{23} - u_{21} + u_{22} + 0.25\,c_{23}\,c_{22} + 0.1$$

$$y_{31}^{*} = F_{31}^{*}(c_3, u_3) = 0.8\,c_{31} - 2.5\,c_{32} - 4.2\,u_{11}$$

$$y_{11} = F_{11}(c_1, u_1) = c_{11} - c_{12} + 2\,u_{11}$$

$$y_{21} = F_{21}(c_2, u_2) = c_{21} - c_{22} + u_{21} - 3\,u_{22}$$

$$y_{22} = F_{22}(c_2, u_2) = 2\,c_{22} - c_{23} - u_{21} + u_{22}$$

$$y_{31} = F_{31}(c_3, u_3) = c_{31} + 2.5\,c_{32} - 4\,u_{31}$$

where:

$c_1 = (c_{11}, c_{12})$ $\qquad$ $c_2 = (c_{21}, c_{22}, c_{23})$ $\qquad$ $c_3 = (c_{31}, c_{32})$

$u_1 = u_{11}$ $\qquad$ $u_2 = (u_{21}, u_{22})$ $\qquad$ $u_3 = u_{31}$

$y_1 = y_{11}$ $\qquad$ $y_2 = (y_{21}, y_{22})$ $\qquad$ $y_3 = y_{31}$

The subsystem performance indices are:

$$Q_1(c_1, u_1) = (u_{11} - 1)^4 + 5(c_{11} + c_{12} - 2)^2$$

$$Q_2(c_2, u_2) = 2(c_{21} - 2)^2 + c_{22}^2 + 3c_{23}^2 + 4u_{21}^2 + u_{22}^2$$

$$Q_3(c_3, u_3) = (c_{31} + 1)^2 + (u_{31} - 1)^2 + 2.5\,c_{32}^2$$

The system constraints are:

$$\mathcal{CU}_1 \triangleq \{(c_1, u_1) \in \Re^3 \mid c_{11}^2 + c_{12}^2 \leq 1 \;,\; 0 \leq u_{11} \leq 0\}$$

$$\mathcal{CU}_2 \triangleq \left\{ (c_2, u_2) \in \Re^5 \mid \quad 0.5\, c_{21} + c_{22} + 2\, c_{23} \mid \leq 1 , \right.$$

$$\left. 4c_{21}^2 + 2\, c_{21}\, u_{21} + 0.4u_{21} + c_{21}\, c_{23} + 0.5\, c_{23}^2 + u_{21}^2 \leq 4 \right\}$$

$$\mathcal{CU}_3 \triangleq \left\{ (c_3, u_3) \in \Re^3 \mid \quad c_{31} + u_{31} + 0.5 \geq 0 \quad , \; 0 \leq c_{32} \leq 1 \right\}$$

Finally, the coupling equation is:

$$\begin{bmatrix} u_{11} \\ u_{21} \\ u_{22} \\ u_{31} \end{bmatrix} = \begin{bmatrix} 0 & 1 & 0 & 0 \\ 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} y_{11} \\ y_{21} \\ y_{22} \\ y_{31} \end{bmatrix}$$

Example 5:

This final example is modified to be a non-convex problem (Stoilov, 1980). The model equations, performance indices, system constraints and coupling equations are the same as in example 4. The system equations are as follows:

$$y_{11}^* = 1.3c_{11} - c_{12} + 2\, u_{11} + 0.15\, c_{11}\, u_{11}$$

$$y_{21}^* = c_{21} - c_{22} + 1.2\, u_{21} - 3\, u_{22} + 0.1\, c_{22}^2$$

$$y_{22}^* = 2\, c_{22} - 1.25\, c_{23} - u_{21} + u_{22} + 0.25\, c_{22}\, c_{23} + 0.1\, u_{21}\, u_{22}$$

$$y_{31}^* = 0.9\, c_{31} + 2.5\, c_{32} - 4.2\, u_{31}$$

5.5.2 Simulation Results

The simulation work was performed on a Prime 750 computer and the local optimisation problems were solved using the NAG Library subroutine E04UAF which uses the augmented sequential Lagrangian technique (see Chapter 3).

The simulation results show that the algorithm is applicable to a wide class of non convex problems. It is also useful for convex problems due to the reduced number of controller set-points changes which results in reducing on-line control correction time.

A comprehensive study of the algorithm is carried out to investigate the effects of non linearity and penalty coefficient on convergence and to investigate the sensitivity of the algorithm to various gains. Five numerical examples were used to test the applicability of the algorithm and to investigate the effects of the various factors on the algorithm's convergence. Table 5.1. shows the obtained simulation results for the simulation examples. In the first two columns the gains ϵ_1, ϵ_2 are shown which are used for updating the price vector λ and the shift vector s in intermediate and outer loops respectively.

The tolerance for the various loops for the inner, intermediate and outer loop iterations Tol_1, Tol_2, Tol are presented in the next three columns. The following column shows the penalty coefficient ρ.

The simulation results are presented in the last four columns; where IS is the number of system set-points changes, and IT is the total number of model based optimisations.

From an extensive simulation study, it is evident that the algorithm is applicable to both convex and non convex problems. It is also applicable to problems with output dependent inequality constraints (Example 3).

Representative results obtained from the algorithm, applied to example 2, to study the sensitivity of the algorithm for all combinations of ϵ_1 and ϵ_2, and are presented in Tables 5.2. and 5.3. The step length of the intermediate and outer loops, ϵ_1 and ϵ_2, which were used in the formula to update λ and s respectively were studied over the range [0.1, 1.0] for different values of penalty coefficient, ρ = 5, 20, 40 and 100.

It is generally observed that the algorithm is less sensitive to the choice of the step length of the outer loop ϵ_2. Exceptionally, when the value of step length ϵ_2 is approaching 1.0, there is a small difference of one or two iterations, as is shown in Figure (5.3). Concerning the choice of the step length ϵ_1, it can be remarked that the higher the value of step length ϵ_1, the lower the number of iterations will be achieved. This is observed for the used values of penalty coeffiecients ρ = 100, 40 and 20 as shown in Figure (5.4.).

Example	ϵ_1	ϵ_2	Tol_1	Tol_2	Tol	ρ	IS	IT	Real Perform.	Real Optimum
1	0.5	0.5	10^{-3}	10^{-2}	10^{-2}	16	6	19	-0.00570	-0.068839
	0.95	0.5	10^{-4}	10^{-3}	10^{-3}	16	7	30	0.03140	
	0.95	0.7	10^{-5}	10^{-4}	10^{-3}	16	4	37	0.03633	
2	0.5	0.5	10^{-4}	10^{-3}	10^{-2}	32	5	155	-9.38003	-9.550154
	0.9	0.7	10^{-4}	10^{-3}	10^{-3}	32	7	103	-9.35289	
	1.0	1.0	10^{-5}	10^{-4}	10^{-3}	16	4	142	-9.35189	
3	1.0	1.0	10^{-5}	10^{-4}	10^{-3}	120	4	91	6.03335	5.9826
	1.0	1.0	10^{-5}	10^{-4}	10^{-3}	100	4	109	6.04031	
	1.0	0.9	10^{-5}	10^{-4}	10^{-3}	80	4	140	6.04028	
4	0.5	0.5	10^{-4}	10^{-3}	10^{-3}	5	5	47	6.33367	6.326566
	0.5	0.5	10^{-4}	10^{-3}	10^{-3}	6	5	72	6.33204	
	0.5	0.5	10^{-4}	10^{-3}	10^{-3}	40	5	312	6.34416	
5	0.5	0.7	10^{-4}	10^{-3}	10^{-3}	5	3	42	6.37062	6.314206
	0.5	0.5	10^{-4}	10^{-3}	10^{-3}	5	4	53	6.33203	
	0.5	0.5	10^{-4}	10^{-3}	10^{-3}	30	3	309	6.39479	

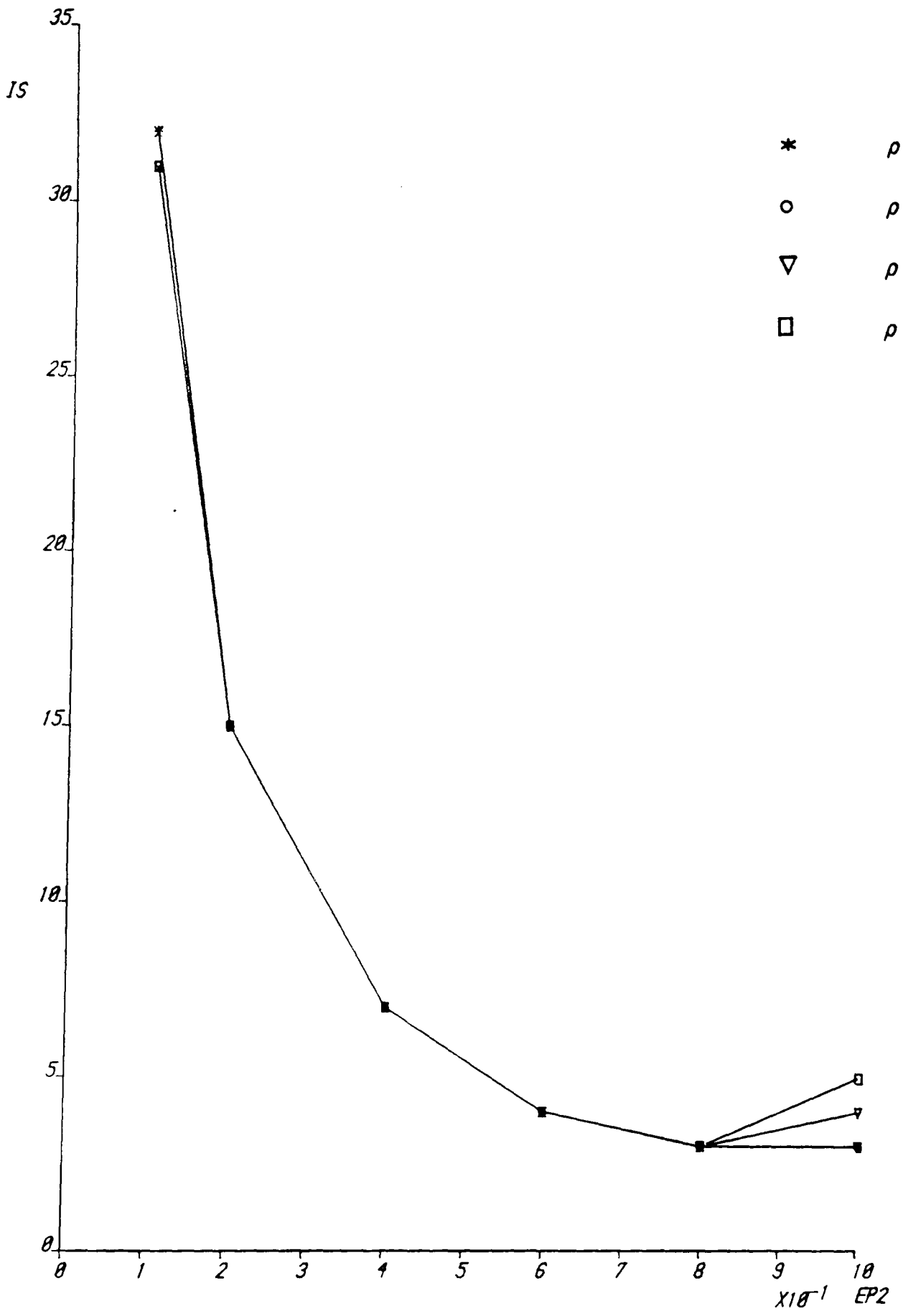
Table (5.1) Simulation Results .

ϵ_1	$\rho = 5$		$\rho = 20$		$\rho = 40$		$\rho = 100$	
	IT	IS	IT	IS	IT	IS	IT	IS
0.1	1208	6	820	6	720	5	564	5
0.2	755	6	449	6	390	5	295	5
0.3	557	6	319	6	232	5	200	5
0.4	438	6	235	6	193	5	144	5
0.5	366	6	190	6	149	5	118	5
0.6	304	6	155	6	132	5	94	5
0.7	843	6	140	6	107	5	81	5
0.8	DIVERG.		121	6	97	5	86	5
0.9	DIVERG.		113	6	79	5	63	5
1.0	DIVERG.		98	6	66	5	50	5

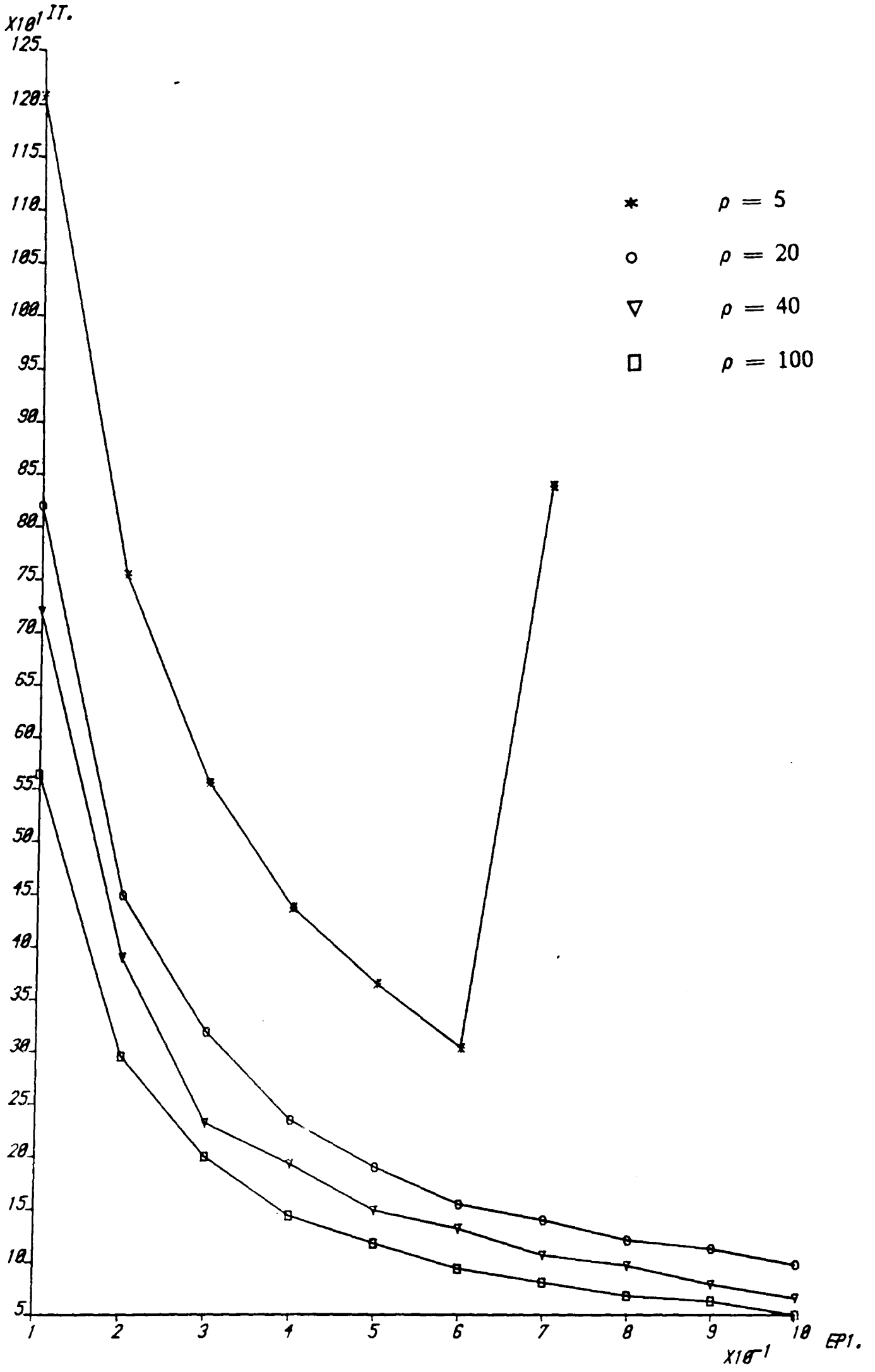
Table (5.2) Sensitivity of the algorithm to ϵ_1 (ϵ_2 = 0.5), Example 2.

ϵ_2	$\rho = 5$		$\rho = 20$		$\rho = 40$		$\rho = 100$	
	IT	IS	IT	IS	IT	IS	IT	IS
0.1	1010	32	449	31	336	31	269	31
0.2	693	15	304	15	263	15	214	15
0.4	396	7	210	7	189	7	149	7
0.6	270	4	141	4	129	4	97	4
0.8	194	3	106	3	97	3	77	3
1.0	220	3	124	3	125	4	128	5

Table (5.3) Sensitivity of the algorithm to ϵ_2 ($\epsilon_1 = 0.5$), Example 2.



FIGURE(5.3) SENSITIVITY OF ALGORITHM TO EP2 (ε_2), EX. 2



FIGURE(5.4) SENSITIVITY OF ALGORITHM TO ε_1 FOR DIFFERENT VALUES OF ρ (EXAMPLE 2)

However, for ρ = 5, the total number of iterations is decreased as the step length ϵ_1 is increased until ϵ_1 = 0.7, then increasing ϵ_1 any further will cause divergence.

The effect of the penalty coefficient ρ on convergence of the algorithm has been studied. Examples 4 and 5 were used in this study. Figures (5.5.) and (5.7.) represent the effect of varying the penalty coefficient on the number of system iterations for examples 4 and 5 respectively, while Figures (5.6.) and (5.8.) show the effect of varying the penalty coefficient on the total number of iterations for both examples. It seems evident that the number of system iterations IS not much affected by varying ρ, where IS remains at approximately 5 iterations for example 4 and 4 iterations for example 5 over the range $0 < \rho \leq 50$. However, choosing the value of ρ affects the total number of iterations, where the minimum value of IT may be achieved in the range $4 \leq \rho \leq 10$ for example 4 and in the range $4 \leq \rho \leq 8$ for example 5.

Finally, a simulation study of the algorithm to investigate the effects of non linearity is illustrated by Figure (5.9.). Example 2 was used, where a non linear term is introduced to F_1^* and various degrees of non linearity were investigated. That is with:

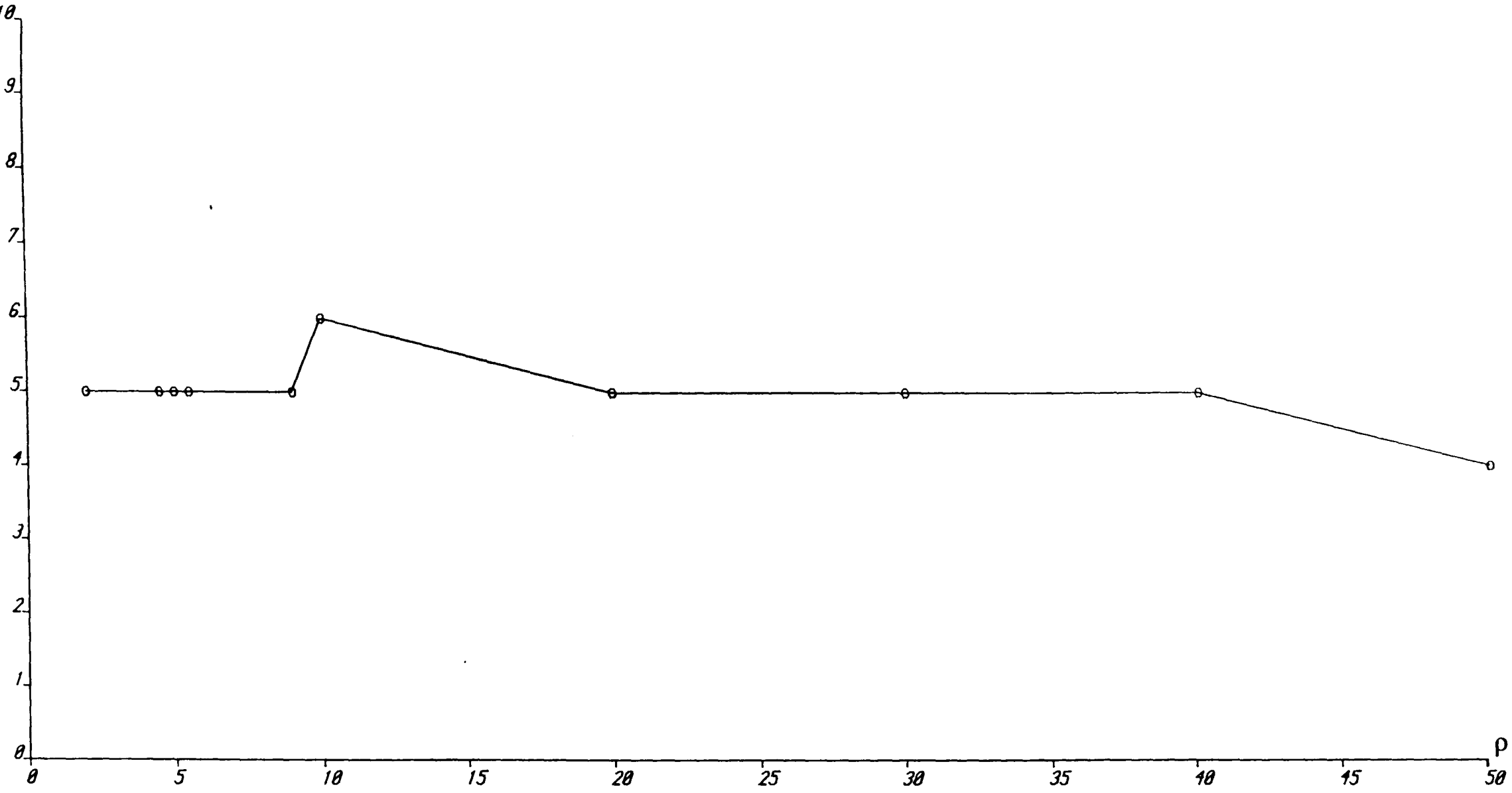
$$F_1^* (c, u) = 2.1 \; c_1 + u_1 + \beta \; c_1 \; u_1$$

Where $\beta \geq 0$ is a parameter.

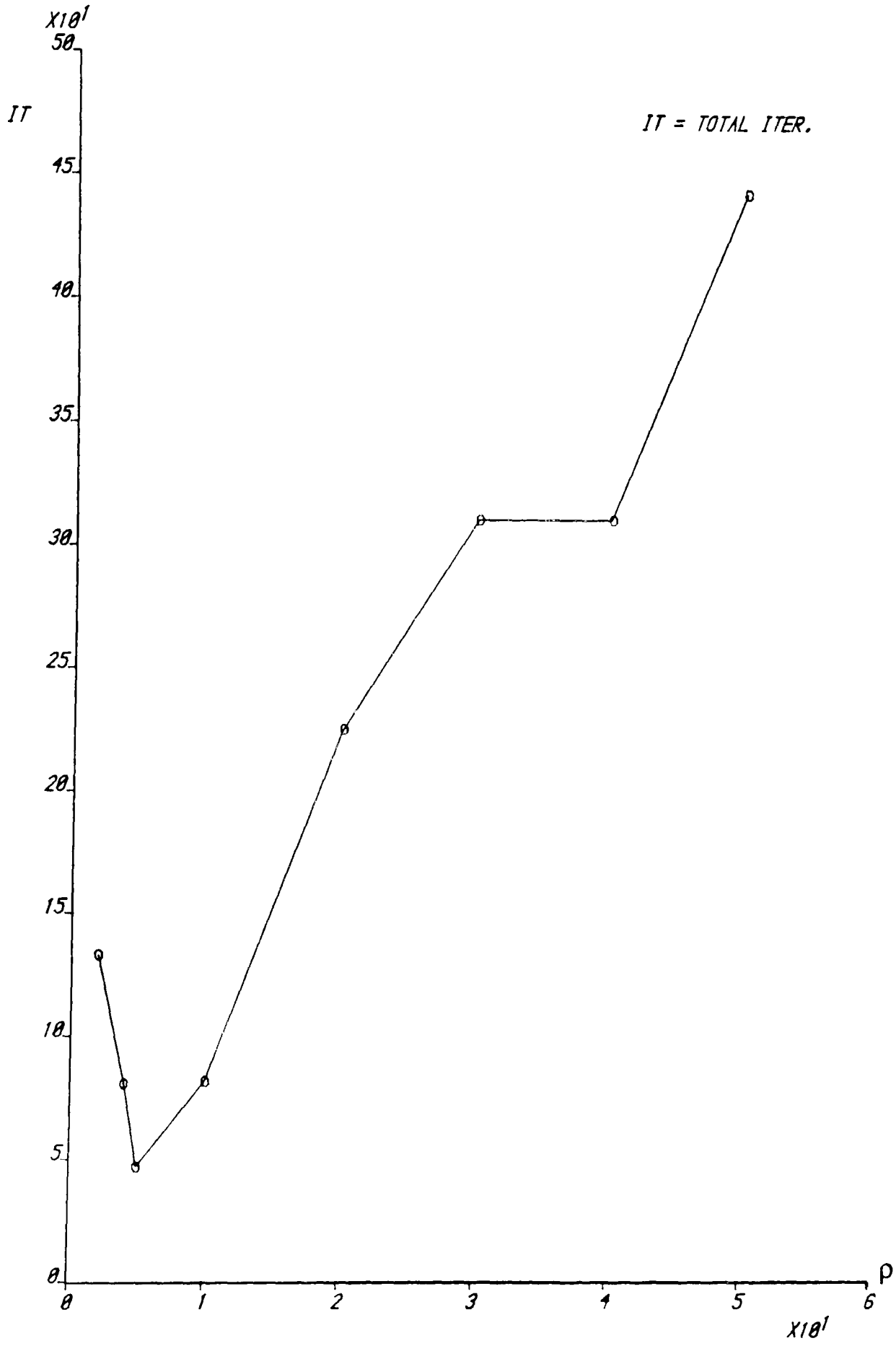
Results for $\beta \in [0, 0.75]$ are shown in Figure (5.9.), for ρ = 32. Over that range, the algorithm is shown to perform satisfactory, within a range of 3 to 5 set point changes to achieve the various loop's accuracies.

A selective simualtion comparative study between the Augmented Price Correction Mechanism (APCM) which is described in this chapter and the previous correction mechanism (Shao V1, Shao V2) is presented in Table (5.4) for two convex examples; 3 and 4. The study shows that the present algorithm is applicable on convex and non-convex problems, and in addition

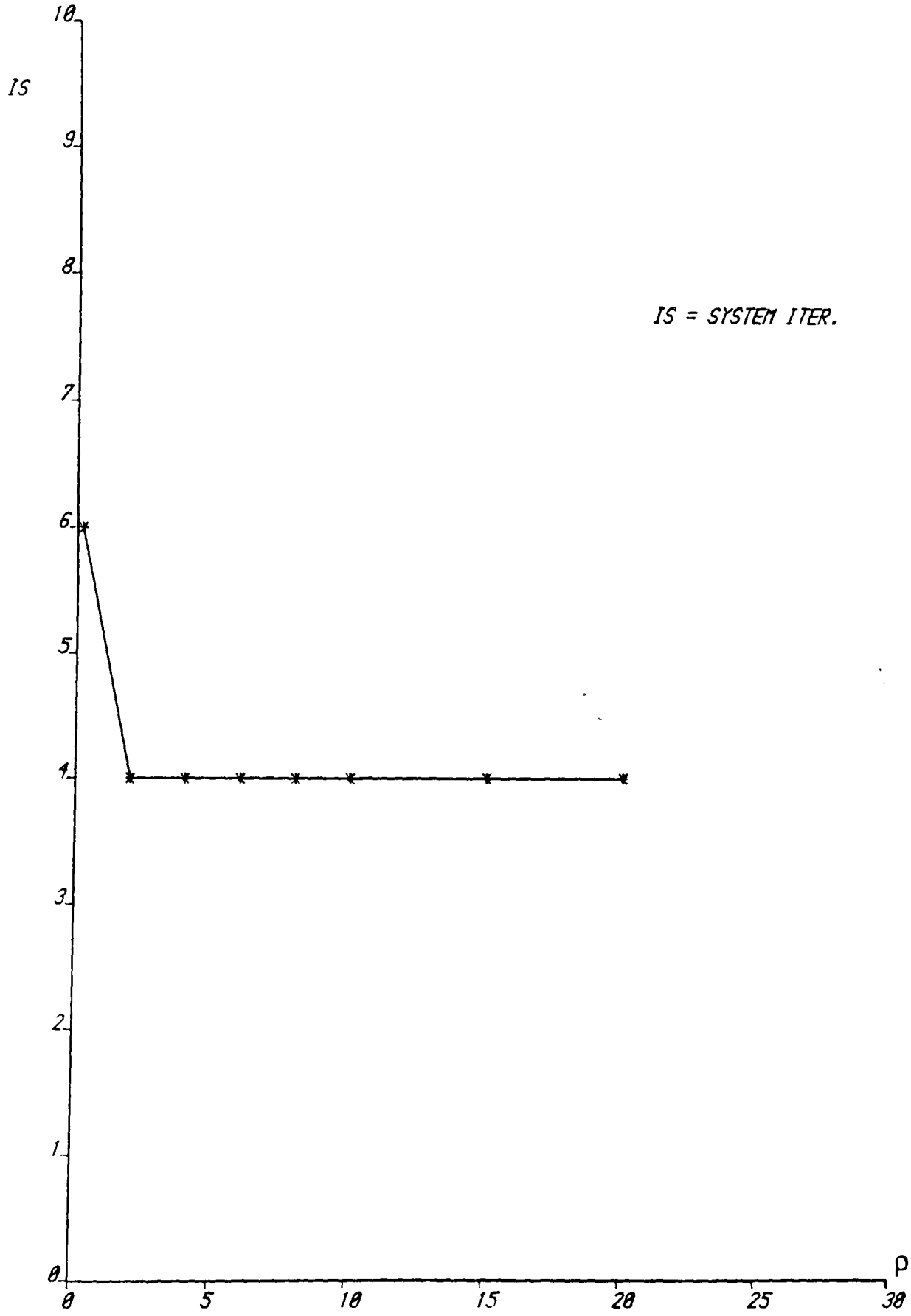
it reduces the number of system iterations as well as the total number of iterations for both examples. However, Shao's technique V1 provides a final converged solution closer to the real optimum than the present algorithm. Table (5.5) shows another simulation study to compare the Augmented Price Correction Mechanism with the Augmented Interaction Balance Method with feedback (AIBMF) by Tatjewski, 1985. The results shown in table (5.5) are obtained from a variety of problems; where example 1 is convex but example 2 is non-convex. Both examples are taken from the paper by Tatjweski, 1985 while example 3 is taken from the work by Shao and Roberts, 1983. It is convex, but its performance indices and constraints involve system outputs. From that study, it seems evident that augmented price correction mechanism (APCM) achieves the converged solution to the required tolerance using a number of set point changes less than AIBMF at the expense of increasing the total number of iterations. However the converged results of AIBMF is closer to the optimum.



