ALGORITHM DEVELOPMENT AND ANALYSIS FOR ON-LINE OPTIMISING CONTROL OF LARGE SCALE INDUSTRIAL PROCESSES

By

ZAHRA AMINI-LARGANI

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

CONTROL ENGINEERING CENTRE
DEPARTMENT OF ELECTRICAL, ELECTRONIC
AND INFORMATION ENGINEERING

CITY UNIVERSITY, LONDON

OCTOBER 1990

To my dear Mother,

may she rest in peace

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 advice, guidance, encouragement, understanding and caring attitude throughout my research work. Without him, this project would not have been completed.

I am also indebted to Dr. J. Lin of the Control Engineering Centre of City University for his invaluable advice, contribution, guidance and most of all his patience. Most of the work in this thesis has been suggested by him.

Thanks are also due to Dr. J. Ellis of the Control Engineering Centre of City University for his help and advice in this research.

I would also like to thank the following friends for their help, support and encouragement : Chaffei, Jerry, Mr. M. Mogford, Iman and Zoheir.

Special thanks are due to Mrs. J. Rivellini, the secretary of the Control Engineering Centre who assisted me during the work on this project.

I would like to thank the Iranian Government for supporting this research.

At last, but not least, my thanks to my family, specially my father, sister, brother, uncle and my brother in law for their enduring patience and constant support and encouragement.

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 papers, based on the work described in this thesis have been submitted for publication :

Amini, Z., Lin, J. and Roberts, P.D. (1990) : "A Single Iterative Algorithm with Global Feedback for Integrated System Optimisation and Parameter Estimation of Large Scale Industrial Processes: Optimality, Convergence and Simulation", IEE proceedings Pt. D., (to appear).

Lin, J., Amini, Z. and Roberts, P.D. (1990) : "A New Coordination Strategy for On-line Hierarchical Optimising Control of Large Scale Industrial Processes", Submitted to Int. J. Control (to appear).

ABSTRACT

The work presented in the thesis is concerned with on-line optimising control of large scale industrial processes. Theoretical analysis has been carried out to investigate optimality and convergence features of various optimising control algorithms in both centralised and hierarchical forms, providing a basis for algorithm design and assessment. Important issues, such as iterative strategies, coordination methods and feedback structures, concerning the improvement of algorithm efficiency are explored. An improved price updating formula is proposed and implemented in the single iterative loop Integrated Optimisation and Parameter Estimation (ISOPE) structure with global feedback to further improve the convergence features of the algorithm. A new coordination technique — Modifier Coordination (MC) method is proposed and implemented in both single and double iterative ISOPE structures. Approaches for coping with output dependent constraints are examined and the Penalty Relaxation (PR) technique is integrated into the ISOPE structure to extend the existing ISOPE algorithms so as to cover many output dependent cases. Comparative studies of some newly developed algorithms, techniques and methods based on substantial computer simulations are also provided. Issues concerning software implementation of optimising control algorithms are discussed providing a general guideline to such practices. Suggestions for future research as a continuation of the work presented in this thesis are also made.

LIST OF SYMBOLS

Symbol	Description	Chapter
$\mathbb{C}\mathbb{U}\mathbb{Y}$	feasible set of control, input and output mapping	2,3,4,5
$\mathbb{R}^m$	m-dimensional real space	2,3,4,5
$Q(\cdot\ ,\ \cdot)$	performance index	2,3,4,5
$q(\cdot)$, $q(\cdot\ ,\ \cdot)$	model-based performance indices	2,3,4
$q'(v)$	gradient of $q(v)$ which is the partial derivative of model-based performance index q with respect to v	2,3,4
$q_*(\cdot)$	real performance index	2,3,4
$q_*'(v)$	gradient of $q_*(v)$ which is the partial derivative of the real performance index q_* with respect to v	2,3,4
c	model-based solution for controller set-point	2,3,4,5
v	controller set-point	2,3,4
u	interactive input	3,4,5
y	output	2,3,4,5
$K_*(c)$	real process mapping	2,3,4
$F(\cdot)$, $F(\cdot\ ,\ \cdot)$	process model	2,3
$f(\cdot)$, $f(\cdot\ ,\ \cdot)$	local inequality constraint mapping	2,3,4
$G(\cdot\ ,\ \cdot)$	output dependent inequality constraint mapping	2

$\mathbb{A}$	finite dimensional space of dimension t	2,3,4
α	model parameter vector	2,3,4,5
$\hat{c}$	model-based optimum controller set-point	2,3,4
ϵ_v	iterative gain for updating the control vector v	2,3,4
L	normal Lagrangian function	2,3,4
$\tilde{L}_*$	augmented Lagrangian function	4,5
λ	modifier	2,3,4
η, ξ, μ	Lagrange multipliers	2,3,4
$(\cdot)^T$	transpose of $(\cdot)$	2,3,4
$(\hat{\cdot})$	optimum of $(\cdot)$	2,3,4
$(\cdot)^0$	starting point of $(\cdot)$	2,3,4
$(\cdot)_*$	denoting reality of $(\cdot)$	2,3,4,5
$\lVert \cdot \rVert$	Euclidean norm of $(\cdot)$	2,3,4
$(\cdot)'$	gradient of $(\cdot)$ which is the transpose of the partial derivative of $(\cdot)$	2,3,4
Ω	solution set of the model-based optimisation problem	2,4
Ω_*	optimum solution set of the real optimisation problem	2,4
$\cup$	union	2,5
$\cap$	intersection	4

$\subset$	subset	2
$\exists$	there exists	2,4
A_*	Lipschitz constant	2,4
b, b(α)	monotone constant	2
P(v,ξ)	projection operator	2
$\in$, $\notin$	belongs to, does not belong to	2,3,4,5
ρ	penalty coefficient	2,3,4
c_i, u_i, y_i	i^{th} subsystem control, input and output vectors, respectively	3,4
H	interconnection matrix	3,4
p	price vector	3,4
ϵ_p	iterative gain for updating the price vector p	3,4
$\tilde{c}_i$, $\tilde{u}_i$	optimum controller set-point and input for the i^{th} subsystem in the iteration	4
$u(\cdot\ , v^k)$	linear approximate model	3
τ	augmentation coefficient	3,4,5
$O(x^r)$	all the terms x, of order r and more	4
o(r)	infinitesimal of r	4
$Z(\cdot\ , \cdot)$	Zangwill function	4

LIST OF ABBREVIATIONS

Abbreviation	Description
ISOPE	integrated system optimsation and parameter estimation method
TS	two-step method
MTS	modified two-step method
NM	new modifier method
PR	penalty relaxation technique
ROP	real optimisation problem
EOP	equivalent real optimisation problem
MOP	model-based optimisation problem
MMOP	modified model-based optimisation problem
HROP	hierarchical real optimisation problem
HEOP	hierarchical equivalent real optimisation problem
HMMOP	hierarchical modified model-based optimisation problem
MC	modifier coordination
min	minimise
Lim	limit of

sup	supremum
inf	infimum
SIA	single iterative algorithms
DIA	double iterative algorithms
AL1	single iterative ISOPE algorithm with global feedback
AL2	single iterative ISOPE algorithm without input dependent inequality constraints, using modifier coordination strategy
AL3	single iterative ISOPE algorithm with input dependent inequality constraints, using modifier coordination strategy
AL4	double loop ISOPE algorithm using price coordination
AL5	double loop ISOPE algorithm using modifier coordination
IMTS	improved modified two-step
VA	variable augmentation
TM	two model approach
PC	price coordination
MC	modifier coordination
PPFA	process perturbation and finite difference approximation
CAD	computer aided design

CHAPTER 1

INTRODUCTION

1.1 ON-LINE PROCESS OPERATIONAL CONTROL

The purpose of process operational control is to find, when the process is first put into operation, and to maintain, when the process is continuously disturbed by a slowly changing environment, the optimal operating conditions for a production process, so that it brings about the maximum benefit at the minimum cost.

An early attempt to implement such a process operational control scheme may be traced back to 1970 when computers were introduced in on-line process control (Youle and Duncanson, 1970). Simple approaches, such as trial and error and hill-climbing, were employed to adjust the controller set-points of the process so that the process performance would be improved. With the rapid development of the computer industry, the power and capacity of computers have been greatly increased. This makes it possible to approach many complicated on-line process operational control problems quantitatively, which were considered impractical before. This is in fact the reason why process operational control is becoming more and more popular nowadays.

Even in the early studies of process operational control, it was recognised that it was different from dynamic control in principle. The important feature, which distinguishes process operational control from dynamic control, is that the quality of the control is evaluated according to the resulting steady state performance of the process rather than its transient behaviour. This implies, from a practical point of view, that after each application of the new controller set-points, it is necessary to wait, before taking any process measurements, until the process resumes its steady state (Findeisen et al., 1980). As a result, the total number of controller set-point changes required to reach the optimum becomes crucial in the application of process optimising control (Ellis, Roberts and Michalska, 1986; Lin, Kambhampati and Roberts, 1989). This is in fact one of the reasons why the above mentioned simple approaches are seldom used in the practical world.

Since steady state optimisation is involved when locating the optimum operating conditions, finite dimensional optimisation plays an important role in process operational control. The application of optimisation methods and modelling techniques results in a systematic approach to process operational control. The main idea of such an approach is to use a mathematical model, which is built upon the available knowledge of the process, to replace the real process in the optimisation. Since the real process information is not required in the model-based optimisation, a great deal of reduction in the number of controller set-point changes may be expected. However, due to the inadequacy of the available knowledge of the process and also to the changing nature of the process, there is bound to be model-reality differences which, consequently, result in inaccuracy in the application of pure model-based optimisation methods (Roberts, 1979). This is the reason why pure model-based optimisation techniques generally fail to produce the true optimum where there are significant model-reality differences. To tackle this problem, feedback techniques are employed to adjust the model-based optimisation by making use of the real process information in modifying optimisation algorithms (Findeisen et al., 1980) and in model adaptation (Roberts, 1979).

1.2 OPTIMISING CONTROL OF INDUSTRIAL PROCESSES

As mentioned in Section 1.1, in order to keep up with the ever changing nature of process and environment, the model has to be adaptive. This results in the model parameter estimation problem. Combining parameter estimation and system optimisation processes together establishes the Two-Step (TS) method (Haimes and Wismer, 1972). Since there is interaction between parameter estimation and system optimisation, the TS method is generally sub-optimal (Durbeck, 1965; Foord, 1974). It is observed that the sub-optimality of the TS method is dependent on the model-reality differences and the choice of initial values (Chen, 1986). This makes it impossible, in most cases, to estimate accurately the deviation of the algorithm solution from the real optimum. In order to find and maintain the true optimum, regardless of the changing environment, it is necessary to develop an optimising control method such that the optimality of the method is independent of the model-reality differences.

1.2.1 Integrated System Optimisation and Parameter Estimation

In order to design an algorithm which is capable of producing true optimum regardless of model-reality differences, it is necessary to find a way to cater for the interaction between parameter estimation and model-based optimisation problems. This is achieved by introducing a modifier into the model-based optimisation problem so that the interaction between system optimisation and parameter estimation is compensated for at the end of algorithm iteration. This leads to a Modified Two-Step (MTS) approach which is also called Integrated System Optimisation and Parameter Estimation (ISOPE) (Roberts, 1979). The appealing feature of the ISOPE method is that it is capable of producing the true optimum, regardless of model-reality differences. The ISOPE method has later been extended to hierarchical process optimisation with different feedback structures for the purpose of handling large scale problems (Brdyś and Roberts, 1986).

1.2.2 Decomposition and Coordination

As the optimising control problems to be solved are getting closer to reality, their complexity is increasing sharply. In the past decade distributed computer control has become more and more popular in the process industry, resulting in a rapid expansion in the scale of the process operational control problems. Thus, the overall or centralised process operational control approaches have become less attractive to todays process industry. In the practice of distributed computer control, it is common to have a two-level computer system dedicated to the purpose of process supervision and control. Since this two-level computer system assumes a hierarchical structure, a decomposition and coordination scheme may be more favourable than a centralised optimising control approaches.

A practical way of handling large scale process optimisation consisting of several interconnected sub-processes is to decompose the overall problem into interconnected sub-problems of manageable size so that they can be solved hierarchically (Singh and Titli, 1978). This leads to a two-level structure in which the sub-problems are solved independently in the lower level and the individual sub-problem solutions are coordinated by a coordinator in the upper level.

Two different coordination strategies have been mostly used in hierarchical process operational control, Interaction Balance method (Price method) and Interaction Prediction method (Direct method) (Findeisen et al., 1980; Singh and Titli, 1978). Since the direct method is usually difficult to converge and there are restrictions on its application, the price method has been widely used in the coordination of hierarchical optimisation (Findeisen et al., 1980; Singh and Titli, 1978). A major disadvantage of the price method, recognised through the application of hierarchical optimising control, is that the coordination efficiency depends closely on the Hessian structure of the real optimisation problem and the convergence of the algorithm may be very slow towards the end of iteration especially when the problem is ill-conditioned (Lin, Roberts, Wang and Wan, 1990; Lin, Amini and Roberts, 1990). To tackle this problem, a new coordination strategy is proposed (Lin, Amini and Roberts, 1990). In this approach, the interaction between subsystems is considered as a kind of model-reality difference and is, therefore, taken care of by a specially designed modifier (Ellis, Michalska and Roberts, 1984; Lin, Amini and Roberts, 1990). Since the coordination task is performed by the modifier which has a more desirable convergence feature than the price coordinator, the coordination efficiency is significantly improved.

1.2.3 Iterative Strategies

According to the nature of hierarchical optimisation, the interaction balance between subsystems is achieved in an iterative way. Two different iterative strategies, namely single iterative loop and double iterative loop strategies, can be employed to implement a hierarchical optimising control algorithm. In the single iterative strategy, the controller set-point is updated simultaneously with the coordinator while, in the double iterative strategy, a nested double iterative loop structure is employed such that the subsystem coordination is separated from the controller set-point optimisation.

The single iterative strategy aims at reducing the total number of algorithm iterations while the double iterative strategy is designed to reduce the real process iteration at the price of a possible increase in off-line computation (Shao and Roberts, 1983; Hendawy, 1989). The single iterative strategy becomes attractive when there is a limitation in computer power and the process is of fast transient nature. On the other hand, the double iterative strategy is more suitable for processes with slow transient nature.

1.2.4 Output Dependent Constraints

It is often required that the controlled process always operates within the region specified by the safety standards and capacity limits. This results in certain constraints on the process operational control. Generally, there are two kinds of constraints, i.e. local and output dependent constraints. The constraints imposed on the controller set-points only, are referred to as the local constraints, while the constraints imposed on both controller set-points and process outputs are called the output dependent constraints. Due to the difficulties in analysis, output dependent constraints are usually avoided in the early development of the ISOPE methods (Brdyś and Roberts, 1985; Tatjeweski, Abdullah and Roberts, 1990), although such cases are often found in practical situations.

The earliest attempt to extend the ISOPE method so that it can deal with output dependent constraints was to augment the modifier to take into account the model-reality differences in constraints (Brdyś, Chen and Roberts, 1986). The major disadvantage of this technique is that the feasibility of the model-based optimisation problem is not ensured during the course of iteration (Lin, Chen and Roberts, 1988) and this will cause a great deal of difficulty in the on-line implementation of the algorithm. To get around this problem, a New Modifier method (NM) is proposed in which the model-reality differences in the constraints are compensated for by introducing a new modifier into the model-based optimisation problem (Lin, Chen and Roberts, 1988; Lin, Hendawy and Roberts, 1988a). The disadvantage of the NM method is that the convergence is rather slow, especially when the output dependent constraints are tight at the optimum. A more efficient approach to problems with output dependent constraints is the Penalty Relaxation (PR) technique. In this technique, the convergence features of the standard ISOPE method are retained and it has been found to be the most efficient technique to date for solving such problems (Lin, Kambhampati and Roberts, 1989; Lin, Li, Wan and Roberts, 1989).

1.3 SCOPE AND AIMS OF THE THESIS

In the past decade, a great deal of effort has been put into the research concerned with on-line optimising control theory. Efficient algorithms have been developed in an attempt to meet requirements from industry, which are summarised in Chapter 6, and theoretical analyses have been carried out to ensure their stability. Important issues such as development of more efficient coordination methods, convergence and stability studies for hierarchical optimising control algorithms and investigation of their applicability conditions, techniques for dealing with output dependent constraints, have been receiving ever increasing attention from both practical and theoretical fields.

During the past few years, optimising control theory has been successfully applied in the field of process design. This has initiated the research on the software development of optimising control theory. Emphasis is given to issues such as algorithm specification, design policies, modular structures and man-machine interface.

The aim of the thesis is to further explore the above mentioned topics and to push forward the research by improving the existing results. The scope and contribution of the thesis are briefly summarised as follows :

1.3.1. Contribution to Theoretical Analysis

A thorough study on optimality and convergence of ISOPE algorithms in both centralised and hierarchical cases is provided in the thesis. In the centralised case, the applicability conditions for the standard MTS method are investigated, providing an insight into the practical meaning of such conditions. In the hierarchical case, detailed analysis has been carried out to study the optimality and convergence features of the four newly developed hierarchical optimising control algorithms. The importance of the theoretical work included in this thesis lies in the fact that it provides a more complete theoretical basis for algorithm specification in on-line hierarchical optimising control.

1.3.2. Contribution to Algorithm Development

A new coordination method, which is much more efficient than the price method in many cases, has been developed (Lin, Amini and Roberts, 1990). The main feature of this method is that it retains the convergence properties of the standard MTS method which is usually considered efficient in most cases. To improve the efficiency of the single loop ISOPE algorithms, a global feedback structure is employed giving rise to an improved single iterative ISOPE method (Amini, Lin and Roberts, 1990). In order to extend the ISOPE algorithms so that they can be applied to output dependent cases, a penalty relaxation technique has been integrated into the modifier coordination structure. Augmentation techniques are also introduced into the newly developed ISOPE algorithms to further increase on-line efficiency. These works are considered as a continuation of the efforts made in this area to develop and improve optimising control algorithms.

1.3.3 Contribution to Algorithm Assessment

A number of newly developed optimising control algorithms in both centralised and hierarchical forms have been assessed in the thesis. Techniques for dealing with output dependent constraints, such as a new modifier method and a penalty relaxation technique, and approaches for increasing the algorithm efficiency, such as the use of augmentation techniques, are also examined. Substantial simulations have been carried out to justify these assessments. This provides useful information for the algorithm specification in the practice of on-line optimising control of industrial processes.

1.3.4. Contribution to Software Implementation Methodology

Important issues concerning software implementation of optimising control algorithms are discussed in the thesis. Based on requirements from industry, a number of criteria for algorithm specification are established. A new design idea of forming an algorithm bank, which can produce a family of optimising control algorithms to cover most practical situations, is proposed and applied in a design scheme for an optimising control Computer Aided Design (CAD) package. A possible software configuration of such a package is also discussed. The work presented provides a general guideline for software development of an optimising control CAD package.

1.4 OUTLINE OF THE THESIS

The outline of the thesis is as follows :

In Chapter 2, the basic concept of on-line process optimisation is introduced in a centralised framework. Theoretical analysis is carried out to investigate optimality and convergence features of three basic solution approaches i.e. Two-Step (TS), Modified TS (MTS) and Improved MTS (IMTS) methods. Applicability conditions for the MTS method are discussed, providing insight into the convergence features of the algorithm. Techniques for dealing with output dependent constraints are also introduced and assessed according to their convergence features.

Chapter 3 mainly focuses on the issue of developing efficient hierarchical Integrated System Optimisation and Parameter Estimation (ISOPE) algorithms in different iterative structures. A new coordination strategy — modifier coordination is proposed to increase the coordination efficiency of hierarchical optimising control algorithms. The Penalty Relaxation (PR) technique is integrated with the modifier coordination strategy to extend ISOPE algorithms so as to cover output dependent cases. Augmentation techniques are introduced to reduce the algorithm sensitivity and the number of on-line iterations.

In Chapter 4, a theoretical study of optimality and convergence of the hierarchical ISOPE algorithms developed in Chapter 3 is presented. Sufficient conditions for the convergence of these algorithms are provided.

Chapter 5 is dedicated to the simulation studies of the convergence features of the hierarchical ISOPE algorithms developed in Chapter 3. Based on the simulation results obtained from three different examples, a comparative study of on-line and off-line efficiency and algorithm sensitivity is presented.

In Chapter 6, some general views on software development of ISOPE algorithms for on-line process operational control are presented. A novel idea of forming an algorithm bank to cover different practical situations is proposed. A number of criteria are worked out for the purpose of algorithm specification. A practical scheme for the development of a CAD package for process operational control is also presented.

Chapter 7 is the concluding part of the thesis, which provides a brief summary of the results achieved and suggestions for future research in this area.

CHAPTER 2

ON-LINE PROCESS OPTIMISATION --- CENTRALISED CASE

2.1 INTRODUCTION

The aim of on-line process operational control is to find and maintain the optimum operating conditions for a production process, bringing about the maximum profit at the minimum cost. According to the nature of process control a change in the operating conditions for a process is realised by changing its controller set-points. Generally, there are two different ways to locate the optimum controller set-points. One is called the direct approach where the optimum operating conditions are achieved by directly improving the controller set-points applied in a trial and error manner while, the other is based on a systematic optimisation procedure in which a mathematical model is used in the process of system optimisation. In general, the direct approach requires a much larger number of controller set-point changes to locate the correct optimum operating condition than the optimisation approach. This explains why the optimisation approach receives much more attention than the direct method in the practice of process operational control.

In the optimisation approach mentioned above, a process model, which is built upon the available knowledge of the system, is used in the process optimisation. Since the available knowledge of the process may not be adequate and the process itself as well as the environment in which it operates keeps changing, there is bound to be model-reality differences (Roberts, 1979). This will result in inaccuracy in the process optimisation. In order to keep up with the change in the process and environment, the model has to be adaptive. This establishes the model parameter estimation problem. Combining parameter estimation and system optimisation processes together one obtains a method called the Two-Step (TS) method. Since there is interaction between parameter estimation and system optimisation, the TS method is sub-optimal and the sub-optimality is dependent on the model-reality difference. To separate these two processes a modifier is introduced in the model-based optimisation problem and this leads to the Modified Two-Step (MTS) approach which is also called Integrated System Optimisation and Parameter Estimation (ISOPE) (Roberts, 1979; Brdyś and Roberts, 1986).

In this chapter the standard TS and MTS as well as its improved version are introduced and thoroughly analysed. The New Modifier (NM) method and the Penalty Relaxation (PR) technique are presented for problems with output dependent constraints.

2.2 FORMULATION OF THE PROBLEM

In order to preserve the physical meaning of the formulation, we start with clarifying the fundamental elements of an optimisation problem i.e. objective function, decision variables and constraints, following the basic principles of process operational control.

2.2.1 Performance Index

Since the purpose of process optimisation is to achieve the optimum operational performance for the process, a certain criterion should be clarified to measure the performance of the process. In order to perform optimisation on system performance, this criterion needs to be expressed in a quantitative form, so that different degrees of optimum system performance can be analysed. In practice the aim of adjusting process operation is usually to achieve maximum profit at the minimum cost. This can easily be expressed in terms of the selling price of the products produced and the cost of the materials and energy consumed in the production process. Generally, the products, materials and energy mentioned above are associated with the output of the process and the controller set-point which is also called the manipulative variable in some cases. Therefore, a mathematical function of c, the controller set-point, and y, the output of the process, can be used to represent the above mentioned criterion, i.e.

$$Q : \mathbb{C} \times \mathbb{Y} \rightarrow \mathbb{R} \quad \text{with } \mathbb{C} \subset \mathbb{R}^n,\ \mathbb{Y} \subset \mathbb{R}^m \text{ and}$$

$$Q = Q\,(c,y) \tag{2.1}$$

2.2.2 Optimisation Variables

Based on the idea of hierarchical multi-layer process control the entire system is divided, according to different functions, into at least two layers, i.e. direct control layer and optimisation layer (Findeisen, et al., 1980). The task of the direct control layer is to keep the output of the controlled process as close as possible to the given controller set-points under whatever disturbances. In the optimisation layer the task is to generate the optimum controller set-points. Applying the optimum controller set-points to the direct control layer results in the optimum steady state performance of the process. Unlike the dynamic control problem in the direct control layer the optimisation of the controller set-point is carried out in a purely steady state fashion. Therefore, here the optimisation variables should be the controller set-points which are defined in a finite dimensional space.

2.2.3 Process Mapping

From a mathematical point of view process mapping is nothing more than equality constraints in an optimisation problem. It is, however, important to identify the mapping from the constraints imposed by the process designer or operator because this mapping reflects reality rather than the desire of a human being. In general, the process mapping is unknown and it is impossible to obtain accurate knowledge about it without taking proper measurements. In the practice of process operational control, given a controller set-point one can always expect, if the process is properly regulated, a steady state process output after a sufficiently long period of time. This means the process mapping exists and is well defined. It maps from the domain for the controller set-points onto the region for the steady state process outputs, i.e.

$$K_* : \mathbb{C} \rightarrow \mathbb{Y}$$

$$y = K_*(c) \tag{2.2}$$

2.2.4 Process Operation Constraints

In order to ensure that the process under control always operates within the region specified by the safety standards and capacity limits it is necessary for the designer to impose some constraints on the process operation and for the operator to direct the process within these limitations. Although these constraints are usually imposed on both controller set-point and process output, it is worth while identifying the local constraint, which is dependent solely on the controller set-point, from the output dependent constraint which is dependent not only on the controller set-point but also on the process output i.e.

$$f(c) \leq 0 \quad \text{with } f: \mathbb{C} \rightarrow \mathbb{R}^r \tag{2.3}$$

$$G(c,y) \leq 0 \text{ with } G: \mathbb{C} \times \mathbb{Y} \rightarrow \mathbb{R}^s \tag{2.4}$$

Based on the above disscusion, a steady state optimising control problem is formulated as follows:

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

(ROP)

s.t.

$$y = K_*(c)$$

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

$$c \in \mathbb{C}$$

where

$$\mathbb{C} = \left\{ c \in \mathbb{R}^n : f(c) \leq 0 \right\} \tag{2.5}$$

c and y are the controller set-point and process output vectors, respectively.

In order to develop efficient solution algorithms for the optimisation problem ROP several basic assumptions are made on the above formulation. It is recognised in the practice of optimising control that these assumptions are usually met due to the nature of the problem.

Assumption 2.1

It is assumed throughout this chapter that:

1. The performance index $Q(\cdot\ ,\ \cdot)$, the constraint mappings $f(\cdot)$ and $G(\cdot\ ,\ \cdot)$, and the process mapping $K_*(\cdot)$ are continuously Fréchet differentiable;

2. $Q(\cdot\ ,\ \cdot)$ is strictly convex;

3. $f(\cdot)$ and $G(\cdot\ ,\ \cdot)$ are convex.

2.3 STANDARD TWO-STEP AND MODIFIED TWO-STEP METHODS

It was seen in Section 2.2 that the formulation of ROP is virtually a non-linear programming problem. It is, however, worth mentioning that unlike ordinary optimisation problems, an optimising control problem usually has uncertain constraints due to the fact that the process mapping $K_*(\cdot)$ in ROP is generally unknown.

Two different types of approaches can be developed for the solution of ROP. One possibility is to employ a direct on-line search approach which requires a great number of process measurements and the other is to use a system model to approximate the process mapping $K_*(\cdot)$ in ROP. In practice, algorithms for the direct on-line optimisation approach are not easy to implement because, due to the slow nature of the process dynamics and also the presence of noise, the performance of such algorithms deteriorates. Therefore, model-based methods provide alternative approaches to optimising control problems. The mathematical model used in the model-based optimisation problem contains parameters whose values are estimated by comparing output responses from the model with the corresponding responses from the real process, defining the system identification or parameter estimation problem. In turn, the mathematical model is used to determine the optimum controller set-points, so that a model-based optimum process performance is achieved, defining the model-based optimisation problem. A successive solution of the above mentioned problems forms an iterative procedure which is repeated until convergence is achieved. This is in fact the basic idea of the standard Two-Step (TS) method (Haimes and Wismer, 1972).

2.3.1 Two-Step Technique

As mentioned above, the solution of ROP using direct on-line search is inefficient and therefore, a process model is used to approximate the process mapping $K_*(\cdot)$ in ROP. i.e.

$F: \mathbb{C} \times \mathbb{A} \rightarrow \mathbb{Y}$ with

$$y = F(c\ ,\ \alpha) \tag{2.6}$$

where α is the parameter of the process model and $\mathbb{A}$ is a finite dimensional space of dimension t.

In order for the process mapping $F(\cdot\ ,\ \cdot)$ to match that of the real process, a parameter estimation problem is defined as follows

$$F(c\ ,\ \alpha) = K_*(c) \tag{2.7}$$

The following assumption is needed to clarify the parameter estimation problem :

Assumption 2.2

It is assumed that the model mapping $F(\cdot\ ,\ \cdot)$ in equation (2.7) is point-parametric (Brdyś, 1983).

It can be proved under assumption 2.2 that the real process optimisation problem ROP is equivalent to the following optimisation problem (Brdyś, 1983):

$$\text{(EOP)} \qquad \min_{c,\alpha} \ Q(c,\ F(c,\alpha))$$

s.t.

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

$$G(c\ ,\ F(c\ ,\ \alpha)) \leq 0$$

$$(c\ ,\ \alpha) \in \mathbb{C} \times \mathbb{A}$$

Following the basic idea discussed at the begining of section 2.2, the standard Two-Step (TS) method can be implemented as follows :

Choose an initial set-point $v^0 \in \mathbb{C}$ and set $k = 0$;

Step 1 Apply v^k to the real process and measure its output $K_*(v^k)$. Then estimate the model parameter α^k by solving the following equation:

$$F(v^k, \alpha^k) = K_* (v^k) \tag{2.8}$$

Step 2 Solve the following model-based optimisation problem

$$\min_c Q(c, F(c,\alpha^k))$$

(MOP)

s.t.

$$F(c, \alpha^k) = K_* (v^k)$$

$$G(c , F(c , \alpha^k)) \leq 0$$

$$c \in \mathbb{C}$$

to obtain a model-based optimum $\hat{c}(v^k)$ or $\hat{c}^k$ for short.

The controller set-point can be updated by passing the value of $\hat{c}^k$ directly to v^{k+1} and the convergence of the algorithm is achieved when $\hat{c}^k = v^k$. The information structure for this technique is shown in Figure 2.1.

Remark 1 In the practical implementation of the algorithm, the requirement $\hat{c}^k = v^k$ is replaced by an inequality $\left\| \hat{c}^k - v^k \right\| < \epsilon$, where $\epsilon > 0$ is a desired accuracy threshold.

Remark 2 An alternative to the controller set-point updating strategy described above is to use the following simple relaxation technique :

$$v^{k+1} = v^k + \epsilon_v (\hat{c}^k - v^k) \tag{2.9}$$

where $0 < \epsilon_V \leq 1$ is the gain factor which is chosen to control the convergence rate of the algorithm. This technique is found to be very helpful in some cases to the convergence of the algorithm (Brdyś and Roberts, 1987).

Remark 3 In general the standard TS method only gives a sub-optimal solution to the real optimisation problem ROP as is observed in the example given by Chen (1986). This is due to the fact that it is difficult to match the output as well as the first order derivatives of the real process and the corresponding model at the optimum controller set-point when there is a structural difference between the model and reality.

In general, a mathematical model will not be a true representation of the real system. This causes the two problems to interact in that the solution of the optimisation problem is dependent upon the values of the model parameters and the parameter estimates will change according to the set-points. Such a two-step approach as mentioned above, cannot overcome the model-reality structure differences and, therefore, a sub-optimal solution results.

It is discussed by Foord (1974), Ellis and Roberts (1982) and Chen (1986) that a sufficient condition for the standard TS method to give the correct optimum for ROP is that there are no model-reality differences up to the first order. This explains the reason why the standard TS usually produces sub-optimum results when there are model-reality structure differences.

2.3.2 The Modified Two-Step Technique

The modified two-step approach (Roberts, 1987) is an improved version of the two-step technique in that it incorporates a modifier to compensate for the model-reality differences, and results in an optimal solution. Since this technique integrates the parameter estimation and optimisation problems together, the above algorithm is referred to as the Integrated System Optimisation and Parameter Estimation (ISOPE) technique. A constructive way of deriving the ISOPE method is to use the Lagrangian approach (Roberts, 1979).

Rewrite EOP as:

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

(EOP1)

s.t.

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

$$g(c, \alpha) \leq 0$$

$$c \in \mathbb{C}$$

where :

$$q(c, \alpha) = Q(c, F(c, \alpha)) \tag{2.10}$$

$$g(c, \alpha) = G(c, F(c, \alpha)) \tag{2.11}$$

In order to decouple the optimisation and parameter estimation problems incorporated in EOP1, an additional equality constraint $v = c$ is introduced to convert the above optimisation problem into:

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

(EOP2)

s.t.

$$F(v, \alpha) = K_*(v)$$

$$g(c, \alpha) \leq 0$$

$$c \in \mathbb{C}$$

$$v = c$$

The problems EOP1 and EOP2 are equivalent (Chen, 1986) and in turn EOP is equivalent to ROP as discussed earlier. Therefore, the optimality conditions of EOP1 and EOP2 are identical. The Lagrangian function associated with EOP2 is :

$$L(c,v,\alpha,\lambda,\eta,\xi,\mu) = q(c, \alpha) + \lambda^T(v - c) + \eta^T \left[F(v, \alpha) - K_*(v) \right]$$

$$+ \xi^T g(c, \alpha) + \mu^T f(c) \tag{2.12}$$

where λ, η, ξ and μ are Lagrange multipliers and f(c) is defined in (2.5).

Assuming that all the required derivatives, including $\frac{[\partial^T F(v,\alpha)]^{-1}}{\partial \alpha}$ exist and the regularity conditions for constraints are satisfied then the Kuhn-Tucker conditions are as follows :

$$\frac{\partial^T L}{\partial c} = \frac{\partial^T q(c,\alpha)}{\partial c} - \lambda + \frac{\partial^T g(c,\alpha)}{\partial c}\, \xi + \frac{\partial^T f(c)}{\partial c}\, \mu = 0 \tag{2.13}$$

$$\frac{\partial^T L}{\partial v} = \lambda + \left[\frac{\partial^T F(v,\alpha)}{\partial v} - \frac{\partial^T K_*(v)}{\partial v} \right] \eta = 0 \tag{2.14}$$

$$\frac{\partial^T L}{\partial \alpha} = \frac{\partial^T q(c,\alpha)}{\partial \alpha} + \frac{\partial^T F(v,\alpha)}{\partial \alpha}\, \eta + \frac{\partial^T g(c,\alpha)}{\partial \alpha}\, \xi = 0 \tag{2.15}$$

$$\frac{\partial^T L}{\partial \lambda} = v - c = 0 \tag{2.16}$$

$$\frac{\partial^T L}{\partial \eta} = F(v\ ,\ \alpha) - K_*(v) = 0 \tag{2.17}$$

$$\frac{\partial^T L}{\partial \mu} = f(c) \leq 0 \tag{2.18}$$

$$\frac{\partial^T L}{\partial \xi} = g(c\ ,\ \alpha) \leq 0 \tag{2.19}$$

$$\xi^T \frac{\partial^T L}{\partial \xi} = \xi^T g(c\ ,\ \alpha) = 0\ ,\ \ \xi \geq 0 \tag{2.20}$$

$$\mu^T \frac{\partial^T L}{\partial \mu} = \mu^T f(c) = 0\ ,\ \ \mu \geq 0 \tag{2.21}$$

Solving η from (2.15) and substituting it into (2.14), it is not difficult to obtain the formula for λ :

$$\lambda = \lambda(c,v,\alpha,\xi) = \left[\frac{\partial^T F(v,\alpha)}{\partial v} - \frac{\partial^T K_*(v)}{\partial v} \right] \left[\frac{\partial^T F(v,\alpha)}{\partial \alpha} \right]^{-1} \left[\frac{\partial^T q(c,\alpha)}{\partial \alpha} + \frac{\partial^T g(c,\alpha)}{\partial \alpha}\, \xi \right] \tag{2.22}$$

Since v = c, (2.22) can be rewritten as :

$$\lambda = \lambda(v,\alpha,\xi) = \left[\frac{\partial^T F(v,\alpha)}{\partial v} - \frac{\partial^T K_*(v)}{\partial v} \right] \left[\frac{\partial^T F(v,\alpha)}{\partial \alpha} \right]^{-1} \left[\frac{\partial^T q(v,\alpha)}{\partial \alpha} + \frac{\partial^T g(v,\alpha)}{\partial \alpha}\, \xi \right] \tag{2.23}$$

Therefore, EOP2 can be reformulated in the following form :

$$\min_{c} \left\{ q\,(\,c\,,\,\alpha\,) - \lambda^{T}(\,c\,) \right\}$$

s.t.

$$g\,(\,c\,,\,\alpha\,) \leq 0$$

$$c \in \mathbb{C}$$

For any given v and ξ, $\hat{\alpha}(v)$ and λ are calculated according to formulae (2.17) and (2.23). The solution of the above optimisation problem and associated Lagrange multipliers are denoted by $\hat{c}(v,\alpha,\xi)$, $\hat{\xi}(v,\alpha,\xi)$ and $\hat{\mu}(v,\alpha,\xi)$, respectively. Then any solution of the operator equations :

$$\hat{c}(v\,,\,\hat{\alpha}(v)\,,\,\xi) = v \tag{2.24}$$

and

$$\hat{\xi}(v\,,\,\hat{\alpha}(v)\,,\,\xi) = \xi \tag{2.25}$$

is a solution of (2.13) to (2.21).

For the time being, we assume that the system inequality constraints do not depend upon the system output, and output dependent cases are covered in Section 2.5. Then,

$$\frac{\partial^{T} g(v,\alpha)}{\partial \alpha}\,\xi = 0$$

and (2.24) and (2.25) reduce to :

$$\hat{c}(v\,,\,\hat{\alpha}(v)) = v$$

and (2.22) to :

$$\lambda = \lambda(c,v,\alpha) = \left[\frac{\partial^{T} F(v,\alpha)}{\partial v} - \frac{\partial^{T} K_{*}(v)}{\partial v} \right] \left[\frac{\partial^{T} F(v,\alpha)}{\partial \alpha} \right]^{-1} \frac{\partial^{T} q(c,\alpha)}{\partial \alpha} \tag{2.26}$$

Based on the above arguments, the algorithmic implementation of the method can be described as follows :

Choose some initial set-points $v^o \in \mathbb{C}$, Lagrange multipliers $\xi \in \mathbb{R}^r$ and parameters β.

<u>Step 1</u> Apply v^k to the real process and measure its outputs $K_*(v^k)$. Determine a new parameter value α^k by solving :

$$F(v^k,\alpha) = K_*(v^k)$$

Perform additional perturbations about v^k and measure corresponding process outputs to compute finite difference approximations of the derivatives :

$$\frac{\partial^T K_*(v^k)}{\partial v}$$

Finally, calculate $\lambda^k = \lambda\ (\ \hat{c}^{k-1},\ v^k,\ \alpha^k,\ \xi^k)$ using equation (2.22).

λ^k can also be calculated using equation (2.23). In this case, $\lambda^k = \lambda\ (\ v^k,\ \alpha^k,\ \xi^k\ )$.

<u>Step 2</u> Solve the following model-based optimisation problem,

$$\text{(MMOP)} \qquad \begin{array}{l} \min\limits_{c} \left\{ q\,(\,c\,,\,\alpha\,) - \lambda^T(\,v\,) \right\} \\ \\ \text{s.t.} \quad c \in \mathbb{C} \end{array}$$

to obtain $\hat{c}^k$ for given α^k and λ^k.

The iteration is terminated when $\hat{c}^k = v^k$. In practice, the procedure is terminated when :

$$\left\| \hat{c}^k - v^k \right\| < \epsilon \tag{2.27}$$

where ϵ is a prescribed tolerance.

Otherwise, update the control v^k using the following simple relaxation formula :

$$v^{k+1} = v^k + \epsilon_v\ (\hat{c}^k - v^k) \tag{2.28}$$

where $0 < \epsilon_v \leq 1$ is a gain factor as used in equation (2.9). This technique has been successful in solving many example problems and gives optimal solutions (Roberts, 1979; Brdyś and Roberts, 1987).

It is seen from the above derivation that in the model-based optimisation problem MMOP a modifier is introduced to modify the performance index so that when the algorithm converges the solution of MMOP will coincide with the real optimum even if there are model-reality differences. It is also observed from equation (2.23) that the modifier becomes zero when the model-reality derivative differences vanish. This reveals the important role of the modifier in the ISOPE algorithms and that is the reason why the ISOPE method is also called the modified two-step method MTS. The information structure for the MTS method is shown in Figure 2.2.

2.3.3 An Improved Version Of The MTS Method

In this section, the standard MTS method is improved to give a wider choice for model selection. In this improved version of MTS, the parameter estimation as well as the matching between model ouputs and real outputs is no longer required.

For any given initial $v^0 \in \mathbb{C}$, the k^{th} iteration of the proposed algorithm is described as follows :

<u>Step 1</u> For given v^k, calculate the modifier according to the following equation (Brdyś and Roberts, 1987; Lin, Han, Roberts and Wan, 1989) :

$$\lambda^k = \lambda(v^k) = q'(v^k) - q'_*(v^k) \tag{2.29}$$

where $q(v) = Q(v \; , \; F(v))$, $q'(\cdot)$ is the partial derivative of q with respect to v and $F\colon \mathbb{R}^n \to \mathbb{R}^m$ is a properly chosen model.

<u>Step 2</u> For given λ^k, solve the following model-based optimisation problem

$$\text{(MOP)} \qquad \begin{aligned} &\min_c \left\{ q(c) - (\lambda^k)^T c \right\} \\ &\text{s.t.} \;\; c \in \mathbb{C} \end{aligned}$$

Denote $\hat{c}(v^k)$ as the optimum of MOP, or simply as $\hat{c}^k$.

The controller set-point v^k is updated by using the relaxation technique described in Section 2.3.2, that is :

$$v^{k+1} = v^k + \epsilon_v^k (\hat{c}^k - v^k) \tag{2.30}$$

where $0 < \epsilon_v \leq 1$ is the iterative gain factor.

The iteration is terminated when $v^{k+1} = v^k$. Figure 2.3 shows the information structure for the improved MTS method.

2.4 OPTIMALITY AND CONVERGENCE ANALYSIS

Optimality and convergence analysis for the standard and improved versions of MTS method are presented in this section. This analysis is based on the previous work of Brdyś and Roberts (1987), Chen (1986) and Kambhampati (1989).

2.4.1 Optimality

The following theorems establish the optimality of the standard and improved MTS method in the sense that every algorithmic solution satisfies the Kuhn-Tucker conditions of the real optimisation problem ROP. The following sets are defined to clarify the proof of the theorems.

Definition 2.1

$$\Omega_1 = \left\{ v : v \in \mathbb{C},\ v = \hat{c}(v\ ,\ \hat{\alpha}(v)) \right\} \text{ and } \Omega_2 = \left\{ v \in \mathbb{C},\ v = \hat{c}(v) \right\} \tag{2.31}$$

and denote Ω^* as the optimum solution set of the real optimisation problem ROP.

Theorem 2.1

Let Assumption 2.1 be satisfied, and $v \in \Omega_1$ and η be the associated Lagrange multiplier vector. Assume that for any $c \in \mathbb{C}$, the regularity conditions (Luenberger, 1984) are satisfied. Then v satisfies the Kuhn-Tucker necessary optimality conditions for ROP. If, in addition, $\mathbb{C}$ is convex and q_* is convex on $\mathbb{C}$ then v is a solution of ROP.

Proof

Obviously, v is feasible and

$$\hat{c}(v , \hat{\alpha}(v)) = v \tag{2.32}$$

According to the assumption made in the theorem, there is an $\eta \geq 0$ such that :

$$\frac{\partial^T q(\hat{c}(v) , \alpha(v))}{\partial c} - \lambda(v , \hat{\alpha}(v)) + \frac{\partial^T f(\hat{c}(v))}{\partial c} \eta = 0 \tag{2.33}$$

$$f(\hat{c}(v)) \leq 0 \tag{2.34}$$

$$\eta^T f(\hat{c}(v)) = 0 \text{ and } \eta \geq 0 \tag{2.35}$$

Since $F(v , \alpha(v)) = K_*(v)$, using the definition of modifier vector λ (equ. 2.26) as well as equation (2.32), we obtain :

$$\frac{\partial^T q_*(v)}{\partial c} + \frac{\partial^T f(v)}{\partial c} \eta = 0 \tag{2.36}$$

$$f(v) \leq 0 \tag{2.37}$$

$$\eta^T f(v) = 0 \text{ and } \eta \geq 0 \tag{2.38}$$

which is precisely the Kuhn-Tucker conditions at v for ROP. The Kuhn-Tucker conditions become sufficient for optimality if q_* is convex (Luenberger, 1984).

QED

Theorem 2.2

Let Assumption 2.1 be satisfied, and $v \in \Omega_2$ and η be the associated Lagrange multiplier vector. Assume that for any $c \in \mathbb{C}$, the regularity conditions (Luenberger, 1984) are satisfied. Then v satisfies the Kuhn-Tucker necessary optimality conditions for ROP. If, in addition, $\mathbb{C}$ is convex and q_* is convex on $\mathbb{C}$ then v is a solution of ROP.

Proof

Obviously, v is feasible and

$$\hat{c}(v) = v \tag{2.39}$$

According to the assumption made in the theorem, there is an $\eta \geq 0$ such that :

$$\frac{\partial^T q(\hat{c}(v))}{\partial c} - \lambda(v) + \frac{\partial^T f(\hat{c}(v))}{\partial c} \eta = 0 \tag{2.40}$$

$$f(\hat{c}(v)) \leq 0 \tag{2.41}$$

$$\eta^T f(\hat{c}(v)) = 0 \text{ and } \eta \geq 0 \tag{2.42}$$

Since $F(v) = K_*(v)$, using the definition of modifier vector λ (equ. 2.29) as well as equation (2.39), we obtain :

$$\frac{\partial^T q_*(v)}{\partial c} + \frac{\partial^T f(v)}{\partial c} \eta = 0 \tag{2.43}$$

$$f(v) \leq 0 \tag{2.44}$$

$$\eta^T f(v) = 0 \text{ and } \eta \geq 0 \tag{2.45}$$

which is precisely the Kuhn-Tucker conditions at v for ROP. The Kuhn-Tucker conditions become sufficient for optimality if q_* is convex (Luenberger, 1984).

QED

In order to establish the convergence theorems for the standard and improved MTS in a systematic fashion, it is helpful to express the algorithmic mapping into a composition of a series of point-to-set mappings.

2.4.2 The Algorithmic Mappings for the Standard and Improved MTS Methods

An inspection of the algorithmic descriptions for the standard and improved MTS methods shows that at each iteration, the algorithmic mapping of these two methods can be decomposed into several point-to-set mappings.

2.4.2a The Standard MTS Method

1. Parameter estimation: For $v^k \in \mathbb{C}$ determine the parameter vector α^k by solving equation (2.8). Since the solution of α need not be unique, we have in general,

$$\alpha^k \in \hat{\alpha}(v^k) \tag{2.46}$$

where $\hat{\alpha}(v^k)$ is the solution set of equation (2.8).

2. Model-based optimisation :

$$\min_{c \in \mathbb{C}} \left\{ q\,(\,c\,,\,\alpha^k\,) - (\,\lambda^k)^T\,c \right\}$$

where $\alpha^k \in \hat{\alpha}(v^k)$ and $\lambda^k = \lambda\,(v^k,\,\alpha^k)$ is calculated according to (2.26). Let c^k be a solution.

3. Iterative variable updating: If $c^k = v^k$ the procedure is terminated.

Otherwise, $v^{k+1} = v^k + \epsilon_v\,(c^k - v^k)$ and the procedure is repeated.

In the following, the mappings representing these stages are defined. The overall algorithmic mapping then can be expressed as a composition of these mappings.

