DUBLIN CITY UNIVERSITY

School of Electronic Engineering

Master of Engineering
Thesis

INTELLIGENT CONTROL OF INDUSTRIAL PROCESSES

Author: Mark McDonnell B.Eng.

Supervisor: Dr. Charles McCorkell

September 1989

THIS THESIS IS BASED ON THE AUTHORS OWN RESEARCH RESULTS

ACKNOWLEDGEMENTS

I wish to sincerely thank my supervisor, Dr. Charles McCorkell, for his much valued guidance and support throughout this project. Many thanks also to the staff of Dublin City University and to ADERSA (Paris) for their cooperation on predictive control and the LBS development. The time and effort spent by Mary Genockey in typing this thesis is much appreciated. Lastly, but not least, my sincerest gratitude to my wife for her constant support and understanding throughout and her help in compiling the thesis.

CONTENTS

ABSTRACT

A detailed survey of the field of intelligent control is presented. Current practices are reviewed and the need for a unifying framework to identify and strengthen the underlying core principles is postulated. Intelligent control is redefined to make explicit use of human systems in control as a reference model. Psychological theories of intelligent behaviour reveal certain basic attributes. From these a set of necessary and sufficient conditions for intelligent control are derived. Learning ability is identified as a crucial element. Necessary attributes for learning are prediction capabilities, internal world model, estimation of the model parameters, and active probing to reduce uncertainties. This framework is used to define a Learning Based Predictive Control (LBPC) strategy. LBPC is derived from Predictive Functional Control techniques with an adaptive layer implemented by recursive least squares. Improved performance above conventional adaptive control is demonstrated. Distributed parameter systems are identified as a suitable application area requiring an intelligent control approach. Such systems are invariably complex, ill-defined, and nonlinear. Plasticating extrusion processes are considered in particular. LBPC is applied to control of the primary loop to regulate melt temperature and pressure at the die. A novel control technique is proposed for dynamic profile control of extruder barrel wall temperature. This is a two-level hierarchical scheme combining the benefits of LBPC control blocks at the lowest level with decision logic operating at the higher level as a supervisor. This Logic Based Strategy allows multivariable control of non-square systems with more outputs than inputs. The application of LBS to an extruder is demonstrated.

LIST OF CONTRIBUTIONS

The following list details the contributions contained in this thesis:

(1) A detailed review of the current state of the field of intelligent control.

(2) The need for a unifying framework to identify and strengthen underlying core principles.

(3) A new definition of intelligent control which explicitly utilises human systems in control as a reference model.

(4) A set of necessary and sufficient conditions to allow design or classification of intelligent control systems.

(5) Reformulation of Predictive Functional Control in terms of polynomial input-output models.

(6) Development of an adaptive PFC algorithm.

(7) Identification of distributed parameter systems as a class of application problems suitable for an intelligent control approach.

(8) Application of polynomial PFC to a plasticating extruder (a distributed parameter system).

(9) Development of a hierarchical logic based strategy for multivariable control of non-square systems.

(10) Application of LBS to extruder barrel wall temperature control, including actuator placement and controller structure considerations.

INTRODUCTION

The field of control engineering has a rich history with a wealth of practical and intellectual achievements. There has been rapid developments since the work of Bode[1] and Nyquist[2] on *feedback* theory. Feedback allows good performance in the presence of uncertainty. This is important as most dynamical systems (eg. industrial processes, machines, etc.) operate in changing environments which cannot be modelled easily. In the last twenty five years progress has accelerated, due mostly to the development of the digital computer. Recognition of uncertainty due to random environmental inputs has led to the development of *stochastic control*[3]. This theory uses explicit models of the disturbances as random processes and tries to minimise the probability that a system output will move outside a "safety" zone or operational range (eg. minimum variance control).

Complete models of realistic systems are seldom available to the control engineer. Accepting this, designs utilising incomplete models have been proposed that reduce the uncertainty online. Adaptive control[4,5] attempts this reduction by actively estimating the model (or controller) parameters online and using these estimates to produce improved control. It can thus cope with a much larger range of uncertainty than the nonadaptive systems mentioned above. An adaptive system may be considered to learn since it reduces the uncertainty in a stochastic environment as it evolves. At present however, it is limited in that only very structured uncertainties may be learned, i.e. the unknown parameters of a fixed order plant model. There are also problems with the robustness of adaptive controllers[4,5].

As the type and size of systems considered by modern control engineers grows in complexity so too does the task of meeting rigorous performance requirements. As efficiency and cost factors increase in importance, performance conditions become tighter and more difficult to achieve. These challenges become more difficult because of the uncertainty of the system model and its environment. New control systems are required that can tolerate greater degrees of uncertainty than current adaptive systems. These must also have better learning abilities. Theories that can handle incompletely known systems, or systems described by nontraditional models (eg. symbolic models) are needed. Process knowledge in addition to dynamic or static models such as operating procedures and specifications would improve controller operation. As the complexity of the controller grows so too does the amount of heuristics needed to implement it[6], and the lack of faith of the operator in the controller. Hence, efficient methods to encapsulate the necessary heuristics and to interact with the operator are desired.

Increasingly, control designs are requested for complex, large-scale, spatially distributed systems. Examples of such systems include robotic workcells, flexible manufacturing systems, large space structures, communications systems, power systems, etc. In this new scenario of plant-wide control new issues and problems arise other than those encountered in more conventional or traditional systems. Issues such as location of actuators and sensors, numbers of actuators, and control topology become active control system design issues and thus assume greater importance. Control systems will have more autonomy and interact with operators at higher levels. Thus more abstract (eg. linguistic) control objectives may be received and these must be decomposed into primitive elements which may be executed as part of an overall plan. This process requires symbolic reasoning and planning abilities which can deal with uncertainties and failure of plans in a robust manner. It may be required to make decisions as to what must be controlled and to select different strategies in the face of altered configurations. Such a system must deal with environmental input from many sensors and an overwhelming amount of data. Methods of *fusing* this information to obtain a coherent view of the 'environment' is a major problem but is needed to enable high level planning and to provide the operator with a linguistic summary of the state of the plant. In these complex systems of many sensors and actuators attention must also be given to the problem of *failure*. The control system must be tolerant to faults or failures in the system components such that stability is maintained and preformance is gracefully degraded.

This ambitious agenda of controller capabilities defines the discipline of *intelligent control*. It has many definitions in the literature [7-10] but each deals with only one or more of the above mentioned topics. It is a relatively new and immature field of research and will undoubtedly undergo many changes and modifications in the future. Future research and industrial needs will further refine its definition. This may lead to the encapsulation of the previous ideas into a coherent framework or some of these ideas may be discarded while others strenghtened in importance.

A detailed survey of intelligent control is presented with the many diverse and varied research strands highlighted in chapter 1. The need for a unifying framework based on underlying characteristics and not on application specific solutions is postulated in chapter 2. A particular framework is proposed based on explicit use of intelligent human behaviour as a reference model. Learning is identified as a sufficient condition for such behaviour with prediction, internal models, and probing as important attributes or necessary conditions. A Learning Based Predictive Control strategy is formulated based on these principles in chapter 3. This is designed around Predictive Functional Control, a long range predictive control technique, with an adaptive layer implemented by Recursive Least Squares. The application of this control technique is examined for control of a distributed parameter system. Distributed systems contain many of the problems discussed above. In chapter 4 the application of LBPC to an extrusion process, a particular distributed parameter system, is demonstrated. A two level hierarchical approach to the control of multivariable systems using decision logic in conjunction with predictive control is detailed in chapter 5. This technique mixes control theory with decision logic, a computer science tool, in a fashion similar to current approaches to intelligent control. A particular example of this strategy is examined for extruder barrel wall temperature control in chapter 6. This describes a Distributed Actuator Control (DAC) problem which considers issues such as positioning of actuators and sensors, classification of controlled outputs, and choice of feedback topology. This particular application derives considerable benefits from the use of Learning Based Predictive Control and the logic based multivariable control strategy.

CHAPTER 1

REVIEW OF INTELLIGENT CONTROL

1.1 INTRODUCTION

A detailed review of the field of intelligent control is presented. Some of the structures and architectures that have been proposed for intelligent control are first described. Particular tools have been intimately associated with this research field; the use of expert systems (a successful AI tool that allows human expertise or knowledge to be encapsulated in a computer program) and fuzzy logic (a method for emulating human reasoning mechanisms) are discussed. These are useful when mathematical models of a plant are difficult to determine. Artificial neural networks are finding increased use and are also within the realm of intelligent control. The cognitive aspects of intelligent controllers are discussed as well as learning based controllers. Other aspects presented here include sensor data fusion and reasoning with uncertainty.

Technology is a very important element of intelligent control. Control theory developed rapidly due to the advancements made in computer technology during the sixties, seventies, and eighties. Intelligent control is dependant on future developments in symbolic processing machines and software, numeric-symbolic interfacing, and real-time AI software and hardware. Aspects of technology are referred to where approriate.

The progress of the field is also related to the integration of ideas from many disciplines, not just mathematics and physics but also from artificial intelligence, operations research, and behavioural psychology. Norbert Weiner's *cybernetics* may have a contribution to make. The design of 'intelligent' control systems may not be achieved without resort to the study of the human brain and the biological mechanisms which embody so many of the characteristics of intelligent control.

1.2. FORMAL STRUCTURES AND ARCHITECTURES

Formal structures for intelligent control have been proposed by many researchers. Due to the nature of the problem as discussed in section 1.1, such structures are typically hierarchical[11] in organisation. Problems are decomposed at high levels into small units which can be further subdivided to produce control or actuator inputs to the lowest level. Similarly information is abstracted at the higher levels. The upper levels are mainly concerned with operator interaction, planning, coordinating, and reporting. AI has most influence in these levels due to the nature of the tasks addressed.

1.2.1 Hierarchical Architectures

A Hierarchically Intelligent Control System has been proposed[12,18,19] which has a three level structure. This system follows closely the definition of Intelligent Control[13] as a control which *"would replace the human mind in making decisions, planning control strategies and learning new functions whenever the environment does not allow and does not justify the presence of a human operator".* It is suggested that this can be achieved through the interaction of three major disciplines (see fig. 1.1): Artificial Intelligence, Operations Research, and Control Theory. The overall system is based on a principle called *Increasing Precision with Decreasing Intelligence* as a unified theoretical approach to the combination of cognitive and control system methodologies[14]. A comprehensive mathematical formulation has been presented[12-18]. At the higher levels information is more abstract and less precise although working with such knowledge is considered intelligent. Similarly less abstract but more precise knowledge requires less intelligence.

The upper (Organisation) level performs such operations as planning, high level decision making, learning, and data storage and retrieval from "*long-term memory*" . It performs high level information processing using knowledge techniques derived from AI. Because it is a *knowlege-based system*[19], a formalism based on flow of information (or knowledge) may be used to define its operation. To this end, entropies and entropy rates are employed. The second (Coordination) level consists of groups of automata, or linguistic decision schemata[16], see section 1.6.3. This level performs further decision making and planning, and co-ordinates the operation of the lower level "*execution*" devices (i.e. actuators and sensors). Subjective probabilities for each possible action are assigned from which respective entropies may be obtained. Low-level learning is performed through updating of the subjective probabilities of the automata using feedback from the lowest level. The lowest (Execution) level or hardware layer, also incorporates the use of entropies. The cost of the control problem is expressed as an entropy which measures the uncertainty of selecting an appropriate control to execute the task. Entropy is minimised by selecting an optimal control. Equivalent measures between information theoretic and optimal control problems have been established[17]. Learning as such is not considered within the context of the lowest level which is regrettable as adaptive control techniques could allow parameter estimation, and hence learning, at this level.

A very similar intelligent control architecture was proposed[20] which is also based upon the principle of increasing precision with decreasing intelligence and contains three levels. A functional description of each layer was presented with the upper two layers implemented with AI technology and the lowest execution layer incorporating a combination of hardware and conventional software. Again the application used was an autonomous space vehicle. Learning within the upper two levels is considered to be of prime importance and adaptive

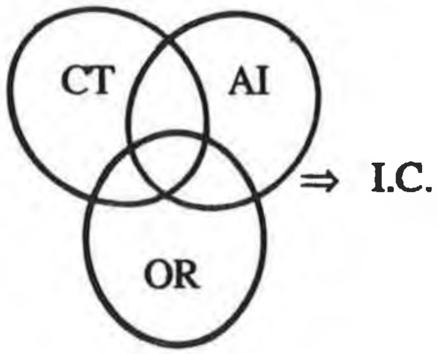

FIG. 1.1: Definition of Intelligent Control

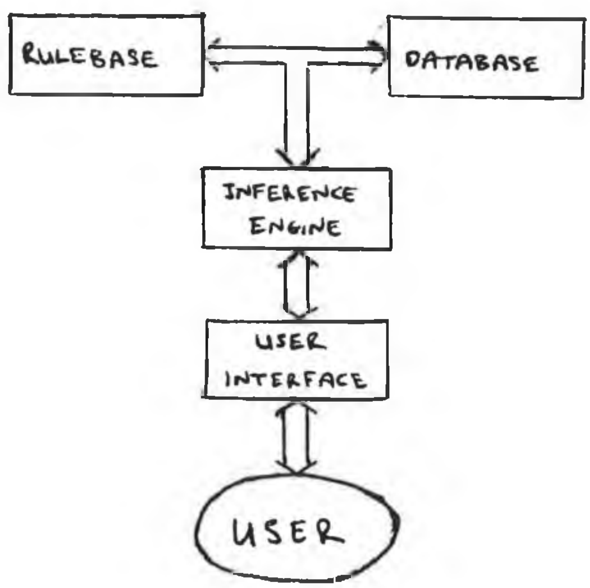

FIG. 1.2: Rulebased Expert System

controllers are used at the lowest level.

Several formal structures for incorporating knowledge based systems in an intelligent control framework have been presented[6,21-24]. One approach[21,25] is to utilise an evidential technique to the problem of inexact reasoning in intelligent control systems where the available knowledge and evidence is incomplete or even inconsistent. This is a *rulebased* approach to control using *evidential reasoning* (see section 1.7.3). The knowledge base consists of a set of 'prototypical' control rules as specified by human domain experts. Each rule represents a control decision, i.e. a mapping from a control situation to a control action. In instances where there is no direct rule that is applicable, a fuzzy classification algorithm[21,25] determines a new control decision by *"analogy"* with the most similar rules (prototypical decisions) in the knowledge base. The need to look at human expertise for guidance in the design of intelligent control systems has been expressed[22] and a method for incorporating analytical and symbolic models to represent real systems was presented. Another formal hierarchical structure was proposed[6] as a means to incorporate expert systems in a level above the real time controller. This will be looked at in detail in section 1.3. The knowledge-based system incorporates several control, parameter estimation, and supervision algorithms as well as operator and procedural information about the plant. The appropriate algorithms are chosen based on the current circumstances of the plant.

A final although important structure to look at is that of cognitive controllers[26]. Obviously to endow controllers with more intelligence or autonomy to work in hazardous conditions or complex systems it is necessary to provide them with cognitive capabilites. Such structures entertain principles similar to those utilised in human cognition. This presumes the existance of several features: eg. multisensor perception subsystems, knowledge organisation abilities, extensive world representation, decision making and planning facilities, and also learning capabilities. Such structures may be utilised through the use of neural networks (see section 1.5) although as will be shown later, the incorporation of human like abilities may be performed without the need to resort to neural networks, see section 1.7.1. One approach is based on the theories of brain structure and function[91,92]. The basic architectureis hierarchical consisting of multiple levels with an ascending *"sensory processing hierarchy"* coupled to a descending *"goal decomposition hierarchy"* via *world models* at each level. This is called a *sensory-interactive control hierarchy* and is composedof many CMAC modules. Each CMAC (Cerebellar Modular Arithmetic Computer) is basically a look-up table for reproducing functions with multiple input and output variables over particular regions of the state space. Learning is accomplished by altering the values of the table through a training procedure[91,92]. At each level of the sensory processing hierarchy, feedback information from the lower level is abstracted by a CMAC module to fit into the world model. Also, at each level a CMAC decomposes the goal from a higher level into a set of subtasks for a lower level. Prediction is also incorporated by a CMAC which considers the current goals and context (sensory information) to provide predictions of expected feedback. This sensory-interactive structure has been successfully applied to the control of a robot manipulator[92].

1.2.2 Discussion

Several noticeable problems are evident from examining the many and varied structures and architectures proposed for Intelligent Control. None really address the lowest level actuator controller, i.e. the interface to the physical system. All are concerned with the addition of extra hierarchies to handle the decomposition of goals into subtasks, planning, dealing with unexpected occurances, abstracting information from intelligent sensors, and providing high level interfaces to operators. Most structures have assumed standard adaptive or optimal controllers in the lowest loop. Although it is undoubtedly worthwhile to consider the upper hierarchical levels it would also seem sensible to examine what type of controller should be used for optimum performance of the complete intelligent control system. Should optimal controllers be used as in [15] which have no learning capabilities? Are adaptive controllers better suited with their primitive rote learning functions? If the principle of cognitive controllers are accepted should not the actuator controller also exhibit human like abilities? Is there a need to provide several control laws and allow switching between these to meet varying objectives?

The lack of diverse applications is another noticable inadequacy. Invariably implementations or simulations of the above architectures are with regard to robotic or unmanned vehicles. Such systems obviously need a high degree of intelligence and humanlike abilities in order to operate autonomously. However it is important to diversify the type of applications to which intelligent control may be applied. In particular non-robotic systems should be looked at. Complex and highly uncertain systems are perfect examples requiring advanced learning abilities, as mentioned in the introduction. Also systems that vary considerably both in their parameters, structure and control objectives (eg. bioreactors or distributed parameter systems) provide useful applications.

1.3 EXPERT SYSTEMS

The field of expert systems is a rapidly expanding area of Artificial Intelligence. It is one of the truly practical tools to emerge from AI[19,27,28] and is currently finding application in a multitude of diverse scenarios, from legal aids to data base management, to oil exploration. The use of expert systems in control is new gaining much recognition and considered by some to be the embodiment of intelligent control.

Expert systems seek to model the knowledge and procedures used by a human expert in solving problems within a well defined domain. To this end the model reflects aspects of a problem which are not naturally amenable to numerical representation or which can be more efficiently represented by heuristics. The implementation of practical control laws require a considerable amount of heuristic logic[6] which is often implicitly incorporated with the actual control law. Expert system methodologies provide a systematic approach for separating the heuristic logic and actual control law resulting in considerable improvements[6].

1.3.1 Expert System Architecture

The typical architecture of such a system has four principle components (see fig. 1.2):

(1) System data base to store declarative knowledge - a repository of facts about the present and past states of the system, eg. data measurements, alarm conditions, etc.

(2) Rule base of procedural knowledge implemented as production rules.

(3) Inference engine or control structure for manipulating the date base and rule base, ie to decide which production rule to apply next given the present context of the system (obtained from the date base).

(4) User interface which allows the operator to question the expert system on why certain decisions were made, actions performed, allow qualitative definitions of goals by the user, and provide a qualitative description of the present system state.

It is this explicit separation of the knowledge base from the problem solving or control strategy that distinguishes expert systems from conventional application programs.

The major problems encountered with the use of expert systems are:

(1) *Real-time capability*

The available symbolic processing capabilities do not allow the use of expert systems for fast motor control, although they are sometimes used for path planning in the higher levels of a robot controller. Most expert systems are finding application in the field of process control where response times are much longer.

(2) *Development tools and knowledge engineering*

The lack of available general purpose tools for building expert systems to work in a control environment. There is a major problem with determining the appropriate human operator production rules which is a difficult and time consuming task. Knowledge engineers however specialise in such tasks.

Another area which must be looked at is efficient communication methods between the real time numerical processing of the control algorithm and the slower symbolic processing of the expert system. Learning, i.e. automatic adaptation of the rulebase, is an area receiving much attention[29]. However this is mostly concentrated on developing a rule base automatically when presented with long streams of quantitative data. Online adaptation of the rule base and also restructuring of the data base would be useful functions for an intelligent control system.

1.3.2 Applications in Control

Expert systems find most use in complex systems with long response times, for example process control, and tend to perform supervisory functions on a hierarchical level above the control law implementation[30]. One of the first tasks to be assigned to expert systems in process control is fault diagnosis and repair. Multiple data sources and online information are used to deduce problems with the plant and to estimate the probable cause of the problem and indicate a possible repair/replacement strategy. Process supervision tasks require: process management, alarm handling, and optimizing control. The expert system must monitor the overall state of a plant, scan alarm sensors to watch for possible problems or failures, and also ensure optimum performance through selective tuning of the control laws, switching between different controls, or variation of setpoints. The plant status may change drastically within a few minutes and the large number of variables that are monitored and alarms that are provided necessitates the use of an expert system with reasoning schemes similar to those of a human operator (expert) who when confronted with such situations responds and takes actions within the limited time available. The role of the expert system is in an overall plantwide control hierarchy above that of the normal distributed control system.

Many proposals have been made for the application of expert systems in process control and supervision. One example is the "*general intelligent supervisory control scheme*"[30]. This has a three level hierarchical structure. The first level corresponds to the classical 'controller-process' loop in which MV's are computed at each sampling instant. The second level is the "information generator" which continuously provides the third level with condensed useful information. In this level, information perception is only at an analytical or numerical level. The third level is the "expert supervisor" in which both quantitative and qualitative information is used. Intelligent functions like decision making are handled here via symbolic processing. This system has been applied to the control of a turbocharged diesel engine[31]. Other applications employing expert systems follow very similar lines[32-34]. Particular areas have been looked at in detail, eg operator interface[35] and interfacing of numerical and symbolic processing[36]. A rule based system has been used with an inductive learning algorithm to obtain its own rule base[37]. This can be done from examples or historical data of the process. The inductive learning algorithm applies information theory to develop decision trees from examples based on maximising the information content.

Examination of robotics literature reveals quite widespread use of expert systems. Mostly they are employed with advanced sensors, eg vision or tactile sensors, to interpret data and recognise objects. One particular use[38] employs an evidential reasoning approach to recognise 3D objects and determine the most suitable grasp configuration for a robot arm. The other most common use is for task decomposition, operator interfacing, and path planning[39-42]. Other regularly encountered applications of expert systems in the control field are ship positioning[43,44] and self tuning of PID and other control algorithms online[6,45]. Tuning of PID controllers is an obvious area for expert systems due to the heuristic rules (Ziegler-Nicholls, Cohen-Coon) used for this purpose.

1.3.3 Expert Control

A detailed framework for blending numerical algorithms with expert system technology has been presented under the name of *expert control*[6]. This recognises the fact that engineering of control laws require a good deal of heuristics. For example the implementation of a PID algorithm requires consideration of issues like operater interface, operational issues like switching smoothly between manual and automatic operation, transients due to parameter changes, windup of the integral term, maximum and minimum values, etc (see fig. 1.3).

Given that heuristics are used, expert control considers the improvements to be observed by more extensive use of heuristics and by explicitly making them available in the form of a rule base. It also explores the possibility of designing systems that combine a range of algorithms, i.e. control, identification and supervision. This entails orchestration of the different algorithms to achieve varying control objectives (see fig. 1.4). Selection of different control structures is made in current control systems to a limited extent by hard-wired logic. The objective of expert control is to encode knowledge representations and decision capabilities to allow intelligent decisions and recommendations automatically rather than to preprogram logic which treats each case explicitly. To this end it deals with qualitative or symbolic knowledge rather than quantitative data used in more conventional control systems. Learning is considered to be very important for the control system to behave intelligently. Learning is performed by successively increasing and refining the knowledge about the plant through online compilation or modification of the data base. It is proposed to store processing history in a manner suitable for automatic learning as

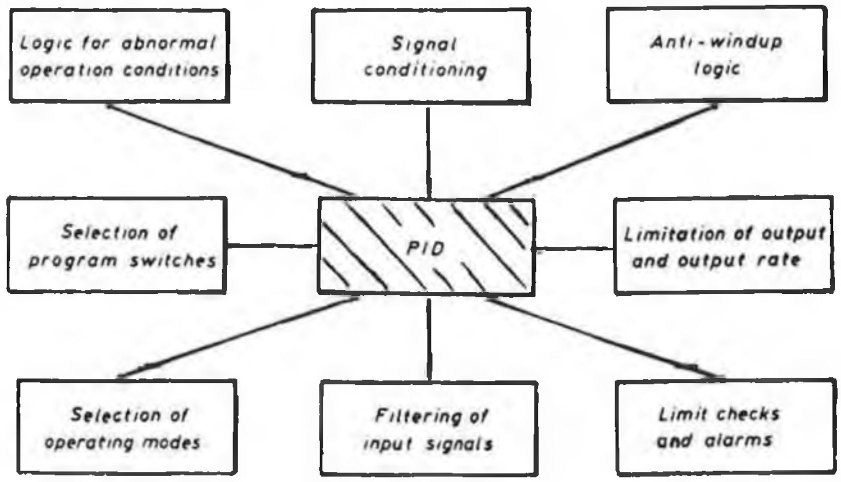

Fig. 1.3: A PID and Associated Heuristics

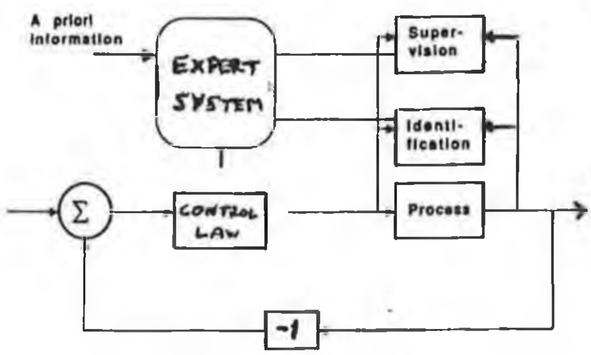

FIG. 1.4: Expert Controller

suggested by the AI field.

This framework has been used to design an expert controller[6,46] which uses a minimum variance algorithm for the main control law with PID control as backup. Heuristic logic is included to monitor the algorithms and provide functions like parameter estimation, ensuring persistency of excitation, controller switching, stability checking, etc. The most serious drawback encountered in this implementation was the lack of an expert system with true realtime capabilities. Much research is required in this field to enable advances in intelligent control.

1.3.4 Implementation Issues

Expert systems may be written in conventional procedural languages (eg. C, PASCAL, etc.) since the distinguishing feature of expert systems from conventional programs is the explicit separation of knowledge and control structures. However most systems are written in an AI language, like LISP or PROLOG, due to the nature of the symbolic computations. Also LISP provides a ready made interactive environment. To reduce the time needed to create an expert system, expert system shells are commonly employed. Shells are packages that have inbuilt control structures, eg. forward or backward changing, but are delivered with blank rule bases and data bases.

Speed is a major consideration. Several methods have been proposed to improve their realtime capabilities:

(a) use of a shared memory structure (generally called a blackboard architecture[47]) for coping with rapidly changing data.

(b) use of an appropriate interface between the expert system inference engine and a data-scanning component and what are called "focus of attention" rules[48].

To improve runtimes of expert systems for symbolic processing use is being made of recently available dedicated AI integrated circuits, eg. dedicated LISP engines. These can process LISP code to the order of 10/50 times faster than conventional processors. PICON[48] (*P*rocess *I*ntelligent *CON*trol) is a commerically available system than employs this idea.

1.3.5 Discussion

The ever increasing roll that expert systems have to play in control has been examined as well as appropriate technologies for their implementation and efficient use. It has been shown that their use can considerably improve the performance of control systems; particularly when used on large complex plants. For processes with multistage operating modes they provide an elegant means to switch between different algorithms to meet varying operating priciples and desired objectives.

But expert system technology is still only a tool. Why then is their use so effective? They allow intelligent behaviour of the control system through the incorporation of human expertise and experience with heuristic logic. This allows them to behave as would human experts. It is thus the underlying principles and not the tool itself which endows the control system with 'intelligent' abilities. The intelligence arises from the embodiment of human characteristics in the expert system.

Expert systems should thus be considered as a means to enhance control system performance as part of a hierarchically intelligent control system. To ensure efficient or optimal use of the expert system however, much investigation and research is needed. New development tools to design expert systems are required. Learning abilities and realtime capabilities must be investigated as well as methods for handling uncertainities. Expert system shells more suited to control operations would be very beneficial as well as an efficient interface between numerical and symbolic processing elements. Despite these present inadequacies, expert systems would provide an even greater impact on control system development in the future.

1.4 FUZZY LOGIC

Fuzzy logic[49] is a method of emulating human thought processes. It is based on the premise that the key elements in human thinking are not numbers but labels of fuzzy sets. That is, classes of objects in which the transition from membership to nonmembership is gradual rather than abrupt. It seems that the logic behind human reasoning is not two-valued or even multi-valued logic, but a logic with fuzzy truths, fuzzy connectives, and fuzzy rules of inference. There is a fundamental difference between the imprecision of fuzzy set theory (or possibility theory) and that of probability theory. Probability deals with randomness of future events whereas possibility theory deals with the imprecision of current or past events. Fuzzy logic provides a method for manipulating fuzzy sets and hence a scheme for reasoning about possible actions based upon current or past measurements. The fundamental notion of fuzzy set theory has been elaborated mathematically[49,50]. The theory has substantially matured and there has been considerable development of applications to a wide range of problems.

A clear outline of the use of fuzzy sets for the analysis and control of complex systems has been presented[50,51]. The use of fuzzy set theory for control of industrial processes provides an alternative to the more traditional well developed concepts of control theory[54-57]. This approach gives most benefit in situations where the precise mathematical models of the industrial process are unobtainable. Typical of such fuzzy processes are those involving biological or chemical reactions, or those of such complexity that mathematical modelling is not practical (eg. cement kilns, steel furnaces, and sewage treatment plants). Such processes are controlled by human operators who are capable of controlling under imprecision. It is this 'human' aspect of the control problem that fuzzy controllers emulate through the use of qualitative information gleaned from the operators.

1.4.1 Fuzzy Control Structures

Basic to fuzzy control algorithms is their interpretation in terms of linguistics, or linguistic control rules. The control rule is a statement of actions to be taken given certain conditions. Typically a fuzzy control rule says that 'if the temperature is high and the pressure is rising rapidly then reduce the fuel by a large amount'. Such statements are imprecise, or fuzzy, but contain a lot of heuristic knowledge gleaned from an experienced operator. The similarity with expert systems is obvious. Fuzzy controllers have the same internal structures as expert systems[52]. The knowledge base consists of linguistic rules similar to that above. The inference mechanism is a form of forward chaining where the result is

$$K_3[du(kt)] = F[K_1e(kt),K_2c(kt)]$$

where F denotes the fuzzy relationship defined by the rule base and K the scaling factor between the measurement space and the universe of discourse for the appropriate controller variable. Although there is a similarity between the two types of systems, fuzzy controllers tend to be used at a lower level than expert systems. They are used to directly control a process whereas expert systems tend to be employed for supervision of other control algorithms. A major problem with fuzzy controllers is one applicable also to expert systems, i.e. the elucidation of the heuristic rules for control. There is also a problem with possible contradictory rules and also ensuring the existance of an output for each input[52].

1.4.2 Self-Organisation

Improved versions of fuzzy controllers have been proposed[53]. One such controller has more 'intelligence' in the sense that it is capable of automatically modifying the rules applied online. That is, it has the ability to learn. Such a controller is termed a self organised fuzzy controller (SOC). The rules are modified according to a measure of the deviation of each output from a trajectory p(kt), where

$$p(kt) = \Theta[e(kt),c(kt)]$$

and Θ represents the performance decision table used. The input correction r(kt) is fed to the rule modifier which alters the linguistic rules such that future control actions lead to the appropriate output improvement.

The SOC developed[53] was a SISO strategy but this was extended for the multiple input multiple output case[54]. Theoretical results where also presented[55] where an algebraic model of the controller was developed. This was used to study the loop stability conditions following a purely linguistic approach. Other applications[56] looked at have considered chemical plants[57-59] and robotics[60].

1.4.3 Supervision and Managerial Applications

Fuzzy logic has also found application in similar areas to expert systems, i.e. at higher levels than the direct control loop. In industrial control scenarios it has been used to supervise the operation of standard low-level direct control loops. One particular use[30] has been to use fuzzy logic to tune the parameters of a standard PID controller. The idea is to slightly change the parameter values (initially provided by some existing technique such as Ziegler-Nichols) during the system transient so as to improve the characteristics of the step response. Empirical rules were used in a 14x14 fuzzy control matrix. The rules were obtained by observation of experienced human controllers.

Some applications have been in social, ecomonic, and managerial domains[58]. The optimisation of fuzzy goals in the context of fuzzy constraints is seen as a key form of management control. Production scheduling was looked at and a model derived that simulated the approximate reasoning abilities of managers facing production scheduling decisions. A domestic airline maintenance control system has been demonstrated, and models to simulate organizational behaviour are other examples of the use of fuzzy set theory in this field[58].

1.4.4 Discussion

The area of fuzzy set theory was looked at and it was seen that fuzzy logic provides a mechanism for emulating human performance. It allows us to use explicit rules derived from the observation of human behaviour. In this way, automatic controllers may be built for plants which do not permit precise mathematical modelling. However, there still remains the difficult problem of elucidating these rules of behaviour from the experienced operators.

All fuzzy controllers use quantized values of the error signal and its rate of change. Hence there is an obvious association with Proportional plus Integral (PI) controllers. A method for endowing fuzzy systems with more intelligence tries to provide them with learning abilities. This self-organising fuzzy controller allows a system to adapt and to change its rule base online in response to performance inadequacies. Such an 'intelligent controller' would be very beneficial, in particular for systems that are difficult to model or exhibit drastic changes in structure during their operation.

Fuzzy logic tends to be used mostly in the lower-level or direct control loop. But why restrict variables to quantization levels when accurate sensors are available which can provide reliable and fast measurements. Undoubtedly, direct fuzzy controllers can play a significant part in the control of systems that are poorly modelled. But other systems could benefit from the integration of modern (or classical) control techniques with fuzzy managers or supervisors operating at a higher hierarchical level. Again the comparsion with expert systems is unavoidable. Fuzzy logic however appears to be just another tool or method for encapsulating human behavioural rules. A technique for performing logical or symbolic reasoning, rather than numerical processing, and reaping the benefits of using human experience to control this symbolic processing.

1.5 ARTIFICIAL NEURAL NETWORKS

As stated previously the objective of intelligent control is to design a system with acceptable performance characteristics over a very wide range of uncertainty. The system must be robust enough to deal with unexpected occurences, large parameter variations, unquantified data, or extremely large quantities of data. Besides the approaches considered already, i.e. expert systems and fuzzy logic, an increasingly popular approach is to augment control systems with artificial neural networks.

Neural networks provide several appealing features for use in an intelligent control scenario. They allow nonalgorithmic information processing, i.e. no "programming" is required as in more conventional algorithmic signal processing. Neural nets purport to represent or simulate simplistically the activities of processes that occur in the human brain. Indeed multi-layered networks have been shown to develop very similar structures to existing human physiological structures with no human interaction or guidance. Also, the development of fast architectures makes implementation in realtime feasible, unlike artifical intelligence techniques which are infamous for their lengthy computation times.

Within engineering, neural networks are seen as an alternative technology by which information processing may be accomplished quickly and easily. To this end they have predominantly found application in pattern recognition and signal processing[61,62]. Until recently relatively little has been published on the application of neural nets to control. For control purposes, the control problem is reformulated as a pattern recognition problem. Control is seen as the mapping of measured signals for "*change*" into calculated controls for "*action*"[10].

1.5.1 Learning Algorithms

The power of neural networks is in their ability to learn and to store knowledge. Both of these important functions are achieved through adaptation of the synaptic weights assigned to each node's input. The weights are adjusted by a learning algorithm. Learning algorithms fall into two categories: supervised and unsupervised. The more popular supervised learning techniques employ a 'teacher' who presents the desired output to the net for a given input pattern. Unsupervised methods have no teacher and usually employ a local gradient algorithm to adjust the networks weights based around the activity near each particular node. The most commonly used learning algorithm is "*error-back propagation*"[61]. This allows adaptation of nets with hidden layers but requires an external teacher to guide it by supplying desired responses to each output node. Deviations from the desired results are used to train the network. Back propagation is similar in form to the basic "*delta rule*"[63] used to update weights in a single layered network[62].

Problems with these learning rules are that multiple training patterns are required to allow sufficient generalisation for good operation. Also, for each training pattern, the exact performance of each output node of the net must be known in order to produce the deltas. Obviously these constraints are quite severe.

To overcome these drawbacks a type of network has been developed which requires a '*critic*' instead of a teacher, called an *Associative Search Network* (ASN)[71]. ASN's employ evaluative feedback to learn using an algorithm called the *Associative Reward-Penalty*, or A_{r-p}, algorithm. It is recognised that environmental feedback may not be so informative as to provide individualised instruction to each adaptive element. Hence a scalar evaluation signal (critic) is used to assess the general performance (success/failure) of the ASN and this common scalar signal is used by all of the elements to adapt their weights accordingly. This type of learning is closely related to reinforcement learning as identified by psychologists in their study of human or animal learning[64]. It does not require a training sequence to be presented but will need a 'commissioning phase' in which it will act on the environment and adapt its weights in accordance with its task under the influence of the critic element.

Neural nets that employ unsupervised learning are often based upon Hebb's rule. Hebb's hypothesis is that repeated pairings of pre- and post-synaptic activity strengthens synaptic efficacy. This is really a clustering algorithm and hence is not necessarily useful to a system for improving performance in tasks determined by external factors. For example, the control of a plant with initially unknown dynamics requires a learning system that does not just cluster information but actively forms clusters that are useful. To do this some form of evaluative feedback from the environment is required either in the detailed form needed for error-back propagation or else a simple scalar success/failure signal as required by an ASN.

1.5.2 Applications in Control

As previously stated applications of neural networks to control are sparse in the literature although their use is gaining momentum. Applications tend to be more oriented towards pattern recognition, e.g. controller parameter updating and model generation, rather than as the actual feedback or feedforward regulator. Documented results are mostly simulations, both of the controlled plant and the neural net. Some examples of the approaches taken to apply neural networks to control are discussed below.

Very fast controller parameter updating can be achieved through the use of a neural network[65]. The advantage of using neural nets for this purpose is that the rate of convergence toward a steady state is essentially independent of the number of neurons in the network[66]. This compares favourably to other large scale dynamic systems, but is a feature of neural nets that is currently underutilized[65]. In the proposed architecture[66] the state variables of the neural estimator correspond exactly to the parameters of the controller. Thus a stable-state topology of this space can be designed so that the local minima corresponds to optimal control laws for the parameters of the controller. A major drawback of this approach is that to design a stable-state topology[67] much *a priori* information about the plant is required. Thus it is recognised that for simple plants this approach is not beneficial. However it is very worthwhile for large plants because it allows multiparameter adaptive control schemes to be realized with similar architectures and convergence rates. In [65] an example is presented using proportional derivative control of a single degree of freedom manipulator.

Proposals to use neural nets as regulators in place of conventional algorithmic control laws have been suggested. In this manner, the neural controller performs a specific form of adaptive control with the controller taking the form of a nonlinear multilayered network and the adaptable parameters as the weights between the neurons. Suggested topologies are shown in fig. 1.5[46] and fig. 1.6[68] each with a combined feedback/feedforward structure. It is to be noticed that as the learning process tunes the weights of the neural networks the error signal between the desired and actual plant responses is minimized. Since the error signal becomes small, training of the network will lead to a gradual switching from feedback to feedforward action.

Accounting for this fact some feedforward only schemes have been presented[68], i.e. only the feedforward controller implemented as a neural network. During the training sequence, features of the plant that are initially unknown and not taken into account by the control algorithm are learned. That is, the feedforward controller is adapted to compensate for the characteristics of the plant that are only learned during training. Random uncertainties are unpredictable and hence cannot be learned.

Fig. 1.7 illustrates a topology (*Specialised Learning Architecture*)[68] that allows training of the controller to operate in regions of specialisation only. The error ϵ=d-y is minimised but because of the location of the plant error back propagation may not be used directly. However if the plant is thought of as an additional, although unmodifiable layer, then a modified error back propagation algorithm may be used. The error (ϵ) is propagated back through the plant (the first layer of the 'network') using the partial derivatives of the plant at its operating points[68]. If the plant is a function of unknown form then the plant Jacobian may be approximated at each iteration using an equation given in [68]. This structure also has the advantage that it may be trained online - fine tuning itself while operating usefully (active adaptation). Stability problems will not arise with active adaptation if the learning rate is sufficiently slower than the time constants of the other components of the control system.

The application of ASN's to control has also been considered[69-71]. Their application to balancing an inverted pendulum is discussed in [69,71]. Two ASN's are employed: an *Associative Search Element* (ASE) and an *Adaptive Critic Element* (ACE). The ASE generates the actions to be applied to the plant. Since desired responses are not available active learning must be employed, i.e. the system learns while it is actually controlling the plant. Because learning only occurs upon failure, a more informative evaluation signal may be generated through the use of the ACE. The ACE receives the externally generated reinforcement signal which it uses along with the current states of the system to generate an improved reinforcement signal, used by the ASE. Effectively the ACE generates future predictions of reinforcement based upon a particular action being chosen. This allows

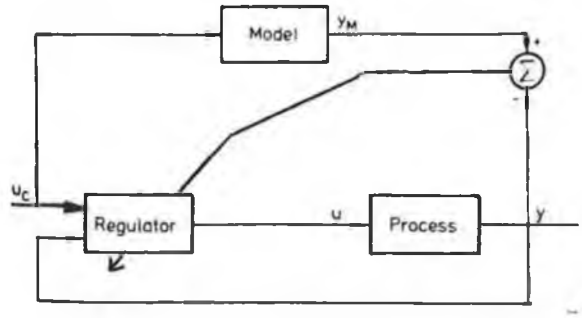

FIG. 1.5: Astrom's Neural Net Controller

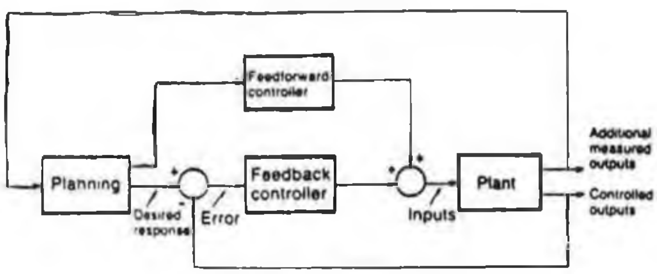

FIG. 1.6: Psaltis' Neural Net Controller

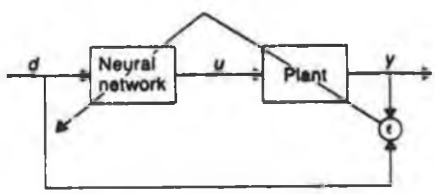

FIG. 1.7: Specialised Learning Architecture

learning throughout the system operation and not just upon failure. The predictive element is seen as the critical component of the system and it is expressed that the lack of prediction is a drawback of other neural networks[72].

1.5.3 Discussion

To summarise, neural nets are a new and useful tool for the design of control systems. Although at an early stage of development some useful and exciting applications have been proposed. Their use provides another form of adaptative control, but one based on physiological ideas rather than algorithmic control laws. This brings them into the realm of intelligent control although many problems have been highlighted. The topology of the control system is very important and determines exactly how learning may occur. A proposal to consider the plant as an additional unmodifiable layer seems to be worth further investigation. Another problem is the learning algorithm. The usual error back propagation employed by most neural networks requires a teacher which can supply each adaptive component of the output layer with its individual desired response for each pattern of the training sequence. Such knowledge is not always available in control situations. If detailed knowledge could be provided then more conventional control techniques could perhaps be applied more easily. Associative Search Networks seem to counter this problem by only requiring a scalar evaluation signal from the environment which is transmitted to all the adaptive elements. Learning performed by ASNs is analagous to animal reinforcement learning and utilizes active adaptation, i.e. the system learns itself as it actually operates on the plant. This is a desirable feature of an intelligent control system as is that of prediction. Predictions of the likely success of actions are very useful and can help to guide the learning process and thus deserve more research. Other aspects need to be further investigated before neural nets will interest practical end-users. Stability issues must be addressed both for online and offline training of the network. Application tools and techniques are required as are hardware implementations of neural nets for fast operation. Despite these problems neural networks should provide a useful system tool for the design of intelligent control systems.

1.6 LEARNING CONTROL SYSTEMS

The previous section introduced one of the central tenets of intelligent human behaviour, that is the ability to *learn*, to deliberately change the knowledge or knowledge structure of a system in such a way that the system can perform better on later repetitions of some given task. Not only is learning important to human systems but it has also been identified as an important aspect of intelligent control[7,73]. This can be seen from the previous discussion and the current trend of modern control theory. Control is moving towards consideration of systems which are imprecisely defined or about which there is very little or no information available[74], examples include adaptive control, robust design, use of neural networks, etc. Most of these systems are based on models or theories of human learning. The need to consider theories of human intelligent behaviour for intelligent control has been stressed[74].

1.6.1 Adaptive Control

Adaptive Control is probably the best known form of control that actively employs learning in the control loop. Adaptive controllers automatically adjust their controller settings to compensate for unexpected changes in the process or environment. The system learns or adapts itself to retain good control in response to uncertain time-varying process parameters. Thus, this technique is used for processes that are poorly modelled and/or which change in an unpredictable manner.

Adaptive controllers can be divided into two categories: Model Reference Adaptive Control (MRAC) and Self Tuning Control (STC)[4,5]. The objective of MRAC is to make the response of the closed loop system follow that of the reference model. One or more of the controller parameters is adjusted to achieve this.

STC strategies may be subdivided into two classes: explicit and implicit algorithms. With explicit or indirect methods the parameters of a process model are directly estimated, eg. the coefficients of a polynomial transfer function. These are then used by the control law to generate the control input. Typically it is assumed that the parameter estimates are correct ("certainty equivalence") thus leading to a suboptimal control[5]. However, as the estimates converge to their true values the control approaches the optimal[5]. With explicit algorithms the two stages are distinct and each may be designed separately. That is, any suitable estimation technique, eg. least squares, may be used as may any appropriate control law design, eg. pole placement or LQG. Implicit or direct algorithms estimate the actual parameters of the regulator, bypassing the model. Hence, only appropriate control law designs which can be configured in a suitable fashion may be used.

The actual learning is achieved through the use of the parameter estimation algorithm. Several procedures are available for this purpose: recursive least squares, extended least squares, instrumental variables, maximum likelihood, or stochastic approximation. Most of these techniques are very similar and only implement a form of learning equivalent to parameter learning as discussed previously. The learning process is completely pre-programmed and does not require any cognitive abilities in the control system. The most commonly used method is recursive least squares (RLS).

The problems with this form of learning is the great deal of *a priori* information required. For example, RLS needs good initial estimates of the error covariance matrix and also the parameter estimate matrix itself. An accurate estimation of the model order is required. Depending on the control law used, information such as deadtime, tuning values, etc, may also be needed. The adaptive system will also suffer from problems such as bursting, estimator windup, lack of presistent excitation, etc.[75]. Such a learning control scheme is very primitive but yet it has been proven to fulfill expectations and industrial needs very favourably.

1.6.2 Iterative Learning Control

A quite recent learning control approach does not require detailed knowledge about the process dynamics. This is called *iterative learning control* and is based on repeated iterations of the same task. It is thus ideally suited to robotic manipulator tasks as evidenced in the literature since it is a recursive online control method that relies on less calculation and requires less *a priori* knowledge about the system dynamics than the previously described adaptive control approach. Iterative learning control also allows tight tracking of a command trajectory, as required by robot manipulators, and can easily handle fast dynamics. This is unlike adaptive control which can achieve asymptotic positioning convergence but may not be capable of achieving tight trajectory tracking within an error bound.

There are several variations of iterative learning control in the literature. They are all based on similar principles, that is, repeated application of a simple algorithm to an unknown plant until perfect tracking is achieved. An early algorithm is[76,77]:

$$u_{k+1}(t) = u_k(t) + \Gamma\dot{e}_k(t)$$

$$e_k(t) = y_d(t) - y_k(t)$$

where $u_k(t)$ is the control input, $e_k(t)$ is the error, $y_k(t)$ is the actual system output, $y_d(t)$ is the desired system output. 'k' is the current iteration number and Γ is the controller gain matrix. Problems with this algorithm are that a noncausal operator is used (differentiator) and also that there is no guarantee of positional convergence. The choice of controller gain matrix is not trivial and requires specific knowledge of the manipulator dynamics.