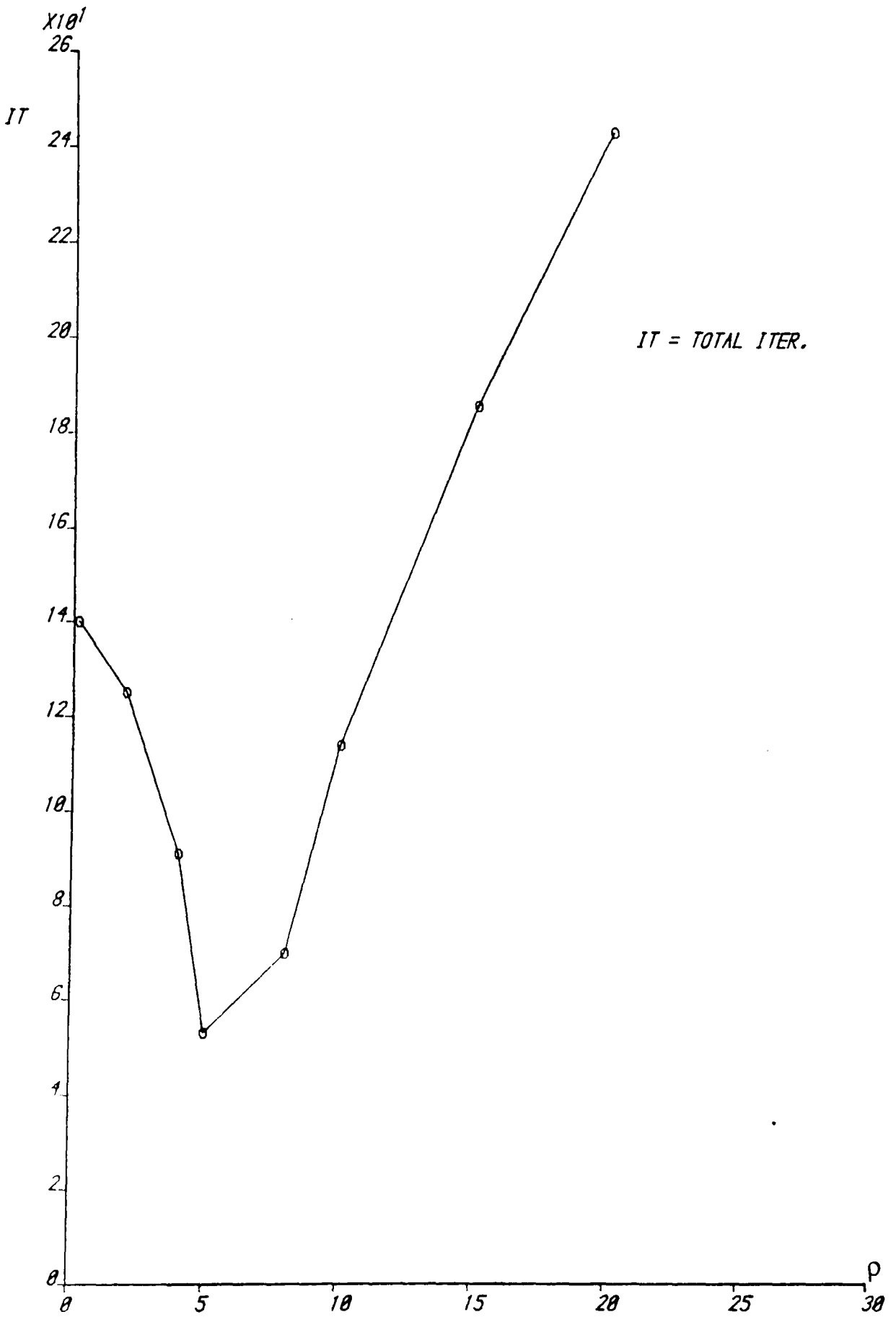
FIGURE(5.5) EFFECT OF PENALITY COEFFICIENT ON CONVERGENCE OF SYSTEM ITER., EX.4



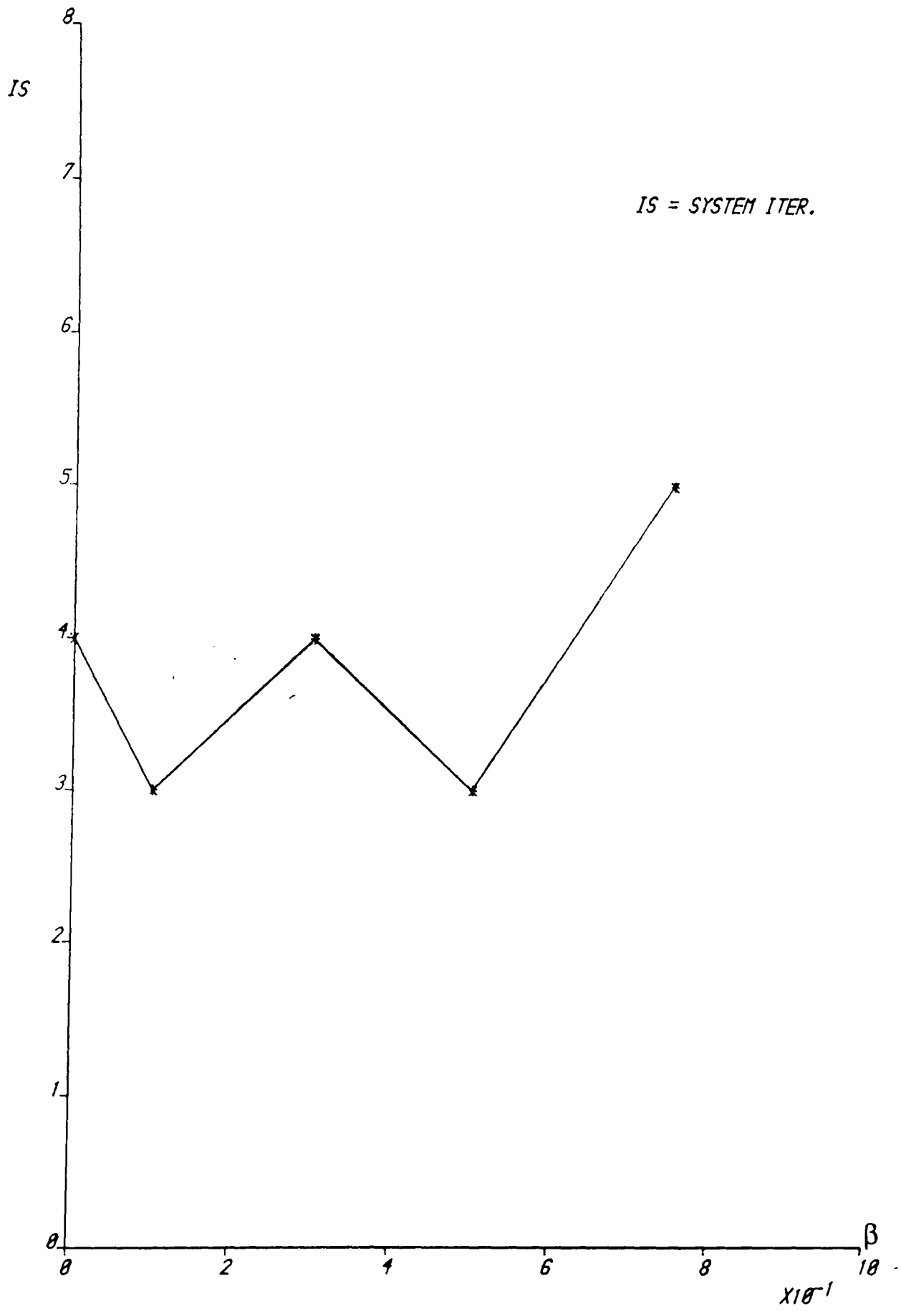
FIGURE(5.6) EFFECT OF PENALITY COEFFICIENT ON CONVERGENCE OF IT, EX. 4



FIGURE(5.7) EFFECT OF PENALITY COEFFICIENT ON CONVERGENCE OF IS, EX. 5



FIGURE(5.8) EFFECT OF PENALITY COEFFICIENT ON CONVERGENCE OF IT, EX. 5



FIGURE(5.9) EFFECT OF NONLINEARITY ON CONVERGENCE OF IS, EX. 2(ρ =32)

Algorithm	ϵ_1	ϵ_2	Tol_1	Tol_2	Tol	P	IS	IT	Real Perform.	Real Optimum
Present	1.0	1.0	10^{-5}	10^{-4}	10^{-3}	120	4	91	6.03335	
Shao V1	1.0	1.0	10^{-5}	-	10^{-3}	-	6	168	5.98438	5.9261
Shao V2	1.0	1.0	10^{-5}	-	10^{-3}	-	5	144	6.04314	
Present	0.5	0.5	10^{-4}	10^{-3}	10^{-3}	5	5	47	6.33187	
Shao V1	0.2	0.6	10^{-4}	-	10^{-3}	-	6	192	6.33560	6.3261
Shao V2	0.2	0.6	10^{-4}	-	10^{-3}	-	6	190	6.33555	

Table (5.4) Comparison of simulation results for examples 3 & 4.

Algorithm	Example	Tol_1	Tol_2	Tol	ϵ_1	ϵ_2	P	IS	IT	Real Perform.	Real Optimum
Present	1	10^{-5}	10^{-4}	10^{-4}	1.0	1.0	16	6	41	6.03335	-0.0692
AIBMF	1	10^{-5}	-	10^{-4}	-	1.0	32	9	26	5.98438	
Present	2	10^{-5}	10^{-4}	10^{-3}	1.0	1.0	16	4	142	-9.35189	-9.5502
AIBMF	2	10^{-5}	-	10^{-3}	-	1.0	32	7	121	-9.54818	
Present	3	10^{-5}	10^{-4}	10^{-3}	1.0	1.0	120	4	91	6.03335	5.9261
AIBMF	3	10^{-5}	-	10^{-3}	-	1.0	5	9	44	5.98431	

Table (5.5) Comparison of simulation results for examples 1,2 and 3.

5.6 Summary

In this chapter, an algorithm is proposed to extend the applicability of the previous price correction mechanism, by Shao and Roberts, 1983, to cover the case of non-convex problems. It is based on the augmented Lagrangian and shifting of model outputs.

The algorithm is tested using five simulation examples of various types; where convex and non-convex problems were used and also those which involve system outputs. The study shows that the algorithm is applicable on all those types of problems. It is also shown that it reduces the number of system set point changes as well as model based optimisation iterations if it is compared with the previous price correction mechanism.

Comparing this algorithm with the augmented IBMF (Tatjewski, 1985) algorithm shows that using the augmented PCM algorithm, the converged solution is achieved to the required tolerance with a number of set point changes less than that with the augmented IBMF algorithm but with a less degree of optimality.

Chapter 6

Integrated System Optimisation and Parameter Estimation (ISOPE)

6.1 Introduction:

A common method for solving an on-line optimisation problem is to use a steady state mathematical model of the real process (Yole and Duncanson, 1970). In general, the model will not be a faithful representation of the real system for different reasons such as:

1- The model might be deliberately chosen simple to avoid complications of the optimisation problem.
2- All the real process informations are not available.
3- There might be some factors in practical applications which are difficult to represent accurately in the model.

The differences between model and reality give rise to a suboptimal solution in the optimisation problem. This may be overcome by using output feedback from the process to update the model parameters. A common technique is to employ a two-step approach in which the system optimisation problem and parameter estimation problem are solved iteratively until convergence is obtained. Even though the model parameters are updated, as in the Standard Two-Step approach (Roberts, 1977; 1978), it still does not guarantee that the solution of the optimisation problem will be optimal. This is because when the model is inaccurate, the derivatives of the real process outputs with respect to the controller set point values are not exactly matched with the

corresponding derivatives in the model (Durbeck, 1965; Foord,1974).

The Modified Two-Step (MTS) algorithm was proposed by Roberts (1979). In this algorithm, the interactions between the system optimisation and parameter estimation problems have been taken into account, where an extra term is introduced to the optimisation performance index to allow for any mismatch between model and real process output derivatives. In spite of model-reality differences, the MTS algorithm converges to the correct optimum. The algorithm has been successfully applied to solve many example problems (Roberts and Roussias, 1980; Roberts and William, 1981; Ellis and Roberts, 1981, 1985; Stevenson, Brdyś and Roberts, 1985).

Originally, these algorithms were developed for centralised optimisation without decomposition. They have been extended (Brdyś and Roberts, 1986; Ellis, Michalska and Roberts, 1984; Michalska, Ellis and Roberts, 1985) to be applied to interconnected large scale systems. This was achieved by combining the two-step approach with the price technique, and this combined method is known as the joint coordination method.

Brdyś and Roberts (1987) gave optimality and global convergence conditions of the algorithm in the situation where the system inequality constraints do not depend on the outputs. These optimality conditions are readily valid for general inequality constraints (Brdyś and Roberts, 1986; Chen, 1986).

The modified two-step approach has been extended by Lin , Chen and Roberts (1986) to take into account general inequality constraints. The resulting algorithm has a similar structure to that given by Brdyś, Chen and Roberts (1986). Lin, Hendawy and Roberts (1988a) modified the algorithm proposed by Lin, Chen and Roberts (1986) to extend its applicability to interconnected large scale systems. The resulting algorithm is a combination of the algorithm by Lin, Chen and Roberts (1986) and the price method. The technique uses an output feedback structure and a single loop iteration strategy. The algorithm is described in detail in chapter 7.

6.2 Centralised ISOPE Techniques:

In this section we will be looking at methods of determining the optimal set points of the process feedback controllers using centralised integrated system optimisation and parameter estimation.

This means that the whole system is treated in its entirety where all the information is available to a single decision unit. Kambhampati (1988) has listed most of the available ISOPE techniques but the selected algorithms to be examined here are:

i- The standard two-step approach,

ii- The modified two-step approach,

iii- The approximate linear model approach.

Consider the optimising control problem for the centralised case as follows:

$$\min_{c,y} \; Q(c\ ,\ y)$$

s.t: (6.1)

$$y = F^{*}(c)$$

$$G(c\ ,\ y) \leq 0$$

where c and y are the vectors of controller set points and real process outputs respectively. The performance index function is described by Q, and G is the inequality constraints imposed on the system. F^{*} is the process input-output mapping which is usually assumed to be unknown. The approximate model input-output mapping F will be used instead of F^{*}, where:

$$y = F(c\ ,\ \alpha) \tag{6.2}$$

where α is a vector of parameters of the process model.

The model equation (6.2) is assumed to be point parametric on $\mathcal{C}$ (Brdyś, 1983); that is for every $c \in \mathcal{C}$ there is an $\alpha(c)$ such that:

$$F^*(c) = F(c\ ,\ \alpha(c))$$

and that mappings F, F^*, Q, G are twice continuously differentiable.

6.2.1 The standard two-step approach:

The standard two-step approach considers the integrated problem as two separate problems to be solved repeatedly; these are optimisation and parameter estimation. The optimisation control problem can be written in an equivalent form by substituting from equation (6.2) in (6.1) as follows:

$$\min_{c,\alpha}\ q(c\ ,\ \alpha)$$

$$\text{s.t:}\quad F(c\ ,\ \alpha) = F^*(c) \qquad (6.3)$$
$$g(c\ ,\ \alpha) \leq 0$$

where: $q(c\ ,\ \alpha) = Q(c\ ,\ F(c\ ,\ \alpha))$

$$g(c\ ,\ \alpha) = G(c\ ,\ F(c\ ,\ \alpha))$$

Starting from a given initial set point c^0, the algorithm has the following form:

Step 1: Apply the current set point c^k to the real process and obtain steady state measurement $F^*(c^k)$. Solve the parameter estimation problem to determine the new value of $\hat{\alpha}(c^k)$:

$$F(\hat{c}^k\ ,\ \hat{\alpha}^k) = F^*(c^k)$$

Step 2: With the parameter estimates $\hat{\alpha}(c^k)$ obtained from step 1, solve the optimisation problem (6.3) to obtain $\hat{c}^k$.

Steps 1 and 2 are repeated until condition (6.4) is satisfied within a desired tolerance.

$$\hat{c}^k - c^k = 0 \tag{6.4}$$

otherwise, update the control c^k and go to step 1 until no further improvement is observed.

The standard two-step approach will not converge to the correct optimum operating point (Roberts, 1979) as has been verified by simulation study (Brdyś, Chen and Roberts; 1986). This is due to the mismatching between the gradients of the real process outputs with respect to the set points and the corresponding gradients of the model (Durbeck, 1965). The information structure of the standard two-step approach is shown in figure (6.1).

6.2.2 The Modified Two-Step Approach:

The standard two-step approach has been modified by introducing an extra term to the optimisation performance index to allow for any mismatch between model and real output derivatives. The resulting algorithm is known as the modified two-step approach (Roberts, 1979). In deriving this approach, a new set of variables v is introduced into the problem. The optimisation problem (6.3) then becomes:

$$\begin{aligned} &\min_{c,v,\alpha} \; q(c\ ,\ \alpha) \\ \text{s.t:}\ \ &F(v\ ,\ \alpha) = F^*(v) \\ &g(c\ ,\ \alpha) \leq 0 \\ &v = c \end{aligned} \tag{6.5}$$

The Lagrangian function $\mathcal{L}(c\ ,\ v\ ,\ \alpha\ ,\ \lambda\ ,\ \beta\ ,\ \mu)$ for equation (6.5) is

$$\begin{aligned} \mathcal{L}(c\ ,\ v\ ,\ \alpha\ ,\ \lambda\ ,\ \eta\ ,\ \xi) = q(c\ ,\ \alpha) &+ \lambda^t(v - c) \\ &+ \eta^t[F(v\ ,\ \alpha) - F^*(v)] + \xi^t g(c,\alpha) \end{aligned} \tag{6.6}$$

where λ and η are the Lagrangian multipliers associated with the equality

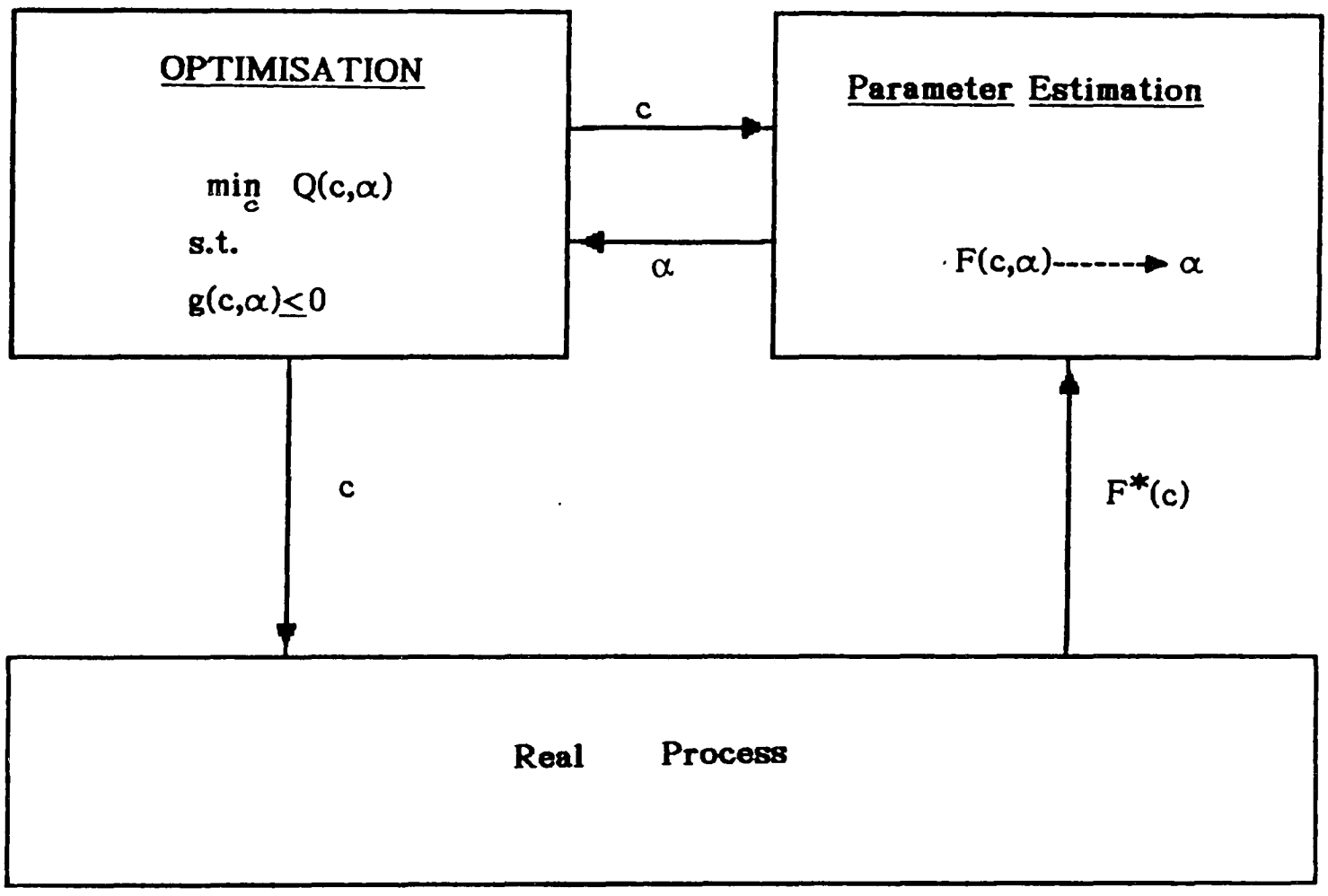


Figure (6.1) The Standard Two Step Approach.

constraints in problem (6.5) and ξ is the Kühn-Tucker (K-T) multiplier associated with the inequality constraint $g(c , \alpha)$.

Assuming that all required derivatives exist and regularity conditions are satisfied, the first order necessary optimality conditions for the existence of a stationary point of (6.6) are:

$$\nabla_c \mathcal{L}(.) = \frac{\partial q(c , \alpha)}{\partial c} - \lambda + \frac{\partial^t g(c , \alpha)}{\partial c} \xi = 0 \qquad (6.7)$$

$$\nabla_v \mathcal{L}(.) = \lambda + [\frac{\partial^t F(v , \alpha)}{\partial v} - \frac{\partial^t F^*(v)}{\partial v}] \eta = 0 \qquad (6.8)$$

$$\nabla_\alpha \mathcal{L}(.) = [\frac{\partial^t q(c , \alpha)}{\partial \alpha}] + [\frac{\partial^t F(v , \alpha)}{\partial \alpha}] \eta + [\frac{\partial^t g(c , \alpha)}{\partial \alpha}] \xi = 0 \qquad (6.9)$$

$$\nabla_\lambda \mathcal{L}(.) = v - c = 0 \qquad (6.10)$$

$$\nabla_\eta \mathcal{L}(.) = F(v , \alpha) - F^*(v) = 0 \qquad (6.11)$$

together with

$$\nabla_\xi \mathcal{L}(.) = g(c , \alpha) \leq 0 \qquad (6.12)$$

$$\xi^t \nabla_\xi \mathcal{L}(.) = \xi^t g(c , \alpha) \quad , \xi \geq 0 \qquad (6.13)$$

Solving equations (6.7) to (6.13) to obtain a formula for λ (Brdyś, Chen and Roberts, 1986), we obtain:

$$\lambda = \lambda(c , v , \alpha , \xi) =$$

$$= [\frac{\partial^t F(v , \alpha)}{\partial v} - \frac{\partial^t F^*(v)}{\partial v}].[\frac{\partial^t F(v , \alpha)}{\partial \alpha}]^{-1}.[\frac{\partial^t g(c , \alpha)}{\partial \alpha} \xi + \frac{\partial^t q(c , \alpha)}{\partial \alpha}] \qquad (6.14)$$

It is worth mentioning here that the formula (6.14) for updating λ is the same as in the original modified two-step approach (Roberts, 1978; 1979; Roberts and Ellis, 1981; Roberts and Williams, 1981), except for the extra term $[\frac{\partial^t g(c,\alpha)}{\partial \alpha}. \xi]$ in calculating λ, which was introduced by Brdyś, Chen and Roberts (1986) to cater for the influence of system inequality constraints.

For a given (c , v , α , ξ), λ can be evaluated such that conditions (6.8) and (6.9) are satisfied. The conditions (6.7), (6.10) - (6.13) are satisfied by solving the modified model based optimisation problem (Brdyś, Chen and Roberts, 1986):

$$\left.\begin{array}{l} \min_{c} \{q(c , \alpha) - \lambda^t c\} \\ \text{s.t: } g(c , \alpha) \leq 0 \end{array}\right\} \qquad (6.15)$$

where α is a parameter value such that:

$$F(v , \alpha) = F^*(v) \qquad (6.16)$$

Implementation aspects of the modified two-step algorithm:

Given some initial $v^0 \in \mathcal{C}$ and some initial guess ξ^0, the on-line implementation involves repeatedly the following procedure:

Step 1: Apply the controls v^k to the real system and obtain the steady-state output measurements $F^*(v^k)$. Solve the parameter estimation problem; equation (6.16); to determine a new parameter value $\hat{\alpha}^k$.

Perform additional perturbations around v^k and take the corresponding measurements to obtain finite difference approximations for $F^{*'}(v^k)$. Then calculate $\lambda^k = \lambda(\hat{c}^{k-1}, v^k, \alpha^k, \xi^k)$ using equation (6.14).

Step 2: Solve the model based optimisation problem (6.15) to obtain $\hat{c}^k$ for given α^k and λ^k.

Step 3: The iteration is terminated when conditions (6.17) and (6.18) are satisfied within a desired tolerance.

$$\hat{c}^k - v^k = 0 \qquad (6.17)$$

$$\hat{\xi}^k - \xi^k = 0 \tag{6.18}$$

otherwise, update v and ξ and go to step 1.

A simple relaxation may be used to update v and ξ (Brdyś, Chen and Roberts, 1986) as:

$$v^{k+1} = v^k + \epsilon_v (\hat{c}^k - v^k) \tag{6.19}$$

$$\xi^{k+1} = \xi^k + \epsilon_\xi (\hat{\xi}^k - \xi^k) \tag{6.20}$$

The algorithm was shown to be successful in solving many example problems and produced optimal results in spite of differences between model and reality. The information structure of the modified two-step approach is shown in figure (6.2).

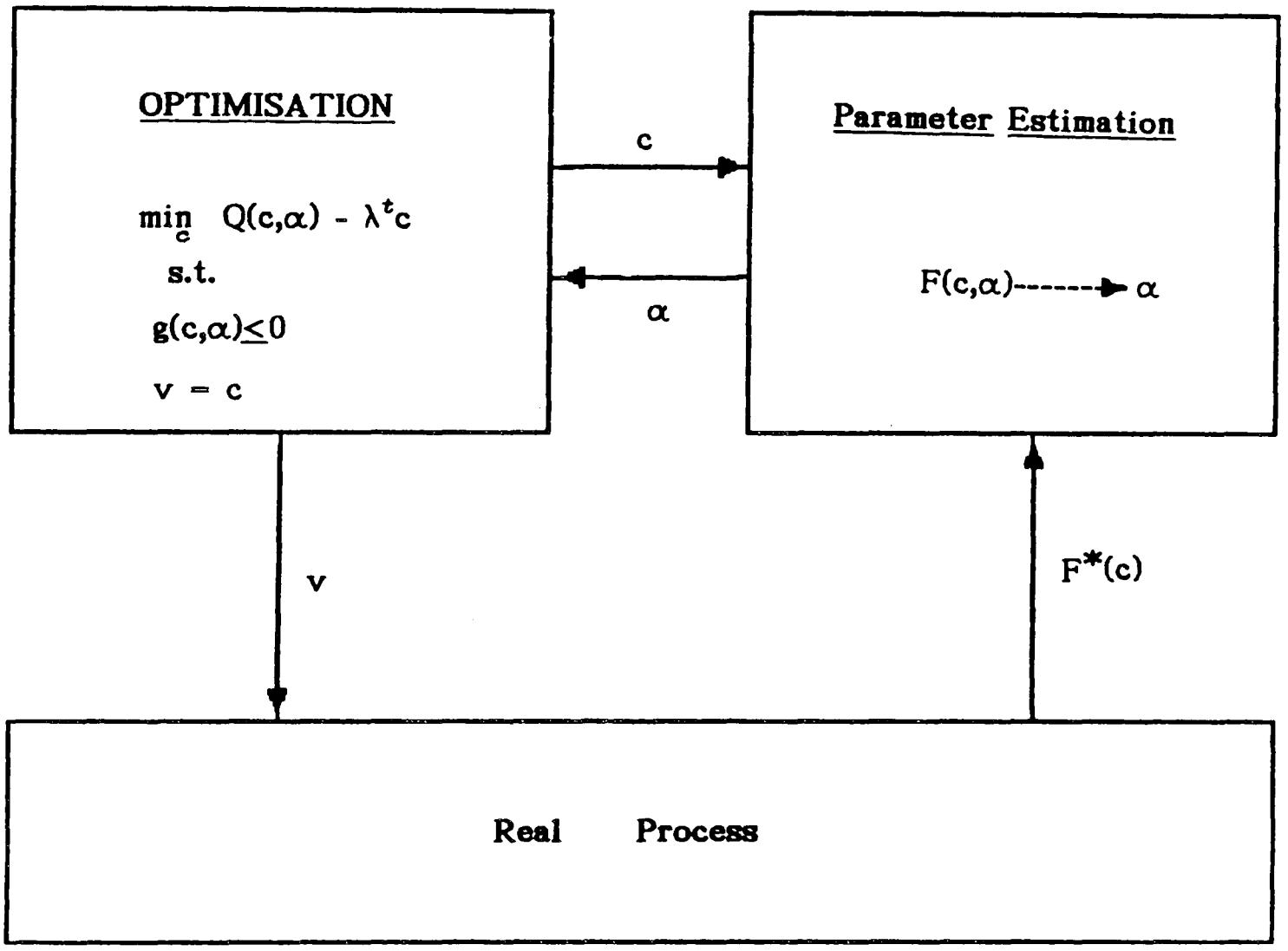


Figure (6.2) The Modified Two Step Approach.

6.2.3 The Approximate Linear Model Approach:

The Approximate Linear Model approach (ALM) is a modification of the modified two-step approach and involves system linearization techniques. It was proposed by Sheng and Ellis (1985) and proved to be successful in providing optimal results (Ellis, Roberts and Michalska , 1986; Kambhampati and Ellis, 1987).

The idea of the ALM approach is to utilize the sufficient conditions (Durbeck, 1965; Ellis and Roberts, 1982) which are required for the model based optimisation problem to produce optimum results, these are:

$$\text{i} - F(\hat{c}\ ,\ \hat{\alpha}) = F^{*}(\hat{c}) \tag{6.21}$$

$$\text{ii} - \frac{\partial F(\hat{c}\ ,\ \hat{\alpha})}{\partial c} = \frac{\partial F^{*}(\hat{c})}{\partial c} \tag{6.22}$$

where $\hat{c}$ represents the optimal value of the control, and $\hat{\alpha}$ is the corresponding model parameter at this point.
The model used is a linear function in the form:

$$F(c\ ,\ \alpha) = Wc + \alpha \tag{6.23}$$

$$\text{and } W = \frac{\partial F(c\ ,\ \alpha)}{\partial c} = \frac{\partial F^{*}(c)}{\partial c} \tag{6.24}$$

where $W \in \Re^{n\times m}$ and $\alpha \in \Re^{n}$.

Since the ALM approach uses the sufficient conditions (6.21) - (6.22), it means that the model matches the system. In that case, the modifier λ (as in the modified two-step approach) which exists because of the differences in the system and its model, will be zero. The optimisation and parameter estimation of the ALM approach are:
Optimisation Problem:

$$\begin{aligned} &\min_{c}\ Q(c\ ,\ y) \\ \text{s.t:}\ &y = Wc + \alpha \\ &G(c\ ,\ y) \leq 0 \end{aligned} \tag{6.25}$$

and

Parameter Estimation Problem:

$$\frac{\partial F(v\ ,\ \alpha)}{\partial v} \;=\; W \;=\; \frac{\partial F^{*}(v)}{\partial v} \tag{6.26}$$

$$\alpha \;=\; F^{*}(v) \;-\; Wv \tag{6.27}$$

Structure of the ALM approach:

As in the MTS approach, starting with some initial $v^0 \in \mathcal{C}$, and ξ^0, the on-line implementation of the ALM approach involves the following procedure:

Step 1: Apply perturbations about v^k to estimate the finite difference approximation for the derivative $\frac{\partial F^{*}(v^k)}{\partial v}$ and set it to W. Determine α by applying equation (6.27). Set the new model to

$$Y = Wc + \alpha$$

Step 2: Solve the optimisation problem (6.25) to obtain the controls $\hat{c}^k$.

Step 3: This is the same as in the MTS approach.

The control structure of The ALM approach is shown in figure (6.3).

In the ALM approach, the model can be considered as a local linear approximation to the process (Ellis, Sheng, Roberts and Michalska, 1985). This can be seen from equations (6.23) and (6.24) and assuming that α is estimated in a manner similar to the MTS approach. To clarify this, consider the Taylor series expansion of a function $F^{*}(v)$ about a point v^k:

$$F^{*}(v) = F^{*}(v^k) + F^{*'}(v^k)\ v^k + \| 0 \| \tag{6.28}$$

where $\| 0 \|$ denotes the higher terms of the expansion.
If the parameter α is set to:

$$\alpha = F^{*}(v^k) + \| 0 \| \tag{6.29}$$

the expression (6.28) will be:

$$F^{*}(v) = F^{*'}(v^{k})\, v^{k} + \alpha \qquad (6.30)$$

Equation (6.30) is the same structure as that used in the ALM approach (equation 6.23).

6.3 Hierarchical ISOPE Techniques:

The ISOPE techniques were originally developed for centralised optimisation without decomposition, but they have been extended for application within decomposed hierarchical optimisation structures (Michalska, Ellis and Roberts, 1985; Ellis, Michalska and Roberts, 1984). By incorporating the modified two-step approach with the price method, several approaches were proposed (Brdyś and Roberts, 1986) depending on how to utilize the available real process measurements efficiently. In this section, two of the proposed schemes are examined. These are:

i - The structure with output feedback

ii - The structure with input and output feedback.