For $v \in \mathbb{C}$, if we define the sets :

$$\hat{\alpha}(v) = \left\{ \alpha \in \mathbb{R}^t : F(v, \alpha) = K_*(v) \right\} \tag{2.47}$$

and

$$\mathbb{A} = \bigcup_{v \in \mathbb{C}} \hat{\alpha}(v) \tag{2.48}$$

then a point-to-set mapping is given as :

$$\hat{\alpha} : \mathbb{C} \rightarrow \mathbb{A} \tag{2.49}$$

and stage (1) can be described as :

For a given $v^k \in \mathbb{C}$ find α^k such that $\alpha^k \in \hat{\alpha}(v^k)$.

Let us define another point-to-set mapping :

$$\hat{c} : \mathbb{C} \times \mathbb{A} \rightarrow \mathbb{C} \tag{2.50}$$

as

$$\hat{c}(v, \alpha) = \operatorname{Arg} \min_{c \in \mathbb{C}} \left\{ q(c, \alpha) - \lambda^T(v, \alpha)\, c \right\} \tag{2.51}$$

where $\alpha \in \hat{\alpha}(v)$ and the capital A of Arg indicates that the optimisation solutions are not required to be unique and

$$\lambda : \mathbb{C} \times \mathbb{A} \rightarrow \mathbb{R}^n \tag{2.52}$$

is determined by the relation (2.26). Then stage (2) is described as :

For a given $(v^k, \alpha^k) \in \mathbb{C} \times \mathbb{A}$, find c^k such that $c^k \in \hat{c}(v^k, \alpha^k)$.

We will allow the iterative gain ϵ_V to vary during the iterations, i.e.

$$v^{k+1} = v^k + \epsilon_V^k\,(c^k - v^k) \quad \text{with } 0 < \tau \leq \epsilon_V^k \leq B(v^k) \tag{2.53}$$

where B: $\mathbb{C} \rightarrow \mathbb{R}$ is an appropriately defined function.

Now let us define :

$$\hat{\omega} : \mathbb{C} \times (\mathbb{C} - \mathbb{C}) \rightarrow \mathbb{C} \tag{2.54}$$

with

$$\hat{\omega}\,(v\,,\,d) = \left\{\, v + \epsilon\, d \in \mathbb{C} : \tau \leq \epsilon \leq B\,(v) \,\right\} \tag{2.55}$$

where d = c - v

Notice that the appropriate choice of B(v) is of great importance for guaranteeing $v + \epsilon\, d \in \mathbb{C}$. Stage (3) can be stated as :

For a given v^k and c^k, find v^{k+1} such that $v^{k+1} \in \hat{\omega}\,(v^k\,,\,c^k - v^k)$. If $c^k = v^k$, then $v^{k+1} = v^k$ and the iterative procedure is terminated.

Next, the following compositions of $\hat{\alpha}$, $\hat{c}$ and $\hat{\omega}$ are given. The cartesian product of $\hat{\alpha}$ and the identity mapping I :

$$\hat{\delta} : \mathbb{C} \rightarrow \mathbb{C} \times \mathbb{A} \tag{2.56}$$

with

$$\hat{\delta}\,(v) = \left\{\, (v,\alpha) \in \mathbb{C} \times \mathbb{A} : \alpha \in \hat{\alpha}\,(v) \,\right\} \tag{2.57}$$

The composite mapping of $\hat{\delta}$ and $\hat{c}$:

$$\hat{c}\hat{\delta} : \mathbb{C} \rightarrow \mathbb{C} \tag{2.58}$$

Stages (1) and (2) together can be expressed as :

For a given $v^k \in \mathbb{C}$ find $c^k \in \hat{c}\hat{\delta}\ (v^k)$. The sum mapping of $\hat{c}\hat{\delta}$ and - I

$$\hat{c}\hat{\delta} - I : \mathbb{C} \rightarrow \mathbb{C} - \mathbb{C} \tag{2.59}$$

with

$$(\hat{c}\hat{\delta} - I)(v) = \left\{ d = c - v \in \mathbb{C} - \mathbb{C} : c \in \hat{c}\hat{\delta}\ (v) \right\} \tag{2.60}$$

The cartesian product of $\hat{c}\hat{\delta}$ - I and I :

$$\nu : \mathbb{C} \rightarrow \mathbb{C} \times (\mathbb{C} - \mathbb{C}) \tag{2.61}$$

with

$$\nu\ (v) = \left\{ (v\ , d) \in \mathbb{C} \times (\mathbb{C} - \mathbb{C}) : d \in (\hat{c}\hat{\delta} - I)\ (v) \right\} \tag{2.62}$$

Finally, the overall algorithmic mapping of the standard MTS method is defined as the composition of ν and $\hat{\omega}$, i.e.

$$\Phi = \hat{\omega}\ \nu \tag{2.63}$$

and the k^{th} iteration of the algorithm is expressed as :

For a given $v^k \in \mathbb{C}$ find $v^{k+1} \in \Phi\ (v^k)$.

In a more general sense, the set of all fixed points of Φ can be defined as the algorithmic solution set :

$$\Omega = \left\{ v \in \mathbb{C} : v \in \Phi\ (v) \right\} \tag{2.64}$$

If the solutions at stages (1) and (2) are unique and a fixed iterative gain ϵ_V is chosen the above mappings are all point-to-point mappings.

2.4.2b The Improved MTS Method

1. Model-based optimisation :

$$\min_{c \in \mathbb{C}} \left\{ q(c) - (\lambda^k)^T c \right\}$$

where $\lambda^k = \lambda(v^k)$ is calculated according to (2.29). Let c^k be a solution.

2. Iterative variable updating: If $c^k = v^k$ the procedure is terminated.

Otherwise, $v^{k+1} = v^k + \epsilon_V (c^k - v^k)$ and the procedure is repeated.

In the following, the mappings representing these stages are defined. The overall algorithmic mapping then can be expressed as a composition of these mappings.

Stage (1) can be described as :

Define the point-to-set mapping :

$$\hat{c} : \mathbb{C} \rightarrow \mathbb{C} \tag{2.65}$$

as

$$\hat{c}(v) = \operatorname{Arg} \min_{c \in \mathbb{C}} \left\{ q(c) - \lambda^T(v) c \right\} \tag{2.66}$$

and

$$\lambda : \mathbb{C} \rightarrow \mathbb{R}^n \tag{2.67}$$

is determined by the relation (2.29).

Then stage (1) is described as :

For a given $(v^k) \in \mathbb{C}$, find c^k such that $c^k \in \hat{c}(v^k)$.

We will allow the iterative gain ϵ_V to vary during the iterations, i.e.,

$$v^{k+1} = v^k + \epsilon_V^k (c^k - v^k) \quad \text{with } 0 < \tau \leq \epsilon_V^k \leq B(v^k) \tag{2.68}$$

where B: $\mathbb{C} \rightarrow \mathbb{R}$ is an appropriately defined function.

Now let us define :

$$\hat{\omega} : \mathbb{C} \times (\mathbb{C} - \mathbb{C}) \rightarrow \mathbb{C} \tag{2.69}$$

with

$$\hat{\omega} (v , d) = \left\{ v + \epsilon d \in \mathbb{C} : \tau \leq \epsilon \leq B (v) \right\} \tag{2.70}$$

where $d = c - v$

Notice that the appropriate choice of $B(v)$ is of great importance for guaranteeing $v + \epsilon d \in \mathbb{C}$. Stage (2) can be stated as :

For a given v^k and c^k find v^{k+1} such that $v^{k+1} \in \hat{\omega} (v^k , c^k - v^k)$. If $c^k = v^k$, then $v^{k+1} = v^k$ and the iterative procedure is terminated.

Stages (1) and (2) together can be expressed as :

For a given $v^k \in \mathbb{C}$ find $c^k \in \hat{c} (v^k)$. The sum mapping of $\hat{c}$ and - I

$$\hat{c} - I : \mathbb{C} \rightarrow \mathbb{C} - \mathbb{C} \tag{2.71}$$

with

$$(\hat{c} - I) (v) = \left\{ d = c - v \in \mathbb{C} - \mathbb{C} : c \in \hat{c} (v) \right\} \tag{2.72}$$

The cartesian product of $\hat{c}$ - I and I :

$$\nu : \mathbb{C} \rightarrow \mathbb{C} \times (\mathbb{C} - \mathbb{C}) \tag{2.73}$$

with

$$\nu (v) = \left\{ (v , d) \in \mathbb{C} \times (\mathbb{C} - \mathbb{C}) : d \in (\hat{c} - I) (v) \right\} \tag{2.74}$$

Finally, the overall algorithmic mapping of the improved MTS method is defined as the composition of ν and $\hat{\omega}$, i.e.

$$\Phi = \hat{\omega}\ \nu \tag{2.75}$$

and the k^{th} iteration of the algorithm is expressed as :

For a given $v^k \in \mathbb{C}$ find $v^{k+1} \in \Phi\ (v^k)$.

2.4.3 Convergence Analysis

The following theorems provide a set of sufficient conditions for the standard and improved MTS method to be globally convergent.

Theorem 2.3 (Convergence Theorem for The Standard MTS Method)

Assume :

(1) The set $\mathbb{C}$ is bounded and the set $\mathbb{A}$ is compact.

(2) The functions f_j , j = 1, ... , r are convex and continuous on $\mathbb{C}$.

(3) The function $q_*(\cdot)$ is Frèchet differentiable on $\mathbb{C}$ and its derivative $q_*'(\cdot)$ is Lipschitz continuous on $\mathbb{C}$ with constant $A_* > 0$, i.e.

$$\left|\left| q_*'(c+h) - q_*'(c) \right|\right| \leq A_* \left|\left| h \right|\right| , \text{ for all } c,\ c+h \in \mathbb{C}$$

where $\left|\left| h \right|\right|$ denotes the Euclidean norm of vector h.

(4) The function $K_*(\cdot)$ is continuous on $\mathbb{R}^n$ and continuously Frèchet differentiable on $\mathbb{C}$.

(5) The model mapping $F(\cdot\ ,\ \cdot)$ is continuous on $\mathbb{C} \times \mathbb{A}$ and $F(\ .\ ,\alpha)$ is continuously Frèchet differentiable on $\mathbb{C}$ for every $\alpha \in \mathbb{A}$.

(6) The process model $F(\cdot\ ,\ \cdot)$ is point-parametric on $\mathbb{C}$ i.e., the set $\hat{\alpha}(v)$ is not empty for all $v \in \mathbb{C}$ (Brdyś, 1983).

(7) The function $Q(\cdot\ ,\ \cdot)$ is continuously differentiable on $\mathbb{C} \times \mathbb{Y}$.

(8) For every $\alpha \in \mathbb{A}$, the function $q(.\ ,\ \alpha)$ is Frèchet differentiable on $\mathbb{C}$ and its derivative is uniformly monotone on $\mathbb{C}$ with constant $b\ (\alpha) \geq 0$, i.e.

$$\left[q_c'\ (c+h\ ,\ \alpha) - q_c'(c\ ,\ \alpha) \right]^T h \geq b\ (\ \alpha\ )\ \left|\left|\ h\ \right|\right|^2 \text{ for all c and c+h} \in \mathbb{C}$$

where $b : \mathbb{A} \rightarrow \mathbb{R}$ is upper semicontinuous on $\mathbb{A}$ and the following holds :

$$\inf_{\alpha \in \mathbb{A}} b\ (\alpha) > 0$$

(9) The point-to-set mapping $\hat{\alpha}$ is open on $\mathbb{C}$. Then,

(i) If τ and ϵ are chosen such that $0 < \tau \leq 1$ and $\epsilon > 0$ and

$$B(v) = \min \left\{ 1\ ,\ \frac{2}{A_*} \inf_{\alpha \in \hat{\alpha}(v)} b\ (\alpha) - \epsilon \right\} \tag{2.76}$$

$$\tau + \epsilon \leq \frac{2}{A_*} \inf_{\alpha \in \mathbb{A}} b\ (\alpha) \tag{2.77}$$

then the algorithmic mapping Φ is well-defined on $\mathbb{C}$ and closed outside the set Ω.

(ii) There is at least one cluster point of the sequence $\{v^k\}$ generated by the algorithm $v^{k+1} \in \Phi\ (v^k)$. Each cluster point belongs to Ω and, moreover, each point v^k satisfies the real process constraints and corresponding process performace values satisfy the condition :

$$q_*(v^{k+1}) < q_*(v^k) \text{ if } v^k \notin \Omega \quad \text{for } k = 1\ ,\ 2\ ...$$

The proof of this theorem is a modified version of the one given by Brdyś and Roberts (1987) and is given by Chen (1986).

An inspection of Theorem 2.3 shows that assumptions (1) and (2) are mild. The set $\mathbb{C}$ is often bounded in practical problems. If q_* is twice continuously differentiable on $\mathbb{C}$ assumption (3) is satisfied and can usually be taken for granted in practical situations. Assumptions (4) to (7) are standard and are usually fulfilled.

Sufficient conditions for assumption (9), openness of $\hat{\alpha}$, can be derived based on the general results reported in Brdyś and Ulanicki (1978). However, if an estimation result is unique for all $v \in \mathbb{C}$, then $\hat{\alpha}$ is a point-to-point mapping. Based on the previous assumptions it can be proved that $\hat{\alpha}$ is a closed point-to-point mapping (Chen, 1986). Hence, due to compactness of $\mathbb{A}$, $\hat{\alpha}$ is a continuous function.

The important assumption is assumption (8). It requires that the composition of the performance index and the process mathematical model is a uniformly convex function on the feasible control set, but it is usually satisfied when $Q(\cdot\ ,\ \cdot)$ is strictly convex (Chen, 1986; Lin, Wang and Roberts, 1989).

It is important to observe that no assumption requires that $q_*(\cdot)$ is convex and, therefore, the convergence theorem applies to non-linear processes and to the situation where the $q_*(\cdot)$ is not necessarily convex.

Theorem 2.4 (Convergence Theorem for The Improved MTS Method)

Assume :

(1) The set $\mathbb{C}$ is bounded.

(2) The functions f_j , $j = 1, \dots , r$ are convex and continuous on $\mathbb{C}$.

(3) The function $q_*(\cdot)$ is Fréchet differentiable on $\mathbb{C}$ and its derivative $q_*'(\cdot)$ is Lipschitz continuous on $\mathbb{C}$ with constant $A_* > 0$, i.e.

$$\left\| q_*'(c+h) - q_*'(c) \right\| \le A_* \left\| h \right\| , \text{ for all } c,\ c+h \in \mathbb{C}$$

where $\left\| h \right\|$ denotes the Euclidean norm of vector h.

(4) The function $K_*(\cdot)$ is continuous on $\mathbb{R}^n$ and continuously Fréchet differentiable on $\mathbb{C}$.

(5) The model mapping $F(\cdot)$ is continuous and continuously Fréchet differentiable on $\mathbb{C}$.

(6) The function $Q(\cdot\ ,\ \cdot)$ is continuously differentiable on $\mathbb{C} \times \mathbb{Y}$.

(7) The function $q(\cdot)$ is Frèchet differentiable on $\mathbb{C}$ and its derivative is uniformly monotone on $\mathbb{C}$ with constant $b > 0$, i.e.

$$\left[\, q_c'(c+h) - q_c'(c) \,\right]^T h \geq b \,\left|\left|\, h \,\right|\right|^2 \text{ for all } c \text{ and } c+h \in \mathbb{C}$$

Then,

(i) If τ and ϵ are chosen such that $0 < \tau \leq 1$ and $\epsilon > 0$ and

$$B(v) = \min \left\{ 1 \, , \frac{2b}{A_*} - \epsilon \right\} \tag{2.78}$$

$$\tau + \epsilon \leq \frac{2b}{A_*} \tag{2.79}$$

then the algorithmic mapping Φ is well-defined on $\mathbb{C}$ and closed outside the set Ω.

(ii) There is at least one cluster point of the sequence $\{v^k\}$ generated by the algorithm $v^{k+1} \in \Phi\ (v^k)$. Each cluster point belongs to Ω and, moreover, each point v^k satisfies the real process constraints and corresponding process performance values satisfy the condition :

$$q_*(v^{k+1}) < q_*(v^k) \text{ if } v^k \notin \Omega \quad \text{for } k = 1, 2 \ldots$$

The proof of this theorem is given by Lin, Han and Roberts (1989). An inspection of Theorem 2.4 shows that the assumptions made in the theorem are basically the same as those for Theorem 2.3. Therefore, the comments made for Theorem 2.3 apply also to this theorem.

2.5 STUDIES ON THE CONVERGENCE CONDITIONS FOR THE STANDARD MTS METHOD

In this section, the convergence conditions for the MTS method (assumptions of Theorem 2.3) are examined. To gain some insight into the above mentioned conditions we present a simple case of linear quadratic problems.

2.5.1 Convergence Analysis for Unconstrained Linear Quadratic Cases

In this case both model and reality are linear mappings and the performance index in ROP is quadratic (Amini and Roberts, 1988) i.e.

$$\text{Model} \qquad y = D\,c + \alpha = F\,(c\,,\,\alpha) \tag{2.80}$$

$$\text{Reality} \qquad y_* = D_* c + e_* = K_*(c) \tag{2.81}$$

$$\text{Performance index} \quad Q = \Big[\,y - d\,\Big]^T A \Big[\,y - d\,\Big] + c^T\,B\,c \tag{2.82}$$

Real optimum

By solving equations (2.82) and (2.81) simultaneously we obtain the real performance :

$$q_*(c) = \Big[\,D_* c + e_* - d\,\Big]^T A \Big[\,D_* c + e_* - d\,\Big] + c^T\,B\,c \tag{2.83}$$

Applying the optimality condition for ROP yields :

$$\frac{\partial q_*(c)}{\partial c} = 2\Big[\,Bc + D_*^T\,A\,(\,D_* c + e_* - d\,)\,\Big] = 0$$

$$c_* = \Big[\,B + D_*^T\,A\,D_*\,\Big]^{-1} D_*^T\,A \Big[\,d - e_*\,\Big] \tag{2.84}$$

where c_* is the real optimum of ROP.

Model-Based Optimisation

The model-based performance index is obtained from equations (2.82) and (2.80), i.e.

$$q\,(c) = \Big[\,D\,c + \alpha - d\,\Big]^T A \Big[\,D\,c + \alpha - d\,\Big] + c^T\,B\,c \tag{2.85}$$

According to the principle of the MTS method the modified model performance index then becomes :

$$q\ (c) - \lambda^T c = \Big[D\,c + \alpha - d \Big]^T A \Big[D\,c + \alpha - d \Big] + c^T\, B\,c - \lambda^T\, c \tag{2.86}$$

By minimising q(c) we obtain :

$$2 \Big[B\,c + D^T A\ (\ D\,c + \alpha - d\ ) \Big] - \lambda = 0$$

giving

$$\hat{c} = \Big[B + D^T\ A\ D \Big]^{-1} \Big[0.5\ \lambda + D^T\ A\ (\ d - \alpha\ ) \Big] \tag{2.87}$$

Modifier Calculation

The modifier in equation (2.86) is calculated according to :

$$\lambda = \Big[\frac{\partial^T F\ (\ v\ ,\ \alpha\ )}{\partial\ v} - \frac{\partial^T K_*(\ v\ )}{\partial\ v} \Big] \frac{\partial^T Q(v\ ,\ F(v\ ,\ \alpha))}{\partial\ y} \tag{2.88}$$

i.e.

$$\lambda = 2 \Big[D - D_* \Big]^T A \Big[y - d \Big] \tag{2.89}$$

Parameter Estimation

The parameters of the model are determined by solving :

$$F\ (\ v\ ,\ \alpha\ ) = K_*(\ v\ ) \tag{2.90}$$

where v is the current application of set-point c.

i.e.

$$D\ v + \alpha = D_* v + e_*$$

or

$$\alpha = \left[D_* - D \right] v + e_* \tag{2.91}$$

Updating Controller Set-Point and Deriving Algorithmic Mapping

The controller set-points are updated using the following formula :

$$v^{k+1} = v^k + \epsilon_v^k \left(\hat{c}^k - v^k \right) \tag{2.92}$$

Let $\epsilon_v^k = \epsilon_v$ throughout the iteration, then the algorithmic mapping is obtained by combining equations (2.87), (2.89), (2.91) and (2.92), i.e.

$$v^{k+1} = \left[I + \epsilon_v \left[\left[B + D^T AD \right]^{-1} \left[D^T AD - D_*^T AD_* \right] - I \right] \right] v^k$$

$$+ \epsilon_v \left[B + D^T AD \right]^{-1} D_*^T A \left[d - e_* \right] \tag{2.93}$$

Convergence Study

Using Ostrowski's theorem (Bertsekas, 1982) convergence will occur when

$$\sigma \left[I + \epsilon \tilde{A} \right] < 1$$

where $\sigma \left[\quad \right]$ is the spectral radius of $\left[\quad \right]$ and

$$\tilde{A} = \left[B + D^T AD \right]^{-1} \left[D^T AD - D_*^T AD_* \right] - I \tag{2.94}$$

which is symmetric

Lipschitz Condition

The applicability conditions of the MTS method provided in Theorem 2.3 require that the derivative of the real performance is Lipschitz continuous with constant $A_* > 0$, i.e.

$$\left\| q_*'(c+h) - q_*'(c) \right\| \leq A_* \left\| h \right\| \qquad \text{for all } c, c+h \in \mathbb{C}.$$

Hence

$$A_* \geq \frac{\left\| q_*'(c+h) - q_*'(c) \right\|}{\left\| h \right\|}$$

Now, from equation (2.83)

$$q_*'(c) = 2\left[B + D_*^T A D_* \right] c + 2\, D_* A \left[e_* - d \right]$$

and

$$q_*'(c+h) = 2\left[B + D_*^T A D_* \right](c+h) + 2 D_* A \left[e_* - d \right]$$

Hence,

$$q_*'(c+h) - q_*'(c) = 2\left[B + D_*^T A D_* \right] h$$

and

$$A_* \geq 2 \left\| \left[B + D_*^T A D_* \right] \right\| \tag{2.95}$$

Hence, we use

$$A_* = 2\, \lambda_{max} \left[B + D_*^T A D_* \right] \tag{2.96}$$

where $\lambda_{max} \left[B + D_*^T A D_* \right]$ = max eigenvalue of $\left[B + D_*^T A D_* \right]$ (Bertsekas, 1982) .

Monotone Condition

In Theorem 2.3 it is assumed that the derivative of the model performance is uniformly monotone with constant $b(\alpha) \geq 0$ according to :

$$\left[q_c'(c+h , \alpha) - q_c'(c) \right]^T h \geq b(\alpha) \left\| h \right\|^2 \quad \text{for all } c, c+h \in \mathbb{C}$$

Using equation (2.85) gives

$$q_c'(c , \alpha) = 2 \left[B + D^T AD \right] c + 2 D^T A \left[\alpha - d \right]$$

and

$$q_c'(c+h , \alpha) = 2 \left[B + D^T AD \right] (c+h) + 2 D^T A \left[\alpha - d \right]$$

Hence,

$$q_c'(c+h , \alpha) - q_c'(c , \alpha) = 2 \left[B + D^T A D \right] h$$

and

$$b(\alpha) \leq \frac{2 h^T \left[B + D^T A D \right] h}{h^T h} \tag{2.97}$$

But, according to a well-known result (Bertsekas, 1982);

$$\lambda_{min} \left[2 (B + D^T AD) \right] \leq \frac{2 h^T \left[B + D^T A D \right] h}{h^T h} \leq \lambda_{max} \left[2 (B + D^T A D) \right]$$

where $\lambda_{min} ([\;\;])$ = minimum eigenvalue of $[\;\;]$.

Hence,

$$b(\alpha) \leq \lambda_{min} \left[2 (B + D^T A D) \right] \tag{2.98}$$

Stability

From Theorem 2.3 the maximum iterative gain for the algorithm convergence is :

$$\epsilon_{max} = \frac{2\ b(\alpha)}{A_*} \tag{2.99}$$

Hence, from equations (2.96) and (2.98) :

$$\max.\ \epsilon_v = \frac{2\ \lambda\min\left[\ B + D^T\ A\ D\ \right]}{\lambda_{max}\left[\ B + D_*^T\ A\ D_*\ \right]} \tag{2.100}$$

It is important to note that the above estimation of the upper bound for gain selection is in general a conservative one (Chen, 1986; Amini and Roberts, 1988). The reason for this is as follows; the global convergence theorem states that convergence is achieved if ϵ_v is not greater than B(v), however, having convergence does not necessarily mean that ϵ_v must be less than B(v). Hence $\epsilon_v < B(v)$ is only a sufficient condition. The simulation results obtained using a MATLAB program support the above arguments for linear quadratic cases. Convergence conditions of Theorem 2.3 can also be investigated through simulation by referring to the two examples, one of linear and one of non-linear nature, presented by Amini and Roberts (1988).

2.6 OPTIMISING CONTROL PROBLEMS WITH OUTPUT DEPENDENT CONSTRAINTS

In previous sections it was assumed, for the sake of the simplicity of the algorithm development, that the process constraints are independent of the process outputs. It is, however, important to extend the ISOPE method so that it can cope with output dependent constraints as such cases are often found in the industrial application of optimising control.

The earliest attempt to extend the ISOPE method so that it can deal with output dependent constraints was to augment the modifier to take into account the model-reality differences in constraints (Brdyś, Chen and Roberts, 1986). The major disadvantage of this technique is that the feasibility of the model-based optimisation problem is not ensured during the course of iteration (Lin, Chen and Roberts, 1988) and this will cause a great deal of difficulty in the on-line implementation of the algorithm. To get around this problem, a New Modifier method (NM) is proposed in which the model-reality differences in the constraints are compensated for by introducing a new modifier into the model-based optimisation problem (Lin, Chen and Roberts, 1988; Lin, Hendawy and Roberts, 1988a). The major advantage of this method is that it preserves the feasibility of the model-based optimisation problem during the course of iteration. It is, however, found through simulation that the convergence of the NM method is rather slow mainly due to the fact that the convergence features of the new modifier are effectively the same as those of the price coordination mechanism in the hierarchical optimisation problems (Lin, Kambhampati and Roberts, 1989). A more efficient approach to problems with output dependent constraints is the Penalty Relaxation (PR) technique. In this technique, the main features of the standard ISOPE method are retained and it has been found that it is a highly efficient technique for solving such problems (Lin, Kambhampati and Roberts, 1989; Lin, Li, Wan and Roberts, 1989).

In this section, the above mentioned New Modifier (NM) method and the Penalty Relaxation (PR) technique are described and thoroughly analysed. A review of these two methods based on comparative studies is also provided.

2.6.1 The New Modifier Method

The main idea of the New Modifier (NM) method is to introduce a new modifier into the model-based optimisation problem MMOP and the model-reality differences in the constraints are compensated for by modifying the performance index of MMOP so that the real optimum is obtained towards the end of iterations.

In this section, the NM method is derived directly rather than using the standard Lagrangian approach (Lin, Chen and Roberts, 1988). Optimality and convergence conditions are also studied in this section.

2.6.1a Algorithm Description

The following definition is needed to clarify the algorithmic description.

Definition 2.2

Define :

$g_*: \mathbb{R}^n \times \mathbb{R}^m \rightarrow \mathbb{R}^p$ with

$$g_* (v) = G (v , K_*(v)) \tag{2.101}$$

and

$q_*: \mathbb{R}^n \times \mathbb{R}^m \rightarrow \mathbb{R}$ with

$$q_* (v) = Q (v , K_*(v)) \tag{2.102}$$

Starting from any initial controller set-point $v^0 \in \mathbb{C}$ and initial guess of ξ^0, the k^{th} iteration is described as follows :

Step 1 Apply v^k to the real process and obtain the corresponding steady state measurements $K_*(v^k)$. Determine $\hat{\alpha}^k = \hat{\alpha}(v^k)$ by solving :

$$F(v^k , \alpha) = K_*(v^k) \tag{2.103}$$

Perform additional perturbations about v^k and measure the corresponding process outputs to compute a finite difference approximation of $K'_*(v^k)$, where :

$$K'_*(v^k) = \frac{\partial^T K_*(v^k)}{\partial v} \tag{2.104}$$

Finally, calculate the modifiers $\lambda^k = \lambda(v^k , \hat{\alpha}^k , \xi^k)$ according to :

$$\lambda(v , \alpha , \xi) = \left[\frac{\partial^T F(v,\alpha)}{\partial v} - \frac{\partial^T K_*(v)}{\partial v} \right] \left[\frac{\partial^T Q(v , y)}{\partial y} + \frac{\partial^T G(v , y)}{\partial y} \xi \right] \tag{2.105}$$

where $y = F(v , \alpha)$ is a process model.

Step 2 With given $\hat{\alpha}^k$ and λ^k, solve the following model-based optimisation problem

$$\text{(MMOP)} \quad \min_{c} \left\{ q(c, \hat{\alpha}^k) - (\lambda^k)^T c + \left[\frac{\partial g(v^k, \hat{\alpha}^k)}{\partial v} \xi^k \right]^T c \right\} \quad \text{s.t. } c \in \mathbb{C}$$

The solution is denoted as $\hat{c}(v^k, \hat{\alpha}^k, \xi^k)$, or $\hat{c}^k$ for short.

The vectors v^k and ξ^k are updated according to the formulae :

$$v^{k+1} = v^k + \epsilon_v (\hat{c}^k - v^k) \tag{2.106}$$

$$\xi^{k+1} = \xi^k + \epsilon_\xi^k (v^k - \xi^k) \left[g_*(v^k) + g_*'(v^k)(\hat{c}^k - v^k) \right] \tag{2.107}$$

where $g_*'(v^k)$ is the derivative of $g_*(v^k)$ as defined in (2.104) for $K_*'(v^k)$. In order to clarify the strategy for choosing the iterative gain factor ϵ_ξ^k a projection operator $P(v, \xi)$ is introduced satisfying :

$$\left[P(v, \xi) \{ g_*(v) + g_*'(v)(\hat{c}(v, \hat{\alpha}, \xi) - v) \} \right]_i = 0 \qquad \forall i \in N^1(v, \xi)$$

$$\left[P(v, \xi) \{ g_*(v) + g_*'(v)(\hat{c}(v, \hat{\alpha}, \xi) - v) \} \right]_i = \left[g_*(v) + g_*'(v)(\hat{c}(v, \hat{\alpha}, \xi) - v) \right]_i \qquad \forall i \notin N^1(v, \xi) \tag{2.108}$$

where $\hat{\alpha}$ is the short form of $\hat{\alpha}(v)$ and

$$N^1(v, \xi) = \left\{ i : \xi_i = 0 \text{ and } \left[g_*(v) + g_*'(v)(\hat{c}(v, \hat{\alpha}, \xi) - v) \right]_i \leq 0 \right\} \tag{2.109}$$

where $[\cdot]_i$ denotes the i^{th} element of the vector. The gain parameter ϵ_ξ^k is determined as follows :

Let

$$N^2(v\ ,\ \xi) = \left\{ i\!: \xi_i > 0 \text{ and } \left[g_*(v) + g_*'(v)\,(\hat{c}(v\ ,\ \hat{\alpha}\ ,\ \xi) - v) \right]_i < 0 \right\} \tag{2.110}$$

and

$$\bar{\epsilon}_\xi^k = \min_{i \in N^2(v^k\ ,\ \xi^k)} \left\{ \frac{-\ \xi_i^k}{\left[g_*(v^k) + g_*'(v^k)\,(\hat{c}^k - v^k) \right]_i} \right\} \tag{2.111}$$

then

$$\epsilon_\xi^k = \begin{cases} \min\{\ \epsilon_\xi\ ,\ \bar{\epsilon}_\xi^k\ \} & \text{if } N^2(v^k\ ,\ \xi^k) \text{ is not empty} \\ \epsilon_\xi & \text{if } N^2(v^k\ ,\ \xi^k) \text{ is empty} \end{cases} \tag{2.112}$$

In the above, ϵ_v and ϵ_ξ are prescribed positive scalars. The purpose of (2.100) to (2.112) is to exclude the possibility of ξ_i^{k+1} becoming negative. The iteration is terminated when $\hat{c}^k = v^k$ and $\xi^{k+1} = \xi^k$. Figure 2.4 shows the information structure for the NM method.

Inspecting the model-based optimisation problem, it is seen that $g(c\ ,\ \alpha)$ is excluded from the constraint set of MMOP. On the other hand, its influence on the optimisation is taken into account by the term $\left[\frac{\partial g(v^k\ ,\ \hat{\alpha}^k)}{\partial v}\ \xi^k \right]^T c$ added to the objective function. This has an effect of simplifying the optimisation calculation, especially if there are no $f(c)$ constraints. In this case, MMOP becomes an unconstrained problem. Unlike the previous algorithm given by Brdyś, Chen and Roberts (1986), the constraints of the model-based optimisation problem in the present algorithm do not change during the course of iteration. Because of this, the possibility of having no feasible solution in the model-based optimisation is ruled out. The technique employed to iterate ξ originates from the active set method (Luenberger, 1984). Note that there is no new requirement because $K_*(v^*)$ and $K_*'(v^k)$ needed to update ξ have already been provided in Step 1.

2.6.1b Optimality and Convergence

It is shown in this section that the NM method is optimal in the sense that every algorithmic solution point satisfies the Kuhn-Tucker conditions for ROP (Lin, Chen and Roberts, 1988). The following definitions are introduced to clarify the proof of the optimality theorem :

$$\Omega = \Big\{ (v , \xi) \text{: } v \in \mathbb{C}, \; v = \hat{c}(v , \hat{\alpha} , \xi); \text{ and } \xi_i \geq 0 ,$$

$$\Big[P(v , \xi) \; (g_*(v) + g_*'(v) (\hat{c} - v)) \Big]_i = 0, \;\; i = 1, \dots , P \Big\} \tag{2.113}$$

$$\Omega_1 = \Big\{ v\text{: } \exists \; \xi \text{ such that } (v , \xi) \in \Omega \Big\} \tag{2.114}$$

$$\Omega_2 = \Big\{ \xi\text{: } \exists \; v \text{ such that } (v , \xi) \in \Omega \Big\} \tag{2.115}$$

and denote Ω_1^* as the optimal solution set of ROP.

It can be seen that Ω is the algorithmic solution set. In the following η is denoted as the Lagrange multiplier vector associated with the constraints f.

Theorem 2.5

Let the assumptions for ROP and EOP 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 necessary optimality conditions of ROP.

Proof

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 :

$$\frac{\partial^T q (\hat{c} , \hat{\alpha})}{\partial c} - \lambda(v , \hat{\alpha} , \xi) + \frac{\partial g (v , \hat{\alpha})}{\partial c} \; \xi + \frac{\partial f(\hat{c})}{\partial c} \; \eta = 0 \tag{2.116}$$

$$\eta_i f_i(\hat{c}) = 0, \;\; \eta \geq 0, \;\; \text{and} \;\; f_i(\hat{c}) \leq 0, \;\; i = 1, \dots , r \tag{2.117}$$

Since $F(v , \hat{\alpha}) = K_*(v)$, (2.105) can be rewritten as :

$$\lambda(v , \hat{\alpha} , \xi) = \frac{\partial q (v , \hat{\alpha})}{\partial c} - \frac{\partial q_*(v)}{\partial c} + \frac{\partial g (v , \hat{\alpha})}{\partial c} \; \xi - \frac{\partial g_*(v)}{\partial c} \; \xi \tag{2.118}$$

Substituting (2.118) into (2.116), and using $\hat{c} = v$ in both (2.116) and (2.117), yields :

$$\frac{\partial q_*(v)}{\partial v} + \frac{\partial g_*(v)}{\partial v}\,\xi + \frac{\partial f(v)}{\partial v}\,\eta = 0 \tag{2.119}$$

$$\eta_i f_i(v) = 0, \;\; \eta_i \geq 0, \;\; \text{and} \;\; f_i(v) \leq 0, \;\; i = 1, \dots, r \tag{2.120}$$

According to the definition of Ω, ξ and v satisfy

$$\xi_i \left[g_*(v) \right]_i = 0, \;\; \xi_i \geq 0, \text{ and} \left[g_*(v) \right]_i \leq 0, \; i = 1, \dots, p \tag{2.121}$$

(2.119) to (2.121) are precisely the Kuhn-Tucker conditions of ROP at v.

QED

It can be proved, subject to certain conditions, that the presented NM method is locally convergent (Lin, Chen and Roberts, 1988). As seen from the proof in the given reference, convergence conditions for the NM method are nearly the same as those for the single loop ISOPE method for hierarchical optimising control given by Brdyś and Roberts (1985). This means that the NM method requires more strict conditions for its convergence than the standard centralised ISOPE method. Simulation results also show that in the NM method the convergence of the Kuhn-Tucker multiplier ξ^k is slower than that of the controller set-point v^k (Lin, Kambhampati and Roberts, 1989; Lin, Li, Wan and Roberts, 1989). In fact, it is not difficult to observe in the algorithmic description for the NM method that the updating innovation for ξ^k is effectively the gradient of the dual problem and, therefore, the convergence of ξ^k should be of the same nature as that of the price coordination mechanism in hierarchical optimisation algorithms.

2.6.2 Penalty Relaxation Technique

In order to develop a more efficient algorithm, which can be used for solving optimising control problems with output dependent constraints, than the NM method it is necessary to devise a technique which can cope with the output dependent constraints within the structure of the iterative mechanism in the ISOPE method. This is precisely the motivation for the introduction of the Penalty Relaxation (PR) technique.

In the PR technique, the output dependent constraints are linearised at the given controller set-point in each iteration to form the model constraints in the model-based optimisation problem MMOP. A set of relaxation variables are introduced to relax the model constraints so that the feasibility of the model-based optimisation problem MMOP is ensured during the course of iteration. A penalty term of the relaxation variables is so constructed that by adding it to performance index in MMOP, the feasibility of the original output dependent constraints is preserved towards the end of iteration (Lin, Kambhampati and Roberts, 1989; Lin, Li, Wan and Roberts, 1989).

2.6.2a Algorithm Description

The PR technique given in this section fits in both the standard MTS method (Roberts, 1979; Brdyś and Roberts, 1987) and the improved version of MTS (Lin, Han, Roberts and Wan, 1989), which were introduced in Sections 2.3.2 and 2.3.3. In the following, the algorithm description is dedicated to the combination of the PR technique and the improved version of MTS.

For any initial v^0, the k^{th} iteration of the proposed algorithm is described as follows :

Step 1 Apply v^k to the real process and measure its output $K_*(v^k)$, and hence $q_*(v^k)$ and $g_*(v^k)$. Perform additional perturbations about v^k to compute $q'_*(v^k)$ and $g'_*(v^k)$ using the finite difference approximation method. Then, calculate the modifier according to the following equation (Lin, Kambhampati and Roberts, 1989; Lin, Han, Roberts and Wan, 1989)

$$\lambda^k = \lambda(v^k) = q'(v^k) - q'_*(v^k) \tag{2.122}$$

where

$$q(v^k) = Q(v , F(v)) \tag{2.123}$$

and

$$F: \mathbb{R}^n \rightarrow \mathbb{R}^t, \quad \text{with} \quad y = F(v)$$

is a properly chosen model.

Step 2 For given λ^k, solve the following model-based optimisation problem

$$\min_{c} \left\{ q(c) - (\lambda^k)^T c + \rho \left|\left| w \right|\right|^2 \right\}$$

(MMOP)

s.t.

$$g_*(v^k) + g'_*(v^k)\,(c - v) + w \leq 0$$

$$c \in \mathbb{C}$$

where $\rho > 0$ is a penalty coefficient which is normally assumed to be large, Denote $\hat{c}(v^k)$ and $\hat{w}(v^k)$ as the optimum of MMOP, or simply as $\hat{c}^k$ and $\hat{w}^k$ for short.

The controller set-point v^k is updated using the relaxation technique described in Section 2.3.2.

$$v^{k+1} = v^k + \epsilon_v^k\,(c^k - v^k) \tag{2.124}$$

The iteration is terminated when $v^{k+1} = v^k$.

It is observed from the above description that unlike the NM method (Lin, Chen and Roberts, 1988), in the PR method the output dependent constraints are linearised at each given set-point and no iteration on the Kuhn-Tucker multipliers are needed. As a result, the convergence features of the standard MTS method are preserved. Finally, since the PR technique is combined with the improved MTS (Lin, Han, Roberts and Wan, 1989), the matching of process outputs and model outputs are not required. This makes the model selection very flexible. The information structure for the penalty relaxation method is shown in Figure 2.5.

2.6.2b Optimality and Convergence

In this section, optimality of the PR method is investigated. It can be proved that subject to mild conditions, it is optimal and globally convergent (Lin, Kambhampati and Roberts, 1989).

It is well known that under mild conditions the real optimisation problem ROP is equivalent to the following :

$$\min_{c} \left\{ q_*(c) + \rho \left|\left| w \right|\right|^2 \right\}$$

(EOP1)

s.t.

$$g_*(c) + w \leq 0$$

$$c \in \mathbb{C}$$

in the sense that the optimum solution set of EOP1 can be made arbitrarily close to that of ROP provided that the penalty coefficient ρ is sufficiently large (Luenberger, 1984), where $q_*(\cdot)$ and $g_*(\cdot)$ are defined in (2.102) and (2.101), respectively. Therefore, it is sufficient to use EOP1 for investigation of the optimality for the PR method. Define :

$$\Omega = \left\{ v\text{: } v = \hat{c}(v) \right\}$$

where $\hat{c}(v)$ is the optimum solution of the model-based optimisation problem MMOP.

Denote Ω^* as the optimal solution set of EOP1 and assume that the constraint set $\mathbb{C}$ is defined in equation (2.5). With the above preliminary definitions we are ready to prove the optimality of the PR method.

Theorem 2.6

Let Assumption 2.1 be satisfied and assume that every $v \in \Omega$ is a regular point of the constraints $g_*(v) \leq 0$ and $f(v) \leq 0$. Then, for every $v \in \Omega$, there exists w, ξ and η such that (v , w , ξ , η) satisfies the Kuhn-Tucker conditions of EOP1, where ξ and η are the Kuhn-Tucker multipliers for the constraints $g_*(v) \leq 0$ and $f(v) \leq 0$, respectively.

Proof

For every $v \in \Omega$, $(\hat{c}(v) , \hat{w}(v))$ is the optimum solution of MMOP. Since $v \in \Omega$, we have $\hat{c}(v) = v$. Hence, for any given $v \in \Omega$, $\hat{c}(v)$ is a regular point of the the constraints $g_*(v) + g_*'(v)\ (c - v) \leq 0$ and $f(v) \leq 0$, and $(\hat{c}(v) , \hat{w}(v))$ is a regular point of the constraints $g_*(v) + g_*'(v)\ (c - v) + w \leq 0$ and $f(v) \leq 0$. Therefore, there exists ξ and η such that :

$$q'(c) - \lambda(v) + g'_*(v)\,\xi + f'(c)\,\eta = 0 \text{ and } \xi + 2\rho w = 0 \tag{2.125}$$

$$\xi_i \Big[g_*(v) + g'_*(v)\,(c - v) + w \Big]_i = 0,\ i = 1, \ldots, r \tag{2.126}$$

with

$$\xi_i \geq 0, \Big[g_*(v) + g'_*(v)\,(c - v) + w \Big]_i = 0,\ i = 1, \ldots, r \tag{2.127}$$

$$\eta_i f_i(c) = 0 \text{ with } \eta_i \geq 0,\ f_i(c) \leq 0,\ \ i = 1, \ldots, s \tag{2.128}$$

where $\hat{c}$ and $\hat{w}$ are the short forms for $\hat{c}(v)$ and $\hat{w}(v)$, respectively.

Substituting (2.122) into (2.125) and the equation $\hat{c}(v) = c$ into (2.126) — (2.127) yields

$$q'_*(c) + g'_*(v)\,\xi + f'(v)\,\eta = 0 \text{ and } \xi + 2\rho w = 0 \tag{2.129}$$

$$\xi_i \Big[g_*(v) + w \Big]_i = 0, \text{ with } \xi_i \geq 0,\ \Big[g_*(v) + w \Big]_i \leq 0, \quad i = 1, \ldots, r \tag{2.130}$$

$$\eta_i f_i(v) = 0 \text{ with } \eta_i \geq 0,\ f_i(v) \leq 0,\ \ i = 1, \ldots, s \tag{2.131}$$

Equations (2.129) — (2.131) are precisely the Kuhn-Tucker conditions of EOP1.

QED

It has been proved that subject to nearly the same conditions as those of the standard MTS method the PR method is globally convergent (Lin, Kambhampati and Roberts, 1989). It is also observed from the applicability conditions, given by Lin, Kambhampati and Roberts (1989), for the PR method that the penalty term introduced into the model-based optimisation problem MMOP has an effect of convexifying the real optimisation problem ROP. Compared with the NM method the applicability conditions of the PR method are much easier to satisfy and the convergence of the iterative mechanism is much more efficient.

2.7 SUMMARY

In this chapter basic concepts of on-line process optimisation are discussed. Based on these concepts a general mathematical formulation for centralised optimising control problems is derived. Optimum solution approaches to such problems are presented. Emphasis has been given to the Modified Two-Step (MTS) method which is designed to produce the true optimum regardless of model-reality differences. Optimality and convergence of the MTS method is thoroughly investigated. Techniques for coping with output dependent constraints, such as the New Modifier (NM) and Penalty Relaxation (PR) methods, are introduced and integrated with the MTS method to cover more general cases in optimising control. The ideas behind the presented centralised optimising control algorithms are further extended to hierarchical optimisation in the next chapter.