An approach which alleviates the difficulty of determining *a priori* the controller gain utilizes a recursive least square parameter estimator to estimate the A and B state-space model matrices[78]. Effectively the method operates by generating a form of inverse system model which is used to update the control inputs, $u_{k+1}(t)$. Problems with this algorithm are the large amounts of initial data required. Initial estimates of A and B must be supplied as well as the initial error covariance matrix and forgetting factors. It was found that to operate correctly the system had to start from a known 'rest' position each time and also that the algorithm only corrected for small perturbations with respect to a nominal trajectory[78].

The previous 'conventional' algorithms only use the last iteration data pair (i.e. $u_k(t)$, $e_k(t)$) to learn. Intuitively, human learning systems gather several adjacent points around the point to be computed, to acquire knowledge of the 'trend' and to decide what to do for the next iteration. It also seems intuitive that faster convergence would be achieved if more past history data points are used by the learning process. Using this reasoning, the following algorithm was derived[79]:

$$u_{k+1}(t) = u_k(t) + \frac{1}{h} \cdot \sum_{j=-m}^{m} a_j . e_k(t-j+1)$$

where h is the distance between adjacent points (eg. sampling interval) and a_j's are weighting values determined by the number of points summated. The net effect of this algorithm is that it obviates the need for the noncausal operator in [77].

Use of multiple past history data points can siqnificantly improve the convergence performance of iterative learning controllers. The previous algorithms looked at can be classed as first order methods in that they only utilize one previous past history pair (i.e. $u_k(t)$ and $e_k(t)$). An N^{th} order algorithm uses N consecutive pairs[79]:

$$u_{k+1}(t) = P_1 u_k(t) + P_2 u_{k-1}(t) + \cdots + P_N u_{k-N+1}(t) + Q_1 e_k(t) + Q_2 e_{k-1}(t) + \cdots + Q_N e_{k-N+1}(t)$$

The P_i's and Q_i's must be chosen for fast convergence conditions. An example is presented in [79], however positional convergence may not be guaranteed as the output velocity, $\dot{y}(t)$, only converges to the desired velocity, $\dot{y}_d(t)$.

Some other variations of iterative learning control have been presented[80-82]. A major problem with all of the algorithms is that as the variety of specified tasks increases the total number of operations also increases and the amount of data to be stored in the form of input patterns grows excessively large. To solve this problem some early development work is being carried out to investigate if an input pattern corresponding to a new desired trajectory may be produced from previously stored patterns[83]. This involves investigation of how to decompose the problem of following a complex trajectory into a set of subproblems, each one referring to an already learned simpler trajectory. To this end, research involves consideration of reproducing the same spatial trajectory but in a different time frame or time scale, and also how to combine basic input patterns when there is no spatial restriction[83].

The previous discussion of iterative learning control methods has highlighted some behavioural similarities with neural networks. The task objective must be thought to the iterative learning controller before it can track a trajectory with no error. This is similar to the training sequences that must be presented to a neural network. They each update their "internal state" on each iteration of the training pattern, and more training results in better performance from the system. However, neural networks can generalise if presented with enough sufficiently different examples. This compares very favourably to the total lack of generalisation available with iterative learning control. Also it has been shown that neural networks may classify patterns or discover important 'concepts' which are held in their hidden layers which could be considered a form of concept learning.

1.6.3 Learning Automata

Learning automata are another learning tool that has been suggested and used for intelligent control. Learning automata are based on mathematical learning models developed in psychology[84]. They attempt to find an optimal action from a set of allowable actions. Typically they start with no information as to which action is optimal, with equal probabilities attached to each action initially. One action is chosen at random and the response of the environment is observed. Based on this, the action probabilities are changed and a new action is chosen. This process is then repeated, ensuring that the optimal action is always chosen.

Mathematically, at each instant 'n' the automaton selects an action $\alpha(n)$ from a finite set $\{\alpha_i \mid i=1,2,\dots,r\}$. The selection is based on a probability distribution $p(n)$, where $p = [p_1,p_2,\dots,p_r]^T$. The environment gives a response $b(n)$ at time 'n', where $b(n)$ is an element of $\beta=\{0,1\}$ ('1' is a penalty and '0' is a reward). The environment penalizes the automaton with the penalty c_i, the probability of failure when the input is α_i

$$\Pr[b(n)=1 \mid \alpha(n)=\alpha_i] = c_i$$

The environment characteristics are described by the set of penalty probabilities $C = \{c_i \mid i=1,2,\dots r\}$. The environment is said to be stationary (non-stationary) if the c_i's are (are not) constant from stage to stage. Based on the response $b(n)$, the action probabilities vector p(n) is updated and a new action is chosen at 'n+1'.

Updating of the action probabilities is called "reinforcement" and several schemes, mostly based on psychological models of animal and human learning, have been proposed. A linear reward-penalty (L_{R-P}) scheme increases the probability $p_i(n)$ if action α_i leads to a success. All $p_j(n)$ ($j \neq i$) are also linearly decreased. If failure results from action α_i then $p_i(n)$ is decreased while all $p_j(n)$ ($j \neq i$) are increased. A linear reward-inaction (L_{R-I}) scheme works exactly as the L_{R-P} scheme when successful actions are observed, while all the probabilities are left unchanged in the event of failure. A linear reward-ϵpenalty ($L_{R-\epsilon P}$) scheme decreases $p_i(n)$ due to a failure but only fractionally in comparsion to the increase in $p_i(n)$ due to a success. Convergency proofs have been given for each of these schemes[84]. Other schemes have been presented[eg.85,86] but most are derivations of the previous three.

There are obvious similarities between learning automata and neural networks. Each requires 'training' before optimum performance is achieved. Error differences between the desired and actual outputs are used to update the 'internal state' of the system to guide it towards its objective. There are particular similarities between learning automata and associative search networks (see section 5.2.3). Each operates on the environment directly in order to learn optimum actions. The ASN uses an associative reward-penalty scheme to penalise actions which lead to failure and reward successful actions. This is very similar to the L_{R-P} scheme mentioned above. Integration of the two methods could perhaps lead to some further advancement for control. It would perhaps be worthwhile to investigate the use of similar schemes to L_{R-I} and $L_{R-\epsilon P}$ in ASN's which have better convergence properties than the L_{R-P} scheme. The importance of, and the significant improvements obtained through the use of prediction with ASN's could perhaps be employed by learning automata. The previous discussion of learning automata only dealt with P-models where the response set of the environment is binary. Q-models allow the response to take on a finite number of values and S-models allow values on a closed interval. Results for P-models have been extended to Q and S-models and their use could be tried with ASN's to improve performance further.

Another relationship exists between learning automata and fuzzy controllers, in particular with Q-models in which the environment response may take on one of several possibilities. Integration of the reinforcement learning schemes into fuzzy controllers would provide an ideal framework for advanced development of learning in a fuzzy set environment. The use of learning in fuzzy controllers has thus far been limited and of an "*ad hoc*" nature (see section 4).

1.6.4 AI Approaches to Learning

Learning automata still only employ a form of quantitative, or parameter, learning. That is they simply update parameters according to some pre-programmed scheme. In contrast to this approach to learning the discipline of AI is trying to develop concept learning techniques. Typically however research has been in a "*blocks world*", where everything is known and there can be no conflict of evidence or lack of information. Although such work may lead to general principles of concept learning there is still much research needed to deal with realistic environments which are highly uncertain, in particular for control problems.

A common successful AI approach to learning is by *induction*. Sets of examples, or tables, are presented to the system. The system examines these examples and can produce a set of production rules:

IF situation DO action *OR* IF situation MAKE inference.

Using a 'teacher' as before rules may be altered in order to specialise, i.e. by adding a conjunct to the antecedent of the rule, or indeed to generalize the rule. Thus concept definition is automatically performed by considering both positive and negative examples of a concept. A well known program which does this is that of Winston[87,88]. The concept of an "arch" was learned in a blocks world environment. Other approaches to learning by induction include matching, analogy, and indexing[87]. Most of these centre around some form of knowledge organisation, eg. schemas, scripts, "isa" hierarchy, etc., and a method of searching among this knowledge. The main problem however is that all examples are taken as being equally true and important but no allowance is made for "noisy data", i.e. a false example. Obviously this scenario is unsuitable for realistic application domains.

Learning by exploration defines a situation where a system starts with little or no knowledge and does not even have a goal-state or objective as its aim. Two programmes which implement this form of learning are *AM* and *Eurisko*[87]. They do not search a space as the previously described learning programmes but *'explore'* their domain looking for interesting patterns and generalizations. AM was designed for a mathematics domain while Eurisko may operate in several different domains. They maintain and grow a large database of "concepts" (eg. 'set' and 'function' in mathematics). Each concept has "slots" which point to other objects (eg. concepts) which are related. New concepts are created and their slots are filled in by the learning process. This is done by keeping an agenda of *tasks* which have to be done. Each task is an attempt to fill in a slot of a concept, but this usually creates more concepts, slots, and tasks during this process. The *agenda* of tasks is sorted in order of "interestingness", where each task is rated depending on how many examples there are to look at, the rating of the task which created it originally, and other tasks which have since suggested it again. If all task ratings are below a certain threshold then a "*suggestions*" slot for each concept is used to generate new tasks. Suggestions are heuristics written by the designer to keep the system running. Working on a task is also performed using heuristics. These systems have been very successful and have given an insight into how learning may be implemented automatically. For instance, AM was able to discover numbers, addition, multiplication, and to make conjectures about these ideas simply by starting with the concepts of 'set' and 'bag' (sets which may have duplicate elements).

Other approaches to investigate learning in AI include GPS (General Problem Solver), Hacker (which uses *failure driven learning*), Foo/Bar (game playing), and others[87,88]. However, most of these attempts to simulate higher level concept learning operate in naive environments, eg. the blocks world, and need much further research to enable successful use of the techniques in a realworld environment as part of the upper hierarchical levels in an intelligent control system.

1.7 COGNITIVE CONTROLLERS AND OTHER ASPECTS OF INTELLIGENT CONTROL

7.1 Cognitive Structures

Neural network controllers may be classified under the category of *Cognitive Controllers*. Cognitive controllers entertain structures or principles similar to those utilized in human cognition. Such systems embody recognition or perception in the control loop and use 'knowledge' to achieve their goal in complex or uncertain environments. They can learn about their environment as they operate within it. It is recognised that cognitive controllers form an important part of intelligent control[26,89,90].

Control of complex systems requires the use of multiple sensors and often hierarchical organisation of the control system[91,92]. Integration of the data for multiple sensors and propagation of observations among levels of the hierarchy leads to *"perception in the control loop"*[89,91]. Sensor fusion in itself is a separate issue of intelligent control (see section 1.6.2). Once the principle of perception is recognised as necessary to intelligent control another issue arises: knowledge and knowledge representation[26,89,93]. Other aspects which arise are decision making and planning using this knowledge[89], and learning which involves the incorporation and proper utilization of new information gained during operation[26,89,90]. Processes and techniques to deal with such aspects of *intelligence* have arisen from studies of human cognitive abilities. Cognitive and behavioural psychologists have proposed theories of how humans organise knowledge and data in memory, how decisions and plans are made, and how learning occurs (from quantitative to conceptual learning)[64,94]. These theories parallel techniques developed in AI to reproduce cognitive capabilities on a computer [87,88] and indeed there has been much interaction between the two fields. Although metaphorical models of human cognition may be unsuited to control, the development of an intelligent controller would appear to necessitate consideration of human cognitive abilities and to design along these lines.

Human cognitive systems have been shown to be organised in a hierarchical fashion, with perception and a hierarchical knowledge base[91]. A control system called CMAC (Cerebellar Modullar Arithmetic Controller) was designed along the lines of this model [91,92]. Cognitive abilities allow not only incorporation of knowledge about the controlled system but also a *world representation* to incorporate knowledge of the environment, and the context of the performance. In addition, learning may also be encapsulated. However, learning modes beyond the simple quantitative learning of adaptive controllers are looked for, that is "conceptual learning" (the creation of new concepts). Further discussion of learning systems is presented in section 6.4.

Symbolicprocessing is the medium by which higher level intelligent functions are performed[88,94] whereas interfacing with the real world in realtime still requires numerical processing techniques. The integration of numerical and symbolic processing is a problem to be dealt with in the design of cognitive controllers[22]. It can be seen that much research is still required in the context of cognitive controllers. Although there are few references to this class of controllers in the literature many current strands of research are encompassed, eg. expert systems, fuzzy logic, neural nets, and learning controllers. Among the issues that need to be looked at are measures of performance and reliability of cognitive controllers in comparasion to conventional controllers and cognitive controller design.

7.2 Sensor Fusion

Sensor data fusion is an important aspect of intelligent control. Architectures of intelligent controllers are generally conceived as being hierarchical in nature. Thus, the integration of perception and action at multiple levels of resolution is required. Although intelligent control is not constrained to a particular application environment, this particular aspect is best described with respect to control of an autonomous robot in an uncertain environment. The literature on sensor fusion is mostly concerned with this scenario.

To operate effectively and efficiently autonomous robots must be able to organise and integrate observations from many different, or disparate, sensors to provide information with which to build a robust world model. Each level of the hierarchy is thus characterised by incomplete knowledge structures with varying degrees of refinement. Sensor fusion deals with the propagation and organisation of disparate sensory evidence within and between these incomplete knowledge structures. Multiple sensors are required because each sensor is limited both in the observations or uses that it can make of the environment and also in its useful range in which its observations are realible and accurate. Sensors of autonomous robots can include stereo vision, sonar data, active ranging, and tactile sensing. The use of many disparate sensors provides a robust, realiable, and consistent world model. However, to achieve this it is required to find computationally efficient algorithms for the propagation of incomplete evidence over a network of incompatible frames of discernment. The main problems with this objective are:

1) *sensor model*: each sensor provides information in its own sensor specific level of abstraction (eg. vision vs. range data).
2) *conflict resolution*: sensors can provide conflicting evidence of the same object.
3) *uncertainty*: there are degrees of uncertainty of sensor data (there is also the problem of a lack of information from the sensor).
4) *sensor failure*: recognition of the failure of a sensor and disregard for its observations.

There has been much research effort in the development of advanced individual sensor subsystems, eg. vision and tactile sensing. Subsystems are then combined into an overall architecture for control of flexible manipulators[23,95,96]. There are many examples in the literature of highly application specific techniques of combining disparate sensors subsystems, eg.[95-99]. These fall under the classification of *guiding* techniques. That is, data from one sensor is used to guide the control or processing of other more accurate estimates from other sensors.

More general mathematical approaches to the sensor fusion problem are emerging[100]. Some simple methods have been described for the fusion of sensor data about the same object/environment features[101]. These methods typically use a distributed blackboard architecture. One technique chooses one of the sensor values as the "fused" value. The selection could be based on the value with the highest confidence measure, or through the use of heuristics of the present situation (i.e. state of the environment). Another method calculates an average value from all the sensor readings. Corresponding confidence values may also be averaged.

Another more formal probabilistic approach uses Gaussian probability density functions and measures of confidence for the sensor readings[102]. A '*confidence distance measure*' is defined as a criterion for detecting sensor errors. These are distance measurements for two sensors measuring the same object property. An mxm matrix ('m' sensors) is formed of elements d_{ij}, relating each sensor to one another. Thresholding the elements of this matrix allows a directed graph to be compiled. This highlights the group of sensors whose data values agree with and support each other. All other sensors are suspected of error. The optimal fused sensor data property value can be computed from the group of agreeable sensors. This technique was implemented using a frame type scheme for knowledge representation[102].

In another approach each sensor is treated as an individual decision-maker, i.e. a member of a team with common goals[103,104]. Each sensor is considered as a source of uncertain geometric information, able to communicate to and coordinate with other members of the sensing team through a blackboard architecture. A model of the environment based on stochastic geometry is used, where the uncertainty in each vector is described as a probability distribution on the parameter vector. Uncertain points, lines, curves, and surfaces are considered. The invariant topology of relations between uncertain geometric features is used for propagating observations through the world model. The integration process uses a Bayes procedure to compare disparate observations of geometric features rejecting spurious measurements, and providing partial updates of object locations to the world model. The method has been applied to a robot system comprising an active stereo camera and a force/tactile gripper[103].

1.7.3 Reasoning with Uncertainty

Accepting that intelligent control requires autonomous actions and planning and that real life situations embody uncertainty in all aspects of the environment then clearly an intelligent controller should be provided with mechanisms for handling uncertain information. Handling uncertainty is in fact one of the key issues of expert systems. Elucidating heuristic rules from experts provides the basis for a workable system but even experts do not have complete understanding of complex domains. The expert system must be able to represent uncertain heuristics and perform inferences with them. A knowledge representation scheme must be used that can encode uncertain knowledge and data, and the inference mechanisms must be provided with abilities to handle uncertainties and conflicts. Such abilities are required not alone by expert systems but by any *intelligent machine* trying to operate fully or semi-autonomously and which must make decisions and plans in a highly uncertain real world environment.

To express uncertain knowledge, a scheme is required that allows a proposition to have a truth value other than true or false. The truth space, for example, may be expanded to allow levels of truth or falsity to be expressed, perhaps as a real number or as some subinterval between 0 and 1 (representing falsity and truth respectively). One approach to this has already been looked at, i.e. fuzzy logic. In rule-based expert systems, 'if-then' rules generally have attached some indication of the strength of correlation between the condition and the consequent. This may be quantified as a number or as a 'linguistic' qualification of the rule. Inference techniques employed under uncertainty propagate numeric values through the inference chain using one of the uncertainty calculi available. The four common calculi are: probability (or Bayes) theory, MYCIN (logic of confirmation), belief (orDempster-Shafer) theory, and possibility (or Fuzzy Set) theory[105,106]. Other approaches are similar to the above and include plausible inferences, INFERNO, linguistic reasoning, nonmonotonic logic, theory of endorsement[106], and evidential reasoning[107].

Each of the calculi offers a different perspective and each manipulates uncertain information in a different manner. Each has advantages and disadvantages and the choice of which to build into the inference engine of an expert system should be carefully considered based on the nature of the application.

1.8 CONCLUSIONS

A detailed survey of intelligent control has been presented. The need for and the development of this discipline has been precipitated by progress in computer science and technolgy, developments in control theory, the movement toward more complete overall or plant-wide control with greater autonomy from human intervention, and advancements in the understanding of human intelligence and perception. Intelligent control has been proposed as the search toward systems that can operate with greater levels of uncertainty than currently accepted methods (eg. adaptive control). It should provide an approach to the control of large-scale complex systems and be able to deal with the inherent problems that such systems present. For example, high-level planning and decision-making capabilities which are robust in the event of failure or large uncertainties are required. Needed too are techniques for sensor fusion and high-level knowledge representation (eg. symbolic models of the environment). Methods and technologies to accomodate efficient communications and interfacing of numeric and symbolic processing is required.

Several major strands of research which fall into the category of intelligent control have been discussed. Typically architectures proposed have been hierarchical with some level of cognitive representation or relationship to human knowledge organisation. Expert systems are finding widespread use as a means to expand the operation range or performance of control systems through the integration of human expertise in the form of heuristics. Fuzzy controllers have been developed which can operate in environments with very large degrees of uncertainty. Fuzzy logic is a decision making tool similar to human reasoning abilities. The problem of reasoning with uncertainty is being examined and several calculi have been proposed for this purpose as well as techniques to use these in expert system architectures. Research into methods of multiple sensor data fusion is gaining momentum as is investigations of artificial neural networks for control. Such systems require no explicit programming and 'learn' how to perform in an optimum fashion. Learning has been identified as a key element of further advancement. To perform the tasks mentioned some learning capability is either highly desirable or essential. Several approaches to learning have been developed from low-level iterative learning control and adaptive control to higher level AI learning programs. It is essential that learning abilities are extended beyond that of simple adaptive controllers which operate with very limited uncertainties and that effective means of bridging the gap between these low-level parameter learning systems and the naive high-level concept learning systems developed in AI.

Research in intelligent control will depend less on traditional engineering principles and more on ideas originating in other scientific fields. Areas which have already had major influence are artificial intelligence, cognitive psychology, and connectionism. Other areas may be highlighted in the future. Norbert Weiner's dream of *cybernetics*, machines in the service of man and machines imitating man, may yet have a significant role to play as a source of new ideas. Is it not realistic to suggest that better and more intelligent machines and controllers may be designed inspired by the structures discovered in biology and psychology? Could animal and human behaviour be used as a benchmark against which to judge the performance of intelligent machines? Obviously, we are not at the stage where machines and controllers may be designed that can imitate man and his abilities. Thus the use of the word *intelligence* may seem premature. However, the current state of control theory development is moving beyond simple engineering of better feedback loops and toward better and easier performance attainment and system integration. It is moving into areas which seem to require 'intelligent capabilities'.

The field of intelligent control is still new and immature and hence is very difficult to review within a unified framework. The approaches presented here reflect current views of the subject both of the researchers within those approaches and those of the author. It is expected that as the subject developes and progresses further refinement may be made and some of the ideas presented here will be strenghtened while others may be discarded. However, due to the nature of the discipline, it is anticipated that future approaches will still have the flavour of control theory, artificial intelligence, and behavioural\cognitive psychology.

CHAPTER 2

A FRAMEWORK FOR INTELLIGENT CONTROL

2.1 INTRODUCTION

The exposition of intelligent control, presented in the previous chapter, highlighted the many diverse and varied research areas that may be classified in this novel field of control research. It is a multi-disciplinary area with research progressing along many independent paths with little coordination or cooperation between them[73]. It is clear from chapter 1 that there is some controversy over what exactly constitutes intelligent control. Several different definitions have been proposed. Some application areas have received much more attention than others eg. robotics and autonomous systems, resulting in an exclusive relationship between these areas and intelligent control. Similarly, use of particular tools is considered to imply 'intelligent' control, eg. use of expert systems or neural networks. Another very noticeable aspect of the survey is the lack of attention given to the low-level controller used as the basis of the overall control system. Most research is geared towards the development of higher level strategies for organising, planning, fault detection and correction, etc. It is often simply stated that the low-level controller is an adaptive algorithm derived from one of the many modern control design techniques available. Sometimes a nonadaptive or even PID control law is assumed. Despite the improvements in performance that may be achieved with the use of AI and other methods in combination with current control laws, there still exists a limitation due to the type of low-level control algorithm employed. It is thus essential that intelligent control research should also focus on suitable low-level controllers and on the characteristics which these should have.

To enhance future development a unifying framework is required. This should incorporate the diverse areas currently under investigation and suggest new areas for consideration. The framework should encompass research on both high and low levels of a control system and also the tools that are necessary to implement intelligent controllers.

Several themes underlie the nature of the topics discussed in chapter 1. The major theme is the pursuit of better control via the emulation of human intellectual abilities (eg. thinking, planning, memory) or by the incorporation of human expertise and intelligence in a controller (eg. heuristics in an expert system). This is seen in the sections concerned with expert systems, fuzzy logic, and decision making. Neural networks are models of the neuronal structure of human thinking and action. Multi-sensor integration is one of the most important human attributes providing the great flexibility and robustness associated with human performance. Learning control systems try to emulate the adaptibility of human systems and their learning capabilities.

In line with these 'intelligence' themes a new definition and framework for the design of intelligent controllers will be proposed. This will then be used to design a suitable low-level control algorithm as the basis of an hierarchical system. To do this aspects of human intelligence will be explored. Psychological theories of intelligent behaviour will be discussed and important necessary conditions extracted. Learning is identified as a necessary condition and possible simulations of animal learning theories are examined. Other important attributes of learning are extracted as desirable for control.

2.2 DEFINITION OF INTELLIGENT CONTROL

From earliest considerations, intelligent control has been intimately linked to human system emulation[7]. It's objective was viewed as the transferral of human abilities to control systems to achieve greater autonomy. Necessary abilities for this were identified as learning, problem solving and planning, environment sensing and world modelling[7]. The need for hierarchical structures was also expressed. The attainment of these goals was conceptualized as resulting from the merging of research in AI and automatic control. As can be seen from the survey of chapter 1, much current intelligent control research retains this flavour. Later definitions (eg. see fig. 1.1) were extensions of this viewpoint[9,12]. Although this was a useful starting point for incorporating intelligence into control strategies the result has been a concentration of research on the higher level intelligent functions required.

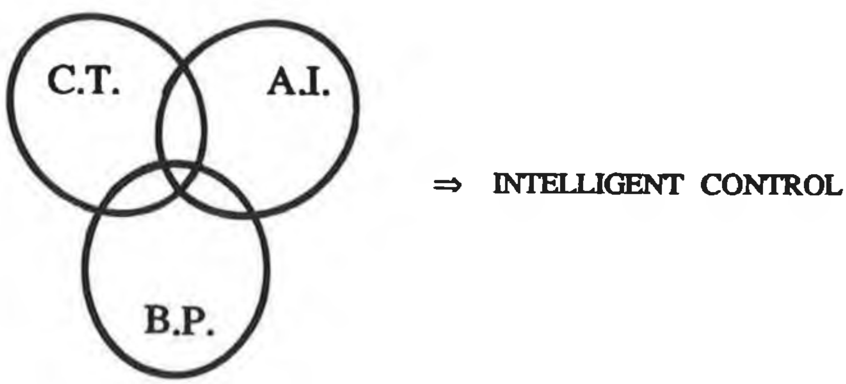

FIG. 2.1: A New Definition of Intelligent Control

The current state of the field of intelligent control is not unlike that of the early stages of computer vision research. Most work was concentrated on application specific solutions to problems. There was a need for a unifying framework within which the many diverse and unconnected strands of research could be gathered together. In response a computational theory was proposed based on information processing guided by the study of human visual systems[108]. This led to a strengthening of the core principles and a flourishing of new ideas.

A similar theory is required for the field of intelligent control research. This should encompass current research strands yet concentrate on the underlying characteristics desired and not on the tools used or the applied areas. Intelligent control is redefined here as (see fig. 2.1):

> *"The pursuit of improved control performance through emulation of intelligent human control behaviour by application of techniques from the fields of AI, behavioural psychology, and control theory".*

This definition is similar to earlier ones but differs in that the use of intelligent human behaviour as a reference model is explicitly stated. The study of intelligent behaviour provides a source of constraints for building a framework similar to the computational theory for computer vision[108]. A set of processes or attributes may thus be defined as necessary conditions for intelligent control. The framework becomes independent of the tools and application specific solutions and focusses attention on the underlying principles or attributes employed[73].

Of the definition given above, it is envisaged that the field of control theory shall have a dominant role. This is a rich field in which a great body of control knowledge already exists. It is expected that Intelligent Control will provide a vehicle for the further development and expansion of current control methods and ideas. The role of the fields of AI and behavioural psychology will be to guide and direct these developments by the provision of novel ideas and tools. Artificial Intelligence[87,88] is concerned with the simulation of higher level intelligent abilities and has provided some useful tools for their implementation, e.g. expert systems. Behavioural and cognitive psychology[64,94] concerns itself with the study of animal and human behaviour. It provides the principles upon which the framework shall be built by highlighting different attributes or processes that humans exhibit in control situations.

The framework will encompass current research work. The use of expert systems provide a successful tool to explicitly embody human expertise in the form of heuristics. Production systems are in fact a model of stimulus-response behaviour. Fuzzy logic and uncertainty calculi are methods to simulate human reasoning processes. Approaches to sensor fusion have often taken inspiration from the mechanisms by which humans accomplish this task so easily. Neural networks and iterative learning control are approaches to emulate the learning abilities demonstrated in human performance. However, no emphasis has been placed on determining general attributes that an intelligent controller should exhibit. The framework will allow such a task to be performed. It also provides impetus to consider suitable low-level control techniques and not just high level intelligent functions. These should be designed based on human behaviour and the study of motor skills acquisition.

2.3 INTELLIGENT BEHAVIOUR

Many different theories have been proposed to try to explain the phenomena of human intelligence, or in particular intelligent behaviour[64,91,94,108,109]. Intelligence is a concept which is much easier to recognise in behaviour than it is to define or measure. Theories of human intelligence and behaviour are usually classified into two categories:

Behaviourism: An entity responds only to changing environmental inputs. *Purposive* behaviour considers goal-seeking as an extension of this.

Cognitivism: Intelligence is defined by the structure and existence of mental faculties for understanding and cognition.

Both theories can contribute to the framework as will be shown later. However it should be noted that although it is proposed to emulate intelligent behaviour only aspects relating to control will be considered.

2.3.1 Behaviourism

The school of behavioural psychology believes that correct scientific study of human actions and behaviour should not be concerned with introspective measurements of internal, or cognitive processes[110]. Behavioural theories are based purely on environmental stimuli and response pairings. This school is typified by the classical conditioning experiments of Ivan Pavlov (in 1904) causing a dog to salivate in response to a bell ring. The bell stimulus is paired with food presented to the dog and eventually can be used to generate the same response as would the food stimulus.

In psychological terms this can be described as follows. A subject is repeatedly presented with a neutral conditioned stimulus (CS), eg. the bell, that does not cause any particular response. The CS is followed by an unconditioned stimulus (UCS), eg. food, which causes a reflexive or unconditioned response (UCR), eg. salivation. After several pairings of the CS and UCS-UCR the CS comes to elicit a response of its own, i.e. a conditioned response (CR), which closely resembles all or part of the UCR. This is the basis of behavioural explanations of learning[64].

Behavioural psychology ignores internal mechanisms or drives and relates all responses\behaviours to the set of external stimuli then present. Although this approach may explain the actions of lower forms of animals it is not sufficient to explain the complex intelligent behaviour of humans. Despite this, theories of behavioural responses and learning may be useful for the design of low-level control laws as part of an intelligent control hierarchy. The control law may be envisaged as a black-box which acts on the environment or plant and monitors the corresponding outputs. This behaviour may be considered in terms of stimulus-response actions and behavioural theories used to improve the control performance by emulation of human performance. Learning theories may be used to enhance the operation of controllers used on plants with large uncertainties. Theories of behavioural learning have also been used as updating rules for neural networks, eg. associative reward-penalty scheme for ASNs (see chapter 1).

An attempt has been made to extend behavioural psychology to describe more complex actions comparable to human behaviour[111]. These *purposive* behaviour theories includes the type of reflexive responses mentioned earlier, eg. salivating, pulling away from fire, etc., called *respondent* behaviour. They also describe *operant* behaviour which is an unsolicited action on the environment to secure a desired response. The relationship of behaviour with goals or purposes is emphasised. Operant behaviour is originated by the organism and is not a result of external stimuli, although attainment of the desired response will still be guided by external stimuli present. Human behaviour of operant type includes walking, talking, playing, and working. Goal-seeking behaviour is analagous to feedback control. It is thus realistic to expect that models of such behaviour may be used to enhance feedback control design.

Purposive behaviour takes account of an individual's cognitions, i.e. perceptions and beliefs about the world. These are termed intervening variables and experience with certain stimuli results in the formation of corresponding cognitions. In addition, particular needs produce demands for certain goal objects (eg. thirst produces a demand for water). Cognitions and demands are intervening variables which act together to generate responses. Learning is said to occur when cognitions are formed relating responses to rewards, i.e. the internal world model is updated. Cognitions are formed or modified when predicted and observed responses differ. Many forms of cognitions are formed, particularly about the way the environment is structured. Cognitions tend to model the world and the inter-relationship of objects and entities. These are used to predict the outcomes of planned actions. Individuals can respond adaptively to new situations by combining cognitions from several different learning experiences. Several other cognitive behaviouristic theories[64,91] have been proposed for explaining intelligent behaviour.

2.3.2 Cognitivism

In contrast to behaviourism, cognitivists believe that intelligent action cannot simply be described in terms of stimulus-response pairs or interaction. Instead, the underlying principles that enable intelligent action are the internal computational structures based around the central nervous system and the brain[94]. Intelligence is an information processing activity in which sensory input is processed by certain processing modules. Cognitive psychology is usually defined[112] as the study of processes by which the sensory input is transformed, reduced, elaborated, stored, recovered, and used. This viewpoint is closely related to the work of researchers in AI where methods of synthesizing intelligent behaviour are sought through the use of a symbolic processing approach[87,88].

Since information or knowledge manipulation is considered as the route of intelligent behaviour one aspect of human cognitivist processes normally studied is memory. The physical mechanisms are ignored and study is concentrated on abstracting the methods by which information is stored, remembered, and forgotten. Other aspects of cognitive psychology include investigation of processes like perception (transforming sensory input into a cognitive code), attention, language, decision making abilities, etc[94]. Learning is considered as the acquisition, restructuring, and generalisation of knowledge via the set of computational processes provided.

Sensory processing is initially a pattern recognition task which tries to abstract the input patterns into cognitive codes or units of information. The process of selective attention is very important here since much of the sensory input is unnecessary information and only a small proportion is usually important or useful. The information channels for reception of sensory data are normally considered to be of restricted capacity. Such studies have obvious interest for the design of multiple sensor fusion systems especially with regard to the task of data integration.

Generally, it is agreed that there are three types of memory: sensory (inital images), short term (fixed capacity 'workpad'), and long term storage (large capacity with complex knowledge structures). Methods of encoding, storing, and retrieving data from long term storage are discussed. A very strong link exists with the AI field in that some theories of how knowledge is stored in memory have been used to structure knowledge for AI programs and vice versa. Theories of the organisation and operation of memory are very useful for the design of large complex systems as required by intelligent control.

Another important subtopic of cognitivism is how to use effectively the knowledge stored in memory. Associated with this is planning, problem solving and reasoning. Prediction of possible outcomes is again intimately associated with these tasks. Theories of human thinking and reasoning have been developed and some formal mathematical algebras proposed. Some of these were reviewed in chapter 1.

Cognitivism, it may be observed, is mostly concerned with conscious experience. It can provide much stimulus to the design of sensory and cognitive processing systems for higher level intelligent functions as described above. But experience must be translated into objective control behaviour. Thus behaviourism should not be disregarded totally. Both theories have a role to play within the defined framework.

2.4 LEARNED BEHAVIOUR

Although the exact nature of intelligence remains unclear, the ability to learn is a crucial and multi-faceted element of intelligent behaviour. It is reflected in both behaviourist and cognitivist theories in which it has a major role. In relation to human and animal learning there are again two main schools of thought: behaviourist and cognitivist.

Behaviourists describe learning as the association of new and unique stimuli with particular responses[64,91]. A famous example is classical conditioning as demonstrated by Pavlov's dog experiment. Learning is guided by negative and positive reinforcers. Positive reinforcements, rewards (eg. pleasure, success), tend to increase the probability of a particular response to certain stimuli. Negative reinforcement with punishment (eg. fear, pain) reduces the probability of a particular response. Classical conditioning is used to explain respondent behaviour. The CS is always followed by the UCS regardless of the response. Operant or instrumental conditioning can explain the learning of new operant behaviours. Reinforcement is only provided when the response is suitable.

Cognitivists view learning with regard to intervening variables and information flow[94]. Learning is recognised as the aquisition of knowledge. Intervening variables are information models about the organisation and structure of the environment. Cognitivist theories are based around this information processing paradigm in which sensory data is abstracted and integrated at different levels to provide knowledge about the environment (i.e. internal world models). Learning therefore is the gathering of information for the refinement of these models and generation of new models. Learning is also considered as the generalisation of knowledge discovered in one application for use in other similar situations.

2.4.1 Role for Prediction

Prediction is a major element of human learning. Once a pattern, or model, is recognised then deviations from the predicted norm can be recognised. There is large information content in the difference between what is predicted and what is actually observed. Recognitions of gradual or small systematic deviations from the predictable lead to the learning of more sophisticated recognitions and predictions. It can also lead to the modification of the model, or expectations, to match that observed. Humans in fact seem unable to cope with their environment if the outcome of their actions are unpredictable[91]. Abrupt deviations from the predictable produce strong emotions, eg. surprise, anger, fear, etc. Successful animal behaviour and survival is much more likely if prediction is possible.

In classical conditioning it has been observed that a response can begin earlier than a stimulus that previously elicited it[72]. A particular behaviourist theory[113] states that learning can only occur *"when events violate their expectations"*. This implies a form of error-correction learning strategy. Expectations or predictions are generated by current stimuli patterns. When subsequent events disagree modifications are made to improve future expectations. From a cognitive viewpoint learning can also occur by deviations from a predicted trajectory. When predictions of planning results deviate from those actually observed then plans or actions can be updated accordingly. Internal models can be updated or learned by the use of prediction to guide parameter adjustments. Similarly, new models can be used when there are large deviations between predicted and observed outcomes. There is obvious interplay between learning and prediction. Prediction guides learning while learning improves the prediction processes.

2.4.2 Hierarchy of Learning

Several hierarchical learning theories have been proposed[64]. It is of course obvious that learning can occur at many levels of cognition. It can, for example, range from simple gathering of static information about objects, eg. colours and shapes, to devising rules about how objects in the environment interact.

Piaget's theory of intellectual development[114] describes four discrete stages in the development of children. Increasingly well-articulated and interrelated representations are learned for interpretating the world. The first *sensorimotor stage* (years 0-2) involves trial-and-error manipulation experiments. A simple cause-and-effect understanding of how to physically interact with the environment is learned. In particular the ability to predict the effects on the environment of specific motor actions is achieved. The second *symbolic-operational stage* (years 2-7) concerns the development of a symbolic understanding of the environment. The ability to communicate, eg. natural language, reading and writing, and to form internal representations of the external world is learned. In the *concrete-operational stage* (years 7-11) concepts and general principles which govern cause-and-effect relationships are attained. The *formal-operational stage* (years 11+) defines the procurement of the ability for full symbolic reasoning and conception opf possibilities beyond those present in reality. Central to this theory is that learning is best performed through physical manipulation of objects within the environment. However it provides very little insight into the specific mechanisms that underlie learning behaviour. Other similar theories have been proposed[64,115].

Accepting the model-based nature of behaviour and learning, autonomous learning techniques may be divided into three distinct categories[109] which are related to the above theories.

Parameter:

Parameter learning can be equated with the tuning of the parameters of a fixed structure internal world model by means of some simple learning mechanism based on experience.

Description:

Description learning could be described as the process of building new models to represent unique situations. The models would be constructed using *primitives* from a library of items and relationships.

Concept:

This is the highest learning form. When the available primitives are no longer suitable new ones must be created to allow efficient representations of the relevant concepts.

Engineering research into learning systems can be related to these categories. Adaptive control uses a parameter estimation technique to evaluate the variable parameters of an explicit model. Neural network learning schemes are also examples of parameter learning. Updating the weights of the network corresponds to estimating the parameters of an implicit model of the environment. Another example are game playing programs developed in AI which usually have a predefined tree structure which is filled in as the game progresses. Examples of approaches to the other two levels are highly specific AI programs[87,88]. Winston's 'concept' building program as discussed in chapter 1 is an example of description learning. AM and EURISKO, also described in chapter 1, implement a form of concept learning.

2.5 FRAMEWORK FOR INTELLIGENT CONTROL

A new definition of intelligent control was presented in section 2.2. This made explicit use of human systems as a reference model for design of intelligent control systems. Using this definition as a basis, aspects of intelligent human behaviour were examined. Several necessary attributes or characteristics were determined which can form the basis of a *framework* for intelligent control.

2.5.1 Necessary and Sufficient Conditions

The discussion above of psychological theories of intelligent behaviour has highlighted several attributes that underlie such behaviour. *Learning* is a crucial element. It is also highly complex and is a result of many internal mechanisms and characteristics. Some of the components that help to guide learning may be distinguished. One such characteristic is *prediction* or internal expectations. When the observed results of actions differ from expectations the internal mechanisms employed are updated to improve future predictions. Both cognitivist and behaviourist theories imply the use of some explicit or implicit *internal world model* to make logical inferences about the possible outcome of planned actions. Another important element is the active gathering of information using actions with dual purposes, i.e. both to achieve an objective and also to *probe* the environment to learn about it. The lowest form of learning (i.e. parameter) may be implemented by a mechanism to update the variable parameters of an internal world model.

From the discussion above it may be seen that learning ability is a prerequisite for intelligent behaviour. It is learning that distinguishes intelligent behaviour from more programmed responses. Hence learning may be stated as a ***sufficient*** condition for intelligent control.

Several mechanisms were identified as necessay for learning to occur efficiently. The interplay between learning and prediction is important although other attributes such as probing and internal prediction models also help to guide successful learning. ***Necessary*** conditions may thus be identified as: online prediction, internal world models for prediction, active probing of the environment, and simple parameter learning on the weights of the prediction model.

The combination of necessary and sufficient conditions defines a framework for intelligent control. Current research and control theory may be examined with respect to this. It provides a starting point for further development of a more rigorous framework, based on human reference systems, from which more design constraints may be determined.

2.5.2 Perspective on Adaptive Control

Current control techniques should be reviewed in the light of this framework. Feedforward strategies do not provide the opportunity to use prediction. The outcome of actions may not be observed to compare to predicted results. Thus feedforward cannot play a role in intelligent control. Feedforward compensation of measurable disturbances in combination with a feedback strategy may however be used. Feedback control may be classified as adaptive and non-adaptive. Non-adaptive strategies which assume constant or completely known plant models have no learning capabilities and hence are not intelligent.

Adaptive controllers as developed using control and estimation theory are one of the first approaches to the development of control algorithms with learning abilities. Incorporated in these controllers is a parameter or low-level learning capability. A fixed structure model is employed whose parameters are tuned (or learned) online based upon analysis of the input/output data. A learning requirement is essential at all levels of an intelligent control system, including the low-level loop. Thus adaptive control may be considered as the first steptowards intelligent control. But there are inadequacies with classical adaptive controllers[4,5,75]. They can suffer from the phenomenon of bursting or blow-up and a lack of persistent excitation. They have also been shown to fail to control more exotic systems with nonminimum-phase or very large deadtimes. These learning abilities must therefore be enhanced. Such an approach is proposed within the defined framework for intelligent control.

Intelligent control must also progress to incorporating the higher modes of learning. Design and commissioning phases would be significantly reduced if a general purpose learning controller could be placed on site to learn which model is most applicable and also tune that model for optimum performance (description and parameter learning). Such a controller would obviously require a library of possible models and hence would be restricted by the extent of this library. Designing a mechanism for concept learning into a controller would impart it with even greater autonomy and widen its range of applications.

2.6 LEARNING BASED PREDICTIVE CONTROL

A new approach to the design of a control algorithm suitable as the basis of an intelligent control system, using the proposed definition and framework is described here. This strategy will demonstrate improved performance over classical adaptive techniques.

There are many active elements involved in creating intelligent and learning behaviour. However, incorporating the four characteristics mentioned above, i.e. prediction, internal world model, probing, and parameter estimation, into a control technique would improve the robustness and performance of a low-level learning controller beyond that of classical adaptive control. Such a Learning Based Predictive Control (LBPC) approach could be used at the lower levels of a hierarchical intelligent control system.

In line with the proposed framework the desired characteristics of a control system, extracted using behavioural psychology, will be implemented using current control theory and techniques as a basis. A particular control design technique currently receiving much attention is long range predictive control. This technique has shown very good performance for control of difficult processes. It is based on predictions of the future behaviour of the plant due to proposed future input signals. This is achieved with the online use of an internal model of the plant. These strategies thus provide an excellent basis for LBPC as they employ similar principles to those which play a central role in intelligent and learned behaviour. Developed adaptive controllers and recursive identification methods provide several techniques for implementing parameter learning. These adaptation methods inherently incorporate use of an internal plant model. Current estimation techniques are based on plant input-output data. The actual control inputs are used by the learning process, hence introducing a form of sub-optimal probing. This would be acceptable as a first approach to the design of LBPC although dual control ideas could provide a method to make explicit use of criteria that consider the uncertainty of the plant and the need for cautionary inputs which have the dual role of controlling and probing.

2.7 CONCLUSIONS

A new definition of intelligent control was presented and a framework for design proposed. This makes explicit use of the link between intelligent control and intelligent human behaviour. It was suggested that examination of human behavioural processes could provide a unifying theory for current research and also suggest new areas and ideas for future development.

Some of the tenets central to intelligent and learned behaviour were investigated and described from a psychological viewpoint. Current intelligent control research was related to these characteristics and it was shown how they could be used at all levels of a hierarchical control system. Although the exact nature of intelligence remains unclear, the ability to learn is identified as a crucial element.

Learning occurs at all levels of human behaviour, from high level cognitive planning to low-level motor neuron responses. It also benefits from other important aspects identified as part of behaviour. Prediction is used to guide the learning process. At all levels, purposive behaviour estimates or predicts the expected outcome or results of planned actions. When these predictions do not match with actually observed responses, the method of prediction is changed to improve the accuracy of future predictions. To make predictions some form of internal world model is required. Use of internal models was also shown to be central to intelligent behaviour. Finally, it was noted that best learning performance is obtained by actual manipulations on the plant. That is, control inputs are used both for regulation and also for probing to help the learning process.

Necessary and sufficient conditions for intelligent control were identified. *Learning* is a sufficient condition and *prediction*, *probing*, *online models*, and *parameter learning* are necessary conditions. Using this framework and in line with the definition of section 2.2 a Learning Based Predictive Control strategy was proposed based on these elements.

In the following chapters the full development of an LBPC strategy is performed. The performance of this technique is demonstrated on academic plants and then for control of temperature and pressure at the die of a plasticating extruder. Extrusion processes are complex nonlinear distributed parameter systems which are difficult to accurately model. Such application areas require intelligent control and are suitable vehicles to demonstrate it's advantages. The sub-problem of extruder barrel wall temperature control is also considered. The solution proposed utilises several LBPC low-level loops with a second hierarchical level orchestrating the operation of the lower level. The second level is designed with decision logic to enable control of non-square multivariable systems with more outputs than inputs.

CHAPTER 3

LEARNING BASED PREDICTIVE CONTROL

3.1 INTRODUCTION

As part of the framework for the design of an intelligent controller it is proposed that existing long range predictive control (LRPC) strategies be considered. It is intended that a predictive strategy form the basis of controller design with enhancements made according to the framework of chapter 2.

In recent years LRPC methods have proven very successful in industrial applications[116,117,118]. They are very robust and reliable and provide many advantages above those of other control laws. Most LRPC methods can handle nonminimum phase processes, systems with large deadtimes, and can cope with uncertain or unknown delay times[116,117,119]. LRPC is particularly suitable as part of the proposed framework because it is based on a forecast of the future output from the process with the use of an internal model. There is a background of considerable practical and theoretical work on these techniques with several different strategies now available.

With the selection of LRPC two elements of LBPC design are achieved, namely the use of prediction and an internal model. The other elements may be incorporated by the inclusion of an adaptive layer around the LRPC method. Such a layer would form the first stage in the development of an advanced intelligent controller. Classical adaptive control algorithms lack robustness, depend on the choice of deadtime and model order and often go unstable controlling processes with nonminimum phase structures[45,120]. Adaptive predictive control demonstrates improved performance in these circumstances[]. The inclusion of a form of learning through the use of recursive least squares (RLS) meets the last two criteria of LBPC within the framework, i.e. learning and probing. The self-probing feature is automatically included through the use of RLS which utilises past measurements of controls and responses to estimate the parameters of the internal model, i.e. a form of parameter learning.

In this chapter, LRPC strategies are briefly reviewed and compared. One particular method, predictive functional control (PFC)[121], is selected as the basis of LBPC. PFC is reformulated in terms of an ARMAX model in order to make it more amenable to the inclusion of a RLS adaptive layer. The stability and robustness of the reformulated PFC method is considered and the design of an adaptive version is presented. Simulation results are included to demonstrate the performance of the algorithm in different circumstances.

3.2 LONG RANGE PREDICTIVE CONTROL

LRPC strategies are all based on similar principles[117,118]. At each sampling instant a forecast of the process output over a long-range time horizon (h sampling periods) is made. The forecast is made using a mathematical model of the process dynamics and is a function of the future control scenario that it is proposed to apply from the present instant to the end of the horizon, h. A reference trajectory is defined as the 'best way' for the process output to approach the setpoint or track the command signal. The control input is then calculated in order to make the predicted output follow the reference trajectory. Usually only the first computed control action, i.e. that for the present sampling instant, is applied to the process. At the next instant the whole procedure is repeated leading to an updated control action with corrections based on the latest measurements. This is called the *receding horizon principle.*

Several different LRPC strategies have appeared in the literature. Surveys and comparisons of these have also been compiled[117,118,122]. Differences between the algorithms are based on the following:

(1) type of internal process model
(2) structuration of future control scenario
(3) handling of noise and perturbations
(4) choice of tuning parameters

A brief review of some of the major strategies will be presented under headings based on the type of internal model employed.