The ISOPE techniques which are described in chapters seven and eight of this thesis use respectively the single loop technique with output feedback and the input-output feedback structures using double loop technique.

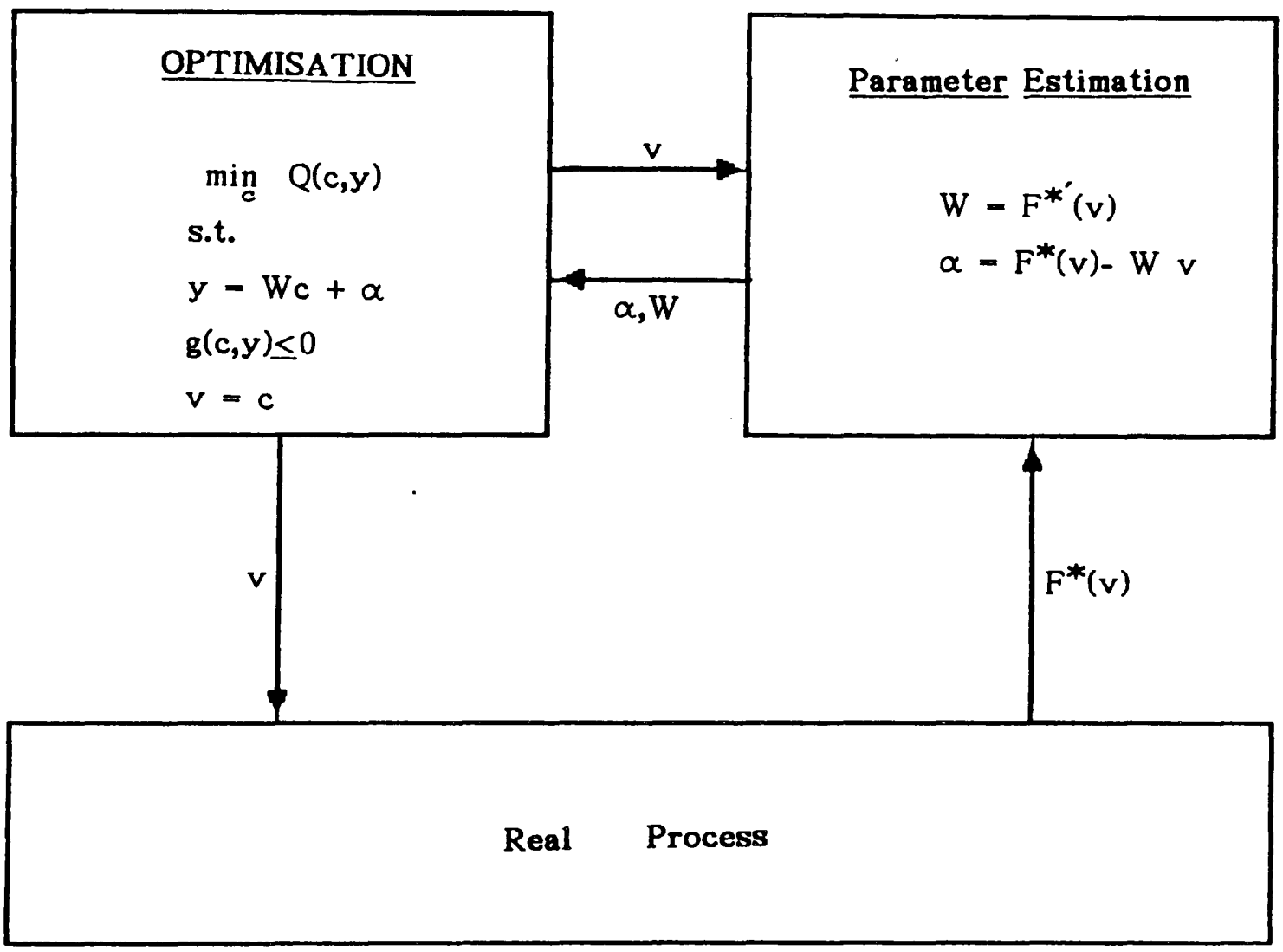


Figure (6.3) The Approximate Linear Model Approach.

6.3.1 Control Problem of the Decentralised ISOPE Technique:

It is assumed that external disturbances which affect the system are varying slowly and can be considered as constant in the considered time interval of determining optimum control. The controlled system is assumed to be described in a decomposed way by the set of subsystem input-output mappings:

$$F_i^* : \mathcal{C}_i \times \mathcal{U}_i \longrightarrow \mathcal{Y}_i \qquad i = 1,\ldots,N \tag{6.31}$$

where N denotes the number of subsystems and $\mathcal{C}_i$, $\mathcal{U}_i$, $\mathcal{Y}_i$ are finite dimensional spaces, as follows:

$$y_i = F_i^*(c_i , u_i) \qquad i = 1,\ldots,N \tag{6.32}$$

where the variables c_i, u_i, y_i are the i-th subsystem control, interaction input (interaction) and interaction output (output), respectively, and $c_i \in \mathcal{C}_i$, $u_i \in \mathcal{U}_i$, $y_i \in \mathcal{Y}_i$. The subsystems are interconnected through the coupling equations:

$$u_i = H_i\, y = \sum_{j=1}^{N} H_{ij}\ y_j \qquad i = 1,\ldots,N \tag{6.33}$$

where H_i and H_{ij} are interconnection matrices composed of zeros and ones.

Consider the usual assumption that (6.32) and (6.33) are uniquely solvable with respect to the controls, so that the input-output mapping can be expressed as:

$$y = K^*(c) \tag{6.34}$$

where:

$$K^*: \mathcal{C} \longrightarrow \mathcal{Y}$$

$$K^*(c) = (K_1^*(c), \ldots , K_N^*(c))$$

$$c = (c_1 , \ldots , c_N) \in \mathcal{C}_1 \times \ldots \times \mathcal{C}_N \triangleq \mathcal{C}$$

$$u = (u_1, \ldots, u_N) \in \mathcal{U}_1 \times \ldots \times \mathcal{U}_N \triangleq \mathcal{U}$$

$$y = (y_1, \ldots, y_N) \in \mathcal{Y}_1 \times \ldots \times \mathcal{Y}_N \triangleq \mathcal{Y}$$

The constraints can usually be partitioned into two parts. The local constraint set which is output independent takes the form:

$$c\ u_i = \{(c_i, u_i) \in \mathcal{C}_i \times \mathcal{U}_i \mid f_i(c_i, u_i) \leq 0\} \qquad i = 1, \ldots, N \tag{6.35}$$

or globally,

$$C\ U = C\ U_1 \times \ldots \times C\ U_N \tag{6.36}$$

where:

$$f_i : \mathcal{C}_i \times \mathcal{U}_i \longrightarrow \mathfrak{R}^S \qquad i = 1, \ldots, N$$

The global constraints which are generally output dependent can be expressed as:

$$G(c, u, y) \leq 0 \tag{6.37}$$

where:

$$G: \mathcal{C} \times \mathcal{U} \times \mathcal{Y} \longrightarrow \mathfrak{R}^L$$

In general, the real system relations are not known exactly, so their approximate models are used.

$$\left.\begin{aligned} &F_i : \mathcal{C}_i \times \mathcal{U}_i \times \mathcal{A}_i \longrightarrow \mathcal{Y}_i \\ &y_i : F_i(c_i, u_i, \alpha_i) \qquad i = 1, \ldots, N \end{aligned}\right\} \tag{6.38}$$

where $\mathcal{A}_i$ is a finite dimensional space and $\alpha_i \in \mathcal{A}_i$ is the i-th subsystem model parameter variable.

As in the case of the real system relations, the global model mapping can be written as:

$$\left.\begin{aligned} &F : \mathcal{C} \times \mathcal{U} \times \mathcal{A} \longrightarrow \mathcal{Y} \\ &y = F(c, u, \alpha) \end{aligned}\right\} \tag{6.39}$$

where:

$$\alpha \triangleq (\alpha_1 , \ldots , \alpha_N) \in \mathcal{A}_1 \times \ldots \times \mathcal{A}_N \triangleq \mathcal{A} \tag{6.40}$$

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

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

which is required to be minimised. The overall performance function

$$Q : \mathcal{C} \times \mathcal{U} \times \mathcal{Y} \longrightarrow \mathfrak{R}$$

is assumed to be:

$$Q(c , u , y) = \sum_{i=1}^{N} Q_i(c_i , u_i , y_i) \tag{6.41}$$

The task for determining the optimal operating condition for a real process can generally be defined by the following optimisation problem (ROP):

$$\min_{c,u,y} Q(c , u , y)$$

s.t:

$$\begin{aligned} & y = K^*(c) \\ & u = H\, y \\ & G(c , u , y) \leq 0 \\ & (c , u) \in \mathcal{CU} \end{aligned} \tag{6.42}$$

It is usually assumed that the following assumptions are satisfied:

A1: Mappings $G(c , u , y)$, $f(c , u)$, $F(c , u , y)$, $K^*(c)$ and the performance index $Q(c , u , y)$ are continuously Fréchet differentiable.

A2: Mappings $G(c , u , y)$ and $f(c , u)$ are convex.

A3: The model input-output mapping is point parametric on $\mathcal{CU}$ (Brdyś, 1983).

That is; a simple property is required for a chosen model, such that there exists $\alpha \in \mathfrak{R}^t$ such that $F(c,\alpha) = F^*(c)$.
The model is point parametric on $\mathcal{C}$ if it is point parametric at every point $c \in \mathcal{C}$. According to these assumptions (Brdyś, 1983), the real optimisation problem ROP (6.42) is equivalent to the following optimisation problem EOP1 (6.43):

$$\begin{aligned} &\min_{c,u,\alpha} \; q(c , u , \alpha) \\ &\text{s.t:} \\ &\left.\begin{aligned} &F(c , u , \alpha) = K^*(c) \\ &u = H\, F(c , u , \alpha) \end{aligned}\right\} \\ &g(c , u , \alpha) \leq 0 \\ &(c , u) \in \mathcal{CU} \end{aligned} \tag{6.43}$$

where:

$$g : \mathcal{C} \times \mathcal{U} \times \mathcal{A} \longrightarrow \mathfrak{R}^L \tag{6.44}$$

$$q : \mathcal{C} \times \mathcal{U} \times \mathcal{A} \longrightarrow \mathfrak{R} \tag{6.45}$$

with:

$$g(c , u , \alpha) \triangleq G(c , u , F(c , u ,\alpha)) \tag{6.46}$$

$$q(c , u , \alpha) \triangleq Q(c , u , F(c , u ,\alpha)) \tag{6.47}$$

similarly,

$$g^*: \mathcal{C} \times \mathcal{U} \longrightarrow \mathfrak{R}^L \tag{6.48}$$

$$q^*: \mathcal{C} \times \mathcal{U} \longrightarrow \mathfrak{R} \tag{6.49}$$

with:

$$g^*(c , u) \triangleq G(c , u , K^*(c)) \tag{6.50}$$

$$q^*(c , u) \triangleq Q(c , u , K^*(c)) \tag{6.51}$$

6.3.2 Single loop ISOPE with output feedback information:

The equivalent optimisation problem (6.43) is extended by introducing new variables v and w as follows (Brdyś and Roberts, 1986):

$$\min_{c,u,v,w,\alpha} q(c\ ,\ u\ ,\ \alpha)$$

s.t:

$$\begin{aligned} & u = H\ F(c\ ,\ u\ ,\ \alpha) \\ & g(c\ ,\ u\ ,\ \alpha) \leq 0 \\ & F(v\ ,\ w\ ,\ \alpha) = K^{*}(v) \\ & v = c \\ & w = u \end{aligned} \qquad (6.52)$$

The Lagrangian optimisation problem for (6.52) is:

$$\begin{aligned} & \mathcal{L}(c\ ,\ u\ ,\ v\ ,\ w\ ,\ \alpha\ ,\ p\ ,\ \lambda\ ,\ t\ ,\ \eta\ ,\ \xi) = \\ & = q(c\ ,\ u\ ,\ \alpha) + p^{t}(u - HF(c\ ,\ u\ ,\ \alpha)) + \lambda^{t}(v - c) + t^{t}(w - u) + \\ & \eta^{t}(F(v\ ,\ w\ ,\ \alpha) - K^{*}(v)) + \xi^{t} g(c\ ,\ u\ ,\ \alpha) \end{aligned} \qquad (6.53)$$

where λ, t, ξ, η are Lagrangian multipliers and p is the price vector.
For given values of v, w, α and p, the Lagrangian analysis provides the following optimisation problem (Brdyś and Roberts, 1986):

$$\left.\begin{aligned} & \min\ [q(c\ ,\ u\ ,\ \alpha) - \lambda^{t}\ c + p^{t}(\ u - HF(c\ ,\ u\ ,\ \alpha)) - t^{t}u] \\ & \text{s.t.}: \quad g(c\ ,\ u\ ,\ \alpha) \leq 0 \end{aligned}\right\} \qquad (6.54)$$

where:

$$\lambda(v\ ,\ w\ ,\ \alpha\ ,\ p\ ,\ \xi) = \left[\frac{\partial^{t} F(v\ ,\ w\ ,\ \alpha)}{\partial v} - \frac{\partial^{t} K^{*}(v)}{\partial v}\right] \left[\frac{\partial^{t} F(v\ ,\ w\ ,\ \alpha)}{\partial \alpha}\right]^{-1}$$

$$\left[\frac{\partial^{t} q(c\ ,\ u\ ,\ \alpha)}{\partial \alpha} - \frac{\partial^{t} F(c\ ,\ u\ ,\ \alpha)}{\partial \alpha} H^{t} p + \frac{\partial^{t} g(c\ ,\ u\ ,\ \alpha)}{\partial \alpha} \xi\right] \qquad (6.55)$$

$$t(v\ ,\ w\ ,\ \alpha\ ,\ p\ ,\ \xi) = [\frac{\partial^t F(v\ ,\ w\ ,\ \alpha)}{\partial w}\ ]\ [\frac{\partial^t F(v\ ,\ w\ ,\ \alpha)}{\partial \alpha}]^{-1}$$

$$[\frac{\partial^t q(c\ ,\ u\ ,\ \alpha)}{\partial \alpha}\ -\ \frac{\partial^t F(c\ ,\ u\ ,\ \alpha)}{\partial \alpha}\ H^t p\ +\frac{\partial^t g(c\ ,\ u\ ,\ \alpha)}{\partial \alpha}\xi] \qquad (6.56)$$

The parameter $\hat{\alpha}(v\ ,\ w)$ is determined by solving the parameter estimation problem:

$$F(v\ ,\ w\ ,\ \alpha(v\ ,\ w)) = K^*(v) \qquad (6.57)$$

It is assumed that all the required inverse matrices exist.

The real optimisation problem (6.42) can be solved by the equivalent problem (6.54) obtaining a solution c , u. The parameter α is obtained from equation (6.57) and the variables v, w, ξ, p are adjusted in an appropriate way (Abdullah, 1988; Brdyś, Abdullah and Roberts, 1986) such that:

$$\hat{c}\ (v\ ,\ w\ ,\ \xi\ ,\ p) = p \qquad (6.58)$$

$$\hat{u}\ (v\ ,\ w,\ \xi\ ,\ p) = w \qquad (6.59)$$

$$\hat{\xi}\ (v\ ,\ w,\ \xi\ ,\ p) = \xi \qquad (6.60)$$

$$\hat{u}\ (v,\ w,\ \xi,\ p) = HF(\hat{c}(v,\ w,\ \xi,\ p),\ \hat{u}(v,\ w,\ \xi,\ p),\ \hat{\alpha}(v,\ w)) \qquad (6.61)$$

Three different strategies were proposed (Brdyś, Abdullah and Roberts, 1985) to solve equations (6.58) to (6.61). These are:

1. System based double loop technique,
2. Model based double loop technique,
3. Single loop technique.

In single loop techniques, all variables (p , v , w and ξ) are iterated with the same frequency. The variables are updated using a simple relaxation technique as suggested by Brdyś and Roberts (1985) and Chen, Brdyś and Roberts (1986):

$$p^{k+1} = p^k + \epsilon_p\ [\hat{u}^k - HF(\hat{c}^k,\ \hat{u}^k,\ \hat{\alpha}^k)] \qquad (6.62)$$

$$v^{k+1} = v^k + \epsilon_v (\hat{c}^k - v^k) \tag{6.63}$$

$$w^{k+1} = w^k + \epsilon_w (\hat{u}^k - w^k) \tag{6.64}$$

$$\xi^{k+1} = \xi^k + \epsilon_\xi (\hat{\xi}^k - \xi^k) \tag{6.65}$$

where ϵ_p, ϵ_v, ϵ_w, and ϵ_ξ are positive constants named gain coefficients which are less than or equal to unity.

Figure (6.4) shows the information structure of the algorithm. The procedure of the algorithm can be described as follows:

a- For a given control v^k applied to the real system, take measurements $K^*(v)$.

b- Solve the parameter estimation problem (6.57) and obtain a new parameter $\hat{\alpha}^k$.

c- Perform additional perturbation around v^k and take the corresponding measurements to find finite difference approximations $\frac{\partial^t K^*(v)}{\partial v}$ which are required to calculate λ.

d- For a given α, λ, t, ξ and p, solve the optimisation problem (6.54) to obtain the solution $\hat{c}$, $\hat{u}$.

e- The overall convergence is achieved when successive solutions of v, ξ are unchanged and interaction balance is satisfied:

$$v^k - \hat{c}^k = 0 \tag{6.66}$$

$$\xi^k - \hat{\xi}^k = 0 \tag{6.67}$$

$$\hat{u}^k - HF(\hat{c}^k, \hat{u}^k, \alpha^t) = 0 \tag{6.68}$$

f- In practice, the overall process may be terminated when v, $\hat{u}$ and ξ satisfy the conditions (6.66) to (6.68) within some desired tolerances, otherwise update the parameters (p, v, w, ξ) according to equations (6.62) - (6.65) and start again from step (a).

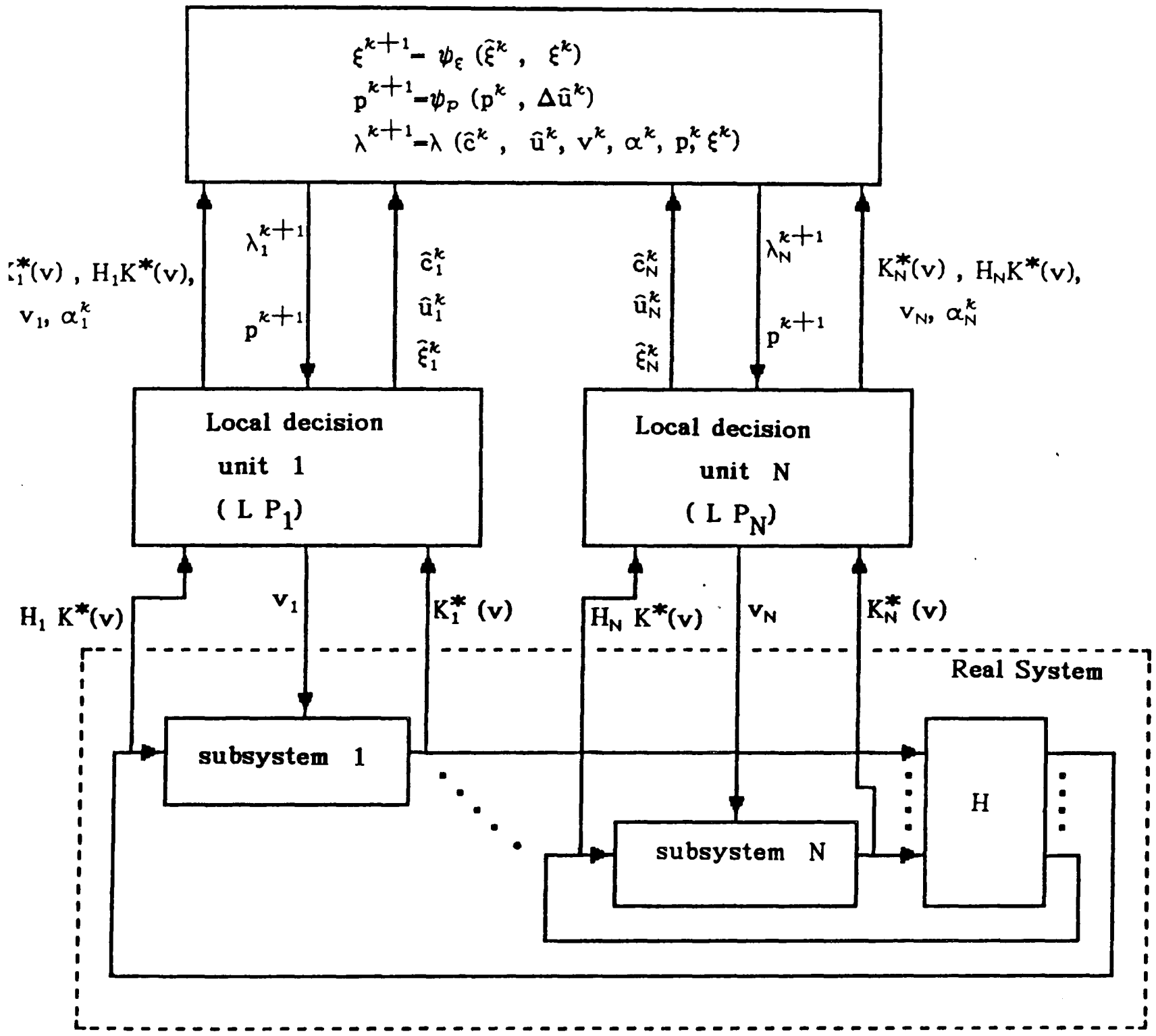


Figure (6. 4) Information Siructure of SIA (ISOPE).

6.3.3 Model based double loop ISOPE with input-output feedback information:

The equivalent optimisation problem (6.43) can be extended by introducing a new variable v as follows :

$$\begin{aligned} &\min_{c,u,v,\alpha} \; q(c\ ,\ u\ ,\ \alpha) \\ &\text{s.t:} \\ &\quad u = H\ F(c\ ,\ u\ ,\ \alpha) \\ &\quad g(c\ ,\ u\ ,\ \alpha) \leq 0 \\ &\quad F(v\ ,\ HK^{*}(v)\ ,\ \alpha) = K^{*}(v) \\ &\quad v = c \end{aligned} \tag{6.69}$$

The Lagrangian associated with problem (6.69) is:

$$\begin{aligned} \mathcal{L}(c\ ,\ u\ ,\ v\ ,\ \alpha\ ,\ p\ ,\ \lambda\ ,\ \xi,\ \eta) &= q(c\ ,\ u\ ,\ \alpha) + \\ &+ p^{t}\ [u - HF(c\ ,\ u\ ,\ \alpha)] + \lambda^{t}(v - c) + \xi^{t} g(c\ ,\ u\ ,\ \alpha) + \\ &+ \eta^{t}\ [F(v\ ,\ HK^{*}(v)\ ,\ \alpha) - K^{*}(v)] \end{aligned} \tag{6.70}$$

For a given value of v, α, ξ and p, the Lagrangian analysis provides the following optimisation problem (Brdyś and Roberts, 1986):

$$\left.\begin{aligned} &\min_{c,u} \; [q(c\ ,\ u\ ,\ \alpha) + p^{t}(\ u - HF(c\ ,\ u\ ,\ \alpha)) - \lambda^{t} c] \\ &\text{s.t:} \quad g(c\ ,\ u\ ,\ \alpha) \leq 0 \end{aligned}\right\} \tag{6.71}$$

where λ is given by

$$\begin{aligned} \lambda(c\ ,\ u,\ v\ ,\ \alpha\ ,\ \xi,\ p) = &- [\ \frac{\partial^{t} K^{*}(v)}{\partial v} - \frac{\partial^{t} F(v\ ,\ HK^{*}(v)\ ,\ \alpha)}{\partial v}] \\ &[\frac{\partial^{t} F(v\ ,\ HK^{*}(v)\ ,\ \alpha)}{\partial \alpha}]^{-1} .[- \frac{\partial^{t} q(c\ ,\ u\ ,\ \alpha)}{\partial \alpha} \\ &+ \frac{\partial^{t} F(c\ ,\ u\ ,\ \alpha)}{\partial \alpha} H^{t} p - \frac{\partial^{t} g(c\ ,\ u\ ,\ \alpha)}{\partial \alpha} \xi] \end{aligned} \tag{6.72}$$

In double loop algorithms, the idea is to separate the function of improving the modifier λ and the local parameter estimation from that of improving the price p and local optimisation. The task of the local control problem (LP) and the coordinator can be described as follows (CP):

a- For given controls v, obtain the corresponding measurements $K^*(v)$ and $HK^*(v)$, then find the model parameters α which satisfy:

$$F(v\ ,\ HK^*(v),\ \alpha) - K^*(v) \ = \ 0$$

b- For given α, λ, ξ and p, find a solution $(\hat{c}\ ,\ \hat{u})$ for the optimisation problem (6.71).

The coordinator task can be stated as follows:

Find p such that:

$$\left.\begin{aligned} &\hat{u} - HF(\hat{c},\ \hat{u}\ ,\ \alpha) \ = \ 0 \\ &\hat{\xi}^k - \xi^k \ = \ 0 \end{aligned}\right\} \qquad \text{(CP)}$$

where the price vector p is adjusted in the inner loop and λ, ξ are updated in the outer loop.

The overall process is terminated where v and ξ remain sufficiently unchanged between successive iterations.
The information structure of the algorithm is shown in figure (6.5).

It can be observed that all the parameters which need real system information (λ, α) are determined or adjusted in the outer loop. In the inner loop where most of the calculations are required, it remains purely model based. Hence, double loop structures are expected to reduce on-line computing time which has been confirmed by many simulations results (for example Chen, Brdyś and Roberts, 1986; Lin Hendawy and Roberts, 1988b).

6.4 Summary:

The modified two-step approach proposed by Roberts (1979) proved to produce optimal results in spite of model-reality differences.

The applicability of the modified two-step approach was extended by Ellis, Michalska and Roberts (1984); Brdyś and Roberts (1986) to interconnected large scale systems. This was achieved by combining the price method with the modified two-step approach.

Lin, Chen and Roberts (1986) have further modified the modified two-step approach to include the output dependent constraints (Chen, 1986). Sheng and Ellis (1985) had combined the linearization technique with the modified two-step approach to produce the approximate linear model approach (ALM). It has been proved that the ALM approach provides optimal results (Kambhampati, 1988).

Several optimal structures are proposed by Brdyś and Roberts (1986). The choice of a particular structure depends on the measurements capabilities and the method of utilizing the system feedback information. The input-output feedback information structure has proved to be superior than structures with output feedback only (Brdyś, Chen and Roberts, 1986) and the model based double loop algorithm reduces the number of set point changes significantly.

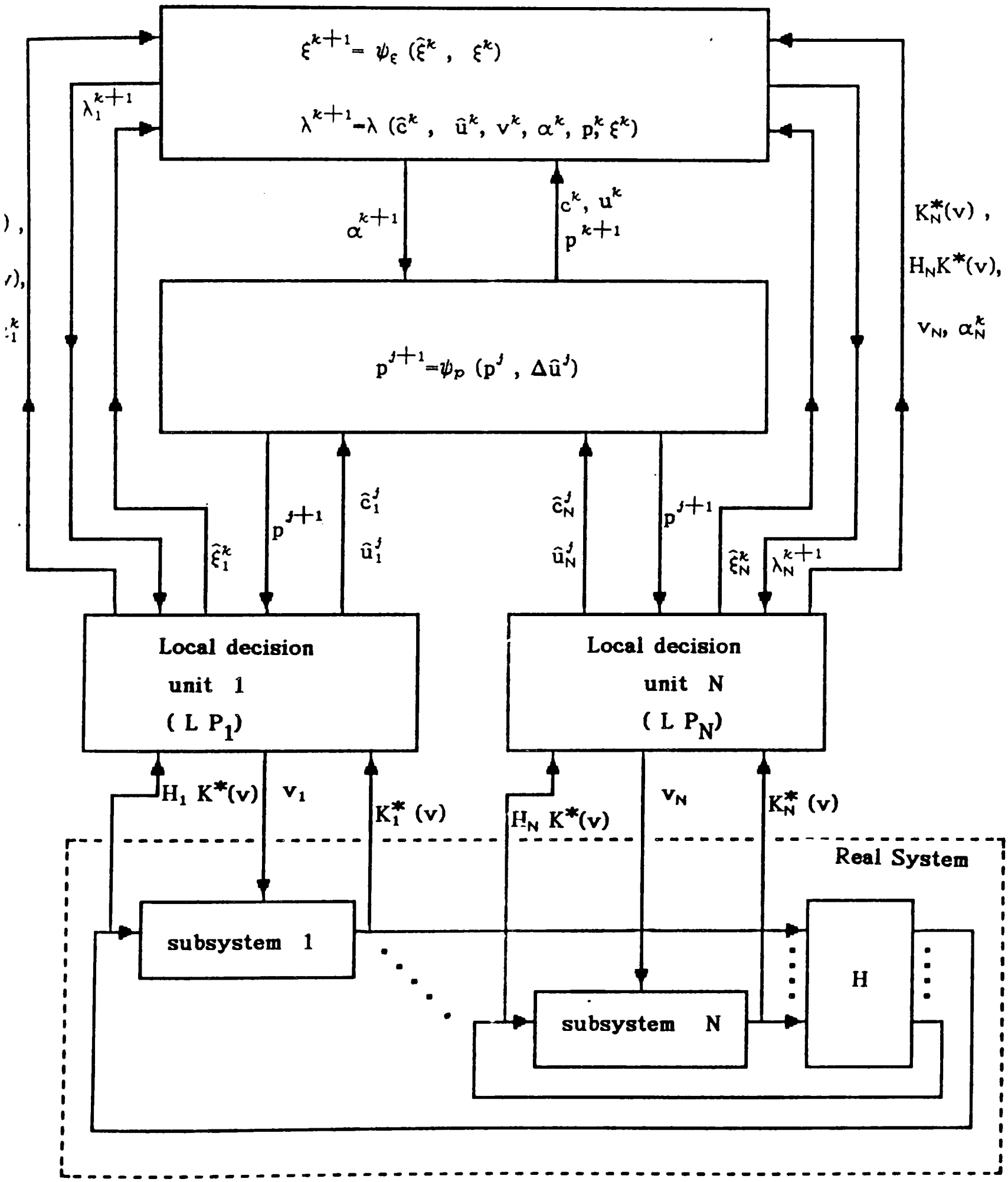


Figure (6. 5) Information Structure of the DIA (ISOPE).

Chapter 7

An extension of ISOPE to hierarchical control of steady state systems with output dependent constraints

7.1 Introduction:

As it is mentioned in Chapter 6, the integrated system optimisation and parameter estimation technique can achieve the real optimum result in spite of model-reality differences. However, in the early versions of integrated system optimisation and parameter estimation, there are some practical limitations. For instance, the steady-state optimal condition can be achieved only when the system inequality constraints are not output dependent. Chen, Brdyś and Roberts (1986) have attempted to introduce a certain technique to deal with output dependent constraints, to encompass a more general class of problems. However, the existence of the model based optimal solution of that algorithm (Chen, Brdyś and Roberts, 1986) is not ensured during the course of iteration. Also, the convergence conditions of this algorithm have not been derived.

In this chapter, an alternative technique is described to take into account general inequality constraints. The resulting algorithm is a combination of the apporach given by (Lin, Chen and Roberts, 1986) and the price method.

This algorithm used the output feedback structure (Brdyś and Roberts, 1986) and the single loop iterative strategy (Brdyś, Abdullah and Roberts, 1986).

The main advantages of the algorithm are that the existence of the model based optimisation solution is ensured during the course of iteration and the model based optimisation problem can be simplified considerably.

7.1 Viability of output dependent constraints:

The importance of considering output dependent constraints lies on some reasons, the most important of them are:

1- For some practical reasons, it is necessary to consider output dependent constraints. By converting these constraints in terms of set point c and interaction balance u, the significance of inequality constraints may be lost. Moreover, in some applications, this conversion is meaningless in spite of its validity from a theoretical point of view.