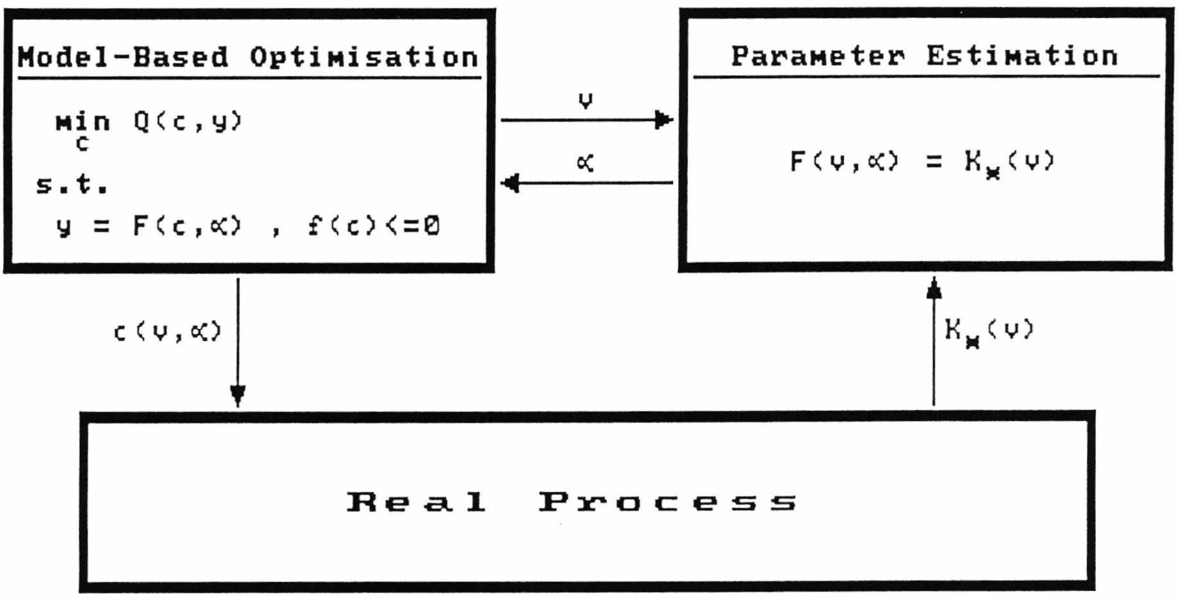

Figure 2.1 The Standard Two-Step Method

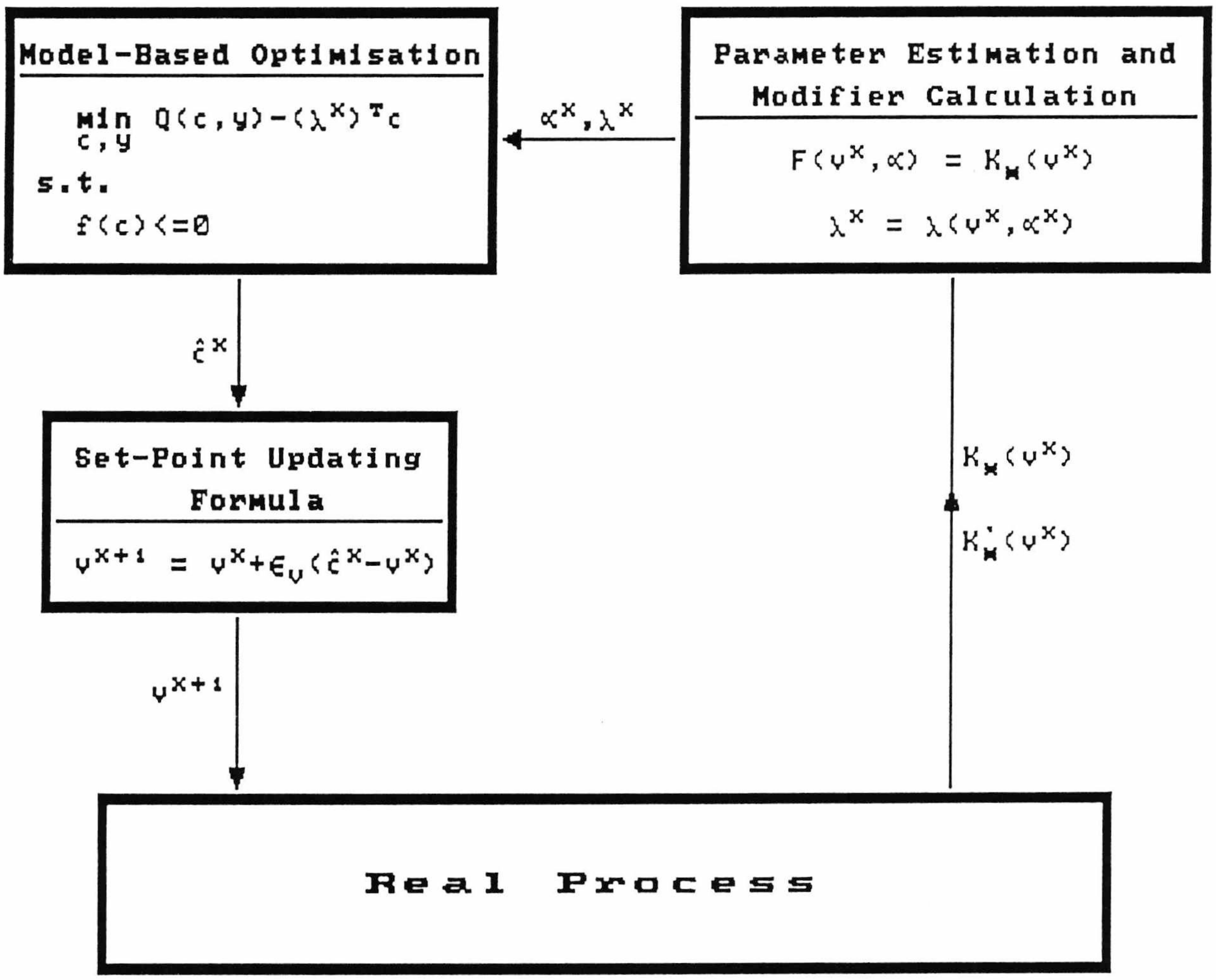

Figure 2.2 The Modified Two-Step Method

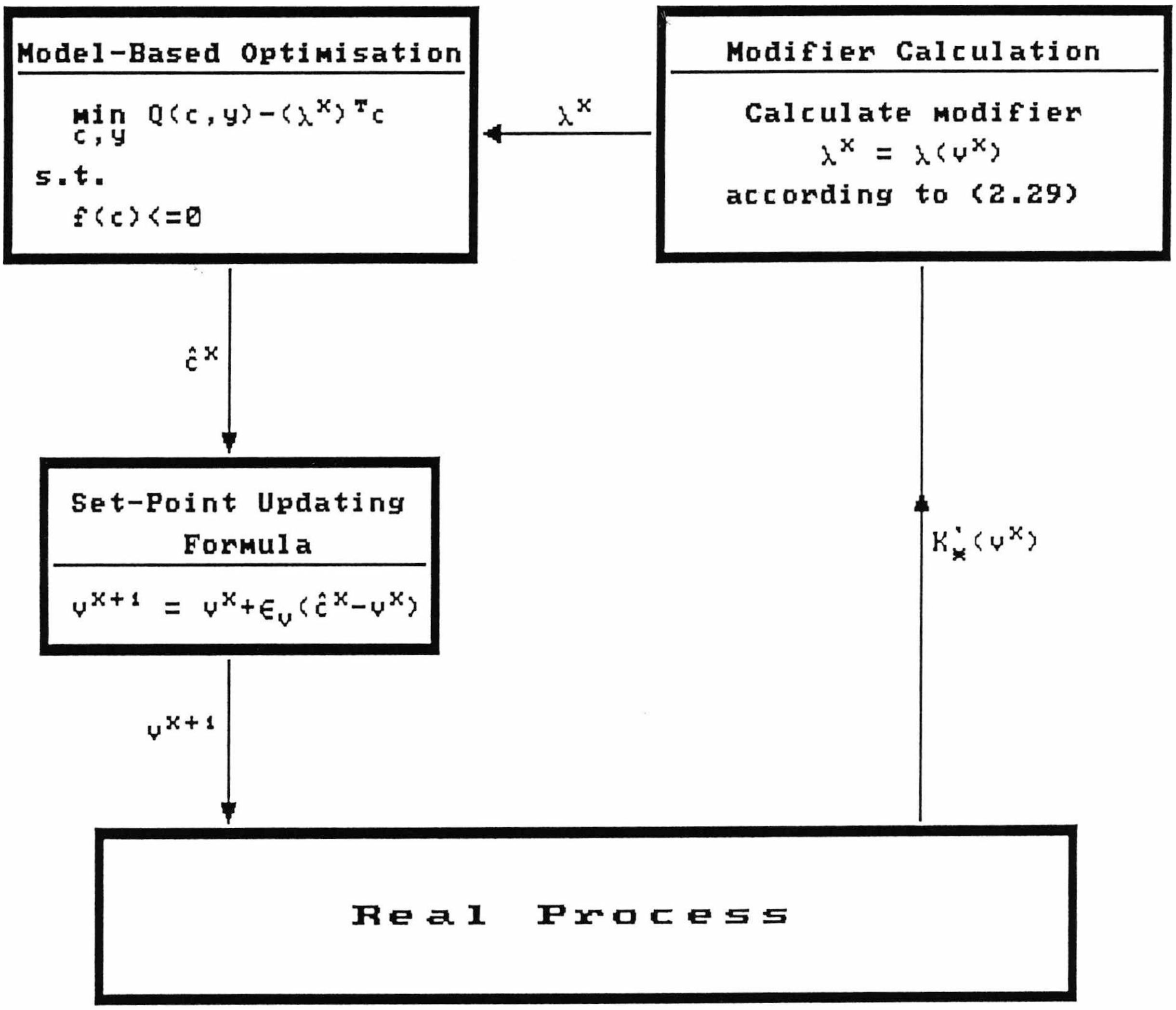

Figure 2.3 The Improved Modified Two-Step Method

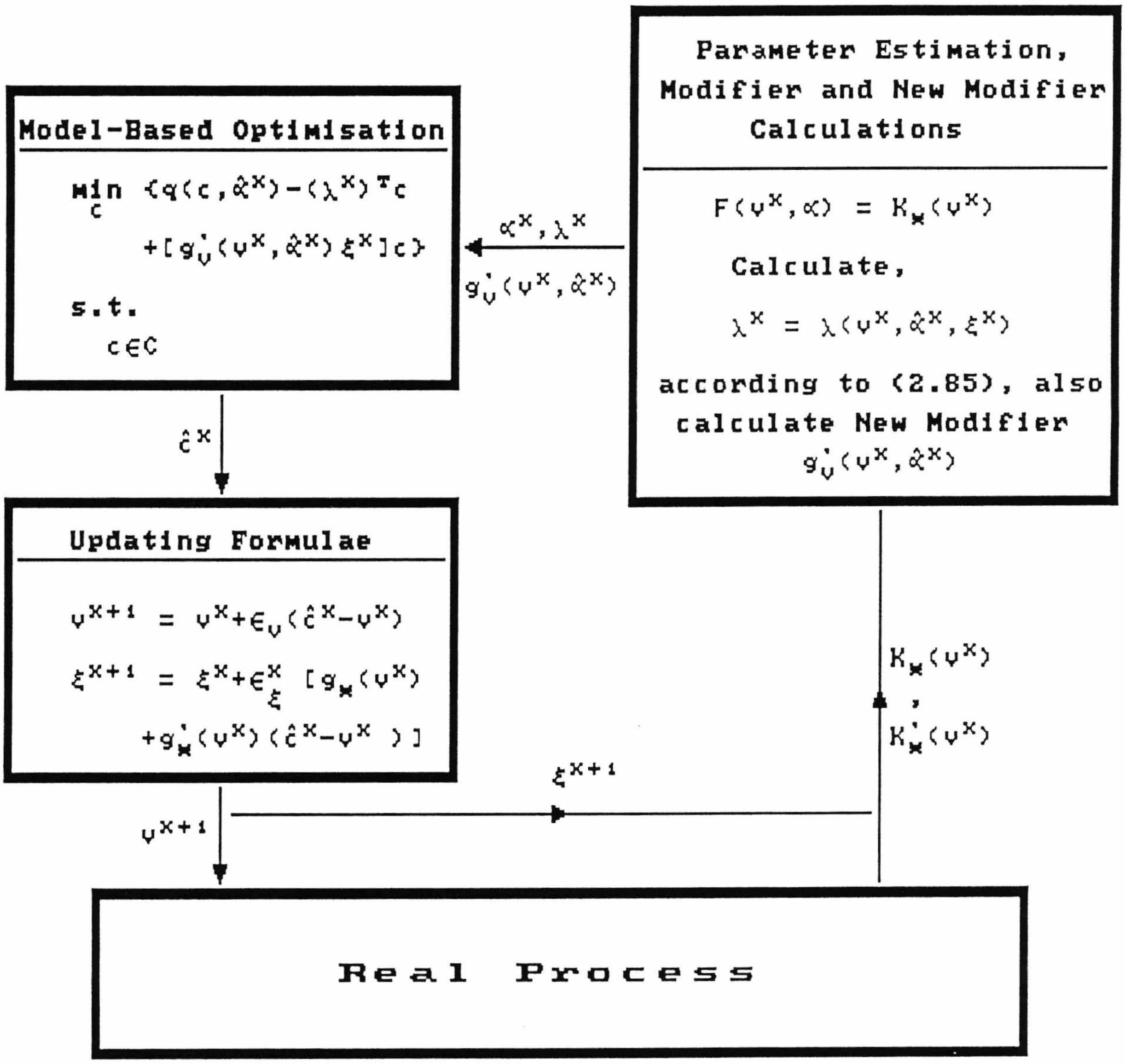

Figure 2.4 The New Modifier Method

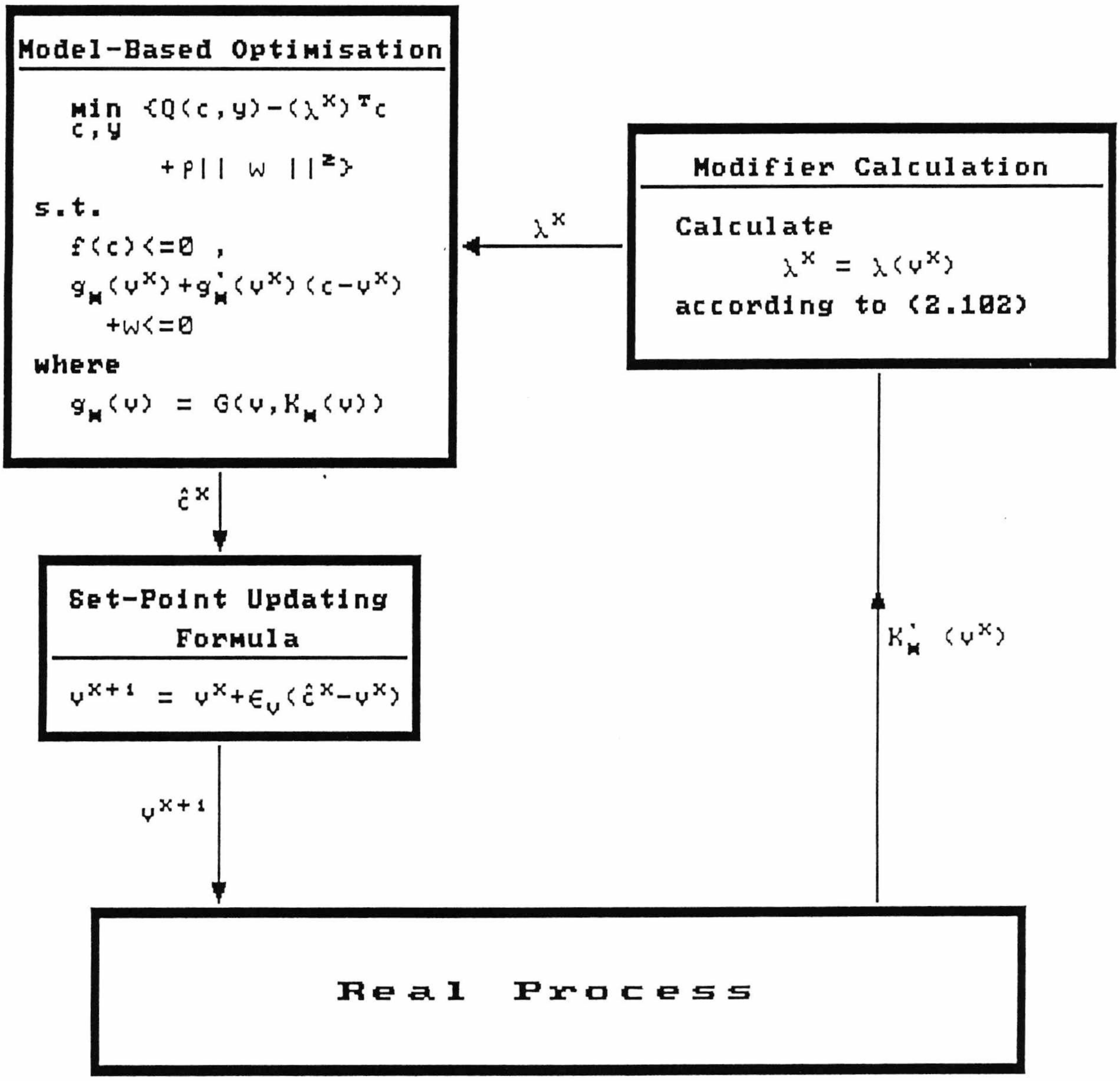

Figure 2.5 The Penalty Relaxation Technique

CHAPTER 3

HIERARCHICAL ALGORITHMS FOR INTEGRATED SYSTEM OPTIMISATION AND PARAMETER ESTIMATION (ISOPE)

3.1 INTRODUCTION TO HIERARCHICAL OPTIMISING CONTROL

A natural way to view a large scale industrial process or plant, which is often too complicated to comprehend in its entirety, is to consider it as a collection of interconnected subprocesses. By exploiting the structure of such a large scale process as an interconnected assembly of sub-processes it is possible to decompose the overall problem of controlling a complex industrial process into interlinked sub-problems of manageable size. Each sub-problem can be solved independently of the other sub-problems, with the interconnections being accounted for by some form of coordination procedure. Such an approach leads to hierarchical optimising control structures.

Generally, a hierarchical control system is structured by arranging a number of decision making units in a priority order where, at each level, some units may operate in parallel giving rise to a pyramid structure. In the practice of on-line process optimisation, this structure is identified by the natural hierarchy of the process.

A large scale industrial process usually consists of several interconnected sub-processes under regulatory and steady state optimising supervisory control. Each sub-process has its own local regulatory and optimising decision units with coordination often required at a second level within the supervisory layer to ensure that overall system objectives, interconnections and constraints are satisfied. The local optimising decision units compute the set-point values to maximise their local performance subject to local model equations, constraints and estimates of disturbances. Coordination is performed at a higher level unit which transmits intervention inputs (coordination variables) to all local decision units, calculated on the basis of previous optimisation results.

In this chapter important issues in on-line hierarchical optimising control such as coordination method, iterative strategy, feedback configuration and information structure are thoroughly discussed. Single and double loop ISOPE algorithms are presented and a variable augmentation technique is introduced to assist the convergence of the algorithm.

3.2 HIERARCHICAL FORMULATION FOR STEADY STATE OPTIMISING CONTROL PROBLEM

As mentioned in Section 3.1, the purpose of introducing hierarchical optimising control is to cater for situations where the production process is composed of a number of interconnected sub-processes and, therefore, the formulation for hierarchical optimising control is rather different from that for centralised cases.

Consider a finite-dimensional control system, composed of N interconnected subsystems whose input-output steady state behaviour is described by the mappings

$$F_{*i} : \mathbb{C}_i \times \mathbb{U}_i \rightarrow \mathbb{Y}_i \qquad i=1, ..., N$$

That is

$$y_i = F_{*i}(c_i, u_i) \qquad i=1, ..., N \tag{3.1}$$

where $\mathbb{C}_i$, $\mathbb{U}_i$ and $\mathbb{Y}_i$ are finite dimensional spaces and $c_i \in \mathbb{C}_i$, $u_i \in \mathbb{U}_i$ and $y_i \in \mathbb{Y}_i$ are the i^{th} subsytem's control, input and output vectors, respectively.

The structure of the interconnections of the subsystems is given by

$$u_i = H_i\, y = \sum_{j=1}^{N} H_{ij}\, y_j \qquad i=1, ..., N \tag{3.2}$$

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

We make the usual assumptions that (3.1) and (3.2) are uniquely solvable with respect to the controls, so that the input-output mapping can be expressed as

$$y = K_*(c) \tag{3.3}$$

where

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

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

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

and $K_* : \mathbb{C} \rightarrow \mathbb{Y}$.

The local constraint set takes the form

$$\mathbb{CU}_i = \left\{ (c_i , u_i) \in \mathbb{C}_i \times \mathbb{U}_i \mid f_i(c_i , u_i) \leq 0 \right\} \qquad i=1, \ldots, N \tag{3.4}$$

or globally,

$$\mathbb{CU} = \mathbb{CU}_1 \times \ldots \times \mathbb{CU}_N \tag{3.5}$$

where $f_i : \mathbb{C}_i \times \mathbb{U}_i \rightarrow \mathbb{R}^{ri}$ $\qquad i = 1, \ldots, N$

In general, the real system relations are not known, so an approximate model is used instead of the system mapping.

$$F_i : \mathbb{C}_i \times \mathbb{U}_i \times \mathbb{A}_i \rightarrow \mathbb{Y}_i \qquad i=1, \ldots, N$$

$$y_i = F_i\ (c_i , u_i , \alpha_i) \qquad i=1, \ldots, N \tag{3.6}$$

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

As before, the global model description can be written as

$$F : \mathbb{C} \times \mathbb{U} \times \mathbb{A} \rightarrow \mathbb{Y}$$

$$y = F\ (c , u , \alpha) \tag{3.7}$$

where

$$\alpha = (\alpha_1, \ldots, \alpha_N) \in \mathbb{A}_1 \times \ldots \times \mathbb{A}_N = \mathbb{A} \tag{3.8}$$

The overall performance index is the summation of the local performance indices:

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

$$Q\,(c, u, y) = \sum_{i=1}^{N} Q_i\,(c_i\,, u_i\,, y_i) \tag{3.9}$$

Just as in Chapter 2, the following assumption is made to clarify the future analysis of the algorithms.

Assumption 3.1

It is assumed throughout this chapter that :

1- Mappings $f(\,\cdot\,,\,\cdot\,)$, $F(\,\cdot\,,\,\cdot\,,\,\cdot\,)$, $K_{*i}(\,\cdot\,)$ and the performance index $Q\,(\,\cdot\,,\,\cdot\,,\,\cdot\,)$ are continuously Frèchet differentiable.

2- Mapping $f\,(\,\cdot\,,\,\cdot\,)$ is convex.

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

Based on the above definitions and assumptions, the task of determining the optimal operating condition for a hierarchical process can be formulated as follows :

$$\text{(HROP)} \qquad \min_{c,u,y} \; Q\,(c, u, y)$$

s.t.

$$y = K_*\,(c)$$

$$u = Hy$$

$$(c, u) \in \mathbb{CU}$$

It can be proved that subject to mild assumptions HROP can be replaced by an equivalent problem, HEOP, of the following form (Brdyś, 1983) :

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

(HEOP)

s.t.

$$F (c, u, \alpha) = K_* (c)$$

$$u = H F (c, u, \alpha)$$

$$(c, u) \in \mathbb{CU}$$

where $q : \mathbb{C} \times \mathbb{U} \times \mathbb{A} \rightarrow \mathbb{R}$ with

$$q(c , u , \alpha) = Q(c, u , F(c , u , \alpha)) \tag{3.10}$$

It is interesting to observe in the above formulation for hierarchical control that a set of variables is introduced to represent the interactive inputs and an extra equation $u = HF(c, u, \alpha)$ is added to preserve interaction balance between subsystems. It is also noted in the derivation of the formulation that in hierarchical cases it is essential to assume the separability of the performance index and the constraints with respect to the subsystem controller set-points and interactive inputs.

3.3 DECOMPOSITION AND COORDINATION

Consider a large scale process consisting of interconnected sub-processes. The hierarchical optimising control is usually implemented in a two-level structure. In the lower level local decision units are implemented to determine the local optimum operating conditions for each sub-process by maximising the subsystem performance index subject to local constraints while, in the upper level, a coordinator is introduced to preserve the interaction balance between sub-processes during the course of iteration. There are two different kinds of coordination strategies, i.e. Interaction Prediction method (Direct method) and Interaction Balance method (Price method) (Findeisen et al., 1980; Singh and Titli, 1978). Since the direct method is usually difficult to converge and there are restrictions on its application, the price method has been widely used in the practice of hierarchical optimising control (Findeisen et al., 1980). A new coordination strategy called Modifier Coordination (MC) mechanism which is found to be much more efficient than the price method in some circumstances is also presented in this chapter (Lin, Roberts, Wang and Wan, 1990; Lin, Amini and Roberts, 1990).

3.3.1 Price Coordination (Interaction Balance)

As mentioned above, the hierarchical optimisation is performed within a two-level structure. In the upper level the main task of the coordinator is to coordinate subsystem optimisation performed in the lower level. The overall optimisation problem is decomposed into independent local optimisation problems which are solved separately in the lower level. The lower level optimisation is arbitrated by the upper level coordinator through either a price correction term (interaction balance method) or a prediction of the interaction between subsystems (direct coordination method). On the other hand, the coordination is updated using the information obtained in the lower level optimisation. In the price coordination method intervention of the upper level coordination into the lower level optimisation problems is achieved by modifying their performance indices. The main feature of this method is that the interaction between the subsystems is ignored in the lower level optimisation and the interaction balance is enforced by the introduction of a price modification term in the local optimisation problems (Findeisen et al., 1980; Singh and Titli, 1978). Compared with the direct coordination approach the price coordination method is much more simple to implement and easier to converge. The major disadvantage of this coordination method is that the coordination efficiency depends closely on the Hessian structure of the real optimisation problem and is sometimes found to be poor in the sense that the convergence of the algorithm becomes very slow towards the end of iteration. The reason for this is that the convergence behaviour of the price coordination mechanism is governed by the features of the Lagrangian for the real optimisation problem around its saddle point.

3.3.2 Modifier Coordination

In order to improve the coordination efficiency and to overcome the other disadvantages suffered by the price coordination method a new coordination strategy for hierarchical process optimisation is proposed. Unlike the price coordination method this approach is not derived from the Lagrangian of the real optimisation problem and is, rather, a direct type approach.

In this approach the overall optimising control problem is decomposed into independent local optimisation problems by employing a separable model structure in the hierarchical optimisation scheme. The interaction between subsystems is considered as a kind of model-reality difference and is, therefore, taken care of by the modifier as found in the standard and improved MTS methods (Lin, Han, Roberts and Wan, 1989; Lin, Wang and Roberts, 1989; Lin, Amini and Roberts, 1990). This idea is also motivated by the early work of Ellis, Michalska and Roberts (1984). Since the coordination task is performed by the modifier which also compensates the model-reality difference, this approach is referred to as the Modifier Coordination (MC) approach. It is observed in the theoretical analysis that this approach retains the desired convergence behaviour of the improved version of the MTS method and is, compared with the price coordination method, very efficient.

3.3.3 Open-Loop Coordination and Introduction of Feedback

In the practice of on-line hierarchical optimising control of industrial processes, two different types of coordination methods are identified according to their use of real process information during the course of coordination, i.e. open-loop coordination and closed-loop coordination methods (Findeisen et al., 1980). In the open-loop methods the process mapping is approximated by a model and the optimisation is performed in an off-line fashion until the optimum is reached. Open-loop coordination methods rely upon an accurate mathematical model of the real process. Since there are inevitably model-reality differences in real circumstances, the accuracy of the optimum achieved is usually poor.

To overcome this problem process measurements are taken and fed back to decision units within the hierarchical optimisation structure (Findeisen et al., 1980). Such algorithms are referred to as closed-loop coordination methods. Since real process information is used in the optimisation, the accuracy of the optimum is significantly improved. There are two principle methods in which feedback information from the real process may be employed, i.e. global feedback and local feedback. In global feedback the process measurements are sent to the coordinator to arbitrate the subsystem coordination, while in local feedback the process measurements are sent to the local decision units to modify the lower level optimisation problems. Global feedback structure is generally more efficient than local feedback since process measurements are used in the coordination. It is, however, important to bear in mind that in the global feedback scheme information interchange between the real process and the coordinator is inevitable and this can be expensive in some circumstances, for instance, where the sub-processes are geographically far apart. These two major feedback schemes can be separately or jointly fitted into the ISOPE structure to form different hierarchical ISOPE methods which are able to achieve the true optimum regardless of the model-reality differences (Brdyś and Roberts, 1986).

3.3.4 Iterative Strategies for Hierarchical ISOPE Algorithms

Generally, the interaction balance between subsystems in the hierarchical optimisation is achieved in an iterative manner. To implement a closed-loop coordination mechanism within the ISOPE structure two different iterative strategies can be employed, i.e. single iterative loop and double iterative loop strategies.

In the single iterative strategy the controller set-point is updated simultaneously with the coordinator while, in the double iterative strategy, a nested double iterative loop structure is employed such that the subsystem coordination is separated from the controller set-point optimisation. Since there is only one iterative loop in the single iterative strategy, the total number of iterations required by this strategy is much less than that of those required by the double iterative strategy (Shao and Roberts, 1983). As a result, this strategy becomes very attractive when there is a limitation on the computing power and the process is of fast transient nature. On the other hand, the purpose of developing the double iterative strategy is to reduce as much as possible the number of controller set-point changes at the expense of an increase in the total number of iterations because in some circumstances, especially when the transient of the process is slow, the number of controller set-point changes plays a more important part than the total number of iterations in reducing the overall process adjusting time (Brdyś and Roberts, 1985).

3.3.5 Information Structures of Hierarchical ISOPE Algorithms

In the practice of optimising control there are two different information feedback structures, namely output and input-output feedback (Brdyś and Roberts, 1986; Chen, 1986). Generally, the input-output feedback structure, in which both input and output process measurements are used, is more efficient than the output feedback structure, in which only output process measurements are used, in the sense that it reduces the number of real system iterations. This becomes obvious when examining the way in which the real process information is extracted because in the modifier calculation the input-output feedback structure extracts the real process information from the process measurements in a more efficient manner. It is, however, important to note that in some circumstances the input-output feedback structure may not be possible to implement due to the physical configuration of the system.

3.4 SINGLE ITERATIVE LOOP ISOPE ALGORITHMS

Two single iterative ISOPE algorithms are presented in this section. In the first algorithm the price coordination method has been implemented in a global feedback structure while in the second algorithm the modifier coordination technique is incorporated with the ISOPE framework (Amini, Lin and Roberts, 1990; Lin, Roberts, Wang and Wan, 1990; Lin, Amini and Roberts, 1990). The input-output information feedback structure is used in both algorithms.

3.4.1 Single Iterative ISOPE Algorithms with Global Feedback (AL1)

We follow the analysis of Brdyś and Roberts (1986) in which we first establish the Lagrangian function associated with HEOP, which is equivalent to HROP, the real optimising control problem, and then develop the optimality conditions for it. It is assumed that the real system input and output measurements are available. In order to separate parameter estimation from system optimisation, HEOP is expanded by introducing a single new variable v, as follows :

(HEOP1)

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

s.t.

$$F\,(v, HK_*(v), \alpha) = K_*\,(v)$$

$$u = H\,K_*\,(v)$$

$$(c, u) \in \mathbb{CU}$$

$$v = c$$

The Lagrangian function associated with HEOP1 takes the form

$$L\,(c, u, v, \alpha, p, \lambda, \xi, \mu) = q\,(c, u, \alpha) + p^T \Big[u - H\,K_*\,(v) \Big]$$

$$+ \xi^T f\,(c, u) + \lambda^T (v - c) + \mu^T \Big[F\,(v, HK_*(v), \alpha) - K_*\,(v) \Big] \qquad (3.11)$$

Hence, the optimality conditions (Luenberger, 1984; Roberts, 1979) are

$$\frac{\partial^T L}{\partial c} = \frac{\partial^T q(c\ ,\ u\ ,\ \alpha\ )}{\partial c} - \lambda + \frac{\partial^T f(c\ ,\ u)}{\partial c}\ \xi = 0 \tag{3.12}$$

$$\frac{\partial^T L}{\partial u} = \frac{\partial^T q(c\ ,\ u\ ,\ \alpha\ )}{\partial u} + p + \frac{\partial^T f(c\ ,\ u)}{\partial u}\ \xi = 0 \tag{3.13}$$

$$\frac{\partial^T L}{\partial v} = \lambda + \left[\frac{\partial^T F\ (v,\ HK_*(v),\ \alpha\ )}{\partial v} - \frac{\partial^T K_*(v)}{\partial v} \right] \mu - \frac{\partial^T K_*(v)}{\partial v}\ H^T p = 0 \tag{3.14}$$

$$\frac{\partial^T L}{\partial \alpha} = \frac{\partial^T q(c\ ,\ u\ ,\ \alpha\ )}{\partial \alpha} + \frac{\partial^T F\ (v,\ HK_*(v),\ \alpha\ )}{\partial \alpha}\ \mu = 0 \tag{3.15}$$

$$\frac{\partial^T L}{\partial p} = \hat{u}\ (v^k) - HK_*(v^k) = 0 \tag{3.16}$$

$$\frac{\partial^T L}{\partial \lambda} = v - c = 0 \tag{3.17}$$

$$\frac{\partial^T L}{\partial \mu} = F\ (v,\ HK_*(v),\ \alpha\ ) - K_*\ (v) = 0 \tag{3.18}$$

$$\xi_{ij}\ f_{ij}\ (c\ ,\ u) = 0 \qquad \xi_{ij} \geq 0 \qquad j \in J_i \qquad i = 1\ ,\ ...,\ N \tag{3.19}$$

Equations (3.12) and (3.13) give the decomposed model-based optimisation problems

$$(\text{HMMOP})_i \qquad \begin{array}{l} \min\limits_{c_i, u_i} \left\{ q_i\ (c_i\ ,\ u_i\ ,\ \alpha_i) - \lambda_i^T\ c_i + p_i^T\ u_i \right\} \\ \text{s.t. } f_i\ (c_i\ ,\ u_i) \leq 0 \qquad i = 1,\ ...,\ N \end{array}$$

The decomposed parameter estimation problems are

$$F_i\ (v_i\ ,\ HK_*(v),\ \alpha_i\ ) = K_*\ (v)_i \tag{3.20}$$

resulting in parameters α.

Finally the modifiers are obtained from equations (3.14) and (3.15)

$$\lambda\ (v\ ,\ \alpha\ ,\ p) = \left[\frac{\partial^T F\ (v,\ HK_*(v),\ \alpha\ )}{\partial v} - \frac{\partial^T K_*(v)}{\partial v} \right] \left[\frac{\partial^T F\ (v,\ HK_*(v),\ \alpha\ )}{\partial \alpha} \right]^{-1}$$
$$\times \left[\frac{\partial^T q(v\ ,\ HK_*(v)\ ,\ \alpha\ )}{\partial \alpha} \right] + \frac{\partial^T K_*(v)}{\partial v}\ H^T p \tag{3.21}$$

Using equation (3.16) the price updating formula is obtained as follows

$$p^{k+1} = p^k + \epsilon_p (\hat{u}^k - u^k) \tag{3.22}$$

where

$$u^k = HK_*(v^k) \tag{3.23}$$

It is worth noticing that the modifier λ takes into account the model-reality differences by modifying the performance index in the model-based optimisation problem. In fact equation (3.21) can be re-written as (Brdyś and Roberts, 1986) :

$$\lambda (v , \alpha , p) = \frac{\partial^T \hat{q}(v , \alpha)}{\partial v} - \frac{\partial^T q_*(v)}{\partial v} + \frac{\partial^T K_*(v)}{\partial v} H^T p \tag{3.24}$$

where

$$\hat{q} (v , \alpha) = q(v , HK_*(v) , \alpha) \tag{3.25}$$

$$q_*(v) = Q (v , HK_*(v)) \tag{3.26}$$

Substituting (3.24) into (3.12) and (3.13), which are the Kuhn-Tucker optimality conditions for HMMOP, yields :

$$\frac{\partial^T q(c , u , \alpha)}{\partial c} - \frac{\partial^T \hat{q}(v , \alpha)}{\partial v} + \frac{\partial^T q_*(v)}{\partial v} - \frac{\partial^T K_*(v)}{\partial v} H^T p + \frac{\partial^T f(c , u)}{\partial c} \xi = 0 \tag{3.27}$$

$$\frac{\partial q^T(c , u , \alpha)}{\partial u} + p + \frac{\partial^T f(c , u)}{\partial u} \xi = 0 \tag{3.28}$$

Combining (3.27) and (3.28) results in :

$$\frac{\partial^T q(c , u , \alpha)}{\partial c} + \frac{\partial^T K_*(v)}{\partial v} H^T \frac{\partial^T q(c , u , \alpha)}{\partial u} - \frac{\partial^T \hat{q}(v , \alpha)}{\partial v} + \frac{\partial^T q_*(v)}{\partial v} +$$

$$\left[\frac{\partial^T f(c , u)}{\partial c} + \frac{\partial^T K_*(v)}{\partial v} H^T \frac{\partial^T f(c , u)}{\partial u} \right] \xi = 0 \tag{3.29}$$

When $\hat{c} = v$, the equilibrium for the optimising algorithm is achieved; Then equation (3.29) takes the following form

$$\frac{\partial^T q_*(v)}{\partial v} + \left[\frac{\partial^T f(v\ ,\ HK_*(v))}{\partial v} + \frac{\partial^T K_*(v)}{\partial v} H^T \frac{\partial^T f(v\ ,\ HK_*(v))}{\partial u} \right] \xi = 0 \tag{3.30}$$

and it is straight forward to see that equations (3.19) and (3.30) are precisely the Kuhn-Tucker conditions for HROP.

Remark 1

By carefully examining the above derivation it is observed that the modifier λ can be partitioned into two separate parts, $\overline{\lambda}$ and $\tilde{\lambda}$ with

$$\overline{\lambda} = \frac{\partial^T \hat{q}(v\ ,\ \alpha\ )}{\partial v} - \frac{\partial^T q_*(v)}{\partial v} \tag{3.31}$$

$$\tilde{\lambda} = \frac{\partial^T K_*(v)}{\partial v} H^T p \tag{3.32}$$

where $\overline{\lambda}$ takes into account the model-reality differences while $\tilde{\lambda}$ together with $p^T u$ in HMMOP looks after the interaction balance.

Remark 2

Substituting (3.24) into the model-based optimisation problem $(HMMOP)_i$, $i=1,...,N$, and differentiating the performance index with respect to p we obtain the dual gradient of the optimisation problem HMMOP,

$$\hat{u} - \frac{\partial^T K_*(v)}{\partial v} H^T c = \text{dual gradient of HMMOP}$$

In order to achieve the desired convergence behaviour in the price updating mechanism, it is important to use the gradient of the dual of the model-based optimisation problem HMMOP to update the price vector (Findeisen et al., 1980; Luenberger, 1984) . However, according to equations (3.22) and (3.23) the innovation term for updating the price is not the same as the gradient of the dual of HMMOP. In fact, it is observed through simulation that the convergence behaviour of the price updating mechanism represented in equations (3.22) and (3.23) is so poor that the method does not converge even for a simple example.

In order to remedy the above problem we have modified the price updating mechanism $p^{k+1} = p^k + \epsilon_p (\hat{u}^k - u^k)$ by changing the previous innovation term to

$$\hat{u}^k - \Big(HK_*(v) + HK'_*(v) (\hat{c} - v) \Big)$$

Hence, the new price updating formula is

$$p^{k+1} = p^k + \epsilon_p \Big[\hat{u}^k - (HK_*(v) + HK'_*(v) (\hat{c} - v)) \Big] \tag{3.33}$$

The terms enclosed by the brackets in equation (3.33) are nothing more than the linearisation of $HK_*(c)$ at v.

It is worth mentioning that introducing the new innovation term does not affect the optimum of HMOP in any way, because $HK_*(v) + HK'_*(v) \times v$ is treated as constant in the model-based optimisation. Therefore, the innovation term in the new price updating mechanism is equal to the dual gradient of an optimisation problem equivalent to HMMOP.

Based on the above discussion the algorithm description is presented as follows :

Step 1

For given values of v^k, the parameters α^k are estimated according to

$$F (v^k, HK_*(v), \alpha) = K_* (v^k)$$

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 modifier

$$\lambda^k = \lambda (v^k, \alpha^k, p^k) = \Big[\frac{\partial^T F (v^k, HK_*(v^k), \alpha^k)}{\partial v} - \frac{\partial^T K_*(v^k)}{\partial v} \Big]$$

$$\times \Big[\frac{\partial^T F (v^k, HK_*(v^k), \alpha^k)}{\partial \alpha} \Big]^{-1} \Big[\frac{\partial^T q (v^k, HK_*(v^k), \alpha^k)}{\partial \alpha} \Big]$$

$$+ \frac{\partial^T K_*(v^k)}{\partial v} H^T p^k \tag{3.34}$$

Step 2

For given λ^k, α^k and p^k, solve the following model-based optimisation problem

(HMMOP)

$$\min_{c,u} \left\{ q\,(c\,,\,u\,,\,\alpha) - \lambda^T c + p^T u \right\}$$

$$\text{s.t.} \;\; f\,(c\,,\,u) \leq 0$$

The solution is denoted by $\hat{c}\ (\alpha^k, \lambda^k, p^k)$, $\hat{u}\ (\alpha^k, \lambda^k, p^k)$ or $\hat{c}^k$, $\hat{u}^k$ for short.

The iterative variables v^k and p^k are updated according to the formulae

$$v^{k+1} = v^k + \epsilon_v \,(\, \hat{c}^k - v^k \,) \tag{3.35}$$

$$p^{k+1} = p^k + \epsilon_p \left[\, \hat{u}^k - (\, HK_*(v) + HK_*'(v)\,(\hat{c} - v)\,)\, \right] \tag{3.36}$$

The above description of the algorithm does not reveal the hierarchical structure of the problem. In order to gain insight into the decomposition and coordination nature of the problem an alternative description of the algorithm is also provided.

The ith local decision unit

The optimisation problem for the i^{th} local decision unit is the model-based optimisation problem $(HMMOP)_i$

$(HMMOP)_i$

$$\min_{c_i,u_i} \left\{ q_i\,(c_i\,,\,u_i\,,\,\alpha_i) - \lambda_i^T c_i + p_i^T u_i \right\}$$

$$\text{s.t.} \;\; f_i\,(c_i\,,\,u_i) \leq 0 \qquad\qquad i = 1, \ldots, N$$

The coordination mechanism

The coordination task is accomplished by enforcing equation (3.36).

The parameter estimation problem

The parameter estimation problem is addressed by the equation listed in step 1.

Finally, the interaction between parameter estimation and system optimisation is balanced when the convergence of (3.35) is achieved. The information structure for this method is shown in Figure 3.1.

3.4.2 Single Iterative ISOPE Algorithms using Modifier Coordination (MC)

The main idea of the Modifier Coordination Strategy (MC) is to convert the interaction between subsystems to a form of model-reality difference by introducing separable subsystem models in the model-based optimisation problem. Then, the decomposition of the whole optimising control problem follows automatically. Since the approach to problems with input dependent constraints is different from that of problems with only set-point dependent constraints, the algorithmic description for each case is presented separately.

3.4.2a MC for problems without input dependent constraints (AL2)

In this case the local constraint set $\mathbb{CU}$ in the real optimisation problem HROP will become input independent and is, therefore, denoted by $\mathbb{C}$, instead of $\mathbb{CU}$, where

$$\mathbb{C} = \mathbb{C}_1 \times \dots \times \mathbb{C}_N \text{ with } \mathbb{C}_i = \left\{ c_i;\ f_i(c_i) \le 0 \right\},\ i = 1, \dots, N \tag{3.37}$$

To clarify the mathematical derivation, we introduce the following definitions :

Definition 3.1

Define

$$q_*(v) = Q(v, HK_*(v), K_*(v)) \tag{3.38}$$

Let

$$\hat{K}'_*(v) = \begin{bmatrix} [K'_*(v)]_{11} & & & & \\ & [K'_*(v)]_{22} & & & O \\ & & \cdot & & \\ & & & \cdot & \\ & O & & & \cdot \quad \cdot \\ & & & & [K'_*(v)]_{NN} \end{bmatrix} \tag{3.39}$$

and

$$\hat{HK}'_*(v) = \begin{bmatrix} [HK'_*(v)]_{11} & & & & \\ & [HK'_*(v)]_{22} & & & O \\ & & \cdot & & \\ & & & \cdot & \\ & O & & & \cdot \quad \cdot \\ & & & & [HK'_*(v)]_{NN} \end{bmatrix} \tag{3.40}$$

where

$\Big[\ K'_*(v)\ \Big]_{ii}$ and $\Big[\ HK'_*(v)\ \Big]_{ii}$, $i = 1, \ldots, N$ are diagonal sub-matrices in the following partitions :

$$K'_*(v) = \begin{bmatrix} [K'_*(v)]_{11} & [K'_*(v)]_{12} & \cdot & \cdot & [K'_*(v)]_{1N} \\ [K'_*(v)]_{21} & [K'_*(v)]_{22} & \cdot & \cdot & [K'_*(v)]_{2N} \\ \vdots & \vdots & & & \vdots \\ [K'_*(v)]_{N1} & [K'_*(v)]_{N2} & \cdot & \cdot & [K'_*(v)]_{NN} \end{bmatrix} \tag{3.41}$$

and

$$
HK'_*(v) = \begin{bmatrix} [HK'_*(v)]_{11} & [HK'_*(v)]_{12} & \cdot & \cdot & [HK'_*(v)]_{1N} \\ [HK'_*(v)]_{21} & [HK'_*(v)]_{22} & \cdot & \cdot & [HK'_*(v)]_{2N} \\ \cdot & \cdot & & & \cdot \\ \cdot & \cdot & & & \cdot \\ \cdot & \cdot & & & \cdot \\ [HK'_*(v)]_{N1} & [HK'_*(v)]_{N2} & \cdot & \cdot & [HK'_*(v)]_{NN} \end{bmatrix} \tag{3.42}
$$

Define

$$
q(c\,,\,v) = \sum_{i=1}^{N} q_i(c_i\,,\,v) \tag{3.43}
$$

with

$$
q_i(c_i\,,\,v) = Q_i\,(\,c_i\,,\,u_i(c_i\,,\,v)\,,\,y_i(c_i\,,\,v)\,) \quad i = 1, \ldots, N \tag{3.44}
$$

and

$$
u_i(c_i\,,\,v) = \Big[\,HK_*(v)\,\Big]_i + \Big[\,HK'_*(v)\,\Big]_{ii}\,(c_i - v_i) \quad i = 1, \ldots, N \tag{3.45}
$$

$$
y_i(c_i\,,\,v) = \Big[\,K_*(v)\,\Big]_i + \Big[\,K'_*(v)\,\Big]_{ii}\,(c_i - v_i) \quad i = 1, \ldots, N \tag{3.46}
$$

or globally,

$$
u(c\,,\,v) = HK_*(v) + \hat{HK}'_*(v)\,(c - v) \tag{3.47}
$$

$$
y(c\,,\,v) = K_*(v) + \hat{K}'_*(v)\,(c - v) \tag{3.48}
$$

It is noted from definition 3.1 that for any given v a fully decomposed optimisation problem may be established if equations (3.47) and (3.48) are chosen as the process model in the model-based optimisation. In general, there is model-reality difference between u(c,v) and $HK_*(v)$ and between y(c,v) and $K_*(v)$ even if c eventually converges to v. In order to compensate for the model-reality differences the modifier should be designed such that the Kuhn-Tucker conditions of the real optimisation problem HROP are preserved in the model-based optimisation when the iterations are complete. This can be easily achieved by subtracting the gradients of the performance index in HROP from those of the model-based optimisation problem (Lin, Han, Roberts and Wan, 1989; Lin, Roberts, Wang and Wan, 1990).

Based on the above discussion the Modifier Coordination Strategy for problems without input dependent inequality constraints is developed and its algorithmic description is written as follows :

Description of algorithm 1 (AL2)

Step 1

For given values of v^k, calculate the modifier $\lambda(v)$ according to :

$$\lambda^k = \lambda\,(v^k) = q_c'\,(v^k, v^k) - q_*'\,(v^k) \tag{3.49}$$

Step 2

For given λ^k solve the model-based optimisation problem HMMOP, decomposed into N sub-problems $(\mathrm{HMMOP})_i$, i=1, ..., N, with the i^{th} sub-problem taking the following form :

$$(\mathrm{HMMOP1})_i \quad \begin{array}{l} \min\limits_{c_i} \left\{ q_i(c_i\,, v^k) - (\lambda_i^k)^T c_i \right\} \\ \\ \text{s.t. } f_i(c_i) \leq 0 \end{array}$$

Denote $\hat{c}^k = \left[\, (\hat{c}_1^k)^T \,, \ldots \,, (\hat{c}_N^k)^T \right]^T$, where $\hat{c}_i^k$ is the short form for $\hat{c}_i(v^k)$, as the optimum of the i^{th} model-based optimisation sub-problem.

The controller set-point vector v is updated using a relaxation technique :

$$v^{k+1} = v^k + \epsilon_v\,(\hat{c}^k - v^k) \tag{3.50}$$

where $0 < \epsilon_v \leq 1$ is an iterative gain factor. Figure 3.2 shows the information structure for AL2.

3.4.2b MC for problems with input dependent constraints (AL3)

It can be proved that the algorithm AL2 may fail to produce the true optimum when the problem has input dependent constraints. The reason for this is that, in contrast to the case without input dependent constraints, the optimum conditions for the model-based optimisation problem HMMOP1 may not match the Kuhn-Tucker conditions for the real problem HROP at the algorithmic solution point because there is, in general, a difference between the derivatives of the constraints in the problems HMMOP1 and HROP.

In order to overcome this problem an adjusting term is added to the modifier calculating formula to compensate for the derivative difference between HMMOP1 and HROP, therefore, making up the mismatch of the Kuhn-Tucker conditions for HMMOP1 and HROP at the algorithmic solution point. The algorithm implementation of this idea is described as follows :

Description of algorithm 2 (AL3)

Step 1

For given values of v^k, calculate the modifier $\lambda(v)$ according to :

$$\lambda^k = \lambda\,(v^k) = q'_c\,(v^k,v^k) - q'_*\,(v^k)$$
$$+ 2\rho\,\hat{f}(\hat{c}^{k-1},v^{k-1})\,\sigma(\hat{c}^{k-1},v^{k-1})\,f'_u(v^k,HK_*(v^k))\Big[\,\hat{HK}'_*(v^k) - HK'_*(v^k)\Big] \tag{3.51}$$

In the above

$$\hat{f}(c,v) = f(v,HK_*(v)) + \Big[\,f'_c(v,HK_*(v)) + f'_u(v,HK_*(v))\,\hat{HK}'_*(v)\Big]\,(c - v) \tag{3.52}$$

$$\sigma(c,v) = \mathrm{diag}\Big[\,\sigma_1(c,v)\,,\,\dots\,,\,\sigma_m(c,v)\Big] \tag{3.53}$$

with $\sigma_i(c,v) = 0$ if $\hat{f}_i(c,v) < 0$ and $\sigma_i(c,v) = 1$ if $\hat{f}_i(c,v) \geq 0$, m is the dimension of f(c,u), and ρ is a penalty coefficient which is normally assumed to be large.