3.2.1 Nonparametric Models

Two of the earlier methods, Model Algorithmic Control (MAC)[123,124] and Dynamic Matrix Control (DMC)[125] were developed around non-parametric internal models. MAC uses a convolution impulse response model while DMC employs step responses. MAC was the forerunner to the PFC algorithm discussed later.

Both strategies are only suitable for setpoint control with offset free performance. MAC defines the prediction error 'e' as:

$$e = y_r - y_p - G.\Delta u \qquad (3.1)$$

where, y_r is the reference trajectory output, y_p is the process output, Δu is the vector of future incremental controls, and G is a triangular matrix consisting of pulse response data,

$$G = \begin{bmatrix} g(1) & 0 & \dots & 0 \\ g(2) & g(1) & \dots & 0 \\ g(h) & g(h-1) & \dots & g(1) \end{bmatrix} \tag{3.2}$$

Minimisation of the cost functional:

$$J = e^T.e + Q.\Delta u^T.\Delta u \tag{3.3}$$

with Q a scalar weighting factor leads to the control law:

$$\Delta u = (G^T.G + QI)^{-1}.G^T.(y_r - y_p) \tag{3.4}$$

DMC is very similar to MAC and uses the prediction error:

$$e = y_r - y_p - H.\Delta u \tag{3.5}$$

where H is a triangular matrix of step response data. The cost function minimised is:

$$J = e^T \cdot e \tag{3.6}$$

leading to the control law:

$$\Delta u = (H^T \cdot H)^{-1}.H^T(y_r - y_p) \tag{3.7}$$

Both MAC and DMC have simple tuning parameters. With MAC the tuning variables are the length of the prediction horizon and the time response of the reference trajectory. Tuning of DMC is performed by varying the number of future input changes considered in minimizing the cost functional at each instant, i.e. it is assumed that the input is zero thereafter. The inclusion of an adaptive layer would require re-estimation of the impulse\step responses at each instant. Theoretical properties and robustness of nonparametric methods have been examined[126,127].

3.2.2. CARMA Models

The two major LRPC strategies formulated with *C*ontrolled *A*uto-*R*egressive *M*oving *A*verage (CARMA) models are:

° Extended predictive self-adaptive control (EPSAC)[128].
° Extended horizon adaptive control (EHAC)[129].

The CARMA model has the form:

$$A(z^{-1}).Y(z) = z^{-d}.B(z^{-1}).U(z) + C(z^{-1}).V(z) \quad (3.8)$$

where A, B, and C are polynomials in the complex operator of the z-transformation. Y(z) is the process output, U(z) the input, and V(z) is an uncorrelated stochastic noise signal (usually assumed white Gaussion noise with zero mean and standard deviation σ^2). The process deadtime is 'd'. Noise and disturbances are explicitly modelled through the last term in equation 3.8.

Each of these methods uses the Diophantine identity

$$C(z^{-1}) = E(z^{-1}).A(z^{-1}) + z^{-j}.F(z^{-1}) \quad (3.9)$$

with,
$$E(z^{-1}) = 1 + e_1.z^{-1} + \ldots + e_{j-1}.z^{-j+1}$$
$$F(z^{-1}) = f_0 + f_1.z^{-1} + \ldots + f_{j-1}.z^{-j+1}$$

to simplify the prediction of future process outputs. The CARMA equation (eq. 3.8) becomes:

$$Y(z).z^{i} = z^{i-d}.E_i(z^{-1}).B(z^{-1}).U(z) + F_i(z^{-1}).Y(z) \quad (3.10)$$

To calculate the predicted output at 'i', $\hat{y}(t+i)$, the Diophantine equation has to be solved for the polynomials $E_i(z^{-1})$ and $F_i(z^{-1})$ [130].

The EHAC technique is defined as computing a sequence of input steps u(t+i) such that the distance between the predicted output and the reference trajectory output at the end of the prediction horizon (h) is zero. Minimizing the cost criterion:

$$J = \sum_{i=0}^{h-1} u^2(t+i) \quad (3.11)$$

leads to the solution

$$u(t) = \beta_k(\sum_{j=1}^{h} \beta^2{}_j)^{-1}[y_r(t+h) - \sum_{i=1}^{n} \alpha_i.y(t+1-i) - \sum_{i=1}^{m} \beta_{h+i}.u(t-i)] \quad (3.12)$$

There is only one tuning parameter, h. Obviously, increases in h lead to longer computation times and hence slower control.

EPSAC is similar to EHAC except that a multistep cost function is used, i.e.

$$J = \sum_{i=d}^{h} [y(t+i) - y_r(t+i)]^2 \tag{3.13}$$

The predicted output is assumed to consist of two parts. One part is the output due to a constant step input (i.e. u(t) = u(t+i), 0<i<h-d), and the other contains the effect of stepwise variagated $\Delta u(t)$. The control law becomes:

$$\Delta u(t) = \sum_{i=d}^{h} \alpha_i [y_r(t+i) - y(t+i)] \tag{3.14}$$

where α_i are weighing terms, and also the only effective tuning parameters available. Usually, a simple choice is to use an exponential weighing function

$$\alpha_i = \lambda^{h-i} \tag{3.15}$$

where λ is a design parameter with influence over the control effort to be exerted. Major disadvantages with EPSAC are that the deadtime is assumed known (see eq. 3.14), and also the large number of computations required to estimate 'h-d' future predicted process outputs. A suboptional prediction routine has been proposed to try to offset this problem[131].

Other predictive controller strategies based on a CARMA model include: Adaptive Control System (APCS)[132,133], MUSMAR[134], and the Matrix Factorisation Algorithm[135]. The major problems with each of these and also the previous two techniques is the large number of computations either through multiple future predictions or solutions of the Diophantine equation, or both.

3.2.3 CARIMA Model

The CARIMA, or Controlled Auto-Regressive Integrated Moving Average, model is based on integral action and automatically ensures that derived control laws have inherent integral performance resulting in zero-offset control:

$$A(z^{-1}).\Delta Y(z) = z^{-d}.B(z^{-1}).\Delta U(z) + C(z^{-1}).V(z) \tag{3.16}$$

with $\Delta = 1 - z^{-1}$.

The main predictive technique to use this model is Generalised Predictive Control (GPC)[136-138,119]. The C polynomial is neglected and the future output is predicted using a modified Diophantine identity:

$$1 = E_i(z^{-1}).A(z^{-1}).\Delta + z^{-i}.F_i(z^{-1}) \tag{3.17}$$

Hence the future output is predicted by:

$$Y(z).z^i = E_i(z^{-1}).B(z^{-1}).\Delta U(z^{i-d}) + F_i(z^{-1}).Y(z) + E_i(z^{-1}).V(z).z^i \tag{3.18}$$

and the output may be written in a time step form as:

$$y(t+i) = \sum_{j=0}^{m} \beta(j).\Delta u(t+i-d-j) + \sum_{j=0}^{n} f(j).y(t-j) \tag{3.19}$$

with v(t+i)=0. Using eq. 3.19 on the prediction horizon **h**, **h** equations may be derived and written in matrix form:

$$y(h) = G.\Delta u(t+h-1) + f \tag{3.20}$$

where $y(t) = [y(t+1) \quad y(t+2) \quad \ldots \quad y(t+h)]^T$

$$G = \begin{bmatrix} \beta(1) & 0 & \ldots & 0 \\ \beta(2) & \beta(1) & \ldots & 0 \\ \beta(h) & \beta(h-1) & \ldots & \beta(1) \end{bmatrix}$$

and f collects all known signals at time t modified by the known parameters β(i), e(i), and f(i).

Using (3.20) with the matrix cost function:

$$J = (y-y_r)^T.(y-y_r) + Q.\Delta u^T.\Delta u \tag{3.21}$$

and minimising results in the control law:

$$\Delta u(t+h-1) = [G^TG + QI]^{-1}.G^T[y_r(t+h) - f] \tag{3.22}$$

This algorithm may deal with nonminimum phase processes (with Q>0) and also unknown or variable deadtime[119]. Criticisms of this method again refer to the number of computations and online solution of the Diophantine equation, although a recursive method of doing this is available[136]. Tuning of the algorithm is also non-trivial.

Another 'argument' against GPC is that it is simply another form of linear quadratic Gaussian (LQG) control[139]. In fact, a relationship between GPC, LQ, and minimum variance has been found[137].

3.2.4 State-Space Models

The only LRPC technique based on state-space modelling currently known to the author is Predictive Functional Control (PFC)[121,140-142]. PFC is designed specifically for both demand following and trajectory tracking. It is characterised mainly by the future control policy and by the objective control function. The future control input, or manipulated variable (MV), is structured as a linear combination of a finite pre-specified set of base functions, eg. step, ramp, etc. The choice of base functions is a design parameter and is dependant on the nature of the process and the maximum degree of the command trajectory polynomial.

Future predicted outputs are obtained from a state-space model:

$$\begin{aligned} X_m(n) &= A_m.X_m(n-1) + B_m.u(n-1) \\ Y_m(n) &= C_m.X_m(n) \end{aligned} \tag{3.23}$$

This is simplified by considering separately the homogenous response and the nonhomogenous response due to the future control senario.

The control is calculated by minimising a quadratic distance between the predicted process output and the reference trajectory for each base function at the end of a separately defined prediction horizon for each, i.e.

$$J = \sum_{j=1}^{nh} [y_m(n+h_j) - y_r(n+h_j)]^2 \tag{3.24}$$

where 'nh' is the number of base functions selected. This leads to a linear regulator of the form:

$$u(n) = \sum_{j=0}^{nc} [k_j . c_j(n)] - y(n) + \underline{V}_x^T . X_m(n) \tag{3.25}$$

where 'nc' is the degree of the command trajectory polynomial, and k_j and $\underline{V}_x$ are calculated weighting values. The full PFC derivation is presented below.

Process model mismatch and disturbances which lead to static errors are compensated by a '*self-compensator*'. This consists of a filter which operates on a window of past error measurements and an extrapolator which predicts the future error over the prediction horizon. This compensation technique alters the form of eq. 3.25 slightly as will be shown later.

The advantages of PFC are that it is easily designed for trajectory tracking control with no lag error, unlike some other LRPC strategies. A simple equivalent linear controller may be derived which is easily implemented with few online calculations. The design procedure is also CAD compatible[] which is a desirable feature as a basis for intelligent control design. PFC tuning parameters are easily understandable and relate well to time-domain characteristics of the closed-loop response. The tuning parameters are the time response of the reference trajectory (corresponding to the desired closed-loop time response) and also the selection of the coincidence points for each base function. Short coincidence points lead to faster but harder control effort and vice versa for longer coincidence horizons. The PFC algorithm also gives high quality performance and robustness compared to other methods, as will be shown later. For these reasons, it was decided that the principles of PFC would be used as the basis for the design of LBPC within the intelligent control framework.

3.3 PREDICTIVE FUNCTIONAL CONTROL

The principles of PFC mentioned in section 3.2.4 will be expanded here. The PFC strategy will then be reformulated in terms of a ARMAX model and the stability\robustness of the algorithm examined.

3.3.1 PFC Principles

The PFC algorithm calculates online the control variable, or MV, according to the receding horizon strategy discussed in section 3.2, and as shown in fig. 3.1.

At each sampling instant the following operations are performed:

- The command trajectory in the future is determined, either pre-specified or extrapolated.

- A reference trajectory, usually a first order decay error, is initialised on the actual measured process output. This defines the way the output should appraoch the command trajectory over a prediction horizon.

- The future control variable is structured as a linear combination of a pre-specified set of base functions.

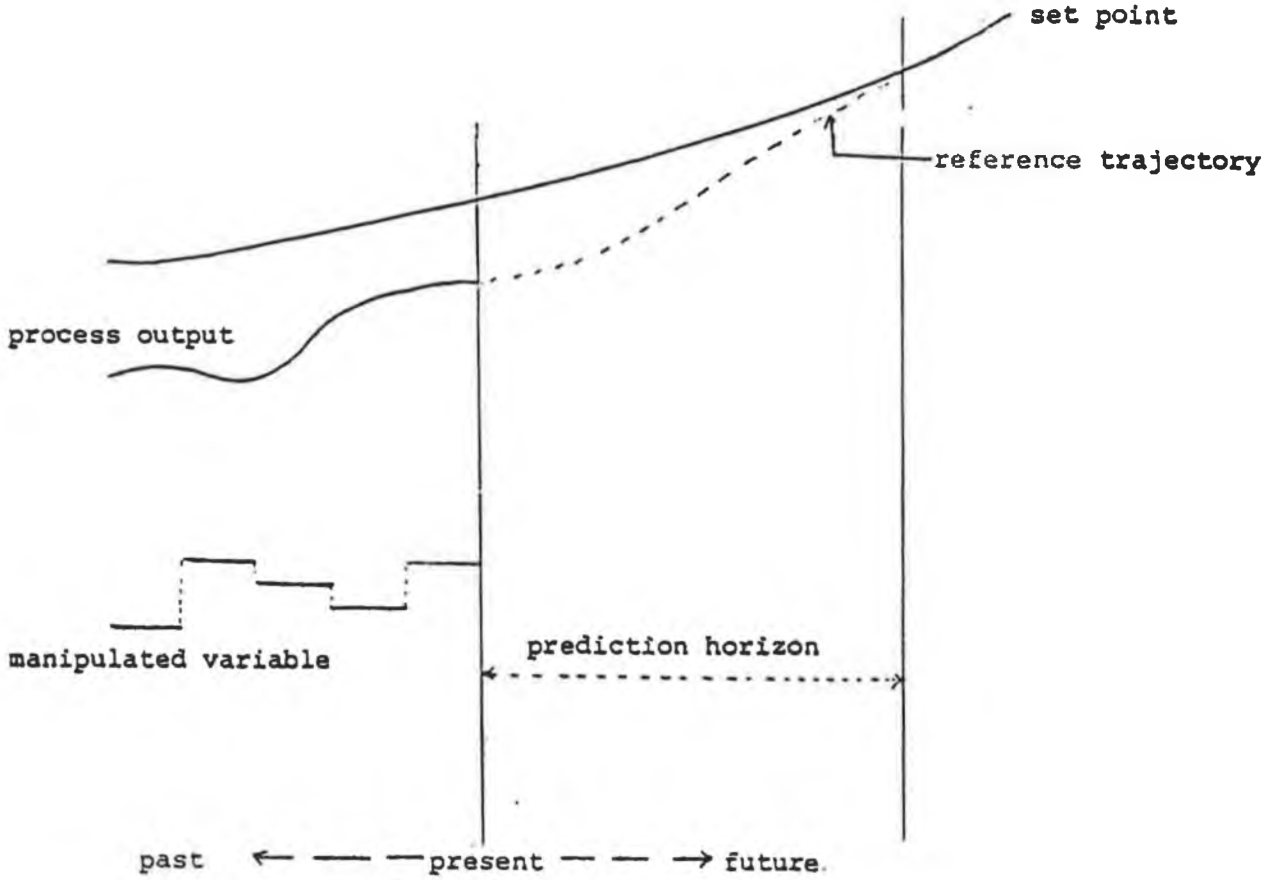

FIG. 3.1: Principles of PFC

- The process model allows the future output to be predicted under the effect of the future controls.

- The control objective is to minimise the sum of the squared errors between the predicted output and the reference trajectory at the coincidence points. The minimum number of these points is at least equal to that of the base functions and their choice influences the stability and robustness aspects of the algorithm.

- The control variable computation consists of determining, through minimisation of the control objective by the *least squares* method, the unknown coefficients of the linear combination of the control expression.

Only the first value of the computed control sequence is executed. The whole procedure is repeated at the next sampling instant, and so on. Extension of PFC to the multiple-input multiple-output (MIMO) case is straightforward. A reference trajectory is defined for each output. The control variables are structured separately and the control objectives (as above) corresponding to the different outputs are minimised.

3.3.2 PFC Derivation with ARMAX Model

Although the PFC algorithm was developed with state-space models (see section 3.2.4), the use of PFC principles are not exclusively reserved for such models. It would be preferable to reformulate PFC in terms of an ARMAX model and thus make it more amenable to the addition of an adaptive layer. Use of an ARMAX model also eliminates the need for a state observer if the model is to be continuously realigned. With the original PFC algorithm the model may either exist independently of the process (i.e. only initial states are matched) or else be realigned using a state observer to estimate $\underline{X}(n)$ at each instant.

The derivation of PFC with an ARMAX model and an adaptive RLS layer is an approach to the design of a learning based predictive controller.

Consider first the derivation for a step setpoint, i.e. a demand following regulator. The future setpoint will remain constant i.e.

$$c(n+i) = c(n) \tag{3.26}$$

To track a step setpoint, given a non-integrative process, will only require a step base function. Thus only one coincidence point on the prediction horizon is chosen for this base function. The control input is therefore

$$u(n) = \mu . u_B(0)$$

where, $u_B(i)$ = unit step base function
μ = scaling factor

The unknown μ must be calculated. The control objective is specified to minimise a quadratic distance between the predicted process output and the reference trajectory at this coincidence point (h):

$$C = [s_P(n+h) - s_R(n+h)]^2 \tag{3.27}$$

where, s_P = process output
s_R = reference trajectory

The process output is predicted using the internal model, plus a predicted error term obtained from a filter and extrapolator as explained in section 3.2.4.

$$s_P(n+h) = s_M(n+h) + \hat{e}(n+h) \tag{3.28}$$

where, s_M = model output
$\hat{e}$ = predicted error

Consider the predicted model output. This may be separated into two parts, a homogenous part and a nonhomogenous part. The homogenous output (s_{hom}) is the output assuming zero future inputs. The nonhomogenous part of the output is that due to the application of the base function, $\mu . u_B(i)$. This output is simply the output due to $u_B(i)$ (i.e. $s_o(i)$) scaled by μ assuming a linear system. Hence, eq. 3.28 may be rewritten as

$$s_P(n+h) = \mu . s_o(h) + s_{hom}(n+h) + \hat{e}(n+h) \tag{3.29}$$

A first order reference trajectory, initialised on the process output (i.e. $s_R(n) = s_P(n)$) is used:

$$c(n+i) - s_R(n+i) = \alpha^i [c(n)-s_P(n)] \tag{3.30}$$

where $\alpha = e^{-3T/\tau}$
T = sampling period
τ = desired closed loop time response

As $c(n+i) = c(n)$, eq. 3.30 may be rearranged and evaluated at h, i.e.

$$s_R(n+h) - s_P(n) = (1-\alpha^h)[c(n)-s_P(n)] \qquad (3.31)$$

Substituting eq.'s 3.28 and 3.29 into 3.27 leads to the control objective:

$$C = [\mu.s_o(h) + s_{hom}(n+h) + \hat{e}(n+h) - s_R(n+h)]^2 \qquad (3.32)$$

Let,

$$\delta(n+h) = s_R(n+h) - s_{hom}(n+h) - \hat{e}(n+h) \qquad (3.33)$$

$$\Rightarrow C = [\mu.s_o(h) - \delta(n+h)]^2$$

It is required to minimise C with respect to the unknown μ, i.e.

$$\frac{dC}{d\mu} = 0 \qquad (3.34)$$

The solution of (3.34) gives

$$\mu = \frac{\delta(n+h)}{s_o(h)} \qquad (3.35)$$

Now, as $u_B(i)$ is the unit step base function,

$$u_B(0) = u_B(1) = \ \dots \ = 1 \qquad (3.36)$$

$$\Rightarrow \quad u(n) = \mu.u_B(0) = \mu \qquad (3.37)$$

$$\Rightarrow \quad u(n) = \frac{\delta(n+h)}{s_o(h)} \qquad (3.38)$$

$s_o(h)$ is precalculated as the forced output from the model at h with the base function applied assuming zero initial conditions. $\delta(n+h)$ must be expanded. From (3.33),

$$\delta(n+h) = s_R(n+h) - s_{hom}(n+h) - \hat{e}(n+h) \qquad (3.39)$$

To determine a term for $\hat{e}(n+h)$ consider the simplest case of an extrapolator of degree zero. This assumes that the future predicted error is equal to the present measured error, i.e.

$$\hat{e}(n+h) = \hat{e}(n+h-1) = \ \dots \ = \hat{e}(n+1) = e(n) \qquad (3.40)$$

and

$$e(n) = s_P(n) - s_M(n) \qquad (3.41)$$

Substituting (3.40) and (3.41) into (3.39)

$$\delta(n+h) = [s_R(n+h)-s_P(n)] - [s_{hom}(n+h)-s_M(n)] \qquad (3.42)$$

Using (3.31) to replace the first term on the R.H.S. above:

$$\delta(n+h) = (1-\alpha^h)[c(n)-s_P(n)] - [s_{hom}(n+h)-s_M(n)] \qquad (3.43)$$

The last term on the R.H.S. consists of model outputs and so may be expressed in terms of the model parameters.

The internal model to be employed is the deterministic form of the ARMAX model (i.e. DARMA). The stochastic part is accounted for by the *self-compensator.*

$$[1-A(z^{-1})].Y(z) = B(z^{-1}).U(z) \qquad (3.44)$$

or,

$$[1-A(z^{-1})].s_M(n) = B(z^{-1}).u(n) \qquad (3.45)$$

with, $A(z^{-1}) = a_1.z^{-1} + a_2.z^{-2} + ... + a_{na}.z^{-na}$

$$B(z^{-1}) = b_1.z^{-1} + b_2.z^{-2} + ... + b_{nb}.z^{-nb}$$

Hence, in time-step form:

$$s_M(n) = a_1.s_M(n-1) + a_2.s_M(n-2) + ... + a_{na}.s_M(n-na)$$

$$+ b_1.u(n-1) + b_2.u(n-2) + ... + b_{nb}.u(n-nb) \qquad (3.46)$$

$s_{hom}(n+h)$ is determined recursively from (3.46) with $u(n) = u(n+1) = ... = u(n+h-1) = 0$, i.e.

$$s_{hom}(n+h) = \alpha_1 s_M(n) + \alpha_2 s_M(n-1) + ... + \alpha_{na} s_M(n-na+1)$$

$$+ \beta_1.u(n-1) + \beta_2.u(n-2) + ... + \beta_{nb-1}.u(n-nb+1)$$

(3.47)

where, $\alpha_i(h) = a_i.\alpha_i(h-1) + \alpha_{i+1}(h-1)$ (3.47a)

$\beta_j(h) = b_j.\alpha_1(h-1) + \beta_j(h-1)$ (3.47b) and,

$$\alpha_i(0) = a_i \ , \ \beta_j(0) = b_j$$

for $i = 1, 2, \ldots, na$; $j = 1, 2, \ldots, nb\text{-}1$

Thus, the last term in (3.43) may be rewritten as:

$$s_{hom}(n+h) - s_M(n) = \underline{\theta}^T.\underline{\varphi}(n) \tag{3.48}$$

where,

$$\underline{\theta}^T = [\alpha_1\text{-}1 \quad \alpha_2 \quad \ldots \quad \alpha_{na} \quad \beta_1 \quad \beta_2 \quad \ldots \quad \beta_{nb\text{-}1}]$$

$$\underline{\varphi}(n) = [s_M(n) \quad s_M(n\text{-}1) \quad \ldots \quad s_M(n\text{-}na\text{+}1) \quad u(n\text{-}1) \quad \ldots \quad u(n\text{-}nb\text{+}1)]$$

Using (3.48) in (3.43) gives

$$\delta(n+h) = (1-\alpha^h).[c(n)-s_P(n)] - \underline{\theta}^T.\underline{\varphi}(n) \tag{3.49}$$

Hence, the PFC regulator becomes

$$u(n) = k_0[c(n)-s_P(n)] - \underline{v}^T.\underline{\varphi}(n) \tag{3.50}$$

where,

$$k_0 = \frac{1}{s_o(h)} \cdot (1-\alpha^h)$$
$$\underline{v} = \frac{1}{s_o(h)} \cdot \underline{\theta}$$

The regulator of (3.50) uses an independent model, i.e. $\underline{\varphi}(n)$ is in terms of past models outputs, not the measured process outputs. A 'realigned' regulator may be obtained by using the past measured process outputs, i.e.

$$\underline{\varphi}(n) = [s_P(n) \quad s_P(n\text{-}1) \quad \ldots \quad s_P(n\text{-}na\text{+}1) \quad u(n\text{-}1) \quad u(n\text{-}2) \quad \ldots \quad u(n\text{-}nb\text{+}1)] \tag{3.51}$$

3.3.3 Stability and Robustness Issues

The stability of the algorithm may be examined by looking at how the error $c(n) - s_P(n)$ progresses as $n \to \infty$.

Consider the regulator equation of (3.50). This may be rewritten as:

$$s_0(h).u(n) = k.\epsilon(n) - \underline{\Theta}^T.\underline{\varphi}(n) \qquad (3.52)$$

where, $k = s_0(h).k_0$
$\epsilon(n)$ = error, i.e. $c(n)-s_P(n)$
$\underline{\Theta}^T$ as before

This may be rewritten in polynomial form:

$$s_0.u(n) = k.\epsilon(n) - \gamma(z^{-1}).s_M(n) - \eta(z^{-1}).u(n-1) \qquad (3.53)$$

Multiply by $B(z^{-1})$, and using the DARMA model (eq. 3.45)

$$\Rightarrow s_0.[1-A(z^{-1})].s_M(n) = k.B(z^{-1}).\epsilon(n) - \gamma(z^{-1}).B(z^{-1}).s_M(n) - \eta(z^{-1}).[1-A(z^{-1})].s_M(n-1) \qquad (3.54)$$

When disturbances are present or process model mismatch exists, $s_P(n) \neq s_M(n)$, an error term must be included as in (3.28), i.e.

$$s_P(n) = s_M(n) + e(n) \qquad (3.55)$$

Hence, (3.54) may be rewritten using (3.55). The z^{-1} dependence terms have been neglected for ease of writing:

$$s_0.(1-A).[s_P(n)-e(n)] = k.B.\epsilon(n) - \gamma.B.[s_P(n)-e(n)] - \eta.(1-A).[s_P(n-1)-e(n-1)] \qquad (3.56)$$

The progression of $\epsilon(n)$ as $n \to \infty$ is desired, so the following relationship:

$$s_P(n) = c(n) - \epsilon(n) \qquad (3.57)$$

is substituted into (3.56) to give

$$Q(z^{-1}).\epsilon(n) = R(z^{-1}).[c(n)-e(n)] \qquad (3.58)$$

where,

$$R(z^{-1}) = s_0.[1-A(z^{-1})] + \gamma(z^{-1}).B(z^{-1}) + z^{-1}.\eta(z^{-1}).[1-A(z^{-1})]$$

$$Q(z^{-1}) = R(z^{-1}) + k.B(z^{-1})$$

and $R(z^{-1})$, $Q(z^{-1})$ are both of degree na+nb.

It may be shown[121] that

$$s_0.[1 - \sum_{i=0}^{na} a_i] + \sum_{i=0}^{na} \alpha_i . \sum_{i=0}^{nb} b_i . + \sum_{i=0}^{nb-1} \beta_i . [1 - \sum_{i=0}^{na} a_i] = 0 \qquad (3.59)$$

Thus, if c(n) is a constant setpoint, then

$$R(z^{-1}).c(n) = 0 \qquad (3.60)$$

Similarly, if the error e(n) is also constant, then

$$R(z^{-1}).e(n) = 0 \qquad (3.61)$$

In this case,

$$Q(z^{-1}).\epsilon(n) = 0 \qquad (3.62)$$

Assuming that $Q(z^{-1})$ is stable (i.e. all roots inside the unit circle, which is achieved by proper selection of the coincidence points) and that $\epsilon(0)$ is bounded, then

$$\lim_{n \to \infty} \epsilon(n) = 0 \qquad (3.63)$$

Hence, zero offset regulation will be obtained for setpoint control using one base function (a step), even considering step disturbances applied to the process. In this instance there is no need for a self-compensator.

If e(n) is a ramp it may be shown[121] that $R(z^{-1}).e(n)$ is a constant value, and hence that

$$\lim_{n \to \infty} \epsilon(n) = \text{constant} \qquad (3.63a)$$

Similarly, only using one base function for a ramp 'setpoint' may be shown to produce an offset (i.e. constant lag error). If the degree of the base functions is chosen equal to that of the setpoint polynomial (assuming a non-integrative plant) then the corresponding $R(z^{-1}).c(n)$ may be shown to equal zero in an analagous fashion to the above case. Thus, zero lag error tracking will be achieved.

In this case, the regulator algorithm of (3.50) is modified by an additional term to account for the future setpoint, as shown in section 3.3.4 below. The stability analysis is performed as before leading to a modified (3.58):

$$Q(z^{-1}).\epsilon(n) = S(z^{-1}).c(n) - R(z^{-1}).e(n) \qquad (3.64)$$

where $Q(z^{-1})$ and $R(z^{-1})$ are as before. $S(z^{-1})$ is also as previously derived but with an additional term representing the future setpoint deviation. As stated above, it may be shown[121] that $S(z^{-1})c(n) \to 0$ producing zero lag error.

The effect of a disturbance or process\model mismatch may also be examined from (3.64). If e(n) is constant then $R(z^{-1}).e(n) \to 0$ as before. Hence no self-compensator is required. Otherwise the term $R(z^{-1}).e(n) \neq 0$ and offset lag will be produced. In general, the trajectory of $R(z^{-1}).e(n)$ will be of degree one less than the degree of polynomial e(n)[121]. A self-compensator is required therefore whenever non-step disturbances are expected.

3.3.4 General PFC Regulator with ARMAX Model

The command trajectory is developed as a polynomial of degree nc:

$$c(n+i) = \sum_{j=0}^{nc} c_j(n).i^j \qquad (3.65)$$

The future MV is structured as a linear combination of nb base functions:

$$u(n+i) = \sum_{j=1}^{nb} \mu_j(n).u_{Bj}(i) \qquad (3.66)$$

The control objective is to minimise a quadratic distance between the predicted process output and the reference trajectory at each of the coincidence points (h_j):

$$C = \sum_{j=1}^{nb} [\hat{s}_P(n+hj) - s_R(n+hj)]^2 \qquad (3.67)$$

The future process output may be written as:

$$\hat{s}_p(n+i) = s_M(n+i) + \hat{e}(n+i) \qquad (3.68)$$

The model output consists of a homogenous and a forced part:

$$s_M(n+i) = s_{hom}(n+i) + s_{forced}(n+i) \qquad (3.69)$$

$$s_{forced}(n+i) = \sum_{j=1}^{nb} \mu_j(n).s_{bj}(i) \qquad (3.70)$$

where s_{bj} are precomputed model outputs due to applied u_{bj} base functions. Equation (3.70) is based on a linear system model.

A first-order reference trajectory as in (3.30) is assumed:

$$c(n+i) - s_R(n+i) = \alpha^i.[c(n)-s_P(n)] \qquad (3.71)$$

Rearranging and evaluating at h_j

$$s_R(n+h_j) - s_P(n) = (1-\alpha^{h_j}).[c(n)-s_P(n)] + c(n+h_j) - c(n) \qquad (3.72)$$

Using (3.68), (3.69), and (3.70) in (3.67) gives:

$$C = \sum_{j=1}^{nh} [\sum_{k=1}^{nb} \mu_k(n).s_{bk}(h_j) - \delta(n+h_j)]^2 \qquad (3.73)$$

where,

$$\delta(n+h_j) = s_R(n+h_j) - s_{hom}(n+h_j) - \hat{e}(n+h_j)$$

Minimising criterion C using least squares w.r.t. the unknown μ_k:

$$\underline{\mu}(n) = [\sum_{j=1}^{nh} \underline{s}_b(h_j).\underline{s}_b(h_j)^T]^{-1}.[\sum_{j=1}^{nh} \delta(n+h_j).\underline{s}_b(h_j)] \qquad (3.74)$$

where,

$$\underline{\mu}(n)^T = [\mu_1(n) \quad \mu_2(n) \quad ... \quad \mu_{nb}(n)]$$

$$\underline{s}_b(h_j)^T = [s_{b,1}(h_j) \quad s_{b,2}(h_j) \quad ... \quad s_{b,nb}(h_j)]$$

The last term in (3.74) may be written as $\underline{s}_B.\underline{\delta}(n)$, where $\underline{s}_B$ is a nb x nh matrix and

$$\underline{\delta}(n)^T = [\delta(n+h_1) \quad \delta(n+h_2) \quad ... \quad \delta(n+h_{nh})]$$

Hence,

$$\underline{\mu}(n) = M.\underline{\delta}(n) \qquad (3.75)$$

where M is derived from (3.74).

Now, the applied control is

$$u(n) = \sum_{j=1}^{nb} \mu_j(n).u_{bj}(0) = \underline{\mu}(n)^T.\underline{u}_B(0) \tag{3.76}$$

Using (3.75):

$$u(n) = [M.\underline{\delta}(n)]^T.\underline{u}_B(0) \tag{3.77}$$

$$\Rightarrow \quad u(n) = [M^T.\underline{u}_B(0)]^T.\underline{\delta}(n) = \underline{v}^T.\underline{\delta}(n) \tag{3.78}$$

Now consider the elements of $\underline{\delta}(n)$:

$$\delta(n+h_j) = s_R(n+h_j) - s_{hom}(n+h_j) - \hat{e}(n+h_j) \tag{3.79}$$

The predicted error is obtained by filtering a finite window of observed past errors (i.e. $s_P(n-i)-s_M(n-i)$) and extrapolating into the future[121], i.e.

$$\hat{e}(n+i) = s_P(n) - s_M(n) + \sum_{j=1}^{ne} f_j(n).i^j \tag{3.80}$$

where, ne = degree of extrapolator

f_j = extrapolator parameters derived as in [121]

Using a DARMA model

$$s_{hom}(n+h_j) - s_M(n) = \underline{\Theta}^T.\underline{\varphi}(n) \tag{3.81}$$

where $\underline{\Theta}^T$ and $\underline{\varphi}(n)$ are as defined for (3.48).

Thus, using (3.72), (3.80) and (3.81) in (3.79):

$$\delta(n+h_j) = (1-\alpha^{h_j})[c(n)-s_P(n)] + c(n+h_j) - c(n) - \underline{\Theta}^T.\underline{\varphi}(n) - \sum_{j=1}^{ne} f_j(n).i^j \tag{3.82}$$

Expanding $c(n+h_j)$ as a polynomial:

$$c(n+h_j) = c(n) + c_1(n).h_j + \ldots + c_{nc}(n).h_j^{nc} \tag{3.83}$$

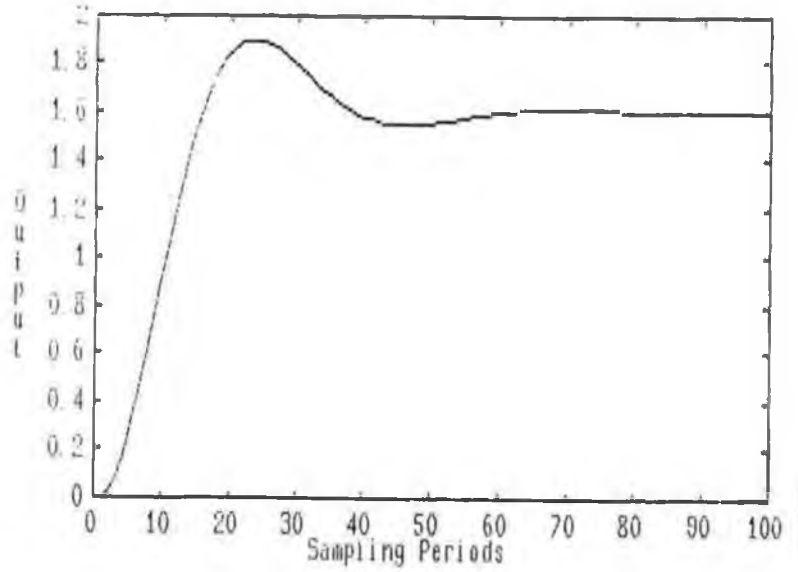

FIG. 3.2: Open Loop Step Response - Second Order Plant

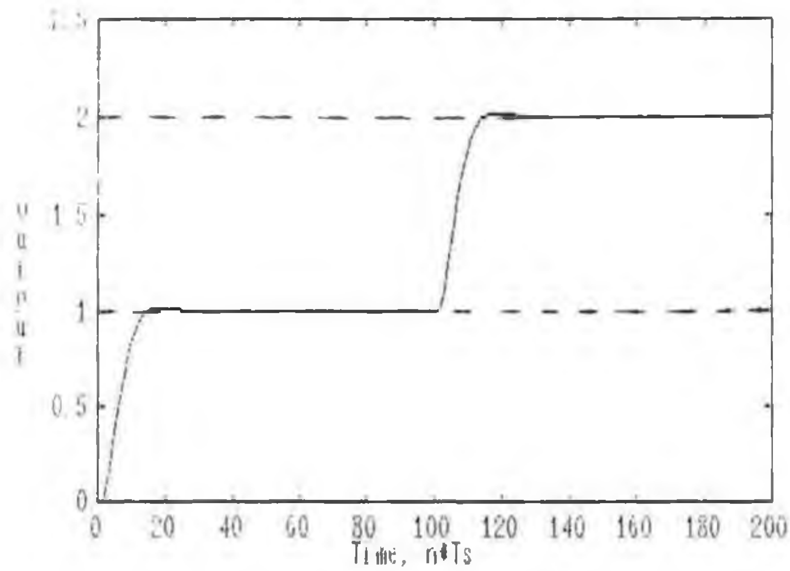

FIG. 3.3(a): Setpoint Tracking

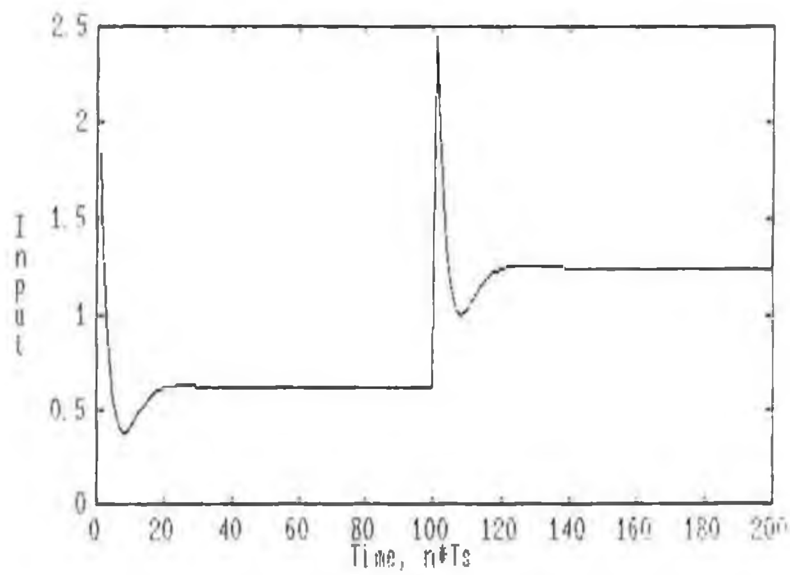

FIG. 3.3(b): Control Input

Defining the following scalars:

$$k_0 = \underline{v}^T . \begin{bmatrix} 1-\alpha^{hj} \\ \vdots \end{bmatrix}, \qquad k_i = \underline{v}^T . \begin{bmatrix} h^i{}_j \\ \vdots \end{bmatrix} \text{ with } i=1,2,\ldots,\max(nc,ne)$$

and the vector:

$$\underline{v}_x = - \begin{bmatrix} \vdots \\ \underline{\theta}(h_j).\underline{v} \\ \vdots \end{bmatrix}$$

allows the linear PFC regulator with ARMAX internal model and self-compensator to be written as:

$$u(n) = k_0.[c(n)-s_p(n)] + \sum_{j=1}^{\max(nc,ne)} k_j.[c_j(n)-f_j(n)] + \underline{v}_x^T.\underline{\varphi}(n) \qquad (3.84)$$

As previously explained, two possibilities exist: independent or realigned model. If an independent model is used it may be shown[] that no static error is obtained but that the poles of the model are cancelled. This effects the robustness of control of unstable processes. Continuously realigning the model on the process will generally induce a static error[121] but unstable plants may be stabilised. The static errors are easily removed by the use of self-compensator.

3.3.5 Simulation Results

The performance of the algorithm was tested by simulations on academic examples. A regulator to control the following underdamped second-order deterministic model was first designed:

$$(1 - 1.837541.z^{-1} + 0.860708.z^{-2}).y(t) = (0.01899718 + 0.0181.z^{-1}).u(t-1)$$

The open-loop step response of this plant is shown in fig. 3.2. Using design parameters of h=0.7 sec. and TR=0.8 sec., a regulator was design for setpoint control giving the control law parameters (sample time = 0.1 sec.):

$$k_0 = 1.825238 \qquad \underline{v}_x^T = [5.514125 \quad -6.220309 \quad 0.130808]$$

A plot of the controlled output is shown in fig. 3.3(a). The setpoint is set to unity initially and then increased to two at t=10 sec. (100 samples). Fig. 3.3(b) shows the