To clarify that, consider the following two subsystem example, shown in figure (7.1), where the interconnection relations are :

$$y_1 = u_2$$
$$y_2 = u_1 \quad , \quad \text{or}$$

$$\begin{bmatrix} u_1 \\ u_2 \end{bmatrix} = \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix} \begin{bmatrix} y_1 \\ y_2 \end{bmatrix}$$

and the system is subjected to the inequality constraint :

$$y_1 \leq 30$$

In this example, the output of the first subsystem y_1 should be equal to or less than 30. If this output dependent constraint is converted to c, u form, it becomes $u_2 \leq 30$ which can be misleading.

2- Handling inequality constraints in their equivalent c, u can lose the

decomposability of the constraints. For example; the global constraints which are generally output dependent can be expressed as:

$$G\ (c,\ u,\ y) \leq \quad 0 \tag{7.1}$$

or in a decomposable form:

$$G_i(c_i,\ u_i,\ y_i) \leq \ 0 \qquad\qquad i = 1,\ \ldots\ ,\ N \tag{7.2}$$

The interconnection of subsystems is given by

$$u = H\ y \tag{7.3}$$

Assuming H is a square and non-singular matrix, equation (7.3) can be written as:

$$y = \ H^{-1}\ u \tag{7.4}$$

Substituting from equation (7.4) in equation (7.2) leads to:

$$G_i\ (c_i,\ u_i,\ [\ H^{-1}\ u]_i) \ \leq \quad 0 \qquad i = 1,\ \ldots\ ,\ N \tag{7.5}$$

Obtaining inequality constraints in the form of equation (7.5), makes the constraints in a not decomposable form.

Therefore, in some cases it is essential to solve the problems subjected to output dependent inequality constraints in their original form.

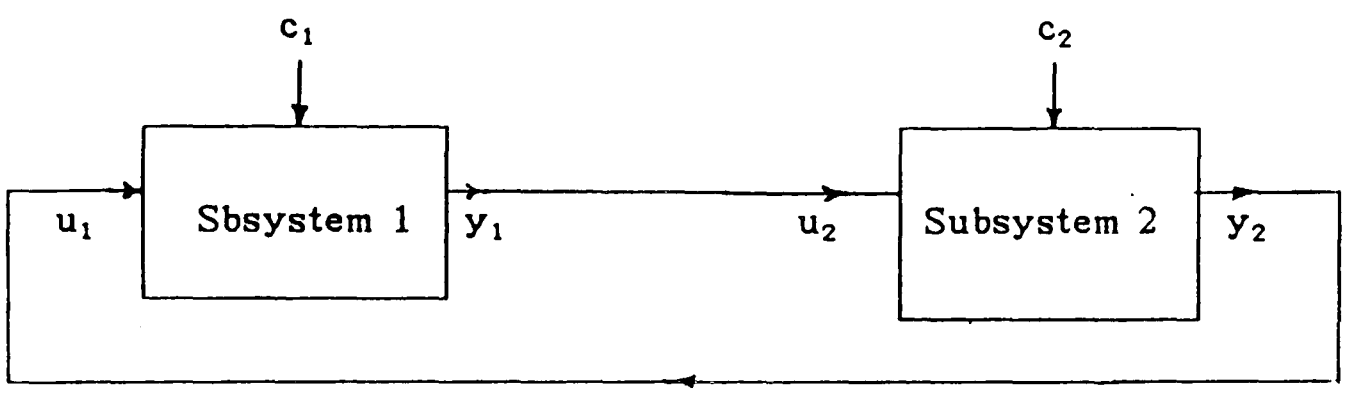


Figure (7. 1) Two Subsystem Example with output-dependent inequality constraints

7.3 Formulation of the algorithm:

To determine the optimal operating condition of a real process, the task is to solve the real optimisation problem (6.42); that is rewritten again as:

$$\min_{c,\,u,\,y} \; Q(c\,,\,u,\,y)$$

s.t:

$$y = K^{*}(c) \qquad \text{ROP}$$

$$G\,(c,\,u,\,y) \leq 0$$

$$(c,\,u) \in \mathcal{CU}$$

If assumptions A_1, A_2 and A_3 that were mentioned in Chapter 6 are satisfied, the ROP is equivalent to the EOP1 (6.43); which is:

$$\min_{c,\,u,\,\alpha} \; q\,(c,\,u,\,\alpha)$$

s.t: $F(c,\,u,\,\alpha) = K^{*}(c)$

$$u = HF(c,\,u,\,\alpha) \qquad \text{EOP1}$$

$$g\,(c,\,u,\,\alpha) \leq 0$$

$$(c,\,u) \in \mathcal{CU}$$

In order to separate the optimisation and parameter estimation problems, additional equations $v = c$, $w = u$ are introduced. The (EOP1) is transferred to the equivalent control problem (EOP):

$$\min_{c,\,u,\,v,\,\alpha} \; q\,(c,\,u,\,\alpha)$$

s.t: $F(v,\,w,\,\alpha) = K^{*}(v)$

$$u = HF(c,\,u,\,\alpha) \qquad \text{EOP}$$

$$g\,(c,\,u,\,\alpha) \leq 0$$

$$v = c$$

$$w = u$$

The Lagrangian associated with (EOP) can then be written as:

$$\mathcal{L}(c, u, v, w, \alpha, p, \lambda, \eta, \xi, t) = q(c, u, \alpha) + p^t[u - HF(c, u, \alpha)] +$$

$$+ \lambda^t(v-c) + t^t(w-u) + \eta^t[F(v, w, \alpha) - K^*(v)] +$$

$$+ \xi^t\, g(c, u, \alpha) \qquad (7.6)$$

where p, λ, η, t and ξ are Lagrange multipliers.
In particular, p is known as the price vector, λ is known as the modifier vector and ξ is denoted as the Lagrange multiplier vector associated with G. Assuming that all required derivatives exist and regularity conditions are satisfied, the Kuhn-Tucker necessary optimality conditions (see, for example, Luenberger, 1984; Gill, Murray and Wright, 1981) of (EOP) can be stated as:

$$\nabla_c \mathcal{L}(\,.\,) = \frac{\partial^t q(c, u, \alpha)}{\partial c} - \frac{\partial^t F(c, u, \alpha)}{\partial c} H^t p - \lambda + \frac{\partial^t g(c, u, \alpha)}{\partial c}\ \xi = 0 \qquad (7.7)$$

$$\nabla_u \mathcal{L}(\,.\,) = \frac{\partial^t q(c, u, \alpha)}{\partial u} - \frac{\partial^t [u - HF(c, u, \alpha)]}{\partial u}\ p - t + \frac{\partial^t g(c, u, \alpha)}{\partial u}\ \xi = 0 \qquad (7.8)$$

$$\nabla_v \mathcal{L}(\,.\,) = \lambda + \left[\frac{\partial^t F(v, w, \alpha)}{\partial v} - \frac{\partial^t K^*(v)}{\partial v}\right] \eta = 0 \qquad (7.9)$$

$$\nabla_w \mathcal{L}(\,.\,) = t + \frac{\partial^t F(v, w, \alpha)}{\partial w}\ \eta = 0 \qquad (7.10)$$

$$\nabla_\alpha \mathcal{L}(\,.\,) = \frac{\partial^t q(c, u, \alpha)}{\partial \alpha} - \frac{\partial^t F(c, u, \alpha)}{\partial \alpha} H^t p + \frac{\partial^t F(v, w, \alpha)}{\partial \alpha}\ \eta$$

$$+ \frac{\partial^t g(c, u, \alpha)}{\partial \alpha}\ \xi = 0 \qquad (7.11)$$

$$\nabla_p \mathcal{L}(\,.\,) = u - HF(c, u\ \alpha) = 0 \qquad (7.12)$$

$$\nabla_\lambda \mathcal{L}(\,.\,) = v - c = 0 \qquad (7.13)$$

$$\nabla_t \mathcal{L}(\,.\,) = w - u = 0 \qquad (7.14)$$

$$\nabla_\eta \mathcal{L}(\,.\,) = F(v, w, \alpha) - K^*(v) = 0 \qquad (7.15)$$

$$\nabla_\xi \mathcal{L}(\,.\,) = g(c,\ u\ \alpha) \leq 0 \tag{7.16}$$

$$\xi^t\ \nabla_\xi \mathcal{L}(\,.\,) = \xi^t\ g(c,\ u,\ \alpha) = 0,\quad \xi \geq 0 \tag{7.17}$$

To obtain a solution of equations (7.7) to (7.17), assume $[\ \frac{\partial^t F(v,\ w,\ \alpha)}{\partial\ \alpha}\ ]^{-1}$ exists, η is solved from (7.11) and substituted into (7.9) to obtain the formula of λ:

$$\lambda = [\ \frac{\partial^t F(v,\ w,\ \alpha)}{\partial\ v} - \frac{\partial^t K^*(v)}{\partial\ v}]\ [\ \frac{\partial^t F(v,\ w,\ \alpha)}{\partial\ \alpha}]^{-1}$$

$$[\frac{\partial^t q(c,\ u,\ \alpha)}{\partial\ \alpha} - \frac{\partial^t F(c,\ u,\ \alpha)}{\partial\ \alpha} H^t p + \frac{\partial^t g(c,\ u,\ \alpha)}{\partial\ \alpha}\ \xi] \tag{7.18}$$

Using the chain rule (Binmore, 1983) and equations (6.46) and (6.47) we obtain:

$$\left.\begin{aligned} \frac{\partial^t q(c,\ u,\ \alpha)}{\partial \alpha} &= \nabla_\alpha\ F\ (c,\ u,\ \alpha)\ .\ \nabla_y\ Q(c,\ u,\ F(c,\ u,\ \alpha)) \\ \frac{\partial^t g(c,\ u,\ \alpha)}{\partial \alpha} &= \nabla_\alpha\ F\ (c,\ u,\ \alpha)\ .\ \nabla_y\ G(c,\ u,\ F(c,\ u,\ \alpha)) \end{aligned}\right\} \tag{7.19}$$

Substituting equation (7.19) into equation (7.18) we obtain:

$$\lambda = [\ \frac{\partial^t F(v,\ w,\ \alpha)}{\partial\ v} - \frac{\partial^t K^*(v)}{\partial\ v}]\ [\ \frac{\partial^t F(v,\ w,\ \alpha)}{\partial\ \alpha}]^{-1}\ .$$

$$[\ \frac{\partial^t F(c,\ u,\ \alpha)}{\partial\ \alpha}\ .\ \frac{\partial^t Q\ (c,\ u,\ F(c,\ u,\ \alpha))}{\partial y} - \frac{\partial^t F(c,\ u,\ \alpha)}{\partial\ \alpha} H^t p\ +$$

$$+\ [\frac{\partial^t F(c,\ u,\ \alpha)}{\partial\ \alpha}.\ \frac{\partial G\ (c,\ u,\ F(c,\ u,\ \alpha))}{\partial\ \alpha}]\ \xi] \tag{7.20}$$

or: $\lambda = [\nabla_v\ F(v,\ w,\ \alpha) - \nabla_v\ K^*(v)]\ [\nabla_y\ Q(v,\ w,\ y) +$

$$- \nabla_y\ G(v,\ w,\ F(v,\ w,\ y))\xi - H^t p] \tag{7.21}$$

Where:

$y = F(v,\ w,\ y)$, and

$F'_v(v,\ w,\ \alpha)$ is the derivative of F with respect to v,

$\nabla_v F(v, w, \alpha) = [F'_v(v, w, \alpha)]^t$ are the gradients of F with respect to v and so on.

To obtain a formula for the modifier t; from equation (7.10) we obtain:

$$t = - [\frac{\partial^t F(v, w, \alpha)}{\partial w}] \eta \tag{7.22}$$

Substituting from equation (7.11) by the value of η in equation (3.2) we obtain:

$$t = [\frac{\partial^t F(v, w, \alpha)}{\partial w}] [\frac{\partial^t F(v, w, \alpha)}{\partial \alpha}]^{-1}$$

$$[\frac{\partial^t q(c, v, \alpha)}{\partial \alpha} - \frac{\partial^t F(c, u, \alpha)}{\partial \alpha} H^t p + \frac{\partial^t g(c, u, \alpha)}{\partial \alpha} \xi] \tag{7.23}$$

Substitution equation (7.19) into (7.23) we obtain:

$$t = [\frac{\partial^t F(v, w, \alpha)}{\partial w}] [\frac{\partial^t F(v, w, \alpha)}{\partial \alpha}]^{-1} [\frac{\partial^t F(v, w, \alpha)}{\partial \alpha} \cdot \frac{\partial^t Q(c, u, y)}{\partial y} -$$

$$\frac{\partial^t F(v, w, \alpha)}{\partial \alpha} H^t p + \{\frac{\partial^t F(v, w, \alpha)}{\partial \alpha} \cdot \frac{\partial^t G(v, w, y)}{\partial y}\} \xi]$$

$$t = [\frac{\partial^t F(v, w, \alpha)}{\partial w}] [\frac{\partial^t Q(v, w, y)}{\partial y} - H^t p + \frac{\partial^t G(v, w, y)}{\partial y} \xi]$$

or

$$t(v, w, \alpha, \xi, p) = [\nabla_w F(v, w, \alpha)] [\nabla_y Q(v, w, y) +$$

$$+ \nabla_y G(v, w, y)\xi - H^t p] \tag{7.24}$$

7.4 Implementation aspects of the algorithm:

Given some initial v^0, w^0 satisfying $(v^0, w^0) \in \mathcal{CU}$ and some initial guess ξ^0, p^0 the procedure of the algorithm is described as follows:

Step 1:

Apply v^k to the real process to obtain the corresponding steady state measurement $K^*(v^k)$.

Determine α^k by solving

$$F(v^k, w^k, \alpha) = K_*(v^k)$$

Perform additional perturbations about v^k and measure the corresponding process outputs to compute the finite difference approximation of $K^{*'}(v^k)$. Finally, calculate the modifiers

$$\lambda^k = \lambda(v^k, w^k, \hat{\alpha}^k, \xi^k, p^k),$$

$$t^k = t(v^k, w^k, \hat{\alpha}^k, \xi^k, p^k)$$

according to equations (7.21) and (7.24) respectively.

Step 2:

For a given $\hat{\alpha}^k$, λ^k, t^k, ξ^k and p^k solve the following model based optimisation problem:

$$\left.\begin{aligned}\min_{c,u} \{&q(c, u, \hat{\alpha}^k) - (\lambda^k)^t c - (t^k)^t u + (p^k)^t [u - HF(c, u, \hat{\alpha}^k)] + \\ &[\nabla_c g(v^k, w^k, \hat{\alpha}^k) \xi^k]^t c + [\nabla_u g(v^k, w^k, \hat{\alpha}^k) \xi^k]^t u\}\end{aligned}\right\} \text{MOP(7.25)}$$

s.t: $(c, u) \in \mathcal{CU}$

The solution of (MOP) is denoted as $\hat{c}(v^k, w^k, \hat{\alpha}^k, \xi^k, p^k)$, $\hat{u}(v^k, w^k, \hat{\alpha}^k, \xi^k, p^k)$ or $\hat{c}^k$, $\hat{u}^k$ for short.

The iterative variables v^k, w^k, ξ^k and p^k are updated according to the formulae (Lin, Hendawy and Roberts, 1988a):

$$v^{k+1} = v^k + \epsilon_v (\hat{c}^k - v^k) \tag{7.26}$$

$$w^{k+1} = w^k + \epsilon_w (\hat{u}^k - w^k) \tag{7.27}$$

$$\xi^{k+1} = \xi^k + \epsilon_\xi T(v^k, w^k, \xi^k, p^k) R(v^k, w^k, \xi^k, p^k) \tag{7.28}$$

$$p^{k+1} = p^k + \epsilon_p [\hat{u}^k - HK^*(v^k) - \nabla K^*(v^k) H^t(\hat{c}^k - v^k)] \tag{7.29}$$

where:

$$R(v, w, \xi, p) = g^*(v, w) + \nabla_c\, g^*(v, w)^t\, [\hat{c}(v, w, \hat{\alpha}, \xi, p) - v] +$$

$$\nabla_u\, g^*(v, w)^t\, [\hat{u}(v, w, \hat{\alpha}, \xi, p) - w] \qquad (7.30)$$

$T(v, w, \xi, p)$ is a projection operator satisfying:

$$\left.\begin{aligned} &[T\,(v, w, \xi, p)\, R(v, w, \xi, p)]_i = 0 \qquad\qquad i \in N^1(v, w, \xi, p) \\ &[T\,(v, w, \xi, p)\, R(v, w, \xi, p)]_i = R\,(v, w, \xi, p)_i \quad \forall\, i \notin N^1(v, w, \xi, p) \end{aligned}\right\} (7.31)$$

$$N^1(v, w, \xi, p) = \{\, i: \xi_i = 0 \ \text{ and } R(v, w, \xi, p)_i \leq 0 \,\} \qquad (7.32)$$

where $[\,.\,]_i$ denotes the ith element of the vector.

The gain parameter ϵ^k_ξ is determined as follows:

$$\text{Define} \quad N^2(v, w, \xi, p) \triangleq \{\, i : \xi_i > 0 \text{ and } R(v, w, \xi, p)_i < 0 \,\} \qquad (7.33)$$

$$\text{and} \quad \bar{\epsilon}^k_\xi = \min \{\, -\xi^k_i \,/\, R(v^k, w^k, \xi^k, p^k)\} \quad i \in N^2\,(v^k, w^k, \xi^k, p^k)\} \quad (7.34)$$

where ϵ_v, ϵ_w, ϵ_ξ and ϵ_p are positive constants, known as gain coefficients.

Equation (7.29) is representing the updating formula for the price vector p, and the innovation term in that equation;

$$[\hat{u}^k - HK^*(v^k) - \nabla K^*(v^k)\, H^t\, (\hat{c}^k - v^k)]$$

is simply the Taylor expansion of $[\hat{u}^k - HK^*(\hat{c}^k)]$ at v^k. Also $R(v^k, w^k, \xi^k, p^k)$ in equation (7.30) is the Taylor expansion of $g^*(\hat{c}^k, \hat{u}^k)$ at (v^k, w^k).

In equation (7.28), the purpose of introducing the projection operator T which is defined by equations (7.31) to (7.35) is to exclude the possibility of ξ becoming negative. This can be seen clearly through equation (7.31).

For those i which belong to N^1, we have $\xi^k_i = 0$ and $R^k_i \leq 0$. Therefore, for any positive gain, the updating formula will produce negative ξ_i^{k+1}. Hence , the gain for those $i \in N^1$ should be zero. Instead of changing gain from element to element, a projection operator T was employed for this purpose. After the multiplication of T in equation (7.28), the updating formula of ξ will be:

$$\xi_i^{k+1} = \xi_i^k \qquad \text{for every } i \in N^1, \text{ and}$$

$$\xi_i^{k+1} = \xi_i^k + \epsilon_\xi^k R_i^k \qquad \text{for every } i \notin N^1.$$

Since for those i which belong to N^2, we have $\xi_i^k > 0$ and $R_i^k < 0$, the gain for updating ξ can not be increased arbitrarily, therefore, the purpose for having equations (7.34) and (7.35) is to calculate the upper bound for the gain. Hence, the possibility of having a negative value of ξ_i^{k+1} is totally excluded.

Then

$$\epsilon_\xi^k = \begin{cases} \min\{\epsilon_\xi, \bar{\epsilon}^k_\xi\} & \text{if } N^2(v^k, w^k, \xi^k, p^k) \text{ is not empty} \\ \epsilon_\xi & \text{if } N^2(v^k, w^k, \xi^k, p^k) \text{ is empty.} \end{cases} \tag{7.35}$$

ϵ_v, ϵ_w, ϵ_ξ and ϵ_p are prescribed gain parameters. The purpose of introducing $N^2(v^k, w^k, \xi^k, p^k)$ is to exclude the possibility that ξ_i^{k+1} becomes negative. The iteration is terminated when:

$$\hat{c}^k = v^k \quad , \quad \hat{u}^k = w^k \quad , \quad \xi^{k+1} = \xi^k \quad \text{and} \quad p^{k+1} = p^k.$$

In the model optimisation problem, the output dependent constraints $g(c, u, \alpha) \leq 0$ are excluded from the constraint set of (MOP). This simplifies the model based optimisation, especially when there are no local constraints $f(c,u) \leq 0$ because in that case (MOP) becomes an unconstrained problem.

The influence of the output dependent constraints is taken into account by introducing the extra terms $[\nabla_v g(v^k, w^k, \alpha^k)\, \xi^k]^t c$ and

$[\nabla_u g(v^k, w^k, \alpha^k)\ \xi^k]^t u$ into the performance index of (MOP). The terms $[\nabla_v g(v^k, w^k, \alpha^k)\ \xi^k]$ and $[\nabla_u g(v^k, w^k, \alpha^k)\ \xi^k]$ can be viewed as additional modifiers.

In the above algorithm, the constraints of the model based optimisation problem do not change during the course of iteration, since they are not dependent on the parameter α (see Chen, Brdyś and Roberts, 1986). Therefore, the possibility of having no feasible solution is excluded.

7.5 Optimality:

The optimality properties of this algorithm are investigated in this section. Define:

$$\Omega \triangleq \{ (v, w, \xi, p) : (v, w) \in \mathcal{CU},$$

$$v = \hat{c}\ (v, w, \hat{\alpha}, \xi, p), \quad w = \hat{u}\ (v, w, \hat{\alpha}, \xi, p)\ ;\ \ \xi_i \geq 0,$$

$$[T(v, w, \xi, p)(g^*(v, w) + \nabla_c\ g^*(v, w)^t(\hat{c}(v, w, \hat{\alpha}, \xi, p) - v) +$$

$$+ \nabla_u\ g^*(v, w)^t(\hat{u}(v, w, \hat{\alpha}, \xi, p) - w]_i = 0,$$

$$i = 1,...,L \qquad \text{and} \qquad W = HF(v, w, \hat{\alpha})\} \tag{7.36}$$

$$\Omega_1 \triangleq \{ (v, w) : \exists\ \xi, p \text{ such that } (v, w, \xi, p) \in \Omega \} \tag{7.37}$$

$$\Omega_2 \triangleq \{ (\xi, p) : \exists\ v, w \text{ such that } (v, w, \xi, p) \in \Omega \} \tag{7.38}$$

and the optimal solution set of the real optimisation problem (ROP) is denoted by Ω_1^*. According to the definition of Ω, it is easily seen that Ω is the algorithm solution set.

Theorem 1: Let assumptions A1, A2 and A3 (in sections 6.3.1) be satisfied, and assume that every $(v, w) \in \Omega_1$ is a regular point of the constraints $f(v, w) \leq 0$. Then for every $(v, w, \xi, p) \in \Omega$ there exists a η such that (v, w, ξ, p, η) satisfies the Kuhn-Tucker conditions of (ROP).

Proof:

For $(v, w, \xi, p) \in \Omega$, $\hat{c}(v, w, \hat{\alpha}, \xi, p)$, $\hat{u}(v, w, \hat{\alpha}, \xi, p)$ is the optimal solution of (MOP). Since $(\hat{c}, \hat{u})$ is a regular point (definitions in Chapter 3) of the constraint $f(\hat{c}, \hat{u}) \leq 0$, there exists an η; which is the Lagrange multiplier vector associated with the local constraints $f(c, u) \leq 0$; such that:

from differentiation of (MOP) w.r.t. c:

$$\nabla_c\, q(\hat{c}, \hat{u}, \hat{\alpha}) - \lambda(v, w, \hat{\alpha}, \xi, p) + \nabla_c\, g(v, w, \hat{\alpha})\, \xi -$$

$$\nabla_c\, F(\hat{c}, \hat{u}, \hat{\alpha})\, H^t p + \nabla_c f\,(\hat{c}, \hat{u})\, \eta \;=\; 0 \qquad (7.39)$$

from differentiation of (MOP) w.r.t. u:

$$\nabla_u\, F(\hat{c}, \hat{u}, \hat{\alpha}) - t(v, w, \hat{\alpha}, \xi, p) + \nabla_u\, g(v, w, \hat{\alpha})\, \xi +$$

$$(I - \nabla_u\, F(\hat{c}, \hat{u}, \hat{\alpha})\, H^t)p + \nabla_u\, f(\hat{c}, \hat{u})\, \eta = 0 \qquad (7.40)$$

$$\eta_i\, f_c(\hat{c}, \hat{u}) = 0, \quad \eta_i \geq 0 \quad \text{and} \quad f_t(\hat{c}, \hat{u}) \leq 0, \quad i = 1,.., N \qquad (7.41)$$

The formulation of the modifier λ in equation (31) can be rewritten as:

$$\lambda(v, w, \alpha, \xi, p) = \nabla_v\, F(v, w, \alpha).\, \nabla_y Q(v, w, y) -$$

$$+ \nabla\, K^*(v) \cdot \nabla_y\, Q(v, w, y)\;\; \nabla_v\, F(v, w, \alpha)\, .\, \nabla_y\, G(v, w, y)\, \xi -$$

$$+ \nabla\, K^*(v)\, .\, \nabla_y\, G(v, w, y)\, \xi - \nabla_v\, F(v, w, \alpha)\, H^t p +$$

$$+ \nabla\, K^*(v)\, H^t p \qquad (7.42)$$

Since $F(v, w, \hat{\alpha}) = K^*(v)$, then

$$\lambda(v, w, \alpha, \xi, p) = \nabla_c\, q(v, w, \hat{\alpha}) - \nabla_c\, q^*(v, w) + \nabla_c\, g(v, w, \hat{\alpha})\, \xi -$$

$$\nabla_c\, g^*(v, w)\, \xi - \nabla_c\, F(v, w, \hat{\alpha})\, H^t\, p + \nabla\, K^*(v)\, H^t\, p \qquad (7.43)$$

Similarly, the formulation of the modifier t in equation (7.24) can be rewritten as:

$$t(v, w, \hat{\alpha}, \xi, p) = \nabla_w\, F(v, w, \alpha)\, .\, \nabla_y\, Q(v, w, y) + \nabla_w\, F(v, w, \alpha)\, .$$

$$\nabla_y\, G(v, w, y)\, \xi - \nabla_w\, F(v, w, \alpha)\, H^t p \qquad (7.44)$$

where $y = F(v, w, \hat{\alpha})$

$$t(v, w, \hat{\alpha}, \xi, p) = \nabla_u q(v, w, \hat{\alpha}) + \nabla_u g(v, w, \alpha) -$$

$$+\nabla_u F(v, w, \hat{\alpha})\, H^t p - \nabla_u q^*(v, w) - \nabla_u g^*(v, w)\, \xi \qquad (7.45)$$

Substituting (7.44), (7.45) into (7.39), (7.40) and using $\hat{c} = v$, $\hat{u} = w$, equations (7.39), (7.41) can be written as:

$$\nabla_c q^*(v, w) + \nabla_c g^*(v, w)\, \xi - \nabla K^*(v)\, H^t p + \nabla_c f(v, w)\, \eta = 0 \qquad (7.46)$$

$$\nabla_u q^*(v, w) + \nabla_u g^*(v, w)\, \xi + p + \nabla_u f(v, w)\, \eta = 0 \qquad (7.47)$$

$$\eta_i f_i(v, w) = 0, \quad \eta_i \geq 0 \quad \text{and} \quad f_i(v, w) \leq 0 \qquad i = 1, \ldots, N \qquad (7.48)$$

According to the definition of Ω, v, w and ξ satisfy:

$$\xi_i [g^*(v, w)]_i = 0, \quad \xi_i \geq 0 \quad \text{and} \quad [g^*(v, w)]_i \leq 0 \qquad i = 1, \ldots, L \qquad (7.49)$$

Examining equations (7.46) to (7.49), they are exactly the Kuhn-Tucker conditions of (ROP) at (v, w).

Theorem 1 Shows that every $(v, w) \in \Omega_1$ satisfies the necessary conditions for optimality. If these necessary conditions are also sufficient, for example, if (ROP) is a convex problem, then $\Omega_1 \subset \Omega_1^*$ is ensured. If, however, the problem is such that there is no equivalence between the necessary and sufficient conditions for optimality, it is preferred that the solutions of (ROP) belong to Ω_1 so that any optimum solution of the original problem can not be missed. The following theorem gives sufficient conditions for $\Omega_1^* \subset \Omega_1$.

Theorem 2: Let assumptions A1, A2 and A3 (Chapter 6) be satisfied, and assume:

1) every $(v, w) \subset \mathcal{CU}$ is a regular point of the constraints $g^*(v,w) \leq 0$ and $f(v, w) \leq 0$;

2) for every $\hat{\alpha} \in \phi$, where $\phi \triangleq \{\alpha : (v, w) \in \mathcal{CU}$ and $F(v, w, \alpha) = K^*(v)\}$, $q(c, u, \alpha)$ is convex on $\mathcal{CU}$.

Then $\Omega_1^* \subset \Omega_1$.

Proof:

For $(v^*, w^*) \in \Omega_1^*$, since (v^*, w^*) is a regular point of the constraints $g^*(v, w) \leq 0$ and $f(v, w) \leq 0$, there exists p^*, ξ^*, η^* such that:

$$\nabla_c q^*(v^*, w^*) + \nabla_c g^*(v^*, w^*) \xi^* - \nabla K^*(v) H^t p^* + \nabla_c f(v^*, w^*) \eta^* = 0 \tag{7.50}$$

$$\nabla_u q^*(v^*, w^*) + \nabla_u g^*(v^*, w^*) \xi^* + p + \nabla_u f(v^*, w^*) \eta^* = 0 \tag{7.51}$$

$$\eta_i^* f_i(v^*, w^*) = 0, \quad \eta_i^* \geq 0 \quad \text{and} \quad f_i(v^*, w^*) \leq 0$$

$$i = 1, \ldots, N \tag{7.52}$$

$$\xi_i^* [g^*(v^*, w^*)]_i = 0, \quad \xi_i^* \geq 0 \quad \text{and} \quad [g^*(v^*, w^*)]_i \leq 0$$

$$i = 1, \ldots, L \tag{7.53}$$

Similar to the case of Theorem 1, differentiating the Lagrangian w.r.t. set point c gives:

$$\nabla_c q^*(v^*, w^*, \hat{\alpha}^*) - \lambda(v^*, w^*, \hat{\alpha}^*, \xi^*, p^*) + \nabla_c g(v^*, w^*, \hat{\alpha}^*) \xi^* - \nabla_c F(v^*, w^*, \hat{\alpha}^*) H^t p^* + \nabla_c f(v^*, w^*) \eta^* =$$

$$\nabla_c q^*(v^*, w^*) + \nabla_c g^*(v^*, w^*) \xi^* - \nabla K^*(v^*) H^t p^* + \nabla_c f(v^*, w^*) \eta^* = 0 \tag{7.54}$$

Differentiation of the Lagrangian w.r.t. u gives:

$$\nabla_u\, q(v^*, w^*, \hat{\alpha}^*) - t(v^*, w^*, \hat{\alpha}^*, \xi^*, p^*) + \nabla_u\, g(v^*, w^*, \hat{\alpha}^*)\, \xi^* +$$

$$(I - \nabla_u\, F(v^*, w^*, \hat{\alpha}^*)\, H^t)\, p^* + \nabla_u\, f(v^*, w^*)\, \eta^* =$$

$$\nabla_u\, q^*(v^*, w^*) + \nabla_u\, g^*(v^*, w^*)\, \xi^* + p^* + \nabla_u\, f(v^*, w^*)\, \eta^* \qquad (7.55)$$

where $\hat{\alpha}^* = \hat{\alpha}(v^*, w^*)$.

Because the (MOP) is a convex problem, equations (7.42) to (7.45) imply that its optimal solution for given $\hat{\alpha}^*$, $\lambda(v^*, w^*, \hat{\alpha}^*, \xi^*, p^*)$ and $t(v^*, w^*, \hat{\alpha}^*, \xi^*, p^*)$ is

$$\begin{cases} \hat{c}\,(v^*, w^*, \hat{\alpha}^*, \xi^*, p^*) = v^* \\ \\ \hat{u}\,(v^*, w^*, \hat{\alpha}^*, \xi^*, p^*) = w^* \end{cases} \qquad (7.56)$$

Equations (7.56) and (7.53) guarantee that $(v^*, w^*, \xi^*, p^*) \in \Omega$, that is

$$(v^*, w^*) \in \Omega_1. \qquad \text{Q.E.D.}$$

The convergence analysis of this algorithm is given by Lin, Hendawy and Roberts, (1988a).

7.6 Simulation study:

7.6.1 Simulation examples:

The following three examples have been used in computer simulation studies to investigate the convergence properties of the algorithm.