Step 2

For given λ^k solve the model-based optimisation problem HMMOP, decomposed into N sub-problems $HMMOP_i$, $i = 1, ..., N$, with the i^{th} sub-problem taking the following form,

$$(HMMOP)_i \quad \begin{aligned} &\min_{c_i, w_i} \left\{ q_i(c_i, v^k) - (\lambda_i^k)^T c_i + \rho \,\|\, w_i \,\|^2 \right\} \\ &\text{s.t. } \hat{f}_i(c_i, v^k) + w_i \leq 0 \end{aligned}$$

Denote $\hat{c}^k = \left[(\hat{c}_1^k)^T, ..., (\hat{c}_N^k)^T \right]^T$ and $\hat{w}^k = \left[(\hat{w}_1^k)^T, ..., (\hat{w}_N^k)^T \right]^T$, where $\hat{c}_i^k$ and $\hat{w}_i^k$ are the short forms for $\hat{c}_i(v^k)$ and $\hat{w}_i(v^k)$, respectively, the optimum of the i^{th} model-based optimisation sub-problem.

The controller set-point vector v is updated using a relaxation technique :

$$v^{k+1} = v^k + \epsilon_v (\hat{c}^k - v^k) \tag{3.54}$$

where $0 < \epsilon_v \leq 1$ is the iterative gain factor.

An inspection of (3.51) shows that a compensation term is added to the modifier calculation formula to make up the constraint derivative difference between HMMOP1 and HROP. In fact, (3.51) can be rewritten as :

$$\begin{aligned} \lambda^k = &\left[q'_c (v^k, v^k) + 2\rho \hat{f}(\hat{c}^{k-1}, v^{k-1}) \, \sigma(\hat{c}^{k-1}, v^{k-1}) \, f'_u(v^k, HK_*(v^k)) \, \hat{HK}'_*(v^k) \right] \\ &- \left[q'_* (v^k) + 2\rho \hat{f}(\hat{c}^{k-1}, v^{k-1}) \, \sigma(\hat{c}^{k-1}, v^{k-1}) \, f'_u(v^k, HK_*(v^k)) \, HK'_*(v^k) \right] \end{aligned} \tag{3.55}$$

At the algorithmic solution point, the first part of the right hand side of (3.55) (enclosed by a pair of square brackets) will cancel with the Lagrangian of HMMOP (with λ^k being taken out) when ρ approaches infinity. The second part of the right hand side of (3.55) is found to be, as ρ approaches infinity, the Lagrangian of (HROP) when c and v are substituted by the algorithmic solution v^*.

It is also observed in the equations (3.51) and (3.55) that instead of $2\rho\ \hat{f}(\hat{c}^k,v^k)\ \sigma(\hat{c}^k,v^k)$, $2\rho\ \hat{f}(\hat{c}^{k-1},v^{k-1})\ \sigma(\hat{c}^{k-1},v^{k-1})$ is used as the Kuhn-Tucker multiplier (Luenberger, 1984). The reason for this is simply because $\hat{c}^k$ and v^k are not yet available when calculating λ^k. Figure 3.3 represents the information structure of AL3.

3.5 DOUBLE ITERATIVE LOOP ISOPE ALGORITHMS

The motivation for introducing double iterative strategies is to reduce as much as possible the real process iterations at the expense of a possible increase in the off-line iterations because in many circumstances the total adjusting time of an optimising control algorithm is dominated by the settling time of the control process and, hence, by the number of controller set-point changes. The basic idea behind the implementation of such a scheme is to separate the iteration for achieving interaction balance from that for controller set-point optimisation. This idea can be realised using a double nested iterative loop structure (Shao and Roberts, 1983).

3.5.1 A Double Loop ISOPE Algorithm using Price Coordination (AL4)

In this section a double loop ISOPE algorithm is presented, which can be considered as the double loop version of the single iterative ISOPE algorithm presented in Section 3.4.1. A similar approach to that used for the derivation of the algorithm presented in Section 3.4.1 may be employed here to derive the algorithm. Since the basic idea behind the double iterative strategy is to separate the operation of achieving interaction balance from real system iteration, we start the algorithm derivation with the following equivalent optimisation problem which takes a slightly different form from HEOP1 :

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

(HEOP2)

s.t.

$$F\ (v,\ HK_*(v),\ \alpha\ ) = K_*\ (v)$$

$$u = H\ K_*\ (c)$$

$$(c,\ u) \in \mathbb{CU}$$

$$v = c$$

Comparing the optimisation problems HEOP1 and HEOP2 it is observed that the constraint $u = HK_*(v)$ in HEOP1 has been changed to $u = HK_*(c)$, reflecting the purpose of separating interaction balance from real process operation.

The Lagrangian function associated with HEOP2 takes the form

$$L(c, u, v, \alpha, p, \lambda, \xi, \mu) = q(c, u, \alpha) + p^T \Big[u - H K_*(c) \Big]$$

$$+ \xi^T f(c, u) + \lambda^T (v - c) + \mu^T \Big[F(v, HK_*(v), \alpha) - K_*(v) \Big] \tag{3.56}$$

Hence, the optimality conditions (Luenberger, 1984; Roberts, 1979) are

$$\frac{\partial^T L}{\partial c} = \frac{\partial^T q(c, u, \alpha)}{\partial c} - \lambda + \frac{\partial^T f(c, u)}{\partial c} \xi + \frac{\partial^T K_*(c)}{\partial c} H^T p = 0 \tag{3.57}$$

$$\frac{\partial^T L}{\partial u} = \frac{\partial^T q(c, u, \alpha)}{\partial u} + p + \frac{\partial^T f(c, u)}{\partial u} \xi = 0 \tag{3.58}$$

$$\frac{\partial^T L}{\partial v} = \lambda + \Big[\frac{\partial^T F(v, HK_*(v), \alpha)}{\partial v} - \frac{\partial^T K_*(v)}{\partial v} \Big] \mu = 0 \tag{3.59}$$

$$\frac{\partial^T L}{\partial \alpha} = \frac{\partial^T q(c, u, \alpha)}{\partial \alpha} + \frac{\partial^T F(v, HK_*(v), \alpha)}{\partial \alpha} \mu = 0 \tag{3.60}$$

$$\frac{\partial^T L}{\partial p} = u - HK_*(c) = 0 \tag{3.61}$$

$$\frac{\partial^T L}{\partial \lambda} = v - c = 0 \tag{3.62}$$

$$\frac{\partial^T L}{\partial \mu} = F(v, HK_*(v), \alpha) - K_*(v) = 0 \tag{3.63}$$

$$\xi_{ij} f_{ij}(c, u) = 0 \qquad \xi_{ij} \geq 0 \qquad j \in J_i \qquad i = 1, \ldots, N \tag{3.64}$$

Equation (3.63) defines the parameter estimation problem

$$F(v, HK_*(v), \alpha) = K_*(v) \tag{3.65}$$

and the modifier calculation formula is obtained from equations (3.59) and (3.60)

$$\lambda (v , \alpha) = \left[\frac{\partial^T F (v, HK_*(v), \alpha)}{\partial v} - \frac{\partial^T K_*(v)}{\partial v} \right] \left[\frac{\partial^T F (v, HK_*(v), \alpha)}{\partial \alpha} \right]^{-1}$$

$$\times \left[\frac{\partial^T q(v , HK_*(v) , \alpha)}{\partial \alpha} \right] \tag{3.66}$$

Through simple calculation it is straight forward to see that the above formula is equivalent to

$$\lambda(v , \alpha) = \left[\frac{\partial^T F(v , H K_*(v) , \alpha)}{\partial v} - \frac{\partial^T K_*(v)}{\partial v} \right] \frac{\partial^T Q(v , HK_*(v) , y)}{\partial y} \tag{3.67}$$

where $y = F(v , H K_*(v) , \alpha)$.

Equations (3.57) and (3.58) define the following optimisation problem

$$\min_{c,u} \left\{ q (c , u , \alpha) - \lambda^T c \right\}$$

s.t.

$$u = HK_*(c)$$

$$f (c , u) \leq 0$$

In order to transfer the above optimisation problem to a pure model-based one it is necessary to replace $K_*(c)$ by a model. This model should be a true representation of the reality at the end of the iteration for otherwise an inaccuracy in the optimum may occur. In the light of this, a linear approximate model is employed to transfer the above problem to the following model-based optimisation problem.

$$\min_{c,u} \left\{ q (c , u , \alpha) - \lambda^T c \right\}$$

(HMMOP)

s.t.

$$u = HK_*(v) + HK_*'(v) (c - v)$$

$$f (c , u) \leq 0$$

Using the price coordination method the above model-based optimisation problem can be solved hierarchically.

The lower level optimisation problem for the i^{th} subsystem is described as follows :

$$(\mathrm{HMMOP})_i \quad \min_{c_i, u_i} \left\{ q_i(c_i , u_i , \hat{\alpha}_i^k) - \lambda_i^T c_i + (p_i^L)^T u_i - d_i^T c_i \right\}$$

$$\text{s.t. } f_i(c_i , u_i) \leq 0$$

where

$$d_i = \left[(p^L) \, HK_*(v^k) \right]_i, \quad \mathbb{CU} = \mathbb{CU}_1 x \cdots x \, \mathbb{CU}_N$$

and the price vector is updated by using the gradient of the dual problem of HMMOP.

Starting from some initial v^0 satisfying $v^0 \in \mathbb{C}$ the k^{th} iteration is described as follows :

Step 1 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 the following equation :

$$F(v^k , HK_*(v^k) , \alpha) = K_*(v^k) \tag{3.68}$$

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 modifier $\lambda^k = \lambda(v^k , \hat{\alpha}^k)$ according to (3.67).

Step 2 For given $\hat{\alpha}^k$ and v^k, solve the following model-based optimisation problem by using the price method

$$(\mathrm{HMMOP}) \quad \min_{c,u} \left\{ q(c , u , \hat{\alpha}^k) - \lambda^T(v^k , \hat{\alpha}^k) c \right\}$$

s.t.

$$u = HK_*(v^k) + HK_*'(v^k) (c - v^k)$$

$$(c,u) \in \mathbb{CU}$$

The solution is denoted by $\hat{c}(v^k , \hat{\alpha}^k)$ and $\hat{u}(v^k , \hat{\alpha}^k)$ or $\hat{c}^k$, $\hat{u}^k$ for short.

Since HMMOP is to be solved hierarchically using the price method, it is necessary to have an inner iterative loop to achieve the interaction balance of the constraints $u = HK_*(v^k) + HK_*'(v^k)(c - v^k)$. For given v^k and $\hat{\alpha}^k$, the inner-loop coordination can be described as follows :

Step 2a For given p^L, the inner-loop optimisation problem for the i^{th} subsystem is

$$\min_{c_i, u_i} \left\{ q_i(c_i , u_i , \hat{\alpha}_i^k) - \lambda_i^T c_i + (p_i^L)^T u_i - d_i^T c_i \right\}$$

$$\text{s.t.} \quad (c_i , u_i) \in \mathbb{CU}_i$$

where

$$d_i = \Big[(p^L) \, HK_*(v^k) \Big]_i, \quad \mathbb{CU} = \mathbb{CU}_1 x \cdots x \, \mathbb{CU}_N$$

The solution is denoted as $\tilde{c}_i(v^k , \hat{\alpha}^k , p^L)$ and $\tilde{u}_i(v^k , \hat{\alpha}^k , p^L)$ or $\tilde{c}_i^k$, $\tilde{u}_i^k$ for short.

Step 2b For given p^L, $\tilde{c}^L$ and $\tilde{u}^L$, where

$$(\tilde{c}^L)^T = \Big[(\tilde{c}_1^L)^T \ \cdots \ (\tilde{c}_N^L)^T \Big] \text{ and } (\tilde{u}^L)^T = \Big[(\tilde{u}_1^L)^T \ \cdots \ (\tilde{u}_N^L)^T \Big]$$

Calculate p^{L+1} according to :

$$p^{L+1} = p^L + \epsilon_p \Big[(\tilde{u}^L - H\, K_*(v^k) - H\, K_*'(v^k) \, (\tilde{c}^k - v^k) \Big]$$

The inner-loop iteration is terminated when $p^{L+1} = p^L$.

The iterative variable v^k is updated according to the following formula :

$$v^{k+1} = v^k + \epsilon_v \, (\hat{c}^k - v^k) \tag{3.69}$$

where ϵ_v is a prescribed gain factor. The iteration is terminated when $\hat{c}^k = v^k$. The configuration of this algorithm is illustrated in block diagrams in Figures 3.4 and 3.5.

Remark 1

Inspecting the algorithm, it is not difficult to see from the above description that a double iterative structure, which is featured by its two nested iterative loops, is employed. In the outer loop, where v^k is updated, real process measurements are required to calculate the modifier while, in the inner loop, where model-based optimisations are carried out, no real process information is needed to achieve the model interaction balance.

Remark 2

Unlike the previous double loop techniques (Brdyś et al. 1986a; Chen et al. 1986; Brdyś et al. 1986; Hendawy, 1989), the derivatives of the model used in the model-based optimisation are adapted each time the inner loop is entered using the derivatives of real process measurements. This is the reason why this algorithm is also referred to as the Two-Model approach (Lin, Kambhampati and Roberts, 1989). 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.

Remark 3

This algorithm can be viewed as an attempt to combine the ISOPE technique with the approximate linear model approach. Normally, the original double loop algorithm (Brdyś et al. 1986a) can be very efficient as long as the difference between the model and reality is not too large. Therefore, if we can adapt the model to reality, the number of set-point changes is bound to be reduced significantly. This is precisely the motivation for combining the ISOPE method with the approximate linear model approach.

3.5.2 A Double Loop ISOPE Algorithm using Modifier Coordination (AL5)

In many cases where the process under consideration is of a slow transient nature, it is desired to reduce, to the maximum extent, the number of controller set-point changes even at the expense of an increase in the number of off-line iterations. This is precisely the motivation of introducing the double loop iterative strategy. The major difference between this algorithm and the double loop ISOPE algorithm presented in Section 3.5.1 is that in this algorithm instead of the price coordinator a modifier coordination mechanism is employed to achieve the interaction balance between subsystems. In addition, the improved version of MTS (Lin, Han, Roberts and Wan, 1989) rather than the standard MTS method is used in the outer loop for controller set-point optimisation.

The basic idea of the presented algorithm is to find the optimum controller set-point in the outer loop using a centralised optimising control algorithm and to dedicate an extra iterative loop to the hierarchical solution of the model-based optimisation problem which is required by the centralised optimising control algorithm. The implementation of the presented algorithm is described below.

For given v^k, the outer loop iteration of the presented double iterative strategy may be described as follows :

<u>Step 1</u> Apply v^k to the process to obtain the process derivative $K_*'(v^k)$ using process perturbation technique (Roberts, 1979; Brdyś and Roberts, 1987). Construct $u(\cdot\ ,\ v^k)$, a linear approximate model for the process, where

$$u(c\ ,\ v) = HK_*(v) + HK_*'(v)\ (c - v) \tag{3.70}$$

Calculate the modifier according to :

$$\lambda^k = q_c'(v^k\ ,\ u(v^k\ ,\ v^k)\ ) + q_u'(v^k\ ,\ u(v^k\ ,\ v^k)\ )\ u_c'(v^k\ ,\ v^k) - q_*'(v^k) \tag{3.71}$$

where $q_*(v^k)$ is defined in equation (3.26).

<u>Step 2</u> For given λ^k and the linear approximate model $u(\cdot\ ,\ v^k)$ solve the following optimisation problem :

$$\text{(HMMOP)} \qquad \begin{aligned} &\min_c \left\{ \sum_{i=1}^{N} q_i(c_i\ ,\ u_i\ ) - (\lambda^k)^T c \right\} \\ &\text{s.t. } u = u(c\ ,\ v^k) \\ &f_i(c_i\ ,\ u_i) \leq 0 \quad i = 1, \ldots, N \end{aligned}$$

Record the optimum of the above optimisation problem, and then the controller set-point v^k is updated according to :

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

The iteration stops when $v^{k+1} = v^k$.

Since the model-based optimisation problem HMMOP is to be solved hierarchically, an inner loop which is nested with the outer loop has to be introduced to achieve the interaction balance. It is seen from the above algorithm description that the model-based optimisation problem HMMOP can be viewed as the real optimisation problem HROP for the single loop ISOPE method presented in Section 3.4.2 (3.4.2b). This makes it possible to solve the optimisation problem HMMOP using the above mentioned single loop MC method. Since the formulation HMMOP fits in the algorithm description in Section 3.4.2 (3.4.2b), the implementation of the inner loop of the presented algorithm is omitted here.

An inspection of the above description shows that the outer loop iteration of the above algorithm is virtually the same as an implementation of the improved version of the MTS method (Lin, Han, Roberts and Wan, 1989). Therefore, this algorithm is expected to retain the convergence features of the improved MTS method in its outer loop iteration. The configuration of this algorithm is illustrated in block diagrams in Figures 3.6 and 3.7.

3.6 AUGMENTATION TECHNIQUE

3.6.1 Introduction to Variable Augmentation

It is often desirable from an implementation point of view that the convergence behaviour of an optimising control algorithm is insensitive to the gain selection, since the optimum gain is unlikely to be known before the optimum is reached. This can be achieved by introducing augmentation terms to the model-based optimisation problem (Brdyś, Abdullah and Roberts, 1987; Tatjeweski, 1985; Lin, Li, Wan and Roberts, 1989).

There are two augmentation techniques which are available for this purpose, i.e. variable augmentation and constraint augmentation. The basic idea behind these two techniques is to add penalty terms, $\tau \, \| c - v^k \|^2$ and $\tau \, \| u - u^k \|^2$, for variable augmentation, and $\tau \, \| u - HF(c, u, \alpha) \|^2$ for constraint augmentation, to the objective function of the model-based optimisation problem HMMOP to temper the controller set-point optimisation. Usually, constraint augmentation or the combination of the two is more effective than variable augmentation. It is, however, important to note that the term $\| u - HF(c, u, \alpha) \|^2$ is generally not separable and, therefore, the implementation of the constraint augmentation technique requires an extra iterative loop within the algorithm iterative structure. This is also the reason behind the use of variable augmentation techniques throughout this chapter.

3.6.2 Effect of Augmentation on Convergence and Sensitivity of ISOPE Algorithms

It has been reported that by incorporating augmentation techniques in the ISOPE framework the convergence behaviour of the ISOPE algorithms may be significantly improved (Brdyś, Abdullah and Roberts, 1987; Lin, Li, Wan and Roberts, 1989; Amini, Lin and Roberts, 1990; Lin, Amini and Roberts, 1990). An explanation for this phenomena is that the introduction of the penalty terms mentioned in the previous section has an effect of convexifying the real optimisation problem. In fact, some non-convex optimising control problems may be convexified through this approach.

It is also noticed that the introduction of the variable augmentation technique may have different convexification effects on the real problem and, hence, results in an improvement of algorithm convergence (Amini, Lin and Roberts, 1990; Lin, Amini and Roberts, 1990). This is also justified in Chapter 4 in establishing applicability conditions for the algorithm. A possible explanation for this is that the introduction of augmentation tempers the controller set-point optimisation and, therefore, results in a significant improvement for less efficient iterative algorithms (ISOPE algorithms with price coordination) but a possible set back in algorithm efficiency for the more efficient algorithms (ISOPE algorithms with modifier coordination), with a decrease in algorithm sensitivity to gain selection (Amini, Lin and Roberts, 1990; Lin, Amini and Roberts, 1990).

Apart from convexification there is another important effect brought about by the augmentation, i.e. algorithm sensitivity reduction. Generally, an augmented version of an ISOPE algorithm has a wider range for iterative gain selection and a flatter sensitivity curve. This can be seen from simulation studies included in Chapter 5.

3.7 SUMMARY

Hierarchical Integrated System Optimisation and Parameter Estimation (ISOPE) algorithms for on-line optimising control of large scale industrial processes are presented in this chapter. Important issues, such as coordination methods, feedback structure and iterative strategies are discussed. Emphasis have been given to a global feedback structure and a new coordination strategy called Modifier Coordination (MC). This gives rise to two different single loop ISOPE algorithms. These two algorithms can also be converted to double loop ISOPE algorithms by employing a double iterative strategy. The variable augmentation technique is introduced and integrated with the four presented hierarchical algorithms to improve convergence and to reduce algorithm sensitivity. The optimality and convergence of the hierarchical algorithms presented in this chapter are investigated in the following chapter.

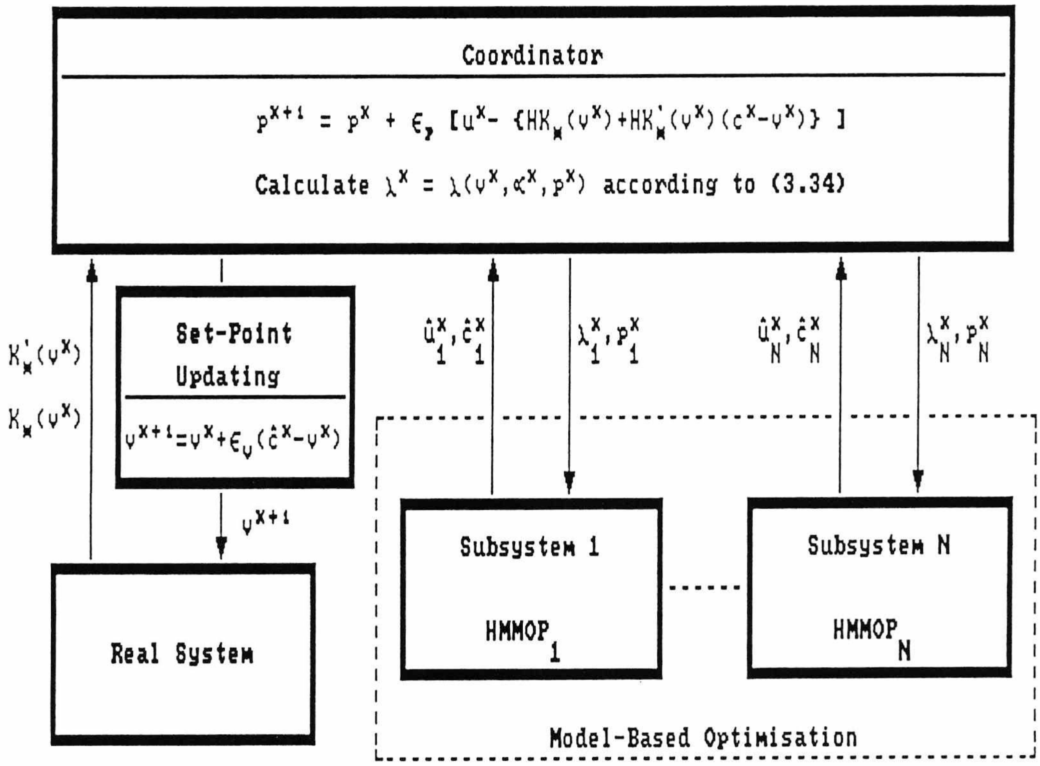

Figure 3.1 The Single iterative ISOPE method with global feedback

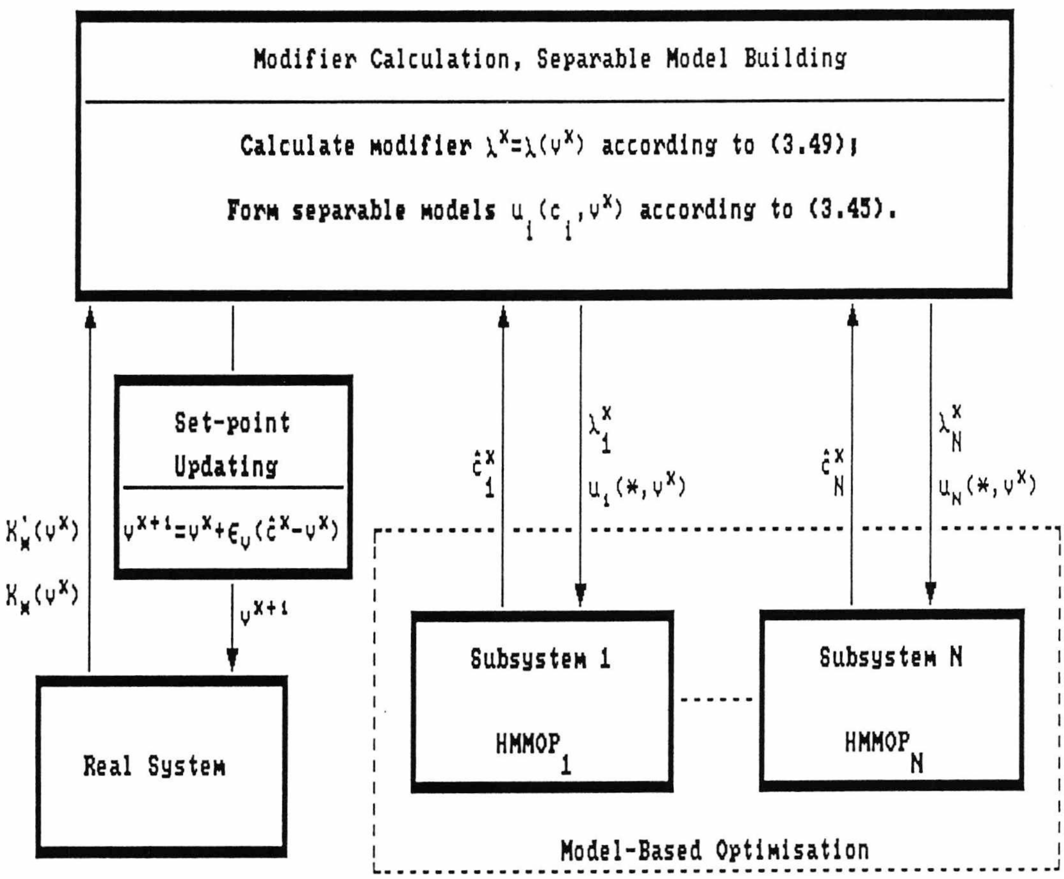

Figure 3.2 The Modifier Coordination method for problems without input dependent constraints

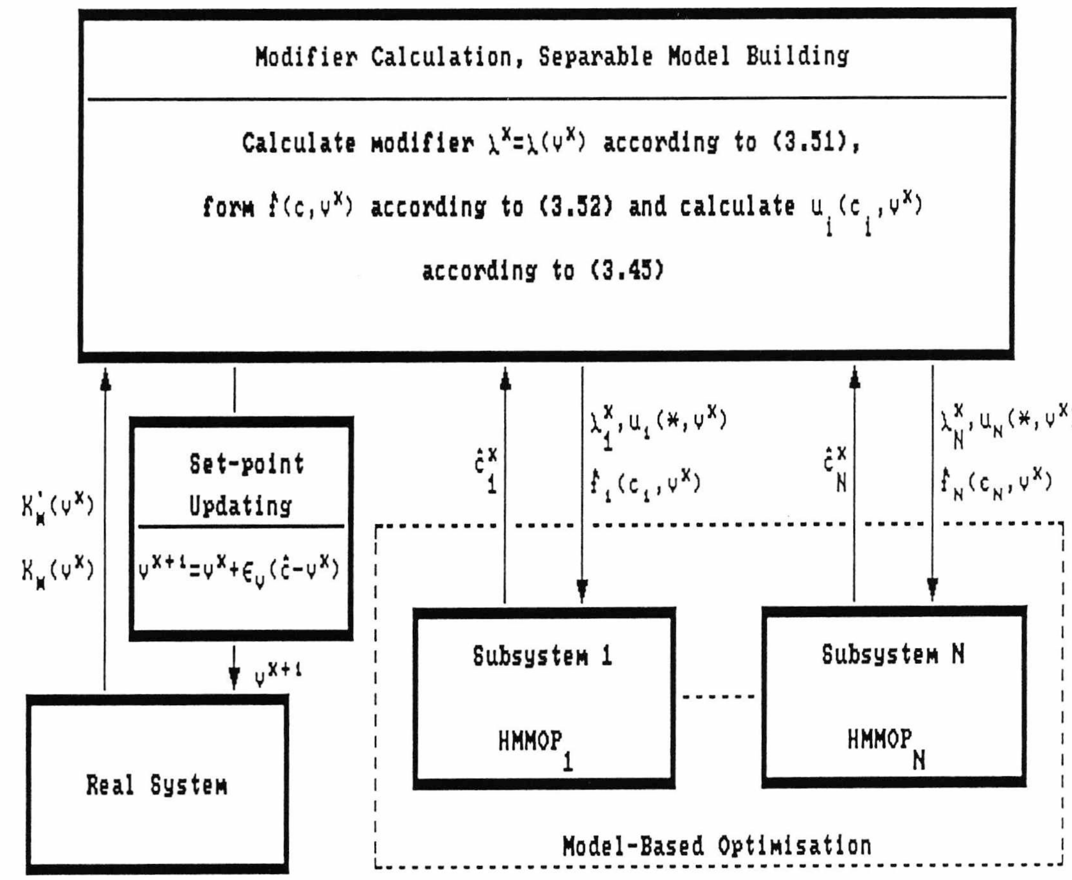

Figure 3.3 The Modifier Coordination method for problems with input dependent constraints

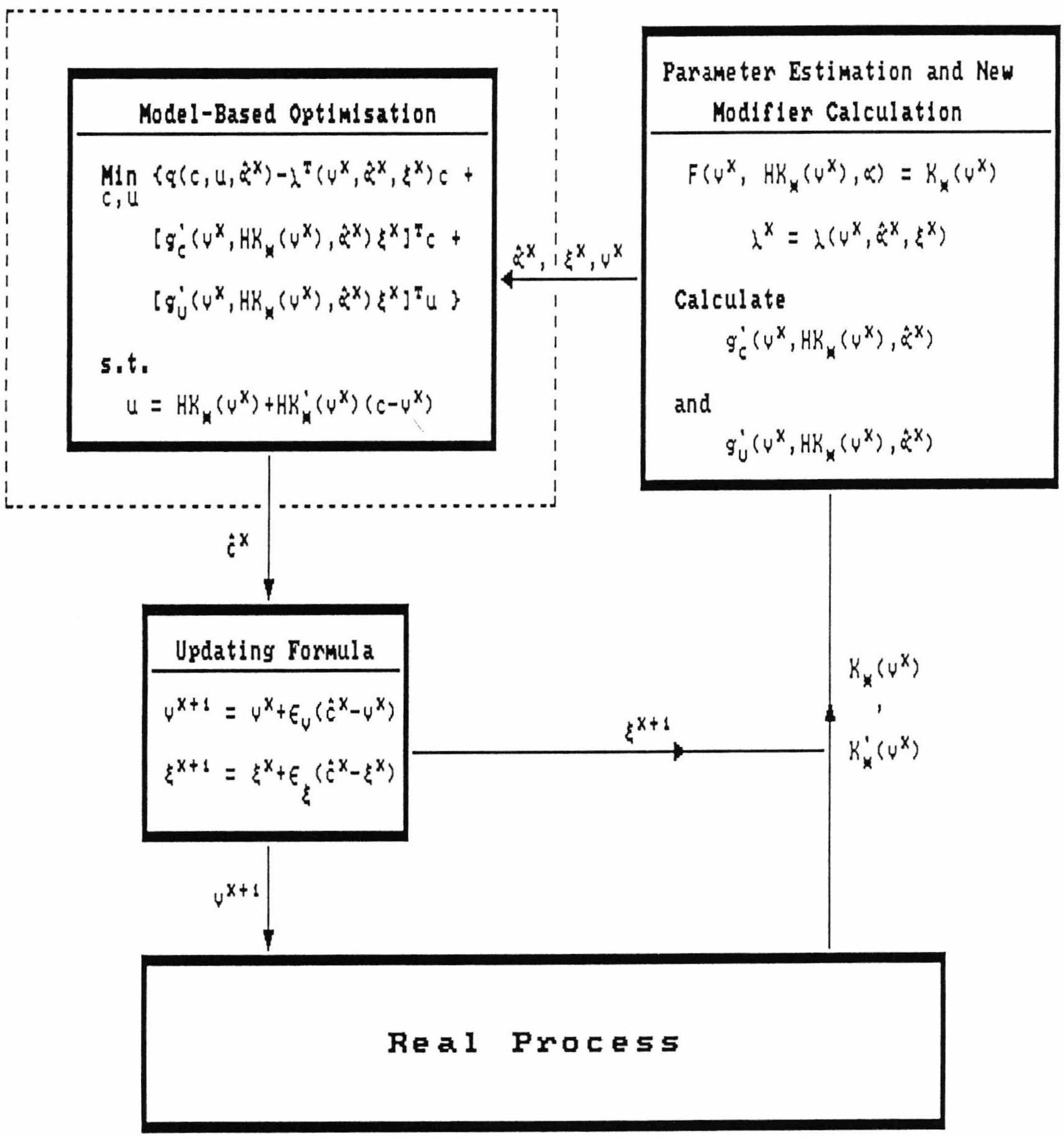

Figure 3.4 The Outer Loop Iteration for Algorithm AL4

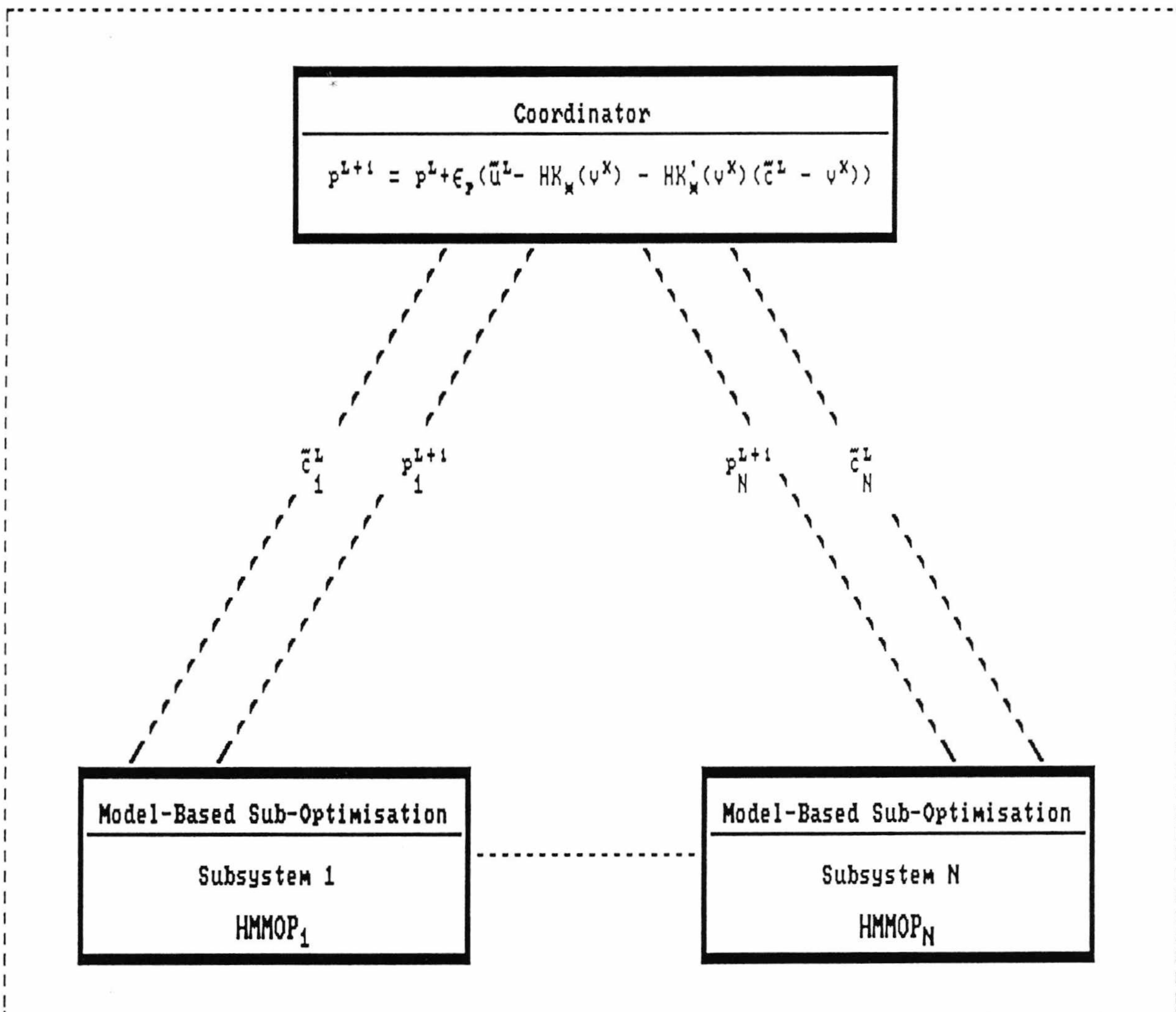

Figure 3.5 The Inner Loop Iteration for Algorithm AL4

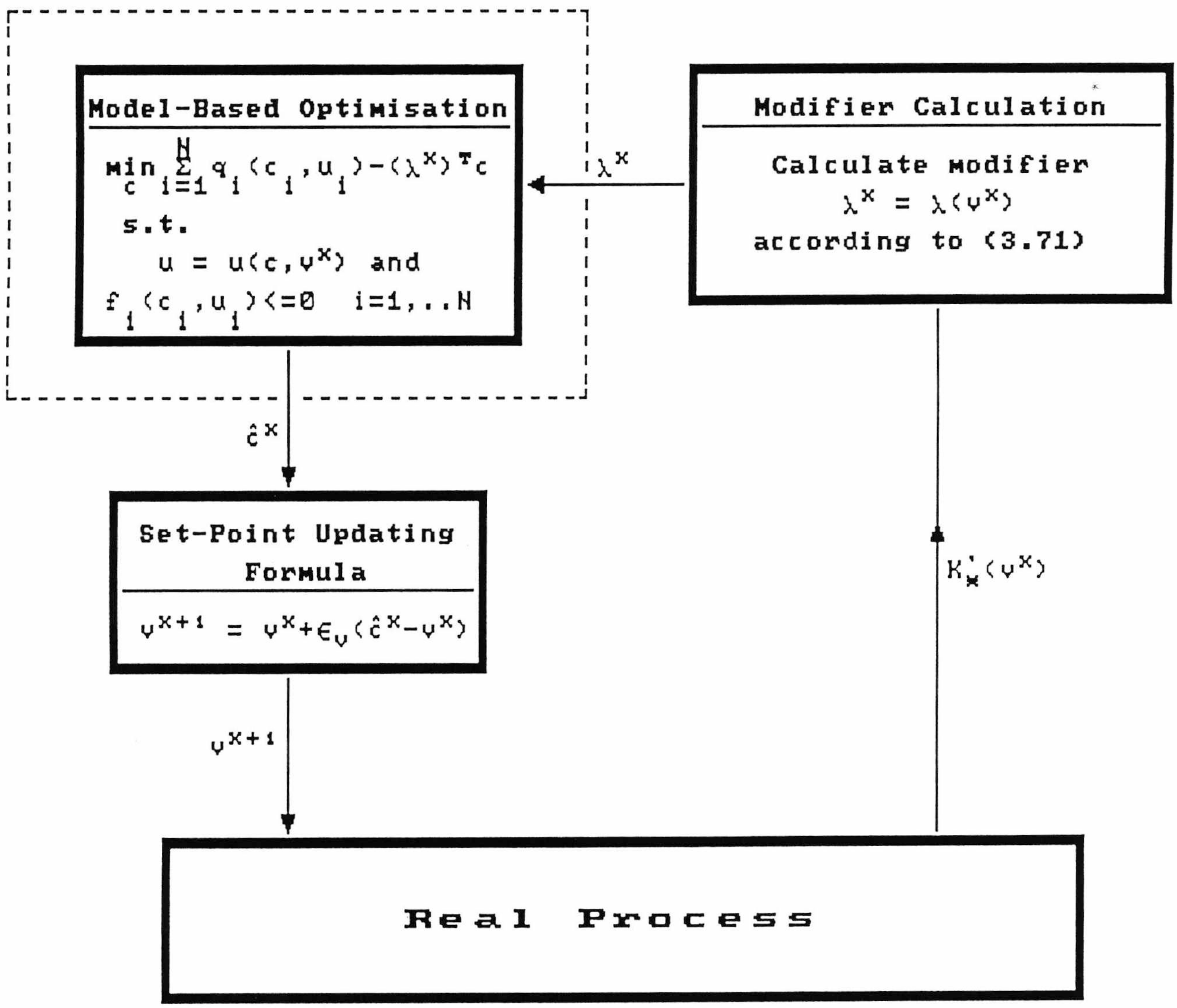

Figure 3.6 The Outer Loop for Algorithm AL5

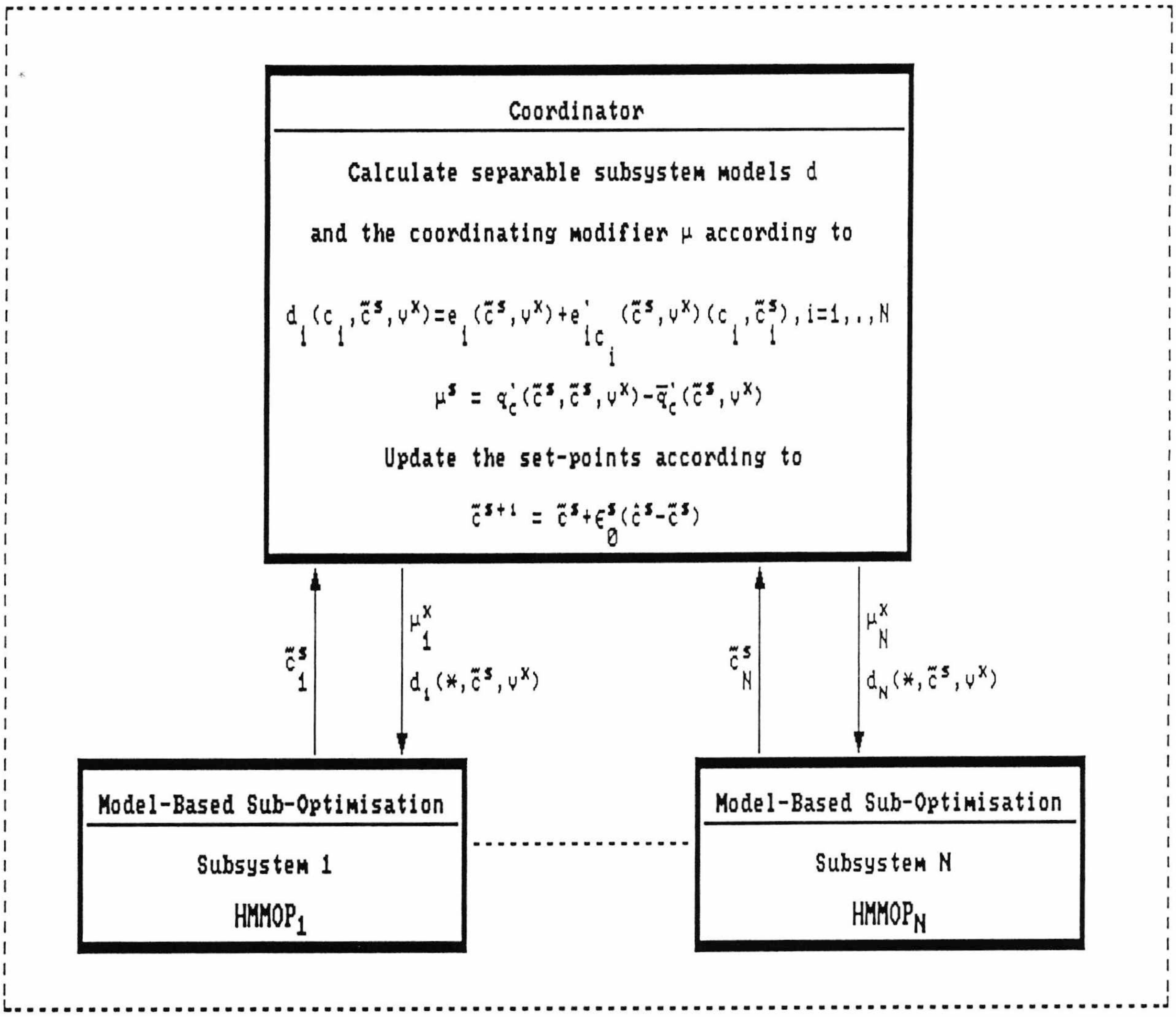

Figure 3.7 The Inner Loop Iteration for Algorithm AL5

CHAPTER 4

OPTIMALITY AND CONVERGENCE ANALYSIS FOR HIERARCHICAL ISOPE ALGORITHMS

4.1 OPTIMALITY

Optimality of the four hierarchical ISOPE methods presented in Chapter 3 (3.4.1, 3.4.2, 3.5.1 and 3.5.2) is examined in this section. Sufficient conditions for optimality of these algorithms are provided.

4.1.1 Single Loop ISOPE Algorithms

It is proved that the two single iterative ISOPE algorithms (3.4.1 and 3.4.2) are optimal in the sense that the algorithmic solution obtained satisfies the Kuhn-Tucker conditions of the real problem.

To simplify the notation we denote the single loop ISOPE algorithm with global feedback, which is presented in Section 3.4.1 (Amini, Lin and Roberts, 1990), as AL1, and the single loop ISOPE algorithms with modifier coordination, which are described in Sections 3.4.2a and 3.4.2b (Lin, Amini and Roberts, 1990), as AL2 and AL3, respectively.

In order to retain mathematical clarity, when investigating the optimality of the iterative mechanism in the presented single and double loop algorithms; we need to introduce the following definitions.

Definition 4.1

$$\Omega = \left\{ (v , p) : (v , Hk_*(v)) \in \mathbb{CU}, v = \hat{c} \ (\hat{\alpha} \ (v) \ , \lambda(v \ , \hat{\alpha}(v) \ , p) \right\} \tag{4.1}$$

$$\Omega_1 = \left\{ v : \exists \ p \text{ such that } (v , p) \in \Omega \right\} \tag{4.2}$$

and denote Ω_1^* as the optimum solution set of ROP.

The following theorem establishes optimality for the algorithm AL1.

Theorem 4.1

Assume that for every $v \in \Omega_1$, $(v , HK_*(v))$ is a regular point of the constraints $f(c , u) \leq 0$. Then, for every $(v , p) \in \Omega$, there exists an ξ such that (v , p , ξ) satisfies the Kuhn-Tucker conditions for ROP.

Proof

For every $(v , P) \in \Omega$, $(\hat{c} (\hat{\alpha} , \lambda , p) , \hat{u} (\hat{\alpha} , \lambda, p))$ is the optimum solution of MOP. Since $(\hat{c} , \hat{u})$ is a regular point of the constraint $f (\hat{c} , \hat{u}) \leq 0$. There exists an ξ such that the Kuhn-Tucker conditions for MOP, ie

$$\frac{\partial^T L}{\partial c} = \frac{\partial^T q(\hat{c} , \hat{u} , \alpha)}{\partial c} - \lambda + \frac{\partial^T f(\hat{c} , \hat{u})}{\partial c} \xi = 0 \tag{4.3}$$

$$\frac{\partial^T L}{\partial u} = \frac{\partial^T q(\hat{c} , \hat{u} , \alpha)}{\partial u} + p + \frac{\partial^T f(\hat{c} , \hat{u})}{\partial u} \xi = 0 \tag{4.4}$$

$$\xi_i \, f_i \, (\hat{c}_i , \hat{u}_i) = 0 \quad , \quad \xi_i \geq 0 \text{ and } f_i \, (\hat{c}_i , \hat{u}_i) \leq 0, \; i=1,...,N \tag{4.5}$$

are satisfied.

Substituting (3.24) into (4.3) and (4.5) results in equation (3.29).

Since $\hat{c} = v$ and $\hat{u} = HK_*(v)$, it is clear that (3.29) is equivalent to (3.30) which is precisely the Kuhn-Tucker conditions for ROP.

QED

Before establishing the optimality for the algorithm AL2, we need the following definition :

Definition 4.2

Define :

$$\Omega = \left\{ v : v \in \mathbb{C}, \hat{c}(v) = v \right\} \tag{4.6}$$

where $\hat{c}(v)$ is the optimum of the model-based optimisation problem HMMOP1, defined in Section 3.4.2a, for given v. Denote Ω^* as the optimum solution set of ROP.