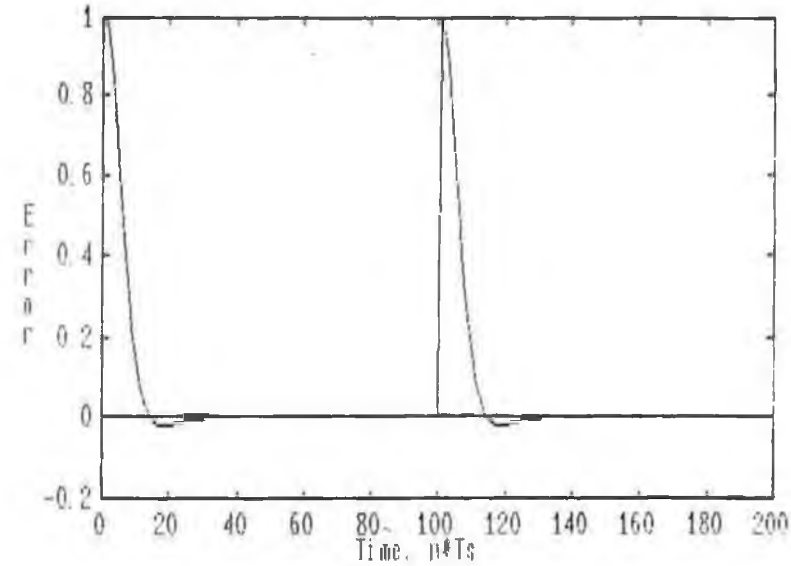

FIG. 3.3(c): Control Output Error

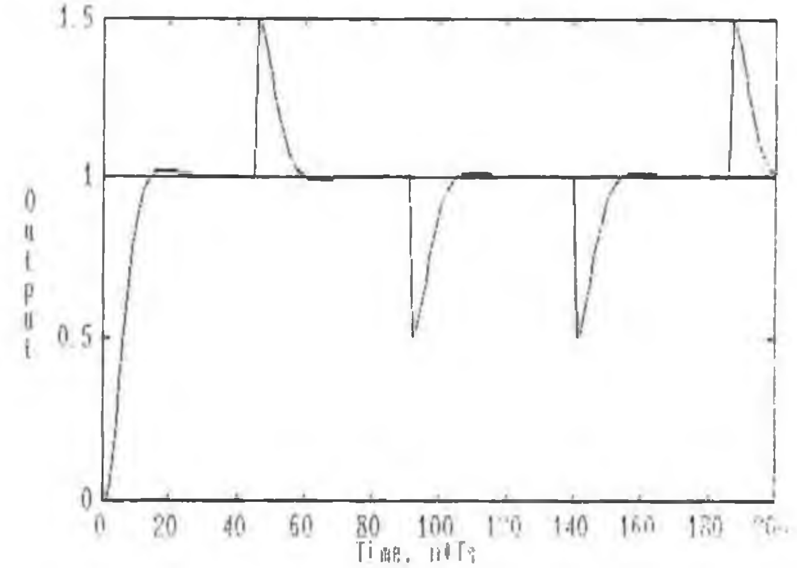

FIG. 3.4(a): Response to Step Disturbances

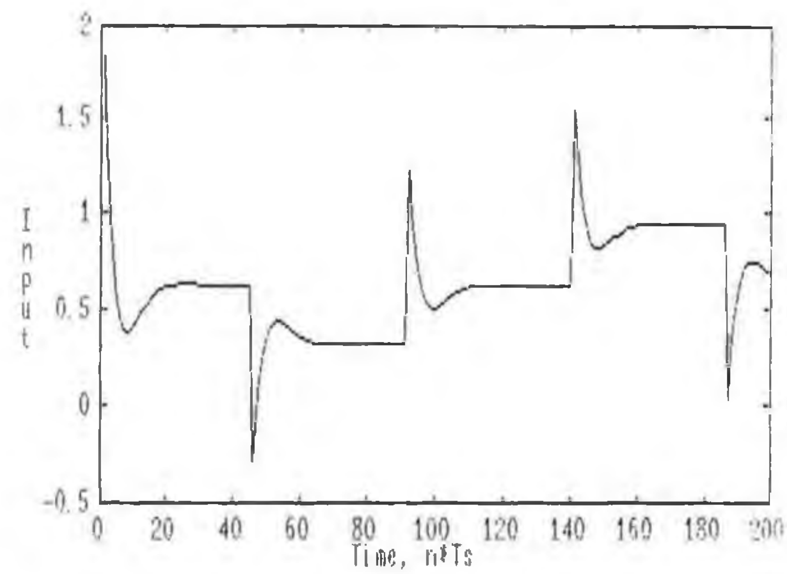

FIG. 3.4(b): Control Input

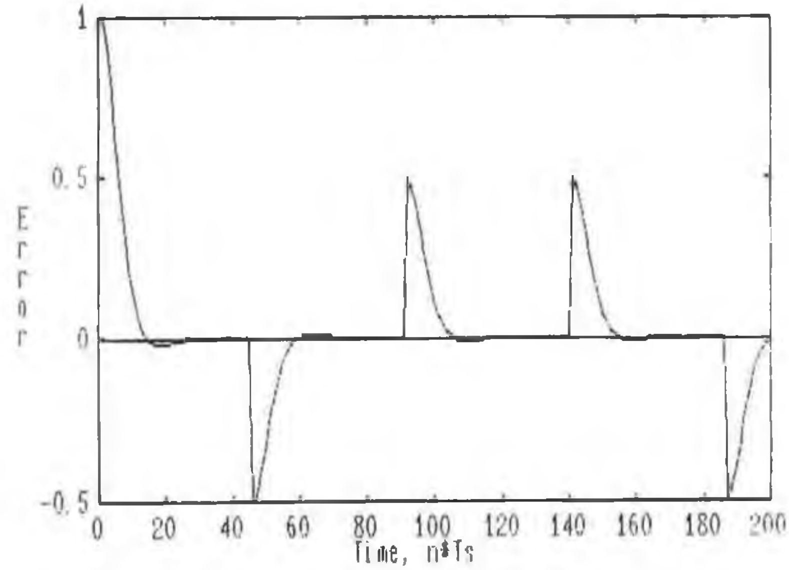

FIG. 3.4(c): Observed Output Error

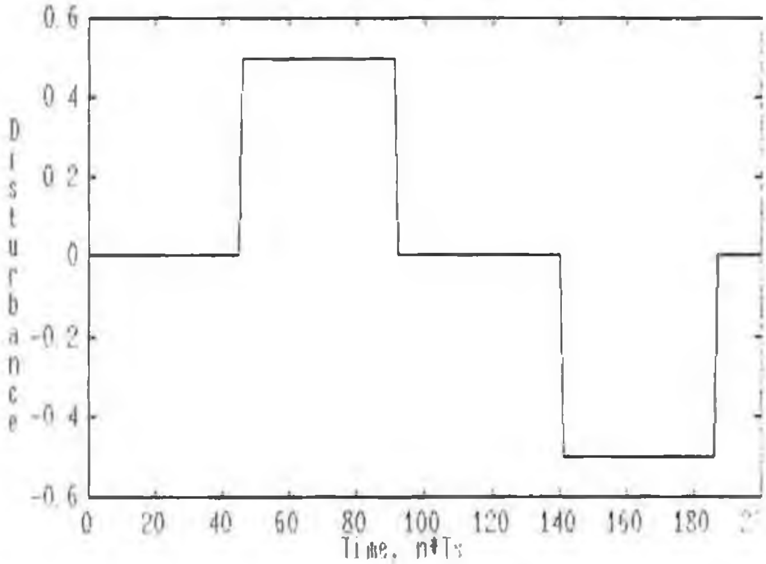

FIG. 3.4(d): Output Disturbance

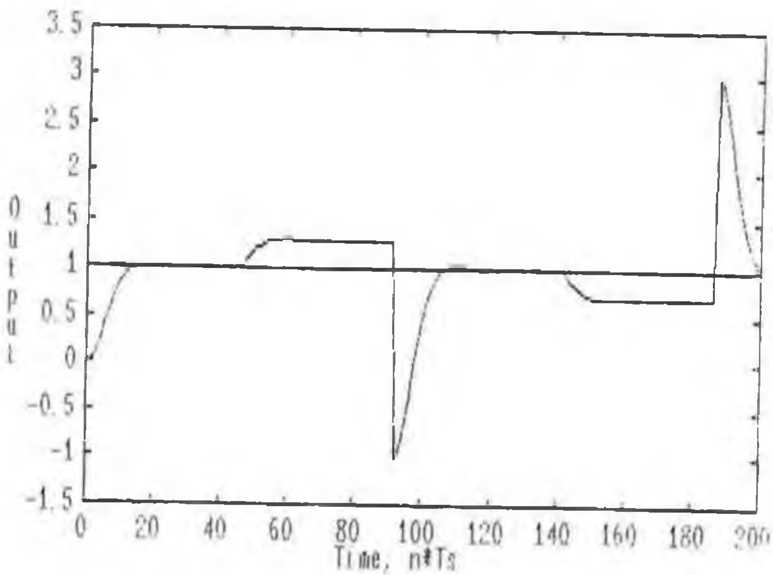

FIG. 3.5(a): Response to Ramp Disturbance

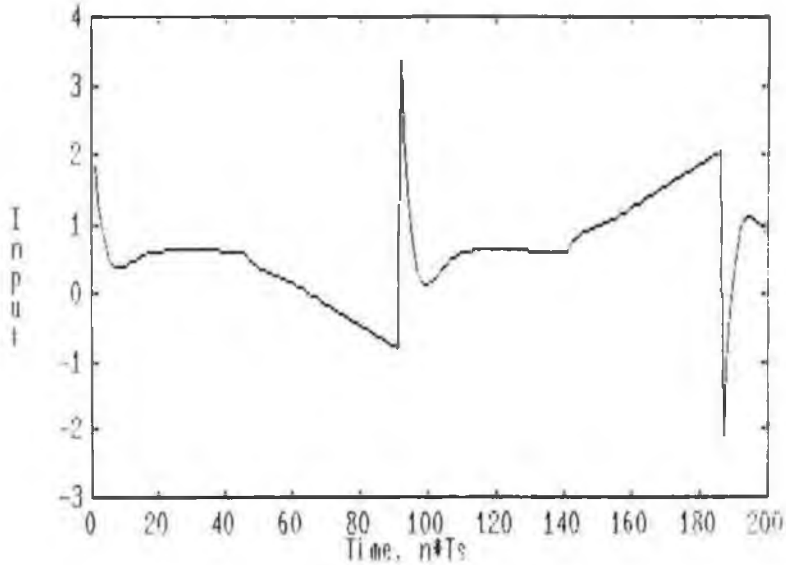

FIG. 3.5(b): Control Input

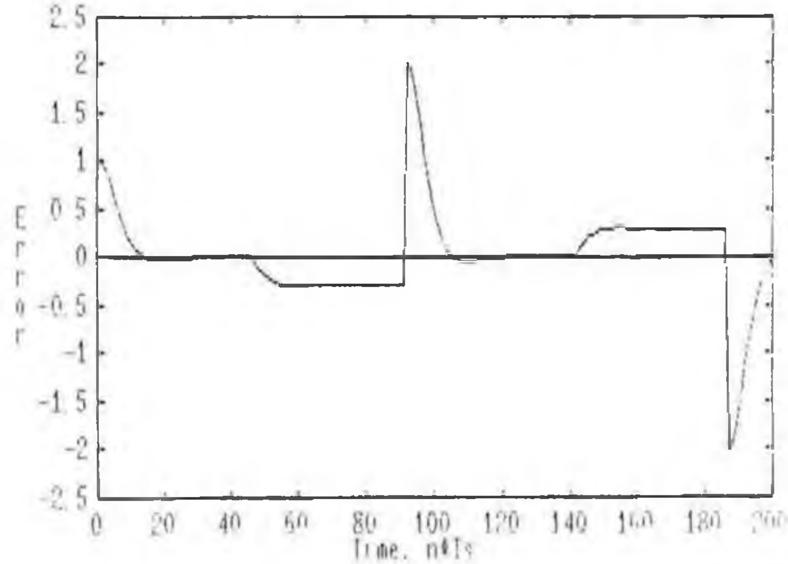

FIG. 3.5(c): Observed Output Error

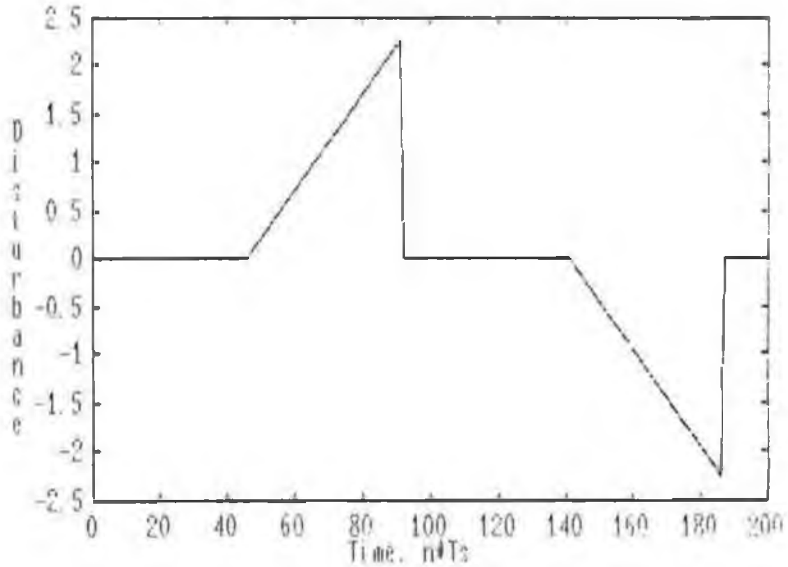

FIG. 3.5(d): Output Disturbance

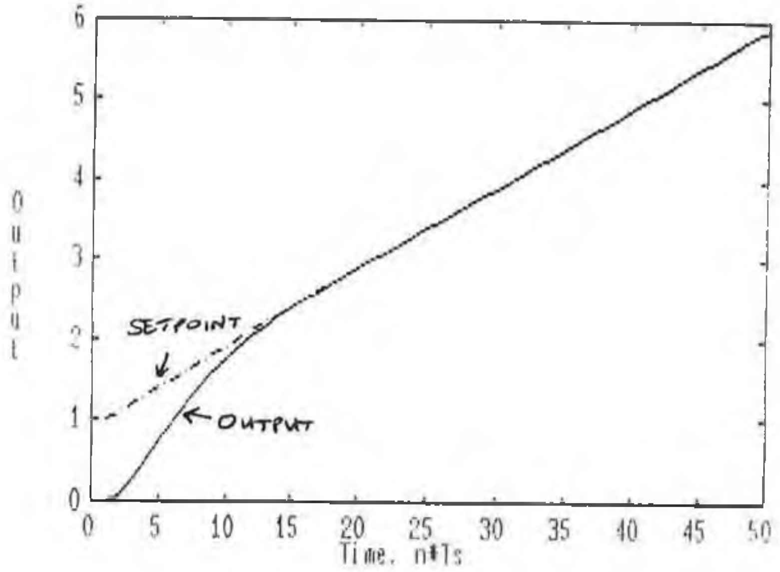

FIG. 3.6(a): Trajectory Tracking

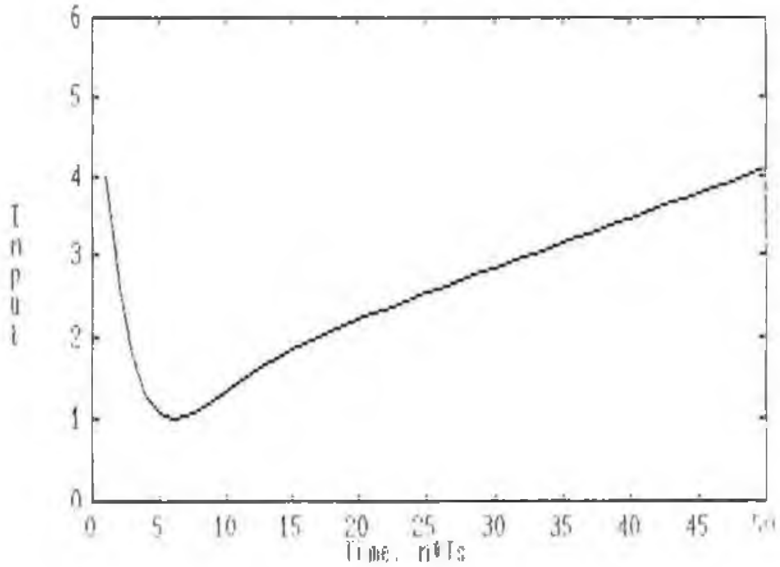

FIG. 3.6(b): Control Input

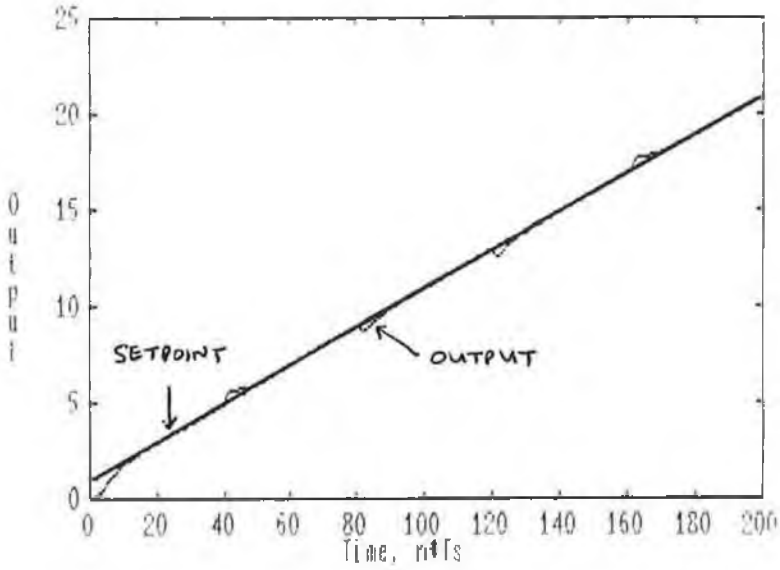

FIG. 3.7(a): Response to Step Disturbances

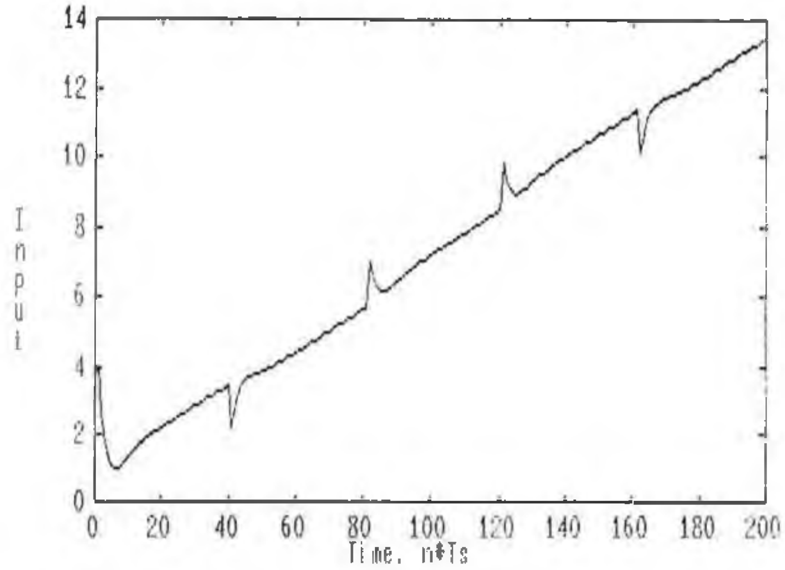

FIG. 3.7(b): Control Input

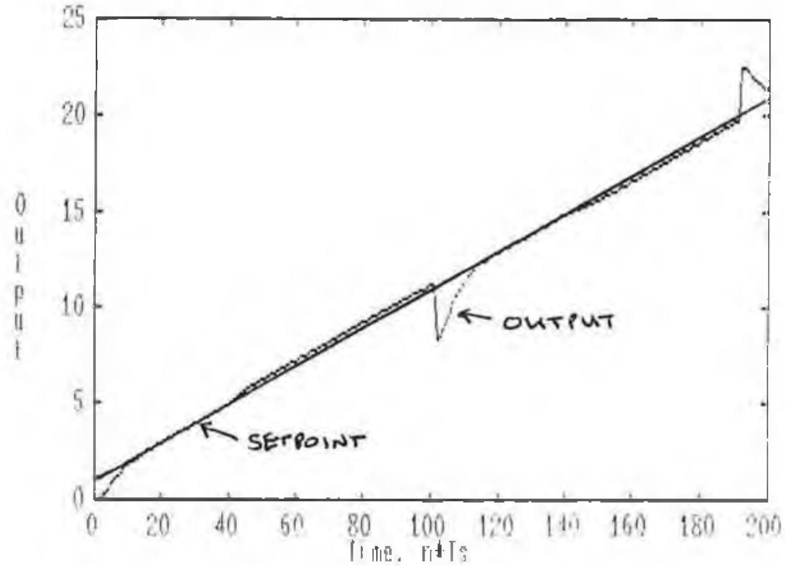

FIG. 3.8(a): Response to Ramp Disturbances

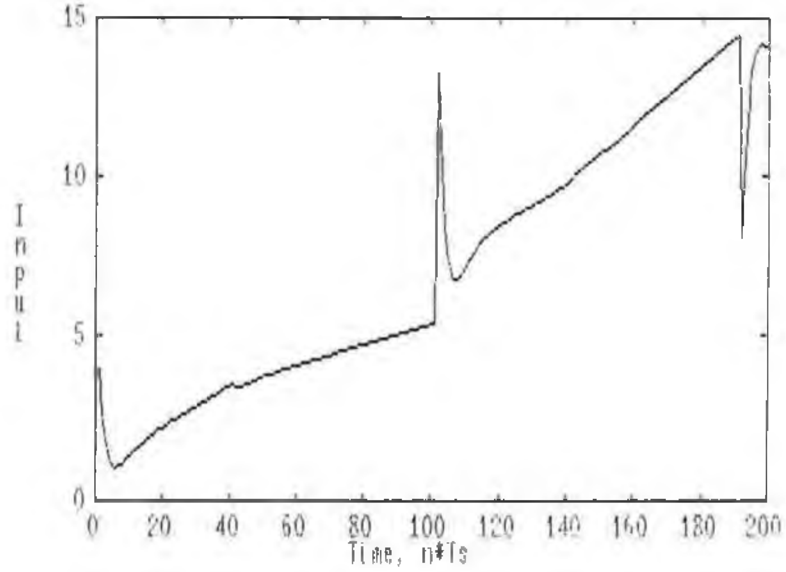

FIG. 3.8(b): Control Input

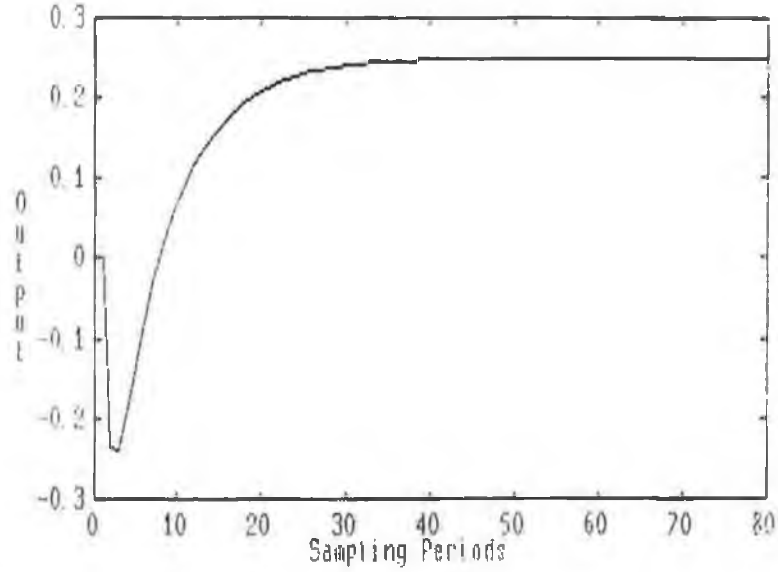

FIG. 3.9: Open Loop Step Response - Nonminimum Phase Process

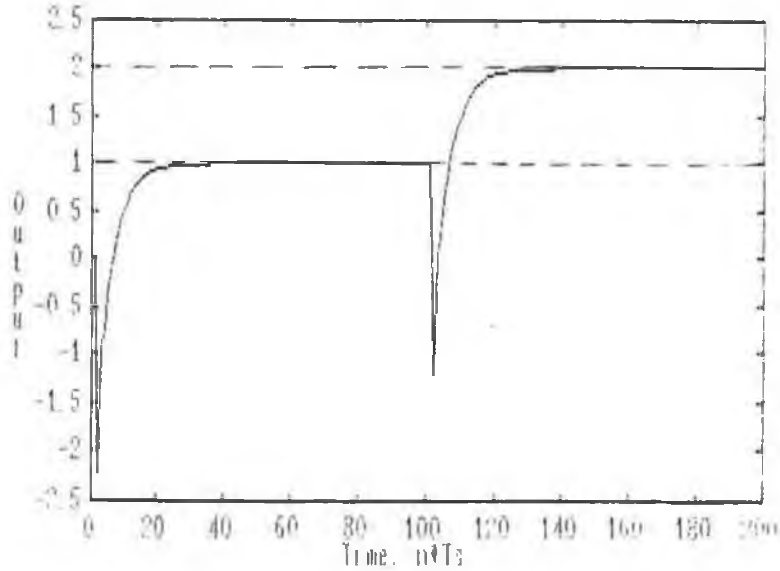

FIG. 3.10(a): Response to Setpoint Change

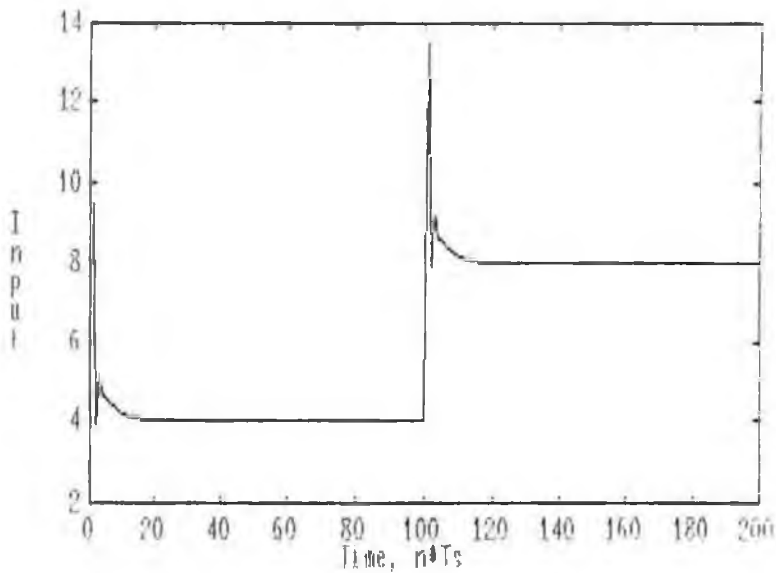

FIG. 3.10(b): Control Input

control input, and 3.3(c) demonstrates the zero-offset performance by plotting the error with time.

The effect of step disturbances directly added to the output were investigated. In practice the disturbance would typically be filtered first. The control results are presented in fig. 3.4(a)-(d) analagous to above. Fig. 3.4(d) illustrates the applied disturbance. Excellent tracking and disturbance compensation may be observed (the setpoint is set at unity in these plots).

The response to ramp disturbances is demonstrated in fig. 3.5(a)-(d). As described in section 3.3.3 an offset may be observed on the output. A self-compensator is required to eliminate this.

A trajectory tracking regulator was also designed for the above second-order plant to follow command trajectories up to first order. This requires the specification of ramp and step base functions. The following design parameters were chosen:

Step:	h = 0.7 sec.	TR = 0.9 sec.
Ramp:	h = 0.8 sec.	TR = 1.0 sec.

The command trajectory is described by a polynomial; 1.0 + 0.1t. The control parameters are thus:

k_0 = 2.586669 k1 = 14.127448

$\underline{v}_x^T$ = [8.641894 -9.3897 0.197458]

Again no self-compensator is employed. The offset free trajectory tracking is shown by fig. 3.6(a). A plot of the control input is given in 3.6(b). The response to step disturbances is shown in fig. 3.7(a). Zero-lag error results as expected. There is an offset when ramp disturbances are applied to the plant output. This is demonstrated in fig. 3.8(a).

Finally, control of a nonminimum phase process was investigated. The plant model used is:

$$(1 - 1.1212016.z^{-1} + 0.22313019.z^{-2}).y(t) = (-0.2348813 + 0.26036357.z^{-1}).u(t-1)$$

with it's step response given in fig. 3.9. A setpoint regulator was designed for this plant using h=1.0 sec. and TR=1.5 sec. The control parameters are:

k_0 = 9.454699 $\underline{v}_x^T$ = [-7.376108 -0.920565 1.074178]

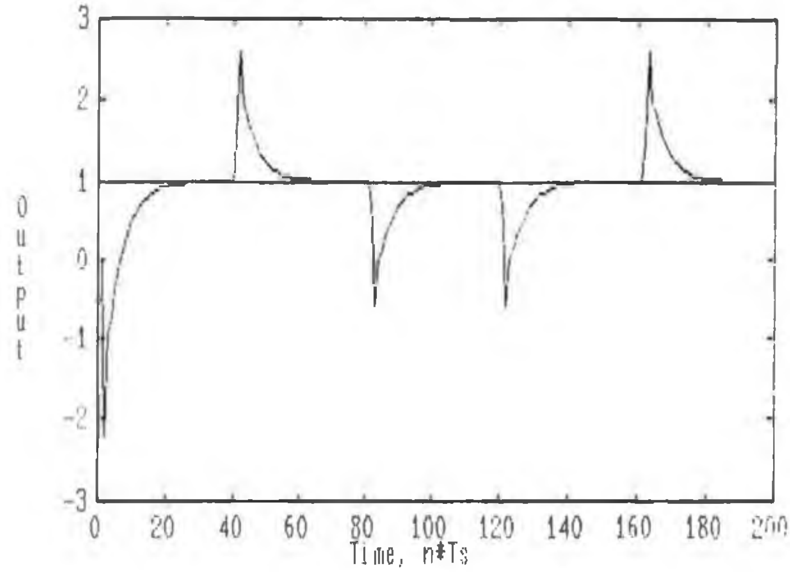

FIG. 3.11(a): Response to Step Disturbances

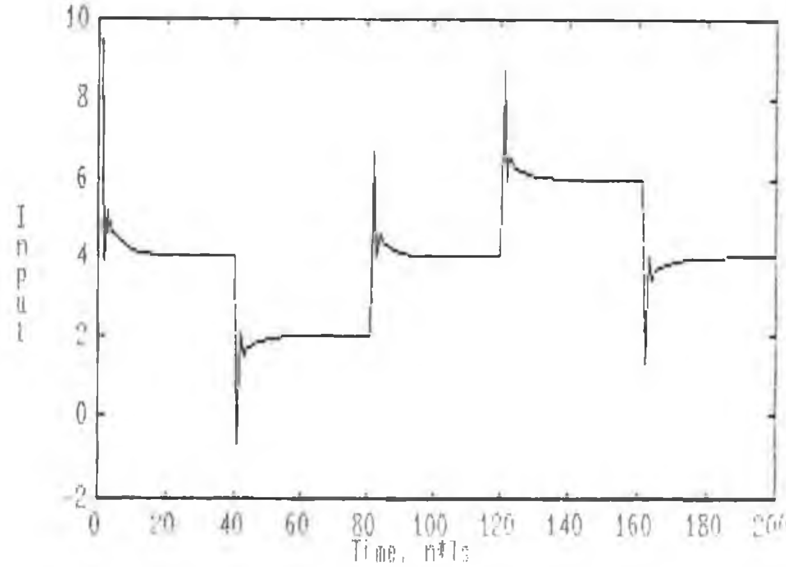

FIG. 3.11(b): Control Input

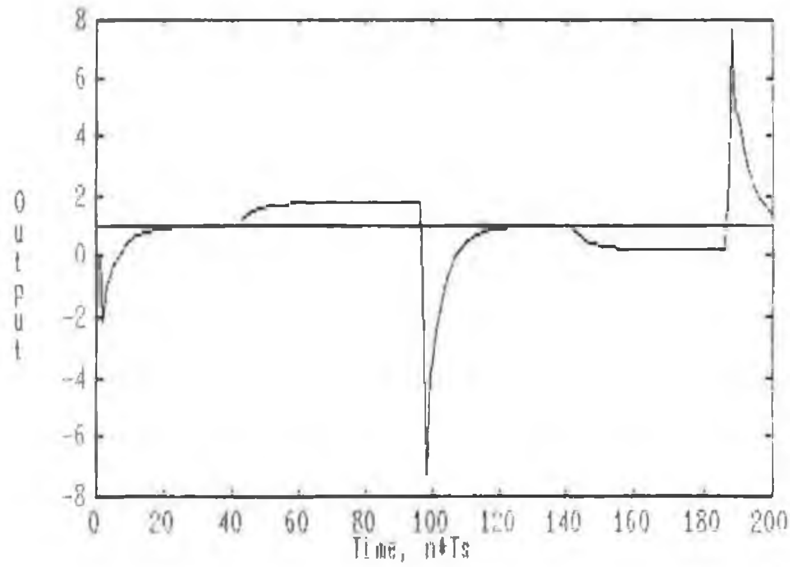

FIG. 3.12(a): Response to Ramp Disturbances

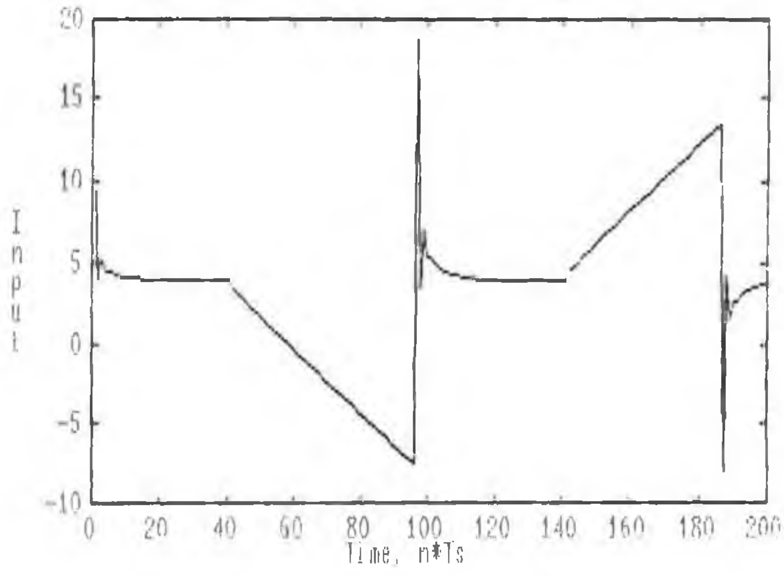

FIG. 3.12(b): Control Input

The response to a setpoint change is indicated in fig. 3.10(a)-(b). Perfect tracking is obtained. Step and ramp disturbances were added to produce the results if fig. 3.11(a)-(b) and fig. 3.12(a)-(b) respectively. Zero-offset regulation is obtained for step disturbances with the ramp producing an output offset as expected. The algorithm can achieve zero-offset regulation despite the nonminimum phase nature of the process.

3.4 ADAPTIVE PFC

An adaptive PFC (APFC) controller may be derived by adding a recursive parameter estimation layer to the feedback structure. This is achieved through the *certainty equivalence principle*[4,5] which allows any control law to be matched with the parameter estimator. Certainty equivalence ignores the stochastic nature of the current estimates and assumes that they are correct. The control law is designed under this assumption for the corresponding deterministic model.

It is important that both the control law and the estimation method are robust to produce a robust adaptive controller. It was seen in section 3.3 that PFC is a very robust control technique. Several different estimation techniques exist[130]. It has been shown[130] that three methods best suited to adaptive control are:

° recursive least squares (RLS)
° recursive extended least squares (RELS)
° recursive maximum likelihood (RML)

RLS is the method most commonly used in adaptive control in the literature. It is a robust estimation technique requiring few online calculations at each iteration. APFC is designed using RLS as the outer adaptive layer.

3.4.1 Recursive Least Squares

Let the process be represented by an ARMAX model of the form

$$[1-A(z^{-1})].S_p(z) = B(z^{-1}).U(z) + E(z) \qquad (3.85)$$

where, $A(z^{-1}) = a_1z^{-1} + a_2z^{-2} + \dots + a_{na}z^{-na}$
$B(z^{-1}) = b_1z^{-1} + b_2z^{-2} + \dots + b_{nb}z^{-nb}$

and $S_p(z)$, $U(z)$, and $E(z)$ are the process output, input, and uncorrelated stochastic noise (including the effects of disturbances) respectively. This may be rewritten as

$$sp(n) = \Theta^T.\Psi(n) + e(n) \tag{3.86}$$

where, $\Theta^T = [a_1 \ a_2 \ ... \ a_{na} \ b_1 \ b_2 \ ... \ b_{nb}]$
$\Psi(n) = [sp(n-1) \ sp(n-2) \ ... \ sp(n-na) \ u(n-1) \ u(n-2) \ ... \ u(n-nb)]$

At each instant 'n' it is required to estimate the parameter vector ($\Theta(n)$) based on the measured or known data vector $\Psi(n)$.

Using the estimated parameter vector, a prediction error between the measured output and the predicted output may be calculated:

$$\epsilon(n) = sp(n) - \Psi^T(n).\Theta(n-1) \tag{3.87}$$

If $\epsilon(n)$ is small then $\Theta(n)$, the new estimates, should not be modified much and vice-versa. A weighting or gain factor k(n) is used to control the parameter updating[]:

$$\Theta(n) = \Theta(n-1) + k(n).\epsilon(n) \tag{3.88}$$

The choice of k(n), also called the Kalman gain, is quite critical for performance. Large k(n) values gives fast convergence but large perturbations while small values improve noise rejection but reduce the rate of convergence[130]. In practice, k(n) is chosen as a time varing gain in terms of the estimation error covariance matrix (i.e. P(n))[130]:

$$k(n) = P(n).\Psi(n) \tag{3.89}$$

The elements of the covariance matrix are minimised by choosing

$$k(n) = \frac{P(n-1).\Psi(n)}{\lambda + \Psi^T(n).P(n-1).\Psi(n)} \tag{3.90}$$

and the covariance matrix is updated by

$$P(n) = [P(n-1) - k(n).\Psi^T(n).P(n-1)]/\lambda \tag{3.91}$$

where λ is an exponential forgetting factor such that new data is relatively important and its importance declines exponentially.

Equations (3.87), (3.88), (3.90), and (3.91) comprise the RLS estimator implemented as part of the APFC stategy. Initial estimates of $\Theta(0)$ and P(0) are required to start the algorithm. If the confidence in $\Theta(0)$ is low then P(0) should be chosen as ρI (where I = identity matrix and ρ = a large number). Large elements of P(0) imply little confidence in $\Theta(0)$ and result in faster convergence. The forgetting factor λ is usually chosen so that $0.95<\lambda<1$. This choice will be discussed below.

The full RLS algorithm is:

(1) Initialise (@n=0): $\Psi(0)$, $\Theta(0)$, P(0). Set n = 1.
(2) $k(n) = P(n-1).\Psi(n)/[\lambda + \Psi^T(n).P(n-1).\Psi(n)]$
(3) $\epsilon(n) = sp(n) - \Psi^T(n).\Theta(n-1)$
(4) $\Theta(n) = \Theta(n-1) + k(n).\epsilon(n)$
(5) $P(n) = [P(n-1) - k(n).\Psi^T(n).P(n-1)]/\lambda$

3.4.2 RLS Extensions

The inclusion of the forgetting factor allows time-varing parameters to be tracked. It does this by ensuring that the error covariance matrix, P(n), does not become too small. This however may also lead to problems of "*estimator windup*" and "*bursting*". These problems and some suggested solutions are given substantial treatment in the literature[4,5,75,120].

Both of these problems essentially stem from a lack of *persistent excitation*[120]. This situation can arise under conditions of good control or when the level of process excitation is low. For example, if the major source of excitation is from setpoint changes then there may be long periods with no excitation at all. In such circumstances, $P(n-1)\Psi(n) \rightarrow 0$ and so $P(n) \rightarrow P(n-1)/\lambda$. As $\lambda<1$, the covariance matrix increases at each iteration (estimator windup). Since P(n) is used to calculate the Kalman gain, k(n) may contain large values causing extreme sensitivity such that small changes in process conditions will cause large jumps in the estimated parameters (i.e. blow-up or bursting phenomenon).

Some solutions to this problem have been proposed[75]. Most solutions are either based on negating the effects of forgetting old data or else the lack of persistent excitation. One attempt is to monitor the system excitation and to add an additional input (eg. PRBS) when appropriate. This is in the spirit of dual control, although it is sometimes undesirable to add perturbations to the input. Alternatively, the trace of the P matrix may be used to determine if it should be reset to some value. A more common method is to use a variable forgetting factor. λ is adjusted automatically as a function of the prediction error: λ is small when $\epsilon(n)$ is large and vice versa. Another solution is to switch off the estimation routine when the estimates are close to their true levels. This may be judged from the prediction error (eq. 3.87), which causes the estimation to stop if it is less than some pre-specified level.

It is proposed to use a variable forgetting factor with the APFC technique. The method used here was presented in [143]. $\lambda(n)$ should be low at start-up for quick initial tuning and should reach some final value. Therefore define:

$$\lambda_1(n) = \alpha.\lambda_1(n-1) + (1-\alpha).\lambda_1(\infty) \qquad (3.92)$$

where, α = rate of change of λ
$\lambda_1(\infty) = 1$ (to avoid bursting)
$\lambda_1(0) < \lambda_1(\infty)$

To be able to track time-varing parameters, define

$$\lambda_2(n) = 1 - \epsilon(n)^2/(1+\epsilon(n)^2) \qquad (3.93)$$

and use the following forgetting factor

$$\lambda(n) = \lambda_1(n).\lambda_2(n) \qquad (3.94)$$

3.4.3 APFC Algorithm

At each sampling instant, RLS re-estimates the parameters of the ARMAX model representing the process. These new model parameters are then used to update the PFC control law. Refering to the regulator equation of (3.84), this requires recalculation of the matrix $\underline{v}$ and the scalars k_i, i=0,1,...,max(nc,ne). In effect, this means recalculating the new base function outputs at the coincidence points and the α_i's and β_i's from the recurrence relations given in (3.47a) and (3.47b).

For ease of presentation, the APFC algorithm will be described in terms of the setpoint controller derived in section 3.3.2. Extension to the general case of section 3.3.4 is straightforward.

The PFC regulator (eq.3.50) is restated as:

$$u(n) = k_0[c(n)-s_p(n)] - \underline{v}^T.\underline{\varphi}(n)$$

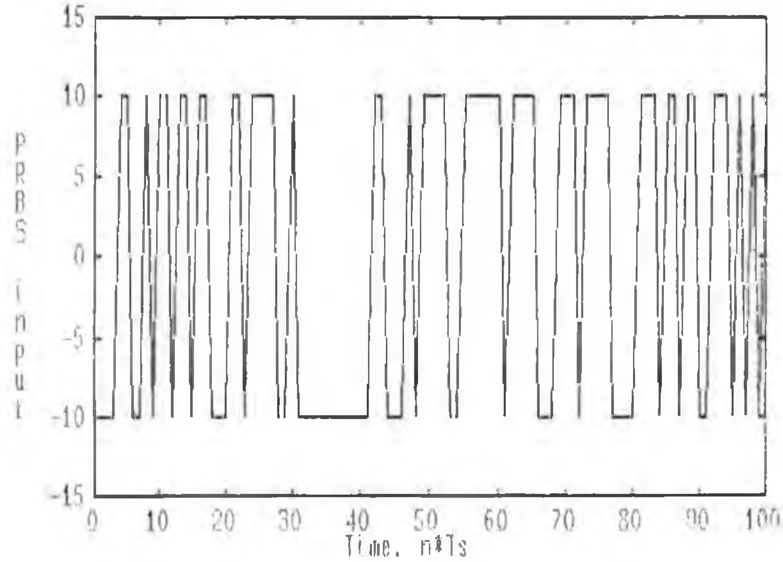

FIG. 3.13: PRBS Input

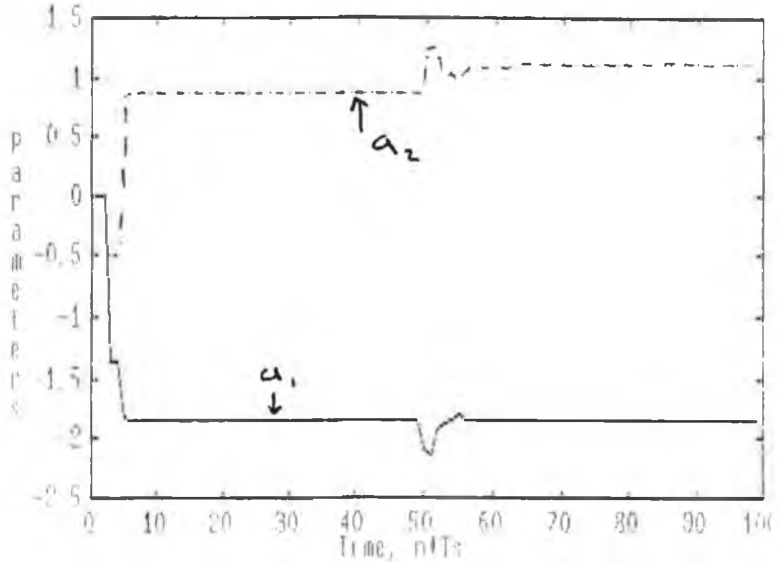

FIG. 3.14(a): Parameter Estimates - a_1, a_2

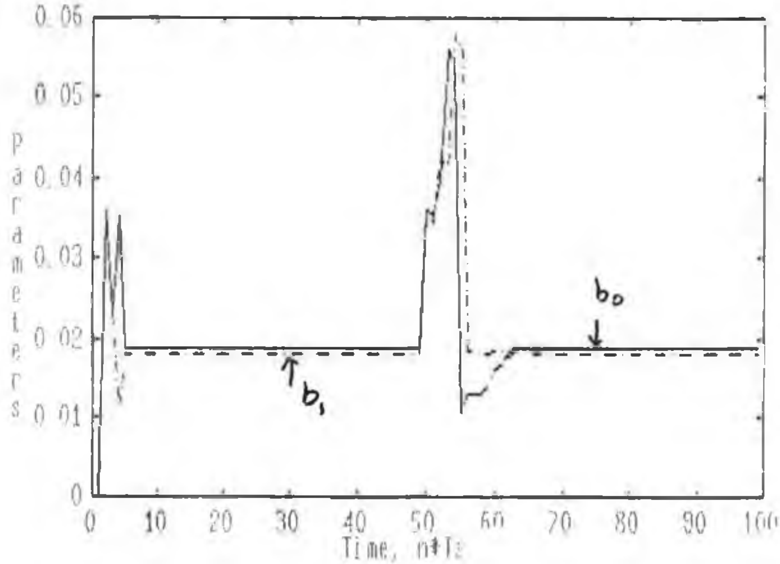

FIG. 3.14(b): Parameter Estimates - b_0, b_1

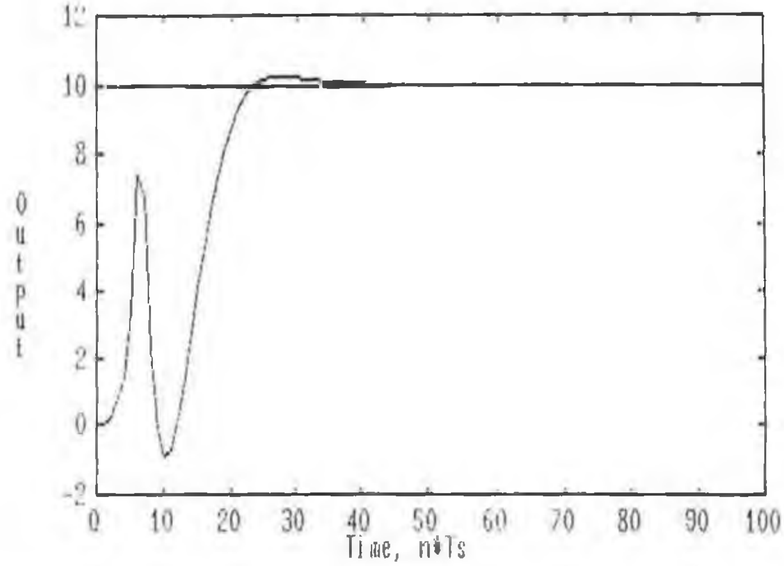

FIG. 3.15: APFC Control Results - Second Order Plant

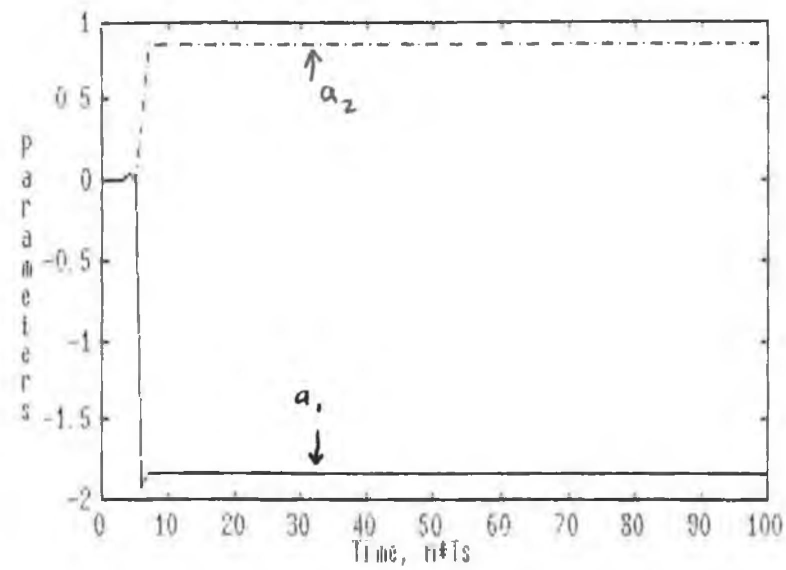

FIG. 3.16(a): Parameter Estimates - a_1, a_2

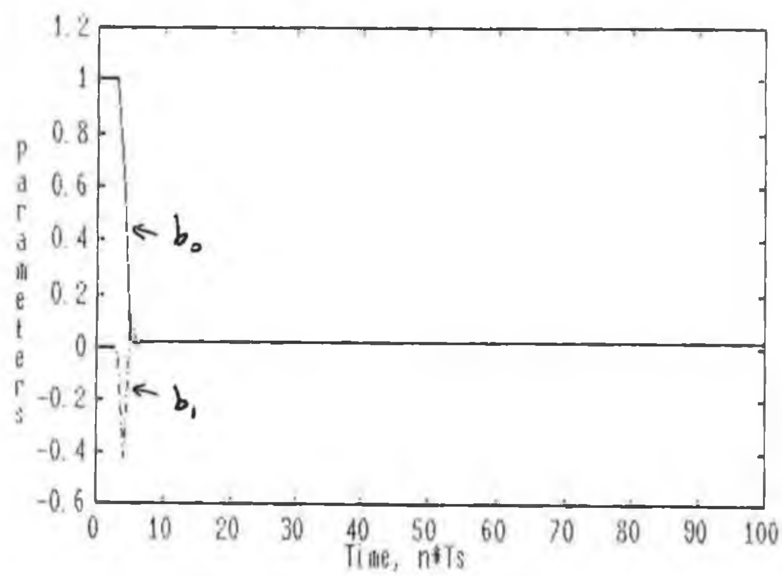

FIG. 3.16(b): Parameter Estimates - b_0, b_1

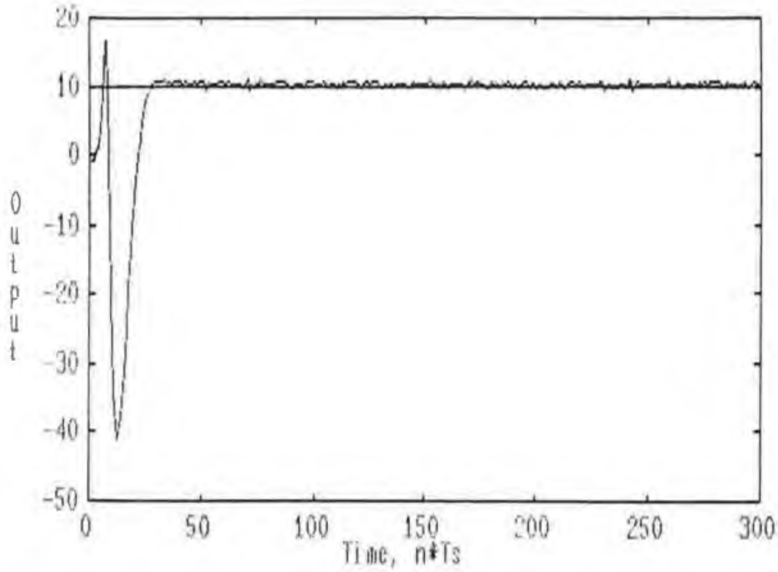

FIG. 3.17: APFC Results with White Noise Additive to Plant Output

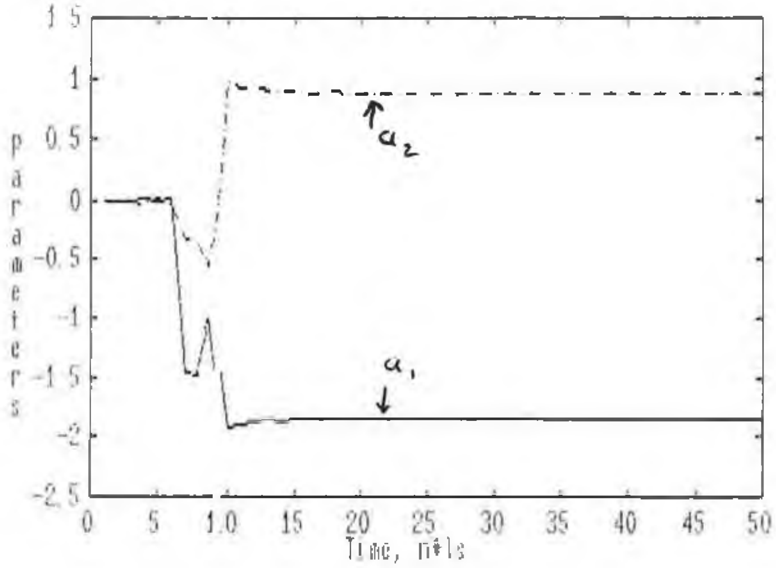

FIG. 3.18(a): Parameter Estimates - a_1, a_2

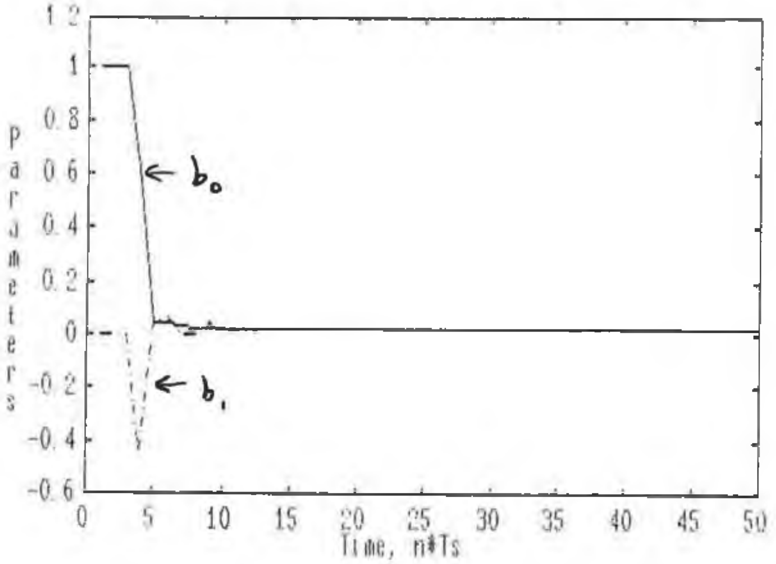

FIG. 3.18(b): Parameter Estimates - b_0, b_1

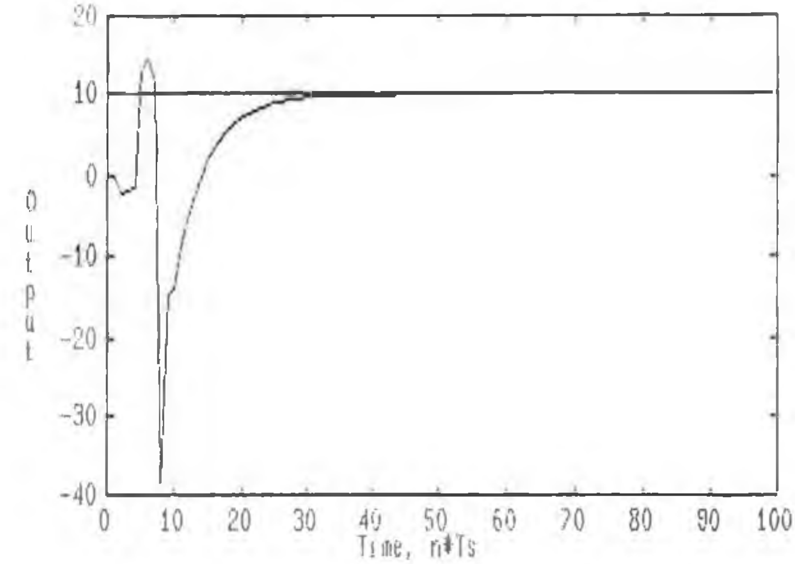

FIG. 3.19: APFC Control Results - Nonminimum Phase Process

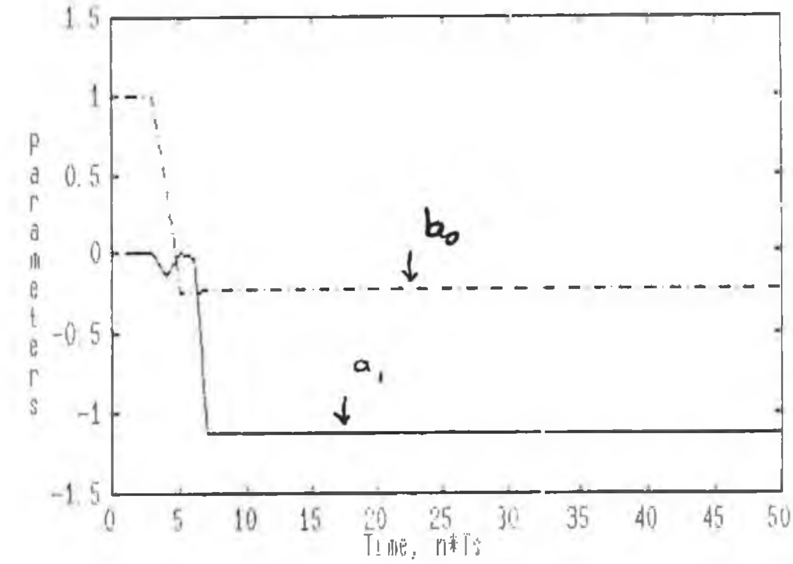

FIG. 3.20(a): Parameter Estimates - a_1, b_0

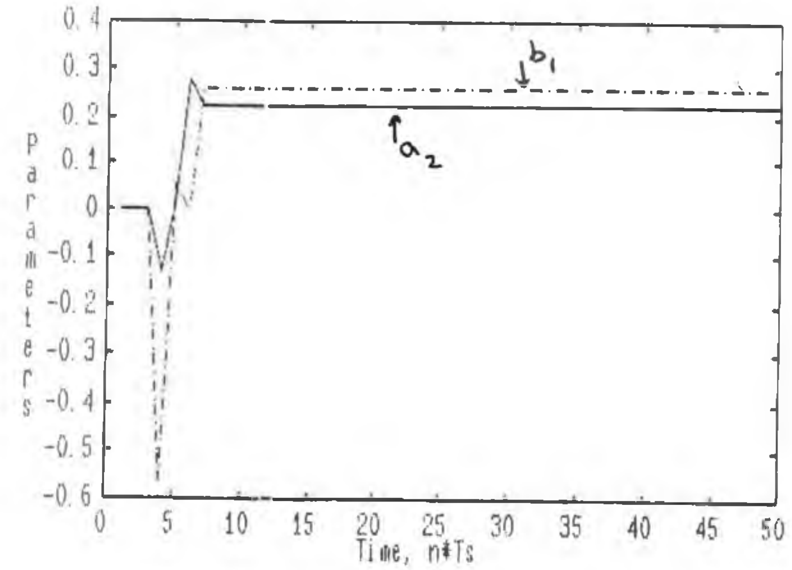

FIG. 3.20(b): Parameter Estimates - a_2, b_1

where, $k_0 = [1/s_0(h)].(1-\alpha^h)$

$\underline{v}^T = [1/s_0(h)].\underline{\Theta}$

$\underline{\Theta} = [\alpha_1\text{-}1 \quad \alpha_2 \quad ... \quad \alpha_{na} \quad \beta_1 \quad ... \quad \beta_{nb-1}]$

The APFC algorithm may be stated as:

(1) Initialise estimator; $\lambda(0)$, $\Theta(0)$, $P(0)$

(2) Calculate $s_0(h)$ using models (3.45) and base function input

(3) Determine α_i's, β_i's using (3.47a), (3.47b)

(4) Calculate $\underline{v}$

(5) Compute the control input and apply

(6) Measure new process output

(7) Recalculate Θ by RLS

(8) n = n+1, goto (2)

3.4.4 APFC Simulation Results

Some simulation results are presented here to demonstrate the performance of the APFC controller. A specific application of APFC to pressure and temperature control in extrusion processes is examined in chapter 4.