Example 1

This example consists of two subsystems as shown in figure (7.2). The model and reality equations respectively are:

$$y_{11} = F_{11}(c_1, u_1, \alpha_{11}) = c_{11} - c_{12} + 2\,u_{11} + \alpha_{11}$$

$$y_{21} = F_{21}(c_2, u_2, \alpha_{21}) = c_{21} - c_{22} + u_{21} + \alpha_{21}$$

$$y_{22} = F_{22}(c_2, u_2, \alpha_{22}) = 2\,c_{22} - c_{23} - u_{21} + \alpha_{22}$$

$$y_{11}^* = F_{11}^*(c_1, u_1) = 1.4\,c_{11} - 0.6\,c_{12} + 1.8\,u_{11}$$

$$y_{21}^* = F_{21}^*(c_2, u_2) = 1.3\,c_{21} - 1.1\,c_{22} + 1.1\,u_{21}$$

$$y_{22}^* = F_{22}^*(c_2, u_2) = 2.3\,c_{22} - 0.7\,c_{23} - 1.1\,u_{21}$$

The subsystem performance indices are:

$$Q_1(c_1, u_1, y_1) = (y_{11} - 1)^2 + 5\,(c_{11}^2 - c_{21}^2) + 5\,u_{11}^2$$

$$Q_2(c_2, u_2, y_2) = 2\,(y_{21} - 2)^2 + (y_{22} - 3)^2 + 5\,(c_{21}^2 + c_{22}^2 + c_{23}^2) + 5\,u_{21}^2$$

The system constraints are:

$$\mathcal{CUY}_1 = \{(c_1, u_1, y_1) \in \mathfrak{R}^4,\ |c_{11}| \leq 1,\ |c_{12}| \leq 1,\ \ 0.8 - c_{12} - 0.6\,u_{11} \geq 0\}$$

$$\mathcal{CUY}_2 = \{(c_2, u_2, y_2) \in \mathfrak{R}^6,\ |c_{22}| \leq 1,\ |c_{23}| \leq 1,$$

$$2.04 + 1.05\,u_{21}^2 - c_{21}^2 - c_{22}^2 - c_{23}^2 \geq 0\}$$

The output dependent constraints are:

$$y_{11} \geq 0 \quad , \quad y_{21} \geq 0 \quad \text{and} \quad y_{22} \geq 0$$

Finally, the structure equation is:

$$\begin{bmatrix} u_{11} \\ u_{21} \end{bmatrix} - \begin{bmatrix} 0 & 1 & 0 \\ 1 & 0 & 0 \end{bmatrix} \begin{bmatrix} y_{11} \\ y_{21} \\ y_{22} \end{bmatrix}$$

Example 2

This example consists of three subsystems, the structure of the subsystems is shown in Figure (7.3).

The model equations and real system equations are :

$$y_{11} = F_{11}(c_1, u_1, \alpha_{11}) = c_{11} - c_{12} + 2\, u_{11} + \alpha_{11}$$

$$y_{21} = F_{21}(c_2, u_2, \alpha_{21}) = c_{21} - c_{22} - 3\, u_{22} + \alpha_{21}$$

$$y_{22} = F_{22}(c_2, u_2, \alpha_{22}) = 2\, c_{22} - c_{23} - u_{21} + u_{22} + \alpha_{22}$$

$$y_{31} = F_{31}(c_3, u_3, \alpha_{31}) = c_{31} + 2.5\, c_{32} - 4\, u_{31} + \alpha_{31}$$

$$y_{11}^{*} = F_{11}^{*}(c_1, u_1) = 1.3\, c_{11} - c_{12} + 2\, u_{11} + 0.15\, u_{11}c_{11}$$

$$y_{21}^{*} = F_{21}^{*}(c_2, u_2) = c_{21} - c_{22} + 1.2\, u_{21} - 3\, u_{22} + 0.1\, c_{22}^2$$

$$y_{22}^{*} = F_{22}^{*}(c_2, u_2) = 2\, c_{22} - 1.25\, c_{23} - u_{21} + u_{22} + 0.25\, c_{22}c_{23} + 0.1$$

$$y_{31}^{*} = F_{31}^{*}(c_3, u_3) = 0.8\, c_{31} + 2.5\, c_{32} - 4.2\, u_{31}$$

Performance indices of the subsystems are:

$$Q_1(c_1, u_1, y_1) = (u_{11} - 1)^4 + 5\,(c_{11} - 1)^2 + 0.1\,(y_{11} - 1)^2$$

$$Q_2(c_2, u_2, y_2) = 4u_{21}^2 + u_{22}^2 + 2\,(c_{21} - 2)^2 + c_{22}^2 + 3\, c_{23}^{\,2} + 0.1\,(y_{21} - 2)^2$$

$$Q_3(c_3, u_3, y_3) = (u_{31} - 1)^2 + (c_{31} + 1)^2 + 2.5\, c_{32}^2 + 0.1\,(y_{31} - 1)^2$$

Subsystem constraints:

$$\mathcal{CU}_1 = \{\, (c_1, u_1) \in \mathfrak{R}^3 : \; c_{11}^2 + c_{12}^2 \leq 1,\; 0 \leq u_{11} \leq 0.5 \,\}$$

$$\mathcal{CU}_2 = \{ (c_2, u_2) \in \mathfrak{R}^5 : \; c_{21}^2 + c_{22}^2 + c_{23}^2 + 0.5\, u_{11}^2 + u_{21}^2 \leq 0.8 \} \quad \text{and}$$

$$\mathcal{CU}_3 = \{ (c_3, u_3) \in \mathfrak{R}^3 : \; c_{31} + u_{31} + 1 \geq 0, \;\; 0 \leq c_{31} \leq 1 \}$$

The output dependent constraint is:

$$0.5\, y_{11} + y_{21} - y_{22} \leq 3$$

Finally, the coupling equation is:

$$\begin{bmatrix} u_{11} \\ u_{21} \\ u_{22} \\ u_{31} \end{bmatrix} = \begin{bmatrix} 0 & 1 & 0 & 0 \\ 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} y_{11} \\ y_{21} \\ y_{22} \\ y_{31} \end{bmatrix}$$

Example 3

This example consists of three subsystems. The structure of the example is the same as that of example 2, Fig(7.3) except that the subsystem constraints are of the form:

$$\mathcal{CU}_1 \triangleq \{ (c_1, u_1) \in \mathfrak{R}^3 \; ; \;\; c_{11}^2 + c_{21}^2 \leq 1, \quad 0 \leq u_{11} \leq 2.5 \}$$

$$\mathcal{CU}_2 \triangleq \{ (c_2, u_2) \in \mathfrak{R}^5 \; ; \; c_{21}^2 + c_{22}^2 + c_{23}^2 \leq 0.5 \}$$

$$\mathcal{CU}_3 \triangleq \{ (c_3, u_3) \in \mathfrak{R}^3 \; ; \;\; c_{31}^2 + c_{32}^2 \geq 1, \quad 0 \leq c_{32} \leq 1 \}$$

and the output dependent constraint is:

$$0.5\, y_{11} + y_{21} - y_{22} + y_{31} + 0.1\, y_{31}^2 \leq 3$$

These examples are chosen such that the performance indices and some of the constraints are output dependent to illustrate the behaviour of the algorithm.

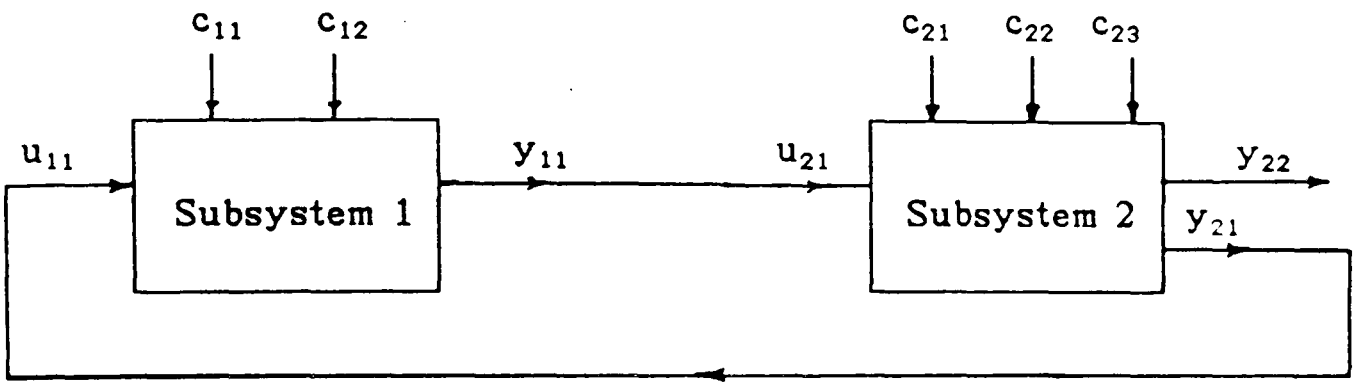


Figure (7. 2) The Structure of the System of Example 1.

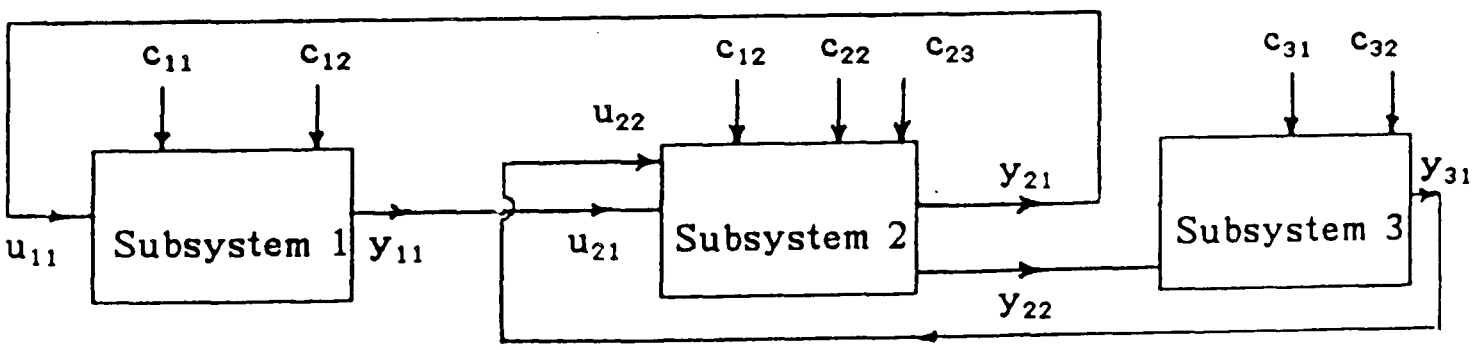


Figure (7. 3) The Structure of the System of Examples 2 and 3.

7.6.2 Simulation results:

The previous three examples were used in a simulation study to investigate the behaviour of the algorithm. The study was carried out using a Prime 750 computer. The NAG library routine E04VBF was used in the optimisation process which uses a sequential augmented Lagrangian method (Gill and Murray, 1974; Murray, 1976).

At the start of the iteration process, all initial conditions of the vectors which were required at that stage were all set to zero. The iterative process is terminated when the stopping criteria are fulfilled within the desired accuracy, which are:

$$\| p^{k+1} - p^k \| = \| \hat{u}^k - HK^*(v^k) - \nabla K^*(v^k) H^t(\hat{c}^k - v^k) \| < 2\times10^{-4}$$

and $\|\xi^{k+1} - \xi^k\| = \| T(v^k, w^k, \xi^k, p^k)\, R(v^k, w^k, \xi^k, p^k) \| < 2\times10^{-4}$

where $R(v^k, w^k, \xi^k, p^k)$ and $T(v^k, w^k, \xi^k, p^k)$ are defined by equations (7.30) and (7.31) respectively. The Euclidean norm was used in each criterion.

The behaviour and convergence of the algorithm is shown in Table (7.1) and Figures (7.4) to (7.6), where ΔQ is the deviation of performance index from the optimal, Δp, Δv and Δw are the deviations from the optimal price, optimal set point values and optimal inputs respectively. These deviations are assigned as:

$$\Delta Q = Q^k - Q^*$$

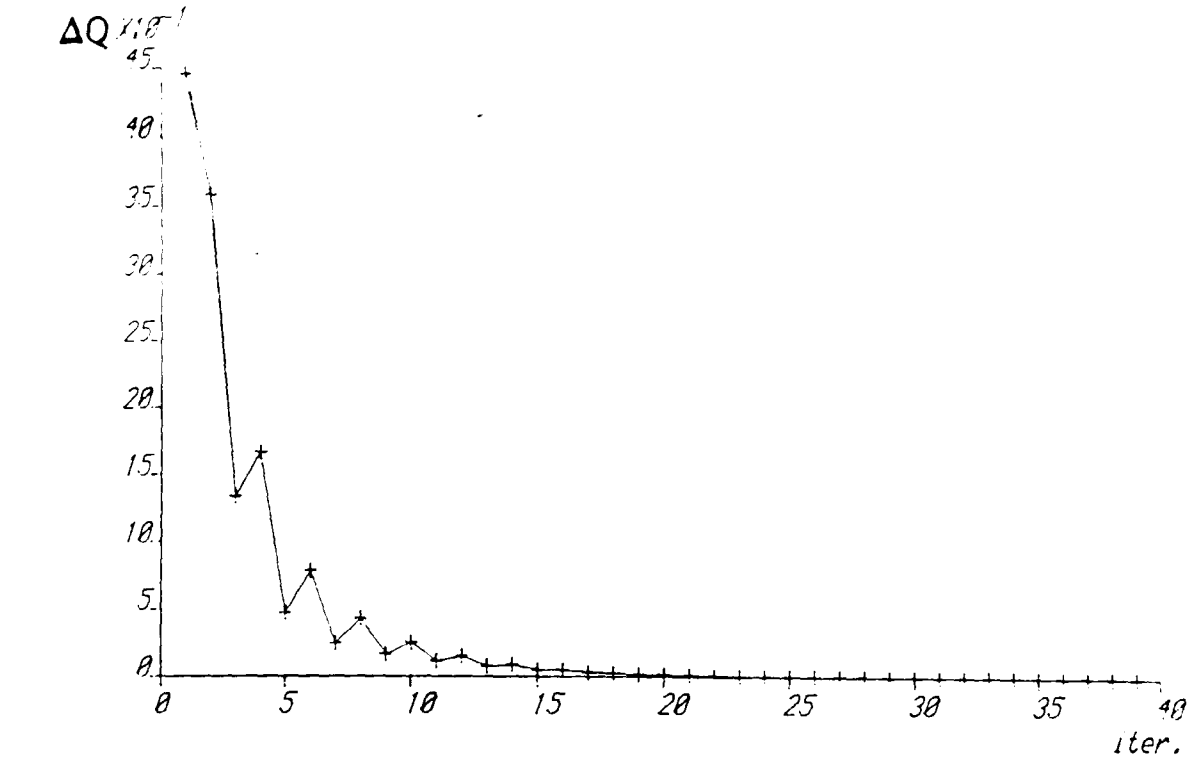
$$\|\Delta p\| = \left[\frac{1}{N}\sum_{i=1}^{N}(p_i^k - p_i^*)^2\right]^{\frac{1}{2}}$$

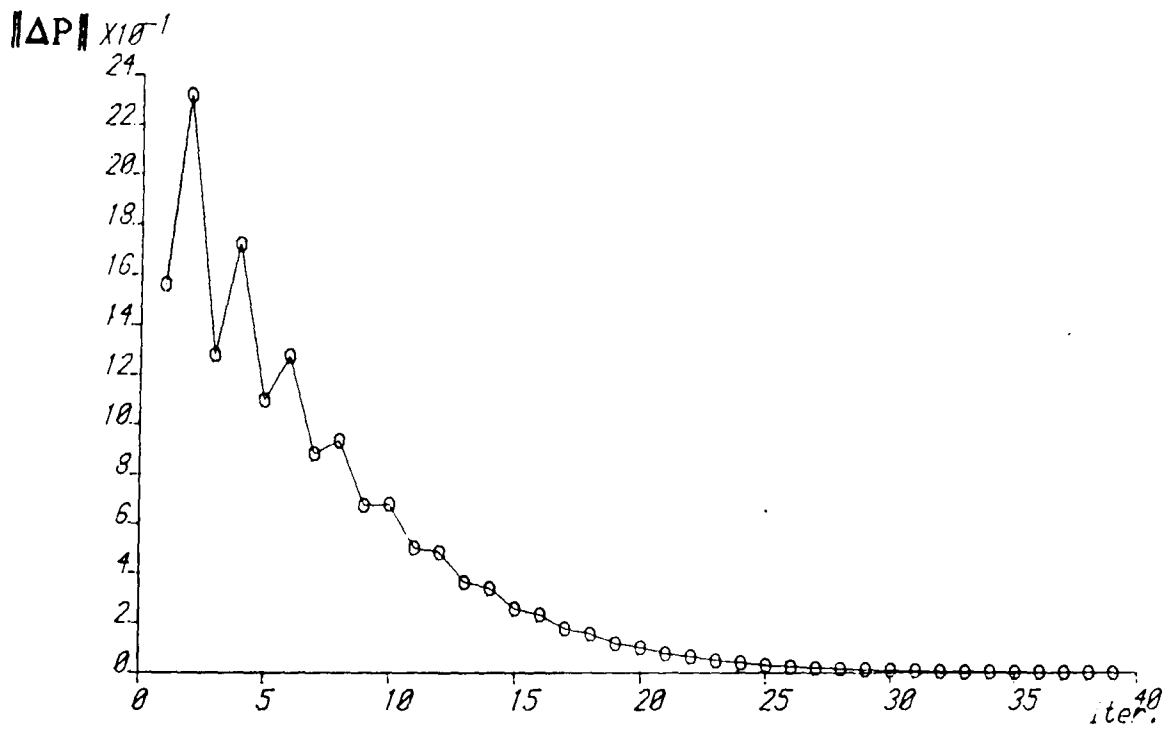
$$\|\Delta v\| = \left[\frac{1}{N}\sum_{i=1}^{N}(v_i^k - v_i^*)^2\right]^{\frac{1}{2}}$$

$$\|\Delta w\| = \left[\frac{1}{N}\sum_{i=1}^{N}(w_i^k - w_i^*)^2\right]^{\frac{1}{2}}$$

EX.	ϵ_w	ϵ_ξ	ϵ_v	ϵ_p	Iter.	Q	Q^*
1	0.4	0.40	0.22	0.66	39	13.54860	13.5489
	0.4	0.50	0.22	0.66	39	13.54860	
2	0.75	0.50	0.5	0.88	44	6.92052	6.9206
	0.75	0.60	0.5	0.88	44	6.92052	
3	0.8	0.50	0.5	0.88	44	6.43886	6.4389
	0.8	0.55	0.5	0.88	44	6.43886	

Table (7 . 1) **Convergence behaviour of the algorithm for Examples 1, 2 and 3.**





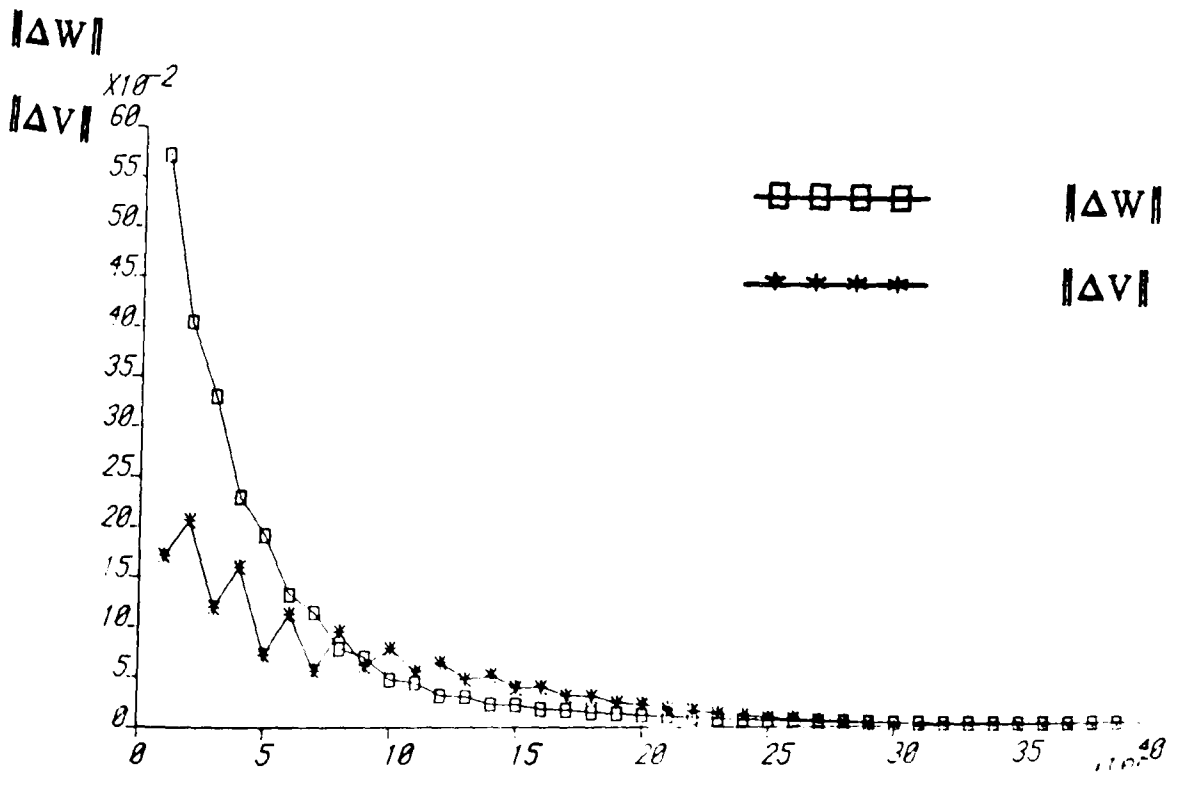
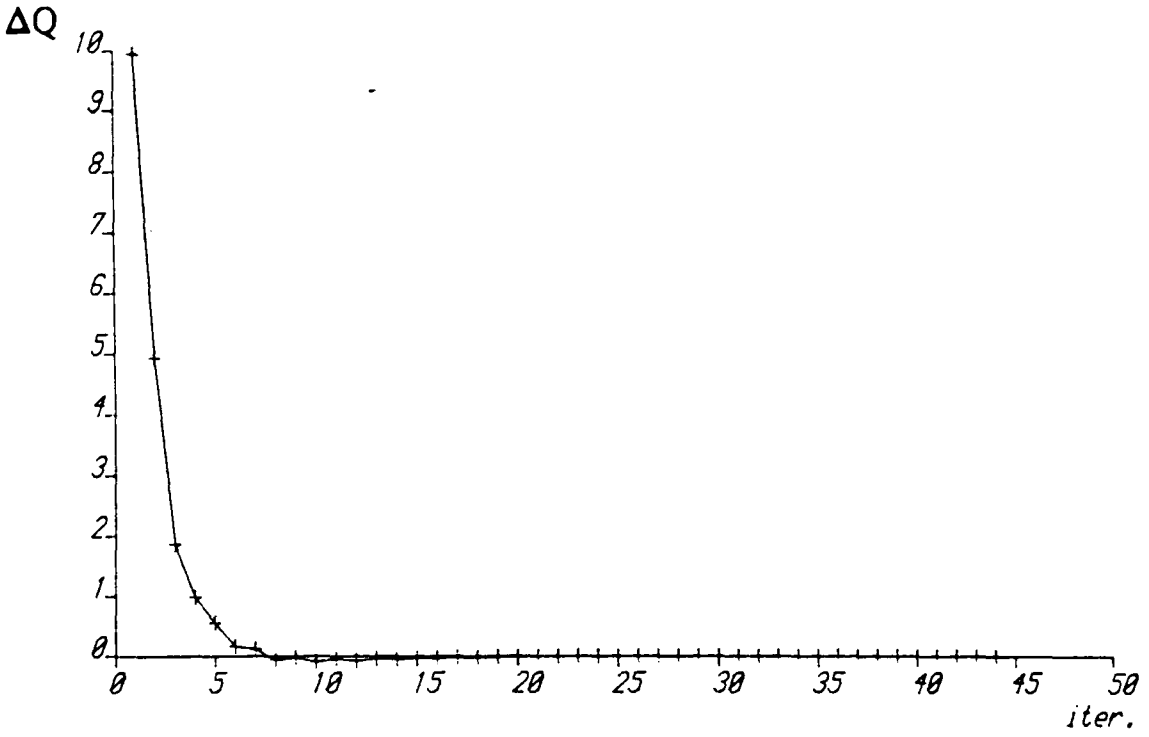
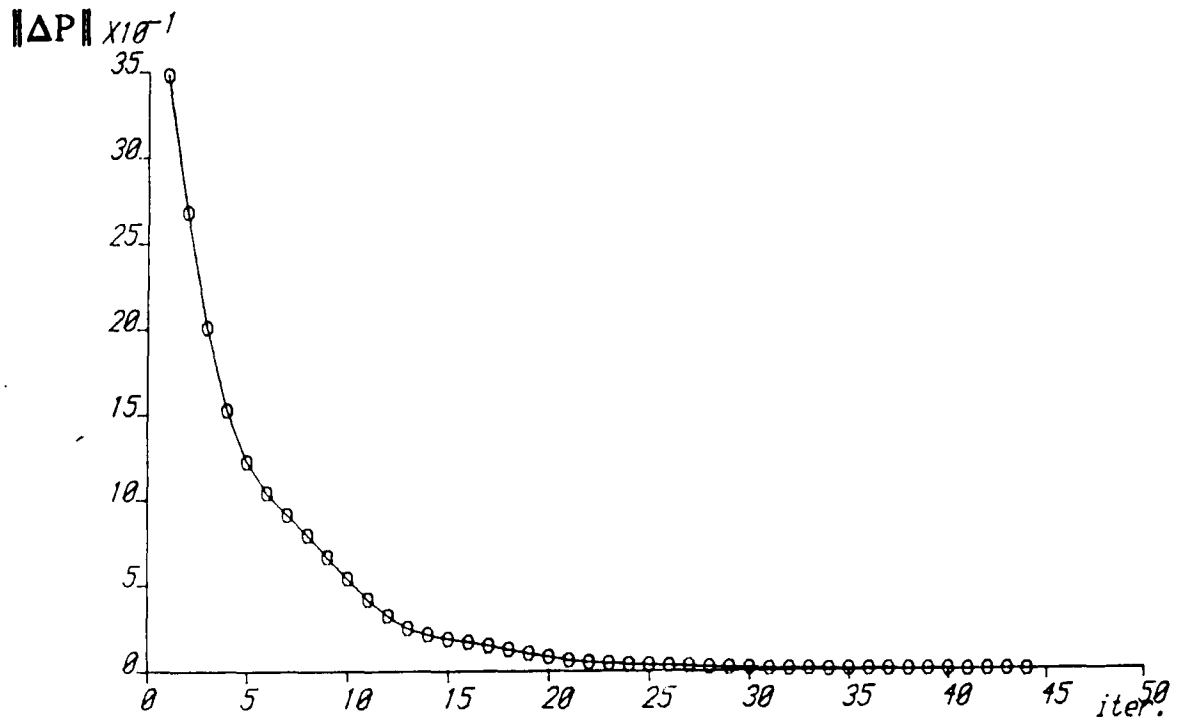


Figure (7.4) Convergence behaviour of SIA for Example 1.





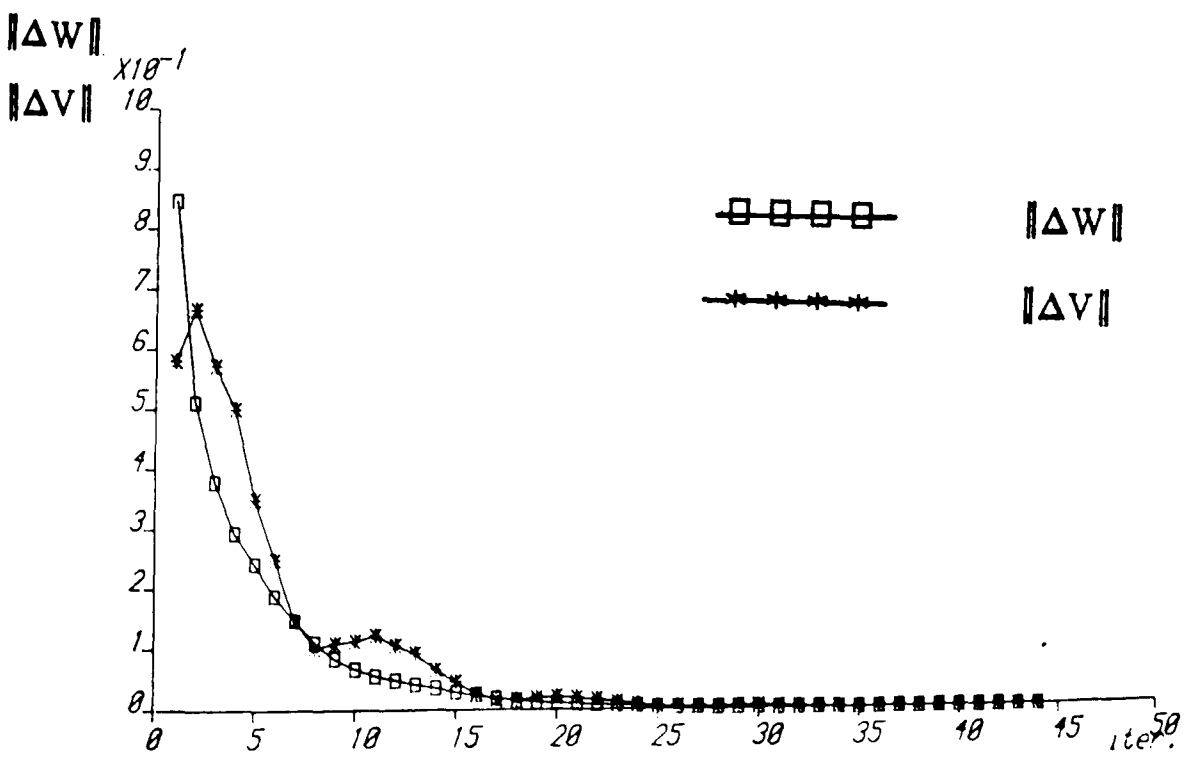
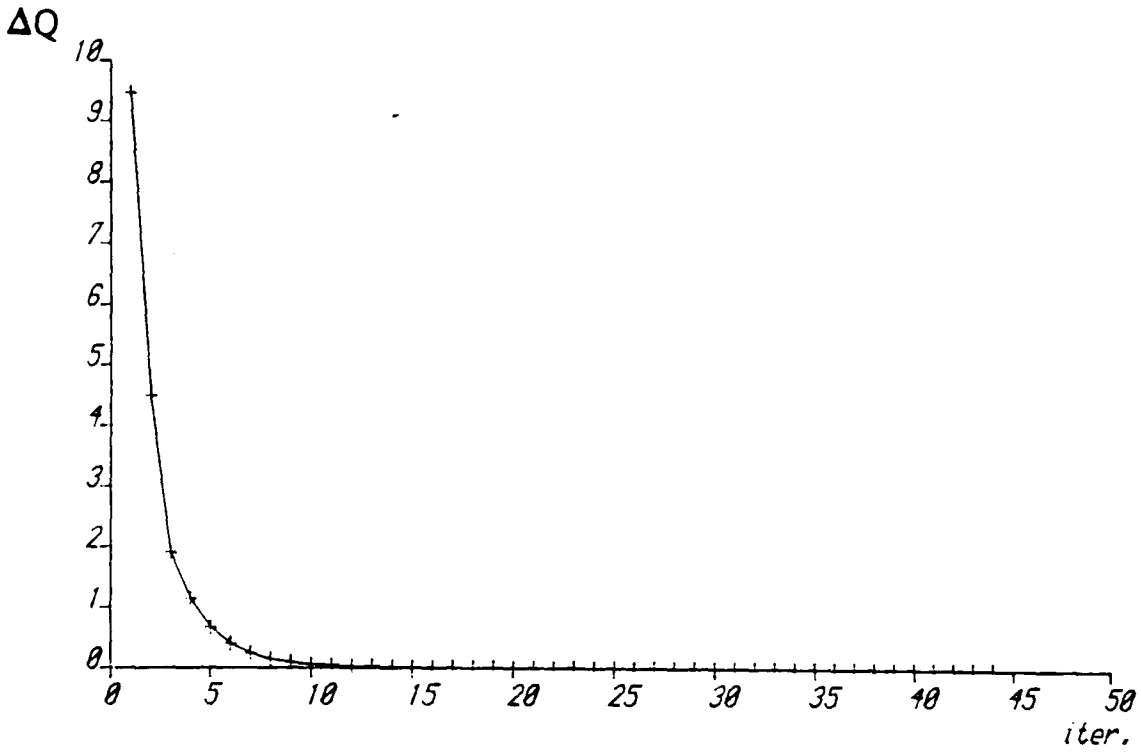
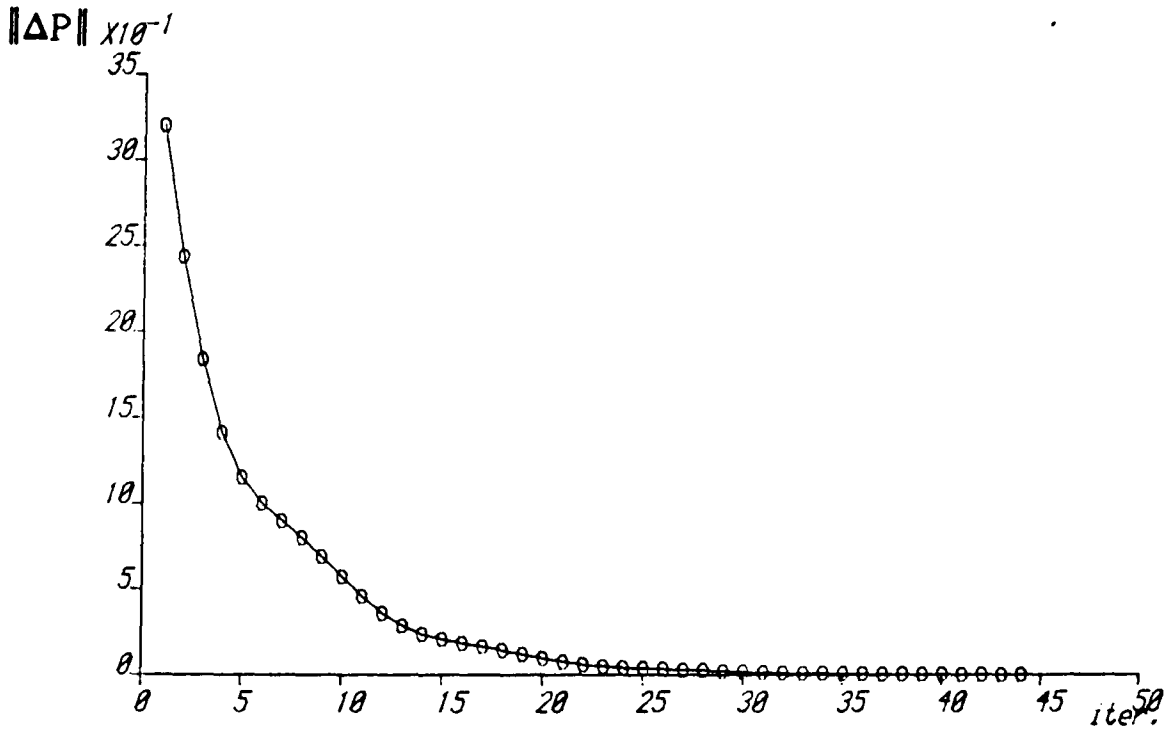


Figure (7.5) Convergence behaviour of SIA for Example 2.