Theorem 4.2

Let assumption 3.1 be satisfied and assume that every $v \in \Omega$ is a regular point of the constraints $f(c) \leq 0$. Then, for every $v \in \Omega$, there exists an ξ such that (v,ξ) satisfies the Kuhn-Tucker conditions for ROP.

Proof

For every $v \in \Omega$, $\hat{c}(v)$ is a regular point of the constraints $f(c) \leq 0$. Hence, there exists an ξ such that the Kuhn-Tucker conditions for MOP1, i.e.

$$q'_c(\hat{c},v) - \lambda^T + \xi^T f'(\hat{c}) = 0 \tag{4.7}$$

$$\xi_i f_i(\hat{c}_i) = 0,\ \xi_i \geq 0 \text{ and } f_i(\hat{c}_i) \leq 0,\ \ i = 1, ..., m \tag{4.8}$$

are satisfied.

Substituting (3.49) into (4.7) and v for $\hat{c}$ in both (4.7) and (4.8) (since $\hat{c}=v$ for every $v \in \Omega$) yields :

$$q'_*(v) + \xi^T f'(v) = 0 \tag{4.9}$$

$$\xi_i f_i(v_i) = 0,\ \xi_i \geq 0 \text{ and } f_i(v_i) \leq 0,\ i = 1,...,m \tag{4.10}$$

Expressions (4.9) and (4.10) constitute the Kuhn-Tucker conditions for ROP.

QED

In order to investigate the optimality of the algorithm AL3, first we need to convert the real optimisation problem HROP into an equivalent form. It is well known that the real optimisation problem HROP is equivalent to the following problem :

$$\min_{c,u,y,w} \left\{ Q(c, u, y) + \rho \, || \, w \, ||^2 \right\}$$

(HROPP)

s.t.

$$y = K_*(c)$$

$$u = Hy$$

$$f(c,u) + w \leq 0$$

in the sense that the optimum solution set of this problem can be made arbitrarily close to that of HROP. The following definition is needed in the investigation of the optimality of algorithm AL3.

Definition 4.3

Define :

$$f_*(v) = f(v, HK_*(v)) \tag{4.11}$$

$$\Omega = \left\{ v : v \in \mathbb{C},\ \hat{c}(v) = v \right\} \tag{4.12}$$

where $\hat{c}(v)$ is the optimum of the model-based optimisation problem HMMOP2 for given v as described in the algorithm description of AL3. Denote Ω^* as the optimum solution set of ROP.

Theorem 4.3

Let assumption 3.1 be satisfied and assume that every $v \in \Omega$ is a regular point of the constraints $f_*(v) \leq 0$ and $\hat{f}(v,v^*) \leq 0$ for any given $v^* \in \Omega^*$. Then, for every $v \in \Omega$, there exists an ξ such that (v,ξ) satisfies the Kuhn-Tucker conditions for HROPP.

Proof

For every $v \in \Omega$, $(\hat{c}(v),\hat{w}(v))$ is the optimum solution of HMMOP2. Since $v \in \Omega$, we have $\hat{c}(v)=v$. Hence, for any $v \in \Omega$, $\hat{c}(v)$ is a regular point of the constraint $\hat{f}(c,v^*) + w \leq 0$. Therefore, there exists an ξ such that :

$$q_c'(v,v) - \lambda(v) + \xi \hat{f}_c'(v,v) = 0 \tag{4.13}$$

$$\xi + 2\rho w^T = 0 \tag{4.14}$$

$$\xi_i \Big[f_*(v) + w \Big]_i = 0 \text{ with } \xi_i \geq 0, \Big[f_*(v) + w \Big]_i \leq 0, \ i = 1, \dots ,m \tag{4.15}$$

It is not difficult to derive from (4.14) — (4.15) that

$$q_c'(v,v) - \lambda(v) + 2\rho \hat{f}(v,v)\ \sigma(v,v)\ \hat{f}_c'(v,v) = 0 \tag{4.16}$$

In fact, if $f_{*i}(v) < 0$ then $w_i = 0$ for otherwise, a contradiction arises from (4.14). We, therefore, conclude that $f_{*i}(v) < 0$ when $\xi_i > 0$ and $f_{*i}(v) \geq 0$ when $\xi_i = 0$. Substituting (3.55) into (4.16) yields :

$$q_*'(v) + 2\rho\ f_*(v)\ \sigma(v,v)\ f_*'(v) = 0 \tag{4.17}$$

Using the same approach as that for deriving (4.16) from (4.13) — (4.15), we can derive from (4.17) that :

$$q_*'(v) + \xi f_*'(v) = 0 \text{ and } \xi + 2\rho w = 0 \tag{4.18}$$

$$\xi_i \Big[f_*(v) + w \Big]_i = 0, \ i = 1,\dots,m \tag{4.19}$$

with

$$\xi_i \geq 0, \Big[f_*(v) + w \Big]_i \leq 0, \ i = 1,\dots,m \tag{4.20}$$

Expressions (4.18) — (4.20) constitute the Kuhn-Tucker conditions for the equivalent real optimisation problem HROPP.

QED

4.1.2 Double Loop ISOPE Algorithms

For simplicity of analysis, we denote the double loop ISOPE algorithm with price coordination, which is presented in Section 3.5.1 (Lin, Hendawy and Roberts, 1988b), as AL4, and the double loop ISOPE algorithm with modifier coordination, which is described in Sections 3.5.2

(Lin, Roberts, Wang and Wan, 1990), as AL5.

4.1.2a Optimality of Algorithm AL4

Define

$$\Omega = \left\{ (v) : v \in \mathbb{C},\ v = \hat{c}(v\ ,\ \hat{\alpha}) \right\} \tag{4.21}$$

and denote Ω^* as the optimal solution set of HROP. In the following, η is denoted as the Lagrange multiplier vector associated with the constraints $f(\cdot\ ,\ \cdot)$.

Theorem 4.4

Let Assumption 3.1 be satisfied, and assume that every $v \in \Omega$ is a regular point of the constraints $f(v\ ,\ HK_*(v)) \leq 0$. Then, for every $v \in \Omega$ there exists a η such that $(v\ ,\ \eta)$ satisfies Kuhn-Tucker necessary optimality conditions of HROP.

A proof of this theorem can be constructed following the same approach as that presented by Lin, Hendawy and Roberts (1988b) and, hence, is omitted here.

4.1.2b Optimality of Algorithm AL5

It is seen from the derivation of AL5 (Section 3.5.2) that the outer loop iteration of the algorithm can be viewed as the standard MTS solution process and the inner loop iteration is designed only for the purpose of model-based optimisation. Therefore, optimality of algorithm AL5 can be investigated without invoking the details of inner loop iteration. This means that optimality of AL5 is covered by that of the standard MTS method for centralised optimisation (Theorem 2.1 in Chapter 2) and, therefore, there is no need to repeat the same analysis.

4.2 CONVERGENCE

Convergence analysis is carried out for the four hierarchical ISOPE methods presented in Chapter 3 (3.4.1, 3.4.2, 3.5.1 and 3.5.2). Sufficient conditions for global or local convergence of these algorithms with and without augmentation are provided in this section.

4.2.1 Single Loop ISOPE Algorithms

It is proved that the presented single iterative ISOPE algorithms (AL1 and AL2 and AL3) are locally convergent and that the introduction of augmentation improves the convergence of an algorithm in the sense that the convergence conditions of the algorithm are relaxed.

4.2.1a Convergence Conditions for Algorithm AL1

The following definitions are introduced to assist the proof of the convergence theorem.

Definition 4.4

Define :

$$\mathbb{A} = \Big\{ \alpha : (v, HK_*(v)) \in \mathbb{CU},\ F (v, HK_*(v) , \alpha) = K_*(v) \Big\} \tag{4.22}$$

$$s_1 (\overline{v} , r) = \Big\{ (u,v) : \Big\| v - \overline{v} \Big\|^2 + \Big\| u - HK_*(\overline{v}) \Big\|^2 \leq r^2 \Big\} \tag{4.23}$$

$$s_2 (\overline{v} , r) = \Big\{ v : \Big\| v - \overline{v} \Big\|^2 + \Big\| HK_*(v) - HK_*(\overline{v}) \Big\|^2 \leq r^2 \Big\} \tag{4.24}$$

$$s_3 (\overline{p} , r) = \Big\{ p : \Big\| p - \overline{p} \Big\|^2 \leq r^2 \Big\} \tag{4.25}$$

$$\mathbb{B} = \Big\{ \alpha :\ F (v, HK_*(v) , \alpha) = K_*(v),\ (v, HK_*(v)) \in \mathbb{CU} \cap s_1(\overline{v} , r) \Big\} \tag{4.26}$$

Definition 4.5

Define :

$$e_*(v) = HK_*(\overline{v}) - HK_*(v^k) \text{ where } \overline{v} \in \Omega_1^* \tag{4.27}$$

$$L_*(v\ ,\ p) = q_*(v) + p^T e_*(v) \tag{4.28}$$

$$h(c\ ,\ v) = q'_u\ (c,\ HK_*(c)\ ,\ \alpha(v)) \tag{4.29}$$

Theorem 4.5

Assume (v, $HK_*(v)$) $\in \mathbb{CU}$ is a regular point of the constraints $f(c\ ,\ u) \leq 0$. Assume :

(1) $K_*(\ \cdot\ )$, $f(\ \cdot\ ,\ \cdot\ )$ and $q(\ \cdot\ ,\ \cdot\ ,\ \alpha)$, $\alpha \in \mathbb{A}$ are twice continuously Frèchet differentiable.

(2) The constraint set $\mathbb{CU}$ is compact.

(3) There exists an $r > 0$ such that

(a) For every $\alpha \in \mathbb{B}$, $q'_c\ (\cdot\ ,\ \cdot\ ,\ \alpha)$ and $q'_u\ (\cdot\ ,\ \cdot\ ,\ \alpha)$ are uniformly monotone on $\overline{\mathbb{CU}} \cap s_1(\ \overline{v},\ r)$ with constant $a_1(\alpha\ ,\ r) > 0$, where $\overline{\mathbb{CU}}$ is the closure of $\mathbb{CU}$ and $(\overline{v}) \in \Omega_1^*$.

(b) For every $\alpha \in \mathbb{B}$, $\hat{q}'_v\ (\ \cdot\ ,\ \alpha)$ is uniformly monotone on $s_2(\ \overline{v},\ r\ )$ with constant $a_2(\alpha\ ,\ r) > 0$.

(c) For every $(c\ ,\ u) \in \mathbb{CU} \cap s_1(\ \overline{v}\ ,\ r\ )$, $h'_v\ (c\ ,\ v)$ is twice continuously differentiable.

(d) For every $p \in s_3(\ \overline{p}\ ,\ r\ )$, $L'_*\ (\ \cdot\ ,\ p)$ is uniformly monotone on $s_2(\ \overline{v}\ ,\ r\ )$ with constant $a_*(r)$.

(4) The real system and its chosen model satisfy :

$$2a_*(r) - \frac{1}{2}\ A_*(r) > \overline{A}_2(r) + \frac{B_1(r)}{2\underline{a}_1(r)} + \frac{B_2(r)}{2\underline{a}_1(r)} \tag{4.30}$$

$$a_2(r) > \frac{3}{4} A_*(r) \tag{4.31}$$

$$\text{where } \overline{A}_2(r) = \sup_{\alpha \in \mathbb{B}} A_2(\alpha, r) \geq \inf_{\alpha \in \mathbb{B}} a_2(\alpha, r) = \underline{a}_2(r) > 0 \tag{4.32}$$

$$\underline{a}_1(r) = \inf_{\alpha \in \mathbb{B}} a_1(\alpha, r) > 0 \tag{4.33}$$

$$B_1(r) = \sup_{v \in S_2(\bar{v}, r),\ c \in S_2(\bar{v}, r)} \left\| h'_v(c, v) \right\|, \ \bar{v} \in \Omega_1^* \tag{4.34}$$

$$B_2(r) = \sup_{v \in s_2(\bar{v}, r)} \left\| HK'_*(v) \right\|, \ \bar{v} \in \Omega_1^* \tag{4.35}$$

$A_2(\alpha, r)$ and $A_*(r)$ are Lipschitz constants of $\hat{q}'_v(\cdot, \alpha)$ and $L'_*(\cdot, p)$ on $s_2(\bar{v}, r)$, $\bar{v} \in \Omega_1^*$, respectively.

Then, there exists $0 < \epsilon_1 \leq 1$ and $\epsilon_2 > 0$ such that for every $\epsilon_v \in (0, \epsilon_1)$ and $\epsilon_p \in (0, \epsilon_2)$, there exists an r such that :

for any $(v^0, HK_*(v^0)) \in \mathbb{C}U \cap s_1(\bar{v}, r)$, $\bar{v} \in \Omega_1^*$ and $p^0 \in s_3(\bar{p}, r)$, the sequences $\{v^k\}$, $\{p^k\}$ generated by the algorithm satisfy

$$\lim_{k \to \infty} v^k = \bar{v} \quad \text{and} \quad \lim_{k \to \infty} p_k = \bar{p} \tag{4.36}$$

Proof

Assume $v^k \in s_2(\bar{v}, r)$ and $p^k \in s_3(\bar{p}, r)$. Then, from the definition of $(\hat{c}^k, \hat{u}^k)$, the following inequality holds (Luenberger, 1984) :

$$\Big[q'_c(\hat{c}^k, \hat{u}^k, \alpha^k) - q'_c(v^k, HK_*(v^k), \alpha^k) - q'_u(v^k, HK_*(v^k), \alpha^k) HK'_*(v^k) + q'_*(v^k)$$

$$- p_k^T HK'_*(v^k) \Big] (\bar{v} - \hat{c}^k) + \Big[q'_u(\hat{c}^k, \hat{u}^k, \alpha^k) + p_k^T \Big] (HK_*(\bar{v}) - \hat{u}^k) \geq 0 \tag{4.37}$$

Since at the optimum $\hat{c} = \bar{v}$ and $\hat{u} = HK_*(\bar{v})$, equations (3.27) and (3.28) produce (Luenberger, 1984) :

$$\Big[- q_u'(\bar{v}, HK_*(\bar{v}), \bar{\alpha})\, HK_*'(\bar{v}) + q_*'(\bar{v}) - \bar{p}\, HK_*'(\bar{v}) \Big] (\hat{c}^k - \bar{v})$$

$$+ \Big[q_u'(\bar{v}, HK_*(\bar{v}), \bar{\alpha}) + \bar{p}^T \Big] (\hat{u}^k - HK_*(\bar{v})) \geq 0 \qquad (4.38)$$

Adding (4.38) to (4.37) yields :

$$q_c'(\hat{c}^k, \hat{u}^k, \alpha^k)(\bar{v} - \hat{c}^k) + q_u'(\hat{c}^k, \hat{u}^k, \alpha^k)(HK_*(\bar{v}) - \hat{u}^k)$$

$$+ q_c'(\bar{v}, HK_*(\bar{v}), \alpha^k)(\hat{c}^k - \bar{v}) + q_u'(\bar{v}, HK_*(\bar{v}), \alpha^k)(\hat{u}^k - HK_*(\bar{v}))$$

$$+ q_c'(v^k, HK_*(v^k), \alpha^k)(\hat{c}^k - \bar{v}) + q_u'(v^k, HK_*(v^k), \alpha^k)\, HK_*'(v^k)(\hat{c}^k - \bar{v})$$

$$+ q_c'(\bar{v}, HK_*(\bar{v}), \alpha^k)(\bar{v} - \hat{c}^k) + q_u'(\bar{v}, HK_*(\bar{v}), \alpha^k)\, HK_*'(\bar{v})(\bar{v} - \hat{c}^k)$$

$$+ q_u'(\bar{v}, HK_*(\bar{v}), \alpha^k)(HK_*(\bar{v}) + HK_*'(\bar{v})(\hat{c}^k - \bar{v}) - \hat{u}^k)$$

$$- q_u'(\bar{v}, HK_*(\bar{v}), \bar{\alpha})(HK_*(\bar{v}) + HK_*'(\bar{v})(\hat{c}^k - \bar{v}) - \hat{u}^k)$$

$$+ q_*'(v^k)(\bar{v} - \hat{c}^k) + q_*'(\bar{v})(\hat{c}^k - \bar{v}) + p_k^T \Big[HK_*(\bar{v}) + HK_*'(v^k)(\hat{c}^k - \bar{v}) - \hat{u}^k \Big]$$

$$+ \bar{p}^T \Big[\hat{u}^k - HK_*(\bar{v}) - HK_*'(\bar{v})(\hat{c}^k - \bar{v}) \Big] \geq 0 \qquad (4.39)$$

According to assumptions (3), it is not difficult to derive (Lin, Chen and Roberts, 1988) :

$$q_c'(\hat{c}^k, \hat{u}^k, \alpha^k)(\bar{v} - \hat{c}^k) + q_u'(\hat{c}^k, \hat{u}^k, \alpha^k)(HK_*(\bar{v}) - \hat{u}^k)$$

$$+ q_c'(\bar{v}, HK_*(\bar{v}), \alpha^k)(\hat{c}^k - \bar{v}) + q_u'(\bar{v}, HK_*(\bar{v}), \alpha^k)(\hat{u}^k - HK_*(\bar{v}))$$

$$\leq - \underline{a}_1(r) \Big[\| \hat{c}^k - \bar{v} \|^2 + \| HK_*(\bar{v}) - \hat{u}^k \|^2 \Big] \qquad (4.40)$$

and

$$q_c'(v^k, HK_*(v^k), \alpha^k)(\hat{c}^k - \bar{v}) + q_u'(v^k, HK_*(v^k), \alpha^k)\, HK_*'(v^k)(\hat{c}^k - \bar{v})$$

$$+ q_c'(\bar{v}, HK_*(\bar{v}), \alpha^k)(\bar{v} - \hat{c}^k) + q_u'(\bar{v}, HK_*(\bar{v}), \alpha^k)\, HK_*'(\bar{v})(\bar{v} - \hat{c}^k)$$

$$\leq -\tfrac{1}{2}\, \underline{a}_2(r) \| \hat{c}^k - v^k \|^2 + \tfrac{1}{2}\, \bar{A}_2(r) \| v^k - \bar{v} \|^2 + \tfrac{1}{2}\, \bar{A}_2(r) \| \hat{c}^k - \bar{v} \|^2 \qquad (4.41)$$

$$q'_u\,(\bar{v},\, HK_*(\bar{v}),\, \alpha^k\,)\,(\,HK_*(\bar{v}) + HK'_*(\bar{v})\,(\hat{c}^k - \bar{v}) - \hat{u}^k\,)$$

$$-\; q'_u\,(\bar{v},\, HK_*(\bar{v}),\, \bar{\alpha}\,)\,(\,HK_*(\bar{v}) + HK'_*(\bar{v})\,(\hat{c}^k - \bar{v}) - \hat{u}^k\,)$$

$$= \Big[\, h(\bar{v}\,,\, v^k) - h(\bar{v}\,,\,\bar{v})\,\Big]\,\Big[\,(\,HK_*(\bar{v}) - \hat{u}^k) + HK'_*(\bar{v})\,(\hat{c}^k - \bar{v})\,\Big]$$

$$\le B_1(r)\,\Big|\Big|\,\hat{u}^k - HK_*(\bar{v})\,\Big|\Big|\;\Big|\Big|\,v^k - \bar{v}\,\Big|\Big| + B_1(r)\,B_2(r)\,\Big|\Big|\,v^k - \bar{v}\,\Big|\Big|\;\Big|\Big|\,\hat{c}^k - \bar{v}\,\Big|\Big|$$

$$+\, O(\,\Big|\Big|\,HK_*(\bar{v}) - \hat{u}^k\,\Big|\Big|\,)\,\Big|\Big|\,v^k - \bar{v}\,\Big|\Big|^2 + O(\,\Big|\Big|\,\hat{c}^k - \bar{v}\,\Big|\Big|\,)\;\Big|\Big|\,v^k - \bar{v}\,\Big|\Big|^2$$

$$\le (\,\underline{a}_1(r) - \tfrac{3}{2}\,\epsilon_p\,)\,\Big|\Big|\,\hat{u}^k - HK_*(\bar{v})\,\Big|\Big|^2 + \frac{B_1^2(r)}{4\,(\,\underline{a}_1(r) - \frac{3}{2}\,\epsilon_p\,)}\,\Big|\Big|\,v^k - \bar{v}\,\Big|\Big|^2$$

$$+ \Big[\,\underline{a}_1(r) - \tfrac{1}{2}\,\bar{A}_2(r)\,\Big]\,\Big|\Big|\,\hat{c}^k - \bar{v}\,\Big|\Big|^2 + \frac{B_1^2(r)\,B_2^2(r)}{2\,(2\underline{a}_1(r) - \bar{A}_2(r)\,)}\,\Big|\Big|\,v^k - \bar{v}\,\Big|\Big|^2$$

$$+\, O(\,\Big|\Big|\,HK_*(\bar{v}) - \hat{u}^k\,\Big|\Big|\,)\,\Big|\Big|\,v^k - \bar{v}\,\Big|\Big|^2 + O(\,\Big|\Big|\,\hat{c}^k - \bar{v}\,\Big|\Big|\,)\;\Big|\Big|\,v^k - \bar{v}\,\Big|\Big|^2 \tag{4.42}$$

where $O(x^r)$ means all the terms x, of order r and more.

$$q'_*(v^k)\,(\bar{v} - \hat{c}^k) + q'_*(\bar{v})\,(\,\hat{c}^k - \bar{v}) + p_k^T\,\Big[\,HK_*(\bar{v}) + HK'_*(v^k)\,(\hat{c}^k - \bar{v}) - \hat{u}^k\,\Big]$$

$$+\, \bar{p}^T\,\Big[\,\hat{u}^k - HK_*(\bar{v}) - HK'_*(\bar{v})\,(\hat{c}^k - \bar{v})\,\Big] = L'_{*v}\,(v^k, p_k)\,(\bar{v} - \hat{c}^k) + L'_{*v}\,(\bar{v}, \bar{p})\,(\hat{c}^k - \bar{v})$$

$$+\, (\,\bar{p} - p_k\,)^T\,(\hat{u}^k - HK_*(\bar{v})) \tag{4.43}$$

$$L'_{*v}\,(\bar{v}, \bar{p})\,(\hat{c}^k - \bar{v}\,) = L'_{*v}\,(\bar{v}, \bar{p}\,)\,(\hat{c}^k - v^{k+1}) + L'_{*v}\,(\bar{v}, \bar{p})\,(v^{k+1} - \bar{v})$$

$$\le L_*\,(v^{k+1}, \bar{p}) - L_*\,(\bar{v}, \bar{p}\,) - \tfrac{1}{2}\,a_*(r)\,\Big|\Big|\,v^{k+1} - \bar{v}\,\Big|\Big|^2 + L'_{*v}\,(\bar{v}, \bar{p}\,)\,(\hat{c}^k - v^{k+1})$$

$$=\; L_*\,(v^{k+1}, \bar{p}) - \bar{p}^T e_*(v^k) + \bar{p}^T e'_*(v^k)\,(v^k - v^{k+1}) - q_*(\bar{v})$$

$$+\, \bar{p}^T\Big[\,e_*(v^k) + e'_*(v^k)\,(\hat{c}^k - v^k)\,\Big] + \bar{p}^T e'_*(v^k)\,(v^{k+1} - \hat{c}^k)$$

$$-\, \tfrac{1}{2}\,a_*(r)\,\Big|\Big|\,v^{k+1} - \bar{v}\,\Big|\Big|^2 + L'_{*v}\,(\bar{v}, \bar{p}\,)\,(\hat{c}^k - v^{k+1}) \tag{4.44}$$

$$L'_{*v}(v^k,p_k)(\overline{v} - \hat{c}^k) = L'_{*v}(v^k,p_k)(\overline{v} - v^k) + q'_*(v^k)(v^k - v^{k+1})$$

$$+ q'_*(v^k)(v^{k+1} - \hat{c}^k) + p_k^T e'_*(v^k)(v^k - \hat{c}^k)$$

$$\leq L_*(\overline{v},p_k) - L_*(v^k,p^k) - \frac{1}{2} a_*(r) \left|\left| v^k - \overline{v} \right|\right|^2 + q'_*(v^k)(v^k - v^{k+1})$$

$$+ q'_*(v^k)(v^{k+1} - \hat{c}^k) + p_k^T e'_*(v^k)(v^k - \hat{c}^k)$$

$$= q_*(\overline{v}) - q_*(v^k) - p_k^T \Big[e_*(v^k) + e'_*(v^k)(\hat{c}^k - v^k) \Big] + q'_*(v^k)(v^k - v^{k+1})$$

$$+ q'_*(v^k)(v^{k+1} - \hat{c}^k) - \frac{1}{2} a_*(r) \left|\left| v^k - \overline{v} \right|\right|^2 \tag{4.45}$$

Substituting (4.44) and (4.45) into the right hand side of (4.43) yields :

$$L'_{*v}(\overline{v},\overline{p})(\hat{c}^k - \overline{v}) + L'_{*v}(v^k,p_k)(\overline{v} - \hat{c}^k) + (\overline{p} - p_k)^T (\hat{u}^k - HK_*(\overline{v}))$$

$$= L_*(v^{k+1},\overline{p}) - L_*(v^k,\overline{p}) + L'_{*v}(v^k,\overline{p})(v^k - v^{k+1})$$

$$(\overline{p} - p_k)^T \Big[\hat{u}^k - HK_*(v^k) - HK'_*(v^k)(\hat{c}^k - v^k) \Big] + \Big[L'_{*v}(\overline{v},\overline{p}) - L'_{*v}(v^k,\overline{p}) \Big]$$

$$(\hat{c}^k - v^{k+1}) - \frac{1}{2} a_*(r) \left|\left| v^k - \overline{v} \right|\right|^2 - \frac{1}{2} a_*(r) \left|\left| v^{k+1} - \overline{v} \right|\right|^2 \tag{4.46}$$

Applying (4.40) — (4.42) and (4.46) to (4.39) produces :

$$- \frac{1}{2} \underline{a}_2(r) \left|\left| \hat{c}^k - v^k \right|\right|^2 + \frac{1}{2} \Big(D(r) - a_*(r) \Big) \left|\left| v^k - \overline{v} \right|\right|^2$$

$$+ O\left(\left|\left| HK_*(\overline{v}) - \hat{u}^k \right|\right| \right) \left|\left| v^k - \overline{v} \right|\right|^2 + O\left(\left|\left| \hat{c}^k - \overline{v} \right|\right| \right) \left|\left| v^k - \overline{v} \right|\right|^2$$

$$- \frac{1}{2} a_*(r) \left|\left| v^k - \overline{v} \right|\right|^2 - \frac{1}{2} a_*(r) \left|\left| v^{k+1} - \overline{v} \right|\right|^2 + L_*(v^{k+1},\overline{p}) - L_*(v^k,\overline{p})$$

$$+ L'_{*v}(v^k,\overline{p})(v^k - v^{k+1}) + \Big[L'_*(\overline{v},\overline{p}) - L'_*(v^k,\overline{p}) \Big] (\hat{c}^k - v^{k+1})$$

$$- \frac{3}{2} \epsilon_p \left|\left| \hat{u}^k - HK_*(\overline{v}) \right|\right|^2 + (\overline{p} - p_k) \Big[\hat{u}^k - HK_*(v^k) - HK'_*(v^k)(\hat{c}^k - v^k) \Big] \geq 0$$

$$\tag{4.47}$$

and

$$D(r) = \overline{A}_2(r) + \frac{B_1^2(r)}{2\left[\underline{a}_1(r) - \frac{3}{2}\epsilon_p\right]} + \frac{B_1^2(r)\, B_2^2(r)}{\left[2\underline{a}_1(r) - \overline{A}_2(r)\right]} \tag{4.48}$$

According to (3.36), it is not difficult to obtain :

$$\frac{1}{2}\left|\left| p_{k+1} - \overline{p} \right|\right|^2 = \frac{1}{2}\left|\left| p_k - \overline{p} \right|\right|^2 + \epsilon_p (p_k - \overline{p})^T$$

$$\times \left[\hat{u}^k - HK_*(v^k) - HK_*'(v^k)\ (\hat{c}^k - v^k) \right]$$

$$+ \frac{1}{2}\epsilon_p^2 \left|\left| \hat{u}^k - HK_*(\overline{v}) + HK_*(\overline{v}) - HK_*(v^k) - HK_*'(v^k)\ (\hat{c}^k - v^k) \right|\right|^2$$

$$\leq \frac{1}{2}\left|\left| p_k - \overline{p} \right|\right|^2 + \frac{3}{2}\epsilon_p^2 \left[\left|\left| \hat{u}^k - HK_*(\overline{v}) \right|\right|^2 + B_2^2(r) \left|\left| v^k - \overline{v} \right|\right|^2 \right.$$

$$\left. + B_2^2(r) \left|\left| \hat{c}^k - v^k \right|\right|^2 + o(\left|\left| v^k - \overline{v} \right|\right|^2) \right]$$

$$+ \epsilon_p (p_k - \overline{p})^T \left[\hat{u}^k - HK_*(v^k) - HK_*'(v^k)\ (\hat{c}^k - v^k) \right] \tag{4.49}$$

where o(r) means infinitesimal of r.

Applying (4.47) to (4.49) yields :

$$\frac{1}{\epsilon_p}\left|\left| p_k - \overline{p} \right|\right|^2 - \frac{1}{\epsilon_p}\left|\left| p_{k+1} - p_k \right|\right|^2 + a_*(r) \left|\left| v^k - \overline{v} \right|\right|^2 - a_*(r) \left|\left| v^{k+1} - \overline{v} \right|\right|^2 \geq$$

$$\left[2a_*(r) - (1 - \epsilon_v)\, A_*(r) - D(r) - 3\epsilon_p\, B_2^2(r) + \frac{o\left|\left| v^k - \overline{v} \right|\right|^2}{\left|\left| v^k - \overline{v} \right|\right|^2} \right] \left|\left| v^k - \overline{v} \right|\right|^2$$

$$+ \left[\underline{a}_2(r) - (1 - \epsilon_v + \epsilon_v^2)\, A_*(r) - 3\epsilon_p\, B_2^2(r) \right] \left|\left| \hat{c}^k - v^k \right|\right|^2$$

$$+ O(\left|\left| HK_*(\overline{v}) - \hat{u}^k \right|\right|) \left|\left| v^k - \overline{v} \right|\right|^2 + O(\left|\left| \hat{c}^k - \overline{v} \right|\right|) \left|\left| v^k - \overline{v} \right|\right|^2 \tag{4.50}$$

where D(r) is defined in (4.42).

If ϵ_v and ϵ_2 are chosen such that ϵ_v is very near to 0.5 and ϵ_2 is sufficiently small, then from assumption (4), the following holds :

$$2a_*(r) - (1 - \epsilon_v)\, A_*(r) - D(r) > 3\epsilon_2\, B_2^2(r) \geq 3\epsilon_p\, B_2^2(r) \tag{4.51}$$

$$\underline{a}_2(r) - (1 - \epsilon_v + \epsilon_v^2)\, A_*(r) > 3\epsilon_2\, B_2^2(r) \geq 3\epsilon_p\, B_2^2(r) \tag{4.52}$$

It can be proved, under the assumptions of the theorem, that $\hat{u}(\alpha(v), \lambda(v, p), p)$ and $\hat{c}(\alpha(v), \lambda(v, p), p)$ are continuous with respect to v and p (Lin, Chen and Roberts, 1988; Lin, Hendawy and Roberts, 1988b). Hence, from (4.50) — (4.52), there is an r such that for any $v^0 \in s_1(\bar{v}, r)$ and $p_0 \in s_3(\bar{p}, r)$, the following holds :

$$\frac{1}{\epsilon_p} \left|\left| p_k - \bar{p} \right|\right|^2 + \frac{1}{\epsilon_p} \left|\left| p_{k+1} - p_k \right|\right|^2 + a_*(r) \left|\left| v^k - \bar{v} \right|\right|^2 - a_*(r) \left|\left| v^{k+1} - \bar{v} \right|\right|^2 \geq 0 \tag{4.53}$$

Furthermore, if $v^k \neq \bar{v}$ or $\hat{c}^k \neq \bar{v}$ then :

$$\frac{1}{\epsilon_p} \left|\left| p_k - \bar{p} \right|\right|^2 + \frac{1}{\epsilon_p} \left|\left| p_{k+1} - p_k \right|\right|^2 + a_*(r) \left|\left| v^k - \bar{v} \right|\right|^2 - a_*(r) \left|\left| v^{k+1} - \bar{v} \right|\right|^2 > 0 \tag{4.54}$$

Define :

$$Z(v, p) = \frac{1}{\epsilon_p} \left|\left| p - \bar{p} \right|\right|^2 + a_*(r) \left|\left| v - \bar{v} \right|\right|^2$$

According to (4.54), Z(. , .) is well qualified to be a Zangwill function for, if $v^k \neq \bar{v}$, (4.54) strictly holds. On the other hand, it is obvious that if $p_k = \bar{p}$ then $\hat{c}^k = v^k$ and (4.54) holds. That is, $Z(v^k, p_k)$ is strictly decreasing provided that $v^k \neq v$ or $p_k \neq p$. It is readily seen that all conditions of the Zangwill theorem (Zangwill, 1969; Lin, Chen and Roberts, 1988) are satisfied. Therefore, $\left\{ v^k \right\}$ and $\left\{ p_k \right\}$ strongly converge to $\bar{v}$ and $\bar{p}$, respectively.

QED

Inspecting the assumptions of the theorem, it is observed that the convergence conditions provided by Theorem 4.5 are sufficient conditions. In fact, the real restriction in theorem 4.5 is assumption (4). However, it is a requirement for the model only rather than the real process. It appears to be difficult to verify the assumptions of theorem 4.5 in some circumstances, especially for non-linear processes. In practice, however, this algorithm is found to be applicable to many practical problems.

4.2.1b Convergence Conditions for the Augmented Version of AL1

It is observed that the convergence conditions of the algorithm may be significantly relaxed by augmentation. The following theorem provides sufficient conditions for local convergence of the augmented version of the algorithm. For simplicity of the mathematical proof, we have assumed in the proof that $\tau_1=\tau_2=\tau$. This, of course, will not affect the generality of the analysis.

Definition 4.6

Define :

$$\tilde{L}_*(v , p) = L_*(v , p) + \frac{8}{17} \tau \left|\left| v - \bar{v} \right|\right|^2 \tag{4.55}$$

$$\tilde{q}_*(v) = q_*(v) + \frac{8}{17} \tau \left|\left| v - \bar{v} \right|\right|^2 \quad , \quad \tilde{q}(c,u,\alpha) = q(c,u,\alpha) + \frac{9}{17} \tau \left|\left| v - \bar{v} \right|\right|^2 \tag{4.56}$$

where τ is the penalty coefficient in the augmented Lagrangian.

$$B_1(r) = \sup_{v \in s_2(\bar{v} , r) , c \in s_2(\bar{v} , r)} \left|\left| h'_v(c , v) \right|\right| , \bar{v} \in \Omega_1^* \tag{4.57}$$

$$B_2(r) = \sup_{v \in s_2(\bar{v} , r)} \left|\left| HK'_*(v) \right|\right| , \bar{v} \in \Omega_1^* \tag{4.58}$$

$$A_3(r) = \sup_{v \in s_2(\bar{v} , r) , c \in s_2(\bar{v} , r)} \left|\left| h''_v(c , v) \right|\right| , \bar{v} \in \Omega_1^* \tag{4.59}$$

$$A_4(r) = \sup_{v \in s_2(\bar{v} , r)} \left|\left| HK'_*(v) \right|\right| , \bar{v} \in \Omega_1^* \tag{4.60}$$

Theorem 4.6

Assume (v, $HK_*(v)$) $\in \mathbb{CU}$ is a regular point of the constraints $f(c , u) \leq 0$. Assume :

(1) $K_*(\cdot)$, $f(\cdot , \cdot)$ and $q(\cdot , \cdot , \alpha)$, $\alpha \in \mathbb{A}$ are twice continuously Frèchet differentiable.

(2) The constraint set $\mathbb{CU}$ is compact.

(3) For any $r > 0$, the following holds :

(a) For every $\alpha \in \mathbb{B}$, $q'_c(\cdot, \cdot, \alpha)$ and $q'_u(\cdot, \cdot, \alpha)$ are uniformly monotone on $\overline{\mathbb{C}\mathbb{U}} \cap s_1(\overline{v}, r)$ with constant $a_1(\alpha, r) > 0$, where :

$$\underline{a}_1(r) = \inf_{\alpha \in \mathbb{B}} a_1(\alpha, r) > 0 \tag{4.61}$$

and $\overline{\mathbb{C}\mathbb{U}}$ is the closure of $\mathbb{C}\mathbb{U}$, $(\overline{v}) \in \Omega_1^*$.

(b) For every $\alpha \in \mathbb{B}$, $\hat{q}'_v(\cdot, \alpha)$ is uniformly monotone on $s_2(\overline{v}, r)$ with constant $a_2(\alpha, r) > 0$, and

$$\inf_{\alpha \in \mathbb{B}} a_2(\alpha, r) = \underline{a}_2(r) > 0 \tag{4.62}$$

(c) For every $(c,u) \in \mathbb{C}\mathbb{U} \cap s_1(\overline{v}, r)$, $h'_v(c,v)$ is twice continuously differentiable.

(d) For every $p \in s_3(\overline{p}, r)$, $L'_{*v}(\cdot, p)$ is uniformly monotone on $s_2(\overline{v}, r)$ with constant $a_*(r)$.

Then, for any given $r > 0$, there exist $\tau > 0$, $\epsilon_v = 0.5$ and $\epsilon_2 > 0$ such that :

for every $\epsilon_p \in (0, \epsilon_2)$ and for any $(v^0, HK_*(v^0)) \in \mathbb{C}\mathbb{U} \cap s_1(\overline{v}, r)$, $\overline{v} \in \Omega_1^*$ and $p^0 \in s_3(\overline{p}, r)$, the sequences $\{v^k\}$, $\{p_k\}$ generated by the augmented version of the algorithm satisfy :

$$\lim_{k \to \infty} v^k = \overline{v} \quad \text{and} \quad \lim_{k \to \infty} p_k = \overline{p} \tag{4.63}$$

Proof

Assume $v^k \in s_2(\overline{v}, r)$ and $p^k \in s_3(\overline{p}, r)$. Then, from the definition of $(\hat{c}^k, \hat{u}^k)$, the following inequality holds (Luenberger, 1984) :

$$\Big[q_c'(\hat{c}^k, \hat{u}^k, \alpha^k) - q_c'(v^k, HK_*(v^k), \alpha^k) - q_u'(v^k, HK_*(v^k), \alpha^k) HK_*'(v^k) + q_*'(v^k)$$

$$- p_k^T HK_*'(v^k) \Big] (\overline{v} - \hat{c}^k) + \Big[q_u'(\hat{c}^k, \hat{u}^k, \alpha^k) + p_k^T \Big] (HK_*(\overline{v}) - \hat{u}^k)$$

$$+ 2\tau (\hat{c}^k - v^k)^T(\overline{v} - \hat{c}^k) + 2\tau (\hat{u}^k - HK_*(v^k))^T(HK_*(\overline{v}) - \hat{u}^k) \geq 0 \tag{4.64}$$

Since at the optimum $\hat{c} = \overline{v}$ and $\hat{u} = HK_*(\overline{v})$, equations (3.27) and (3.28) produce (Luenberger, 1984)

$$\Big[- q_u'(\overline{v}, HK_*(\overline{v}), \overline{\alpha}) HK_*'(\overline{v}) + q_*'(\overline{v}) - \overline{p}\, HK_*'(\overline{v}) \Big] (\hat{c}^k - \overline{v})$$

$$+ \Big[q_u'(\overline{v}, HK_*(\overline{v}), \overline{\alpha}) - \overline{p}^T \Big] (\hat{u}^k - HK_*(\overline{v})) \geq 0 \tag{4.65}$$

Adding (4.65) to (4.64) yields :

$$q_c'(\hat{c}^k, \hat{u}^k, \alpha^k) (\overline{v} - \hat{c}^k) + q_u'(\hat{c}^k, \hat{u}^k, \alpha^k) (HK_*(\overline{v}) - \hat{u}^k) + 2\tau (\hat{c}^k - \overline{v})^T(\overline{v} - \hat{c}^k)$$

$$+ 2\tau (\hat{u}^k - HK_*(\overline{v}))^T(HK_*(\overline{v}) - \hat{u}^k) + 2\tau (HK_*(\overline{v}) - HK_*(v^k))^T(HK_*(\overline{v}) - \hat{u}^k)$$

$$+ q_c'(\overline{v}, HK_*(\overline{v}), \alpha^k) (\hat{c}^k - \overline{v}) + q_u'(\overline{v}, HK_*(\overline{v}), \alpha^k) (\hat{u}^k - HK_*(\overline{v}))$$

$$+ q_c'(v^k, HK_*(v^k), \alpha^k) (\hat{c}^k - \overline{v}) + q_u'(v^k, HK_*(v^k), \alpha^k) HK_*'(v^k) (\hat{c}^k - \overline{v})$$

$$+ q_c'(\overline{v}, HK_*(\overline{v}), \alpha^k) (\overline{v} - \hat{c}^k) + q_u'(\overline{v}, HK_*(\overline{v}), \alpha^k) HK_*'(\overline{v}) (\overline{v} - \hat{c}^k)$$

$$+ q_u'(\overline{v}, HK_*(\overline{v}), \alpha^k) (HK_*(\overline{v}) + HK_*'(\overline{v}) (\hat{c}^k - \overline{v}) - \hat{u}^k)$$

$$+ 2\tau (v^k - \overline{v})^T(\hat{c}^k - \overline{v}) - q_u'(\overline{v}, HK_*(\overline{v}), \overline{\alpha}) (HK_*(\overline{v}) + HK_*'(\overline{v}) (\hat{c}^k - \overline{v}) - \hat{u}^k)$$

$$+ q_*'(v^k) (\overline{v} - \hat{c}^k) + q_*'(\overline{v}) (\hat{c}^k - \overline{v}) + p_k^T \Big[HK_*(\overline{v}) + HK_*'(v^k) (\hat{c}^k - \overline{v}) - \hat{u}^k \Big]$$

$$+ \overline{p}^T \Big[\hat{u}^k - HK_*(\overline{v}) - HK_*'(\overline{v}) (\hat{c}^k - \overline{v}) \Big] \geq 0 \tag{4.66}$$

Following the same reasoning as in theorem 4.5 and using assumptions (3), it is readily derived (Lin, Chen and Roberts, 1988) :

$$q_c'(\hat{c}^k, \hat{u}^k, \alpha^k)(\bar{v} - \hat{c}^k) + q_u'(\hat{c}^k, \hat{u}^k, \alpha^k)(HK_*(\bar{v}) - \hat{u}^k)$$

$$+ q_c'(\bar{v}, HK_*(\bar{v}), \alpha^k)(\hat{c}^k - \bar{v}) + q_u'(\bar{v}, HK_*(\bar{v}), \alpha^k)(\hat{u}^k - HK_*(\bar{v}))$$

$$+ 2\tau(\hat{c}^k - \bar{v})(\bar{v} - \hat{c}^k) + 2\tau(\hat{u}^k - HK_*(\bar{v}))(HK_*(\bar{v}) - \hat{u}^k)$$

$$\leq - \Big[\underline{a}_1(r) + 2\tau \Big] \Big[\big\| \hat{c}^k - \bar{v} \big\|^2 + \big\| HK_*(\bar{v}) - \hat{u}^k \big\|^2 \Big] \tag{4.67}$$

and

$$\tilde{q}_c'(v^k, HK_*(v^k), \alpha^k)(\hat{c}^k - \bar{v}) + \tilde{q}_u'(v^k, HK_*(v^k), \alpha^k) HK_*'(v^k)(\hat{c}^k - \bar{v})$$

$$+ \tilde{q}_c'(\bar{v}, HK_*(\bar{v}), \alpha^k)(\bar{v} - \hat{c}^k) + \tilde{q}_u'(\bar{v}, HK_*(\bar{v}), \alpha^k) HK_*'(\bar{v})(\bar{v} - \hat{c}^k)$$

$$\leq -\tfrac{1}{2} \Big[\underline{a}_2(r) + \tfrac{18}{17}\tau \Big] \big\| \hat{c}^k - v^k \big\|^2 + \tfrac{1}{2} \Big[\overline{A}_2(r) + \tfrac{18}{17}\tau \Big] \big\| v^k - \bar{v} \big\|^2$$

$$+ \tfrac{1}{2} \Big[\overline{A}_2(r) + \tfrac{18}{17}\tau \Big] \big\| \hat{c}^k - \bar{v} \big\|^2 \tag{4.68}$$

where $\overline{A}_2(r) = \sup_{\alpha \in \mathbb{B}} A_2(\alpha, r)$ and $A_2(\alpha, r)$ is the Lipschitz constant of $\hat{q}_v'(\,.\,, \alpha)$.