All of the following tests were performed on the academic second-order underdamped plant used in section 3.3.5.

The proper operation of the RLS scheme is first demonstrated. The process was subjected to a PRBS input of amplitude of ±10, as shown in fig. 3.13. After t=5 sec. a simulated parameter change was made to demonstrate the fast reconvergence of the estimates. Parameter 'a_2' was changed from 0.860708 to 1.10. Plots of the estimated parameters using RLS are in fig. 3.14. The algorithm was started with $\Theta(0)=0$ and $P(0)=\rho I$ with $\rho=10^4$. The variable forgetting factor was initialised with $\lambda(0)=0.95$ and $\alpha=0.9$. It can be seen that the RLS estimation scheme operates correctly.

Control of the above plant with APFC was investigated. Setpoint control of the output at a level of 10 was attempted with the RLS scheme initialised as above except for b_0, one of the estimated plant parameters. This was set to 1.0 to avoid division by zero. In a practical example a more suitable value would be used. The PFC controller parameters used were the same as employed in section 3.3.5 for this plant. The performance of APFC is given in fig. 3.15 with the estimated parameters in fig. 3.16. Fig. 3.17 shows the regulation of the output when white noise is added to the output of the process. The estimated parameters during control are shown in fig. 3.18. Again it may be observed that the APFC algorithm gives excellent results.

APFC was also applied to the nonminimum phase (NMP) process considered in section 3.3.5. Fig. 3.19 shows the regulation of the plant output at a level of 10 with the RLS scheme initialised as above. The estimated parameters are shown in fig. 3.20. It may be deduced that APFC can control NMP processes quite easily unlike some adaptive regulators which tend to go unstable in such circumstances.

3.5 CONCLUSIONS

The use of LRPC as the basis of a learning based predictive controller designed within the intelligent control framework was proposed. The major LRPC strategies were reviewed and compared. Predictive Functional Control (PFC) was selected for several reasons. It is capable of offset free tracking control, an equivalent linear regulator may be easily derived and it requires few online calculations. The design procedure is also CAD compatible and the tuning parameters are easily related to effects on the closed-loop time response.

To complete the LBPC design criteria an adaptive version of the PFC algorithm was derived. This was achieved with use of the certainty equivalence principle by adding a RLS adaptive mechanism around the PFC feedback loop. To facilitate the addition of the adaptive layer PFC was reformulated in terms of an ARMAX model of the plant. The stability and robustness of this new regulator was examined and the need for a self-compensator when the error output perturbation is a ramp or some function of higher degree. A variable forgetting factor was employed in the RLS scheme to counter the causes of estimator windup and the bursting phenomenon.

Simulation performance of the adaptive PFC algorithm was tested on some simple academic plants. It was noted that excellent performance was obtained with zero offset control achieved. It was shown that the I/O model form of PFC could quite easily handle processes with long deadtime and also NMP plants. With APFC, model parameter estimates converged to their true values and it was shown that the algorithm could adaptively control NMP plants. This compares very favourably with classical adaptive controllers which have great difficulty trying to regulate such processes.

Although not considered here, an area deserving of more attention is the theoretical analysis of adaptive predictive controllers. Stability and robustness issues should be investigated and compared to results derived in classical adaptive control. It should however be recognised that the theoretical study of adaptive control and predictive control are both areas still in their infancy.

CHAPTER 4

LBPC APPLIED TO EXTRUDER CONTROL

4.1 INTRODUCTION

The demand for polymeric materials is increasing at an alarming rate. As plastics find more and varied uses quality specifications are becoming more stringent. The cost of plastics raw material (oil-based) grows steadily focussing attention on the expensive 'out-of-spec' waste product. There is obvious motivation therefore to improve polymer processing operations to allow the production of higher specification products and to reduce waste. This can be achieved by better design of production plants and also by the application of modern control techniques both to new plant designs and also to existing older systems.

Plasticating extruders are one of the main items of equipment used by the polymer processing industries. Extrusion processes are highly complex non-linear distributed parameter systems[144] presenting many problems for high performance control[145]. Currently most industrial extruders use standard analog PI controllers to regulate barrel wall temperatures and melt pressure[145,146]. The development of dynamic models has proved extremely difficult with most control models derived empirically, i.e. by step or stochastic identification techniques. As the process is non-linear and distributed in nature these linear empirical models are restricted within a small range of operating conditions. Their successful application as part of a control scheme for stable performance over a range of operating conditions requires some form of automatic online adjustments of the controller parameters (i.e. a learning mechanism which may be implemented as 'simple' adaptive control). Extruder operation will also benefit from the application of hierarchical techniques for systematic control of the full plant. The extruder system is composed of three dynamic sections, shown in fig. 4.1, normally investigated separately for control purposes. Hierarchical schemes have been presented[147,148], giving improved control results.

The extrusion process has many of the elements discussed in chapter 1 identifying it as a suitable application area for intelligent control. The use of LBPC for low-level control should give good performance and the utilisation of a hierarchical scheme, perhaps with an expert system supervisor would seem appropriate. The expert system could supervise the operation of the major control loop (i.e. melt characteristics at the die) and also the barrel wall temperature profile control scheme (see chapter 6). It also provides the opportunity to improve the operator interface and to include additional safety nets and expert fault detection systems.

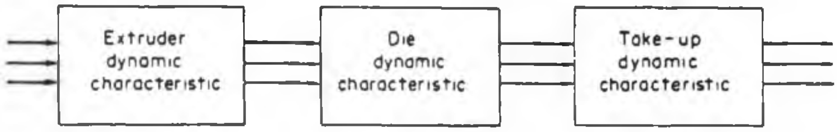

FIG. 4.1: Three Dynamic Stages of an Extrusion Process

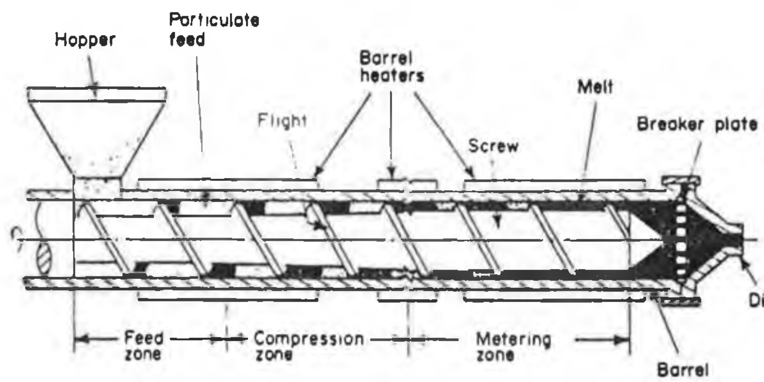

FIG. 4.2: A Single Screw Plasticating Extruder

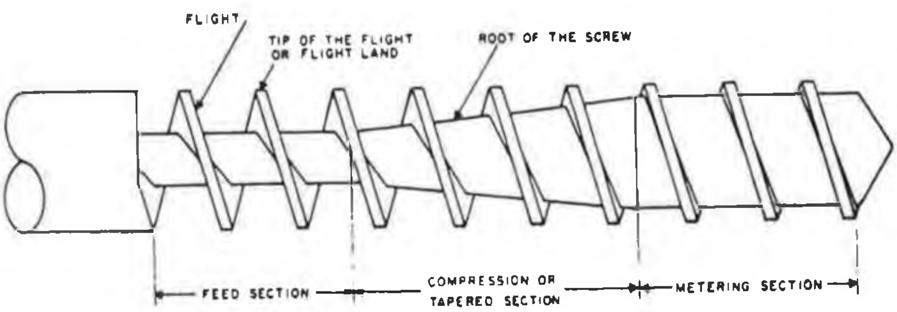

FIG. 4.3: Three Sections of a Metering Screw

This chapter reviews attempts at modelling and control of extrusion processes. A particular extruder example is chosen and simulated. APFC regulators are designed for these models and their operation simulated. The control results are compared to other results presented for the same extruder using different regulator types.

4.2 EXTRUSION PROCESS

A detailed description of the underlying mechanisms of the extrusion system are described below. Consideration is given to the cause and effect of disturbances so that a realistic simulation may be achieved. A survey of modelling and control of extruders is also presented.

4.2.1 Extruder Description

A diagram of a typical single-screw plasticating extruder is shown in fig. 4.2. The feedstock enters the extruder in a solid pellet or powder form via the hopper. The screw rotates in a heated cylindrical barrel with a die at the outlet end. The die is shaped to produce a polymer of desired form. The screw transports the polymer feed to the die with the feed undergoing several changes along its journey. As a consequence of shearing and heat transfer though the barrel the feed is gradually melted and pressurised as it travels downchannel. The material path follows a complex three-dimensional helical path with the screw helix. Ideally the screw should deliver melt to the die at accurately controlled temperature (T_d) and pressure (P_d) to produce a desired output flow rate of high quality product.

Three functional zones may be distinguished[144]:

(1) *Solids Conveying:*

This zone extends from the hopper to the point at which melting first starts. The length of the zone varies as a function of the operating characteristics.

(2) *Melting:*

Both melt and solid coexists in this section with the solid bed profile (SBP) gradually decreasing in size as one progresses downchannel towards the screw. This zone ends when the SBP is reduced to zero.

(3) *Melt Conveying:*

This zone is that length of channel which consists entirely of polymer melt and extends from the end of the melting section to the die.

Analysis and mechanistic modelling of extruders is generally performed by considering each of the above sections individually and deriving appropriate models of temperature, pressure and flowrate within each.

A common type of screw is the *metering* screw which has three geometrical sections[144,149] as shown in fig. 4.3. These are functions of the screw geometry and do not vary with operating conditions, unlike the functional zones above whose lengths may change. The three geometric sections are:

(1) *Feed:*
A section with a relatively deep constant channel depth.

(2) *Compression:*
The part of the barrel where the channel depth decreases linearly with downchannel distance.

(3) *Metering:*
The shallow constant channel depth section just prior to the screw.

The physical mechanisms taking place in the extruder are governed by the differential equations for conservation of energy, momentum and mass, as well as the equations of state and the constitutive properties of the polymer[144,149]. These equations are non-linear and distributed in nature making their solution quite difficult, especially as part of an online control strategy. The oft quoted mechanistic model of [150] deals with the two principal mechanisms of melting and melt conveying. The SBP, melt temperature and pressure may be calculated along the channel length using this model. A static model of solid conveying is given in [144].

The two principal mechanisms consist of several dynamic transport processes[150]. Heat and momentum transport in the thin melt film between the barrel and the solid bed are one mechanism. These determine the rate of melting at every point along the channel. Diffusional heat and momentum transport must also be considered within the melt channel. The other mechanism is melt and solid convection in the downchannel direction. A differential mass balance on the solid bed may be used to calculate the instantaneous SBP:

$$-\rho_s . V_{sz} . \frac{\partial(HX)}{\partial z} \quad - \quad \varphi X^{\frac{1}{2}} \quad = \quad \rho_s . \frac{\partial(HX)}{\partial z}$$

$$\left|\begin{matrix}\text{net rate of}\\ \text{mass flow}\end{matrix}\right| \quad - \quad \left|\begin{matrix}\text{rate of}\\ \text{melting}\end{matrix}\right| \quad = \quad \left|\begin{matrix}\text{rate of mass}\\ \text{accumulation}\end{matrix}\right|$$

where ρ_s is the bulk density of the solid bed, X(z,t) is the solid bed width (SBP) at position z and time t, $V_{sz}(t)$ is the downchannel velocity of solid bed, and H(z) is the channel depth. The term $\varphi(z,t).X^{\frac{1}{2}}$ expresses the rate of melting per unit downchannel length.

The melt temperature profile is determined from a differential energy balance on the melt pool:

$$\rho_m.C_m.\frac{\partial}{\partial t}.[H(W-X).(T-T_r)] \quad + \quad C_m.\frac{\partial}{\partial z}.[G_m(T-T_r)] \quad =$$
$$C_m.\varphi.X^{\frac{1}{2}}.(T_f-T_r) \quad + \quad q_t \quad + \quad q_v$$

$$\begin{bmatrix}\text{accumulation of}\\ \text{heat in melt}\end{bmatrix} + \begin{bmatrix}\text{net convection of heat}\\ \text{in direction of flow}\end{bmatrix} =$$
$$\begin{bmatrix}\text{heat convection}\\ \text{from melt film}\end{bmatrix} + \begin{bmatrix}\text{net heat}\\ \text{transfer}\\ \text{through walls}\end{bmatrix} + \begin{bmatrix}\text{heat generated}\\ \text{by viscous}\\ \text{dissipation}\end{bmatrix}$$

where ρ_m is the melt density, C_m is the melt specific heat, W is channel width, and T_r and T_f are the reference and film temperatures respectively. G_m is the mass flow rate of the melt.

The pressure profile and flow rate readjust instantaneously to the prevailing conditions. Pressure is a function of SBP, temperature, output flow rate (G), screw speed, and physical properties of the polymer (eg. viscosity). Flow rate is dependent on the melt temperature and pressure, die characteristics and extruder characteristics (eg. channel depth). The rate of melting variable (φ) incorporates operating conditions and physical properties such as barrel wall temperature profile, flow rate, screw speed, viscosity, thermal conductivity, heat of fusion, etc.

From these equations it is obvious that the extrusion process is very complex and difficult to model with partial differential equations to solve and many interactions between the process variables. However, computer simulations based on these models have been presented[144,150,151].

4.2.2 Disturbances

The primary goal of extruder control is to maintain high quality product at high throughput rates. The product quality may be determined by several measurable quantities[144,145]. Extruder plants are designed to produce extrudate of a certain quality when operating in steady state. This is usually specified in terms of the melt temperature and pressure at the die and the throughput rate. It transpires that high quality and high throughput can be conflicting requirements. Hence, further motivation for good control methods exists to balance these objectives and achieve optimum performance.

The causes of poor product quality are fluctuations in the steady state operation of the extrusion line. These have been studied in some detail[144,146]. The fluctuations may be divided into four categories according to their frequency:

High Frequency

The highest frequency fluctuations are those that occur at the same frequency as the screw rotations. There are several causes for these disturbances[144]. One cause is periodic changes in the feed rate due to the passage of the flight at the hopper opening. This type of disturbance is rare however and usually only occurs at low back pressure operation. Other reasons include inadequate screw compression ratio or an excessively cooled screw. Disturbances of their nature can usually be eliminated by increasing the back pressure (through the back pressure valve) or increasing the screw temperature. This may be done external to any process control scheme. Another major source of fluctuations at this frequency is due to improper instrumentation. For example, if the pressure transducer is placed close to the tip of the screw it will register fluctuations caused by the passing of the screw flights. This is not an instability of the extrusion process and has no effect on product quality. It can however, cause severe control problems from the use of these measurements.

Intermediate Frequency

Disturbances in the intermediate frequency range (1-15 cycles/minute) are the main cause of poor product quality[144]. The main sources of these problems are periodic breaking up of the solid bed in the melting region or occasional starve feeding in the solid conveying section. At a certain point in the melting zone the solid bed formed by the unmelted polymer pellets starts to disintegrate. Thus blocks of solid polymer float down the channel and melt slowly by conduction. This causes a variation in the length of the melting region and creates a surging effect in the extruder[144]. It has been found that the point at which the solid bed starts to break up is usually constant in the extruder for most polymers. It is thus a non-stationarity in the melting process having a fairly consistent value independent of the operating conditions. A method to counteract these disturbances was proposed which involved cooling the screw. This stabilises the bed by creating a solid layer on the screw with excellent results reported. However, screw cooling reduces throughput[152]. The best method to counteract these fluctuations is thus with improved control methods.

Low Frequency

Disturbances in the range less than 1 cycle/minute are classified as low frequency[144]. The reasons for these fluctuations are usually external to the extruder. Some of the causes include cycling in the heater power controllers, plant voltage variations, water pressure fluctuations, and variations in feed polymer quality. The only effective method of negating these effects is by feedback control.

Random Fluctuations

Besides the cyclic surging problems discussed above, totally random variations also exist. These can be caused by poor feed section design[144] or random measurement noise on the process instrumentation, among others.

There is a very high degree of interaction among the three primary variables (i.e. pressure, temperature and flowrate) and the above disturbances may be observed on each. The objective of any control strategy would be to minimise the deviation of these variables from their steady state operating levels.

4.2.3 Modelling and Control Survey

As plasticating extrusion is basically a steady state process, early theoretical models developed were steady state in nature[144]. These models are usually used for the design of extruders and prediction of the extruder performance according to a given set of operating conditions and material properties. It is not possible to use them to predict the transient behaviour of the process when the operating conditions are changed. The design of an automatic controller for the process however requires a good dynamic model of this behaviour.

Dynamic behaviour modelling has been attempted from a mechanistic, or first principles, approach. The model most often referred to of this type[150] is described in section 4.2.1. This is based on the assumption that all transport processes taking place in the extruder are fast compared to the bulk flow of the solid bed. Partial differential equations are then set up to solve for the rates of heat, mass, and momentum transport. Another model[153] is based on the idealized melting mechanism[144] and assumes that the extruder may be represented by a series of repeating units each consisting of a well-mixed section and a plug-flowsection (corresponding to the melt and solid bed respectively). Ordinary differential equations (O.D.E) may be set up for each repeating unit and solved. A similar O.D.E. model[151] was based on the model of [150]. It proposed simplifying changes in the form of linearizations of some of the partial derivatives. These mechanistics models are too complex and computationally expensive, besides requiring detailed knowledge such as viscosities, heats of fusion, multiple measurements, etc, to be used as part of an online model-based control scheme. Experimental or empirical models would seem to have more utility from a control viewpoint.

Early experimental models were based on classical methods, i.e. step and impulse response measurements. Transfer function models are then matched to the observed responses. A second-order Laplace transfer function model was derived in this way relating flowrate to changes in screw speed and feeding material[154]. Following this approach, more sophisticated dynamic models were developed which could account for both short-term and long-term responses to various forcing functions[155]. First order models relating die pressure and temperature to screw motor power changes were developed[156]. More recently, it has been proposed[157-159] that all transient responses in the extruder may be modelled by one or a combination of simple Laplace transfer functions. The proposed set include first order, second order, and lead-lag equations. Again both short-term and long-term models were presented to represent the dynamic responses of die and barrel temperatures and pressures to step changes in screw speed and polymer feed. Attempts were made to relate each of the models to a particular phenomenon occuring in the extruder to give more physical meaning to the model[157].

Serious defects exist with the classical modelling approach applied to an extrusion system. Simple step or impulse response tests are not sufficient for modelling complex non-linear systems such as extruders. High frequency disturbances are in effect neglected. A more practical approach is that of time series modelling with stochastic or noise terms included. Typically a fixed-order ARMAX model (as in eq. 3.8) or similar is selected and the parameters of the model are estimated from experimental data. The MV is perturbed by a low amplitude PRBS and some statistical technique used to estimate the parameters. Various stochastic identification techniques for modelling melt temperature dynamics have been compared[160]. Time series models for die melt temperature and pressure were presented for screw speed inputs[160,161]. These however were analysed[146] and found to be inadequate from the point of view of process knoweldge. It was suggested that the experimental data sequences used in their development were too short[146]. Time series models were used as part of a hierarchical scheme[147], also models relating melt temperature and pressure to motor power have been developed[156], and a multivariable model for temperature and pressure to screw speed and back-pressure valve inputs has been presented[162]. Unfortunately, time series modelling is highly application specific with developed models only applicable to the extruder for which they were derived.

Many different control strategies have been applied using the above types of models and some comparisons of their performances made. As previously mentioned most control schemes used on extruders are PID-based or extensions of PID[145,146]. A reference-cascade control scheme determining automatic adjustments of screw speed, valve position, and barrel wall temperature setpoints to PID regulators was also proposed[147]. Most are SISO techniques with either melt pressure or temperature at the die controlled by manipulation of screw speed or sometimes by back-pressure valve position[145].

Due to the complex nonlinear nature of the process it is expected that the controller would need to be retuned when process conditions change. The use of adaptive control schemes has thus received some attention. A self-tuning regulator (STR) was used to control die pressure by manipulating screw speed[156]. A PI controller was also used and both systems where found to give comparable results. This was due to the poor positioning of the pressure sensor such that the flight noise level was exceptionally high. The result was that the STR compensated for the flight noise and diminished its performance. Three controllers were tested and compared for both melt temperature and pressure at the die by screw speed manipulation[162,163]. A minimum variance STR, state-space STR, and PI algrorithm were compared. It was found that the STRs maintaining pressure achieved the best results for least effort. Some multivariable controllers have also been investigated. One strategy uses Dahlin algorithms (with variable deadtime), feedforward control and a dynamic decoupler[148]. Temperature, pressure and extrudate thickness are maintained via screw speed, die heater input, and take-up rate. A MIMO model was employed in the design of two multivariable STRs which accounted for process interactions rather than using decoupling[162]. A STR which minimises a Gaussian quadratic performance index and one based on state feedback pole placement were designed and compared for extruder control. It was found that the latter type was less useful as it required considerably more computations and hence a longer sampling time resulting in reduced performance.

Several survey papers exist which consider the modelling and control of extruders[145,146,157,160].

4.3 A PARTICULAR EXAMPLE

It is proposed to investigate the use of APFC for extruder control as this is a very good application area for an intelligent control strategy. The investigation will be performed via simulation so a suitable extruder model must be chosen. As mentioned above, a SISO ARMAX modelling exercise was performed on a particular extruder on which several control strategies were implemented and their performance results compared[162,163]. It was decided to apply APFC to these models as comparisons could then be made to the results presented.

The extruder modelled has a barrel diameter of 19.05mm with an L\D ratio of 25:1. Screw speed and valve position can be manipulated for control. The pressure at the die is measured by a strain gauge (with a range of 0-68.95 MPa) and a thermocouple (with a range of 0-400°C) is used to measure the temperature at the die. The extruder barrel has four zone heaters used to set the barrel wall temperature profile. The extruder product is low density polyethelene.

Model	a_1	a_2	b_0	b_1	c_0	c_1	σ
Pressure	-0.9142	0.0724	0.0236	0.0147	0.0050	0.2512	0.103
Temp.	-1.0860	0.3565	-0.1137	-0.0572	0.0519	0.1746	0.248

TABLE 4.1: Extruder Model Parameters

Controller	Model	σ_P (MPa)	σ_T (°C)
None	-	0.338	2.19
PI	Pressure	0.273	-
Min. Var. STR	Pressure	0.057	0.332
Min. Var. STR	Temp.	0.243	0.479
State STR	Pressure	0.119	0.805
State STR	Temp.	0.152	0.457

TABLE 4.2: Controller Comparison for Models of table 4.1

Temperature and pressure responses to screw speed were obtained from PRBS tests about a mean input value of 40 rev/min. A second-order linear ARMAX model was estimated for both cases. The pressure responded almost instantaneously to input changes and the temperature lagged the input by 0.3 sec. The same sample time of 2 sec. was used for both identification and control experiments. The ARMAX model is of the form:

$$(1+a_1z^{-1}+a_2z^{-2}).y(t) = (b_0+b_1z^{-1}).u(t-1) + (1+c_1z^{-1}+c_2z^{-2}).e(t)$$

where, y(t) is the output (pressure or temperature)
u(t) is the screw speed input
e(t) is uncorrelated Gaussian white noise with standard deviation σ^2

The parameters of the above model for each response are reproduced in table 4.1. Open-loop simulation responses using these models are shown in fig. 4.4 and 4.5. The model does not account for the low frequency disturbances discussed in section 4.2.2 which were observed on the actual process outputs as fluctuations at a frequency of roughly 0.25 cycles per minute[162,163]. These were also simulated to produce the open-loop uncontrolled outputs shown in fig. 4.6 and 4.7.

Calculation of the uncontrolled standard deviations of the process outputs after 1000 sec. gave:

Pressure, σ_P = 0.37 MPa
Temperature, σ_T = 2.26 °C

Using the above models several different SISO control algorithms (a digital PI controller, a minimum variance STR, and a state-space STR) were tested in [162,163]. The results obtained are given in table 4.2 from which it can be seen that the adaptive schemes give excellent results. The results in table 4.2 will be used to compare against the performance of the APFC algorithm applied to the same models.

4.4 APFC DESIGN & RESULTS

PFC regulators, based on ARMAX models (eq. 3.50) were designed for each of the above process models. The linear regulator has the form:

$$u(n) = k_0[c(n) - s_P(n)] - \underline{v}^T.\varphi(n)$$

where,

$$k_0 = s_0(h)^{-1}.[1-\alpha^h]$$
$$\underline{v} = s_0(h)^{-1}.\underline{\Theta}$$

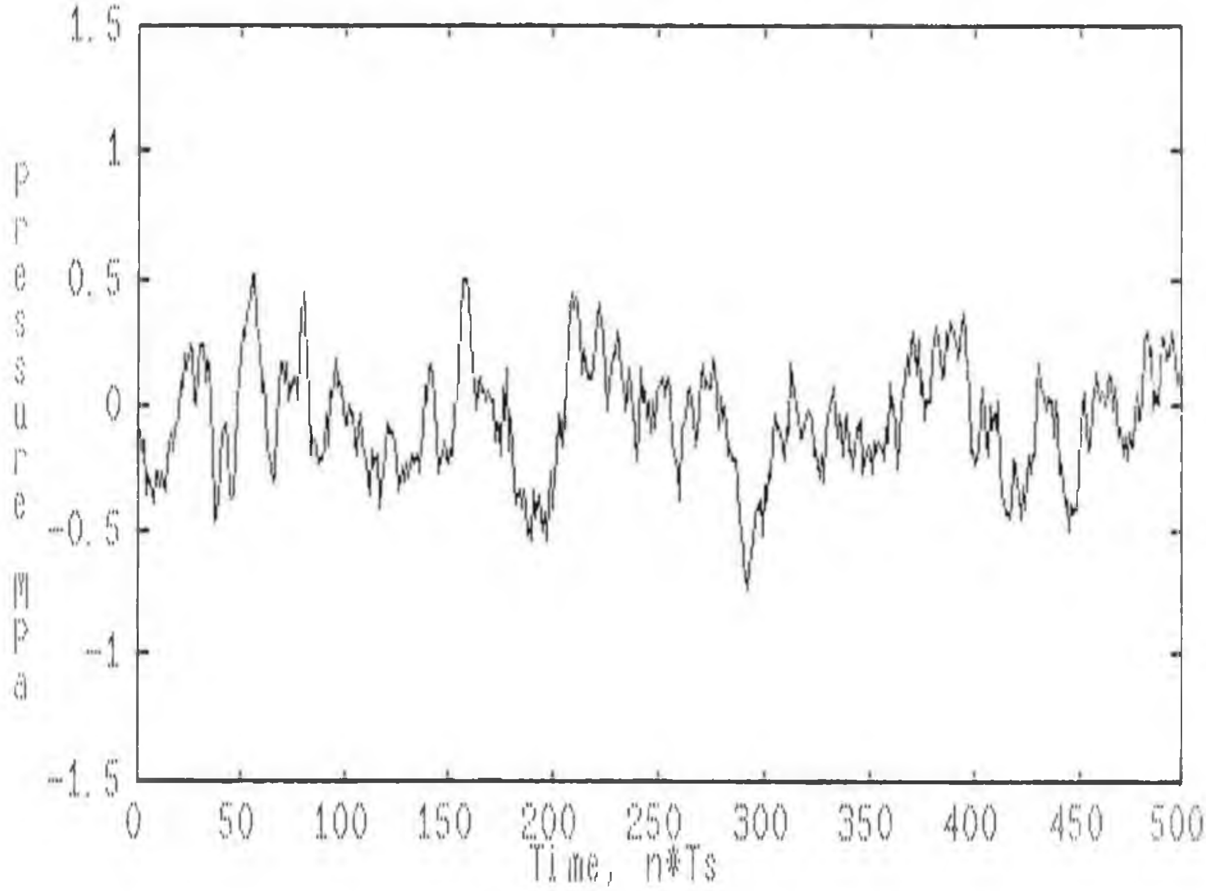

FIG. 4.4: Open Loop Pressure Response

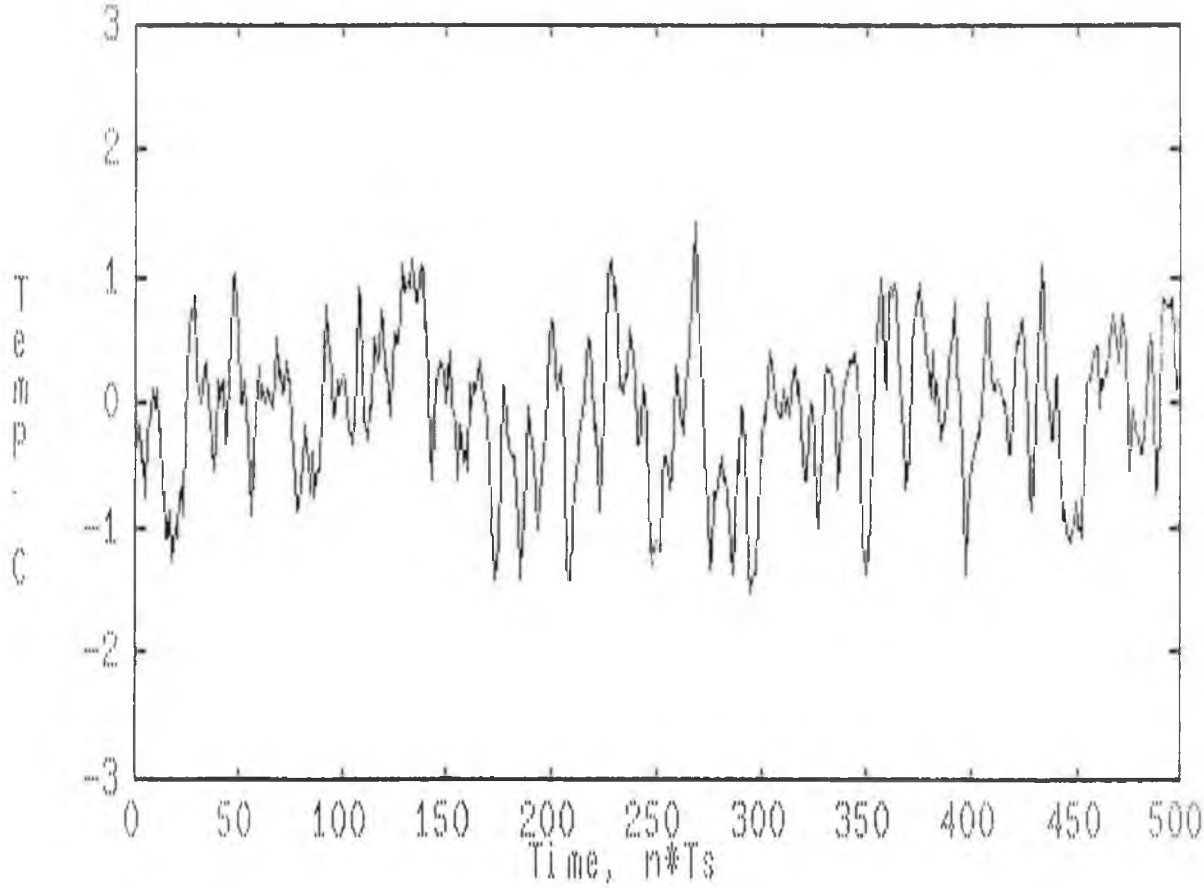

FIG. 4.5: Open Loop Temperature Response

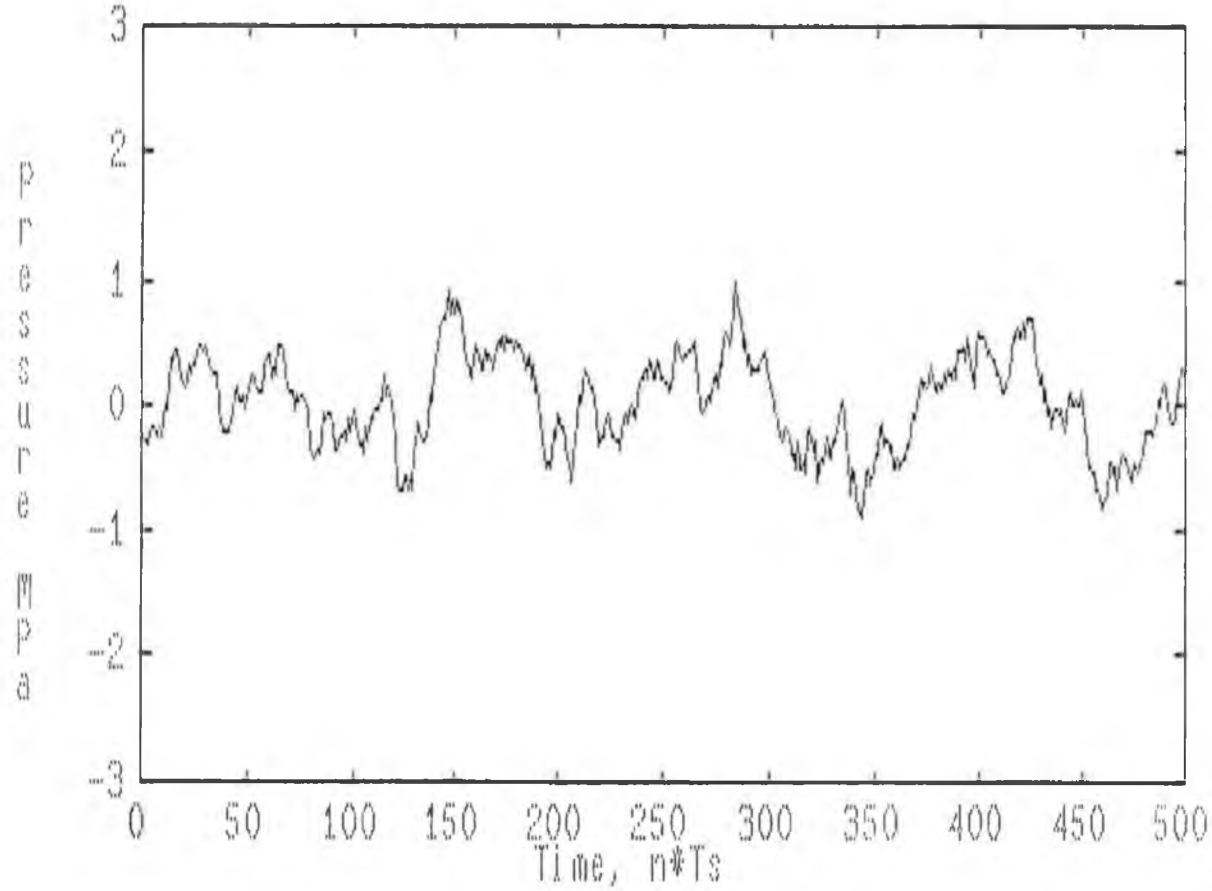

FIG. 4.6: Open Loop Pressure Response (Low Frequency Disturbance Addded)

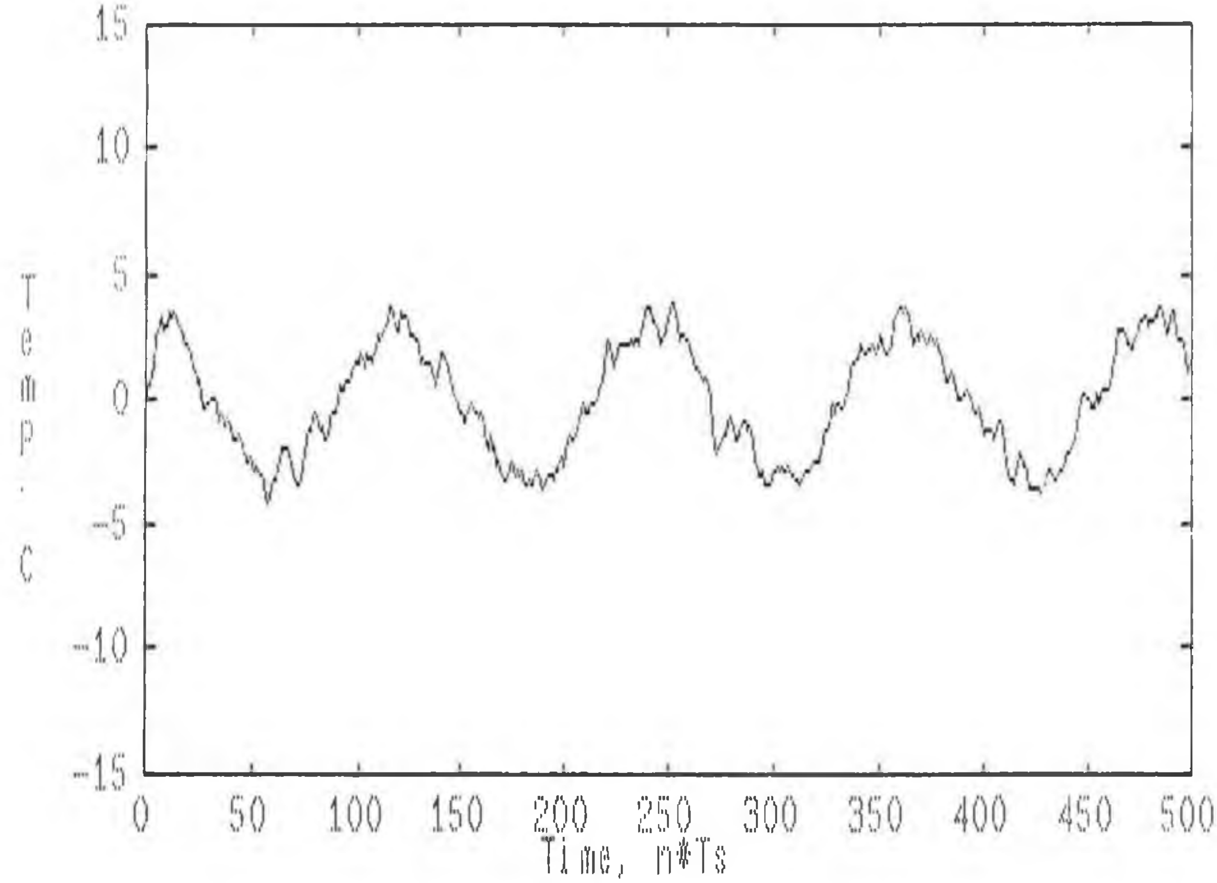

FIG. 4.7: Open Loop Temperature Response (Low Frequency Disturbance Added)

$$\varphi(n) = [sp(n) \quad sp(n-1) \quad u(n-1)]$$
$$\Theta = [\alpha_1 - 1 \quad \alpha_2 \quad \beta_1]$$

with α, h, $s_0(h)$, α_i and β_i as before.

The regulators were tuned for the fixed parameter models with the selected values:

Pressure: h = 4 sec, TR = 6 sec
Temperature: h = 6 sec, TR = 8 sec

The controller parameters were thus derived as:

- Pressure: k_0=14.441135, $\underline{v}_x^T$ = [-3.952199 -1.105435 0.224446]
- Temperature: k_0=-1.987734, $\underline{v}_x^T$ = [1.096505 0.651829 0.104585]

Plots of the controlled responses are given in fig. 4.8 and fig. 4.9. The computed standard deviations after 1000 sec. were:

Pressure:	σ_P = 0.15 MPa
Temperature:	σ_T = 0.55 °C

These compare very favourably with the results presented in [162,163] for control of the models with minimum variance and state-space STR's (as given in section 4.3).

The adaptive PFC algorithm was also tested with these models. Again the model parameters were fixed and the same tuning parameters (h, TR) were used as previously. It was assumed that no knowledge was available and also that the RLS algorithm was initialised with model parameter estimates as zero, except for b0 wfich is set to 1.0. The error covariance matrix was set at 1000.I to indicate poor initial estimates. If such poor estimates were only available then selection of TR and h would have to be cautious. They could be set initially for poor performance and then gradually tightened.

Plots of the controlled responses are shown in fig. 4.10 and fig. 4.11 for pressure and temperature respectively. Calculated standard deviations for these are

Pressure:	σ_P = 0.13 MPa
Temperature:	σ_T = 0.45 °C

which shows better performance with the APFC algorithm than that achieved by the fixed parameter algorithm.