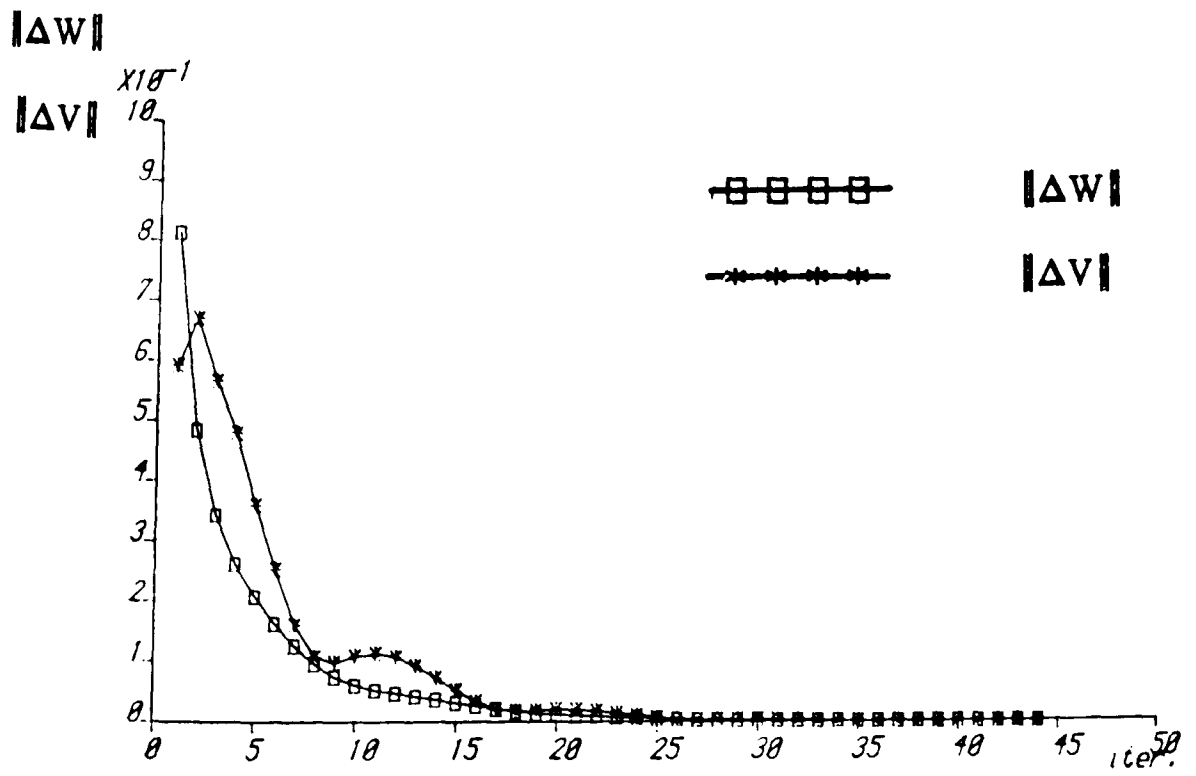


Figure (7.6) Convergence behaviour of SIA for Example 3.

It can be seen that for the particular examples the algorithm converges and achieves real optima. The results of examples two and three show that the convergence behaviour of the algorithm is excellent.

The deviations of the performance index from the optimal solutions are monotonically decreasing except in the case of example one. The convergence behaviour of example one, which is not strictly decreasing, may be due to the fact that the associated convergence theorem of the algorithm gives only sufficient local convergence conditions (Lin, Hendawy and Roberts, 1988a).

The zigzag behaviour associated with the convergence of Δp, Δv and Δw in example one and Δv in examples two and three may be due to the same reason or the used gain values might be too high.

Comparing the convergence rate of ΔQ and Δp, it can be seen, from the simulation results of the shown examples, that the convergence of ΔQ is much faster than Δp. Therefore, double loop algorithms should be much more efficient than single loop ones, in relation to set point changes to the real system.

7.7 Summary:

In this chapter, an algorithm has been introduced, which is a new version of the integrated system optimisation and parameter estimation technique. It is an extension of the algorithm by Lin, Chen and Roberts (1986) to the hierarchical case.

In this algorithm, new modifiers are introduced in the model optimisation problem to cater for the output dependent constraints. As mentioned previously, an attempt was introduced to take output dependent constraints in consideration by Chen, Brdyś and Roberts (1986). However, the feasibility of the solution of the model optimisation problem of that algorithm (Chen, Brdyś and Roberts, 1986) is not ensured during the course of iteration and the algorithm's convergence conditions have not been derived.

In this new algorithm, which is described in this chapter, the feasibility of the solution of the model based optimisation problem is ensured. The simulation results exhibit excellent convergence behaviour of this algorithm.

CHAPTER 8

Modified model-based double loop iterative strategy for ISOPE

8.1 Introduction

When considering hierarchical optimal control of large interconnected steady state systems, the difference between model and reality seem to be the crucial difficulty. Therefore it has been recognised that the model must adapt itself to some extent in order to overcome the model reality differences. This yields the idea of integrated system optimisation and parameter estimation.

As far as on-line steady state optimal control is concerned, the number of set point changes of feedback controllers remains to be a major problem before any elaborate algorithm can be put into application. Efforts have been made to reduce the set point changes (Brdys' and Roberts 1985; Chen, Brdys' and Roberts 1986; Brdys' et.al. 1986a) and have proved to be very successful. Most of these techniques are based in the strategy of splitting the coordination task into two nested iterative loops. The inner loop involves model-based optimisation only while the outer loop requires measurements from the real process.

However, the new double loop technique which is presented in this chapter has important advantages over several other existing methods. In the previous model-based double loop techniques (Brdys' and Roberts 1985;

Chen et. al. 1986; Brdys' et.al 1986a) the derivatives of the model used in the inner loop optimisation are fixed, whereas in this new double loop iterative strategy, derivatives of real process measurements are used to update the model. So, before entering into the inner loop, the derivatives of the real process are available. Therefore, it is possible to linearize the model by using these derivatives . This is also an attempt to combine the approximate linear model approach (Ellis et.al 1985) with the modified two-step method in the decentralised case.

This algorithm has another advantage which is that it can be applied to a more general class of problems in which the inequality constraints may be output dependent. To cope with the output dependent constraints, this algorithm uses the technique given by Lin, Hendawy and Roberts, 1988a. Another advantage for this new double loop method over most of the previous techniques (Brdys' et.al 1986a; Brdys' et.al 1986b Brdys' et.al 1987) is that the convergence conditions of this algorithm are much easier to satisfy than the mentioned previous techniques.

8.2 Formulation of the algorithm

Assumption 1

It is assumed throughout the analysis of the algorithm that:

a) Mappings G(c, u,y), f(c, u), F(c, u, y), $K^*(c)$ and the objective function Q(c, u, y) are continuously Frechet differentiable.

b) Mappings G(c, u, y) and f(c, u) are convex.

c)The model input-output mapping is point parametric on C^2U (Brdys' 1983)

The task of determining the optimal operating condition for a real process can be defined as the following real steady-state optimising control (ROP):

$$\min_{c,\,u,\,y} Q(c, u, y)$$

s.t.

$$y = K^*(c)$$
$$u = Hy \qquad \text{(ROP)}$$
$$G(c, u, y) \leq 0$$
$$(c, u) \in \mathcal{CU}$$

It has been proved by Brdys', 1983 that under very mild assumptions the (ROP) is equivalent to the optimisation problem (EOP). These conditions are also summarized in the previous chapter.

$$\min_{c,\,u,\,\alpha} q(c, u, \alpha)$$

s.t.

$$F(c,\ H\,K^*(c), \alpha) = K^*(c)$$
$$u - HK^*(c) \qquad \text{(EOP)}$$
$$g(c, u, \alpha) \leq 0$$
$$(c, u) \in \mathcal{CU}$$

where

$$q(c, u, \alpha) \triangleq Q\,(c, u, F(c, u,\alpha))$$

$$g(c, u, \alpha) \triangleq G\,(c, u, F(c, u,\alpha))$$

In this algorithm, the c dependent constraints are separable from the constraints set $\mathcal{CU}$. By using the interaction relation u = Hy, the constraints which depend on both c and u can be converted to output dependent constraints. Therefore , we always assume that the local constraints are only c dependent.

In order to separate the optimisation and parameter estimation problems an additional equation, v=c, is introduced. The problem EOP can be written as follows:

$$\min_{c,\, u,\, \alpha} \; q(c,\, u,\, \alpha)$$

s.t.

$$F(v, H\, K^*(v), \alpha)\,) = K^*(v)$$
$$u = HK^*(c) \qquad\qquad (EOP_1)$$
$$g(c,\, u,\, \alpha) \leq 0$$
$$v = c \quad , \quad c \in \mathcal{C}$$

where $c = \{\, v : v \in \Re^n \;,\; f(v) \leq 0 \,\}$, $f : \Re^n \rightarrow \Re^i$ is convex and continuously Ferechet differentiable.

The Lagrangian associated with (EOP_1) can be written as :

$$\mathcal{L}(c\;,\; u\;,\; v\;,\; \alpha\;,\; P\;,\; \lambda\;,\; \eta\;,\; \xi) = q\;(c\;,u,\alpha) + P^t\;[\;u - HK^*\;(c)]$$

$$+\; \xi^t\; g(c,u,\alpha) + \lambda^t\;(v-c) + \eta^t\;[F(v,\, H\; K^*(v),\, \alpha) - K^*(v)] \qquad (8.2)$$

where P , λ , η , and ξ are Lagrange multipliers. In particular, P is known as the price vector, λ is the modifier vector and ξ is denoted as the Lagrange multiplier vector associated with G.

Assuming that all the required derivatives exist and regularity conditions (see chapter 3) are satisfied, the Kuhn-Tucker necessary optimality conditions of (EOP_1) are:

$$\nabla_c \mathcal{L}(\;.\;) = \frac{\partial^t q(c,u,\alpha)}{\partial c} - \frac{\partial K^*(c)}{\partial c}\; H^t P + \frac{\partial^t g(c,u,\alpha)}{\partial c}\; \xi - \lambda \;= 0 \qquad (8.3)$$

$$\nabla_u \mathcal{L}(\;.\;) = \frac{\partial^t q(c,u,\alpha)}{\partial u} + P + \frac{\partial^t g(c,u,\alpha)}{\partial u}\; \xi \;= 0 \qquad (8.4)$$

$$\nabla_v \mathcal{L}(\;.\;) - \lambda + [\; \frac{\partial^t F(v,\, HK^*(v)\;,\alpha)}{\partial v} - \frac{\partial^t K^*(v)}{\partial v}\;]\; \eta = 0 \qquad (8.5)$$

$$\nabla_\alpha\; \mathcal{L}(\;.\;) = \frac{\partial^t q(c,u,\alpha)}{\partial \alpha} + \frac{\partial^t g(c,u,\alpha)}{\partial \alpha}\; \xi + \frac{\partial^t F(v,\, HK^*(v)\;,\alpha)}{\partial \alpha}\; \eta = 0 \qquad (8.6)$$

$$\nabla_\rho\; \mathcal{L}(\;.\;) = u - H\; K^*(\;c\;) \;= 0 \qquad (8.7)$$

$$\nabla_{\lambda}\mathcal{L}(\,.\,) = v - c = 0 \tag{8.8}$$

$$\nabla_{\eta}\mathcal{L}(\,.\,) = F(v\ ,\ H\ K^{*}(v)\ ,\ \alpha)\ ) - K^{*}(v) = 0 \tag{8.9}$$

$$\nabla_{\xi}\mathcal{L}(\,.\,) = g(c,\ u\ ,\ \alpha) \leq 0 \tag{8.10}$$

$$\xi^{t}\nabla_{\xi}\mathcal{L}(\,.\,) = \xi^{t}\ g\ (c,\ u,\ \alpha) = 0 \quad , \quad \xi \geq 0 \tag{8.11}$$

To obtain a solution of equations (8.3) to (8.11), assume $\| \frac{\partial^{t}F(v,\ HK^{*}(v)\ ,\alpha)}{\partial\alpha} \|^{-1}$ exists, η is solved from equation (8.6) and substituted in equation (8.5) to derive the formula for calculating the modifier λ. That is:

$$\lambda(c,\ u,\ v,\ \alpha,\ \xi) = [\ \nabla_{v}\ F(v,\ \ H\ K^{*}(v),\ \alpha) - \nabla_{v}\ K^{*}(v)]\ [\nabla_{y}\ Q(c,\ u,\ y)$$

$$+\ \nabla_{y}G(c,\ u,\ y)\xi] \tag{8.12}$$

where

$$y = F\ (v,\ HK^{*}(v)\ ,\ \alpha).$$

In the stage of calculating λ, the vectors c and u are not available. Therefore, these vectors are replaced by v and $HK^{*}(v)$, respectively. Then, equation (8.13) can be written as:

$$\lambda(v,\ \alpha,\ \xi) = [\ \nabla_{v}\ F(v,\ \ H\ K^{*}(v),\ \alpha) - \nabla_{v}\ K^{*}(v)]\ [\nabla_{y}\ Q(v,\ HK^{*}(v),\ y)$$

$$+\ \nabla_{y}G(v,\ \ HK^{*}(v\ ,\ y)\xi] \tag{8.13}$$

Because the real process description is not known, we approximate it by a linear model:

$$u(c,\ v) = \ \ H\ K^{*}(v) + HK^{*'}(v)\ (c - v) \tag{8.14}$$

Since the global constraints are α dependent, the model based optimisation problem may fail to have a feasible solution during the course of iteration (Lin, Chen and Roberts, 1986). To overcome this, we exclude the global

constraints $g(c,u,\alpha) \leq 0$ from the model optimisation problem and take it into consideration by introducing new modifiers.

8.3 Implementation aspects of the algorithm

Given some initial v^0 inside the feasible set and some initial guess ξ^0, the on-line implementation involves solving repeatedly the following two-step procedure:

step1

Apply v^k to the real process to obtain the corresponding steady state measurement $K^*(v^k)$.

Determine $\hat{\alpha}^k = \alpha\ (v^k)$ by solving

$$F(v^k,\ H\ K^*(v^k),\ \alpha) \ =\ K^*(v^k) \tag{8.15}$$

Perform additional perturbations about v^k and measure the corresponding process outputs to compute finite difference approximations of the derivatives $\frac{\partial^t K^*(v^k)}{\partial v}$.

Finally, use equation (8.13) to calculate

$$\lambda^k = \lambda(\ v^k,\ \hat{\alpha}^k,\ \xi^k)$$

step 2:

For given $\hat{\alpha}^k$, v^k and ξ^k, solve the following model-based optimisation problem by using the price method

$$\min_{c,u} \Big\{\ q(c,\ u,\ \hat{\alpha}^k) - \lambda^t\ (\ v^k,\ \hat{\alpha}^k,\ \xi^k)\ c + [\ \nabla_c\ g\ (v^k,\ H\ K^*(v^k),\ \hat{\alpha}^k)\ \xi^k]^t\ c$$

$$+ [\ \nabla_u\ g\ \ (v^k,\ \ H\ K^*(v^k),\ \hat{\alpha}^k)\ \xi^k\ ]^t\ u\ \Big\} \qquad \text{(MOP)}$$

s.t.

$$u = H\ K^*(v^k) + HK^{*'}(v^k)\ (c - v^k)$$

$$c \in \mathbb{C}$$

The solution is denoted as $\hat{c}(\, v^k, \hat{\alpha}^k, \xi^k)$ and $\hat{u}(\, v^k, \hat{\alpha}^k, \xi^k)$. For simplicity , the minimised function will later be denoted by AA; i.e.

$$AA = \Big\{ q(c, u, \hat{\alpha}^k) - \lambda^t(\, v^k, \hat{\alpha}^k, \xi^k)\, c + [\, \nabla_c\, g\, (v^k,\ H\, K^*(v^k), \hat{\alpha}^k)\, \xi^k]^t\, c$$

$$+ [\, \nabla_u\, g\ \ (v^k,\ H\, K^*(v^k), \hat{\alpha}^k)\, \xi^k\,]^t\, u \Big\} \qquad (8.16)$$

Since the model optimisation problem (MOP) is to be solved hierarchically using the price method, it is necessary to have an inner loop iteration to achieve the interaction balance of the constraints

$$u = H\, K^*(v^k) + HK^{*'}(v^k)\, (c - v).$$

For given ξ^k, v^k and $\hat{\alpha}^k$, the inner loop coordination can be described as follows:

step 2a

For given P^i, the inner loop optimisation problem for the ith subsystem is :

$$\min_{c_i,\, u_i} \Big\{\, q_i(c_i, u_i, \hat{\alpha}_i^k) - \lambda_i^t\, c_i + \gamma_i^t\, c_i \ + \phi_i^t\, u_i \ + P_i^t u_i - d_i^t c_i \,\Big\}$$

s.t. $c_i \in C_i$;

$$\gamma_i \triangleq [\, \nabla_c\, g\ \ (v^k,\ H\, K^*(v^k), \hat{\alpha}^k)\, \xi^k\,]_i$$

$$\phi_i \triangleq [\, \nabla_u\, g\ \ (v^k,\ H\, K^*(v^k), \hat{\alpha}^k)\, \xi^k\,]_i$$

$$d_i \triangleq [(\, P^i)^t\ \ H\, K^*(v^k)\,]_i \quad , \quad \mathcal{C} = C_1 \times \ldots \times C_N$$

The solution is denoted as $\tilde{c}_i(\, v^k, \hat{\alpha}^k, \xi^k, P^i\,)$ and $\tilde{u}_i(\, v^k, \hat{\alpha}^k\ , \xi^k\ , P^i)$, or $\tilde{c}_i^{\,i}$ and $\tilde{u}_i^{\,i}$ for short.

step 2b

For given P^i, $\tilde{c}^i$ and $\tilde{u}^i$, where

$(\tilde{c}^l)^t = [\,(\tilde{c}_1{}^l)^t, \ldots, (\tilde{c}_N{}^l)^t\,]$ and

$(\tilde{u}^l)^t = [\,(\tilde{u}_1{}^l)^t, \ldots, (\tilde{u}_N{}^l)^t\,]$.

calculate

$$P^{l+1} = P^l + \epsilon_p \left(\tilde{u}^l - H K^*(v^k) - HK^{*\prime}(v^k)(\tilde{c}^l - \tilde{v}^k) \right)$$

The inner loop iteration is terminated when $P^{l+1} = P^l$.

The iterative variables v^k and ξ^k are updated according to :

$$v^{k+1} = v^k + \epsilon_v (\hat{c}^k - v^k) \tag{8.17}$$

$$\xi^{k+1} = \xi^k + \epsilon_\xi^k P(v^k, \xi^k) [\, g^*(v^k) + g^{*\prime}(v^k)(\hat{c}^l - v^k) \tag{8.18}$$

P(v, ξ) is a projection operator satisfying some conditions (Lin, Chen and Roberts, 1986), which are mainly to exclude the possibility of ξ_i^{k+1} becoming negative for a selected gain. The conditions for (v, ξ) which need to be satisfied are shown in Appendix A.

The iteration is terminated when $\hat{c}^k = v^k$ and $\xi^{k+1} = \xi^k$.

8.4 Algorithm optimality

In this section, the optimality properties of this new model-based double loop algorithm are studied. The optimal solution set of (ROP) and the algorithmic solution set are denoted respectively by Ω_1^* and Ω, where Ω is defined as:

$$\Omega \triangleq \{ (v, \xi) : \quad v \in C , \quad v = \hat{c}(v, \hat{\alpha}, \xi) ; \text{ and } \quad \xi \geq 0,$$

$$[P(v,\xi) (g^*(v) + g^{*\prime}(v) (\hat{c}-v))]_i = 0, \quad i = 1, \ldots, L \} \tag{8.19}$$

$$\Omega_1 \triangleq \{ v : \exists \xi \text{ such that } (v,\xi) \in \Omega \} \tag{8.20}$$

$$\Omega_2 \triangleq \{ \xi : \exists v \text{ such that } (v,\xi) \in \Omega \} \tag{8.21}$$

Theorem 1

Let assumption 1 and the assumptions for (ROP) and (EOP_1) be satisfied, and assume that every $v \in \Omega_1$ is a regular point of the constraints $f(v) \leq 0$. Then for every $(v, \xi) \in \Omega$ there exists a (η) such that (v, ξ, η) satisfies the Kuhn-Tucker optimality conditions of (ROP).

η is denoted as the lagrangian multiplier vector associated with the constraints f(inequality constraints depend only on controller set points).

Proof of theorem 1

For $(v, \xi) \in \Omega$, $\hat{c}(v, \hat{\alpha}, \xi)$ is the optimal solution of (MOP). Since $\hat{c}$ is a regular point of the constraints $f(c) \leq 0$, there exists a η such that:

$$\nabla_c q(\hat{c}, \hat{u}, \hat{\alpha}) + \nabla K^*(v) H^t \nabla_u q (\hat{c}, \hat{u}, \hat{\alpha}) - \lambda(v, \hat{\alpha}, \xi)$$

$$+ \nabla_c g (v, H K^*(v), \hat{\alpha}) \xi + \nabla K^*(v) H^t \nabla_u g (v, H K^*(v), \hat{\alpha}) \xi$$

$$+ \nabla f(c) \eta = 0 \tag{8.22}$$

$$\eta_i f_i(c) = 0 , \quad \eta_i \geq 0 \text{ and } \quad f_i (c) \leq 0 , \quad i = 1, \ldots, r \tag{8.23}$$

where:

$$\hat{u} = H K^{*}(v) + HK^{*\prime}(v) (\hat{c} - v)$$

Since $F(v, H K_{*}(v), \hat{\alpha}) = K_{*}(v)$, then equation (8.13) for updating the modifier λ can be rewritten as:

$$\lambda(v, \alpha, \xi) = \nabla_c q(v, H K^{*}(v), \hat{\alpha}) + \nabla K^{*}(v) H^{t} \nabla_u q(v, H K^{*}(v), \hat{\alpha})$$

$$- \nabla q^{*}(v) \nabla_c g(v, H K^{*}(v), \hat{\alpha}) + \nabla K^{*}(v) \nabla_u g(v, H K^{*}(v), \hat{\alpha}) \xi$$

$$- \nabla g^{*}(v) \xi \qquad (8.24)$$

Substituting (8.24) into (8.22) and using

$$\hat{c} = v,$$
$$\hat{u} = HK^{*}(v)$$

in both (8.22) and (8.23), we obtain:

$$\nabla q^{*}(v) + \nabla g^{*}(v) \xi + \nabla f(v) \eta = 0 \qquad (8.25)$$

$$\eta_i f_i(v) = 0 \ , \ \eta_i \geq 0 \text{ and } f_i(v) \leq 0 \ , \ i = 1, \ldots, r \qquad (8.26)$$

According to the definition of Ω, the vectors ξ and v satisfy

$$\xi_i [g^{*}(v)]_i = 0 \ , \ \xi_i \geq 0 \text{ and } [g^{*}(v)] \leq 0 \ , \ i = 1, \ldots, L \qquad (8.27)$$

Equations (8.25) to (8.27) are the Kuhn-Tucker conditions of (ROP) at v.

Theorem 1 shows that every $v \in \Omega_1$ satisfies the necessary conditions for optimality. If these necessary conditions are also sufficient, for example, if (ROP) is a convex problem, then $\Omega_1 \subset \Omega_1^{*}$ is guaranteed. If, however, the problem is such that there is no equivalance between the necessary and sufficient conditions for optimality, it is assumed that the solutions of (ROP) belong to Ω_1.

The sufficient conditions that ensure $\Omega_1^* \subset \Omega_1$ are given in theorem 2. Theorem 2 and the proof are given in Appendix B. The convergence analysis of the algorithm is presented in Lin, Hendawy and Roberts (1988b).

8.5 Simulation study

8.5.1 Simulation example

The following three examples have been used in a simulation study on a Prime 750 computer. Local optimizations were solved using the NAG library routine E04VBF (NAG fortran library manual 1983) which uses a direct search and sequential augmented Lagrangian optimisation technique (Gill, Murray and Wright, 1981).

Example 1:

This example consists of three subsystems. The model and the reality equations are:

$$y_{11} = c_{11} - c_{12} + 2\,u_{11} + \alpha_{11}$$

$$y_{21} = c_{21} - c_{22} - 3\,u_{22} + \alpha_{21}$$

$$y_{22} = 2\,c_{22} - c_{23} - u_{21} + u_{22} + \alpha_{22}$$

$$y_{31} = c_{31} + 2.5\,c_{32} - 4\,u_{22} + \alpha_{31}$$

$$y_{11}^* = 1.3\,c_{11} - c_{12} + 2\,u_{11} + 0.15\,c_{11}\,u_{11}$$

$$y_{21}^* = c_{21} - c_{22} + 1.2\,u_{21} - 3\,u_{22} + 0.1c_{22}^2$$

$$y_{22}^* = 2\,c_{22} - 1.25\,c_{23} - u_{21} + u_{22} + 0.25\,c_{22}\,c_{23} + 0.1$$

$$y_{31}^{*} = \quad 0.8 \quad c_{31} + 2.5\ c_{32} - 4.2 \quad u_{31}$$

The subsystem performance indicies are:

$$Q_1(c_1, u_1, y_1) = (u_{11} - 1)^4 + 5(c_{11} - 1)^2 + 0.1(y_{11} - 1)^2$$

$$Q_2(c_2, u_2, y_2) = 4\ c_{21}^2 + u_{22}^2 + 2(c_{21} - 2)^2 + c_{22}^2 + 3\ c_{23}^2 + 0.1\ (y_2 - 2)^2$$

$$Q_3(c_3, u_3, y_3) = (u_{31} - 1)^2 + (c_{31} + 1)^2 + 2.5\ c_{32}^2 + 0.1\ (y_{31} - 1)^2$$

The subsystem constraints are:

$$\mathcal{CU}_1 = \left\{ (c_1 , u_1) \in \mathfrak{R}^3 ;\ c_{11}^2 + c_{12}^2 \leq 1 \ ,\ 0 \leq u_{11} \leq 0.5 \right\}$$

$$\mathcal{CU}_2 = \left\{ (c_2 , u_2) \in \mathfrak{R}^5 ;\ c_{21}^2 + c_{22}^2 + c_{23}^2 + 0.5\ u_{21}^2 + u_{22}^2 \leq 0.8 \right\}$$

$$\mathcal{CU}_3 = \left\{ (c_3 , u_3) \in \mathfrak{R}^3 ;\ c_{31} + u_{31} + 1 \geq 0 \ ,\ 0 \leq c_{32} \leq 1 \right\}$$

The output dependent constraint is:

$$0.5\ y_{11} + y_{21} - y_{22} \leq 3$$

with the coupling equations:

$$u_{11} = y_{21} \ , \qquad u_{21} = y_{11} \ , \qquad u_{22} = y_{31} \ \text{and} \quad u_{31} = y_{22}$$

and interconnection matrix H of the form:

$$H = \begin{bmatrix} 0 & 1 & 0 & 0 \\ 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{bmatrix}$$

Example 2:

The formulation of this example is the same as that of example 1 except that the subsystem constraints are in the following form:

$$\mathcal{CU}_1 = \left\{ (c_1 , u_1) \in \Re^3 ; \; c_{11}^2 + c_{12}^2 \leq 1 \; , \; 0 \leq u_{11} \leq 2.5 \right\}$$
$$\mathcal{CU}_2 = \left\{ (c_2 , u_2) \in \Re^5 ; \; c_{21}^2 + c_{22}^2 + c_{23}^2 \leq 0.5 \right\}$$

$$\mathcal{CU}_3 = \left\{ (c_3 , u_3) \in \Re^3 ; \; c_{31} + u_{31}^2 + 1 \geq 0 \; , \; 0 \leq c_{32} \leq 1 \right\}$$

Example 3

This example also consists of three subsystems. The model and reality equations, performance indicies, local constraints and the coupling equations are the same as in example 2. The output dependent constraint takes the form:

$$0.5\, y_{11} + y_{21} - y_{22} + y_{31} + 0.1\, y_{31}^2 \leq 3$$

The structure of the systems of examples 1, 2 and 3 is shown in figure (8.1) .

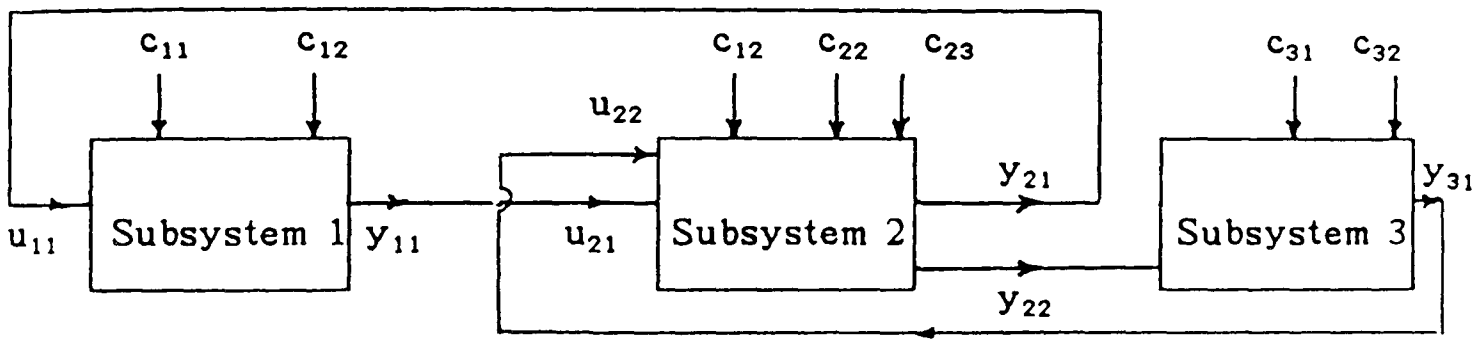


Figure (8. 1) The Structure of the System of Examples 1, 2 and 3.