$$q_u'(v^k, HK_*(v^k), \alpha^k)(HK_*(\bar{v}) + HK_*'(\bar{v})(\hat{c}^k - v^k) - \hat{u}^k)$$

$$- q_u'(\bar{v}, HK_*(\bar{v}), \bar{\alpha})(HK_*(\bar{v}) + HK_*'(\bar{v})(\hat{c}^k - \bar{v}) - \hat{u}^k)$$

$$+ 2\tau(HK_*(\bar{v}) - HK_*(v^k))^T(HK_*(\bar{v}) - \hat{u}^k)$$

$$= \Big[h(\bar{v}, v^k) - h(\bar{v}, \bar{v}) \Big] \Big[(HK_*(\bar{v}) - \hat{u}^k) + HK_*'(\bar{v})(\hat{c}^k - \bar{v}) \Big]$$

$$+ 2\tau(HK_*(\bar{v}) - HK_*(v^k))^T(HK_*(\bar{v}) - \hat{u}^k)$$

$$\leq B_1(r) \big\| \hat{u}^k - HK_*(\bar{v}) \big\| \big\| v^k - \bar{v} \big\| + B_1(r) B_2(r) \big\| v^k - \bar{v} \big\| \big\| \hat{c}^k - \bar{v} \big\|$$

$$+ \tfrac{1}{2} A_3(r) \big\| v^k - \bar{v} \big\|^2 \Big(\big\| HK_*(\bar{v}) - \hat{u}^k \big\| + B_2(r) \big\| \hat{c}^k - \bar{v} \big\| \Big)$$

$$+ 2\tau \big\| HK_*(\bar{v}) - \hat{u}^k \big\| \Big(B_2(r) \big\| v^k - \bar{v} \big\| + \tfrac{1}{2} A_4(r) \big\| v^k - \bar{v} \big\|^2 \Big)$$

$$= \Big[B_1(r) + 2\tau\, B_2(r) + \Big(\tfrac{1}{2}\,(A_3(r) + \tau\, A_4(r)\Big) \, \| v^k - \overline{v} \| \Big) \Big]$$

$$\times \| \hat{u}^k - HK_*(\overline{v}) \| \; \| v^k - \overline{v} \|$$

$$+ \Big[B_1(r)\, B_2(r) + \Big(\tfrac{1}{2}\, B_2(r)\, A_3(r) + \tau\, A_4(r) \Big) \| v^k - \overline{v} \| \Big] \| \hat{c}^k - \overline{v} \| \; \| v^k - \overline{v} \|$$

$$\leq \Big[\underline{a}_1(r) + 2\tau - \tfrac{3}{2}\, \epsilon_p \Big] \| \hat{u}^k - HK_*(\overline{v}) \|^2 + \frac{B_3^2(r,\tau,v^k)}{4\Big[\underline{a}_1(r) + 2\tau - \tfrac{3}{2}\, \epsilon_p \Big]} \| v^k - \overline{v} \|^2$$

$$+ \Big[\underline{a}_1(r) + \tfrac{9}{17}\, \tau - \tfrac{1}{2}\, \overline{A}_2(r) \Big] \| \hat{c}^k - \overline{v} \|^2$$

$$+ \frac{B_4^2(r,\tau,v^k)}{2\Big[2\underline{a}_1(r) + \tfrac{9}{17}\, \tau - \tfrac{1}{2}\, \overline{A}_2(r) \Big]} \| v^k - \overline{v} \|^2 \qquad (4.69)$$

where :

$$B_3(r,\tau,v^k) = B_1(r) + 2\tau\, B_2(r) + \Big(\tfrac{1}{2}\, A_3(r) + \tau\, A_4(r) \Big) \| v^k - \overline{v} \| \qquad (4.70)$$

$$B_4(r,\tau,v^k) = B_1(r)\, B_2(r) + \Big(\tfrac{1}{2}\, B_2(r)\, A_3(r) + \tau\, A_4(r) \Big) \| v^k - \overline{v} \| \qquad (4.71)$$

$$\tilde{q}'_*(v^k)\, (\overline{v} - \hat{c}^k) + \tilde{q}'_*(\overline{v})\, (\hat{c}^k - \overline{v}) + p_k^T \Big[HK_*(\overline{v}) + HK'_*(v^k)\, (\hat{c}^k - \overline{v}) - \hat{u}^k \Big]$$

$$+ \overline{p}^T \Big[\hat{u}^k - HK_*(\overline{v}) - HK'_*(\overline{v})\, (\hat{c}^k - \overline{v}) \Big]$$

$$= \tilde{L}'_{*v}\, (v^k, p_k)\, (\overline{v} - \hat{c}^k) + \tilde{L}'_{*v}\, (\overline{v},\overline{p})\, (\hat{c}^k - \overline{v}) + (\overline{p} - p_k)^T (\hat{u}^k - HK_*(\overline{v})) \qquad (4.72)$$

$$\tilde{L}'_{*v}\, (\overline{v} , \overline{p})\, (\hat{c}^k - \overline{v}) = \tilde{L}'_{*v}\, (\overline{v} , \overline{p})\, (\hat{c}^k - v^{k+1}) + \tilde{L}'_{*v}\, (\overline{v} , \overline{p})\, (v^{k+1} - \overline{v})$$

$$\leq \tilde{L}_*\, (v^{k+1} , \overline{p}) - \tilde{L}_*\, (\overline{v} , \overline{p}) - \tfrac{1}{2} \Big[a_*(r) + \tfrac{16}{17}\, \tau \Big] \| v^{k+1} - \overline{v} \|^2$$

$$+ \tilde{L}'_{*v}\, (\overline{v} , \overline{p})\, (\hat{c}^k - v^{k+1})$$

$$= \tilde{L}_* (v^{k+1}, \bar{p}) - \bar{p}^T u_*(v^k) + \bar{p}^T u'_*(v^k)(v^k - v^{k+1}) - \tilde{q}_*(\bar{v})$$

$$+ \bar{p}^T \Big[u_*(v^k) + u'_*(v^k)(\hat{c}^k - v^k) \Big] + \bar{p}^T u'_*(v^k)(v^{k+1} - \hat{c}^k)$$

$$- \tfrac{1}{2} \Big[a_*(r) + \tfrac{16}{17}\tau \Big] \| v^{k+1} - \bar{v} \|^2 + \tilde{L}'_{*v}(\bar{v}, \bar{p})(\hat{c}^k - v^{k+1}) \tag{4.73}$$

$$\tilde{L}'_{*v}(v^k, p_k)(\bar{v} - \hat{c}^k) = \tilde{L}'_{*v}(v^k, p_k)(\bar{v} - v^k) + \tilde{q}'_*(v^k)(v^k - v^{k+1})$$

$$+ \tilde{q}'_*(v^k)(v^{k+1} - \hat{c}^k) + p_k^T u'_*(v^k)(v^k - \hat{c}^k)$$

$$\leq \tilde{L}_*(\bar{v}, p_k) - \tilde{L}_*(v^k, p^k) - \tfrac{1}{2}\Big[a_*(r) + \tfrac{16}{17}\tau \Big] \| v^k - \bar{v} \|^2 + \tilde{q}'_*(v^k)(v^k - v^{k+1})$$

$$+ \tilde{q}'_*(v^k)(v^{k+1} - \hat{c}^k) + p_k^T u'_*(v^k)(v^k - \hat{c}^k)$$

$$= \tilde{q}_*(\bar{v}) - \tilde{q}_*(v^k) - p_k^T \Big[u_*(v^k) + u'_*(v^k)(\hat{c}^k - v^k) \Big] + \tilde{q}'_*(v^k)(v^k - v^{k+1})$$

$$+ \tilde{q}'_*(v^k)(v^{k+1} - \hat{c}^k) - \tfrac{1}{2}\Big[a_*(r) + \tfrac{16}{17}\tau \Big] \| v^k - \bar{v} \|^2 \tag{4.74}$$

Substituting (4.73) and (4.74) into the right hand side of (4.72) yields :

$$\tilde{L}'_{*v}(\bar{v}, \bar{p})(\hat{c}^k - \bar{v}) + \tilde{L}'_{*v}(v^k, p_k)(\bar{v} - \hat{c}^k) + (\bar{p} - p_k)^T(\hat{u}^k - HK_*(\bar{v}))$$

$$= \tilde{L}_*(v^{k+1}, \bar{p}) - \tilde{L}_*(v^k, \bar{p}) + \tilde{L}'_{*v}(v^k, \bar{p})(v^k - v^{k+1})$$

$$+ (\bar{p} - p_k)^T \Big[\hat{u}^k - HK_*(v^k) - HK'_*(v^k)(\hat{c}^k - v^k) \Big] + \Big[\tilde{L}'_{*v}(\bar{v}, \bar{p}) - \tilde{L}'_{*v}(v^k, \bar{p}) \Big]$$

$$(\hat{c}^k - v^{k+1}) - \tfrac{1}{2}\Big[a_*(r) + \tfrac{16}{17}\tau \Big] \| v^k - \bar{v} \|^2$$

$$- \tfrac{1}{2}\Big[a_*(r) + \tfrac{16}{17}\tau \Big] \| v^{k+1} - \bar{v} \|^2 \tag{4.75}$$

Applying (4.67) — (4.69) and (4.75) to (4.66) produces :

$$-\frac{1}{2}\Big[\underline{a}_2(r) + \frac{18}{17}\tau\Big] \| \hat{c}^k - v^k \|^2 + \frac{1}{2}\Big(D(r,\tau) - a_*(r) - \frac{16}{17}\tau\Big) \| v^k - \overline{v} \|^2$$

$$-\frac{1}{2}\Big[a_*(r) + \frac{16}{17}\tau\Big] \| v^{k+1} - \overline{v} \|^2 - \frac{1}{2}\Big[a_*(r) + \frac{16}{17}\tau\Big] \| v^{k+1} - \overline{v} \|^2$$

$$+ \tilde{L}_*(v^{k+1}, \overline{p}) - \tilde{L}_*(v^k, \overline{p}) + \tilde{L}'_{*v}(v^k, \overline{p})(v^k - v^{k+1})$$

$$+ \Big[\tilde{L}'_{*v}(\overline{v}, \overline{p}) - \tilde{L}'_{*v}(v^k, \overline{p})\Big](\hat{c}^k - v^{k+1}) - \frac{3}{2}\epsilon_p \| \hat{u}^k - HK_*(\overline{v}) \|^2$$

$$+ (\overline{p} - p^k)\Big[\hat{u}^k - HK_*(v^k) - HK'_*(v^k)(\hat{c}^k - v^k)\Big] \geq 0 \qquad (4.76)$$

where :

$$D(r,\tau) = \Big[\overline{A}_2(r) + \frac{18}{17}\tau\Big] + \frac{B_3^2(r,\tau,v^k)}{2\Big[\underline{a}_1(r) + 2\tau - \frac{3}{2}\epsilon_p\Big]} +$$

$$+ \frac{B_4^2(r,\tau,v^k)}{2\Big[2\underline{a}_1(r) + \frac{9}{17}\tau - \frac{1}{2}\overline{A}_2(r)\Big]} \qquad (4.77)$$

According to (3.36) :

$$\frac{1}{2}\| p_{k+1} - p_k \|^2 = \frac{1}{2}\| p_k - \overline{p} \|^2 + \epsilon_p (p_k - \overline{p})^T\Big[\hat{u}^k - HK_*(v^k) - HK'_*(v^k)(\hat{c}^k - v^k)\Big]$$

$$+ \frac{1}{2}\epsilon_p^2 \| \hat{u}^k - HK_*(v^k) - HK'_*(v^k)(\hat{c}^k - v^k) \|^2$$

$$\leq \frac{1}{2}\| p_k - \overline{p} \|^2 + \frac{3}{2}\epsilon_p^2 \Big(\| \hat{u}^k - HK_*(\overline{v}) \|^2$$

$$+ (B_2(r) + A_4(r) \| v^k - \overline{v} \|)^2 \| v^k - \overline{v} \|^2 + B_2(r) \| \hat{c}^k - v^k \|^2\Big)$$

$$+ \epsilon_p (p_k - \overline{p})^T\Big[\hat{u}^k - HK_*(v^k) - HK'_*(v^k)(\hat{c}^k - v^k)\Big] \qquad (4.78)$$

Applying (4.76) to (4.78) yields :

$$\frac{1}{\epsilon_p} \| p_k - \bar{p} \|^2 + \frac{1}{\epsilon_p} \| p_{k+1} - p_k \|^2 + \left[a_*(r) + \frac{16}{17}\tau \right] \| v^k - \bar{v} \|^2$$

$$- \left[a_*(r) + \frac{16}{17}\tau \right] \| v^{k+1} - \bar{v} \|^2$$

$$\geq \left[2 \left(a_*(r) + \frac{16}{17}\tau \right) - (1 - \epsilon_v) \left(A_*(r) + \frac{16}{17}\tau \right) - D(r,\tau) - 3\epsilon_p B_2^2(r) \right] \| v^k - \bar{v} \|^2$$

$$+ \left[\left(\underline{a}_2(r) + \frac{18}{17}\tau \right) - (1 - \epsilon_v + \epsilon_v^2) \left(A_*(r) + \frac{16}{17}\tau \right) - 3\epsilon_p B_2^2(r) \right] \| \hat{c}^k - v^k \|^2 \tag{4.79}$$

where $D(r,\tau)$ is defined in (4.77) and $A_*(r)$ is defined in theorem 4.5.

If ϵ_v is equal to 0.5, then the following holds :

$$2 \left[a_*(r) + \frac{16}{17}\tau \right] - (1 - \epsilon_v) \left[A_*(r) + \frac{16}{17}\tau \right] - D(r,\tau)$$

$$= 2a_*(r) - \frac{1}{2} A_*(r) - D_1(r) + \frac{6}{17}\tau \tag{4.80}$$

$$\left[\underline{a}_2(r) + \frac{18}{17}\tau \right] - (1 - \epsilon_v + \epsilon_v^2) \left[A_*(r) + \frac{16}{17}\tau \right]$$

$$= \underline{a}_2(r) - \frac{3}{4} A_*(r) + \frac{6}{17}\tau \tag{4.81}$$

where $D_1(r) = D(r,\tau) - \frac{18}{17}\tau$

It is seen from (4.80) and (4.81) that the right hand sides of (4.80) and (4.81) will approach $+\infty$ as $\tau \to +\infty$. Therefore, for any given $r > 0$, there is a τ such that :

$$2a_*(r) - \frac{1}{2} A_*(r) - D_1(r) + \frac{6}{17}\tau > 3\epsilon_2 B_2^2(r) \geq 3\epsilon_p B_2^2(r) \tag{4.82}$$

$$\underline{a}_2(r) - \frac{3}{4} A_*(r) + \frac{6}{17}\tau > 3\epsilon_2 B_2^2(r) \geq 3\epsilon_p B_2^2(r) \tag{4.83}$$

From (4.82) — (4.83), it is readily seen that if $v^k \neq \bar{v}$ or $\hat{c}^k \neq \bar{v}$, then :

$$\frac{1}{\epsilon_p} \left\| p^k - \bar{p} \right\|^2 + \frac{1}{\epsilon_p} \left\| p^{k+1} - p^k \right\|^2 + \left[a_*(r) + \frac{16}{17} \tau \right] \left\| v^k - \bar{v} \right\|^2$$

$$- \left[a_*(r) + \frac{16}{17} \tau \right] \left\| v^{k+1} - \bar{v} \right\|^2 > 0 \tag{4.84}$$

Define :

$$Z (v , p) = \frac{1}{\epsilon_p} \left\| p - \bar{p} \right\|^2 + \left[a_*(r) + \frac{16}{17} \tau \right] \left\| v - \bar{v} \right\|^2$$

According to (4.84), Z(. , .) is well qualified to be a Zangwill function for, if $v^k \neq \bar{v}$, (4.84) strictly holds. On the other hand, it is obvious that if $p_k = \bar{p}$ then $\hat{c}^k = v^k$ and (4.84) holds. That is, Z (v^k , p_k) is strictly decreasing provided that $v^k \neq v$ or $p_k \neq p$. It is readily seen that all conditions of the Zangwill theorem (Zangwill, 1969; Lin, Chen and Roberts, 1988) are satisfied. Therefore, $\left\{ v^k \right\}$ and $\left\{ p_k \right\}$ strongly converge to $\bar{v}$ and $\bar{p}$, respectively.

QED

Remark 4.1

It is concluded from Theorem 4.6 that the augmented version of the algorithm is globally convergent subject to mild conditions. Therefore, the major restriction of the Theorem 4.5 (i.e. assumption 4) is completely relaxed by introducing augmentation. In the proof of Theorem 4.6, it is assumed that $\epsilon_v = 0.5$. However, this can be changed by redefining the coefficients of τ in (4.55) and (4.56), i.e.

$$\tilde{L}_*(v , p) = L_*(v , p) + \delta\tau \left\| v - \bar{v} \right\|^2$$

$$\tilde{q}_*(v) = q_*(v) + (2 - \delta)\tau \left\| v - \bar{v} \right\|^2$$

where $0 < \delta < 2$.

4.2.1c Convergence Conditions for Algorithms AL2 and AL3

Convergence analysis for algorithms AL2 and AL3 is presented in this section. The following definitions are needed in establishing the global convergence theorem for the algorithm AL2.

Definition 4.7

Define the following point-to-set mappings :

$\hat{c} : \mathbb{R}^n \to \mathbb{R}^n$, $\hat{c}(v)$ is the optimum solution set of MOP1 for given v;

$$d : \mathbb{R}^n \to \mathbb{R}^n \times \mathbb{R}^n,\ d(v) = \left\{ (v,c) \in \mathbb{R}^n \times \mathbb{R}^n;\ c \in \hat{c}(v) \right\} \tag{4.85}$$

$$w : \mathbb{R}^n \times \mathbb{R}^n \to \mathbb{R}^n,\ w(v,c) = \left\{ w \in \mathbb{R}^n;\ w = (1 - \epsilon_V)v + \epsilon_V c,\ 0 < \delta \leq \epsilon_V \leq \beta(v) \right\} \tag{4.86}$$

where $\beta(v)$ is any function of v defined on $\mathbb{R}^n$;

$$\Phi : \mathbb{R}^n \to \mathbb{R}^n,\ \Phi = w \circ d \tag{4.87}$$

Assumption 4.1

It is assumed in this section that $q_*(\cdot)$ is continuously differentiable on $\mathbb{C}$ and its derivative $q_*'(\cdot)$ is Lipschitz continuous on $\mathbb{C}$ with constant $A_*>0$, i.e.

$$\left\| q_*'(c+h) - q_*'(c) \right\| \leq A_* \left\| h \right\|,\quad \forall c,\ c+h \in \mathbb{C} \tag{4.88}$$

Theorem 4.7

Let assumptions 3.1 and 4.1 be satisfied and assume :

(i) for any given $v \in \mathbb{C}$, function $q(\cdot,v)$ is uniformly monotone on $\mathbb{C}$ with constant $\alpha(v)$, i.e.

$$\left[q_c'(c+h\ ,\ v) - q_c'(c\ ,\ v) \right] h > \alpha(v) \left\| h \right\|^2,\ \alpha(v) > 0 \tag{4.89}$$

(ii) function $\alpha(\cdot)$ is upper semicontinuous on $\mathbb{C}$; and

$$\alpha = \inf_{v \in \mathbb{C}} \left\{ \alpha(v) \right\} > 0 \tag{4.90}$$

(iii) function $\beta(\cdot)$ in (4.86) is chosen such that :

$$\beta(v) \leq \min \left\{ \frac{2\alpha}{A_*} - \epsilon,\ 1 \right\} \tag{4.91}$$

where $0 < \delta \leq 1$, $\epsilon > 0$ and $\delta + \epsilon \leq \frac{2\alpha}{A_*}$.

Then,

I. the point-to-set mapping Φ is well defined and closed on $\mathbb{C} \backslash \Omega$, where Ω is defined in (4.6);

II. there is at least one cluster point of the sequence $\{v^k\}$ generated by the algorithm $v^{k+1} \in \Phi(v^k)$ and every cluster point of the sequence $\{v^k\}$ belongs to Ω.

Proof

Since the algorithm AL2 may be viewed as a kind of the Improved Modified Two-Step (IMTS) method (Lin, Han, Roberts and Wan, 1989), the proof of this theorem can be established in the same way as found in the proof of Theorem 6 of the given reference (Lin, Han, Roberts and Wan, 1989).

QED

A local convergence theorem is established for the algorithm AL3. The following definitions are introduced for the mathematical clarity of the proof.

Definition 4.8

Define :

$$b(r) = \inf_{c,v \in S(v^*,r)} \min \mu \left(\hat{f}_c'(c,v)^T \hat{f}_c'(c,v) \right) \tag{4.92}$$

where $\mu(A)$ indicates the eigenvalue set of A, $v^* \in \Omega$ with Ω defined in (4.6) and

$$S(v^*,r) = \left\{ v;\ \left\| v - v^* \right\|^2 \leq r^2 \right\} \tag{4.93}$$

$$B_1(r) = \sup_{v_1 \in S(v^*,r)\ v_2 \in S(v^*,r)} \left\| \left[f_*'(v_1) \right]^T f_u'(v_2, HK_*(v_2)) \left[\hat{HK}_*'(v_2) - HK_*'(v_2) \right] \right\| \tag{4.94}$$

where $v^* \in \Omega$, $f_*(v)$ is defined in equation (4.11).

$$B_2(r) = \sup_{v \in S(v^*, r)} \left\| \left[f'_c(v, HK_*(v)) + f'_u(v, HK_*(v))\, H\hat{K}'_*(v) \right]^T f'_u(v, HK_*(v)) \times \left[H\hat{K}'_*(v) - HK'_*(v) \right] \right\| \tag{4.95}$$

$$L(c,v) = q(c,v) + \rho \hat{f}(c,v)^T \sigma(v,v)\, \hat{f}(c,v) \tag{4.96}$$

where $\sigma(c,v)$ is defined in (4.53).

$$L_*(v) = q_*(v) + \rho f_*(v)^T \sigma_*(v)\, f_*(v) \tag{4.97}$$

where

$$\sigma_*(v) = \left[\sigma_{*1}(v) \cdots \sigma_{*t}(v) \right]^T \tag{4.98}$$

with

$\sigma_{*i}(v) = 1$, while $f_{*i}(v) > 0$, and

$\sigma_{*i}(v) = 0$, while $f_{*i}(v) \leq 0$, $i = 1, ..., m$ (4.99)

Here, m is the dimension of $f_*(v)$.

Definition 4.9

Define the following point-to-set mappings :

$\hat{c} : \mathbb{R}^n \rightarrow \mathbb{R}^n$, $\hat{c}(v)$ is the optimum solution set of (HMMOP2) for given v;

$\Phi : \mathbb{R}^n \rightarrow \mathbb{R}^n$, $\Phi = w \circ d$ (4.100)

where w and d are defined in definition 4.7.

Theorem 4.8

Let assumptions 3.1 and 4.1 be satisfied and assume :

(i) $Q(\cdot, \cdot, \cdot)$, $f(\cdot, \cdot)$ and $K_*(\cdot)$ are twice continuously Frèchet differentiable;

(ii) $f_*(\cdot)$ is strictly complementary at any $v^* \in \Omega$ (Luenberger, 1984);

(iii) $L_*(\cdot)$ is strictly convex at v^*, the real optimum of HROPP;

(iv) for any given $r > 0$, assume :

(a) for any $v \in S(v^*,r)$, function $q(\cdot,v)$ is uniformly monotone on $S(v^*,r)$ with constant $\alpha(v,r)$, i.e.

$$\Big[q'_c(c+h, v) - q'_c(c, v) \Big] h > \alpha(v, r) \, \| h \|^2, \; \alpha(v, r) > 0 \tag{4.101}$$

(b) function $\alpha(\cdot,r)$ is upper semicontinuous on $\mathbb{C}$; and

$$\underline{\alpha}(r) = \inf_{v \in S(v^*,r)} \Big\{ \alpha(v,r) \Big\} > 0 \tag{4.102}$$

(c) function $\beta(\cdot)$ in (4.86) is chosen such that :

$$\beta(v) \leq \min \left\{ \frac{2\underline{\alpha}(r)}{A_*(r) + 4\rho B_1(r)} - \epsilon, \, 1 \right\} \tag{4.103}$$

where $0 < \delta \leq 1$, $\epsilon > 0$ and

$$\delta + \epsilon \leq \frac{2\underline{\alpha}(r)}{A_*(r) + 4\rho B_1(r)} \tag{4.104}$$

with $A_*(r)$ being the Lipschitz constant of $L'_*(\cdot)$ over $S(v^*,r)$ and, furthermore, the gain factor ϵ_v is kept unchanged during the iteration;

(iv) there is an r such that :

$$b(r) \geq B_2(r) \tag{4.105}$$

Then,

I. for any given $r>0$, the point-to-set mapping Φ is well defined and closed on $S(v^*,r)\backslash\{ v^* \}$, where v^* is the real optimum of HROPP.

II. there is an r such that for any given initial set-point $v^0 \in S(v^*,r)$ the sequence $\{ v^k \}$ generated by the algorithm $v^{k+1} \in \Phi(v^k)$ is convergent in the sense that the sequence has a unique cluster point v^*.

Proof

Clearly, assumption (i) means the continuity of $\hat{c}(\cdot)$ (Findeisen et al., 1980) and, therefore, from assumption (ii), it is not difficult to prove that there exists an r_1 such that for any given $v^{k-1} \in S(v^*,r_1)$, (4.105) is satisfied and, furthermore, we have (Lin, Chen and Roberts, 1988) :

$$\sigma(v^k,v^k) = \sigma(\hat{c}^{k-1},v^{k-1}) \tag{4.106}$$

From assumption (iii), it is clear that for any given $r_1>0$ and $\rho>0$, function $\tilde{L}_*(\cdot,\cdot,r_1,\rho)$, defined as :

$$\tilde{L}_*(v^k,v^{k-1},r_1,\rho) = L_*(v^k) + \frac{\rho}{\epsilon_v}\left[\epsilon_v B_1(r_1) + B_2(r_1)\right] \left|\left| v^k - v^{k-1} \right|\right|^2 \tag{4.107}$$

is strictly convex at (v^*, v^*). Therefore, $\tilde{L}_*(\cdot,\cdot,r_1,\rho)$ is strictly convex in a neighbourhood of (v^*, v^*), because assumption (i) ensures the continuity of $\tilde{L}_*(\cdot,\cdot,r_1,\rho)$ up to the second order. We first assume that function $\tilde{L}_*(\cdot,\cdot,r_1,\rho)$ defined in (4.107) is a descent function with respect to the algorithm AL3 over a certain convex set containing (v^*, v^*), i.e.

$$\tilde{L}_*(v^k,v^{k-1},r_1,\rho) \geq \tilde{L}_*(v^{k+1},v^k,r_1,\rho) \tag{4.108}$$

where v^k is determined according to the updating formula (3.54). This assumption will be verified later. Since $\tilde{L}_*(\cdot,\cdot,r_1,\rho)$ is strictly convex in a neighbourhood of (v^*,v^*), there is an r_2 such that $\tilde{L}_*(\cdot,\cdot,r_1,\rho)$ is strictly convex on the closed ball $\tilde{S}(v^*,v^*,r_2)$, where :

$$\tilde{S}(v_1,v_2,r_2) = \left\{ (v_1,v_2);\ \left|\left| v_1 - v^* \right|\right|^2 + \left|\left| v_2 - v^* \right|\right|^2 \leq r_2^2 \right\} \tag{4.109}$$

Define :

$$L_1(r_1,r_3) = \min_{v_1,v_2 \in \tilde{S}(v^*,v^*,r_2)\backslash\tilde{S}(v^*,v^*,r_3)} \tilde{L}_*(\cdot\,,\cdot\,,r_1,\rho) \tag{4.110}$$

and

$$C(r_1,r_3) = \left\{ (v_1,v_2);\ v_1,v_2 \in \tilde{S}(v^*,v^*,r_3),\ \tilde{L}_*(v_1,v_2,r_1,\rho) \leq L_1(r_1,r_3) \right\} \tag{4.111}$$

where :

$$r_3 = \tfrac{1}{2} \min \left\{ r_1\ ,\ r_2 \right\} \tag{4.112}$$

Suppose that (4.108) is valid over $C(r_1,r_3)$ then, from the definition of $C(r_1,r_3)$, we conclude that if v^k, v^{k-1} are included in $C(r_1,r_2)$ so will be (v^{k+1},v^k).

Now, we are going to verify that (4.108) holds over the set $C(r_1,r_3)$. Since :

$$L_1(r_1,r_3) > \tilde{L}_*(v^*,v^*,r_1,\rho) \tag{4.113}$$

one knows that

$$(v^*,v^*) \in C(r_1,r_3) \tag{4.114}$$

Futhermore, from assumption (i) one knows that $\hat{c}(\cdot)$ is continuous. Therefore, there exists an r, $0<r<r_3$, such that for any given $v^{k-1} \in S(v^*,r)$, we have $(v^k,v^{k-1}) \in C(r_1,r_3)$.

Since $\hat{c}^k$ is the optimum of MOP2 for given v^k, it is seen from the proof of Theorem 4.3 that the following holds :

$$L'_c(\hat{c}^k,v^k) - \lambda(v^k) = 0 \tag{4.115}$$

Substituting (3.55) into (4.115) and multiplying by $(v^k - \hat{c}^k)$ yields :

$$\Big\{ L'_c(\hat{c}^k,v^k) - L'_c(v^k,v^k) + L'_*(v^k) - 2\rho \Big[\hat{f}(v^k,v^k)\, \sigma(v^k,v^k)$$

$$- \hat{f}(\hat{c}^{k-1},v^{k-1})\, \sigma(\hat{c}^{k-1},v^{k-1}) \Big]\, f'_u(v^k,HK_*(v^k)) \Big[\hat{HK}'_*(v^k) - HK'_*(v^k) \Big] \Big\} (v^k - \hat{c}^k) = 0 \tag{4.116}$$

Suppose that $v^{k-1} \in S(v^*,r)$, then, $(v^k,v^{k-1}) \in C(r_1,r_3)$. Using the definitions of r_1, $C(r_1,r_3)$ and equation (4.106), (4.116) can be rewritten as :

$$\Big\{ L'_c(\hat{c}^k,v^k) - L'_c(v^k,v^k) + L'_*(v^k) - 2\rho \Big[\hat{f}(v^k,v^k)$$

$$- \hat{f}(\hat{c}^{k-1},v^{k-1}) \Big] \sigma(v^k,v^k)\, f'_u(v^k,HK_*(v^k)) \Big[\hat{HK}'_*(v^k) - HK'_*(v^k) \Big] \Big\} (v^k - \hat{c}^k) = 0 \qquad (4.117)$$

Since :

$$\Big[\hat{f}(v^k,v^k) - \hat{f}(\hat{c}^{k-1},v^{k-1}) \Big] \sigma(v^k,v^k)\, f'_u(v^k,HK_*(v^k)) \Big[\hat{HK}'_*(v^k) - HK'_*(v^k) \Big] (v^k - \hat{c}^k)$$

$$= \Big[\hat{f}(v^k,v^k) - \hat{f}(\hat{c}^{k-1},v^k) + \hat{f}(\hat{c}^{k-1},v^k) - \hat{f}(\hat{c}^{k-1},v^{k-1}) \Big] \sigma(v^k,v^k)\, f'_u(v^k,HK_*(v^k))$$

$$\times \Big[\hat{HK}'_*(v^k) - HK'_*(v^k) \Big] (v^k - \hat{c}^k)$$

$$= \Big\{ f(v^k,HK_*(v^k)) - f(v^{k-1},HK_*(v^{k-1})) - (\hat{c}^{k-1} - v^{k-1})^T \Big[f'_c(v^{k-1},HK_*(v^{k-1}))$$

$$+ f'_u(v^{k-1},HK_*(v^{k-1}))\, \hat{HK}'_*(v^{k-1}) \Big]^T \Big\} \sigma(v^k,v^k)\, f'_u(v^k,HK_*(v^k))$$

$$\times \Big[\hat{HK}'_*(v^k) - HK'_*(v^k) \Big] (v^k - \hat{c}^k)$$

$$\geq - \Big[\epsilon_v B_1(r_1) + B_2(r_1) \Big] \big|\big| \hat{c}^{k-1} - v^{k-1} \big|\big| \; \big|\big| \hat{c}^k - v^k \big|\big| \qquad (4.118)$$

it can be derived from (4.117) that :

$$L'_*(v^k)\, (v^k - \hat{c}^k) \geq \Big[\underline{a}(r_1) + 2\rho\, b(r_1) \Big] \big|\big| v^k - \hat{c}^k \big|\big|^2 - 2\rho \Big[\epsilon_v\, B_1(r_1) + B_2(r_1) \Big]$$

$$\big|\big| \hat{c}^{k-1} - v^{k-1} \big|\big| \; \big|\big| \hat{c}^k - v^k \big|\big|$$

$$\geq \Big[\underline{a}(r_1) + 2\rho\, b(r_1) - \rho\, (\epsilon_v\, B_1(r_1) + B_2(r_1)) \Big] \big|\big| v^k - \hat{c}^k \big|\big|^2 - \rho\, (\, \epsilon_v\, B_1(r_1)$$

$$+ B_2(r_1)\,)\, \big|\big| v^{k-1} - \hat{c}^{k-1} \big|\big|^2 \qquad (4.119)$$

Brdyś and Roberts (1987) and Lin, Kambhampati and Roberts (1989) show that :

$$L_*(v^k) - L_*(v^{k+1}) \geq \epsilon_v\, L_*'(v^k)\,(v^k - \hat{c}^k) - \tfrac{1}{2}\epsilon_v^2\, A_*(r_1)\, \|\, v^k - \hat{c}^k \,\|^2 \tag{4.120}$$

Substituting (4.119) into (4.120) and using (4.107), (4.105) and (4.103) we obtain the required inequality :

$$\tilde{L}_*(v^k,v^{k-1},r_1,\rho) - \tilde{L}_*(v^{k+1},v^k,r_1,\rho) \geq \epsilon_v \Big[\, \underline{a}(r_1) + 2\rho\,(\, b(r_1) - B_2(r_1)\,)$$

$$-\ \epsilon_v\,(\, \tfrac{1}{2} A_*(r_1) + 2\rho\, B_1(r_1)\,)\Big]\, \|\, v^k - \hat{c}^k \,\|^2 \geq \delta \Big[\, \underline{a}(r_1) + 2\rho\,(\, b(r_1) - B_2(r_1)\,)$$

$$-\ \tfrac{1}{2}\,\beta(v^k)\,(A_*(r_1) + 4\rho\, B_1(r_1))\Big]\, \|\, v^k - \hat{c}^k \,\|^2 \geq \tfrac{1}{2}\,\delta\epsilon\,(A_*(r_1) + 4\rho\, B_1(r_1))\, \|\, v^k - \hat{c}^k \|^2$$

(4.121)

Hence, (4.108) indeed holds. Moreover, it is seen from (4.121) that inequality (4.108) strictly holds if v^{k+1} is different from v^k. Therefore, $\tilde{L}_*(\,\cdot\,,\,\cdot\,,r_1,\rho)$ is qualified to be a Zangwill function (Zangwill, 1969) with respect to the algorithmic mapping Φ defined in (4.100). It can be proved, following the same process as used by Brdyś and Roberts (1987) and Lin, Wang and Roberts (1989), that the algorithmic mapping Φ is closed. Furthermore, for any given r_1 and r_3, $C(r_1,r_3)$ is compact and $v^k \in C(r_1,r_3)$, k=1, 2, ... if $v^0 \in S(v^*,r)$. Therefore, all the assumptions for Zangwill's Convergence Theorem are satisfied (Zangwill, 1969). From Zangwill's Convergence theorem, we know that algorithm AL3 is convergent. Finally, the uniqueness of the cluster point of the sequence follows from the strict convexity of $\tilde{L}_*(\,\cdot\,,\,\cdot\,,r_1,\rho)$ and the uniqueness of v^*.

QED

It has been proved that the introduction of a variable augmentation technique improves the convergence of the algorithm AL2 and reduces the sensitivity of the algorithm efficiency to the gain factor selection (Brdyś, Abdullah and Roberts, 1987; Lin, Li, Wan and Roberts, 1989). We are now going to prove that this statement remains true for the algorithm AL3. The following theorem provides sufficient conditions for local convergence of the Augmented AL3.

Theorem 4.9

Let assumptions 3.1 and 4.1 be satisfied and assume :

(i) $Q(\cdot,\cdot,\cdot)$, $f(\cdot,\cdot)$ and $K_*(\cdot)$ are twice continuously Frèchet differentiable;

(ii) $f_*(\cdot)$ is strictly complementary at any $v^* \in \Omega$;

(iii) $L_*(\cdot)$ is strictly convex at v^*, the real optimum of HROPP;

(iv) for any given $r>0$, assume :

(a) for any $v \in S(v^*,r)$, function $q(\cdot,v)$ is uniformly monotone on $S(v^*,r)$ with constant $\alpha(v,r)$, i.e.

$$\Big[q_c'(c+h\,,\,v) - q_c'(c\,,\,v) \Big]\, h > \alpha(v\,,\,r)\, \|h\|^2,\ \alpha(v\,,\,r) > 0 \tag{4.122}$$

(b) function $\alpha(\cdot,r)$ is upper semicontinuous on $\mathbb{C}$; and

$$\underline{\alpha}(r) = \inf_{v \in S(v^*,r)} \Big\{ \alpha(v,r) \Big\} > 0 \tag{4.123}$$

(c) function $\beta(\cdot)$ in (4.86) is chosen such that :

$$\beta(v) \leq \min \left\{ \frac{2\underline{\alpha}(r)}{A_*(r) + 4\rho\, B_1(r)} - \epsilon,\ 1 \right\} \tag{4.124}$$

where $0 < \delta \leq 1$, $\epsilon > 0$ and

$$\delta + \epsilon \leq \frac{2\underline{\alpha}(r)}{A_*(r) + 4\rho\, B_1(r)} \tag{4.125}$$

with $A_*(r)$ being the Lipschitz constant of $L_*'(\cdot)$ over $S(v^*,r)$ and, furthermore, the gain factor ϵ_v is kept unchanged during the iteration;

Then,

I. for any given $r>0$, the point-to-set mapping Φ is well defined and closed on $S(v^*,r)\backslash\{v^*\}$, where v^* is the real optimum of HROPP;

II. there is an r and τ such that for any given initial set-point $v^0 \in S(v^*,r)$ the sequence $\{v^k\}$ generated by the Augmented AL3 mapping $v^{k+1} \in \Phi(v^k)$ is convergent in the sense that the sequence has a unique cluster point v^*.

Proof

Clearly, assumption (i) implies continuity of $\hat{c}(\cdot)$ (Findeisen et al., 1980) and, therefore, from assumption (ii) it is not difficult to prove that there exists an r_1 such that for any given $v^{k-1} \in S(v^*,r_1)$, we have (Lin, Chen and Roberts, 1988) :

$$\sigma(v^k,v^k) = \sigma(\hat{c}^{k-1},v^{k-1}) \tag{4.126}$$

From assumption (iii), it is clear that for any given $r_1>0$ and $\rho>0$, function $\tilde{L}_*(\cdot,\cdot,r_1,\rho)$, defined as :

$$\tilde{L}_*(v^k,v^{k-1},r_1,\rho) = L_*(v^k) + \frac{\rho}{\epsilon_v}\Big[\epsilon_v B_1(r_1) + B_2(r_1)\Big] \Big|\Big| v^k - v^{k-1} \Big|\Big|^2 \tag{4.127}$$

is strictly convex at (v^*,v^*). Therefore, $\tilde{L}_*(\cdot,\cdot,r_1,\rho)$ is strictly convex in a neighbourhood of (v^*, v^*), because assumption (i) ensures the continuity of $\tilde{L}_*(\cdot,\cdot,r_1,\rho)$ up to the second order. We first assume that function $\tilde{L}_*(\cdot,\cdot,r_1,\rho)$ defined in (4.126) is a descent function with respect to the Augmented AL3 over a certain convex set containing (v^*, v^*), i.e.

$$\tilde{L}_*(v^k,v^{k-1},r_1,\rho) \geq \tilde{L}_*(v^{k+1},v^k,r_1,\rho) \tag{4.128}$$

where v^k is determined according to the updating formula (3.54). This assumption will be verified later. Since $\tilde{L}_*(\cdot,\cdot,r_1,\rho)$ is strictly convex in a neighbourhood of (v^*,v^*), there is an r_2 such that $\tilde{L}_*(\cdot,\cdot,r_1,\rho)$ is strictly convex on the closed ball $\tilde{S}(v^*,v^*,r_2)$, where

$$\tilde{S}(v_1,v_2,r_2) = \Big\{ (v_1,v_2);\ \Big|\Big| v_1 - v^* \Big|\Big|^2 + \Big|\Big| v_2 - v^* \Big|\Big|^2 \leq r_2^2 \Big\} \tag{4.129}$$

Define :

$$L_1(r_1,r_3) = \min_{v_1,v_2 \in \tilde{S}(v^*,v^*,r_2)\backslash\tilde{S}(v^*,v^*,r_3)} \tilde{L}_*(\cdot,\cdot,r_1,\rho) \tag{4.130}$$

and

$$C(r_1,r_3) = \left\{ (v_1,v_2);\ v_1,v_2 \in \tilde{S}(v^*,v^*,r_3),\ \tilde{L}_*(v_1,v_2,r_1,\rho) \leq L_1(r_1,r_3) \right\} \tag{4.131}$$

where

$$r_3 = \tfrac{1}{2} \min \left\{ r_1,r_2 \right\} \tag{4.132}$$

Suppose that (4.128) is valid over $C(r_1,r_3)$, then, from the definition of $C(r_1,r_3)$ we conclude that if v^k, v^{k-1} are included in $C(r_1,r_2)$ so will be (v^{k+1},v^k).

Now, we are going to verify that (4.128) holds over the set $C(r_1,r_3)$. Since :

$$L_1(r_1,r_3) > \tilde{L}_*(v^*,v^*,r_1,\rho) \tag{4.133}$$

one knows that :

$$(v^*,v^*) \in C(r_1,r_3) \tag{4.134}$$

Furthermore, from assumption (i) one knows that $\hat{c}(\cdot)$ is continuous. Therefore, there exists an r, $0<r<r_3$, such that for any given $v^{k-1} \in S(v^*,r)$, we have $(v^k,v^{k-1}) \in C(r_1,r_3)$.

Since $\hat{c}^k$ is the optimum of HMMOP2 for any given v^k, it is seen from the proof of the Theorem 4.3 that the following holds :

$$L_c'(\hat{c}^k - v^k) - \lambda(v^k) + 2\tau\,(\hat{c}^k - v^k) = 0 \tag{4.135}$$

Substituting (3.55) into (4.135) and multiplying by $(v^k - \hat{c}^k)$ yields :

$$\begin{aligned} &\Big\{ L_c'(\hat{c}^k,v^k) - L_c'(v^k,v^k) + L_*'(v^k) + 2\rho \Big[\hat{f}(v^k,v^k)\,\sigma(v^k,v^k) \\ &- \hat{f}(\hat{c}^{k-1},v^{k-1})\,\sigma(\hat{c}^{k-1},v^{k-1}) \Big] f_u'(v^k,HK_*(v^k)) \Big[\hat{HK}_*'(v^k) - HK_*'(v^k) \Big] \Big\} (v^k - \hat{c}^k) \\ &= 2\tau \left\| \hat{c}^k - v^k \right\|^2 \end{aligned} \tag{4.136}$$

Suppose that $v^{k-1} \in S(v^*,r)$, then, $(v^k,v^{k-1}) \in C(r_1,r_3)$. Using the definitions of r_1, $C(r_1,r_3)$ and equation (4.126), (4.136) can be rewritten as :

$$\Big\{ L_c'(\hat{c}^k,v^k) - L_c'(v^k,v^k) + L_*'(v^k) + 2\rho \Big[\hat{f}(v^k,v^k)$$

$$- \hat{f}(\hat{c}^{k-1},v^{k-1}) \Big] \sigma(v^k,v^k)\, f_u'(v^k,HK_*(v^k)) \Big[\hat{HK}_*'(v^k) - HK_*'(v^k) \Big] \Big\} (v^k - \hat{c}^k)$$

$$= 2\tau \,\Big|\Big|\, \hat{c}^k - v^k \,\Big|\Big|^2 \tag{4.137}$$

Since :

$$\Big[\hat{f}(v^k,v^k) - \hat{f}(\hat{c}^{k-1},v^{k-1}) \Big] \sigma(v^k,v^k)\, f_u'(v^k,HK_*(v^k)) \Big[\hat{HK}_*'(v^k) - HK_*'(v^k) \Big] (v^k - \hat{c}^k)$$

$$= \Big[\hat{f}(v^k,v^k) - \hat{f}(\hat{c}^{k-1},v^k) + \hat{f}(\hat{c}^{k-1},v^k) - \hat{f}(\hat{c}^{k-1},v^{k-1}) \Big] \sigma(v^k,v^k)\, f_u'(v^k,HK_*(v^k))$$

$$\times \Big[\hat{HK}_*'(v^k) - HK_*'(v^k) \Big] (v^k - \hat{c}^k)$$

$$= \Big\{ f(v^k,HK_*(v^k)) - f(v^{k-1},HK_*(v^{k-1})) - (\hat{c}^{k-1} - v^{k-1})^T \Big[f_c'(v^{k-1},HK_*(v^{k-1}))$$

$$+ f_u'(v^{k-1},HK_*(v^{k-1}))\, \hat{HK}_*'(v^{k-1}) \Big]^T \Big\} \sigma(v^k,v^k)\, f_u'(v^k,HK_*(v^k))$$

$$\times \Big[\hat{HK}_*'(v^k) - HK_*'(v^k) \Big] (v^k - \hat{c}^k)$$

$$\geq - \Big[\epsilon_v\, B_1(r_1) + B_2(r_1) \Big] \Big|\Big|\, \hat{c}^{k-1} - v^{k-1} \,\Big|\Big| \;\Big|\Big|\, \hat{c}^k - v^k \,\Big|\Big| \tag{4.138}$$

it can be derived from (4.137) that :

$$L_*'(v^k)\,(v^k - \hat{c}^k) \geq \Big[a(r_1) + 2\tau + 2\rho\, b(r_1) \Big] \Big|\Big|\, v^k - \hat{c}^k \,\Big|\Big|^2$$

$$- 2\rho \Big[\epsilon_v\, B_1(r_1) + B_2(r_1) \Big] \Big|\Big|\, \hat{c}^{k-1} - v^{k-1} \,\Big|\Big| \;\Big|\Big|\, \hat{c}^k - v^k \,\Big|\Big|$$

$$\geq \Big[a(r_1) + 2\tau - \rho\, (\epsilon_v\, B_1(r_1) + B_2(r_1) - 2b(r_1)) \Big] \Big|\Big|\, v^k - \hat{c}^k \,\Big|\Big|^2$$

$$- \rho\, (\epsilon_v\, B_1(r_1) + B_2(r_1))\,]\, \Big|\Big|\, v^{k-1} - \hat{c}^{k-1} \,\Big|\Big|^2 \tag{4.139}$$

Furthermore, (Brdyš and Roberts, 1987; Lin, Wang and Roberts, 1989b),

$$L_*(v^k) - L_*(v^{k+1}) \geq \epsilon_v L_*'(v^k)(v^k - \hat{c}^k) - \frac{1}{2}\epsilon_v^2 A_*(r_1) \left|\left| v^k - \hat{c}^k \right|\right|^2 \tag{4.140}$$

Substituting (4.139) into (4.140) and choosing τ such that :

$$\tau - \rho (B_2(r_1) - b(r_1)) \geq 0 \tag{4.141}$$

yields :

$$\tilde{L}_*(v^k,v^{k-1},r_1,\rho) - \tilde{L}_*(v^{k+1},v^k,r_1,\rho) \geq \epsilon_v \left[a(r_1) - \epsilon_v \left(\frac{1}{2} A_*(r_1) + 2\rho B_1(r_1) \right) \right]$$

$$\times \left|\left| v^k - \hat{c}^k \right|\right|^2 \tag{4.142}$$

Using (4.127), (4.125) and (4.123) we obtain from (4.141) the required inequality :

$$\tilde{L}_*(v^k,v^{k-1},r_1,\rho) - \tilde{L}_*(v^{k+1},v^k,r_1,\rho) \geq \epsilon_v \left[a(r_1) - \epsilon_v \left(\frac{1}{2} A_*(r_1) + 2\rho B_1(r_1) \right) \right]$$

$$\times \left|\left| v^k - \hat{c}^k \right|\right|^2$$

$$\geq \delta \left[a(r_1) - \frac{1}{2} \beta (v^k) \left(A_*(r_1) + 4\rho B_1(r_1) \right) \right] \left|\left| v^k - \hat{c}^k \right|\right|^2$$

$$\geq \frac{1}{2} \delta\epsilon (A_*(r_1) + 4\rho B_1(r_1)) \left|\left| v^k - \hat{c}^k \right|\right|^2 \tag{4.143}$$

Hence, (4.128) indeed holds. Moreover, it is seen from (4.143) that inequality (4.128) strictly holds if v^{k+1} is different from v^k. Therefore, $\tilde{L}_*(\cdot , \cdot ,r_1,\rho)$ is qualified to be a Zangwill function (Zangwill, 1969) with respect to the algorithmic mapping Φ defined in (4.100). It can be proved, following the same process as used by Brdyš and Roberts (1987) and Lin, Kambhampati and Roberts (1989), that the algorithmic mapping Φ is closed. Furthermore, for any given r_1 and r_3 $C(r_1,r_3)$ is compact and $v^k \in C(r_1,r_3)$, k=1, 2 ,... if $v^0 \in S(v^*,r)$. Therefore, all the assumptions for Zangwill's Convergence Theorem are satisfied (Zangwill, 1969). From Zangwill's Convergence theorem, we know that algorithm AL3 is convergent. Finally, the uniqueness of the cluster point of the sequence follows from the strict convexity of $\tilde{L}_*(\cdot , \cdot ,r_1,\rho)$ and the uniqueness of v^*.

QED

Comparing the assumptions made in Theorems 4.8 and 4.9 one may observe that the assumption (iv) :

$$b(r) \geq B_2(r) \tag{4.144}$$

in Theorem 4.8 is no longer existent in Theorem 4.9. This means that the convergence conditions of AL3 are significantly relaxed because assumption (iv) is in fact a key condition imposed on the real process.

4.2.2 Double Loop ISOPE Algorithms

It is proved by Lin, Hendawy and Roberts (1988b), subject to nearly the same assumptions made by Brdyś, Abdullah and Roberts (1986) for the convergence of a single loop ISOPE algorithm, that the double loop algorithm AL4 is locally convergent. The convergence theorem as well as its proof may be found in the given reference (Lin, Hendawy and Roberts, 1988b).

As mentioned in Section 4.1.2b, the outer loop iteration of the double loop algorithm AL5 is equivalent to the iteration of the centralised MTS method. It is also observed in the description of the algorithm that the inner loop iteration is equivalent to the iteration of the single loop ISOPE algorithm AL3 if the model-based optimisation problem of AL5 is treated as the real optimisation problem HROP for AL3. Therefore, subject to similar conditions for the convergence of the improved version of the MTS method (Lin, Han, Roberts and Wan, 1989), the outer loop iteration of the double loop ISOPE algorithm AL5 is globally convergent. It can also be proved by referring to Theorem 4.8 that the inner loop iteration of the algorithm is locally convergent.