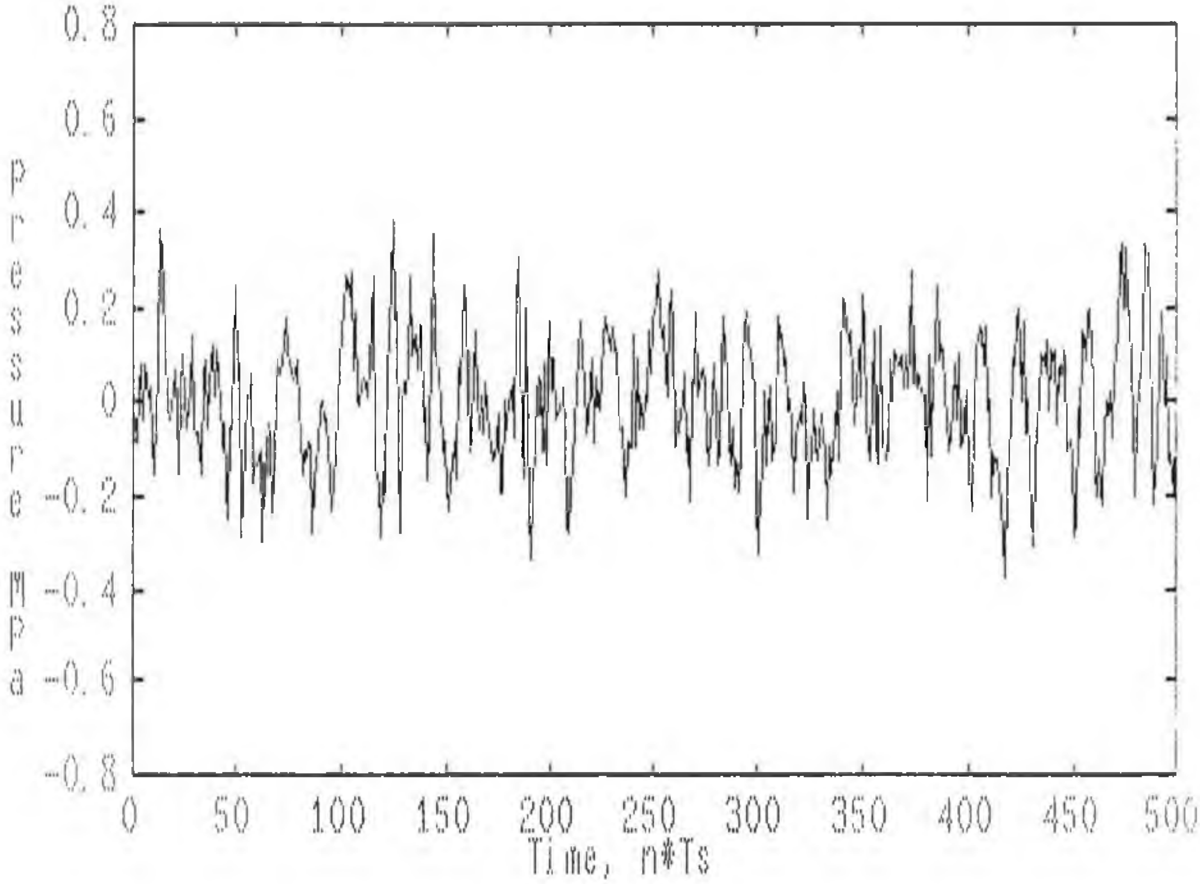

FIG. 4.8(a): Pressure Controlled Response

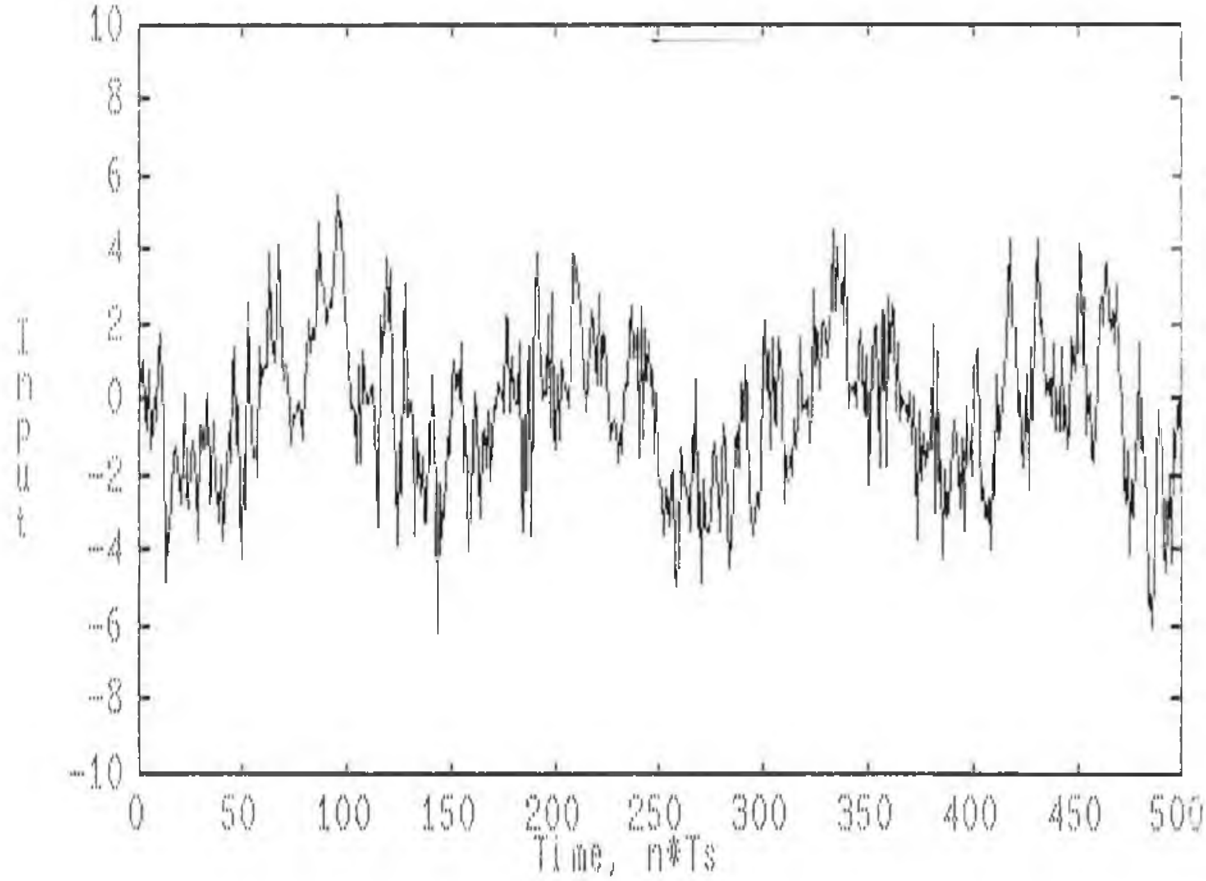

FIG. 4.8(b): Screw Speed Input (Pressure Control)

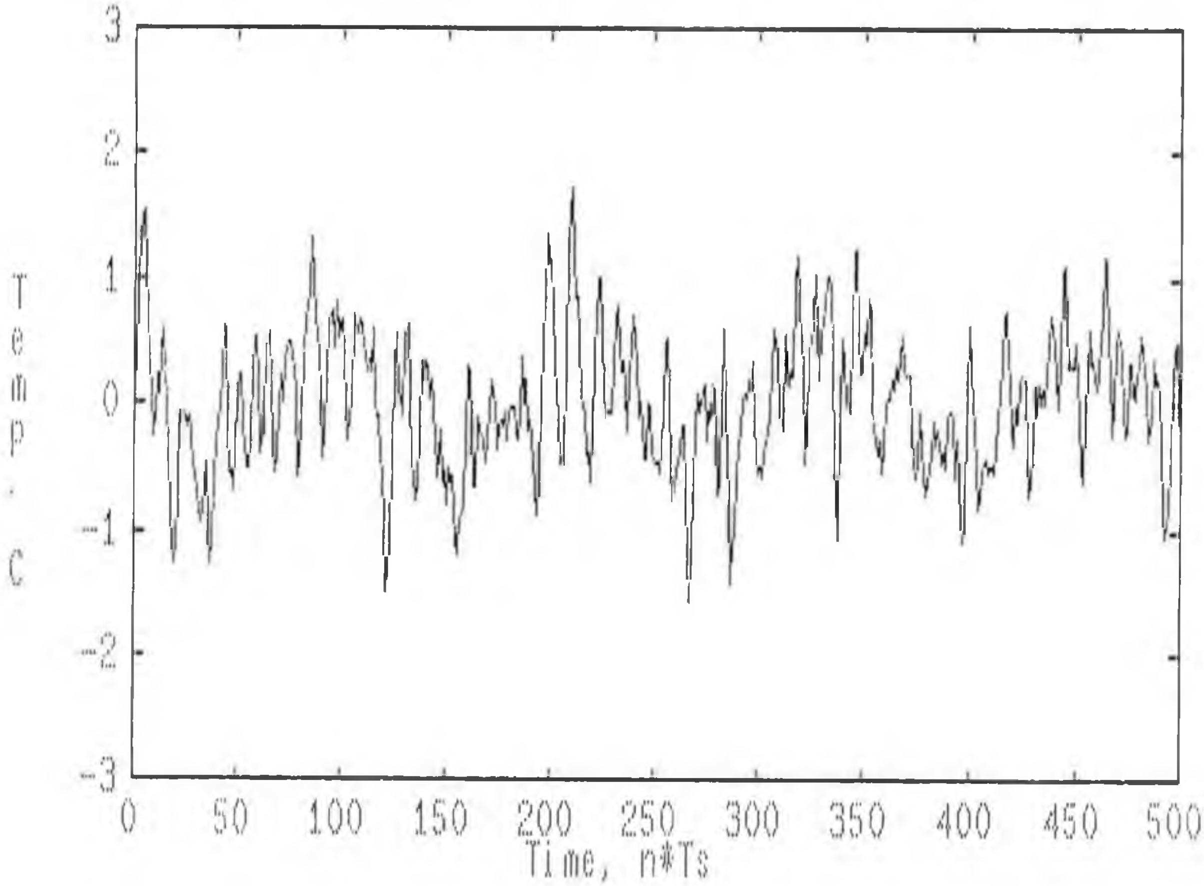

FIG. 4.9(a): Temperature Controlled Response

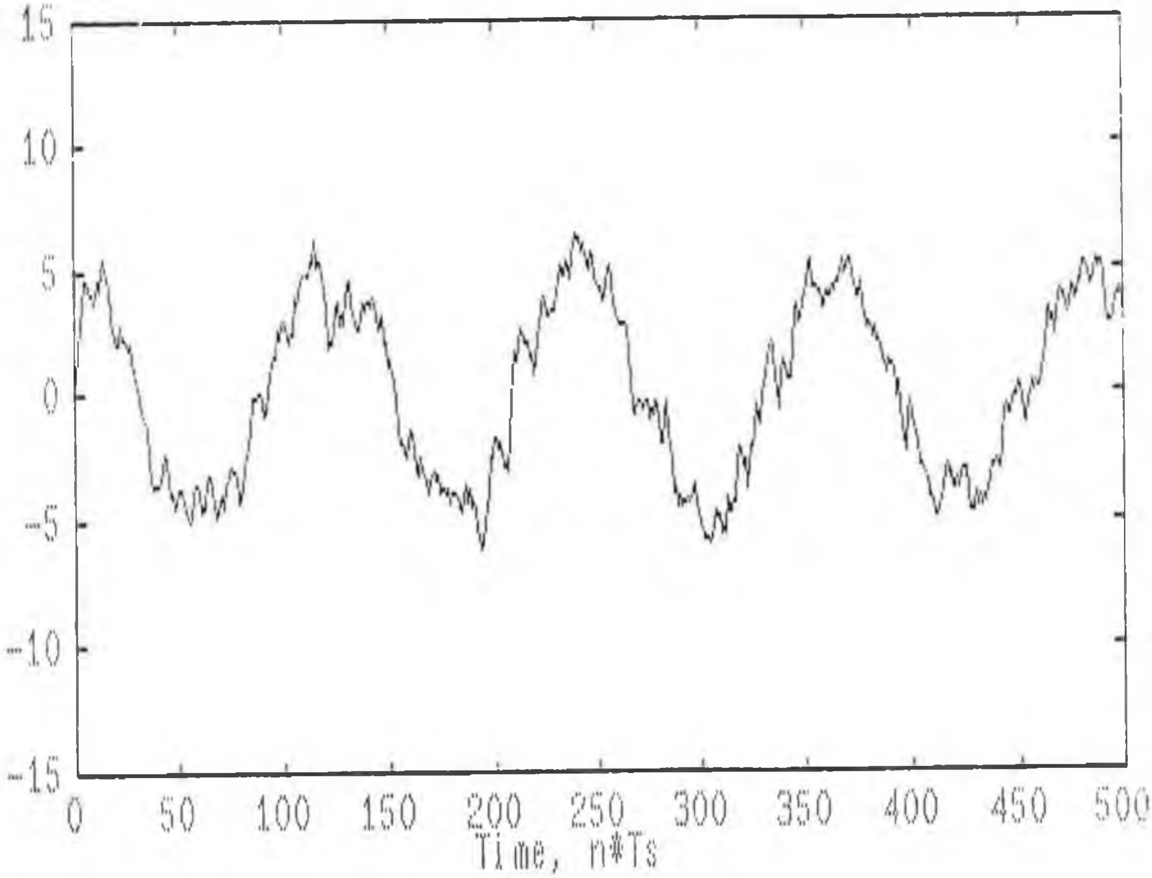

FIG. 4.9(b): Screw Speed Input (Temperature Control)

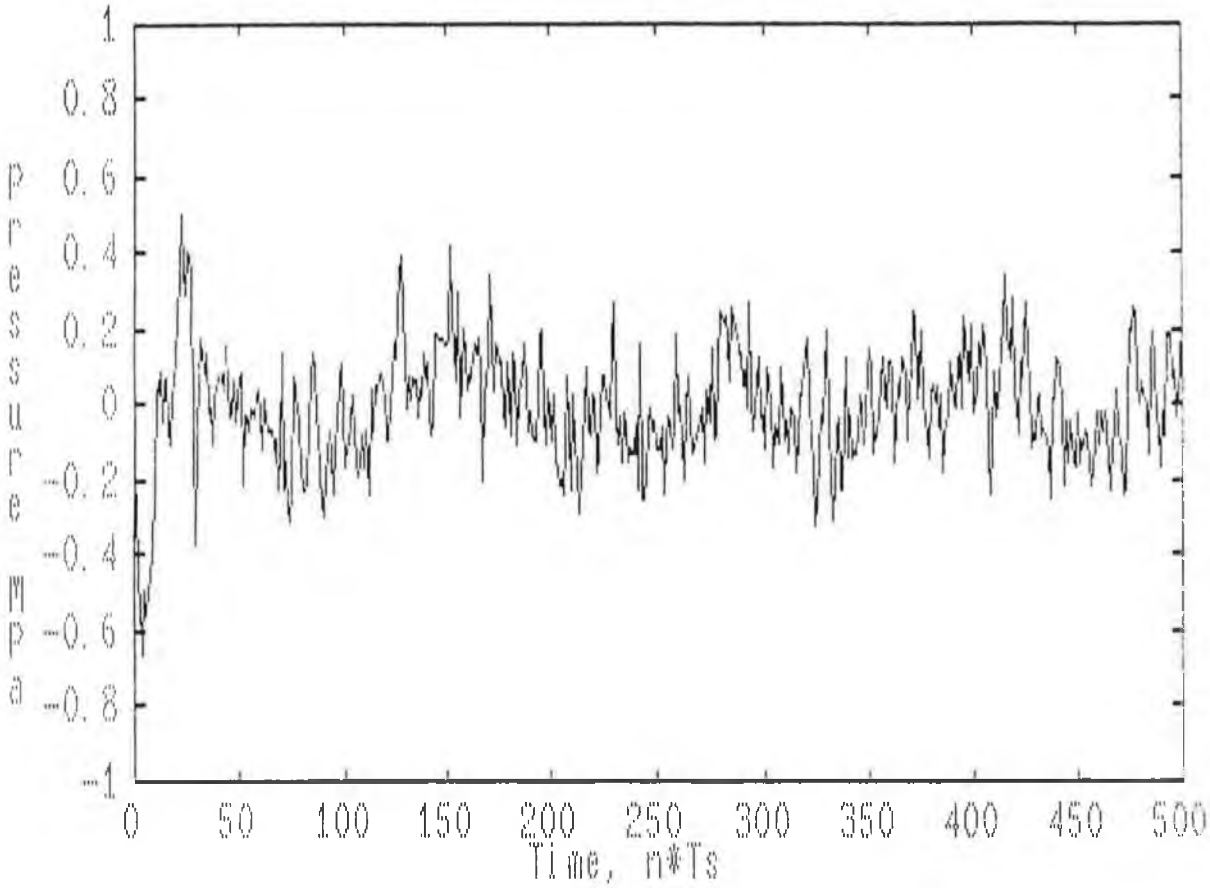

FIG. 4.10(a): APFC Pressure Controlled Response

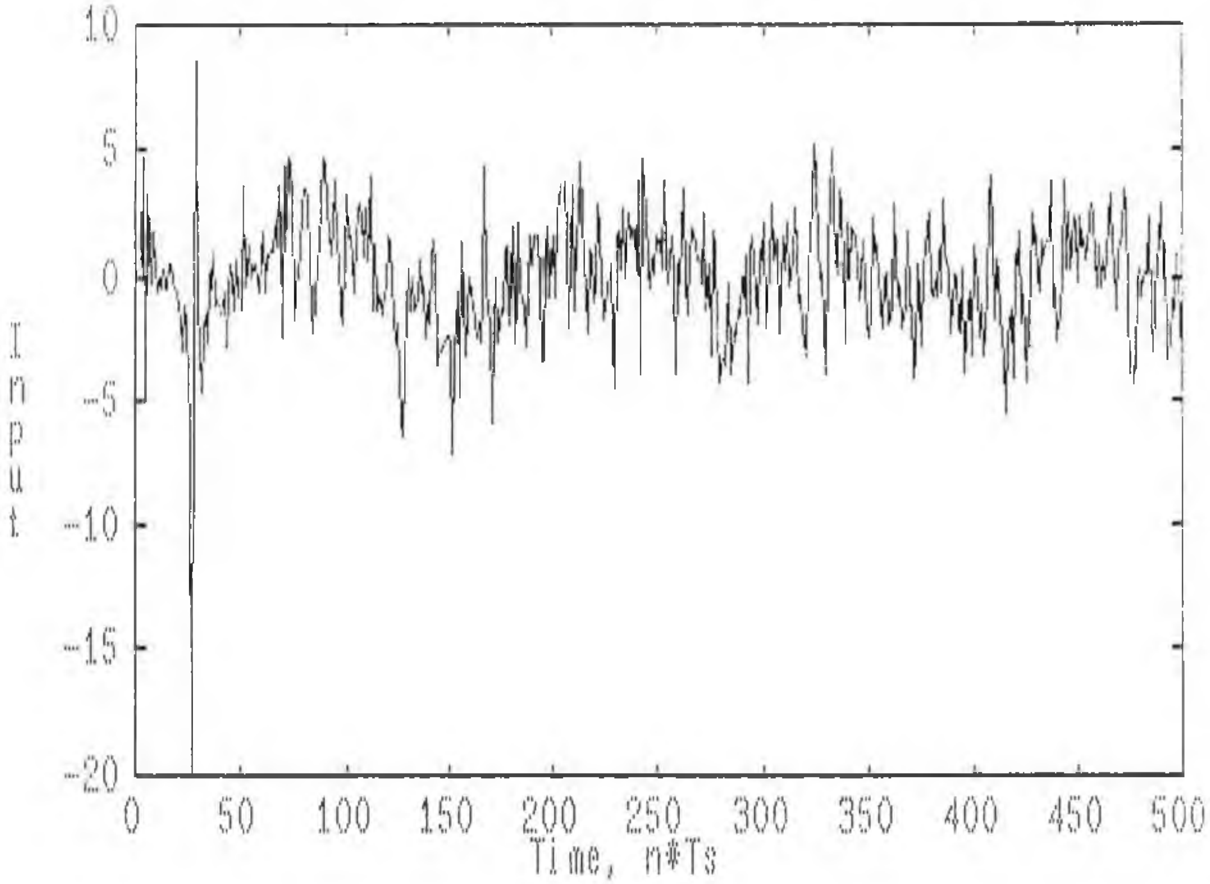

FIG. 4.10(b): Screw Speed Input (Pressure Control)

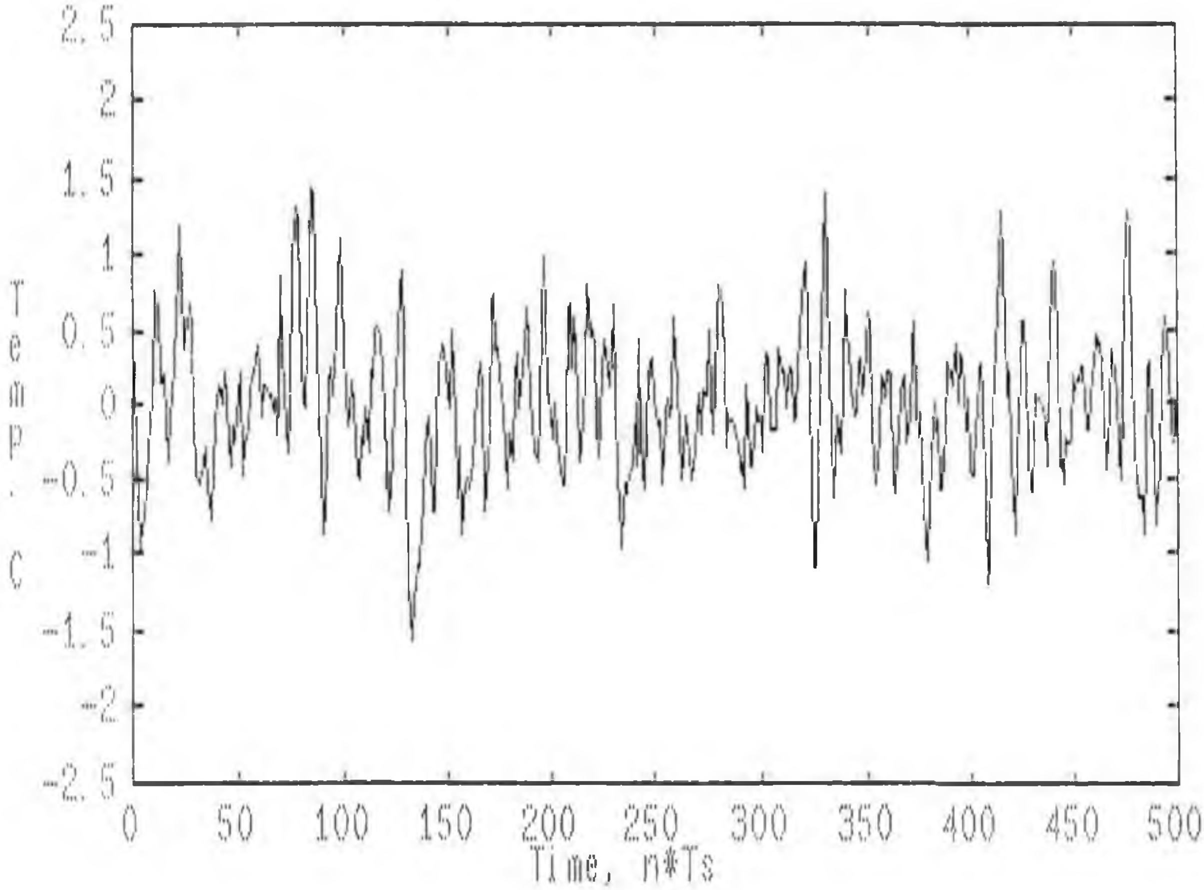

FIG. 4.11(a): APFC Temperature Controlled Response

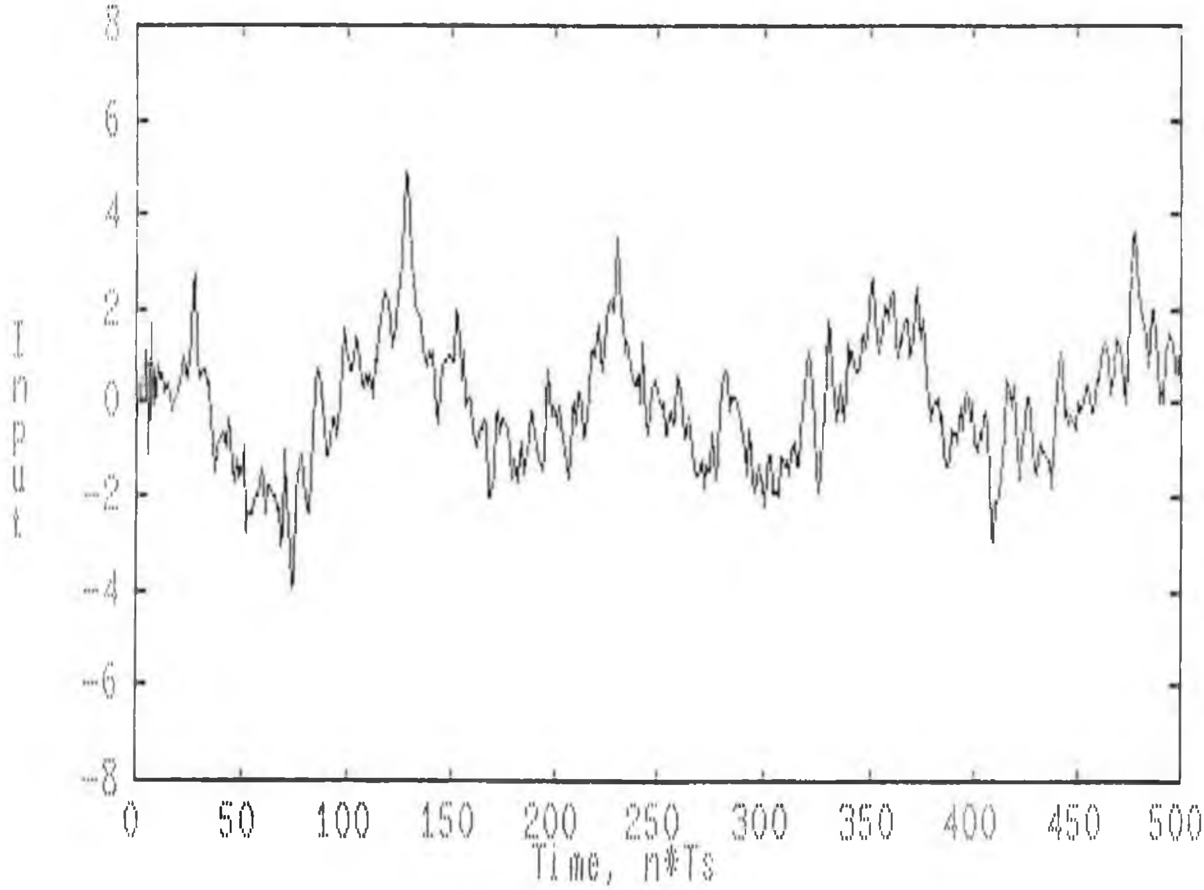

FIG. 4.11(b): Screw Speed Input (Temperature Control)

4.5 CONCLUSIONS

The extrusion process has been identified as a suitable application area for intelligent control. It has many of the elements discussed in chapter 1 which have driven the development of intelligent control. It is a highly complex, nonlinear distributed parameter system which has proven very difficult to model. Successful control requires considerable process knowledge for efficient organisation and integration of the many dynamic elements used in the process. Automatic learning would be required within any control scheme because of the inadequate modelling and nonlinear nature of the process. Hierarchical schemes would be useful and indeed necessary for plant-wide control of the process. The use of an expert system in this respect would also allow heuristic experimental process knowledge to be encapsulated and used for better and more reliable control and fault detection. Improved operator interfaces could also be designed. It is expected therefore that control of this process would benefit from control design within the framework developed in chapter 2.

The extrusion process was described in some detail. The high degree of process knowledge required and the complex interactive multivariable nature of the process could be seen. The distributed nonlinear aspects of the underlying mechanisms were also demonstrated. Typical disturbances affecting product quality and their classification were considered and a brief survey of previous attempts at modelling and control of plasticating extruders was presented.

A particular extruder previously modelled with ARMAX equations was selected and simulated. The application of APFC to the control of temperature and pressure at the die by screw speed manipulation was examined based on these models. The observed results were compared to those found previously for other regulators designed around the same extruder models. Both fixed parameter ARMAX PFC and the adaptive PFC algorithms were found to produce good results.

CHAPTER 5

LOGIC BASED STRATEGY FOR MULTIVARIABLE CONTROL

5.1 INTRODUCTION

Multivariable control design techniques tend to find greatest use in circumstances where the number of inputs equals the number of outputs, i.e. a square system. Typically the specification on control is an extension of the single-input single-output (SISO) case, i.e. each control variable is required to meet strict static and dynamic performance measures. In the operational control of processes where the number of outputs exceeds the number of inputs, the specification on controlled variables will vary. Obviously it is only possible in such a case to tightly regulate as many outputs as there are inputs. This is due to the fact that output function controllability requires that the transfer function matrix have rank equal to the number of outputs. A necessary condition for this is that the number of inputs should be greater than or equal to the number of outputs[164]. Thus, some controlled variables may have a tight regulatory or demand following role while others are specified in terms of an accepable behavioural boundary requiring zone control. Knowledge of the process will dictate the best classification of output variables under the controllability constraint described.

An approach is described here for the control of multivariable systems with more outputs (n) than inputs (m), i.e. limited degrees of freedom arising from the number of inputs or actuators being less than the number of outputs. This situation can arise naturally or a reduced number of actuators may be selected for reasons of economy, reliability, or maintainence reduction. This is utilised for extruder barrel wall temperature control in chapter 6. In addition extra output variables may be employed for safety or quality purposes to provide improved performance using this approach.

Some solutions to this control problem have been proposed in the literature. One defines the control objective so as to keep 'm' of the outputs near their nominal desired values whilst maximising the remaining 'n-m' output variables[165]. Another solution calculates the controls to force the outputs to track given reference signals periodically (i.e. at regular sampling intervals)[166]. The simplest solution is the 'alarm method' in which the additional zone outputs are continuously monitored and corrective action is taken when they enter a danger zone. This however can lead to hard control action. It will also occur after the problem arises or else narrower zone constraint bands will have to be specified reducing efficiency.

As described previously in chapter 1, use of AI and computer science techniques in combination with control theory in a hierarchical architecture can lead to improved control performance. A Logic Based Strategy (LBS) is presented here for multivariable control. In global terms a two-level hierarchy is employed combining the benefits of single-loop or multivariable predictive functional controller (PFC) blocks at the lowest level with a decision logic block at the higher level to offset the degree of freedom deficiency mentioned. The decision logic performs smooth switching between the low-level PFC controllers based on comparison of the computed control signals. As will be explained later this leads to smooth actuator signals and pre-emptive corrective action before a problem occurs. Zone constraints may thus be set at their maximum for greater efficiency. LBS is described in [167,168] and an application to barrel wall temperature control has been presented[169]. The logic based strategy is first developed for the single-input multiple-output case and then generalized to the m-input n-output case. Geometric analysis of its operation in each instance is presented along with simulation results to demonstrate its performance.

5.2 LOGIC BASED STRATEGY

To understand the operation of LBS for control of non-square multivariable systems with more outputs than inputs consider the case of a single-input dual-output plant as shown in fig. 5.1. With this multivariable system one MV affects two outputs, S1 and S2. There is one degree of freedom hence only one output may be tightly regulated. However, the other secondary output is constrained to lie between certain values for reasons of safety, quality, or perhaps economy, and thus requires zone control. The multivariable control problem is to tightly control the primary output while ensuring that the secondary output always remains within its constraints. In fig. 5.1, the primary output S1 is to be controlled at a setpoint C1 and the secondary output S2 is constrained to lie between $C2^+$ and $C2^-$. The structure of the LBS controller to solve this control problem will first be presented. An analysis and explanation of its operation will then be given, followed by some simulation examples.

5.2.1 LBS Controller Structure

The system of fig. 5.1 is treated as two separate single-input single-output systems, i.e.

$$P1: MV \rightarrow S1 \qquad (5.1)$$

$$P2: MV \rightarrow S2 \qquad (5.2)$$

with corresponding models, P1m and P2m respectively, available. Two SISO PFC regulators are then synthesized, one for control of S1 using internal model P1m and the other for control of S2 using internal model P2m. The LBS approach utilises three PFC

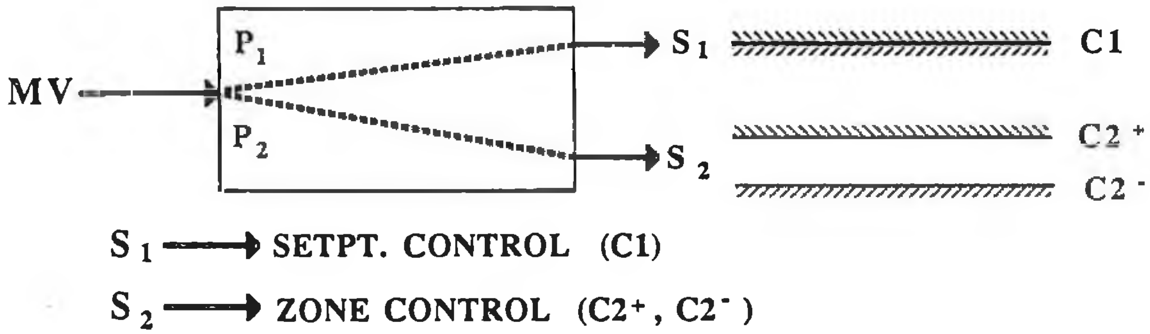

FIG. 5.1: A One-input Two-output Plant

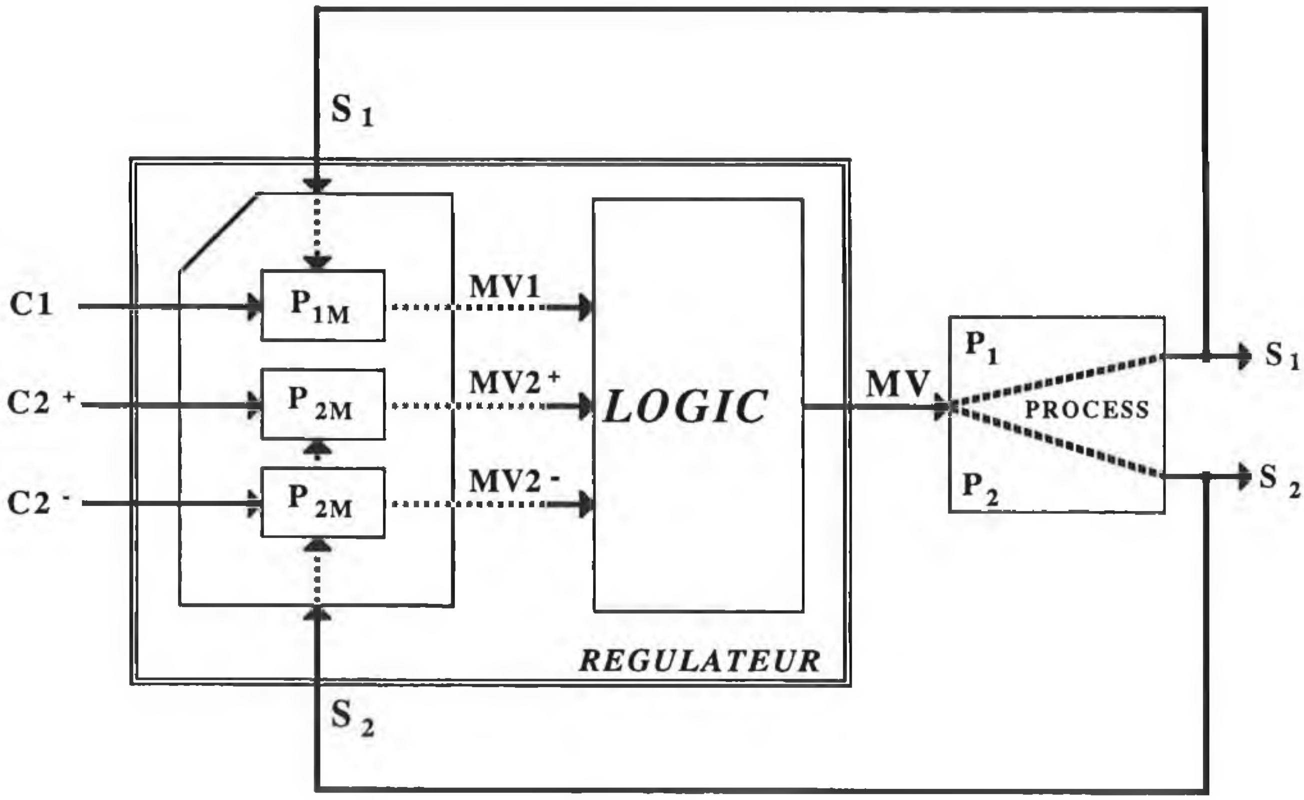

FIG. 5.2: Single-input LBS Controller Topology

regulators in parallel as shown in fig. 5.2. The controller designed around P2m is duplicated but is presented with a different command signal or setpoint (i.e. PFC_b and PFC_c).

Thus at the lowest level, three PFC regulators are operating in parallel. The first, PFC_a, tries to control S1 at setpoint C1 and produces a control signal MV1. It is desirable that PFC_a should always be allowed to operate on the process, while S2 is within its constraints. PFC_b and PFC_c are used to determine when this is not possible. PFC_b produces a control signal $MV2^+$ which attempts to take S2 to its upper constraint $C2^+$ and PFC_c calculates $MV2^-$ which tries to control S2 at its lower constraint $C2^-$. Effectively the output constraints are translated into dynamic constraints on the control signal which may be applied to the process. In this manner they can account for disturbances on the outputs and the dynamics of the process.

5.2.2 Decision Logic

Operating at a level above the three PFC regulators is a logic block which accepts the three computed MV's at each sampling instant and is designed to select the appropriate MV to apply to the process. In the case of systems with only one degree of freedom, i.e. one input as in fig. 5.1, the computed MV's are scalars and the logic is simply to choose the middle scalar value as the applied MV. The reasoning for this is explained below in section 5.2.3. The logic may be expressed simply as:

```
if (MV1>MV2-) and (MV1<MV2+)
      then MV = MV1
else if (MV2->MV1) and (MV2-<MV2+)
      then MV = MV2-
else
            MV = MV2+
```

5.2.3 Geometric Analysis of LBS Operation

As explained above LBS effectively maps the constraints on the secondary output to constraints on the input control signal. Hence, LBS is analysed geometrically on the common input control signal domain. That is, the three command signals (C1, $C2^+$, and $C2^-$) are mapped to a common input domain (MV1, $MV2^+$, and $MV2^-$ respectively) via the PFC regulators as shown in fig. 5.3, i.e.

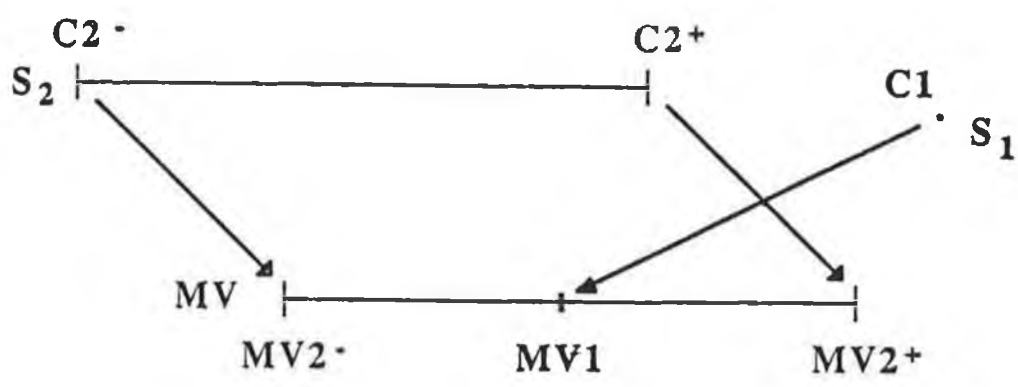

FIG. 5.3: Mapping to Actuator Signal Domain

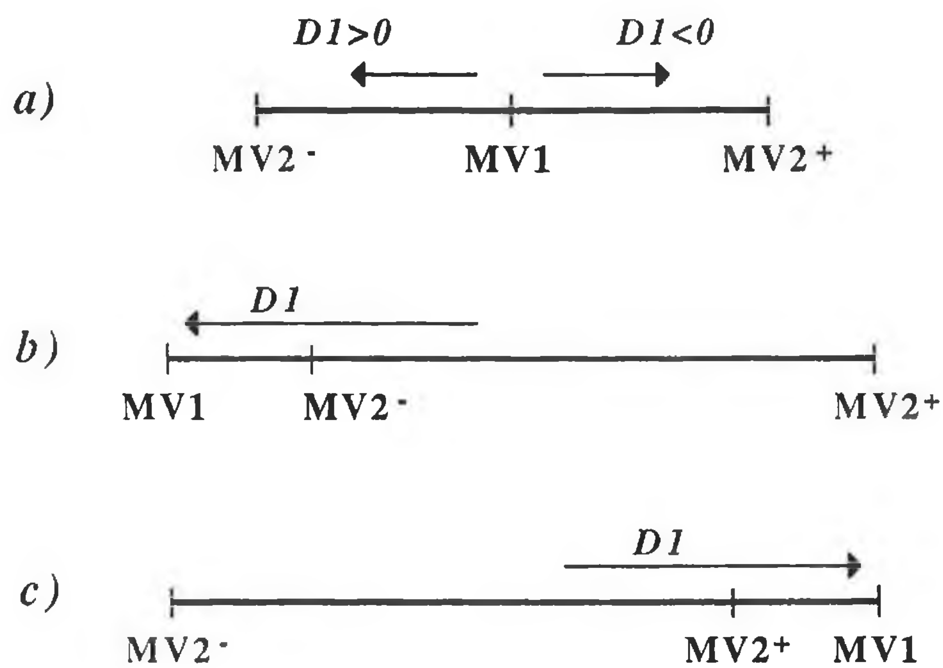

FIG. 5.4: Effects of Disturbances on S1

$$\text{PFC}_a\text{:}\quad MV1 = f(S1,C1) \qquad (5.3)$$

$$\text{PFC}_b\text{:}\quad MV2^{+} = f(S2,C2^{+}) \qquad (5.4)$$

$$\text{PFC}_c\text{:}\quad MV2^{-} = f(S2,C2^{-}) \qquad (5.5)$$

If MV1 lies between $MV2^+$ and $MV2^-$ then this indicates that S1 may be controlled at its setpoint C1 and that S2 will remain between its constraints. Consider the effect of a disturbance, D1, on output S1. As MV1 is a function of S1 (eq. 5.3.), the net result of D1 will be a shift of MV1 along the common input domain to compensate for its effect, as shown in fig. 5.4(a). MV1 will always be chosen while it remains between $MV2^+$ and $MV2^-$ since this implies that S2 will stay within its constraints. As the magnitude of D1 increases it may not be possible to compensate for its effect on S1 while constraining S2. This is signified by MV1 moving beyond the $MV2^+/MV2^-$ boundary on the common input domain as shown in fig. 5.4(b) and (c). In such cases the logic will select a new middle value (i.e. $MV2^-$ in fig. 5.4(b) and $MV2^+$ in fig. 5.4(c)). The MV selected will be such as to control S2 at one of its constraints while keeping S1 as close as possible to its setpoint, i.e. minimum offset. This is demonstrated in section 5.2.5. Analysis of the operation of LBS for disturbances on S2 is exactly the same except that $MV2^+/MV2^-$ move by equal amounts on the common input domain.

The advantages of LBS are obtained because the logic operates on the common input domain and not on the measured process outputs as in the simple 'alarm method' mentioned in section 5.1. One advantage is that the applied MV will always be continuous, i.e. there will be no large jumps in the applied MV. This is because switching between MV trajectories only occurs when they intersect and is thus very smooth. This can be observed by considering the operation of a controller in the time domain as shown in fig. 5.5. Here the effect of a large disturbance, first positive then negative, on S1 is shown. The result is that MV1 moves beyond $MV2^+/MV2^-$ and control is switched at the trajectory intersection. Another result of basing the logic on the control signal is that pre-emptive corrective action is taken. The effect of applying MV1 is considered with respect to keeping S2 within its zone. If this would be invalidated as indicated by a trajectory intersection then control is switched to PFC_b or PFC_c before S2 goes out of zone and is instead brought to its maximum value in a controlled fashion. These results are demonstrated in section 5.2.5.

5.2.4 Extension to 'n' Outputs

LBS is easily generalised to 'n' outputs for the single degree of freedom case, i.e. only one input. Again only one output may be tightly controlled with the remaining n-1 outputs restricted by constraints and thus requiring zonal control. Consider first the formulation of

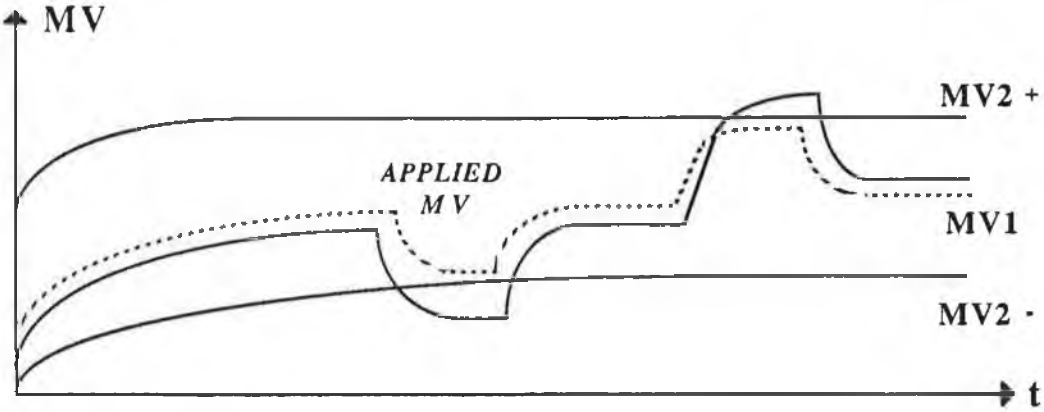

FIG. 5.5: LBS Operation in the Time Domain

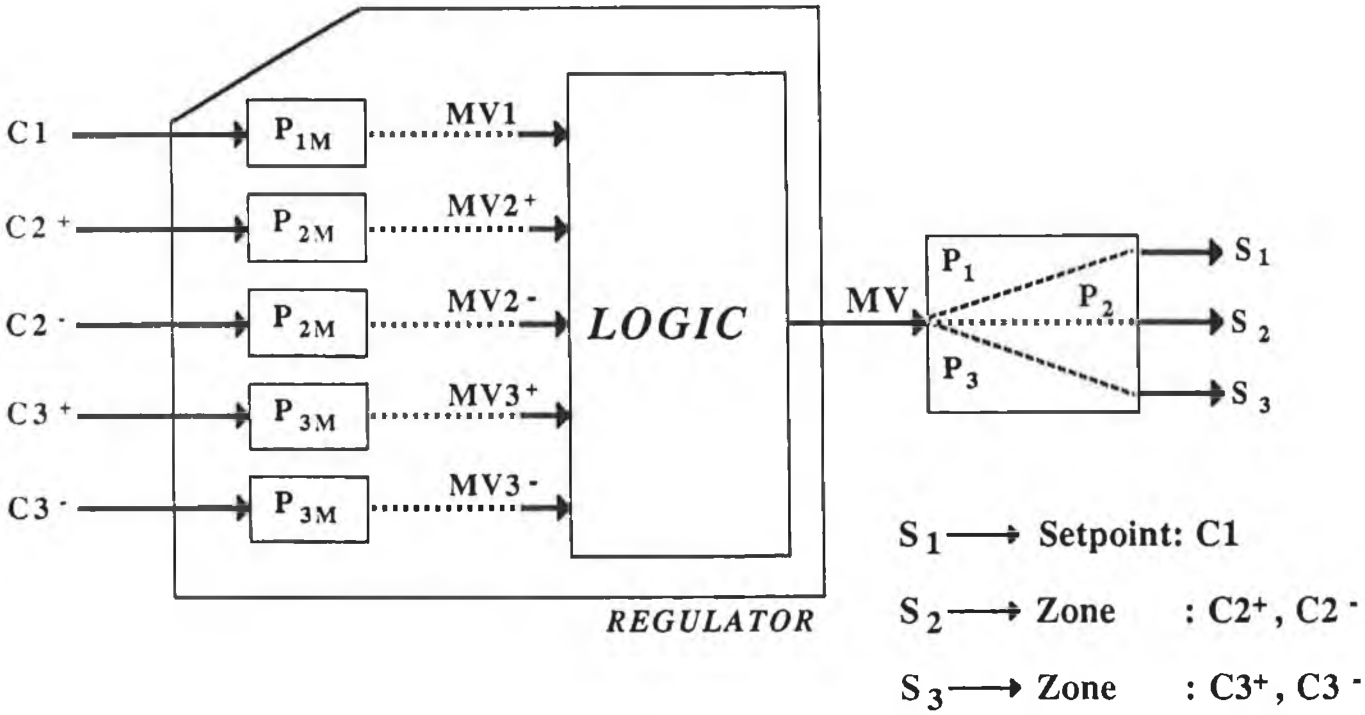

FIG. 5.6: LBS Topology for a One-input Three-output System

the strategy for a one-input three-output system.