8.5.2 Simulation results

The simulation results are given in Table (8.1). The initial conditions for v^0, ξ^0 and P^0 are all zeros. The convergence behaviour of the performance index and the controllers set points of the simulation examples are shown in figures (8.2), (8.3) and (8.4) respectively.

Table (8.1) and figure (8.5) show comparison results between the single loop technique and the new double loop technique.

The stopping criterion of the algorithm is:

$$\| \; P(v^k, \xi^k) \;\; R(v^k, \xi^k) \| \; < 2 \times 10^{-4}$$

together with

$$\| \hat{c}^k - v^k \| < 2 \times 10^{-4}$$

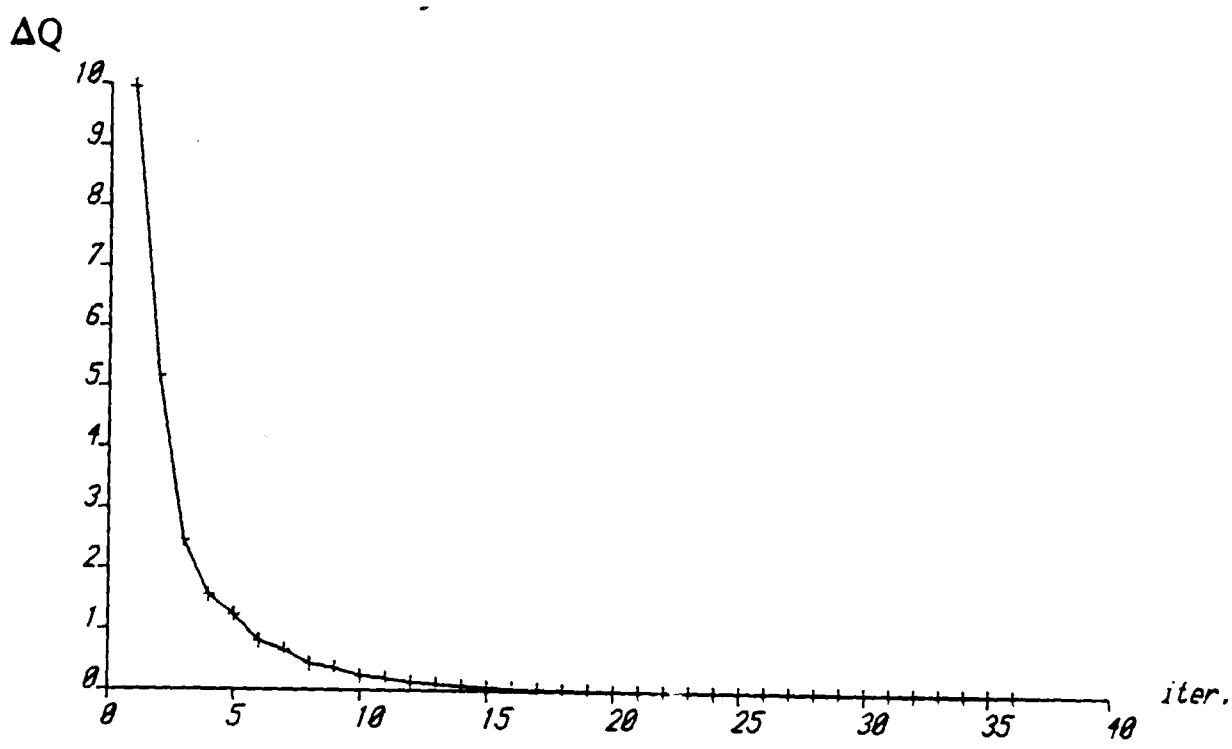
where $P(v^k, \xi^k)$ and $R(v^k, \xi^k)$ are defined by equations (8.27) and (8.28) respectively (see Lin, Hendawy and Roberts, 1988b).

$$[P(v, \xi) \left\{ g_*(v) + \; g^{*\prime}(v) \; (\hat{c}(v, \hat{\alpha}, \xi) - v) \right\}_i$$

$$= \begin{cases} 0 \quad , \quad \forall \; i \in N^1(v, \xi) \\ \\ [\; g^*(v) + \; g^{*\prime}(v) \; (\hat{c}(v, \hat{\alpha}, \xi) - v) \;]_i \quad , \; \forall i \notin N^1(v, \xi) \end{cases} \tag{8.27}$$

and

$$R(v,\xi) \cong \quad g^*(v) + g^{*'} \; (v) \; [\; (\hat{c}(v, \hat{\alpha}, \xi) - v)] \tag{8.28}$$



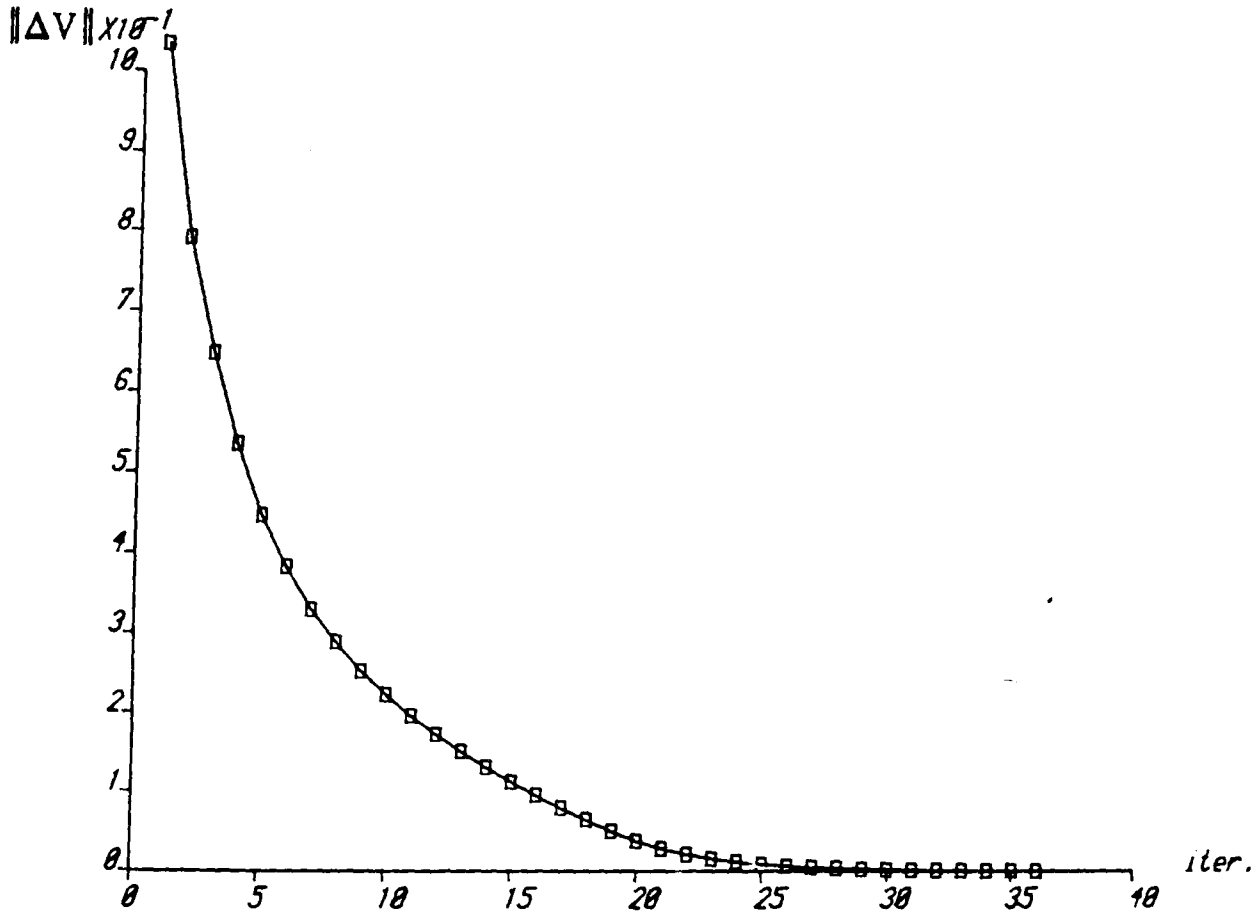
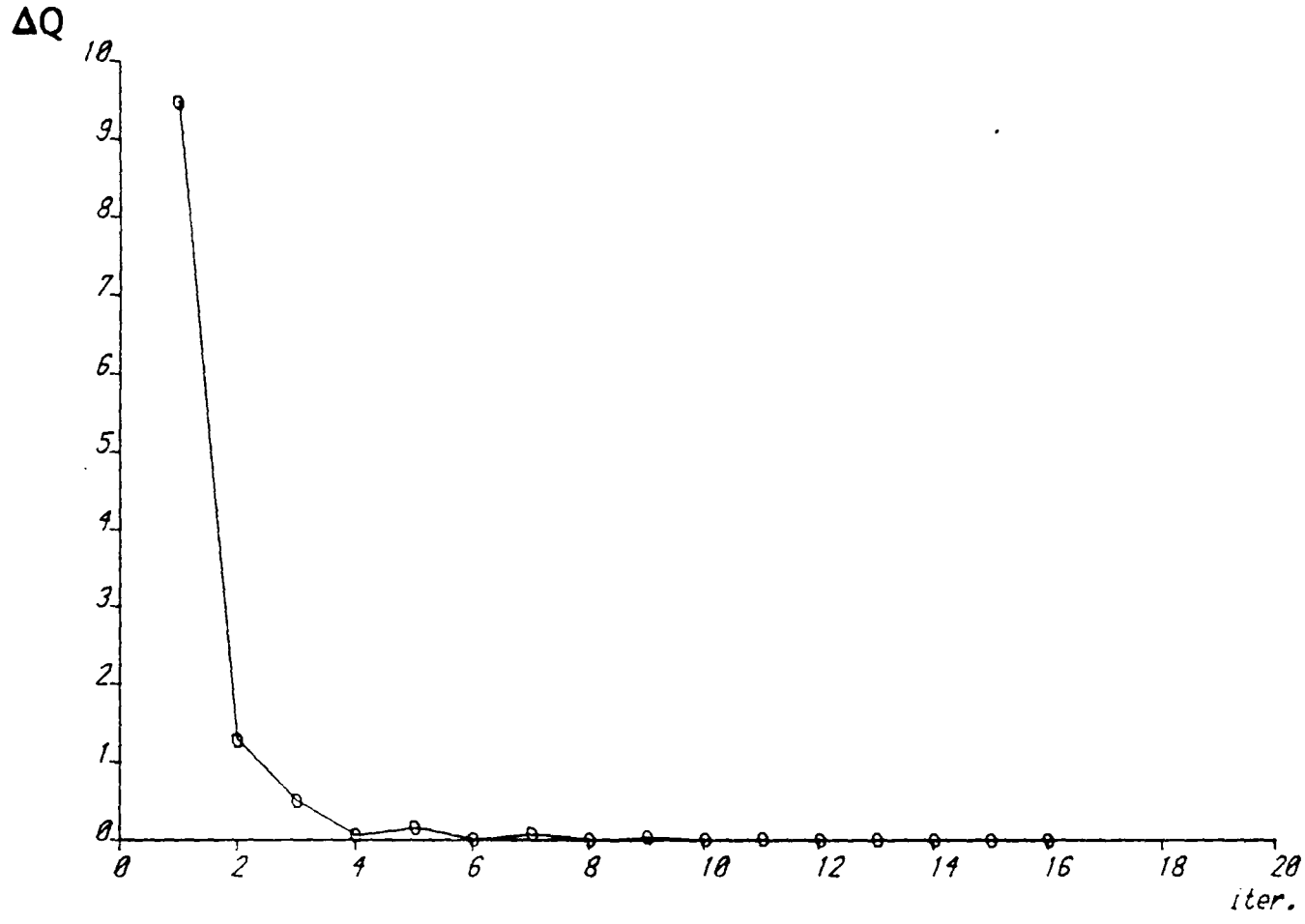


Figure (8.2) Convergence behaviour of the performance index and controller set points for Example 1.



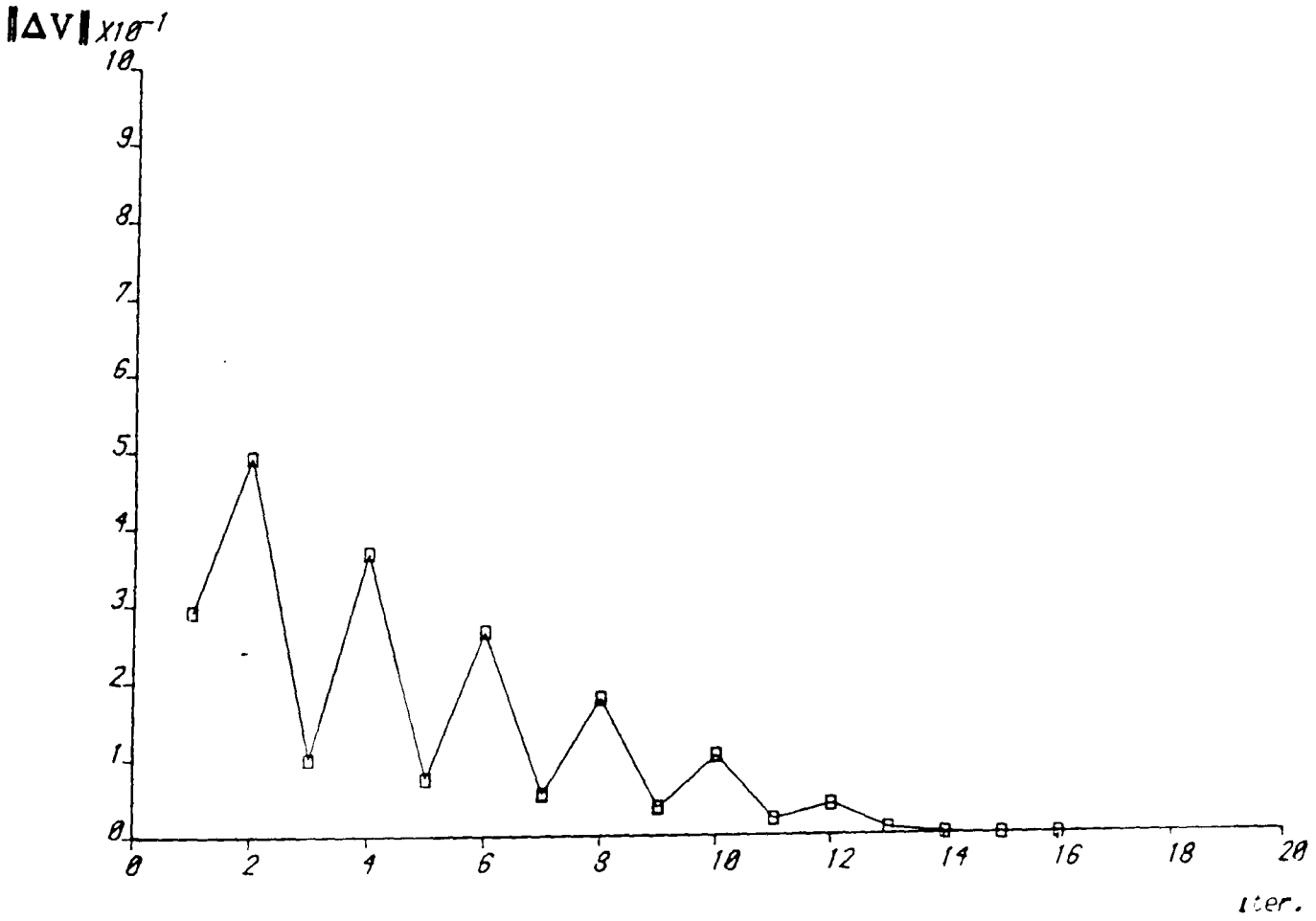
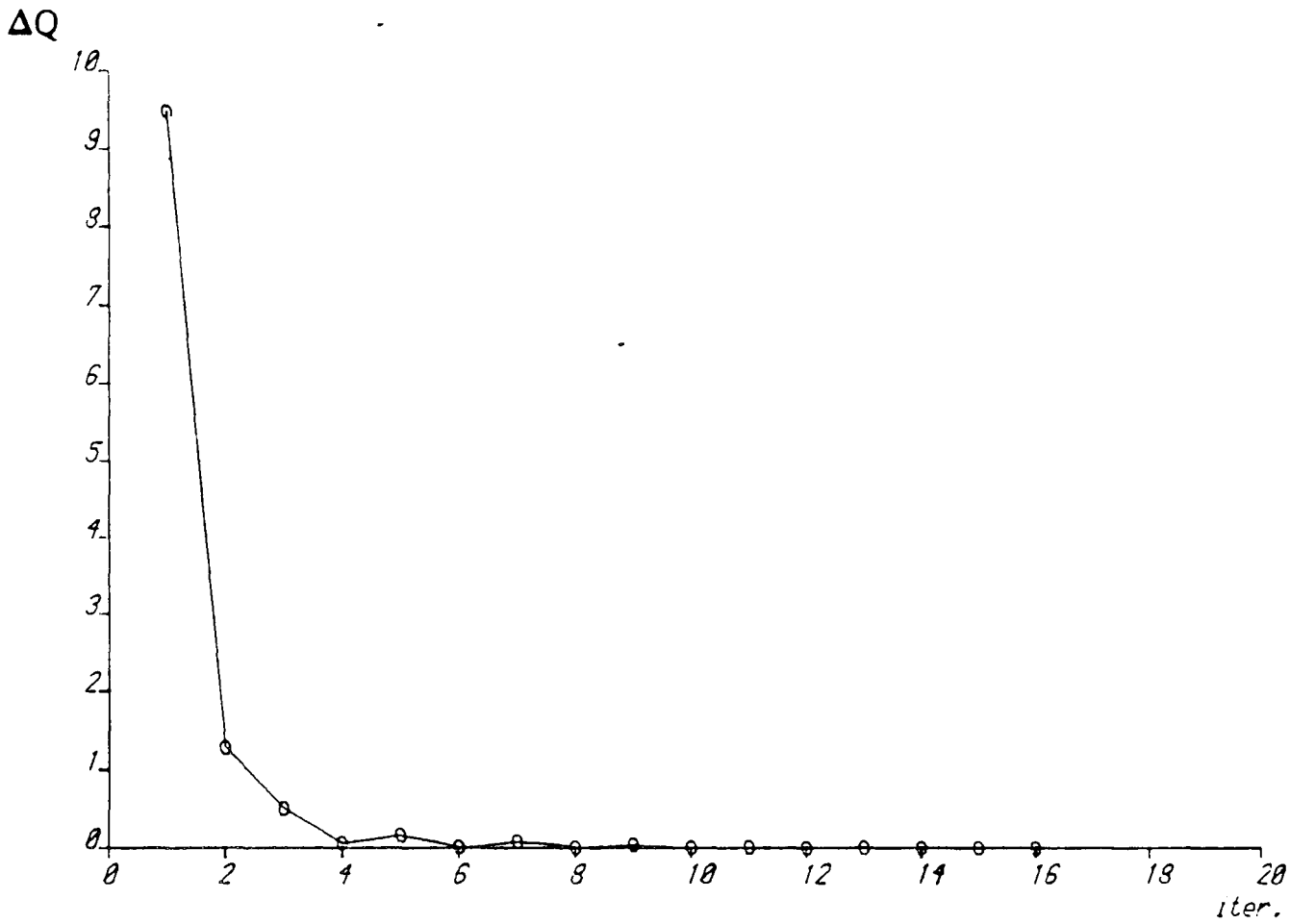


Figure (8.3) Convergence behaviour of the performance index and controller set points for Example 2.



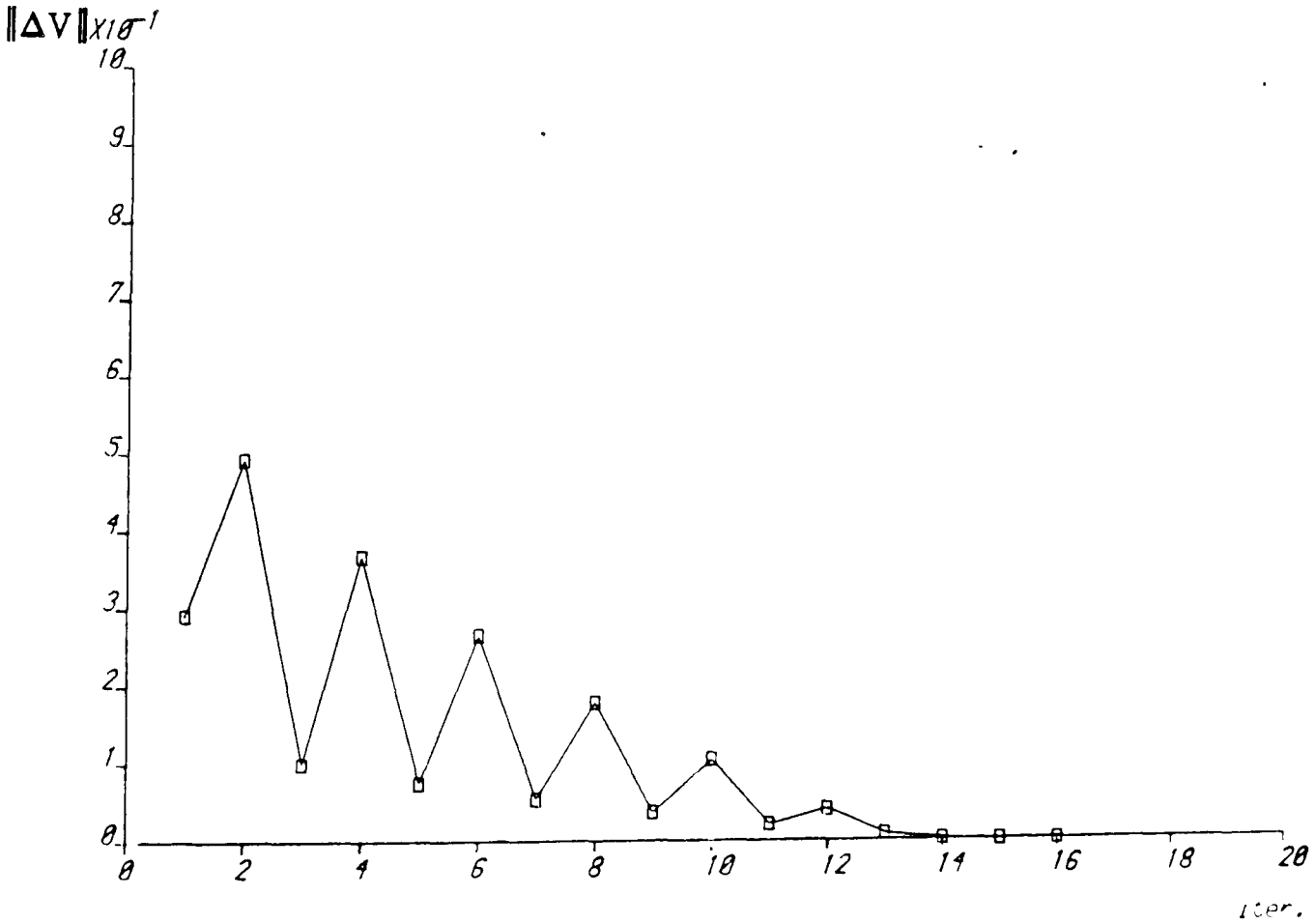


Figure (8.4) Convergence behaviour of the performance index and controller set points for Example 3.

The norm used in the criterion is:

$$\| x^{k+1} - x^k \| = \sqrt{\frac{1}{N} \sum_{i=1}^{N} (x_i^{k+1} - x_i^k)^2}$$

Comparing this algorithm with the single loop iterative algorithm (SIA) of Lin, Hendawy and Roberts, (1988a) shows that this new double loop (NDIA) is much more efficient.

The improvement of efficiency in examples 2 and 3 is much more than in example 1. The main reason is that there are more active c and u dependent constraints at the optimum in example 1 than in examples 2 and 3.

It is also noticed the achieved convergence of all examples. In examples 2 and 3, the deviation from the optimum control $\| \Delta v \|$ is not strictly decreasing. A possible explanation of the zigzags are:

a) The convergence conditions of the algorithm which are given by theorem 3 (see Lin, Hendawy and Roberts, 1988b) only provides sufficient local convergence conditions. Therefore, starting from any arbitrary initial condition the Zangwill function (Zangwill, 1969) need not necessarily be strictly decreasing.

b) Since the Zangwill function ($Z(v, \xi)$) in theorem 3 consists of two terms as:

$$Z(v, \xi) \triangleq \frac{1}{\epsilon_\xi} \| \xi - \bar{\xi} \|^2 + a_* \| v - \bar{v} \|^2$$

where:

$$\bar{v} = \lim_{k \to \infty} v^k, \quad \bar{\xi} = \lim_{k \to \infty} \xi^k \quad \text{and } a_* \text{ is a constant defined by theorem 3.}$$

Then $\| \Delta v \|$ need not necessarily be strictly decreasing even when the Zangwill function is. However, when the local optimisation of this algorithm is slightly modified (Lin, Roberts and Kambhampati, 1987) by adding a convexification slack variable term, there was more improvement of the efficiency and the zigzagging behaviour of $\| \Delta v \|$ has disappeared.

The modified MOP (Lin, Roberts and Kambhampati, 1987) is:

$$\min_{c,\, u,\, \omega} \left\{ AA + \rho \parallel \omega \parallel^2 \right\}$$

s.t.

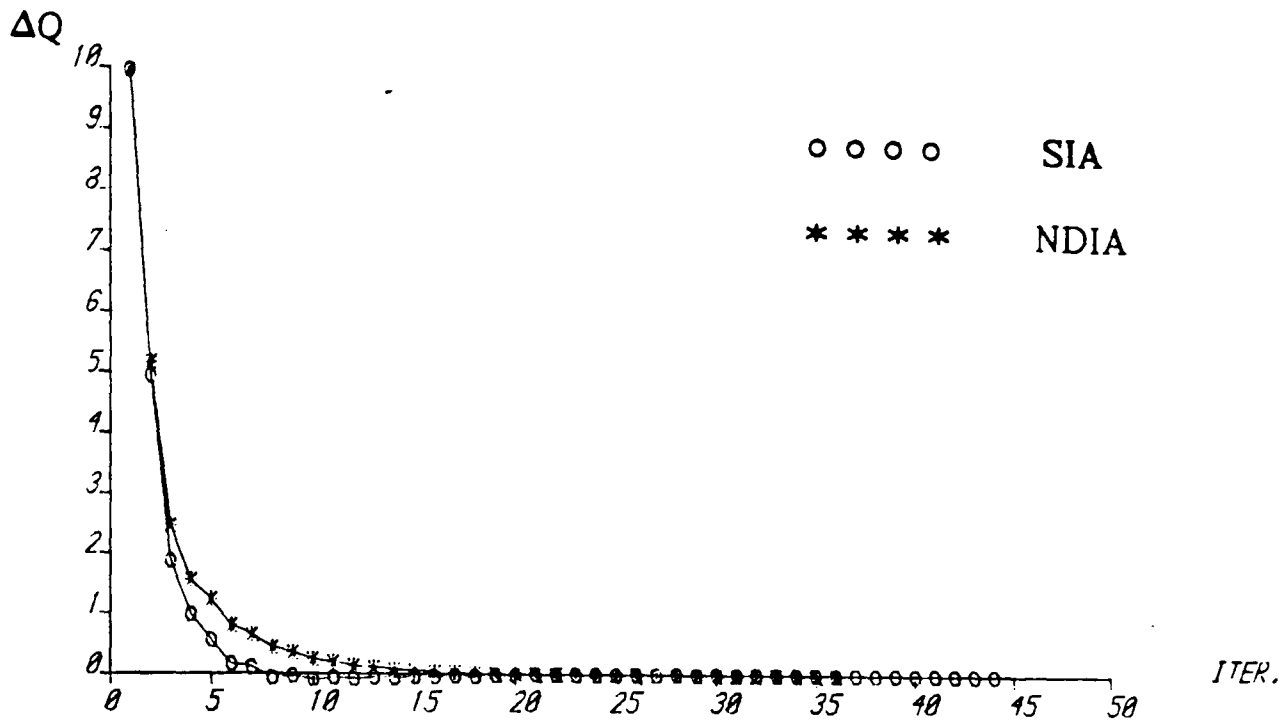
$$u = H K^{*}(v^k) + HK^{*\prime}(v^k)\,(c - v^k)$$

$$(c, u, \omega) \in \mathcal{CUW}$$

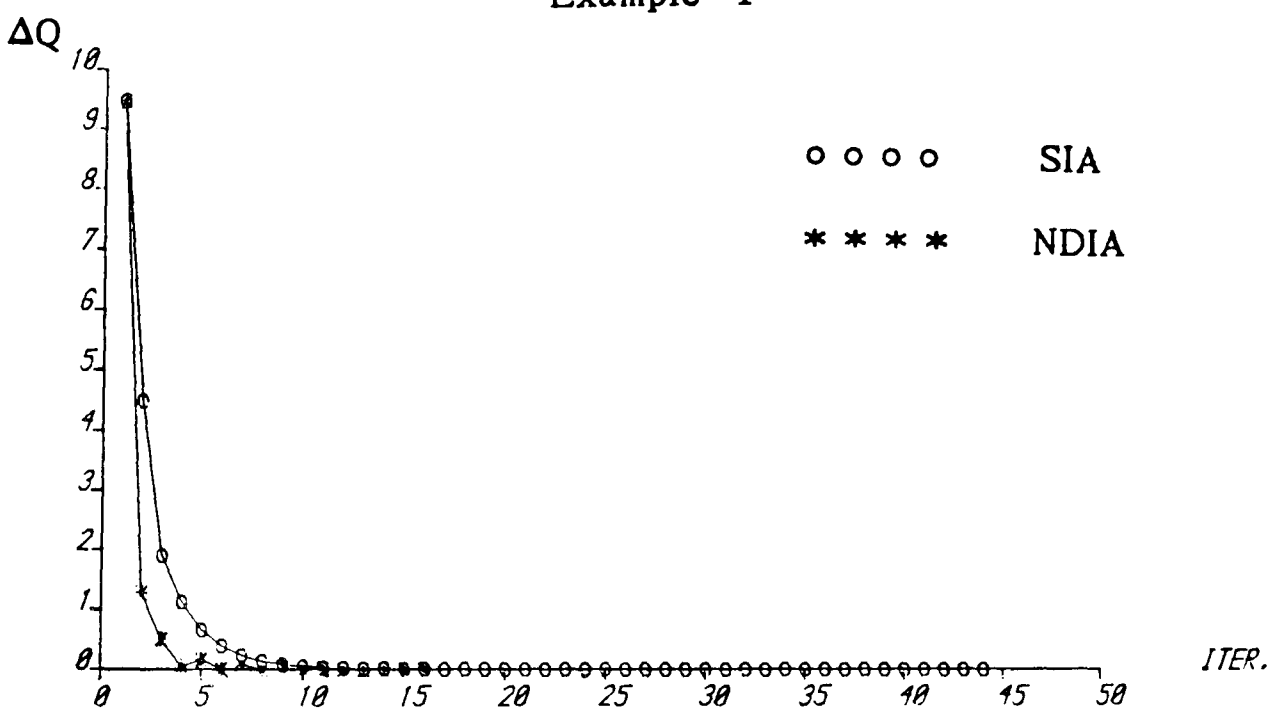
where AA is defined earlier by the MOP.

$$f(c_i, u_i) + \omega_i \leq 0$$

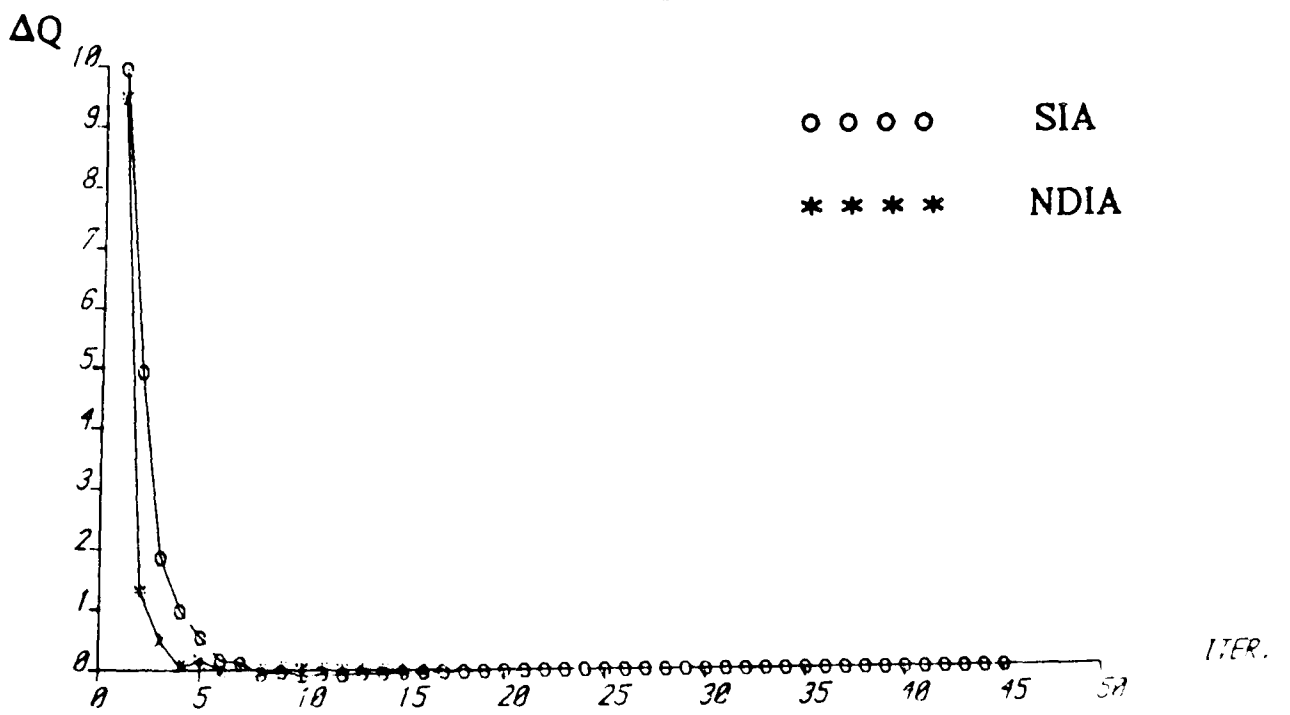
ω_i are slack variables and ρ is a penalty coefficient.