4.3 Summary

Optimality and convergence of the single and double loop ISOPE algorithms (AL1$-$AL5) presented in Chapter 3 is thoroughly investigated. It is proved in this chapter subject to mild conditions that all these algorithms are optimal in the sense that every algorithmic solution satisfies the Kuhn-Tucker conditions for the real optimisation problem.

Going through the convergence analysis presented in this chapter, it is realised that the convergence conditions for double loop ISOPE algorithms are generally easier to satisfy than those for single loop ISOPE algorithms. A possible explanation for this is that at the expense of an increase in off-line calculation double loop algorithms have, in a sense, separated set-point iteration from interaction coordination resulting in a favourable situation for outer loop convergence.

It is also noted in the convergence analysis that by introducing variable augmentation it is possible to further relax the convergence conditions of the algorithms. In fact, the local convergence feature of the single loop algorithm AL1 can even be made global by increasing the penalty coefficient for the variable augmentation term. The reason for this is that the introduction of augmentation terms convexifies the real problem. The convergence results achieved through theoretical analysis on the hierarchical optimising control algorithms, presented in Chapter 3, may also be verified by using computer simulation techniques, as shown in the following chapter.

CHAPTER 5

SIMULATION STUDIES FOR HIERARCHICAL ISOPE ALGORITHMS

5.1 INTRODUCTION

It is usually considered important either from an industrial application point of view or from an algorithm development point of view that any newly developed optimising control algorithm should be tested in a simulated environment before it is applied to a real situation.

This is achieved by using a computer simulation technique in which the real process is simulated by a set of non-linear simultaneous equations. For each given controller set-point, the solution of these simultaneous equations is computed and treated as the real process measurements.

Three important factors are usually included in the analysis of the simulation results, which are : (a) convergence features of the algorithm; (b) algorithm efficiency; and (c) algorithm sensitivity. In the convergence feature analysis, the convergence of the performance index or augmented Lagrangian, the controller set-point vector and the coordinator, is investigated through a careful study of the corresponding convergence curves. The algorithm efficiency is characterised by the on-line efficiency and the off-line efficiency of the algorithm. Finally, the algorithm sensitivity is studied by perturbing the operation parameters such as iterative gains.

The purpose of this chapter is to test the presented hierarchical optimising control algorithms using computer simulation techniques and to compare these algorithms so that general guidelines to the algorithm specification in the practice of optimising control can be obtained.

5.2 DESCRIPTION OF THE EXAMPLES

The hierarchical ISOPE algorithms presented in Chapter 3 (AL1—AL5) have been tested, using a computer simulation technique, against three numerical examples. The three different testing examples are described mathematically as follows :

EXAMPLE 5.1

This example consists of two subsystems. Although the system mapping of this example is linear, the problem is rather difficult to solve because the inequality constraints are non-linear and tight at the optimum. The model and reality take the form :

$$y_{11} = 1.3c_{11} - 0.5c_{12} + 2u_{11} + \alpha_{11}$$

$$y_{21} = 1.2c_{21} - c_{22} + u_{21} + \alpha_{21}$$

$$y_{22} = 2.2c_{22} - 0.6c_{23} - u_{22} + \alpha_{22}$$

$$y^*_{11} = 1.4c_{11} - 0.6c_{12} + 1.8u_{11}$$

$$y^*_{21} = 1.3c_{21} - 1.1c_{22} + 1.1u_{21}$$

$$y^*_{22} = 2.3c_{22} - 0.7c_{23} - 1.1u_{21}$$

The performance index is a summation of the subsystem indices :

$$Q_1 = (y_{11} - 1)^2 + c^2_{11} + c^2_{12}$$

$$Q_2 = 2(y_{21} - 2)^2 + (y_{22} - 3)^2 + c^2_{21} + c^2_{22} + c^2_{23}$$

$$Q = Q_1 + Q_2$$

The subsystem constraints are :

$$\mathbb{CU}_1 = \left\{ (c_1 , u_1) \in \mathbb{R}^3 : | c_{1i} | \leq 1 , i = 1, 2 \text{ and } 0.8 - c_{12} - 0.6u_{11} \geq 0 \right\}$$

$$\mathbb{CU}_2 = \left\{ (c_2 , u_2) \in \mathbb{R}^4 : | c_{2i} | \leq 1 , i = 1, ..., 3 \text{ and } \right.$$

$$\left. 2.04 + 1.5u^2_{21} - c^2_{21} - c^2_{22} - c^2_{23} \geq 0 \right\}$$

The interaction constraints are :

$$u_{11} = y_{21}$$

$$u_{21} = y_{11}$$

EXAMPLE 5.2

This is a larger example consisting of three subsystems. It may be seen from the following simulation studies that a hierarchical optimisation scheme is more favourable than a centralised one for this problem. The model and reality of the example are :

$$y_{11} = 1.3c_{11} - 0.5c_{12} + 2u_{11} + \alpha_{11}$$

$$y_{21} = 1.2c_{21} - c_{22} + u_{21} + \alpha_{21}$$

$$y_{22} = 2.2c_{22} - 0.6c_{23} - u_{22} + \alpha_{22}$$

$$y_{31} = c_{31} + 2c_{32} - 4.5u_{31} + \alpha_{31}$$

$$y^*_{11} = 1.4c_{11} - 0.6c_{12} + 1.8u_{11}$$

$$y^*_{21} = 1.3c_{21} - 1.1c_{22} + 1.1u_{21}$$

$$y^*_{22} = 2.3c_{22} - 0.7c_{23} - 1.1u_{22}$$

$$y^*_{31} = 0.8c_{31} + 2.5c_{32} - 4.2u_{31}$$

The performance index is :

$$Q = Q_1 + Q_2 + Q_3$$

where

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

$$Q_2 = 2(y_{21} - 2)^2 + (y_{22} - 1)^2 + c_{21}^2 + c_{22}^2 + c_{23}^2$$

$$Q_3 = (y_{31} - 3)^2 + c_{31}^2 + 1.5\, c_{32}^2$$

The subsystem constraints are :

$$\mathbb{CU}_1 = \left\{ (c_1 , u_1) \in \mathbb{R}^3 : | c_{1i}| \leq 1 , i = 1, 2 \text{ and } 0.8 - c_{12} - 0.6\, u_{11} \geq 0 \right\}$$

$$\mathbb{CU}_2 = \left\{ (c_2 , u_2) \in \mathbb{R}^5 : | c_{2i}| \leq 1 , i = 1, ..., 3 \text{ and } \right.$$

$$\left. - c_{21}^2 - c_{22}^2 - c_{23}^2 + 1.5 u_{21}^2 + 2.04 \geq 0 \right\}$$

$$\mathbb{CU}_3 = \left\{ (c_3 , u_3) \in \mathbb{R}^3 : | c_{3i}| \leq 1 , i = 1, 2 \text{ and } 2.0 - c_{31} - 2c_{32} + u_{31} \geq 0 \right\}$$

The interaction constraints are :

$$u_{11} = y_{21}$$

$$u_{21} = y_{11}$$

$$u_{22} = y_{31}$$

$$u_{31} = y_{22}$$

EXAMPLE 5.3

This is an example consisting of three subsystems. Unlike Example 5.2 the system mapping in this example is non-linear. Hence, this example is the most difficult one amongst the three presented examples. The real system is described by the following equations :

$$y^*_{11} = 1.4c_{11} - 0.6c_{12} + 0.2u_{11} + 0.15\, c_{12}u_{11}$$

$$y^*_{21} = 1.3c_{21} - 1.1c_{22} + 0.1u_{21} + 0.1c_{22}u_{21}$$

$$y^*_{22} = 2.3c_{22} - 0.7c_{23} + c^2_{23} - 0.2u_{11} - 0.1u_{22}$$

$$y^*_{31} = 0.8c_{31} + 2.5c_{32} - 0.1u_{31} + 0.1c_{31}u_{31}$$

The performance index is a summation of the subsystem indices :

$$Q_1 = (y_{11} - 1)^2 + c^2_{11} + c^2_{12}$$

$$Q_2 = 2(y_{21} - 2)^2 + (y_{22} - 1)^2 + c^2_{21} + c^2_{22} + c^2_{23}$$

$$Q_3 = (y_{31} - 3)^2 + y^2_{21} + 2y^2_{22} + 3.6y_{31} + 1.3y^2_{11} + c^2_{31} + 1.5c^2_{32}$$

$$Q = Q_1 + Q_2 + Q_3$$

The subsystem constraints are :

$$CU_1 = \left\{ (c_1 , u_1) \in \mathbb{R}^3 : |c_{1i}| \le 1 , i = 1, 2 \text{ and } -0.3 - (-0.6 + c_{11})\, u_{11} \le 0 \right\}$$

$$CU_2 = \left\{ (c_2 , u_2) \in \mathbb{R}^5 : |c_{2i}| \le 1 , i = 1, ..., 3 \text{ and } -2.0 - 1.5u^2_{21} + c^2_{21} + c^2_{22} + c^2_{23} \le 0 \right\}$$

$$CU_3 = \left\{ (c_3 , u_3) \in \mathbb{R}^3 : |c_{3i}| \le 1 , i = 1, 2 \text{ and } 0.3 - u_{31} - 0.5c_{31}u_{31} \le 0 \right\}$$

The interconnection of the subsystems assumes a loop structure as follows :

$$u_{11} = y_{21}$$

$$u_{21} = y_{22}$$

$$u_{22} = y_{31}$$

$$u_{31} = y_{11}$$

A process model required by the Single Loop algorithm AL1 is chosen as follows :

$$y_{11} = 1.3c_{11} - 0.5c_{12} + 0.2u_{11} + \alpha_1$$

$$y_{21} = 1.2c_{21} - 1.0c_{22} + 0.1u_{21} + \alpha_2$$

$$y_{22} = 2.2c_{22} - 0.6c_{23} - 0.12u_{22} + \alpha_3$$

$$y_{31} = 1.0c_{31} + 2.0c_{32} - 0.1u_{31} + \alpha_4$$

5.3 SIMULATION RESULTS

Simulation results of the algorithm AL1, with and without augmentation, for Examples 5.1, 5.2 and 5.3, with the best gain for different penalty coefficients, are shown in Tables 5.1 and 5.2. Figures 5.4 and 5.5 show the sensitivity of the algorithm AL1 to the iterative gain varying with the augmentation coefficient τ in solving Example 5.1; while Figures 5.1, 5.2 and 5.3 represent the convergence of the real performance, the price coordination vector and the controller set-point vector during the course of iteration. Figures 5.6, 5.7, 5.8, 5.9 and 5.10 are the equivalents to Figures 5.1, 5.2, 5.3, 5.4 and 5.5, representing the behaviour of the algorithm, when solving the second example.

The simulation results of algorithms AL3 and AL5, with the best gain factor, are shown in Table 5.2 and Table 5.3. Table 5.2 and Table 5.3 also show comparisons of convergence between AL1 and AL3, and AL3 and AL5, respectively. Figures 5.11 and 5.12 provide a comparison of convergence between algorithms AL1 and AL3. Figures 5.14 and 5.15 show a comparison of convergence between algorithms AL3 and AL5. The simulation results for algorithm AL4 are given by Lin, Hendawy and Roberts (1988b).

An inspection of Table 5.1 and Figures 5.4, 5.5 and 5.9, 5.10 shows that the sensitivity of algorithm AL1 to the selection of the gain factors is significantly reduced as the penalty coefficient in the augmented Lagrangian increases. Just as concluded in the convergence analysis of algorithm AL1 (Theorem 4.5), the introduction of the augmented Lagrangian to the model-based optimisation problem relaxes the convergence conditions of the algorithm, as is also seen in Table 5.1 and Figures 5.4 and 5.9, in that the number of real system iterations is significantly reduced after augmentation. The reason for this is that augmentation convexifies the problem and, therefore, improves the convergence behaviour of the algorithm. It is worth mentioning that mainly due to the global feedback structure of AL1, the real performance converges in such a desirable manner that curves shown in Figures 5.1 and 5.6 appear to be sharply descendant and, after a few iterations, the optimum performance is nearly achieved.

It is seen in Table 5.2 that algorithm AL3 converges faster than algorithm AL1 even without augmentation. The reason for this is that the convergence of the modifier coordination mechanism tends to be faster than the price coordination method; hence, little improvement is made by applying an augmentation technique. It is, however, worth mentioning that the introduction of the augmentation technique may help to relax the applicability conditions of the algorithm and to widen the range for selecting the gain factor as shown in Figure 5.13. It is interesting to observe in Figures 5.11 and 5.12 that although algorithm AL3 requires less iterations than algorithm AL1 to reach the optimum, the convergence of the algorithm in the first few iterations appears to be slower than in algorithm AL1. This means that AL3 has a better tail convergence behaviour than AL1, while the latter has more favourable convergence features in the first few iterations. A zigzag is spotted in Figure 5.11 during the course of convergence for the presented algorithm. A possible explanation is that in Theorems 4.8 and 4.9, monotonic convergence is ensured only for the Augmented Lagrangian $\tilde{L}_*$ rather than the real performance index.

An inspection of Tables 5.2 and 5.3, and Figures 5.11—5.16, shows that the smallest number of real system iterations is required by algorithm AL5, compared to that required for algorithms AL1 and AL3. The reason for this is that the convergence of the double iterative algorithm AL5 is dominated by its outer loop iterations and the convergence feature of its outer loop iteration is basically the same as that of the improved MTS method (centralised case). It is also observed that this advantage is obtained at the cost of an increase in the number of inner loop iterations which is performed in an off-line manner.

5.4 COMPARATIVE STUDY

Based on the above analysis of the simulation results, the following conclusions are reached which can serve as general guidelines in the practice of optimising control.

5.4.1 On-line Efficiency

In terms of on-line efficiency, which is characterised by the required number of real system iterations, it is concluded that the double loop ISOPE algorithm AL5 is the most efficient of the optimising control algorithms investigated. It is, therefore, considered that the algorithm AL5 is the most suitable choice for cases where the process adjusting time is dominated by the number of its controller set-point changes, for instance when the transient of the process is slow (Lin, Hendawy and Roberts, 1988b; Lin, Kambhampati and Roberts, 1989).

5.4.2 Off-line Efficiency

It is well recognised that in some circumstances the total efficiency, which is expressed by the total number of algorithm iterations, is of particular importance, for instance when the transient of the process is fast and there is a limitation in the real time computing power. In these cases, since the off-line efficiency of an algorithm is of the same importance as the on-line efficiency, the single iterative algorithms AL1 and AL3 appear to be superior over the double iterative algorithm AL5. Although in most of the cases mentioned above, AL3 seems to be more efficient than AL1, there are situations for which AL1 is superior to AL3 (Amini, Lin and Roberts, 1990; Lin, Amini and Roberts, 1990).

5.4.3 Algorithm Sensitivity

It is concluded that, by introducing the augmentation technique, the algorithm sensitivity to the selection of the iterative gain factor is significantly reduced, resulting in an increase in the algorithm robustness. As is seen from the simulation results, the reduction in algorithm sensitivity is found to be of less significance for algorithms AL3 and AL5 than it is for algorithm AL1. This is a reason why the augmented version of AL1 is usually preferred to the standard version.

5.5 SUMMARY

In this chapter, the presented hierarchical single and double loop ISOPE algorithms have been tested against three numerical examples using computer simulation techniques. Emphasis has been put on issues such as convergence features, algorithm sensitivity and robustness which are considered essential in application of on-line optimising control. In order to derive general guidelines for algorithm specification in optimising control, a comparative study of the presented algorithms is provided.

Example	GP_{opt}	GV_{opt}	GP	GV	τ_1	τ_2	N	RQ_{opt}	Q_{opt}
	0.66	0.3	0.0−0.62	0.0−0.32	0.0	0.0	28	5.92767	5.928
1	0.87	0.3	0.0−1.0	0.0−0.5	0.5	0.5	21	5.9277	
	1.1	0.4	0.0−1.25	0.0−0.85	1.0	1.0	20	5.9277	
	1.5	0.55	0.0−1.7	0.0−1.2	2.0	2.0	22	5.9278	
	0.4	0.08	0.0−0.4	0.0−0.25	0.0	0.0	103	5.0557	5.057
2	1.03	1.2	0.0−1.4	0.0−1.25	1.0	1.0	38	5.05697	
	1.9	1.6	0.0−1.95	0.0−1.65	2.0	2.0	34	5.056	
	1.9	1.7	0.0−2.0	0.0−1.7	3.0	3.0	38	5.0567	

Table 5.1 Simulation results of AL1 for Examples 5.1 and 5.2

Gp_{opt} : Optimum gain for price

GV_{opt} : Optimum gain for input

τ_1 , τ_2 : Augmentation coefficients

N : Number of iterations

RQ_{opt} : Real optimum performance index

Q_{opt} : Optimum performance index

Algorithm	GV_{opt}	τ_1	τ_2	N	RQ_{opt}	Q_{opt}
AL1	0.45	0.5	0.5	22	12.3027	12.3
AL3	0.39	0.0	0.0	13	12.2993	12.3

Table 5.2 Simulation results of AL1 and AL3 for Example 5.3

GV_{opt} :	Optimum gain for input
τ_1 , τ_2 :	Augmentation coefficients
N :	Number of iterations
RQ_{opt} :	Real optimum performance index
Q_{opt} :	Optimum performance index

Algorithm	GV_{opt}	τ_1	τ_2	N	RQ_{opt}	Q_{opt}
AL3	0.39	0.0	0.0	13	12.2993	12.3
AL5	1.2	0.0	0.0	9	12.2976	12.3
AL5	1.2	0.5	0.5	10	12.2976	12.3

Table 5.3 Simulation results of AL3 and AL5 for Example 5.3

GV_{opt} : Optimum gain for input
τ_1 , τ_2 : Augmentation coefficients
N : Number of iterations
RQ_{opt} : Real optimum performance index
Q_{opt} : Optimum performance index

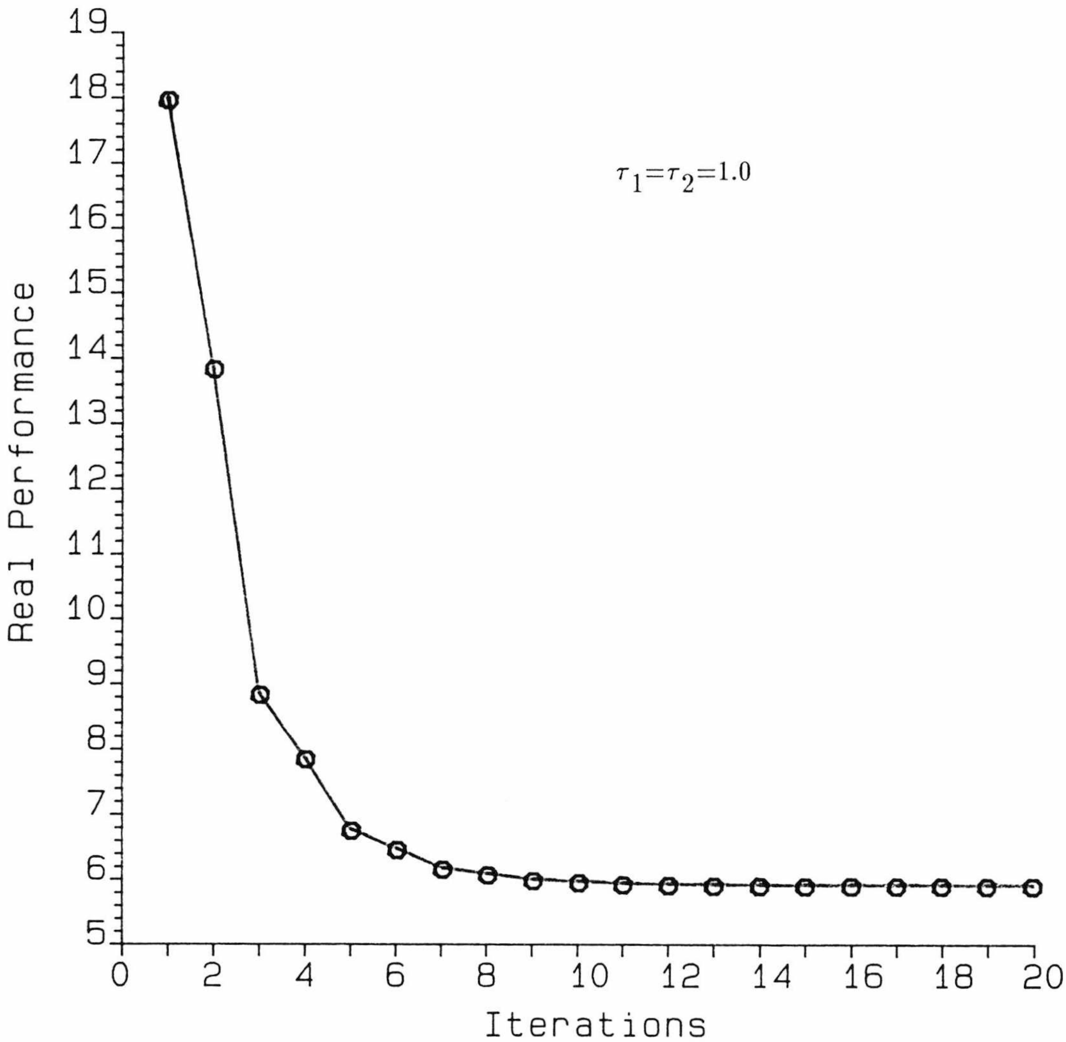

Figure 5.1 Convergence of Performance Index for AL1 (Example 5.1)

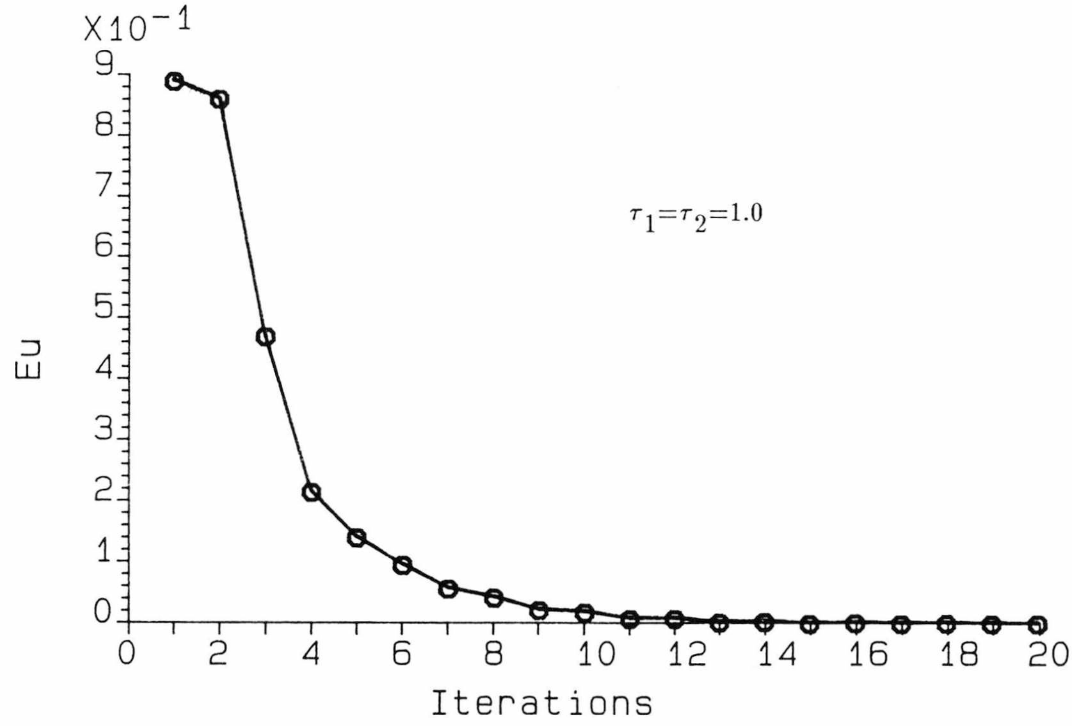

Figure 5.2 Convergence of Interaction Balance for AL1 (Example 5.1)

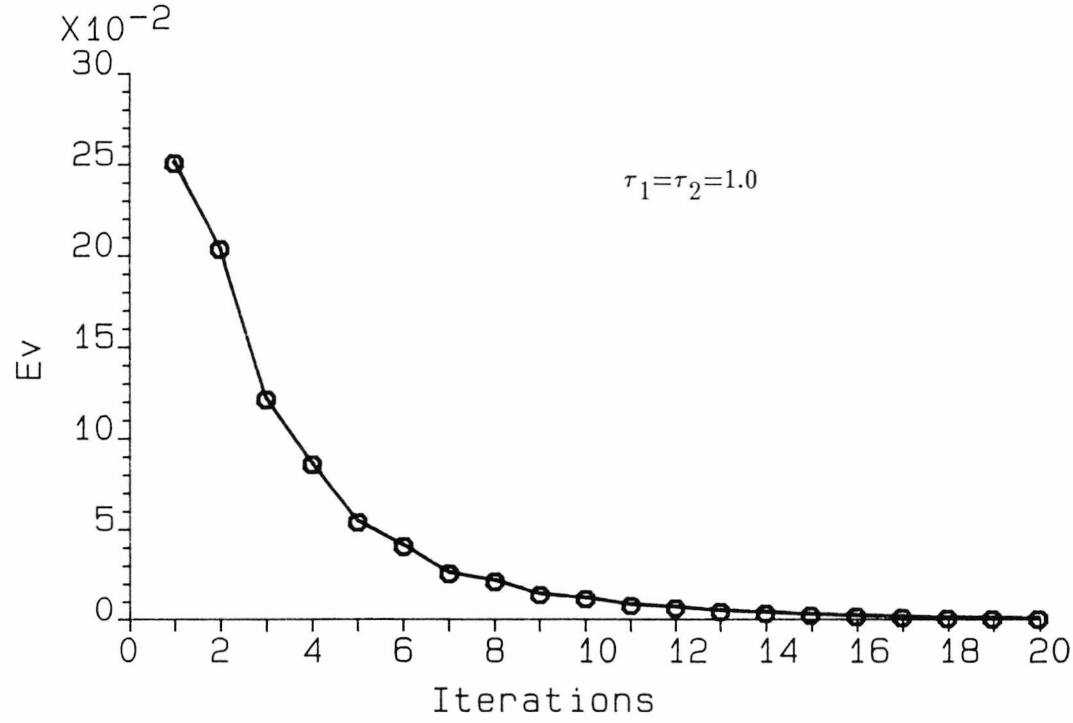

Figure 5.3 Convergence of Controller Set-Point for AL1 (Example 5.1)

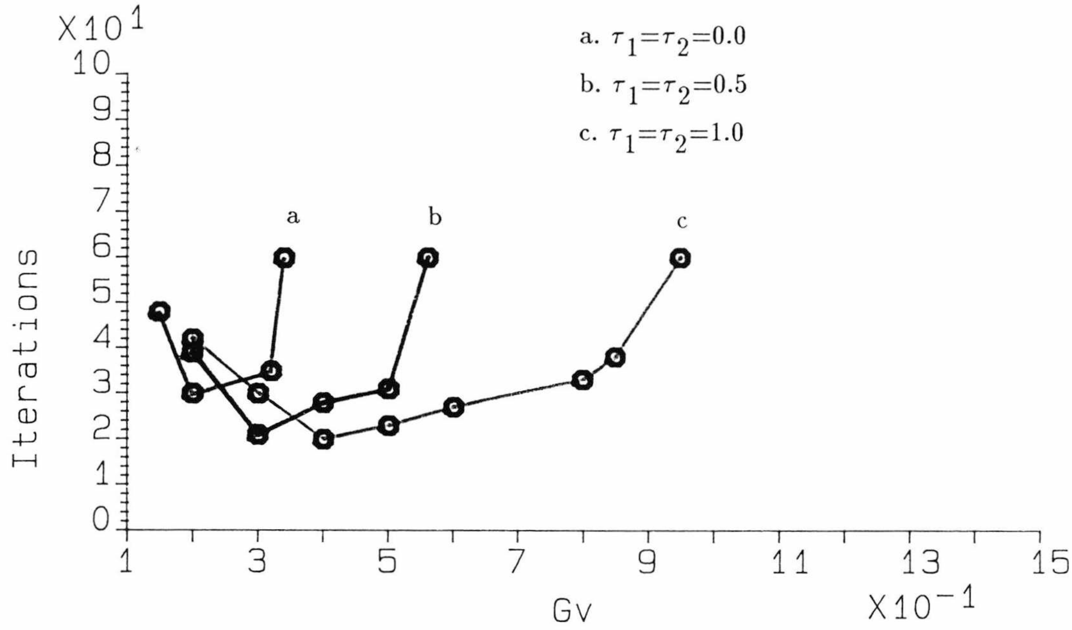

Figure 5.4 Sensitivity of Gain for Set-Points for AL1 (Example 5.1)

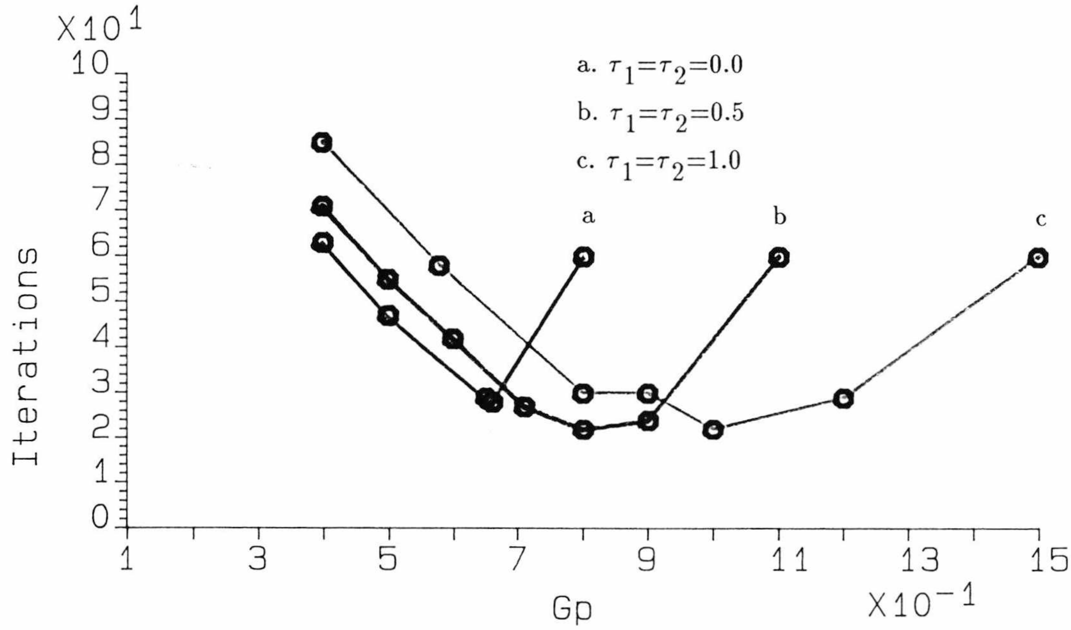

Figure 5.5 Sensitivity of Gain for Price for AL1 (Example 5.1)

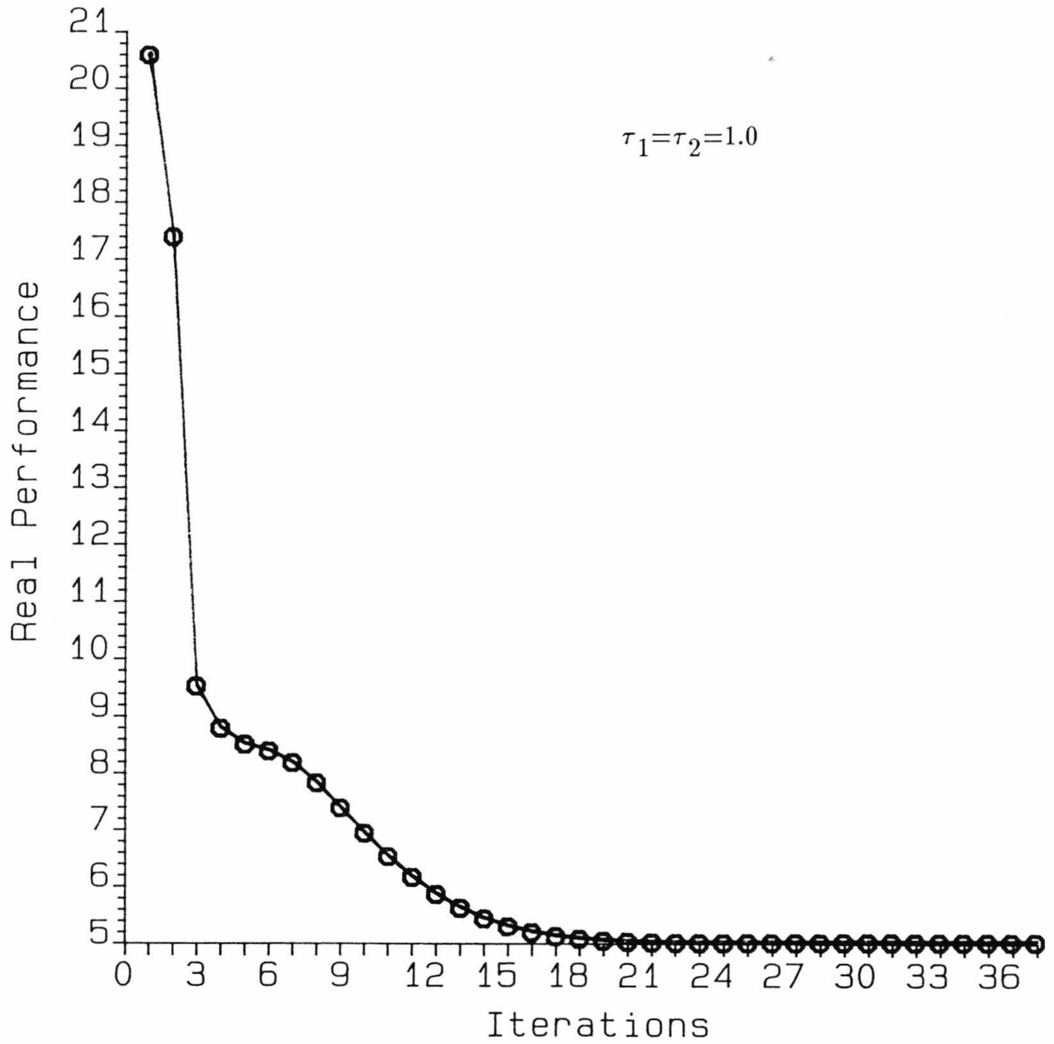

Figure 5.6 Convergence of Performance Index for AL1 (Example 5.2)

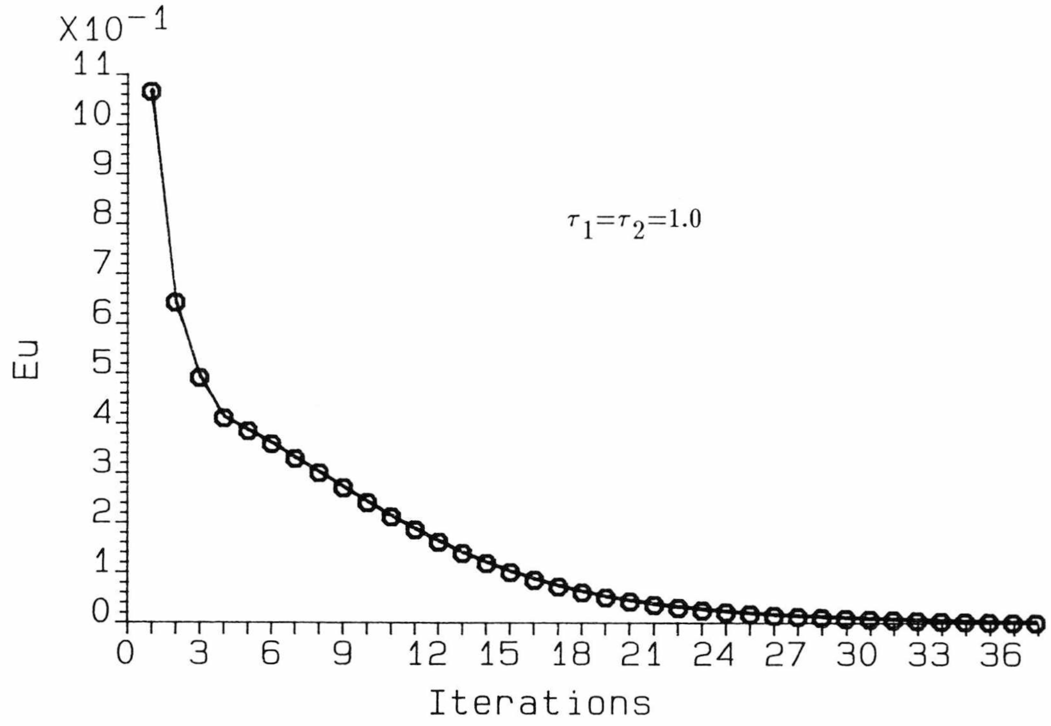

Figure 5.7 Convergence of Interaction Balance for AL1 (Example 5.2)

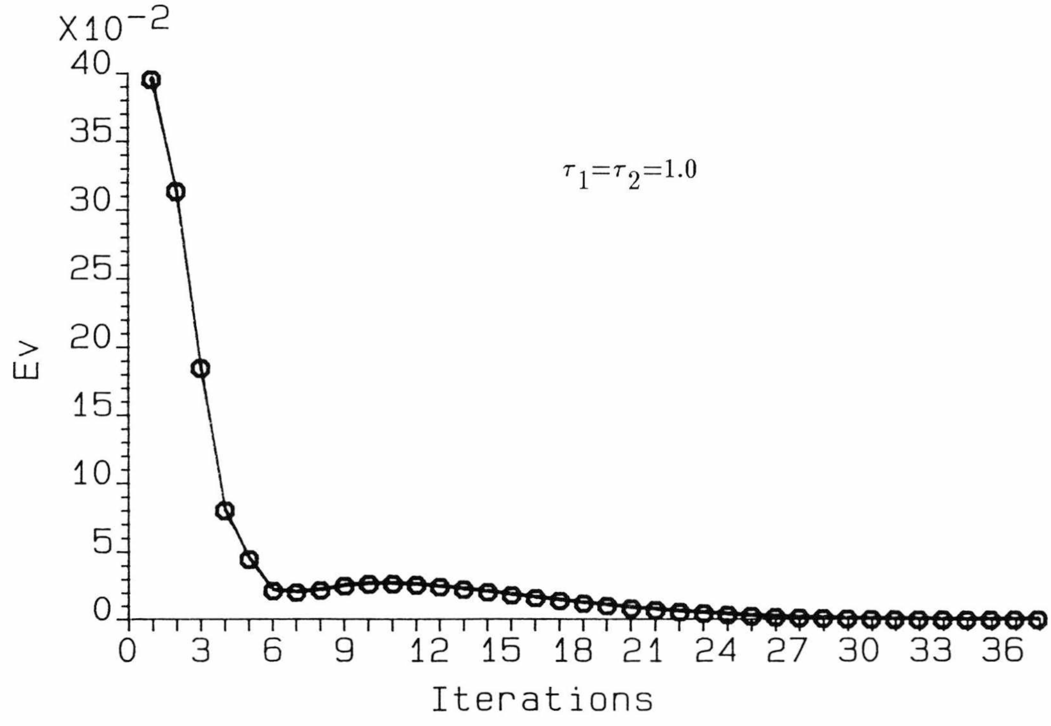

Figure 5.8 Convergence of Controller Set-Point for AL1 (Example 5.2)

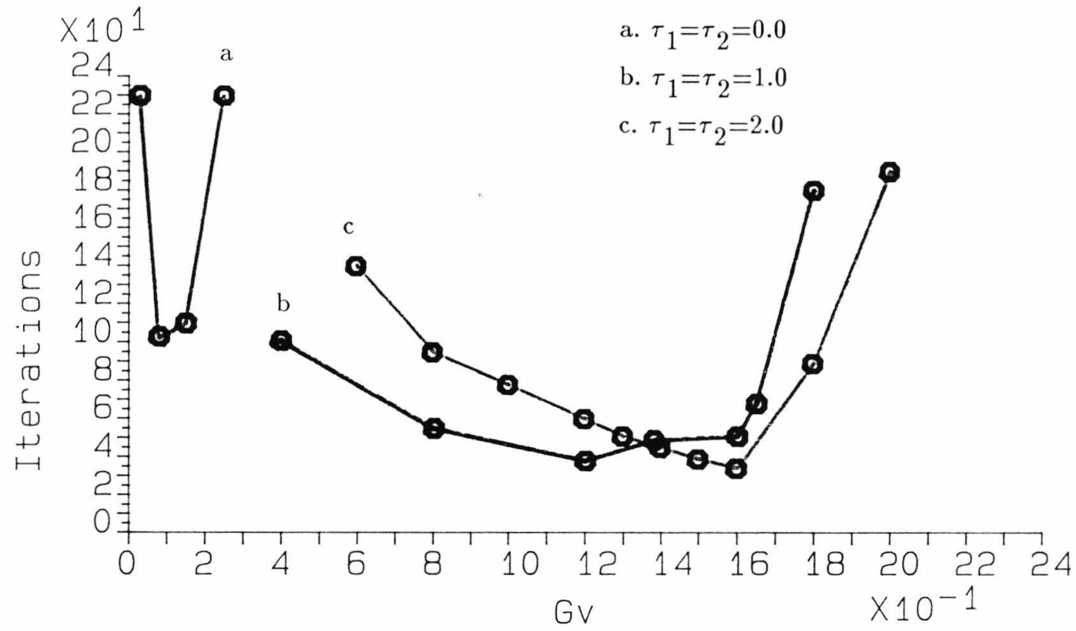

Figure 5.9 Sensitivity of Gain for Set-Point for AL1 (Example 5.2)

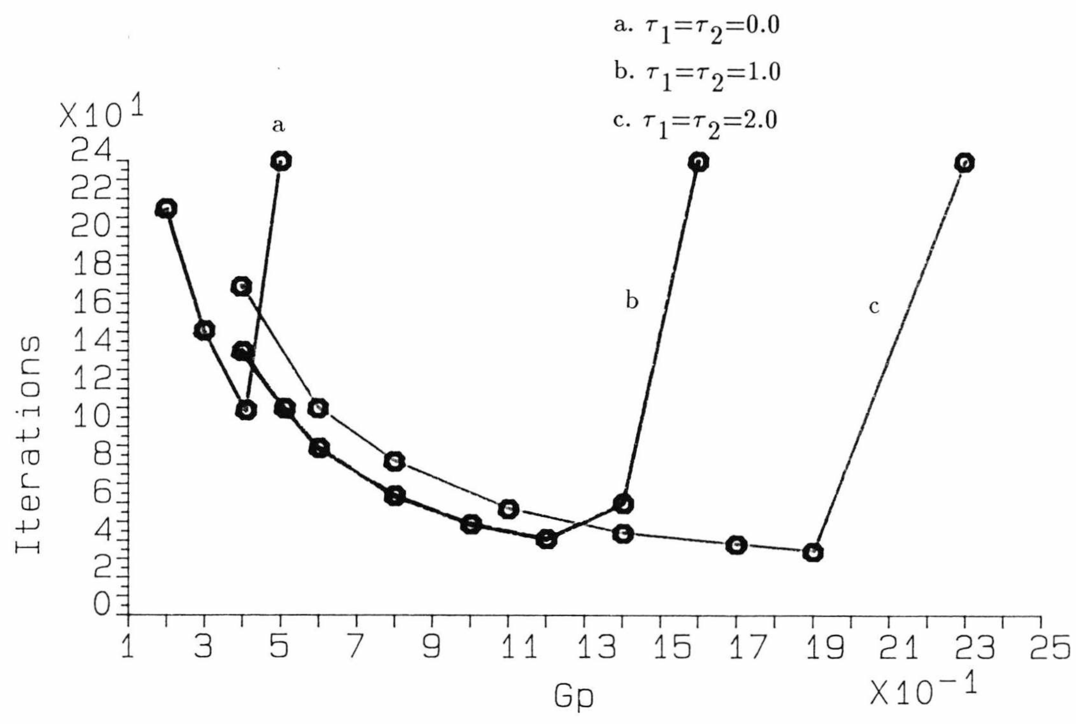

Figure 5.10 Sensitivity of Gain for Price for AL1 (Example 5.2)

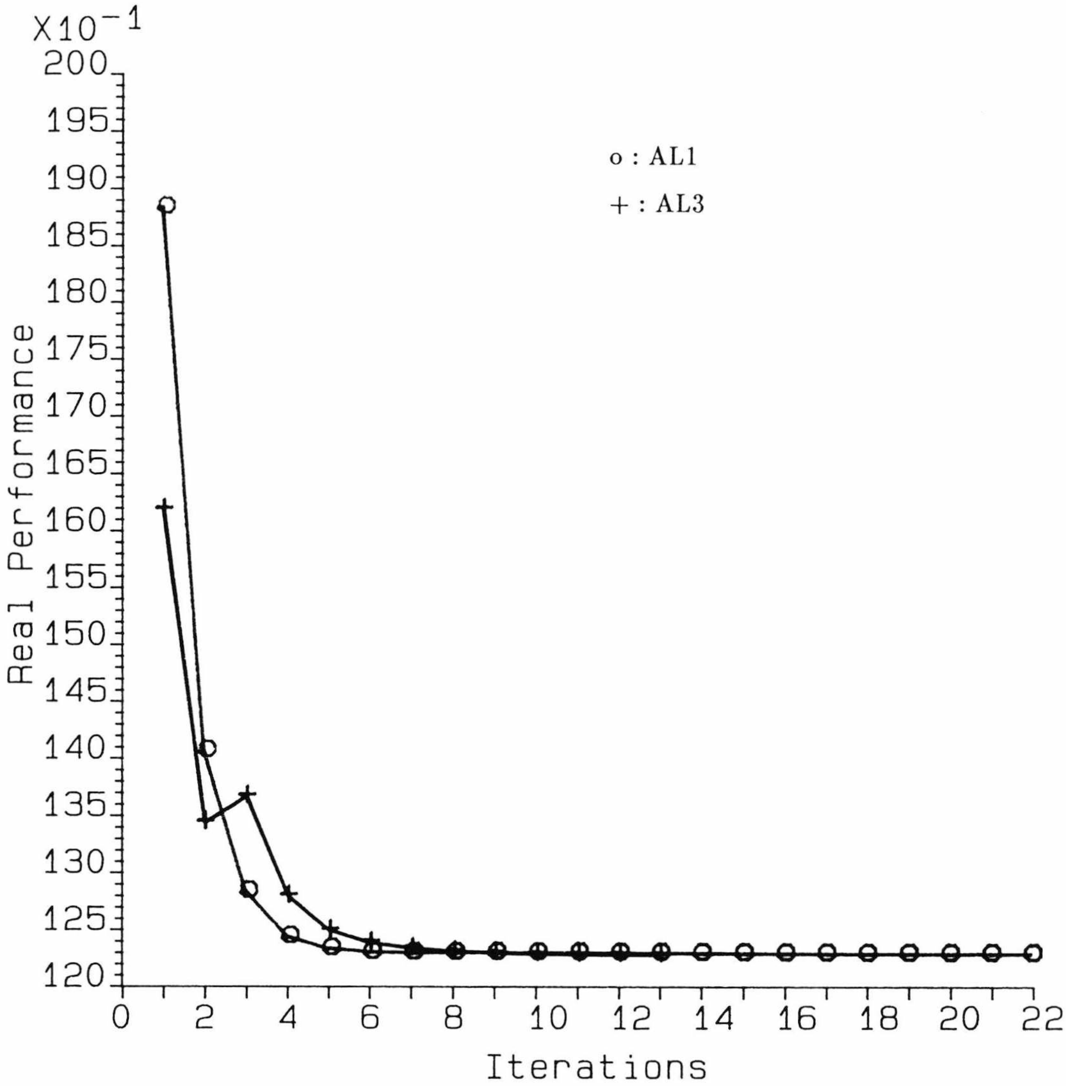

Figure 5.11 Comparison of Convergence between AL1 and AL3

for example 5.3 (Performance Index)