As only one output may be controlled at any one time it is necessary to assign priorities to the zone outputs. The highest priority zone will then take precedence over lower priority zones in extreme cases when only one zonal constraint may be met. The structure of the LBS controller is very similar to that of fig.5.2 but with an additional PFC regulator (again duplicated but with different command signals to each) to map the constraints on the third output to the common input domain, see fig. 5.6.

The logic operates in a recursive manner. First MV1 is compared to the MV-pair for the lowest priority zone output with the middle scalar value selected, eg.

$$MV1,\ MV2^{+},\ MV2^{-} \rightarrow v$$

This result (v) is then compared to the next most important zonal MV-pair, eg.

$$v,\ MV3^{+},\ MV3^{-} \rightarrow MV$$

the result of this last operation is applied to the process.

The general case of 'n' outputs (one tightly controlled and n-1 zones) becomes:

- Assign priorities to the zones (the tightly controlled output has lowest priority by default).
- Synthesize 2n-1 PFC regulators in parallel (actually only n distinct regulators are designed).
- Recursively apply the logic to the lowest priority output with the next lowest priority output (select the middle scalar value each time).
- Apply the final result (MV) to the process.

5.2.5 Simulation Results

The operation of the strategy was tested on some simple plants by simulation. The application of LBS to a real example is discussed in chapter 6.

A simple one-input two-output system, as shown in fig. 5.1 was first considered with both P_1 and P_2 described by first-order dynamics and with unity gains. An LBS controller was designed for this system with the individual PFC blocks tuned as discussed in chapter 3. The following setpoint and constraint values were used:

$$C1 = 10 \qquad C2^{+} = 13 \qquad C2^{-} = 7$$

Fig. 5.7A shows the controlled response to a disturbance D1 on S1 while fig. 5.7B shows the response to D2 on S2. The disturbances D1 and D2 are low-pass filtered to simulate realistic disturbances. Plots (a) and (b) show the controlled outputs S1 and S2 respectively, with the corresponding disturbances also shown. The three computed MV trajectories are shown in (d) with the actual applied control signal given in (c). Switching between PFC blocks occurs at the trajectory intersections. The disturbance values are:

Fig. 5.7A:	D1 = step :	-4, $50<t<100$
		+4, $160<t<210$
	D2 = constant:	-2, $0<t<150$
		0, $t>150$
Fig. 5.7B:	D1 = constant:	-1, $0<t<140$
		0, $t>140$
	D2 = step :	-5, $50<t<130$
		+5, $180<t<260$

It may be observed that when the nature of the disturbances are such that S1 may not be controlled tightly with S2 remaining within zone, then LBS switches to control of S2 at an appropriate constraint value. This produces a minimum offset on S1. Pre-emptive action is demonstrated in fig. 5.7A. Trajectories MV1 and $MV2^-$ intersect at t=170 indicating that the control objective may not be met due to the positive disturbance D1. Hence LBS switches to control of S2 at $C2^-$ acknowledging this fact. This occurs before S2 nears its zone limit and so results in smooth control action to take S2 to its limiting value.

The effect of a disturbance on S2 may be sufficiently hard to take S2 out of zone before corrective action may be taken, as can be seen in fig. 5.7B. Tuning the PFC blocks for better performance (and thus harder control) will bring S2 back within zone much faster but with the obvious tradeoffs on stability, control effort, etc. If disturbances are measurable then feedforward compensation may be used to improve the response of the controller. This is demonstrated later.

Fig. 5.8(a)-(d) contains the same information except that P_1 is a non-minimum phase response. The results are as above with LBS operating successfully.

To demonstrate the operation of the strategy when P_1 and P_2 reflect different gains and operating levels, LBS was tested using data from a milk drying plant[]:

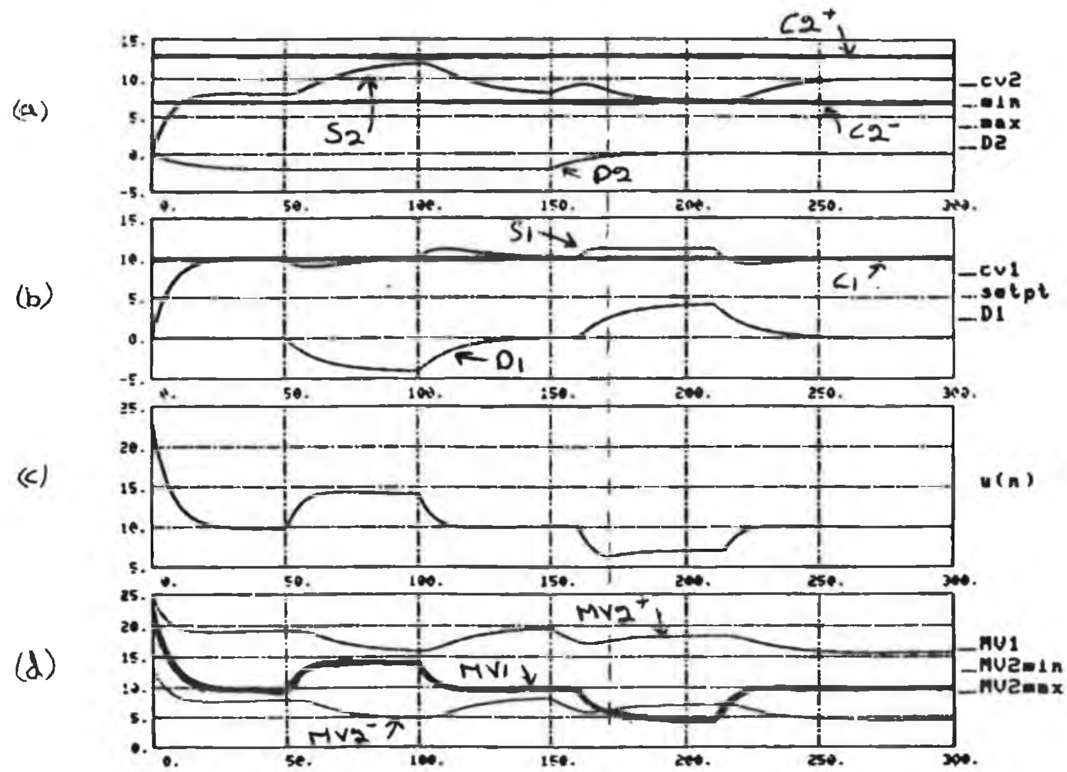

FIG. 5.7A: P_1 - 1st order, P_2 - 1st order - Disturbance D1

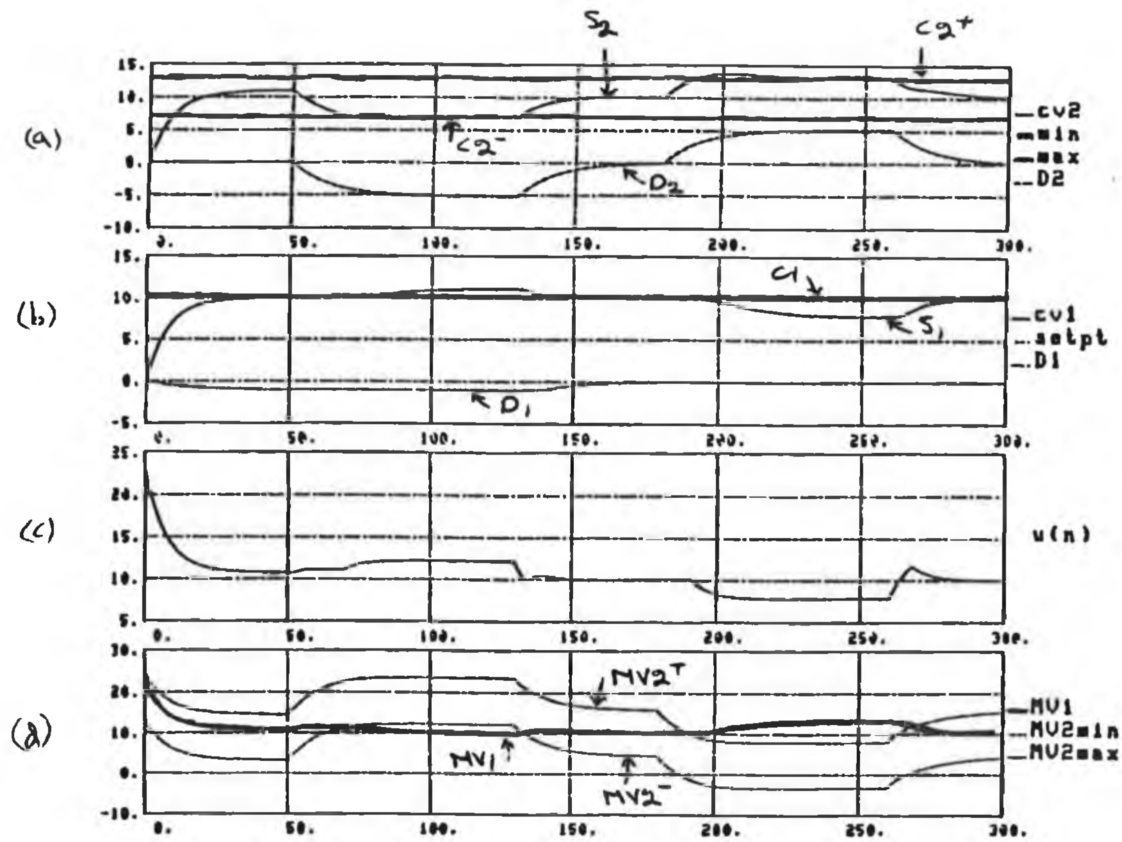

FIG. 5.7B: P_1 - 1st order, P_2 - 1st order - Disturbance D2

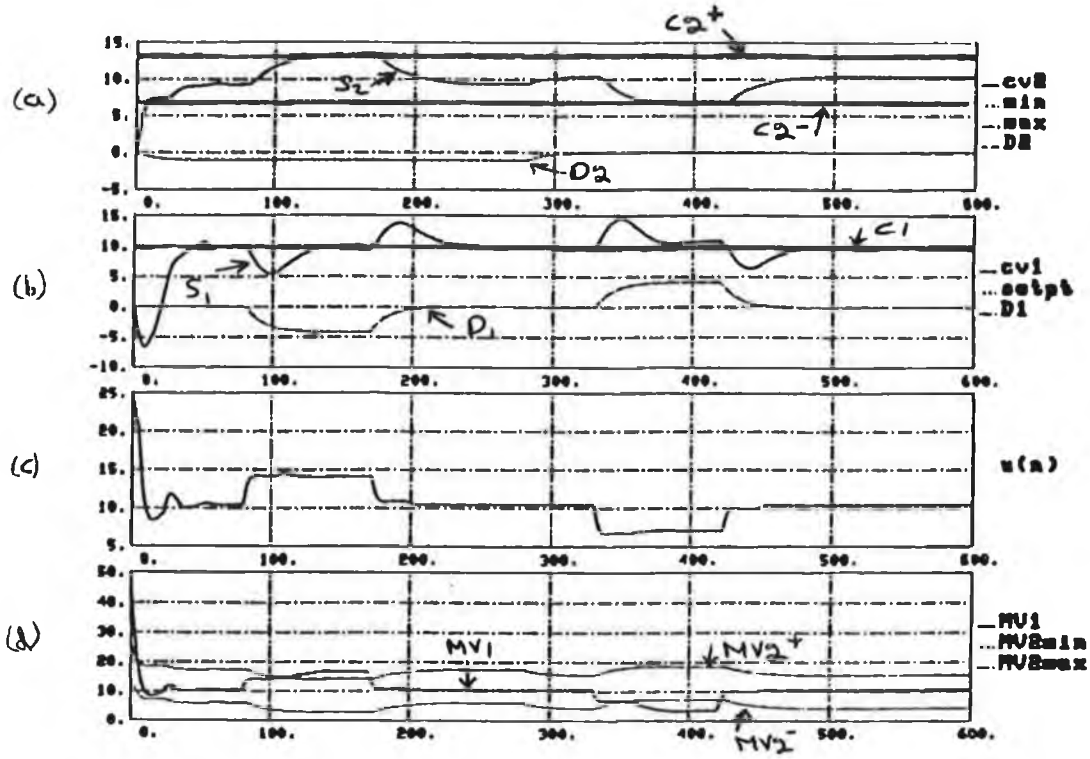

FIG. 5.8: P_1 - 1^{st} order, P_2 - non-minimum phase

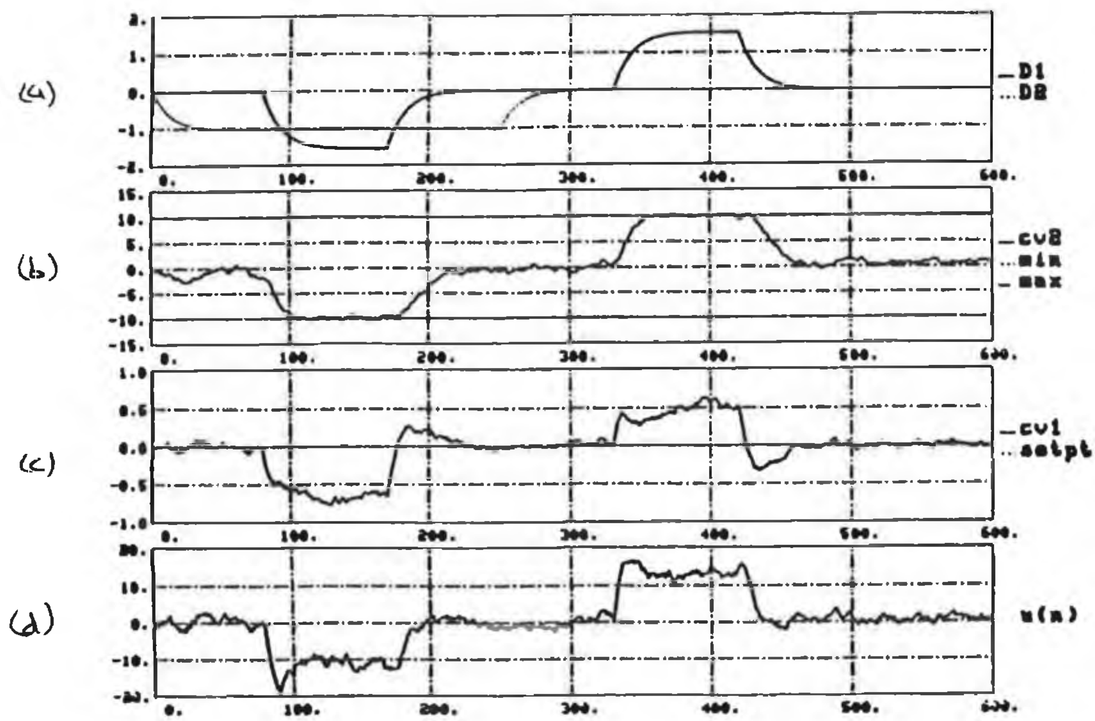

FIG. 5.9: Milk Drying Plant Example

P_1 : 1^{st} order

S.S. gain = -0.08

Output operating level = 3

P_2 : 1^{st} order

S.S. gain = +0.8

Output operating level = 80

Input operating level = 180

Setpoints : $C1 = 3$

$C2^+ = 90$

$C2^- = 70$

Fig. 5.9(b) and (c) show the simulated responses of the system outputs to step disturbances on S1 and S2 as shown in fig. 5.9(a). Note that filtered white noise is added to the process outputs. The applied control signal is given in fig. 5.9(d). The correct operation of LBS may be observed.

The operation of the strategy for a single-input three-output system was also simulated. First-order transfer functions were used to describe the dynamic relationships P_1, P_2 and P_3 (see fig. 5.7). The results are shown in fig. 5.10 in response to a step disturbance on S1. The computed MV trajectories are shown in fig. 5.11. The constraints used are:

$C2^+ = 14$ $\qquad C2^- = 6$

$C3^+ = 13$ $\qquad C3^- = 7$

with setpoint $C1 = 10$

The operation of the recursive logic may be followed by examining fig. 5.11. The first D1 step disturbance (negative) causes LBS to switch to control of S2 at $C2^+$ while the second (positive) D1 causes LBS to concentrate on control of S3 at $C3^-$.

Fig. 5.12 and fig. 5.13 allow the effects of feedforward compensation to be compared. In fig. 5.12, a disturbance (D3) on S3 causes S3 to go out of zone initially due to the nature of D3 (i.e. very hard disturbance). The effects of D3 are eventually compensated for and S3 is brought back to its constraint value by the LBS controller. With feedforward compensation, the PFC blocks can counteract the disturbance as it occurs.

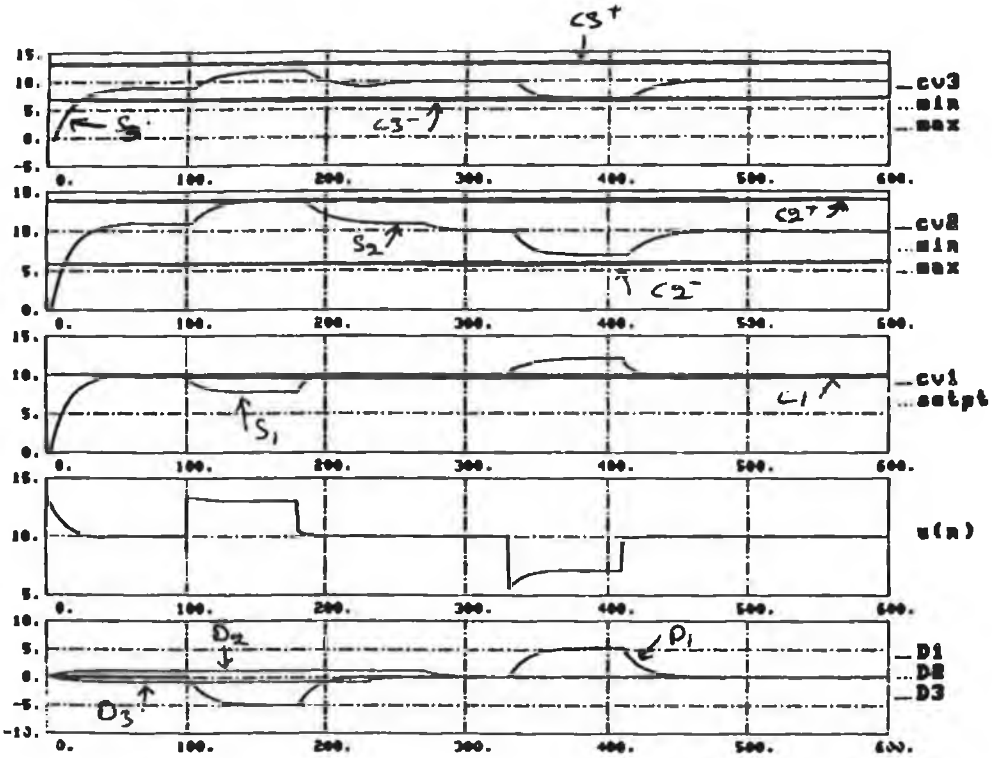

FIG. 5.10: One-input Three-output Example

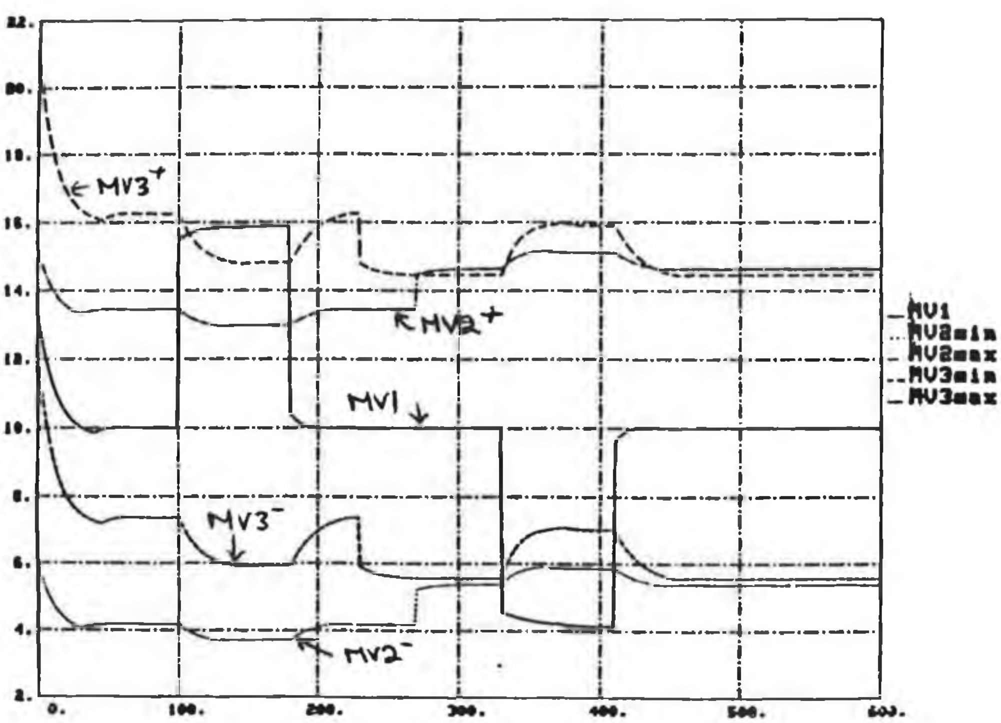

FIG. 5.11: MV Trajectories for example of fig. 5.10

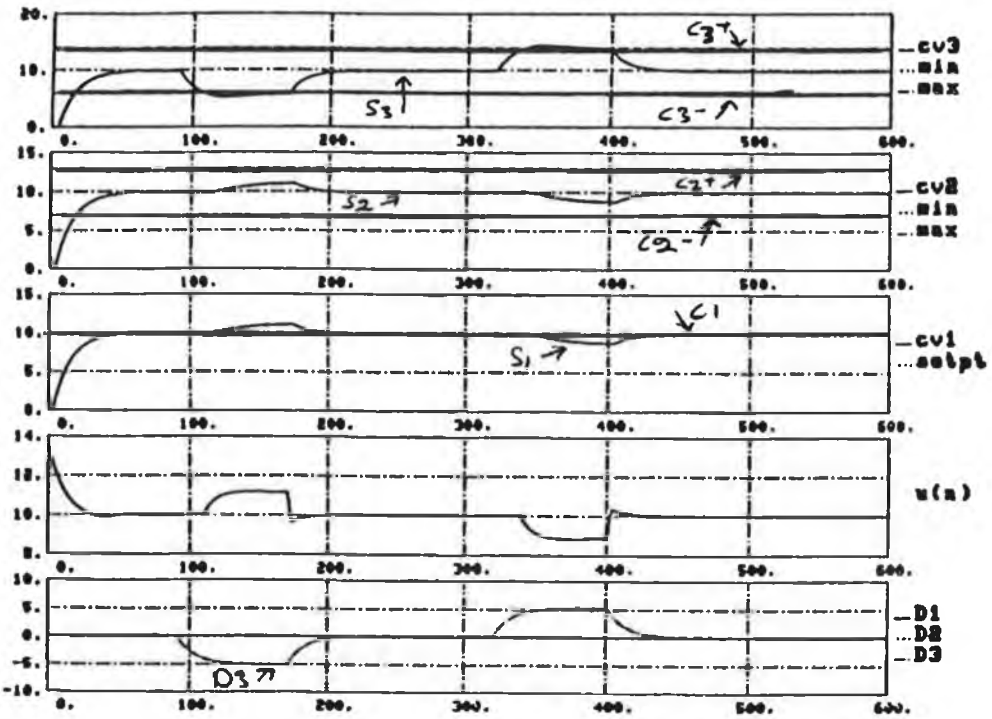

FIG. 5.12: Disturbance D3 on S3 - no feedforward compensation

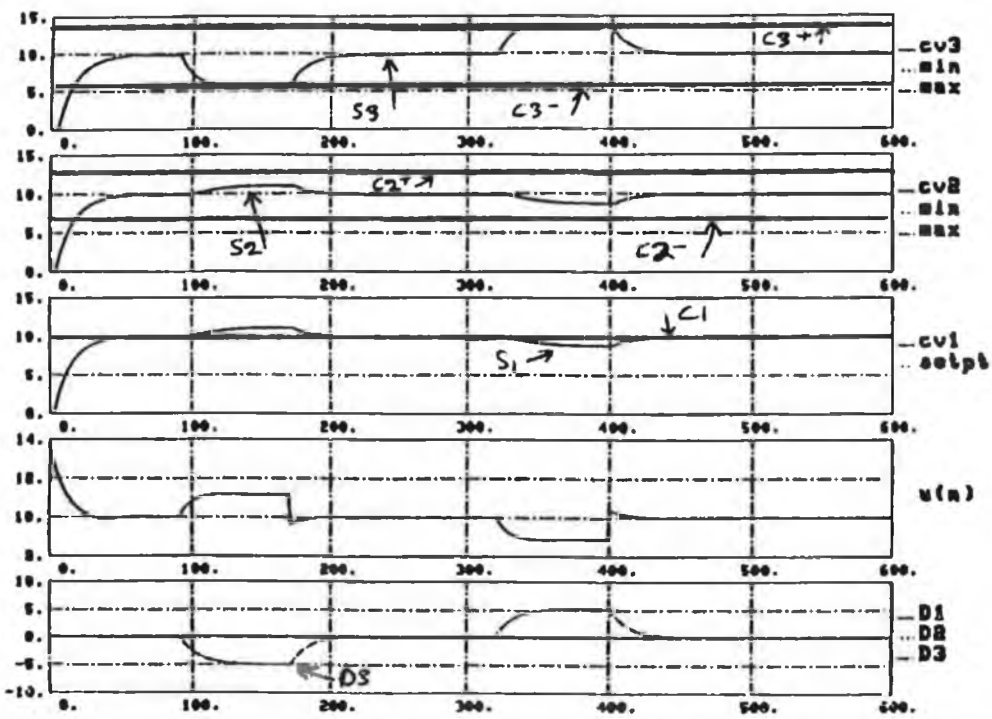

FIG. 5.13: Disturbance D3 on S3 - with feedforward compensation

5.3 MULTIPLE INPUT CASE

The most general form of the logic based strategy will be derived and a geometric interpretation of how it operates will be presented. Systems with 'm' inputs and 'n' outputs (with n>m) shall be treated. First however, a simpler form shall be considered, i.e. a two-input three-output plant as shown in fig. 5.14. With two inputs (MVx and MVy) and hence two degrees of freedom, two of the plant outputs may be tightly controlled (eg. S1 and S2), and the third output constrained within a zone. The control objective is to regulate S1 and S2 at C1 and C2 respectively while keeping S3 within the constraints $C3^+$ and $C3^-$. If disturbances are such that S3 could go out of zone then LBS will control S3 at an appropriate constraint value. But with two degrees of freedom it is also possible to keep tight control of one of the other outputs. There is thus a design decision to be made as to which other output, S1 or S2, should be controlled with S3. This choice will depend on the particular application and other criteria such as economics, safety, etc. It is assumed here that S1 should always be tightly regulated.

5.3.1 Controller Structure

The structure of the LBS controller is shown in fig. 5.15. It is similar to that of the single-input case in fig. 5.2 except that each PFC block is a multivariable 2x2 regulator. PFC_a is designed to control S1 and S2 at C1 and C2 respectively:

$$MVA = f(S1, S2, C1, C2) \qquad (5.6)$$

where MVA is a two element vector of plant control signals. PFC_b and PFC_c are exactly the same except that the former tries to control S1 at C1 and S3 at $C3^+$, and the latter tries to take S1 to C1 and S3 to $C3^-$, i.e.

$$MVB = f(S1, S3, C1, C3^+) \qquad (5.7)$$
$$MVC = f(S1, S3, C1, C3^-) \qquad (5.8)$$

where MVB and MVC are also two element MV vectors. As before the higher-level logic block selects the appropriate MV vector to apply to the plant.

5.3.2 Decision Logic

The logic strategy is again to compare the three computed MV vectors and to apply the "central" vector. That is MVB and MVC define the constraints on the MV vector that

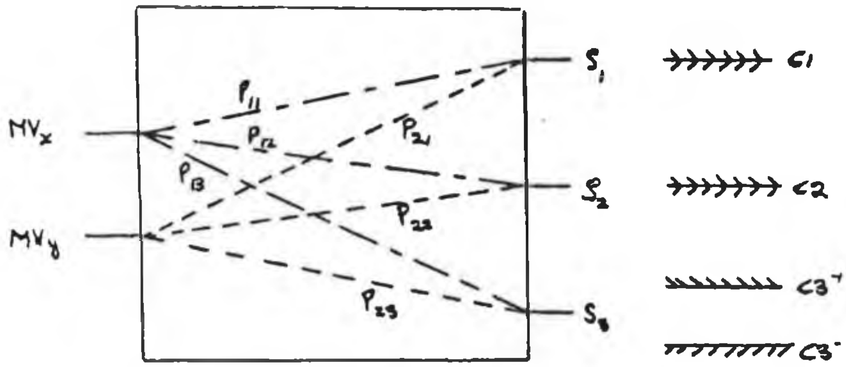

FIG. 5.14: Two-input Three-output System

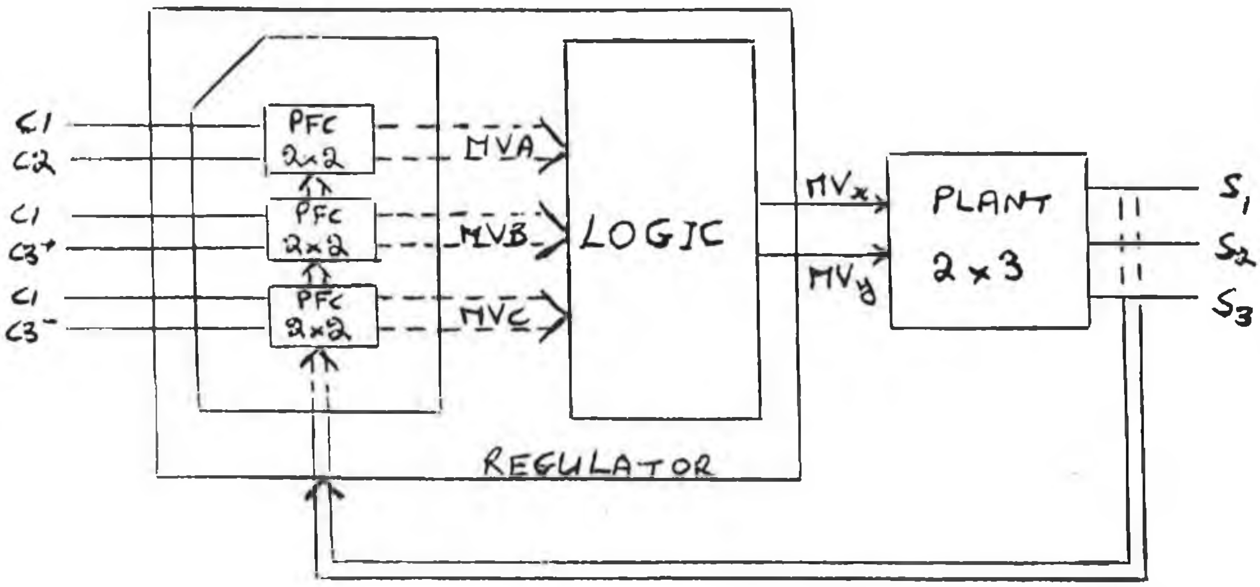

FIG. 5.15: LBS Topology for 2x3 System

may be applied to the plant. If MVA lies 'between' these then its use will ensure that S1 and S2 may be tightly controlled with S3 within zone. If this is not the case, then the central vector chosen (i.e. MVB or MVC) will control S1 at C1, S3 at one of its constraints, and S2 will have a minimum offset from its setpoint value C2. A problem arises in determining how to compare three vectors and to establish the "central" one. It will be shown geometrically in the next section that this problem in fact reduces to a simple scalar comparsion.

5.3.3 Geometric Analysis

The analysis is an extension of the single-input case but is now performed on a two-dimensional common input domain (MVx, MVy). Consider the steady state characteristics of the plant:

$$S1 = k_{11}.MVx + k_{21}.MVy \qquad (5.9)$$
$$S2 = k_{12}.MVx + k_{22}.MVy \qquad (5.10)$$
$$S3 = k_{13}.MVx + k_{23}.MVy \qquad (5.11)$$

where k_{ij} is the steady state gain between the i^{th}-output and the k^{th}-input.

The locus of control vectors (MVx, MVy) such that S1 is controlled at C1 is a line (L1) on the two-dimensional common input domain, i.e.

$$L1 : \quad MVy = -\frac{k_{11}}{k_{21}}.MVx + \frac{C1}{k_{21}} \qquad (5.12)$$

with a slope of $-k_{11}/k_{21}$ and an offset $C1/k_{21}$.

Similarly, the locus of control vectors to keep S2 at C2 is the line:

$$L2 : \quad MVy = -\frac{k_{12}}{k_{22}}.MVx + \frac{C2}{k_{22}} \qquad (5.13)$$

Also, for control of S3 at $C3^-$:

$$L3 : \quad MVy = -\frac{k_{13}}{k_{23}}.MVx + \frac{C3^-}{k_{23}} \qquad (5.14)$$

and S3 at $C3^+$:

$$L4 : \quad MVy = -\frac{k_{13}}{k_{23}}.MVx + \frac{C3^+}{k_{23}} \qquad (5.15)$$

Loci L3 and L4, corresponding to control of S3 at its minimum and maximum constraints respectively, are parallel lines as shown in fig. 5.16. The plane P bounded by L3 and L4 is the locus of all control vectors such that S3 will remain within zone. Hence, the applied MV vector must lie on the plane P.

There is a clear relationship between fig. 5.16 and the single-input case represented in fig. 5.3. Lines L3 and L4 are analagous to the points $MV2^-$ and $MV2^+$ respectively and the plane, P, of allowable control vectors corresponds to the line $[MV2^-, MV2^+]$ for the single-input case.

L1 and L2 have different slopes and offsets, as shown in fig. 5.17. Their point of intersection defines the unique control vector which controls both S1 at C1 and S2 at C2, i.e. the computed vector MVA from PFC_a. Similarly, MVB corresponds to the intersection of L1 and L4 (i.e. S1 at C1 and S3 at $C3^+$) and MVC to the intersection of L1 and L3 (i.e. S1 at C1 and S3 at $C3^-$). Thus at each sampling instant the three computed MV vectors are colinear. The decision logic to select the central vector reduces then to exactly as before, i.e. a scalar comparsion between either the x-components or the y-components of each of the three vectors. The central point is chosen and the vector which this corresponds to is applied to the process.

The effect of disturbances on the outputs must be considered. Equations 5.12-5.15 indicate that the slopes of the loci L1-L4 depend only on the steady state gains of the multivariable process and are independent of disturbances on the process outputs. However, the offsets of L1-L4 will vary corresponding to the magnitude of the disturbances with the offset of L3 and L4 changing by equal amounts. Hence, analagous to the single-input case, three possible scenarios may occur as shown in fig. 5.18. If the disturbances are small then MVA should remain between L3 and L4, i.e. between MVB and MVC as in fig. 5.18(a). Here application of MVA would achieve the desired objective. For the case of disturbances such that the situations in fig. 5.18(b) and (c) occur, application of the middle vector ensures that S1 is controlled at C1, S3 is held at one of its constraints, and S2 is at a minimum offset from C2.

5.3.4 Generalisation to Multiple Input Systems

Generalisation of LBS for an m-input n-output system with n>m is straight forward. 2n-m PFC regulators must be synthesized with each PFC block designed as an mxm regulator. m-outputs may be tightly controlled with zonal constraints placed on the remaining n-m outputs. Each of the zonal outputs must be prioritised and the decision logic applied recursively as described in section 5.2.4.

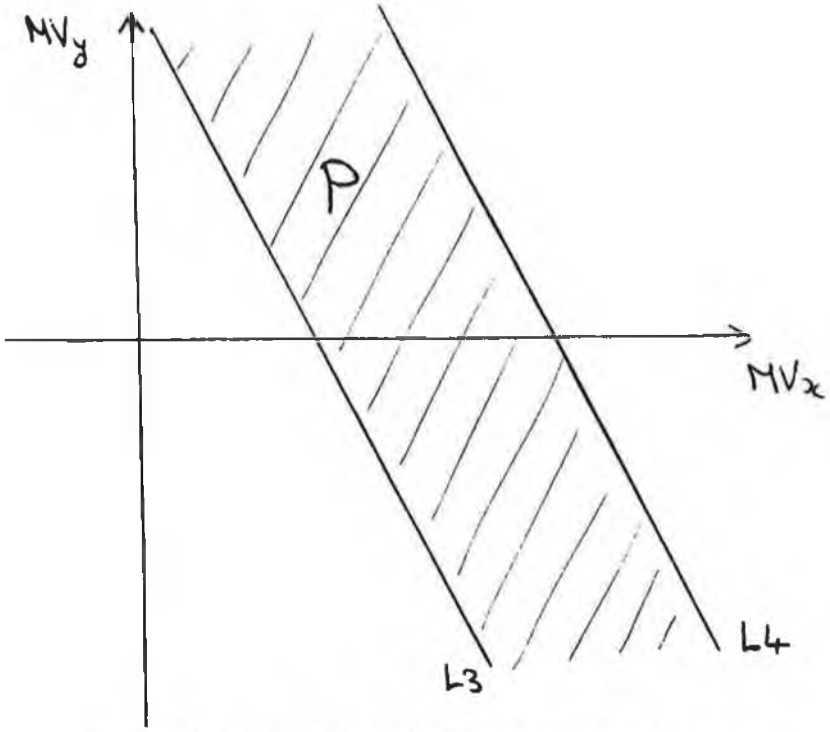

FIG. 5.16: Plane P of Allowable Control Vectors

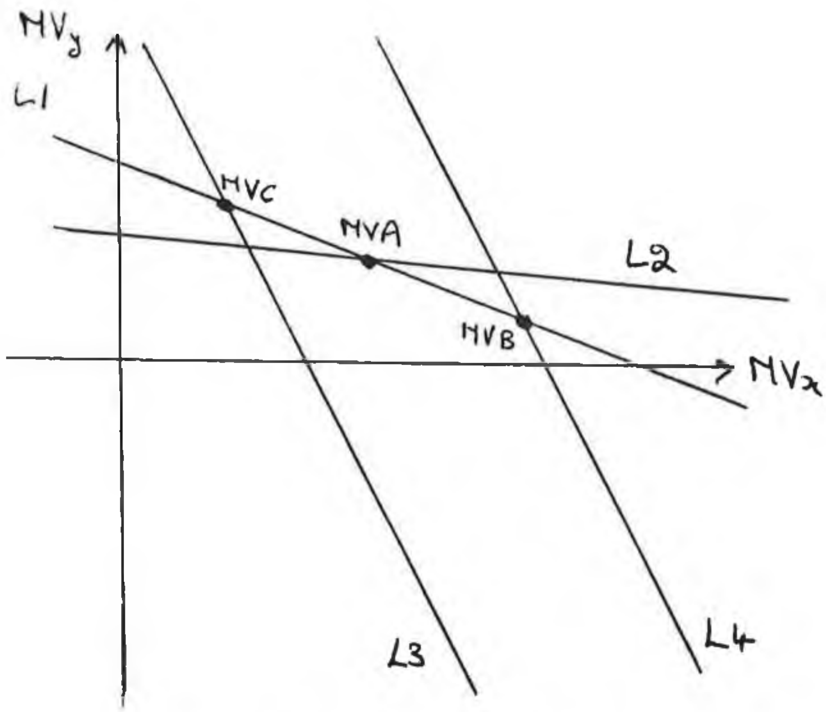

FIG. 5.17: Colinearity of MV Vectors

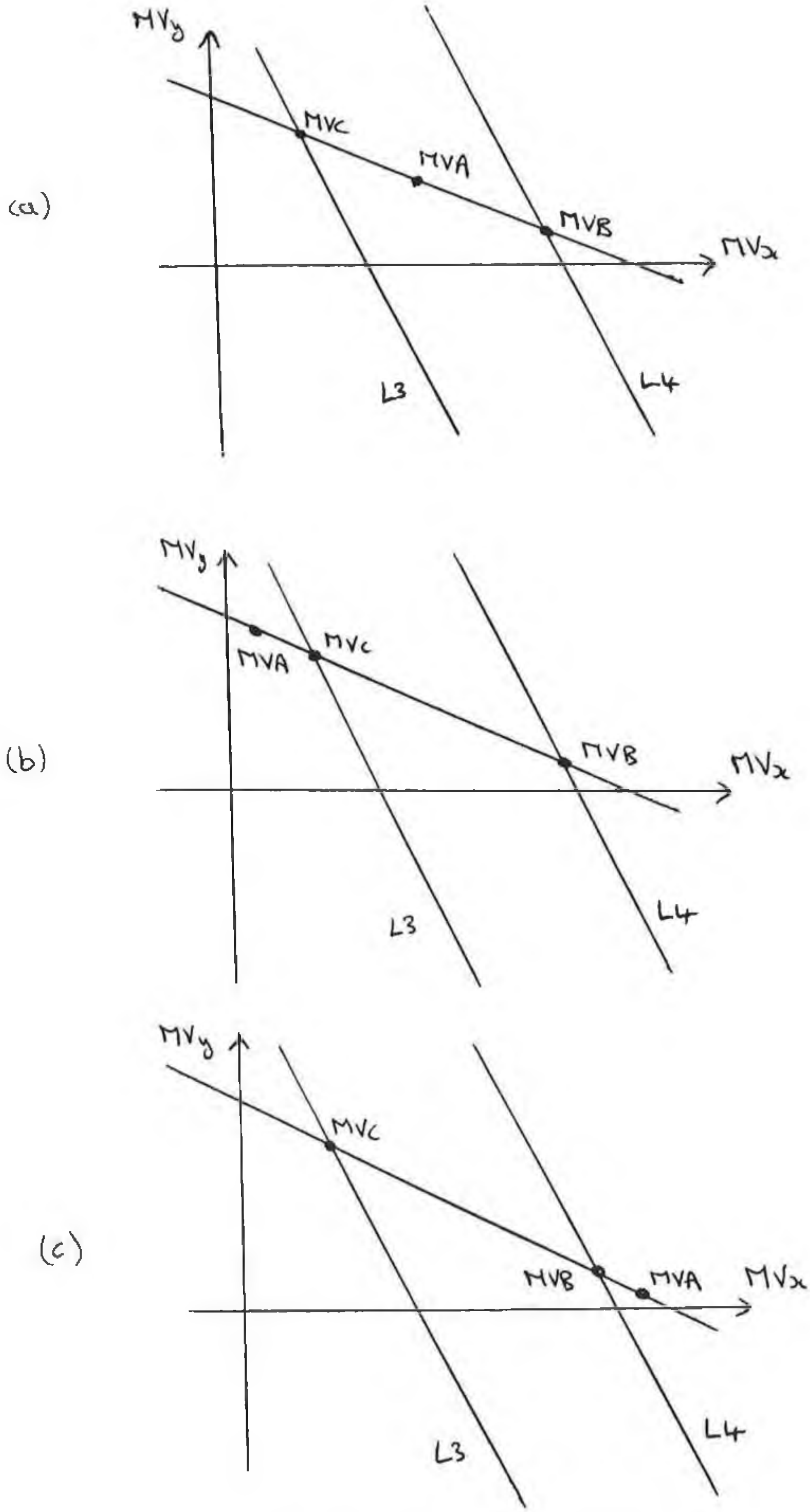

FIG. 5.18: Effects of a Disturbance on S1

Analysis of its operation may be performed on an m-dimensional hyperspace representing the common input domain of actuator signals. In an analagous fashion to the two-dimensional case presented above, L3 and L4 become m-1 dimensional surfaces parallel to each other. The locus of allowable control vectors is the space between L1 and L2. As before the decision logic operates on three colinear vectors and hence can be implemented as scalar comparsions of one of the corresponding elements of each vector.

5.3.5 Simulation Results

The multiple input strategy was tested on a two-input three-output system with the transfer function representation as shown in fig. 5.19. This system has both nonminimum phase, second- order (under and over-damped), and first order elements. A 2x3 LBS controller (as shown in fig. 5.15) was designed for this system. The control objective is to control both S1 and S2 at setpoint values while always ensuring that S3 remains within the constraints. If this is not possible then control on S2 should be relaxed and S1 and S3 should be controlled (i.e. S3 regulated at one of its constraint values).

The constraints and setpoints used were:

$$C1 = 2$$
$$C2 = 6$$
$$C3^{+} = 2$$
$$C3^{-} = -2$$

The effects of step disturbances (D1, D2 and D3) on all three outputs were investigated. Plots of these results are presented in fig. 5.20(a)-(c). Plot (a) shows the response to D1 with (b) and (c) showing the responses to D2 and D3 respectively. In all three cases S1 is regulated at its setpoint C1 at all times. Similarly, S2 is controlled at C2 while S3 remains within it's bounds. If S3 should violate it's constraints then control switches to regulate S3 at it's nearest boundary and control on S2 is relaxed. This results in an output offset on S2 from its setpoint value. This offset is the minimum value achievable in the circumstances.