Example 1



Example 2



Example 3

Figure.(8.5) Comparison between convergence of single and double loop ISOPE techniques.

EX.	Algor.	ϵ_ξ	ϵ_v	ϵ_p	Iter.	Q	Q^*
1	SIA	0.5	0.5	0.88	44	6.4389	6.4389
	NDIA	0.36	0.25	0.4	36	6.4389	6.4389
2	SIA	0.5	0.5	0.88	44	6.9205	6.9206
	NDIA	0.5	0.8	0.4	16	6.9205	6.9206
3	SIA	0.5	0.5	0.88	44	6.9205	6.9206
	NDIA	0.5	0.8	0.4	16	6.9205	6.9206

Table (8. 1) Comparison of Single and Double Loop Algorithm.

8.6 Summary and conclusions

The presented algorithm is a new double loop iterative technique which can be viewed as a combination of the approximate linear model approach and the ISOPE method in the hierarchical case. At the expense of increasing off-line iterations this algorithm has significantly reduced the applied set point changes. New modifiers are introduced in the model optimisation to cover the case where inequality constraints are output dependent.

Inspecting this algorithm, we find that in the outer loop, where v^k, ξ^k are updated, the required real process measurements are also used to calculate the modifier. In the inner loop, where model-based optimisations are carried out, no real process information is needed to achieve the model interaction balance. Unlike the previous double-loop techniques (Brdys', Roberts, Badi and Kokkinos, 1986 ; Chen, Brdys' and Roberts, 1986 ; Brdys', Abdullah and Roberts, 1986), the derivatives of the model used in the model -based optimisation are updated each time the inner loop is entered using the derivatives of real process measurements. Although the derivatives of the real process are required to update the model, no extra set point changes are needed due to the fact that the derivatives are already available when entering the inner loop.

It is shown that the algorithm can achieve the real process optimum in spite of large differences between the model and reality.

CHAPTER 9

Conclusion and Suggestions for Further Work

9.1 Introduction

It is always desirable for an industrial processes operating in real time, to achieve and maintain optimal operating conditions. However, it is not an easy task because of the differences between the process models and reality.

In order to accomodate model deficiencies, measurements from the real system are incorporated in different ways according to the structure of the models. For example, with a mathematical (fixed) model, feedback information is used to improve the model based solution in the form of an iterative procedure (Findesien and co-workers, 1980). Alternatively, with adaptive models which contain uncertain parameters, process measurements may also be used to overcome such uncertainties by adopting a two step approach; optimisation and parameter estimation (for example, Youle and Duncanson, 1970;Roberts, 1978).

This research is concerned with some steady state hierarchical optimising control techniques. Two different kinds of algorithms have been investigated in this thesis. The first class of algorithms uses fixed models and global feedback information. Three different algorithms of this class are involved; these are : the Double Iterative Price Correction Mechanism

(Shao and Roberts, 1983), the Augmented Interaction Balance Method(Tatjewski, 1985) and the Augmented Price Correction Mechanism (Hendawy and Roberts, 1989).

The second class of algorithms employs adaptive models with uncertain parameters, namely: the Integrated System Optimisation and Parameter Estimation (ISOPE) techniques (Roberts, 1978;1979).

The algorithms of both classes are based on either the normal or the augmented Lagrangian technique.

9.2 Conclusion and Suggestions

A detailed simulation study of the Double Iterative Price Correction Mechanism (DIPCM) and the Augmented Interaction Balance Method with Feedback (AIBMF) has been presented. These results have shown that both algorithms, the DIPCM and the AIBMF, are applicable to a wide range and different types of problems. However, the DIPCM are not recommended for non-convex problems which have a duality gap since they are based on the normal Lagrangians, as shown in chapter 4.

The DIPCM and the AIBMF algorithms both converge to sub-optimal results but the sub-optimality of version 2 (V2) of the DIPCM in most cases is worse than the AIBMF and version 1 (V1) of the DIPCM.

The solution of V1 mostly converges within less set point changes than V2 and the AIBMF, but the total number of iterations of the AIBMF are always less than those of V1 and V2 of the DIPCM. This is due to the fast convergence of the inner loop of the AIBMF. Therefore, the computing processing time of this algorithm is always less than those of the DIPCM.

The sensitivity of the algorithms against their dependent parameters is investigated. Both algorithms have two parameters to be selected: (ϵ_1,ϵ_2) for the DIPCM and (ρ,α) for the AIBMF. A guideline for choosing these

parameters for a certain example is presented in chapter 4 of this thesis which may generally be used. However, trial and error must still be applied to assign the best values of the parameters, but V2 is more sensitive than V1 and AIBMF.

Finally, both algorithms can cope with system non-linearity with an exception of an observed deterioration of V2 which, for the particular example investigated exhibited an increase in the total number of iterations for a non-linearity factor $\beta > 0.65$.

In this research, an Augmented Price Correction Mechanism (APCM) algorithm has been proposed and formulated to extend the applicability of the previous Price Correction Mechanism (Shao and Roberts, 1983) to cover the case of non-convex problems. A comprehensive simulation study consisting of five numerical examples shows that the algorithm is applicable to a wide range and type of problems including convex and non-convex problems.

Comparing the algorithm with the previous DIPCM and AIBMF algorithms shows that the APCM can achieve convergence with a number of set point changes less than with the previous mentioned algorithms. Analysis of the sufficiency conditions for the model based solution of the APCM algorithm shows that the algorithm will remain suboptimal since the conditions of optimality can not be achieved. This is due to the model-reality differences and the fixed models they use.

In order to gain a good understanding of any algorithm, a theoretical study and convergence analysis is required. Further research is needed in this line, where most of the existing hierarchical suboptimal optimisation techniques require more theoretical study to analyse the mathematical properties of the algorithms.

Further work is also required to extend the application of the most efficient algorithms for example, (APCM, DIPCM(VI), AIBMF) to on-line implementation using, for example, a laboratory pilot plant. The original version of the price method has been successfully applied to an on-line

control using a vaporiser and tank system (Bakalis, 1986). Hence, applying these algorithms which are modified versions of the price method might improve the efficiency of its control system. An important advantage has been reported (Bakalis, 1986) for these kind of algorithms that they require only a fraction (≈ 1/4) of the on-line time required by the standard ISOPE algorithms.

In this thesis, two optimal hierarchical integrated system optimisation and parameter estimation (ISOPE) algorithms are proposed and investigated.

They are extensions of the modified two step approach(Roberts, 1978) to extend the applicability of the method to cover the case of problems with output dependent constraints and to reduce the required number of set point changes from the real process.

One of the proposed ISOPE algorithms uses an output feedback structure and a single loop iteration strategy. This algorithm has some advantages, compared with the previous algorithm (Chen et. al. 1986), that the existence of the model-based optimal solution is ensured during the course of iteration and the model optimisation problem is simplified considerably.

Generally, the single loop iterative algorithms are not as efficient as the double loop iterative algorithms. Therefore, in this research another hierarchical optimal ISOPE technique is proposed with a double iterative coordination strategy where the measurements from the real process is only required for the outer loop to update the model. The algorithm has several advantages over the previos double loop ISOPE techniques (Brdys' and Roberts, 1985; Brdys' et.al. 1986a; Chen et. al., 1986), in that problems with output dependent constraints are included and a full use of the information measurements is employed. In this algorithm, the derivatives from the real process measurements are used to update the model. Therefore, it is possible to linearize the model by using these derivatives. Simulation studies show that the algorithm is efficient.

Enormous numbers of ISOPE and other algorithms have been and are being developed in the Control Engineering Centre of City University for solving

the system optimisation problem. However, most of these algorithms are based on theoretical analysis and abstract simulation examples except for a very few which have been tested in the laboratory (Bakalis, 1986; Ellis and Roberts, 1985; Chen, 1986).

It is suggested that the most efficient algorithms (for example, those with double iterative coordination strategy, augmented Lagrangian algorithms and the algorithms using approximate linear model and sequence model approaches (Liù and Roberts, 1988)), should be tested on a real exisiting system.

In this research, the main interest was restricted to a study of steady state optimisation techniques. Further work is suggested to consider how to extend the algorithms developed in this thesis and other efficient ISOPE approches to be applied to dynamic systems. The modified two-step approach has been successfully extended to the on-line control of a pilot scale travelling load furnace, which is a batch dynamic plant (Stevenson, 1985). Current research in the Control Engineering Centre is in progress for optimisation of steady state systems with non-linear constraints and slow dynamics.

One of the major drawbacks of the modified two step approach and its extended algorithms is the requirement for the measurements of real output derivatives with respect to the real set points, in order to calculate the modifier vector. Under noisy measurements the performance of these methods are expected to deteriorate considerably. This imposes practical limitations to the ISOPE algorithms because the differentiation amplifies measurement errors. However, employing simple filter techniques to reduce the modifier vector can significantly reduce influence of noise and improve the performance of the methods.The effect of measurement noise on ISOPE methods and the use of various types of filters to improve the method is investigated by Filali and Ellis,(1988).

On the other hand, efforts have been made to either totally eliminate this necessity (Chen, 1986; Liu and Roberts, 1988) or to make the algorithm converge faster (Lin, Hendawy and Roberts, 1988b; Abdullah, 1988; Lin et. al.

1988c, d). Families of double iterative coordination algorithms have been developed to speed up the convergence and have proved to be successful. However, furthur research can be conducted to eliminate the need of derivatives totally where the efficiency of the algorithm developed by Liu et. al.(1988) reduces as the number of set points of the problem increases.

The final concern is the application of the most suitable hierarchical control algorithms to industry. This may be achieved by developing these techniques in the form of software packages.

REFERENCES

1. Abdullah, N (1988) : " Augmented Integrated System Optimisation and Parameter Estimation for On-line Hierarchical Control of Large Scale Industrial Processes " - Ph.D. Thesis, City University, U.K.

2. Allidina, A. Y.; Buro, R. and Malinowski, K. (1985) : " Simulation of PDE System Using Hierarchical Techinques ", IEE Int. Conf. " Control 85 " Publication no. 252, University of Cambridege, England, pp 500 - 504.

3. Bakalis, P. (1986) : " On-line Hierarchical Control " - Ph.D. Thesis, Control Eng. Centre, City University, London, U.K.

4. Bazaraa, M.S. and Shetty, C.M. (1979) : " Nonlinear programming, Theory and Algorithms ", John Wiley and Sons.

5. Bertsekas, D.P. (1982) : " Constrained Minimisation and Lagrange Multiplier Methods ", Academic Press, New York.

6. Binmore, K.G. (1983) : " Calculus ", Cambridege University Press.

7. Brdys', M. (1983) : " Hierarchical Optimising Control of Steady State Large Scale Systems under model-reality difference of mixed type-a mutually interacting approach ", Proc. 3rd IFAC Symp. On Large Scale Systems, Theory and Applications, Warsaw, pp 49-57.

8. Brdys', M.; Abdullah, N. and Roberts, P.D. (1986) : " Hierarchical Adaptive Techniques of Optimising Control of Large Scale Steady State Systems : Optimality, Iterative Strategies and Their Convergence, Research Memorandum, CEC/ MB-NA-PDR/13, City University, Control Engineering

Centre, London, U.K.

9. Brdys', M.; Chen, S. and Roberts, P.D. (1986) : " An Extension of the Modified Two-step Algorithm for Steady-state System Optimisation and Parameter Estimation ", Int. J. Systems Sci., vol. 17, No. 8, pp 1229-1243.

10. Brdys', M. and Roberts, P.D. (1986) : " Optimal Structures for Steady-state Adaptive Optimising Control of Large Scale Industrial Processes ", Int. J. of Systems Sci., vol. 17, No. 10, pp 1449-1474.

11. Brdys', M. and Roberts, P.D. (1987) : " Convergence and Optimising of Modified Two-step Algorithm for Integrated System Optimisation and Parameter Estimation ", Int. J. Systems Sci., vol. 18, No. 7, pp 1305-1322.

12. Brdys', M. and Roberts, P.D. (1985) : " Hierarchical Single and Double Iterative Techniques for Optimising Control of Large Scale Industrial Process, Derivation, Applicability and Convergence ", IEE Int. Conf. " Control 85 " Publication No. 252, University of Cambridge, U.K., pp 89 - 93.

13. Brdys', M.; Roberts, P.D.; Badi, M.M. and Kokkinos, I.C. (1986) : " Model Based Double Iterative Strategy for Integrated System Optimisation and Parameter Estimation of Large Scale Industrial Processes ", Proc. 4th IFAC/ IFROS Symp. on Large Scale Systems : Theory and Applications, 26-29th August, Zurich, Switzerland, pp 634-639 (Pergamon).

14. Chen, S. (1986) : " Integrated System Optimisation and Parameter Estimation for On-line Control of Industrial Processes ", Ph.D. Thesis, City University, London, U.K.

15. Chen, S.; Brdys', M. and Roberts, P.D. (1986) : " An Integrated System Optimisation and Parameter Estimation Technique for Hierarchical Control of Steady State Systems ", Int. J. Systems Sci., vol. 17, No. 8, pp 1209-1228.

16. Durbeck, R.C. (1965) : " Principles of Simplification of Optimising Control Models ", Ph.D. Thesis, Case Western Reserve University.

17. Ellis, J.E.; Hua, S.G.; Roberts, P.D. and Michalska, H. (1985) : Adaptive Models and Optimum Operating Conditions ", Research Memorandum CEC/JEE-SGH-PDR-HM/27, Control Engineering Centre, City University, London, U.K.

18. Ellis, J.E. and Roberts P.D. (1982) : " Measurement and Modelling Trade-Offs for Intergrated System Optimisation and Parameter Estimation ", Large Scale Systems, 3, pp 191-204.

19. Ellis, J.E.; Sheng, G.H.; Roberts, P.D. and Michalska, H. (1985) : " Adaptive Models and Optimum Operating Conditions ", Research Memorandum, CEC/JEE-SGH-PDR-HM/27, City University, London, U.K.

20. Ellis,J.E.; Roberts, P.D. and Michalska, H. (1986) : " Adaptive Models in Decentralised Optimising Control ", 4th IFAC/ IFROS Symp. on Large Scale Systems-Theory and Applications, Zurich, pp 334-339.

21. Ellis, J.E.; Michalska, H. and Roberts, P.D. (1984) : " Coordinated Decentralised in Integrated System Optimisation and Parameter Estimation of Large Scale Processes ", Proc. of the 9th Trieminal World Congress of IFAC, Budapest, Hungary, 2-6 July, pp 261-266 (Pergamon Press).

22. Ellis, J.E. and Roberts, P.D. (1981) : " Simple Models for Integrated System Optimisation and Parameter Estimation ", Int. J. of Systems Sci., vol. 12, pp 455-472.

23. Ellis, J.E. and Roberts, P.D. (1985) : " On the practical viability of Integrated System Optimisation and Parameter Estimation ", IEE Conf. publ. No. 252, vol .1, pp 281-285.

24. Filali, S. and Ellis, J.E. (1988) : " Methods For Real Process Optimisation and The Effects of Measurement Noise ", Research Memorandum CEC/SF-JEE/80, Control Enginering Centre, City University, London, U.K.

25. Findeisen, W.; Bailey, F.N.; Brdys', M.; Malinowski, K.; Tatjewski, P. and Wozniak, A. (1980) : " Control and Coordination in Hierarchical Systems ",

John Wiley and Sons.

26. Findeisen, W.; Brdys', M.; Malinowski, K.; Tatjewski, P. and Woznial, A. (1978) : " On-line Hierarchical of Steady-state Systems ", IEEE Trans. on Automatic Control, vol. AC-23, No. 2, pp 189 - 209.

27. Fletcher, R. (1987) : " Practical Methods of Optimisation ", Second Edition, John Wiley and Sons.

28. Foord, A.G. (1974) : " On-Line Optimisation of a Petrochemical Complex", Ph.D. Thesis, University of Cambridge, U.K.

29. Gill, P.E. and Murray, W. (1974) : " Numerical Methods for Constrained Minimisation ", Academic Press.

30. Gill, P.E.; Murray, W. and Wright, M.H. (1981): "Practical Optimisation", Academic Press.

31. Hendawy, Z.M. and Roberts, P.D. (1989) : " A Price Correction Mechanism using an Augmented Interaction Balance Method With Feedback of Large Scale Industrial Process : Derivation and Applicability ", Journal of AMSE, vol. 16, No. 1, pp 53-63.

32. Hestenes, M.R. (1969) : " Multiplier and Gradient Methods ", Journal of Optimisation Theory and Application, vol. 4, No. 5, pp 303-320.

33. Kambhampati, C. and Ellis, J.E. (1987) : " Approaches to the Centralised On-line Optimisation Problem ", Research Memorandum, CEC/CK-JEE/55, Control Engineering Centre, City University, London, U.K.

34. Kambhampati, C. (1988) : " Algorithms for Optimising Control ", Ph.D. Thesis, Control Engineering Centre, City University, London.

35. Kuhn, H.W. and Tucker, A.W. (1951) : " Nonlinear Programming ", In Proceedings of the second Berkeley Symposium on Mathematical Statistics and Probabilities (J. Neyman, ed.), University of California Press, pp 481-492.

36. Lefkowitz, I. (1966) : " Multilevel Approach Applied to Control System Design ", Trans. ASME. Series D.J. of Basic Eng., vol. 88, No. 2, pp 392-398.

37. Lefkowitz, I. (1977) : " Integrated Control of Industrial Systems ", Royal Society Philosophical Trans., Series A, vol. 287, pp 443-465.

38. Luenberger, D. (1984) : " Linear and Nonlinear Programming ", second edition, Addison-Wesley Publishing Company.

39. Lin, J., Chen, S. and Roberts, P.D. (1986) : " A Modified Algorithm for Steady-State Integrated System Optimisation and Parameter Estimation ", Research Memorandum, CEC/JL-SC-PDR/47, Control Engineering Centre, City University, London.

40. Lin, J., Hendawy, Z.M. and Roberts, P.D. (1988a) : " Extension of Integrated System Optimisation and Parameter Estimation to Hierarchical Control of Steady-State Systems with Output Dependent Constraints", Int. J. Control, vol. 47, No. 2, pp 413-431.

41. Lin, J.; Hendawy, Z.M. and Roberts, P.D. (1988b) : " New Model-Based Double Loop Iterative Strategy For Integrated System Optimisation and Parameter Estimation of Large-scale Industrial Processes ", Int. J. Control, vol. 47, No. 3, pp 753-773.

42. Lin, J.; Hendawy, Z.M. and Roberts, P.D. (1988c) : " A Modified Integrated System Optimisation and Parameter Estimation for Hierarchical Control of Large Scale Systems ", Journal of AMSE Press, vol. 12, No. 2, pp 43-52.

43. Lin, J.; Li, L.Z.; Wan, B.W. and Roberts, P.D. (1988d) : " Augmentation of Several Newly Developed Algorithms for Integrated System Optimation and Parameter Estimation of Large Scale Industrial Processes : Convergence Analysis and Comparative Study ", Research Memorandum, CEC/JL-LZL-BWW-PDR/98, Control Engineering Centre, City University, London, U.K.

44. Liu, Z.Y. and Roberts, P.D. (1988) : " A Sequential Model Approximation Approach for Optimising Control of Steady-state Systems ", Research Memorandum CEC/ZYL-PDR/86, Control Engineering Centre, City University, London, U.K.

45. Lowe, E.I. and Hidden, A. (1971) : " Computer Control in Process Industries ", Peter Preginus.

46. Maxwell, M. (1982) : " Distributed versus Centralised Control ", IFAC Distributed Computer Control Systems, Tallinn, U.S.S.R., pp 189.

47. Maxwell, M. (1983) : " Distributed versus Centralised Control for Profit ", IFAC Distributed Computer Control Systems, Sabi-sabi, South Africa, pp 39-44.

48. Measarovic, M.D.; Macko, D. and Takahara, Y. (1970) : " Theory of Hierarchical Multilevel Systems ", Academic Press, New York.

49. Michalska, H., Ellis, J.E. and Roberts, P.D. (1985) : " Joint Coordination Method for the Steady-state Control of Large Scale Systems ", Int. J. Systems Sci., vol. 16, No. 5, pp 605-618.

50. Murray, W. (1976) : " Methods for Constrained Optimisation ", chapter 12 of Dixon, L.C.W. (Editor), Optimisation in Action, Academic Press.

51. NAG FORTRAN LIBRARY MANUAL Mark 10 , 1983.

52. Powell, M. G. D. (1969) : " A Method for Nonconstraints in Minimisation Problems ", In Optimisation, ed. R. Fletcher, Academic Press, London.

53. Reklaitis, G.V.; Ravindran, A. and Ragsdell, K.M. (1983) : " Engineering Optimisation ", John Wiley and Sons .

54. Roberts P.D. and Lalui, A. (1982) : " Integrated System Optimisation and Parameter Estimation of A Chemical Plant ", IEEE Conference on Applications of Adaptive and Multivariable Control, Hull, 19-21 July, pp 60-64.

55. Roberts, P.D. (1988a) : " Current State of the Art of Process Control ", Proceedings of the first regional seminar and process control 1988 , Faculty of Engineering, Universiti Kebangsaan, Malaysia, November 28-29, pp 1-9.

56. Roberts P.D. (1988b) : " The Potential for Hierarchical Control Algorithms in Process Control ", Proceedings of the first regional seminar and process control, November 28-29, Malaysia, pp 273-279.

57. Roberts, P.D. (1977) : " Integrated System Optimisation and Parameter Estimation of A Chemical Reactor Using A Multivariable Approach ", Digital Computer Applications to Process Control, edited by H.R. Van Nauta Lemke (Amsterdam : North-Holland), pp 551.

58. Roberts, P.D. (1978) : " Algorithms for Integrated System Optimisation and Parameter Estimation ", Electronics Letters, vol. 14, No. 6, March 16, pp 196-197.

59. Roberts, P.D. (1979) : " An Algorithm for Steady-state System Optimisation and Parameter Estimation ", Int. J. System Sci., vol 10, No. 7, pp 719-734.

60. Roberts, P.D. and Roussias, L. (1980) : " Overcoming Model Uncertainties in Optimising Supervising Control ", Proc. 6th IFAC/ IFIP Conf. on Digital Computer Apllication to Process Control, 14-17 October, Dusseldorf, F.R.G., pp 383-388 (Pergamon).

61. Roberts, P.D. and William, T.W.C. (1981) : " On An Algorithm for Combined System Optimisation and Parameter Estimation ", Automatica, vol. 17, pp 199-209.

62. Rockafellar, R.T. (1974) : " Augmented Lagrange Multiplier Functions and Duality in Nonconvex Programming ", SIAM J. Control, vol. 12, No. 2, pp 268-285.

63. Savas, E.S. (1965) : " Computer Control of Industrial Process ", Mc Graw-Hill Book Company.

64. Shao, F.Q. and Roberts, P.D. (1983) : " A Price Correction Mechanism with Global Feedback for Hierarchical Control of Steady-state Systems ", Large Scale Systems 4, pp 67-80.

65. Sheng, G.H. and Ellis, J.E. (1985) : " As Assessment of Adaptive Model Strategies for Optimum Operating Conditions ", Research Memorandum CEC/GS-JEE/21, Control Engineering Centre, City University, London, U.K., pp 505 - 510.

66. Singh, M.G. and Titli, A. (1978) : " Systems : Decomposition, Optimisation and Control ", Pergamon Press.

67. Singh, M.G.; Allidina, A.Y.; Cook, R. and Malinowski, K. (1985) : " Parallel Processing Techniques for Large Scale Simulation Problems ", IEE Int. Conf. " Control 85 ", Publication No. 252, University of Cambridge, England.

68. Stevenson, I.A. (1985) : " On-line Adaptive Computer Control of A Travelling Load Furnace ", Ph.D. Thesis, Control Engineering Centre, City University, London, U.K.

69. Stphanopoulos, G. and Westerberg, A. W. (1975) : " The use of Hestenes Method of Multipliers to resolve Dual Gaps in Engineering System Optimisation ", JOTA , No. 15, pp 285 - 309.

70. Stoilov, E. (1980) : " Interaction Balance Method with Feedback Using Augmented Lagrangians, Arch. Autom. i Telemech. 25, pp 1-20.

71 Tatjewski, P. (1985) : " On-line Hierarchical Control of Steady-state Systems Using the Augmented Interaction Balance Method With Feedback ", Large Scale Systems, Theory and Application, vol. 8, No. 1, pp 1-18.

72. Youle, P. V. and Duncanson, L. A. (1970) : " On-Line Control of Olefine Plant ", Chemical and Process Engineering , vol. 51, pp 49 - 52.

73. Zangwill, W. I. (1969) : " Nonlinear Programming : A Unified Approach ", Prentice-Hall Inc., Englwood Cliffs, U. S. A.

APPENDIX A

The Conditions for Updating ξ

$$\xi^{k+1} = \xi^k + \epsilon_\xi^k \, p\,(\, v^k, \xi^k)\, [g^*\ (v^k) + g^{*\prime}(v^k)\ (\hat{c}^k - \ v^k\ )\ ] \qquad (\text{A. 1})$$

$p(v,\xi\ )$ is a projection operator satisfying :

$$[\ p\ (\ v, \xi\ )\ \{\ g^*(v) + g^{*'}(v)\ (\hat{c}(v, \hat{\alpha}, \xi)\ -\ v)\ \}\ ]_i$$

$$= \begin{cases} 0, & \forall_i \in N^1(v, \xi\ ) \\ [g^*(v) + g^{*\prime}(v)\ \ (\ \hat{c}(v, \hat{\alpha}, \xi) - v)]_i\ , & \forall_i \notin N^1(v,,\xi) \end{cases} \qquad (\text{A.2})$$

$$N^1(v, \xi) = \{\ i : \xi_i = 0 \ \text{ and } \ [g^*(v)+g^{*\prime}(v)\ (\hat{c}(v, \hat{\alpha}, \xi)-v)]_i \leq 0\ \} \qquad (\text{A. 3})$$

where $[\ \ .\ \ ]_i$ denotes the ith element of the vector. The gain parameter ϵ_ξ^k is determined as follows :

Let

$$N^2\ (v, \xi) \triangleq \{\ i : \xi_i > 0 \ \text{ and } \ [g^*(v)+g^{*\prime}(v)\ (\hat{c}\ (v, \hat{\alpha}, \xi)-v]_i < 0\ \} \qquad (\text{A. 4})$$

and

$$\bar{\epsilon}_\xi^k \triangleq \min_{i \in N^2(v^k, \xi^k)} \left\{ - \xi_i^k \,/\, [g^*(v^k)+g^{*\prime}(v^k)\ (\hat{c}^k - v^k)]_i \right\} \qquad (\text{A. 5})$$

then

$$\epsilon_\xi^k = \begin{cases} \min \{ \epsilon_\xi , \overline{\epsilon_\xi^k} \} & \text{if } N^2(v^k, \xi^k) \text{ is not empty.} \quad (A.6) \\ \epsilon_\xi & \text{if } N^2(v^k, \xi^k) \text{ is empty.} \quad (A.7) \end{cases}$$

The equations (A. 4) - (A. 7) are prescribed to exclude the possibility of ξ_i^{k+1} becoming negative.

Appendix B

Theorem 2 and Proof

Let Assumptions 1 and the assumptions for (ROP) and (EOP 1) be satisfied. Assume :

a) every $v \in \mathcal{C}$ is a regular point of the constraints $g^*(v) \leq 0$ and $f(v) \leq 0$,

b) for every $\hat{\alpha} \in \phi$ and $v^* \in \Omega^*$, $q(\cdot, u(\cdot, v^*), \hat{\alpha})$ is convex on $\mathcal{C}$, where

$$\phi \triangleq \{ \alpha : v \in \mathcal{C} \quad \text{and} \quad F(v, HK^*(v), \alpha) = K^*(v) \}$$

$$u(c, v^*) \triangleq HK^*(v^*) + HK^{*'}(v^*)(c - v)$$

Then $\Omega_1^* \subset \Omega_1$.

Proof

For $v^* \in \Omega_1^*$, since v^* is a regular point of the constraints $g^*(v) \leq 0$ and $f(v) \leq 0$,

there exist ξ^* and η^* such that

$$\nabla q^*(v^*) + \nabla g^*(v^*)\, \xi^* + \nabla f(v^*)\, \eta^* = 0 \qquad (B.\ 1)$$

$$\eta_i^*\ f_i(v^*) = 0 \ , \quad \eta_i^* \geq 0 \ , \quad f_i \leq 0, \qquad i = 1, \ldots, r \qquad (B.\ 2)$$

$$\xi_i^* [\, g^*(v^*)\,]_i = 0 \quad , \quad \xi_i^* \geq 0 \quad , \quad [g^*(v^*)]_i \leq 0;$$

$$i = 1, \ldots, t \qquad (B.3)$$

On the other hand

$$\nabla_c q(v^*, HK^*(v^*), \hat{\alpha}^*) + \nabla K^*(v^*) H^t \nabla_u q(v^*, HK^*(v^*), \hat{\alpha}^*)$$

$$- \lambda(v^*, \hat{\alpha}^*, \xi^*) + \nabla_c g(v^*, HK^*(v^*), \hat{\alpha}^*) \xi^*$$

$$+ \nabla K^*(v) H^t \nabla_u g(v^*, HK^*(v^*), \hat{\alpha}^*) + \nabla f(v^*)\eta^*$$

$$= \nabla q^*(v^*) + \nabla g^*(v^*)\xi^* + \nabla f(v^*)\eta^* = 0 \qquad (B.4)$$

where $\hat{\alpha}^* = \hat{\alpha}(v^*)$. Because (MOP) is convex problem, equations (B.2) and (B.4) imply that its optimal solution for a given $\hat{\alpha}^*$ and $\lambda(v^*, \hat{\alpha}^*, \xi^*)$ is

$$\hat{c}(v^*, \hat{\alpha}^*, \xi^*) = v^*$$

Equations (B.5) and (B.3) guarantee $(v^*, \xi^*) \in \Omega$, that is,

$v^* \in \Omega_1$. □