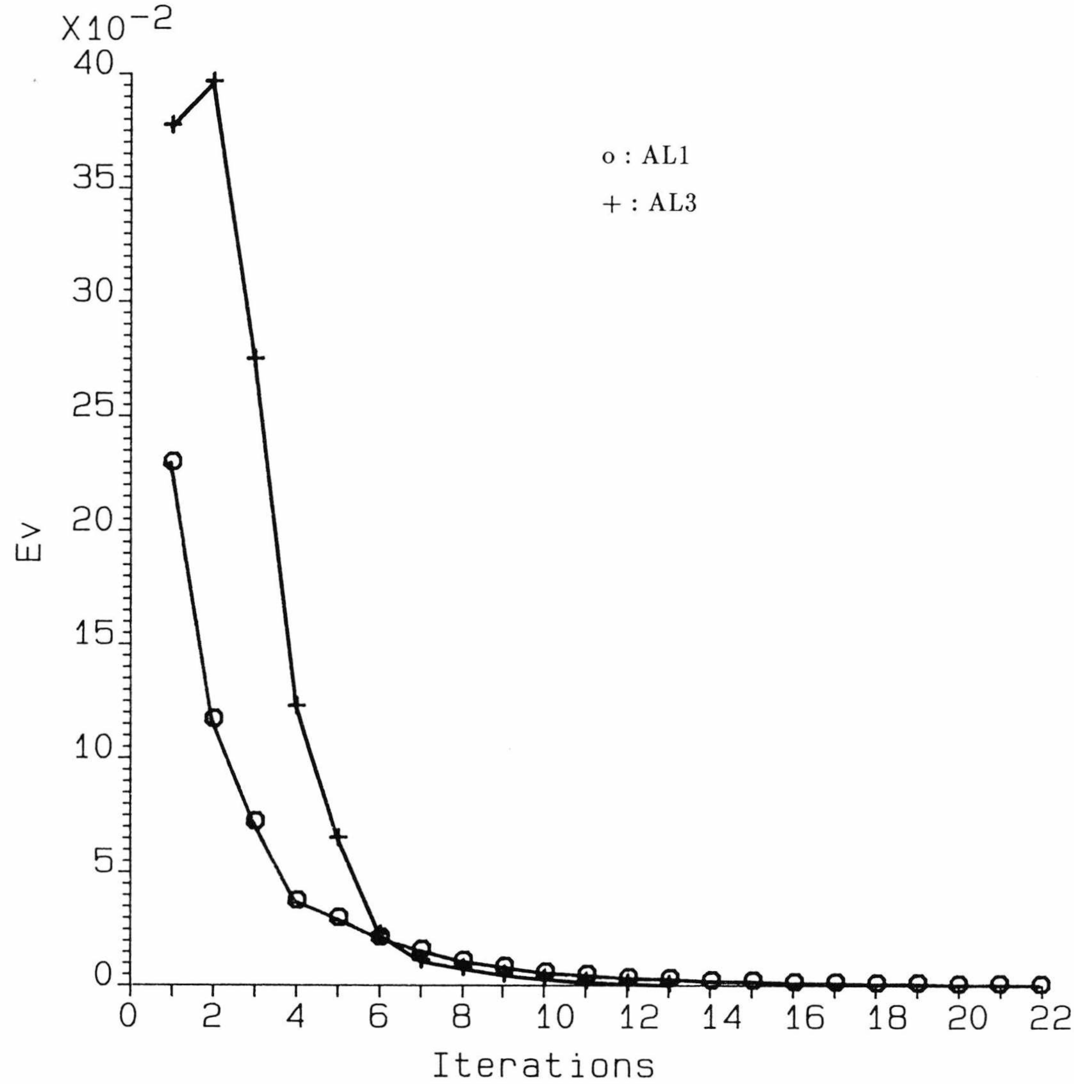

Figure 5.12 Comparison of Convergence between AL1 and AL3 for example 5.3 (Controller Set-Point)

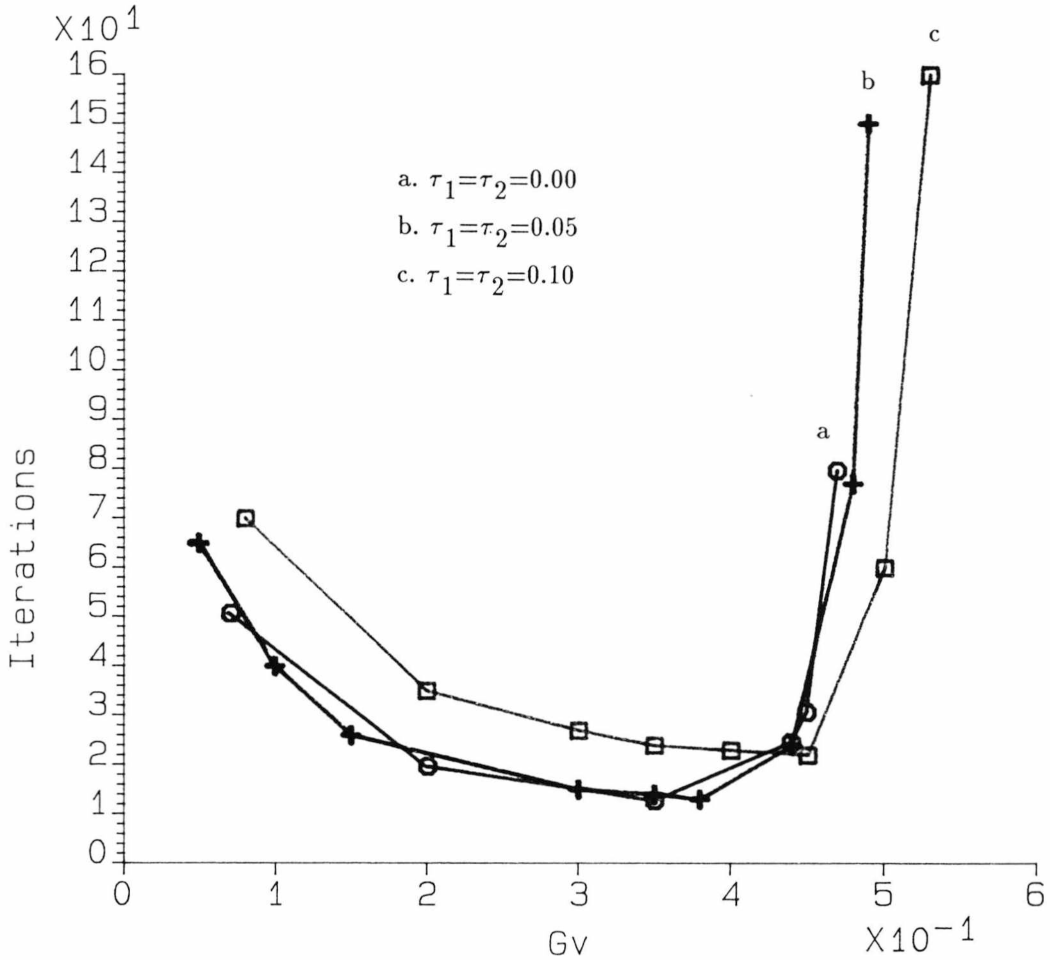

Figure 5.13 Sensitivity of Gain Factor for Controller Set-Point for AL3 (example 5.3)

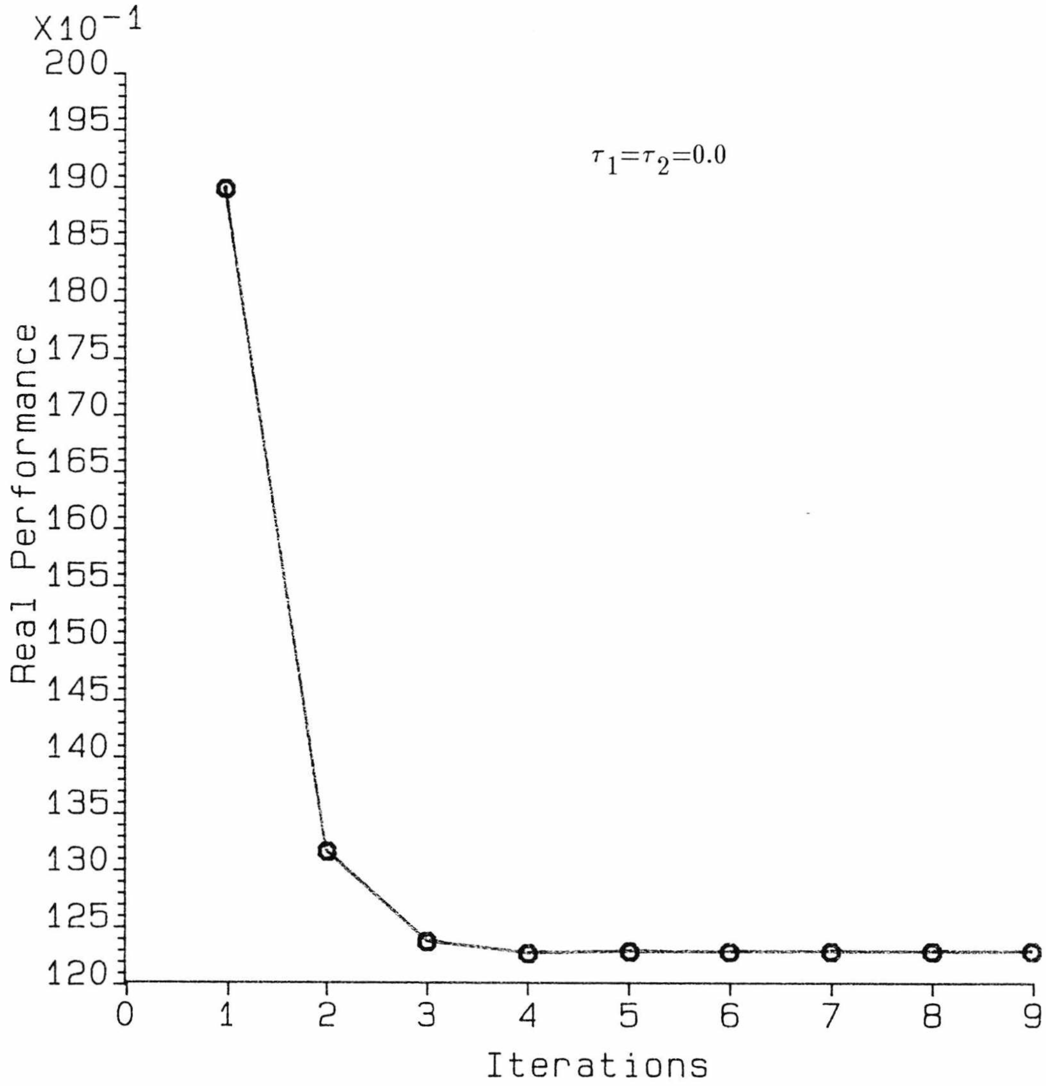

Figure 5.14 Convergence of Performance Index for AL5 (Example 5.3)

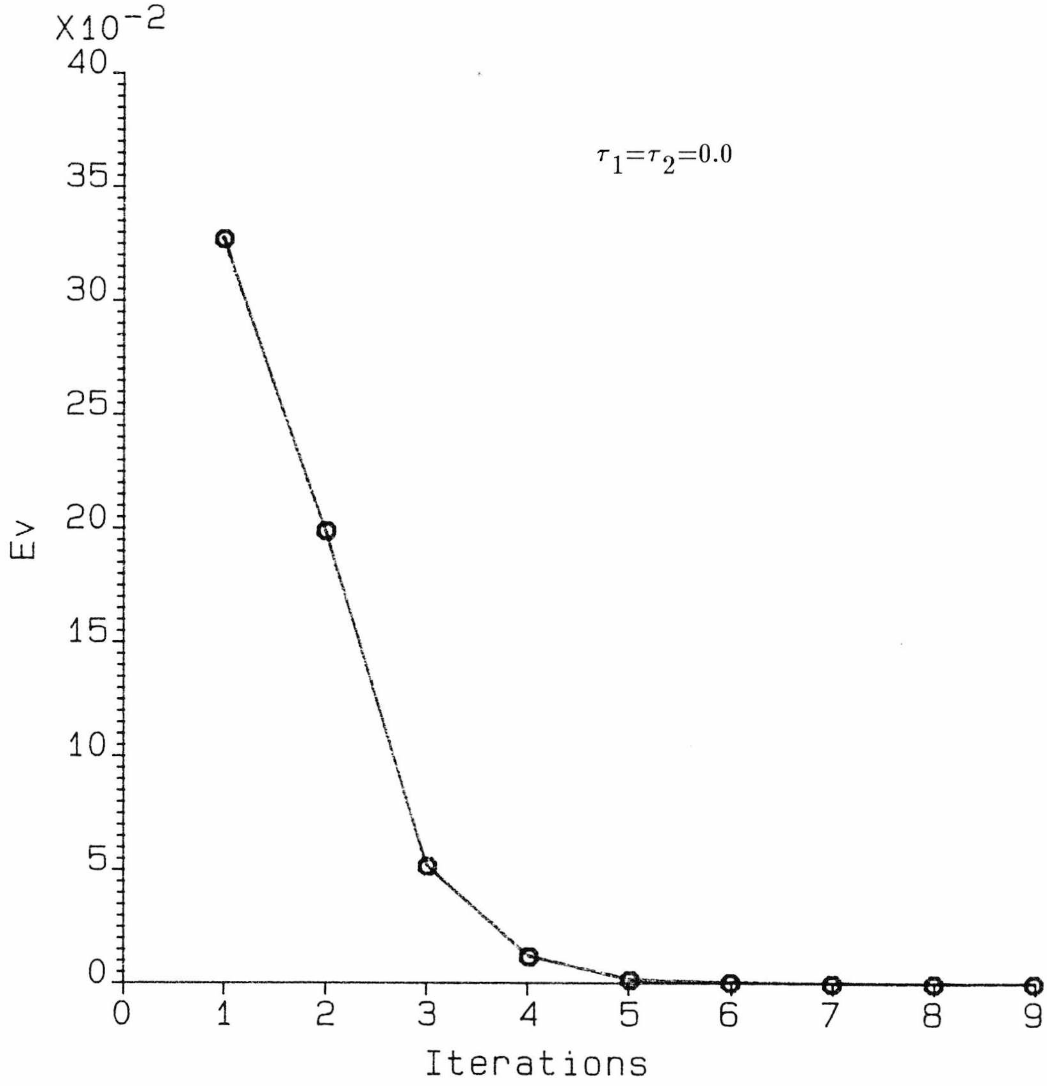

Figure 5.15 Convergence of Controller Set-Point for AL5 (Example 5.3)

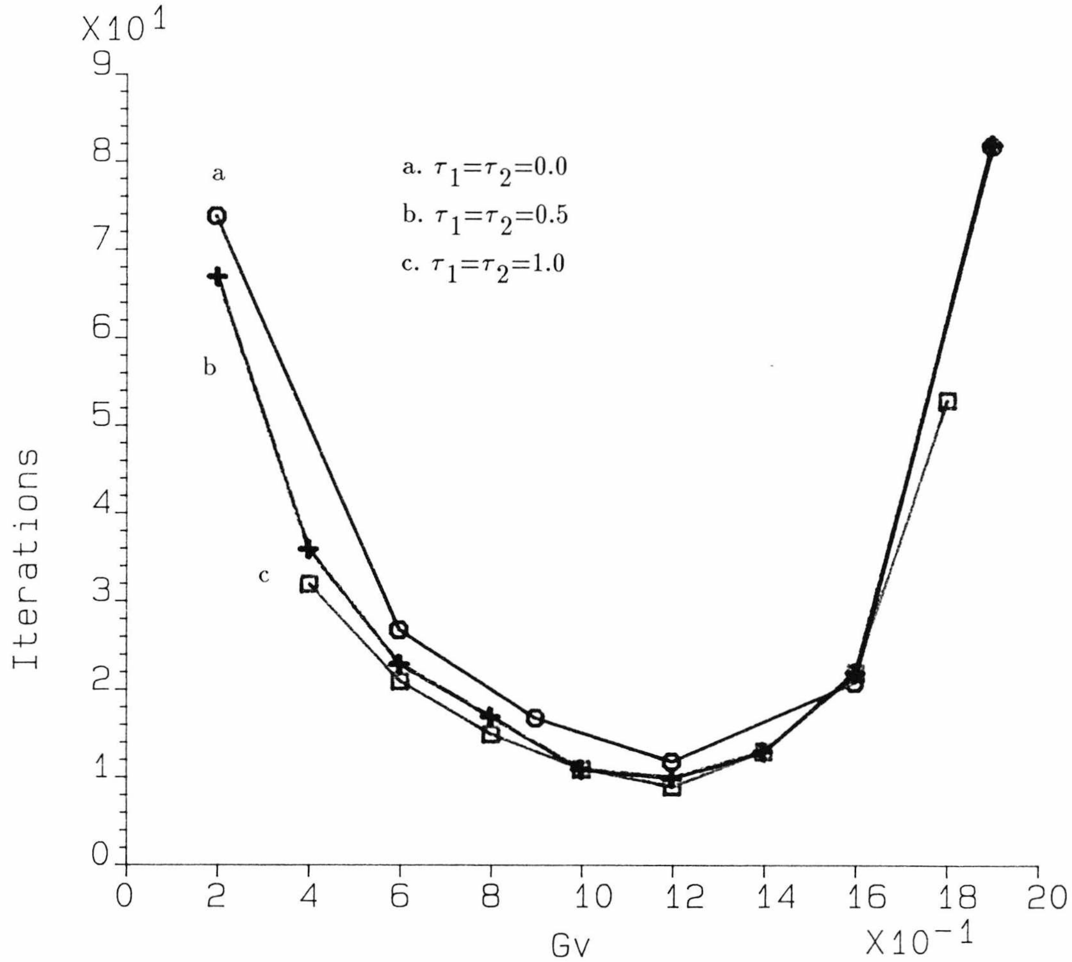

Figure 5.16 Sensitivity of Gain Factor for Controller Set-Point for AL5 (example 5.3)

CHAPTER 6

SOFTWARE DEVELOPMENT FOR PROCESS OPTIMISATION

6.1 INTRODUCTION

Important issues, such as algorithm specification, software configuration and user interface, in the software development of a Computer Aided Design (CAD) package for process optimisation are discussed in this chapter. The aim of this chapter is to give general guidelines for the design and development of such a software package.

During the past decade, a great number of novel on-line hierarchical optimising control algorithms have been developed and tested using computer simulation techniques. These algorithms may be adopted to form an algorithm bank for the package so that an appropriate optimising control algorithm can be chosen according to specific situations. In order to establish a proper algorithm bank for the package, certain criteria are needed for the purpose of algorithm specification. In the practice of on-line process operations control, a number of requirements from industry have been used as the criteria for assessing existing optimising control algorithms. These requirements are, therefore, considered the criteria of algorithm specification for this package.

In general, many optimising control algorithms are very similar in their structure. Therefore, it is worth while extracting a number of basic methods, techniques and approaches from the existing algorithms to form the basic elements of the algorithm bank. With these basic elements, one should be able to create any of the specified optimising control algorithms in the algorithm bank. This idea also fits in with the strong tendency of modular design in todays software development.

A suitable configuration for the aforementioned package should be of a modular design with proper interfacing with the user. The package is then broken down into units specified according to different basic functions of the optimising control algorithms to be included in the algorithm bank. These function units are linked by a management unit which will complete appropriate connections to form the required optimising control algorithm. The management unit also looks after the user interface.

6.2 CRITERIA FOR ALGORITHM SPECIFICATION

In order to establish the algorithm bank mentioned in Section 6.1, a number of criteria are worked out, based on the requirements from the industry, to cover the aspects which are considered important in building up a software package for on-line optimising control of industrial processes (Lin and Roberts, 1990).

Reliability

A desirable hierarchical optimising control algorithm should not introduce too much disturbance to the process when taking necessary measurements. For instance, the controller set-point perturbation technique for finding the process derivatives sometimes introduces too much disturbance because repeated controller set-point changes are required by this technique.

Robustness

A desirable hierarchical optimising control algorithm should be robust in the sense that a slight change in the model parameters or real process mapping should not cause divergence of the algorithm or affect significantly the optimality of the solution. This criterion becomes vital when the process is to be operated in a noisy environment or the process mapping is time dependent and difficult to follow exactly because in that case large model-reality differences are more likely to exist.

Efficiency

In order to keep up with any changing features of the process mapping, the total readjustment time for the optimising control algorithm to achieve the correct optimum should be minimised. Since the readjustment time of an optimising control algorithm depends on the number of controller set-point changes required to locate the optimum operating condition for the process, this number should be as small as possible.

Capability of dealing with large scale problems

A desirable hierarchical optimising control algorithm should be capable of dealing with large scale problems in an efficient manner. This criterion simply means that the coordination strategies adopted in the hierarchical optimising control algorithms should be very efficient and reliable since, in large scale optimisation, the Hessian structure of the problem is likely to be ill-conditioned, resulting in a slow convergence and possible numerical instability.

Optimality

A desirable hierarchical optimising control algorithm should be able to achieve the real optimum to a certain precision regardless of the model-reality differences. In practice, this criterion implies that the precision of the optimum achieved by the algorithm should not depend upon the model-reality differences because only for such cases can approximations be made without losing optimality of the algorithm.

Ease of use

A desirable hierarchical optimising control algorithm should be easy to implement. This means that the selected algorithms should not be too difficult to operate in the sense that there should not be too many user specified algorithm operating parameters and the algorithm efficiency should not be too sensitive to the selection of such operating parameters.

By and large it is difficult, if not impossible, to find an algorithm from existing candidates such that it can meet all the criteria mentioned above regardless of the application circumstances, and the selection of a suitable optimising control algorithm for a specific problem will depend upon the features of the problem. Therefore, the task of the algorithm specification for the above mentioned software package can only be fulfilled by specifying a number of optimising control algorithms which form an algorithm bank covering most practical situations.

6.3 SPECIFICATION OF OPTIMISING CONTROL ALGORITHMS

Since many optimising control algorithms are similar in structure, the best way of creating an algorithm bank is to extract some basic methods, techniques and approaches from the specified optimising control algorithms as the generating elements. With these elements, one can then generate the required algorithm bank consisting of all the specified algorithms. This idea also fits in with the tendency of using modular design in software development.

Based on the criteria listed in Section 6.2 and the above arguments, the following methods, techniques and approaches are considered to be the basic elements for generating the algorithm bank of the package.

I. Double Iterative Strategy

This iterative strategy aims at reducing as much as possible the number of the required controller set-point changes at the price of increasing off-line computations and, therefore, is recommended for the cases where the transient time of the process is slow because in these cases the number of controller set-point changes required to produce the true optimum becomes crucial.

II. Single Iterative Strategy

Unlike the double iterative strategy, this method consists of only one iterative loop and the efficiency counted by the algorithm includes both process iterations and off-line computations. Therefore, this method is recommended for the cases where the transient time of the process is very fast, such as compressors in chemical processes, because in these cases the number of controller set-point changes becomes less important than it would have been in the cases where the process is of slow transient nature.

III. Price Coordination Strategy

In the price coordination method, the lower level optimisation is arbitrated by the introduction of a price correction term so that the interaction balance is restored when the algorithm converges. One of the major advantages of this method is that it is simple to implement and easy to converge. It is, however, worth mentioning that in some cases, where the Hessian structure of the problem is ill-conditioned, the convergence of this method becomes very slow towards the end of the iterations.

IV. Modifier Coordination Strategy

In this approach, the overall optimising control problem is decomposed into independent local optimisation problems by employing a separable model structure in the hierarchical optimisation scheme. The interaction between subsystems is considered as a kind of model-reality difference and is, therefore, taken care of by the modifier. Since the coordination task is performed by the modifier which also compensates the model-reality difference, this approach is often found to be more efficient than the price coordination method. It is, therefore, recommended for the purpose of decomposition and coordination of hierarchical optimising control problems.

V. Standard and Improved Modified Two-Step Methods

The most desirable aspect of these two methods is that they are capable of producing the correct optimum operating condition of the process regardless of model-reality differences which may exist (Roberts, 1979; Brdyś and Roberts, 1987; Lin, Han, Roberts and Wan, 1989). These approaches are recommended in conjunction with different iterative strategies and coordination methods to form various novel hierarchical optimising control algorithms for different situations.

VI. Penalty Relaxation technique

This approach is specially designed for the purpose of handling output dependent inequality constraints which are often found in the practice of optimising control. The main idea of this approach is to relax the model-based optimisation problem by introducing slack variables in the linearised output dependent inequality constraints and to enforce the feasibility of the constraints by adding penalty terms of slack variables to the performance index in the model-based optimisation problem. Theoretical analysis and computer simulation studies show that this technique is excellent for coping with output dependent inequality constraints either in terms of efficiency or reliability (Lin, Kamphampati and Roberts, 1989). This technique is, therefore, recommended for forming, together with other optimising control methods, algorithms for solving optimising control problems with output dependent constraints.

VII. Variable Augmentation Technique

This technique was originally designed for the purpose of problem convexification. It is found that it does not only accelerate the convergence process, but also significantly reduces the sensitivity of the algorithm efficiency with respect to the iterative gain selection. Furthermore, it relaxes the applicability conditions of the algorithm due to the convexification of the problem. This technique is also recommended for use in conjunction with other optimising control methods according to the situation.

VIII. Derivative Acquisition Technique

This technique is used for obtaining the process derivative measurements which is required in the calculation of modifier. In this technique, the real process is perturbed by varying its controller set-points about the given values and the process derivatives are calculated by a finite difference approximation. Since the number of controller set-point changes for each derivative measurement is at least n=1, where n is the number of controller set-points, the overall process regulating time would be significantly prolonged if the process is of slow transient nature. Nevertheless, it remains to be an effective technique for fast processes in on-line process optimisation or for simulated process in CAD package.

6.4 FORMATION OF THE ALGORITHM BANK

Based on the basic elements listed in Section 6.3, we are ready to establish the algorithm bank for the package. This algorithm bank is organised in such a way that all the specified algorithms are classified into two categories. One is for centralised optimising control algorithms while the other is for hierarchical optimising control algorithms. A number of basic methods are listed below and a proper combination of them will form every specified algorithm in each category.

6.4.1. Centralised Optimising Control Algorithms

(A) Modified Two-Step (MTS) method with two options :

a. Standard MTS (Roberts, 1979);
b. Improved MTS (Lin, Han, Roberts and Wan, 1989);

(B) Penalty Relaxation (PR) technique (Lin, Kambhampati and Roberts, 1989).

(C) Variable Augmentation (VA) technique (Lin, Li, Wan and Roberts, 1989).

(D) Process Perturbation and Finite Difference Approximation (PPFA) technique.

In the above, it is possible to combine different characters together to form various algorithms. For instance, AaBCD means a centralised augmented version of the standard MTS technique combined with the PR method, and in this combination the required derivative measurements are obtained by employing a process perturbation technique. By exhausting all the possible combinations, it is possible to form any specified centralised optimising control algorithm in the algorithm bank.

6.4.2. Hierarchical Optimising Control Algorithms

(A) ISOPE Type algorithms with three options :

a. Two model (TM) approach (Lin, Kamphampati and Roberts, 1989);

b. Price Coordination (PC) strategy with two options :

α. Single iterative PC (Amini, Lin and Roberts, 1990);

β. Double iterative PC (Lin, Hendawy and Roberts, 1988b; Lin, Kambhampati and Roberts, 1989);

c. Modifier Coordination (MC) strategy with two options :

α. Single iterative MC (Lin, Roberts, Wang and Wan, 1990; Lin, Amini and Roberts, 1990);

β. Double iterative MC (Lin, Roberts, Wang and Wan, 1990);

(B) Penalty Relaxation (PR) technique (Lin, Kambhampati and Roberts, 1989).

(C) Variable Augmentation (VA) technique (Lin, Li, Wan and Roberts, 1989).

(D) Process Perturbation and Finite difference Approximation (PPFA) technique (Roberts, 1979; Brdyś and Roberts, 1987).

Again it is possible to combine different characters together to form the required algorithm. For instance, AcαBCD means a hierarchical augmented version of the single loop MC strategy combined with the PR method, and in this combination the required process derivative information is obtained by using the process perturbation technique.

6.5 SOFTWARE CONFIGURATION

As mentioned in Section 6.1, the configuration of the package is of a modular design. In accordance with the idea of decomposing optimising control algorithms into basic elements adopted in fulfilling the task of algorithm specification, the whole package is broken down into different function units. These units are specified according to the basic functions of the algorithms included in the algorithm bank, to the task of user interface, and to the need for the package management. All the function units are linked by a management unit where appropriate connections are made to form the user required optimising control algorithm. The management unit also looks after the user interface. The actual configuration of the package as well as its user interface is shown in Figure 6.1. The Specification of these function units are discussed in this section.

Package Management Unit

The major functions of this unit are :

a. Providing appropriate unit links according to the menu selected by the user;

b. Setting proper operating parameters for the chosen algorithm;

c. Interfacing with the user including information exchange and data conversion;

d. Creating a set of control commands for the purpose of program operation control and supervision.

Process Derivative Acquisition Unit

In this unit, the process perturbation technique is used to calculate the real process derivatives required for the purpose of set-point optimisation. The inputs to the unit are the simulated process input and output data-logs and the current controller set-points, and the outputs of the unit are the real process outputs, the derivatives of the process and the gradient of the real performance index.

Performance Index Unit

The function of this unit is to calculate the real performance index. The inputs of the unit are the controller set-points, the formulation of performance index, the corresponding value of the constraints and the penalty coefficient (for penalty relaxation technique). The outputs of the unit for the given controller set-points are the real performance index and the augmented Lagrangian for problems with output dependent constraints.

Constraint Unit

This unit consists of two sub-units. One is dedicated to the calculation of the values and the derivatives of the constraints for the given controller set-points and the other is designed to perform a linearisation on the constraints at the given controller set-points. It is worth mentioning that the unit handles the process constraints in such a manner that the whole contraint set is divided into three parts :

a. Input and output independent constraints;
b. Input dependent only constraints; and
c. Output dependent constraints.

Parameter Estimation and Modifier Calculation Unit

This unit consists of two sub-units. One is designed for the purpose of parameter estimation in the ISOPE method while the other is dedicated to the calculation of the modifier. The inputs are the steady-state output of the process and the structure of the steady-state model specified by the user. The outputs are the values of parameters of the steady-state model and the modifier.

Centralised Model-Based Optimisation Unit

The major function of this unit is to form and solve the model-based optimisation problem for the centralised ISOPE algorithms. Options are provided to form different model-based optimisation problems for the different combinations of the methods described in Section 6.3. The inputs of the unit are the modifier, the parameters of the steady state model, the linearised constraints, the controller set-points and the penalty coefficient. The output is the model-based optimum solution for the given controller set-point.

Hierarchical Model-Based Optimisation Unit

This unit consists of two sub-units (one for the single iterative algorithms and the other for the double iterative algorithms) with options to form different hierarchical model-based optimisation problems for the different combinations of the methods described in Section 6.4. This unit is used together with the system decomposition unit because the model-based optimisation problem to be solved is decomposed by the system decomposition unit into a number of sub-problems. The inputs of the unit are the modifier, the parameters of the steady state model, the linearised constraints, the controller set-points and the penalty coefficient. The output of the unit is the model-based optimum solution for the given controller set-point.

System decomposition unit

The major functions of this unit are :

a. Decomposition of the model-based optimisation problem according to the user requirements; and

b. Separation of the constraints according to the specified subsystem configuration.

These functions will be performed according to the user's specifications which are passed on to the unit through the management unit.

Coordination Unit

This unit consists of two sub-units (one for the implementation of the price coordination mechanism and the other is dedicated to the modifier coordination method). This unit is used in accordance with the system decomposition unit and the option of using different coordination methods is realised by using an indicator variable which takes the corresponding value to the instruction from the management unit. The inputs of this unit are the solutions of the model-based optimisation problem, the controller set-points, real process outputs and their derivatives at the given set-points, while the outputs are the price vectors or the coordination modifiers.

Plant Simulation Unit

In this unit, a number of numerical examples are selected for the purpose of algorithm test and case study for optimum process design. The plant simulation is conducted in a purely steady state manner and the interface with the other function units in the package is taken care of by the management unit.

Output Control Unit

The major task of this unit is to facilitate the package with different program output modes so that the operation of the algorithm can be properly monitored and the results may be interpreted conveniently. The switch between different output modes is realised through the user interface facility provided by the management routine.

The interface between the user and the package is realised through a menu driven facility which is implemented in the management unit of the package. By a proper design of the menus, it is possible to establish the following user interface functions :

* selection of optimisation modes, i.e. centralised or hierarchical process optimisation;

* implementation of any optimising control algorithm specified in the algorithm bank;

* structure of different process decomposition schemes;

* user intervention of the algorithm convergence;

* user intervention of solution precision;

* output and display modes selection.

The configuration of the package is illustrated in Figure 6.1. It is worth mentioning that the recommended configuration of the package as well as the above mentioned specifications of the function units are only standard and alterations may be found necessary in order to further improve the function of the package.

6.6 SUMMARY

Important issues concerning the software development of a CAD package for on-line optimising control of industrial processes are discussed in this chapter. Based on the requirements from the industry and the nature of process optimisation, a number of criteria for algorithm specification are established. In Section 6.3, the idea of forming an algorithm bank, which can produce a family of optimising control algorithms to cover most practical situations, is introduced and in Section 6.4, such an algorithm bank is established based on existing optimising control algorithms. In Section 6.5, the software configuration of an implementation scheme for the aforementioned CAD package is proposed and thoroughly discussed. It is understood that such a scheme may be used in a relevant project (Lin and Roberts, 1990).

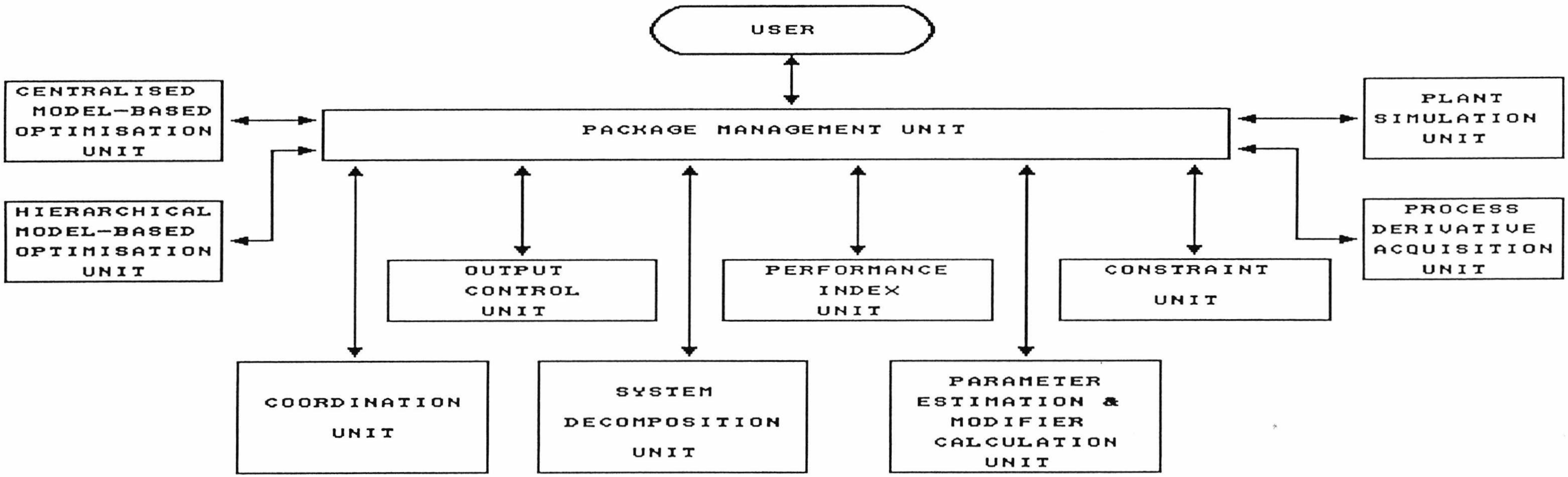

Figure 6.1 Configuration of the CAD package

CHAPTER 7

CONCLUSIONS

7.1 CONCLUSIONS

The research described has been concerned with developing, analysing and testing efficient hierarchical algorithms for on-line optimising control of large scale industrial processes. Emphasis has been given to the improvement of existing Integrated System Optimisation and Parameter Estimation (ISOPE) algorithms.

The global feedback structure has been explored to improve the efficiency of single loop ISOPE algorithms giving rise to an improved version of the single iterative ISOPE method. A new coordination method – Modifier Coordination (MC) method is proposed to increase the efficiency of existing hierarchical optimising control algorithms. Compared with the price coordination mechanism, this method proves to be superior in terms of algorithm efficiency and in terms of algorithm sensitivity. This method has been integrated into the ISOPE framework resulting in a number of hierarchical optimising control algorithms with different iterative structures. The penalty relaxation technique has been brought in to further extend these algorithms to cover output dependent cases. In addition, the Variable Augmentation (VA) technique has also been employed to reduce the algorithm sensitivity to iterative gain selection.

Theoretical analysis has been carried out to investigate the optimality and convergence of these newly developed hierarchical ISOPE algorithms and sufficient conditions for optimality and convergence of these algorithms are provided. It is concluded, based on the theoretical analysis, that the Modifier Coordination (MC) method has a better convergence feature than the price coordination mechanism in the sense that its applicability conditions are easier to satisfy, that by employing the Variable Augmentation (VA) technique it is possible to relax the applicability conditions of the algorithms, and that the integration of the MC method with the penalty relaxation technique retains the convergence features of the standard Modified Two-Step (MTS) method.

Substantial computer simulations have been carried out to test these algorithms. Comparisons of efficiency, sensitivity and reliability have been made amongst the presented algorithms to support an assessment for the algorithm specification in the practice of on-line optimising control of industrial processes. It is concluded that :

i. The MC method is much more efficient and reliable than the price coordination method in many cases;

ii. The variable augmentation technique, which can be easily integrated into the ISOPE structure, helps to increase the algorithm efficiency and reduces the algorithm sensitivity to the iterative gain selection;

iii. The penalty relaxation technique is found to be a highly efficient technique for handling the output dependent constraints.

iv. In general, the double iterative ISOPE algorithm with modifier coordination requires fewer controller set-point changes than its single iterative version. It is, however, worth mentioning that the total number of iterations for the double loop algorithm is usually higher than those for the single loop algorithm. This algorithm is, therefore, recommended for the cases where the controlled process is of slow transient nature, while the single iterative ISOPE algorithms with either modifier coordination or price coordination in a global feedback structure are usually recommended for cases where the transient of the process is fast.

Important issues concerning software implementation of optimising control algorithms have been investigated. A number of criteria for algorithm specification, based on industrial requirements, have been established. The new design strategy for forming an algorithm bank which produces a family of optimising control algorithms has been proposed for an optimising control CAD package. The advantage of using this design strategy is that it is much easier to implement a specified optimising control algorithm by linking different function units in the algorithm bank than by writing a separate code for each individual algorithm. The software configuration of such a package is also presented. It is hoped that this work provides some guidelines for software development of an optimising control CAD package.

7.2 SUGGESTIONS FOR FURTHER RESEARCH

The subject of the research is concerned with ISOPE algorithms, and during the past decade the ISOPE algorithm has been improved in many aspects resulting in a great number of updated versions. There is, however, an important issue, which is the major barrier on industrial application of the approach, to be investigated. As is well known, in the ISOPE type algorithms process derivative measurements are required to produce the true optimum regardless of model—reality differences. Since the process derivative measurements are not directly available, a perturbation technique is often used to estimate them resulting in a significant increase in the number of controller set-point changes. Efforts have been made to avoid process perturbation by using the dynamic response of the process (Bamberger and Isermann, 1978; Lin, Han, Roberts and Wan, 1989). Further research is required to simplify the approach and integrate it with existing ISOPE algorithms.

Up to now, in the practice of on-line optimising control, the algorithm specification has been entirely based on steady state simulation studies where the convergence behaviour and sensitivity of the algorithm are examined. However, it is usually inadequate to appraise an optimising control algorithm merely by steady state simulations for, in reality the algorithm is expected to function in a combined dynamic and steady state environment (Lin, Han Roberts and Wan, 1989). It is, therefore, suggested for the future research that :

a. Combined dynamic and steady state simulation techniques should be employed to create a simulation environment which is closer to the reality than the pure steady state one.

b. In addition to convergence and sensitivity analysis, stability studies need to be carried out to examine the stability of the entire control system.

c. So far, most optimising control algorithms have been tested using artificial examples rather than real ones. It is worth while testing the existing optimising control algorithms against working examples, for example taken from chemical processes utilising distillation columns, reactors, compressors, etc.

In Chapters 3, 4 and 5 the modifier coordination method has been extended to cover output dependent cases. Nevertheless, the approach may be further improved so as to simplify its implementation and analysis. Further research is also needed to improve the inner loop efficiency of double iterative type algorithms.

REFERENCES

Amini, Z., and Roberts, P.D., (1988) : "Convergence Analysis and Investigation of the Centralised Integrated System Optimisation and Parameter Estimation Algorithm", Research Memorandum CEC/ZA–PDR/94, Control Engineering Centre, The City University, London.

Amini, Z., Lin, J. and Roberts, P.D. (1990) : "A Single Iterative Algorithm with Global Feedback for Integrated System Optimisation and Parameter Estimation of Large Scale Industrial Processes: Optimality, Convergence and Simulation", IEE proceedings Pt. D., (to appear).

Bertsekas, D. P. (1982) : "Constrained Optimisation and Lagrange Multipliers Methods", (London: Academic Press).

Brdyś, M. (1983) : "Hierarchical Optimising Control of Steady State Large Scale Systems under model-reality difference of mixed type – a mutually interacting approach", Proceedings of 3rd IFAC Symposium On Large Scale System, Theory and Applications, Warsaw, pp. 49–57.

Brdyś, M. and Ulaniki, B. (1978) : "On the Completely Decentralised Control with Local Feedback in Large Scale Systems", Arch. Automat. Telemech., vol. 23, No. 1–2, 21–36.

Brdyś, M. Chen, S. and Roberts, P. D. (1986) : "An extension to the Modified Two-Step Algorithm for Steady-State Optimisation and Parameter Estimation", Int. J. systems Sci., vol. 17, No. 8, pp. 1229–1243.

Brdyś, M., and Roberts, P. D. (1985) : "Hierarchical single and double iterative techniques for optimising control of large scale industrial processes; derivation, applicability and convergence", IEE Int. Conf. "Control 85", Publications No. 9 –11, University of Cambridge, U.K., pp. 89–94.

Brdyś, M. and Roberts, P. D. (1986) : "Optimal structures for steady state adaptive optimising control of large scale industrial processes", Int. J. System Sci., vol. 17, No. 10, pp. 1449–1474.

Brdyś, M. and Roberts, P. D. (1987) : "Convergence and optimality of modified two-step algorithm for integrated system optimisation and parameter estimation", Int. J. Systems Sci., vol. 18, No. 7, pp. 1305–1322.

Brdyś, 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.

Brdyś, M., Abdullah, N., and Roberts, P. D. (1987) : "Augmented model-based double iterative loop techniques for Integrated System Optimisation and Parameter Estimation of large scale industrial processes", Proceedings of 10th IFAC World Congress on Automatic Control, Munich, F.R. Germany, vol. 7, pp. 97–102.

Chen, S. (1986) : "Integrated System Optimisation and Parameter Estimation Methods for On-line Control of Industrial Processes", Ph.D thesis, City University, U.K.

Durbeck, R. C. (1965) : "Principles for Simplification of Optimising Control Models", Ph.D. Thesis, Case Western Reserve University, Cleveland, Ohio, U.S.A.

Ellis, J. and Roberts, P.D. (1982) : "Measurement and Modelling Trade-offs for Integrated System Optimisation and Parameter Estimation", Large Scale Systems, vol. 13, No. 3, pp. 191–204.

Ellis, J., Michalska, H. and Roberts, P.D. (1984) : "Coordinated Decentralised Integrated Optimisation and Parameter Estimation of Large Scale Processes", Proceedings of 9th IFAC World Congress on Automatic Control, Budapest, Hungary, vol. 10, pp. 261–266 (Pergamon Press).

Ellis, J., Roberts , P.D. and Michalska, H (1986) : "Adaptive models in Decentralised Optimisation Control", Proceedings of 4th IFAC/IFORS Symposium : Large Scale Systems : Theory and Applications, Zurich, Switzerland, 26–29 Aug, pp. 334–339.

Findeisen, W., Bailey, F. N., Brdyś, M., Malinowski, K., Tatjeweski, P. and Wozniak, A. (1980) : "Control and Coordination in Hierarchical Systems", (London : Wiley and Sons).

Foord, A. G. (1974) : "On-line optimisation of a Petrochemical Complex", Ph.D. Thesis, University of Cambridge, England, Publicatiuon No. 252, pp. 281–285.

Haimes, Y. Y. and Wismer, D. D. (1972) : "A computational approach to the combined problem of optimisation and parameter identification", Automatica, vol. 8, pp. 337–347.

Hendawy, Z. (1989) : " Mathematical Algorithms for Optimisation of Large Scale System", Ph.D. Thesis, City University, U.K.

Kambhampati, C. (1988) : "Algorithms for Optimising Control", Ph.D. Thesis, City University, U.K.

Lin, J., Amini, Z. and Roberts, P.D. (1990) : "A New Coordination Strategy for On-line Hierarchical Optimising Control of Large Scale Industrial Processes", Submitted to Int. J. Control (to appear).

Lin, J., Han, C., Roberts, P.D. and Wan, B. (1989) : "New approach to stochastic optimising control of steady state systems using dynamic information", Int. J. Control, vol. 50, No. 6, pp. 2205–2235.

Lin, J., Li, L. Z., Wan, B. W. and Roberts, P. D. (1989) : "Augmentation of several newly developed algorithms for integrated system optimisation and parameter estimation of large-scale industrial processes: Convergence analysis and comparative study", IMA Journal of Mathematical Control & Information, vol. 6, pp. 333–345.

Lin, J., Wang, M. and Roberts, P.D. (1989) : "A composite dynamic and steady state approach to stochastic optimising control of steady state systems with output dependent constraints", Submitted to IEE Proceedings Pt. D.

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.

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.

Lin, J., Roberts, P.D. , Wang, M. and Wan, B.W. (1990) : "Hierarchical integrated identification and optimisation approach for on–line stochastic optimising control of large scale steady state industrial processes", Int. J. Control, (to appear).

Lin, J., Chen, S. and Roberts, P.D., (1988) : "Modified algorithm for steady state integrated system optimisation and parameter estimation", IEE Proceedings, vol. 135, Pt.D., No. 2, pp. 119–126

Lin, J., Kambhampati, C. and Roberts, P.D., (1989) : "A new two-model approach for optimising control of large scale steady state systems", IMA Journal of Mathematical Control and Information, vol. 6, pp. 39–61.

Lin, J. and Roberts, P.D. (1990) : "Manual of a Novel Optimisation Routine for Solving Constrained Non-linear Optimisation Problems "**Update of document EPCR0029"**, ESPRIT II Contract 2090 "Early Process design Integrated with Control (EPIC)"

Luenberger, D. G. (1984) : "Linear and Non-Linear programming", (London : Addison-Welsley).

Roberts, P.D. (1979) : "An algorithm for steady state system optimisation and parameter estimation", Int. J. Syst. Sci., vol. 10, pp. 719–734.

Roberts, P.D. (1987) : "Combined system optimisation and parameter estimation", in Systems & Control Encyclopedia ", ed. M. G. Singh, pp. 634–638. (Pergamon Press).

Shao, F. Q. and Roberts, P. D. (1983) : "A Price Correction Mechanism with Global Feedback for Hierarchiacl Control of Steady State Systems", Large Scale Systems, vol. 4, No. 1, pp. 67–80.

Singh, M. G. and Titli, A. (1978) : "Systems: Decomposition, Optimisation and Control", (London : Pergamon Press).

Tatjeweski, P. (1985) : "On-line Hierarchical Control of Steady-State Systems Using the Augmented Interaction Balance Method with Feedback", Large Scale Systems, vol. 8, No. 1, pp. 1–18.

Tatjeweski, P., Abdullah, N. and Roberts, P. D. (1990) : "Comparative study and development of integrated optimisation and parameter estimation algorithms for hierarchical steady-state control", Int. J. Control, vol. 51, No. 2, pp. 421–443.

Youle, P. V., and Duncanson, L. A. (1970) : "On-line control of olefine plant", Chemical and process Eng., vol. 51, pp. 49–52.

Zangwill, W. I. (1969) : "Nonlinear programming : A Unified Approach", (Englwood Cliffs, NJ : Prentice Hall).