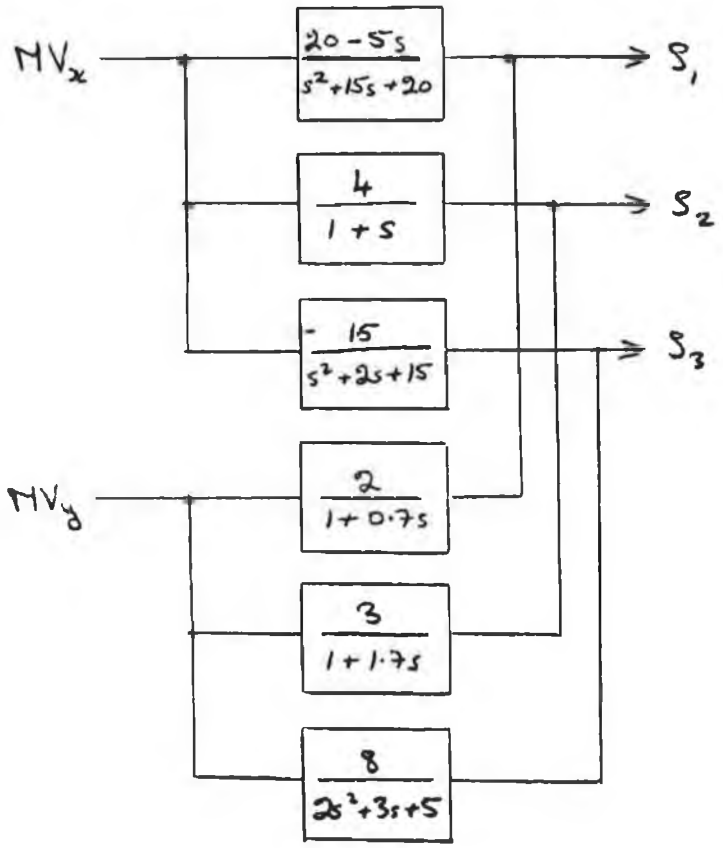

FIG. 5.19: A Two-input Three-output Example

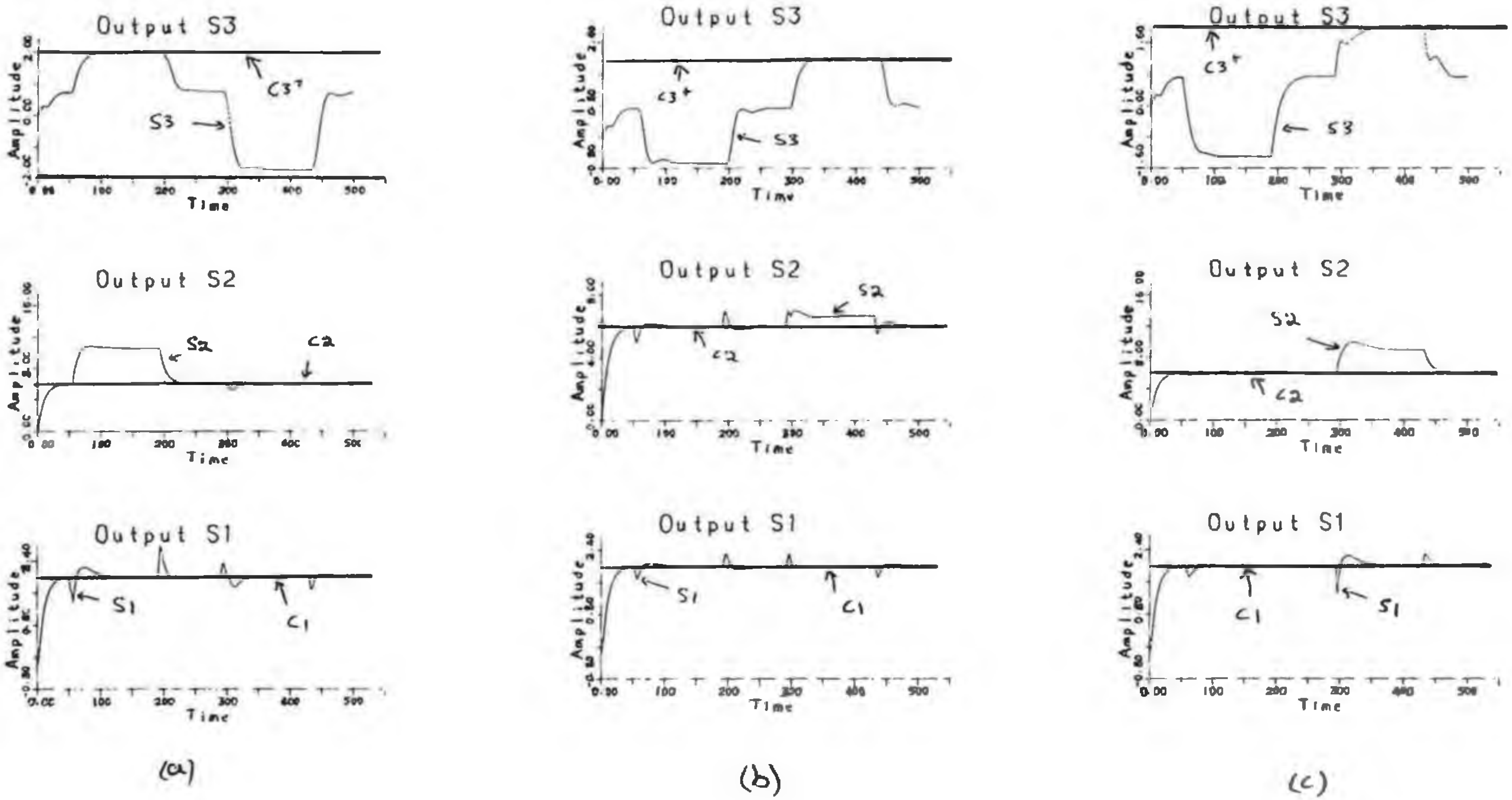

FIG. 5.20: Control Results for Plant of fig. 5.19

5.4 CONCLUSIONS

LBS was presented as a new multivariable control strategy for control of non-square multivariable systems with more outputs than inputs. It is a two-level hierarchical structure combining the benefits of predictive functional control with decision logic, a software tool, superimposed to offset the degree of freedom deficiency. At the lower level several PFC controller blocks designed for control of square systems are implemented in parallel. At the higher level, the decision logic block operates on the computed MV's generated by the PFC blocks. The logic switches the different controllers in and out as appropriate.

A geometric interpretation and analysis of the operation of LBS was presented for both the single-input and two-input cases. It was shown how to extend this analysis to m-input systems using an m-dimensional hyperspace. Simulation results where also presented to demonstrate the performance of the strategy.

Since the logic does not operate on the measured outputs but on the calculated control signals, LBS can respond to possible problems before they take effect on the system outputs. That is, pre-emptive corrective action is taken. Also for this reason, smooth changeover between controllers is obtained because switching only occurs at the intersection of MV trajectories as shown in simulations. This results in the desirable feature of smooth actuator control signals applied to the process. A disadvantage of the strategy is the online computational overhead of two additional PFC regulators per extra zonal output. However, only one regulator need be designed and this is then duplicated. The computational overhead may be reduced as a result of this duplication because some calculations may be combined. A big advantage of the strategy is that the decision logic is quite simple and only requires scalar comparisons regardless of the rank of the controlled system. LBS may be employed with any control law design used in conjuction with the decision logic although it is preferable to use a predictive controller for the reasons discussed in chapter 2.

CHAPTER 6

LBS - AN APPLICATION

6.1 INTRODUCTION

An important class of lumped multivariable control system arises where actuation and measurement is limited as in distributed parameter control. In systems of this nature controlled variable profile is important. Typical control approaches to distributed systems require a distributed control strategy with distributed observation. These are difficult to implement with crucial stability and robustness issues[170,171]. Distributed observation is a serious disadvantage although some solutions to this problem for flexible structures have appeared[172]. An alternative approach is to consider shape or profile control using multipoint controllers with multipoint observations. Standard single-loop or multivariable regulators may then be employed. This approach has been used for dynamic shape control of a flexible beam[173,174]. Issues which derive from this method of control include selection of the position and number of actuators and sensors to be used[173,175]. Profile control however is constrained by the number of actuators employed. Minimisation of the number of actuators is desirable and is motivated by economic, reliability, and maintenance reduction considerations. This form of Distributed Actuator Control (DAC) involves several different components. DAC requires the following choices and design decisions:

- ° number of actuators employed
- ° location of actuators
- ° selection of appropriate measurement points
- ° feedback topology
- ° controller design

Examples of this problem include profile control in sheet metal, paper, and plastics production. Flexible beam control is another example as is temperature gradient control in furnaces and extruders. The application of LBS in combination with DAC to the control of extruder barrel wall temperature profile is considered in this chapter.

The extrusion process was described in some detail in chapter 4. In summary, solid feed in a pellet or powder form is fed into the barrel via a hopper, as shown in fig. 4.2. This is transported along the barrel by a screw and is heated and pressurised. Typically melt pressure and temperature at the die are used to reflect product quality and are controlled by screw speed and valve restriction as in chapter 4. The feed is melted both by friction as it travels along the barrel and also through heat transfer from the heated barrel wall[144].

Primary disturbances result from feed inconsistencies[144] although many other factors affect product quality. One such factor is the barrel wall temperature profile which is important for safety reasons as well as for quality purposes. Because of this most extruder control systems incorporate barrel wall temperature controllers in addition to the primary control loop for melt temperature\pressure at the die[145,147,148,156].

Barrel wall temperature profile control is a distributed parameter problem[144,176]. The dynamic characteristics are described by partial differential equations. However for control purposes the barrel is usually approximated by several zones with individual regulation of each zone by a heater\thermocouple pair. A lumped parameter model is used to approximate the response of each zone. It has been shown that the response, i.e. barrel wall temperature to heater power input, can be accurately approximated by a lumped first-order model with deadtime[145,157,177]. As the deadtime can be quite large the use of predictive control becomes even more attractive. Typical extruders employ four or more zone heaters.

6.2 LUMPED EXTRUDER DESCRIPTION

The simulation results presented here are based on a 2 1/2" single screw extruder with L\D ratio of 25. The product being produced is polyethelene. The barrel is divided into four heating zones as shown in fig. 6.1. The lumped models used for each zone are those presented in [177]. These were obtained experimentally by open loop step tests. The temperature in each zone is measured by a thermocouple embedded in the wall of the extruder and the heater load is manipulated by a time proportioning technique, i.e. the 'percentage time on' (%ton) may be controlled.

As stated in [177] the optimum barrel wall temperature profile for producing polyethelene with this extruder is:

$$T1 = T2 = 170\ ^{\circ}C$$

$$T3 = T4 = 180\ ^{\circ}C$$

The models used are all first-order with deadtime, i.e.

$$T(s) = e^{-sT} \cdot \frac{K}{1 + s\tau} \cdot H(s)$$

where $T(s)$ = barrel wall temperature ($^{\circ}C$)

$H(s)$ = heater power input (%ton)

K = steady state gain ($^{\circ}C$/%ton)

T = deadtime (sec)

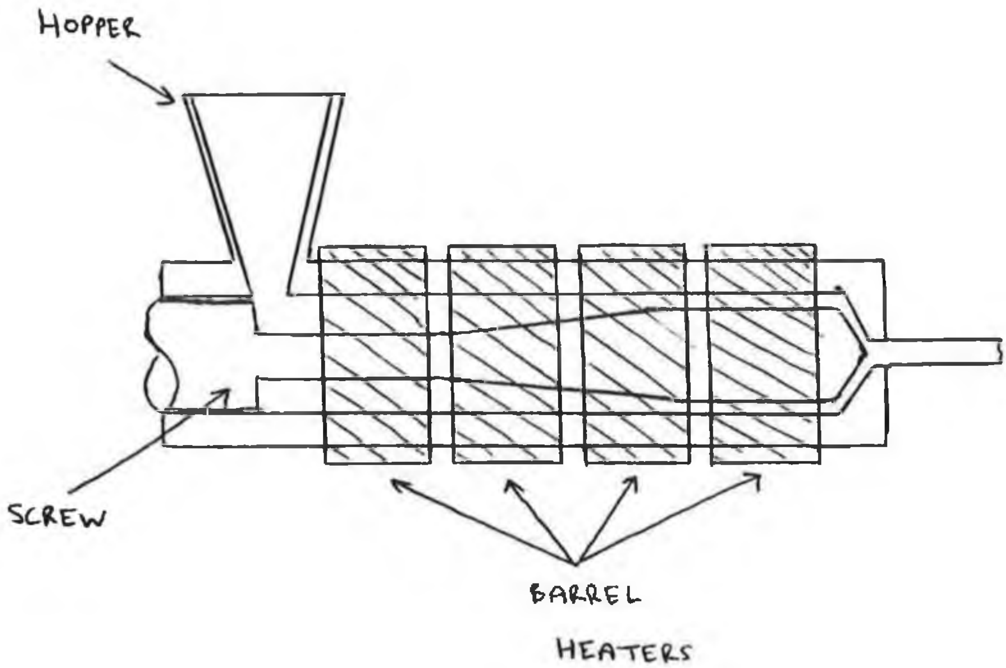

FIG. 6.1: Barrel Heating Zones

τ = time constant (sec)

Although the temperature profile given above is the optimum to produce polyethelene of a certain quality, the temperatures quoted can have acceptable tolerances associated with them. The following facts are applicable to each of the four zones.

Zone 1

This zone is a preheating stage corresponding to the operational feed section of the extruder. The objective of this zone is to sufficiently heat the cold input feed so that constant temperature control of the melt in zone 2 may be achieved more easily. Zone 1 can thus suffer from large variations in the temperature of the cold input feed as the feed may be taken from different storage tanks. Typically the feed is stored at room temperature. The minimum value acceptable for T1 is not critical although it should be high enough to ensure that the melt is sufficiently heated and does not remain solid. This could cause the flights of the screw to be damaged. There is however a highly critical upper limit on T1. Too high a temperature could cause the feed to melt too quickly and result in the phenomenon called *meltback*. This is hot melt flowing *back* up through the hopper and should be avoided at all costs.

Zone 2

T2 is a measurement of the barrel wall temperature at the point just before the screw widens. It is preferable to have this point tightly regulated because a temperature gradient is desired between zones 2 and 3.

Zone 3

This zone measures the barrel wall temperature at the point where the screw has just widened to its maximum. This must be kept above the temperature T2 to ensure a temperature gradient between T2 and T3 is maintained. There is also a lower safety limit on this temperature because the melt cannot be allowed to cool and solidify. This could cause damage to the screw flights. There is an upper limits on T3 for both quality and control purposes. It must be ensured that the melt does not overheat. If this happens the polymer chains may break down with the result that the melt is no longer a polymer. Also, if the melt is heated too much it may not be sufficiently viscous to ensure good compression and hence maintain the pressure of the die.

Zone 4

T4 is a measurement of the barrel temperature in the metering section of the extruder near the die. It is important that this is regulated at a fixed level so that the temperature of the melt at the die may be easily maintained. As mentioned previously, this condition is important to ensure good product quality.

6.3 ACTUATOR MINIMISATION

From the previous discussion it may be seen that this control problem fits the category of industrial processes discussed in section 4.1. That is, some outputs require tight regulatory control while others may be allowed to move within zonal constraints. It is thus quite amenable to the application of LBS for actuator minimisation. That is, the use of LBS would require less actuators than presently employed (four in this particular application). This would result in benefits in costs and reliability without compromising the final product quality or safety.

DAC requires the number and location of actuators to be chosen. From the discussion of section 6.2, two of the outputs required tight regulation (i.e. T2 and T4) while zone control is acceptable for the remaining two outputs. The minimum number of actuators with LBS is thus two. Selection of actuator positioning is best performed following current design practice for heater positioning, i.e. keeping actuators in the same position as in fig. 6.1. Thus, the decision becomes one of selecting which of the four heaters to eliminate.

Performing a steady state analysis will show that the only valid heater positions to achieve the desired temperature profile are as shown in fig. 6.2. This results in the two-input four-output system with dynamic cross-coupling as in fig. 6.3. The parameter values presented in [177] were employed in this model. Coupling across zones must also be considered to design a multivariable controller. These cross-coupling models are also first-order with deadtime. The model parameters are presented in table 6.1. Because of the transportation of the melt down the barrel, heaters have greater influence on downchannel zones than on previous upchannel zones. Similarly, deadtime is less for downchannel coupling than for upchannel coupling.

The open-loop steady state point temperatures are presented in fig. 6.4 with the profile as in fig. 6.5. The achievable steady state temperature profile is thus:

T1 = 169.4°C T2 = 170.0°C T3 = 177.6°C T4 = 180.0°C

6.4 DISTURBANCES AND MEASUREMENT NOISE

To study the performance of LBS for barrel wall temperature control realistic disturbances should be modelled for simulation. The four main disturbances applicable to this problem are[144]:

(1) Constant low temperature cold feed input (usually at room temperature). This exhibits

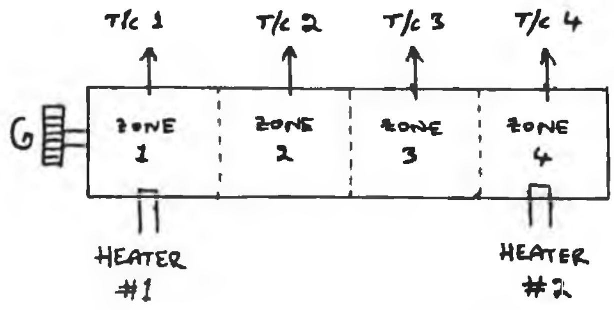

FIG. 6.2: LBS Actuator Positioning

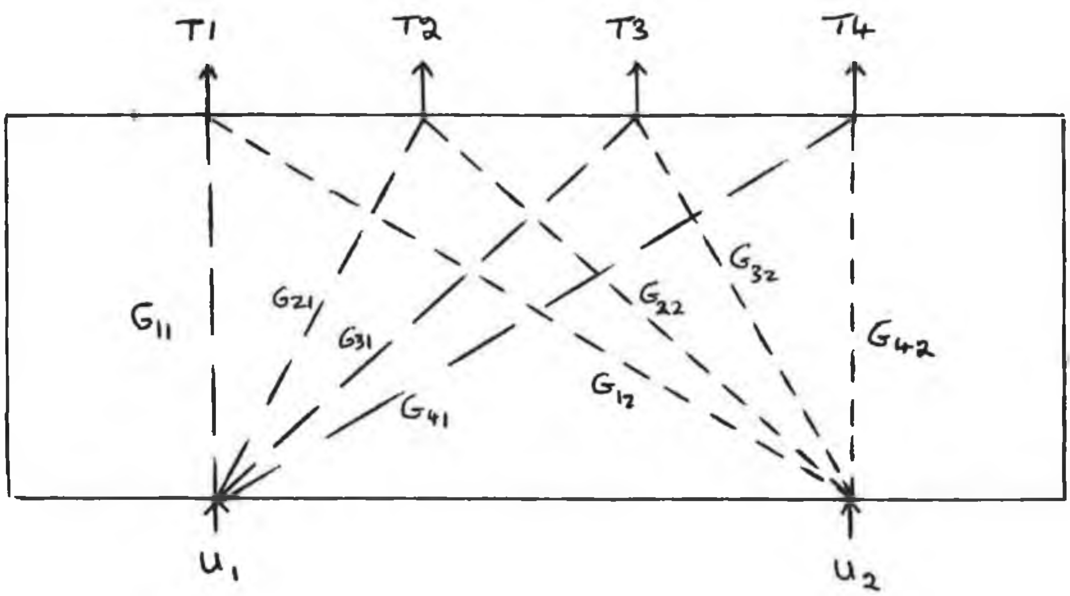

FIG. 6.3: Dynamic Model of Multivariable System

Transfer Function	G_{11}	G_{21}	G_{31}	G_{41}
Gain, K (°C/%ton)	4.4	3.8	1.6	0.5
Time Const., τ (sec)	78	112	122	136
Deadtime, T (sec)	150	210	230	260

Transfer Function	G_{12}	G_{22}	G_{32}	G_{42}
Gain, K (°C/%ton)	0.1	1.0	4.5	6.2
Time Const., τ (sec)	205	160	132	110
Deadtime, T (sec)	390	310	257	215

TABLE 6.1: Lumped Model Estimates (1st order plus deadtime)

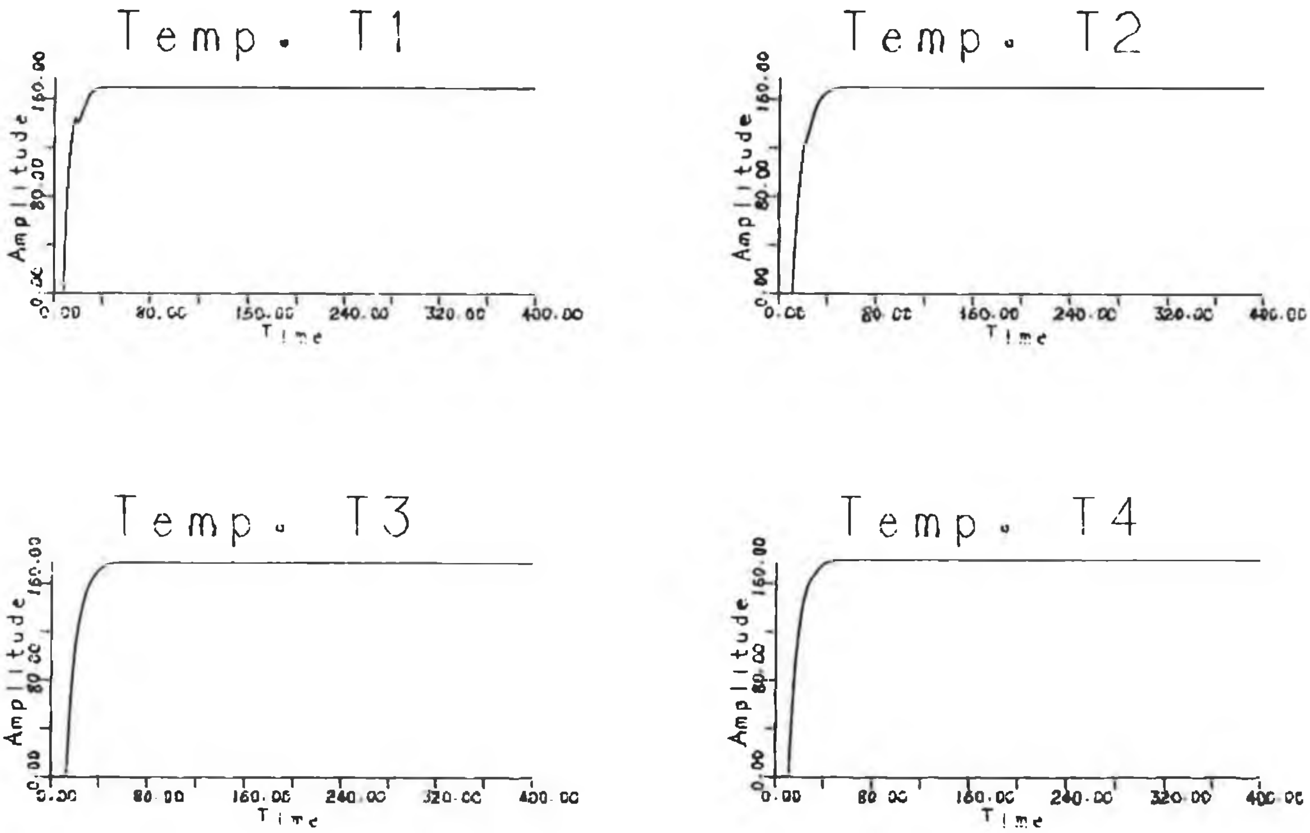

FIG. 6.4: Open-loop Steady State Point Temperatures

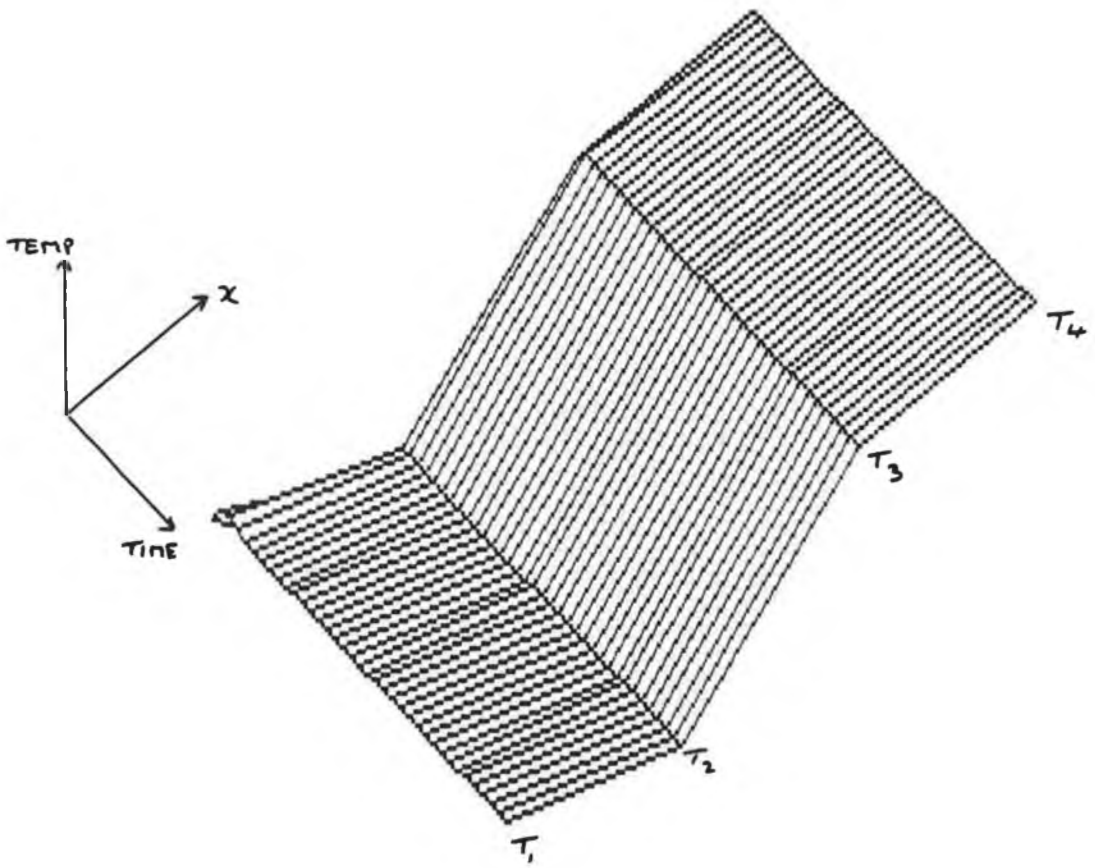

FIG. 6.5: Steady State Temperature Profile

itself as a large constant step disturbance on temperature T1.

(2) Small variations about the low mean value of the cold feed input. These may be modelled as small magnitude short term step disturbances also on measured temperature T1.

(3) Zone 3 experiences large frictional heating due to the stripping of the outer soft skin of the pellets by friction, from rubbing against each other and the blades of the screw. The heat produced depends on the speed of the screw and can be modelled as small step disturbances on temperature T3.

(4) Variations in the mains power causes fluctuations in the heater power input applied to the barrel. This exhibits itself as short step disturbances on the applied MV's.

Measurement noise observed on the four thermocouples is simulated as Gaussian white noise additive to the temperature outputs.

To obtain realistic simulation results mismatch between the process models and the internal control models must be included. Mismatch is introduced in the form of different gains, deadtimes, and time constants.

6.5 FEEDBACK TOPOLOGY

One of the choices to be made as part of DAC is that of the feedback control topology. That is, which actuator input to couple with which outputs, the rank of the controller, etc. For this particular application two possible control topologies may be employed.

6.5.1 Multivariable Topology

One possible topological structure is to treat the system as a two-input four-output plant and to use 2x2 multivariable PFC controller blocks with the recursive form of the decision logic.

With this topology it is possible to always control one of the outputs assigned to a demand following role. Given the discussion of section 6.2 tight control of T4 is more important than that of T2. Thus T4 is chosen as the output to always regulate at its required level. The two zone outputs must be prioritised and two 2x2 PFC blocks implemented for each. Using the assigned priorities recursive decision logic is used to switch between the regulators.

A criticism of this approach is that in extreme circumstances only one of the constrained, or zone outputs, may be guaranteed due to the prioritisation. The output T4 with one of the zone outputs will always be controlled while the other constrained output may deviate outside its bound. This is highly undesirable in the case of an extrusion process for the safety reasons mentioned before. It would be preferable to relax control on T4 and ensure that both zone outputs meet their constraints. This is possible using the two degrees of freedom available and would not compromise the overall control strategy as it is the melt temperature at the die that must be regulated exactly to ensure product quality. This may be achieved with the primary loop, as shown in chapter 4, using either screw speed or back valve position. Relaxing control on T4 just increases the level of difficulty of this task.

This new control objective may also be met using the LBS approach. A multivariable topology, i.e. 2x2 PFC blocks, could be employed similar to the previous discussion. However, the decision logic required becomes more complicated as may be seen by consideration of the solution loci on the common input domain. A simpler topology may be used to meet this objective which only uses SISO PFC blocks.

6.5.2 Feedforward Compensation

This topology considers the extrusion process as two separate stages, each a one-input two-output subsystem. The first two zones (i.e. T1 and T2) are grouped together in stage 1, with the remaining zones (i.e. T3 and T4) forming stage 2 as shown in fig. 6.6.

Heater 1 is then the only MV for stage 1 and heater 2 may be considered a known measurable disturbance. Similarly, stage 2 treats the input of heater 1 as a known measurable disturbance. Thus, using SISO PFC blocks with feedforward compensation a separate one-input two-output LBS controller may be designed for each of the two stages. This strategy is presented in fig. 6.7.

Using this approach the top priority outputs are T1 and T3. The output constraints on these will be guaranteed under extreme conditions and the control of T2 and T4 relaxed. Another advantage of this strategy is that tuning of the SISO PFC blocks is considerably easier than the multivariable case. This feedforward topology is used in the full simulation presented below.

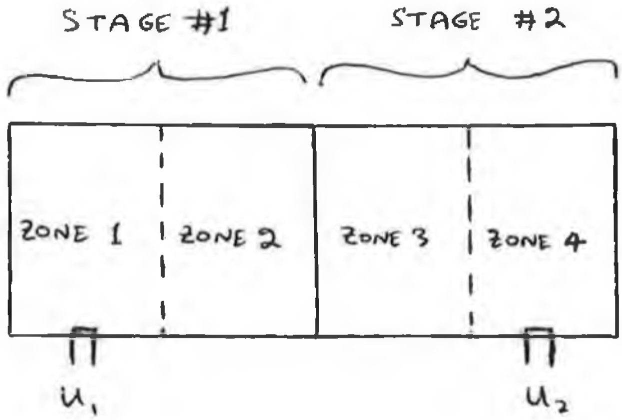

FIG. 6.6: Division of Extruder into Two Stages

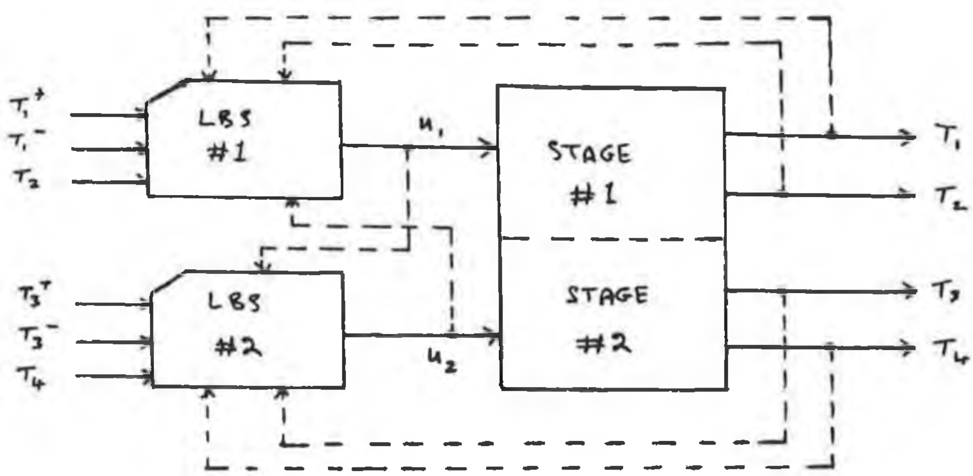

FIG. 6.7: LBS Control Strategy

6.6 SIMULATION RESULTS

From the steady state analysis the heater positioning was decided as shown in fig. 6.2. The achievable temperature profile, with no disturbances present, was then determined as previously shown in fig. 6.4 and fig. 6.5.

The control objective is to regulate T2 and T4 tightly at the following values:

T2 = 170 °C   T4 = 180 °C

and to constrain T1 and T3 within specified limits. The limiting constraints are deduced from the discussion of section 6.2. The zone boundary on T3 is much narrower than that for T1. The critical limit for T1 is the upper constraint to prevent meltback. The lower constraint is less critical. Measurement T3 is taken at a more critical part of the process. It is important that this temperature is kept significantly higher than T2 while not too high to cause the ploymer chains to break down. The zone constraints used are:

$T1^{+}$ = 180 °C $T1^{-}$ = 60 °C
$T3^{+}$ = 182.5 °C $T3^{-}$ = 177 °C

To obtain a realistic simulation disturbances, noise and process-model mismatch must be considered. Typical disturbances were discussed in section 6.4. These were simulated with the following magnitudes:

(1) Constant -10°C step on T1 (cold feed input)
(2) Short term ±5°C additional step on T1 (variations in feedstock temperature)
(3) Short term ±3°C step on T3 (heat changes due to friction caused by screw speed changes)
(4) Intermittent ±0.5% step changes on applied MVs (fluctuations in the power supply to the heaters)

Gaussian white noise is also added to each of the temperature measurements to simulate random measurement noise in the instrumentation.

Process-model mismatch is also considered. The parameters given in table 6.1 are used to simulate the process. The LBS controller is designed with control models of this process which have slightly different deadtimes, gains and time constants. The parameters of the control models are given in table 6.2.

Transfer Function	G_{11}	G_{21}	G_{31}	G_{41}
Gain, K (°C/%ton)	4.5	4.0	1.5	0.5
Time Const., τ (sec)	80	110	130	140
Deadtime, T (sec)	160	220	240	260
Transfer Function	G_{12}	G_{22}	G_{32}	G_{42}
Gain, K (°C/%ton)	0.12	0.9	4.6	6.3
Time Const., τ (sec)	200	150	130	105
Deadtime, T (sec)	380	300	260	220

TABLE 6.2: Control Model Parameters

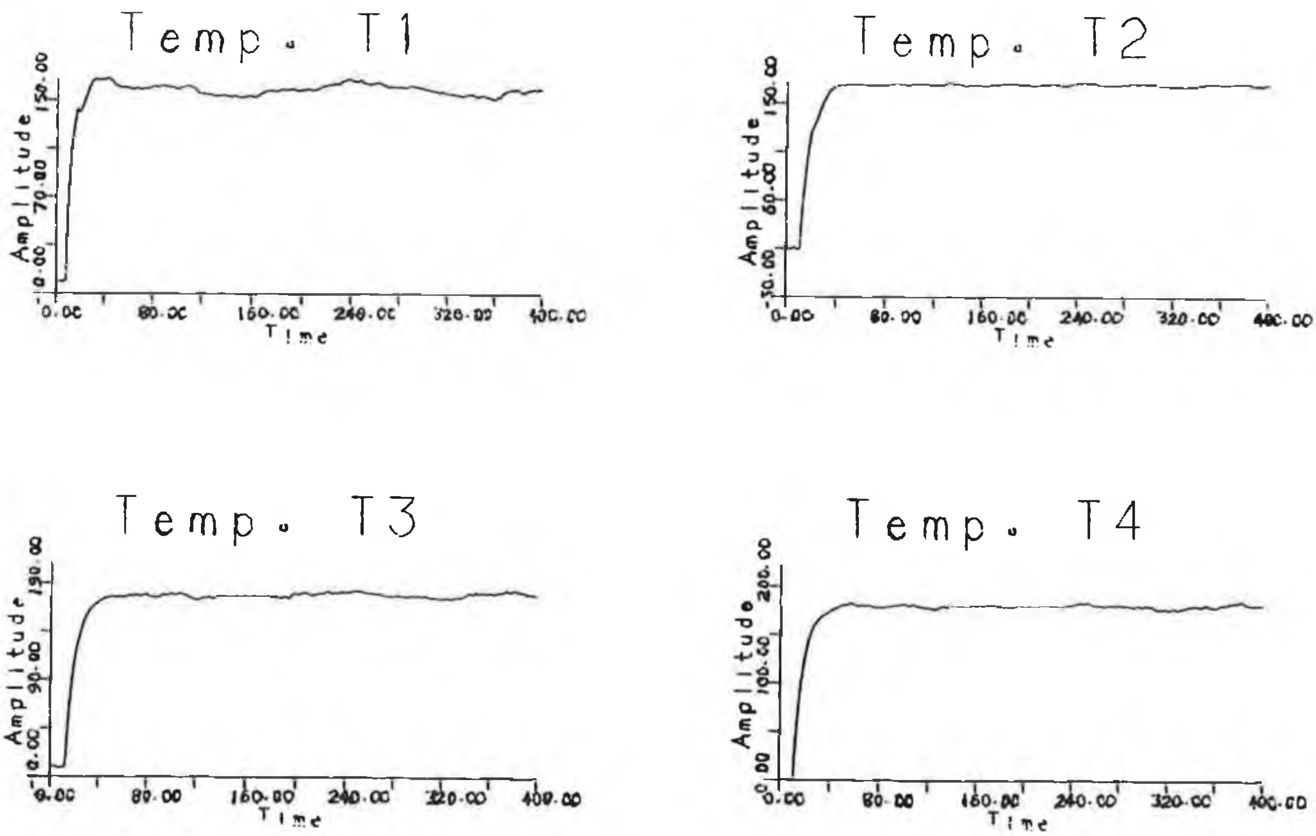

FIG. 6.8: Controlled Outputs with all Disturbances Present

The performance of LBS with the above disturbances, noise and process-model mismatch was investigated. The control results are shown in fig. 6.8 which is a plot of the point temperatures. The results show that T2 and T4 are controlled at their respective setpoints with T1 and T3 allowed to drift within their zone constraints. With the perturbations discussed above T1 and T3 will not violate their safety boundaries and hence LBS does not have to switch control regulators.
To illustrate the performance of LBS, it is useful to consider the effect of individual disturbances. Fig. 6.9 and fig. 6.10 show the point temperatures and profile response of the system to a large step disturbance on T1. T1 is allowed to drift and T2 is tightly regulated. There is no effect on T3 and T4. The responses to a step disturbance on T3 are shown in fig. 6.11 and fig. 6.12.

To show relaxation of the setpoint control when necessary untypically large disturbances have to be introduced. The lower constraint on T1 is also increased to 160 °C to aid demonstration. Fig. 6.13 and fig. 6.14 show the responses in this instance. The disturbance magnitudes are -20°C and +10°C on T1 and T3 respectively. The results show T1 regulated at its lower constraint (i.e. 160°C) and T3 controlled at its upper constraint (182.5°C). This causes offsets on T2 and T4 from their demand levels. These values are minimal at +8°C and -4°C respectively.

6.7 CONCLUSIONS

The application of LBS to a distributed actuator control (DAC) problem was looked at. In particular, control of extruder barrel wall temperature profile was considered. The use of LBS allowed the number of actuators normally employed for this purpose to be reduced by half. This results in considerable savings in cost and maintainence, and increases reliability with a reduced component count. Given this actuator minimisation attention had to be paid to the placement of the actuators. To achieve the temperature profile for this application only one possible configuration exists. This was determined from open-loop steady state analysis.

Several possible controller feedback topologies may be employed. It was found that the simplest was to treat the 2x4 extrusion process as two separate 1x2 subsystems, or stages. Two LBS controllers could then be designed, one for each stage. The basic SISO PFC blocks employed in each utilise feedforward compensation of external measurable disturbances. The heater input for control of stage 2 is treated as such a disturbance by the LBS controller for stage 1, as the control input to stage 2 is treated as a disturbance on stage 1. This topology allows the two zone outputs (T1 and T3) to be assigned equal priorities with the demand outputs (T2 and T4) given a lower priority. Thus, control of

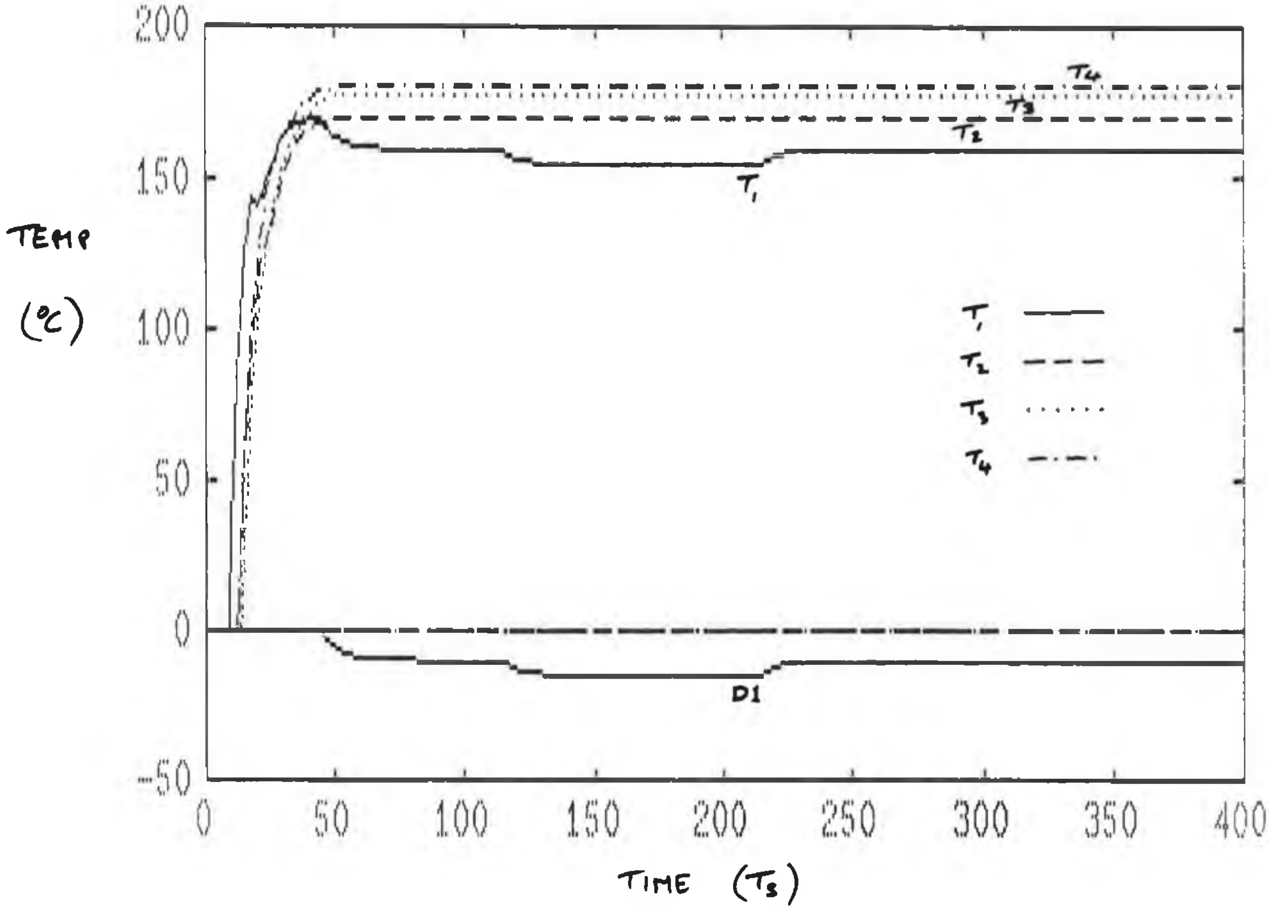

FIG. 6.9: Disturbance on T1

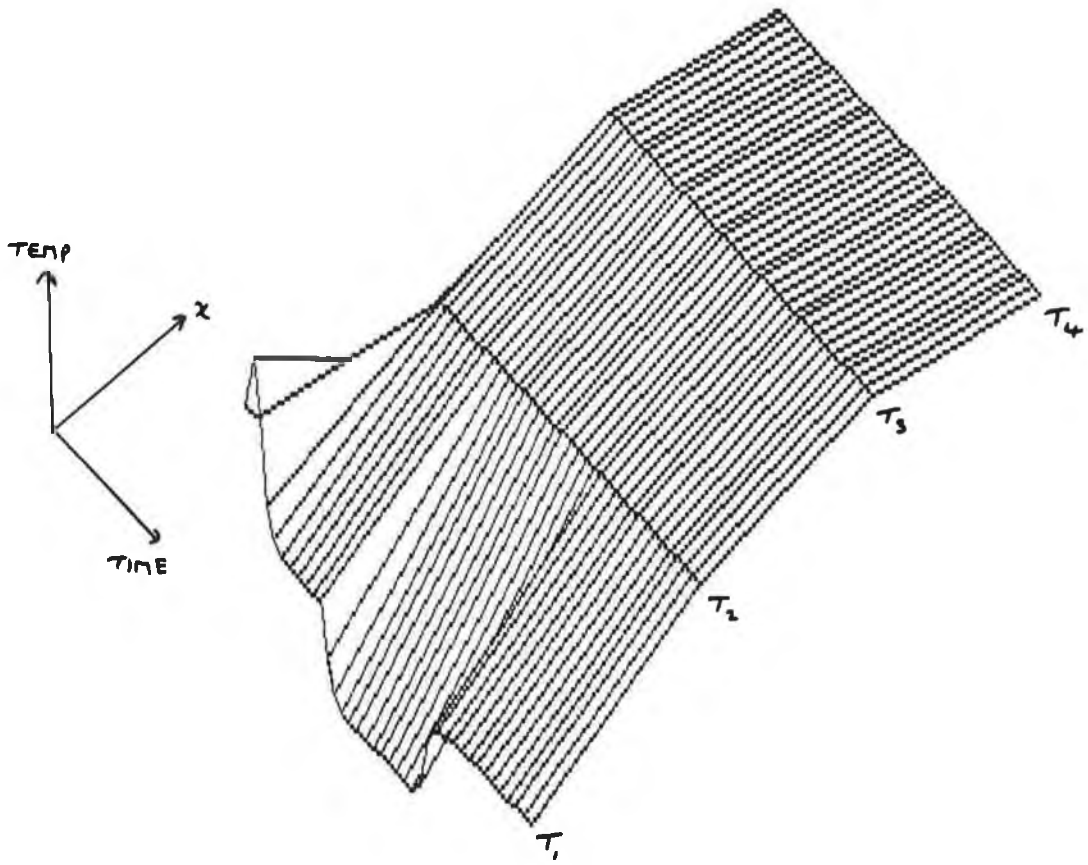

FIG. 6.10: Temperature Profile - Disturbance on T1

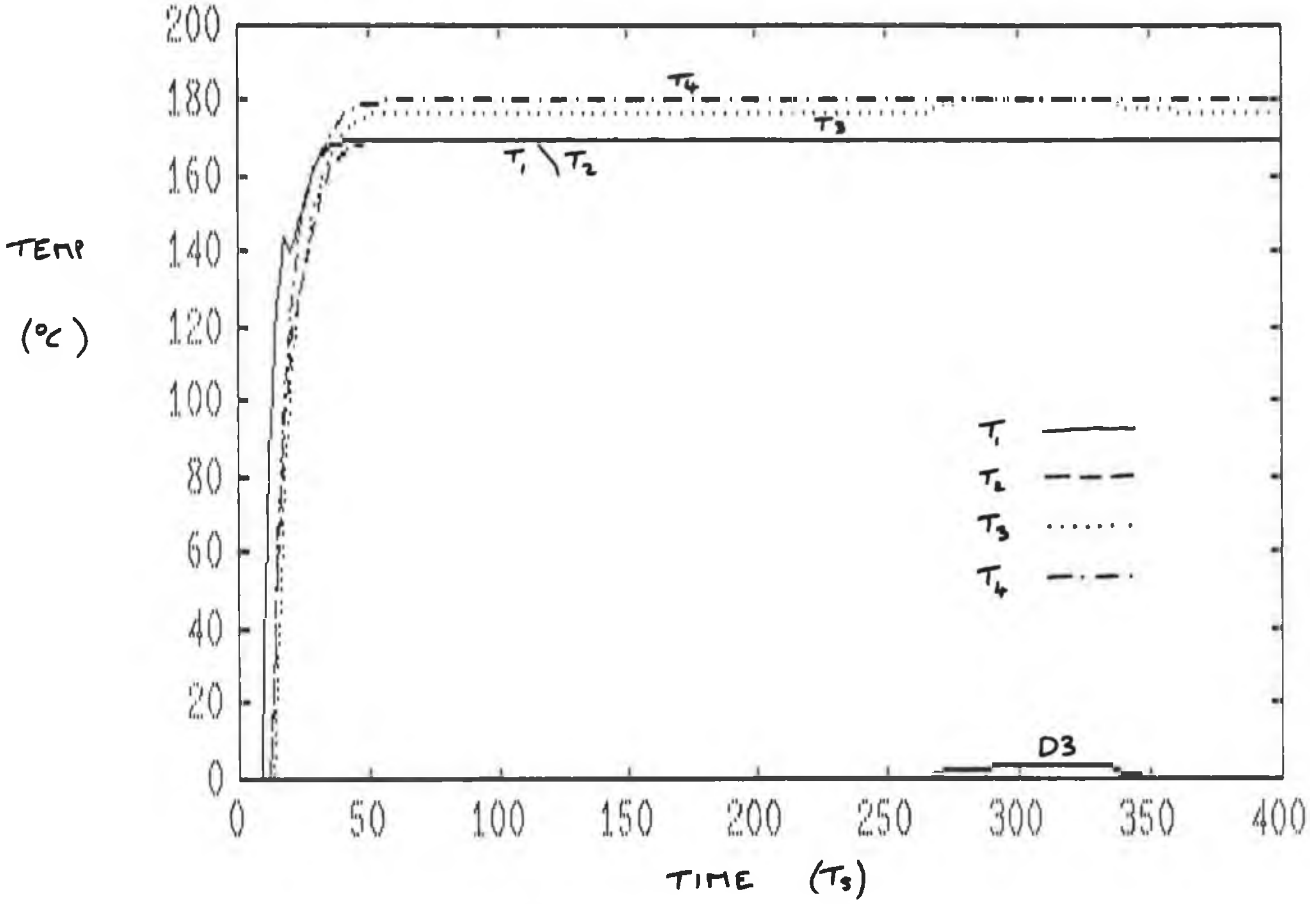

FIG. 6.11: Disturbance on T3

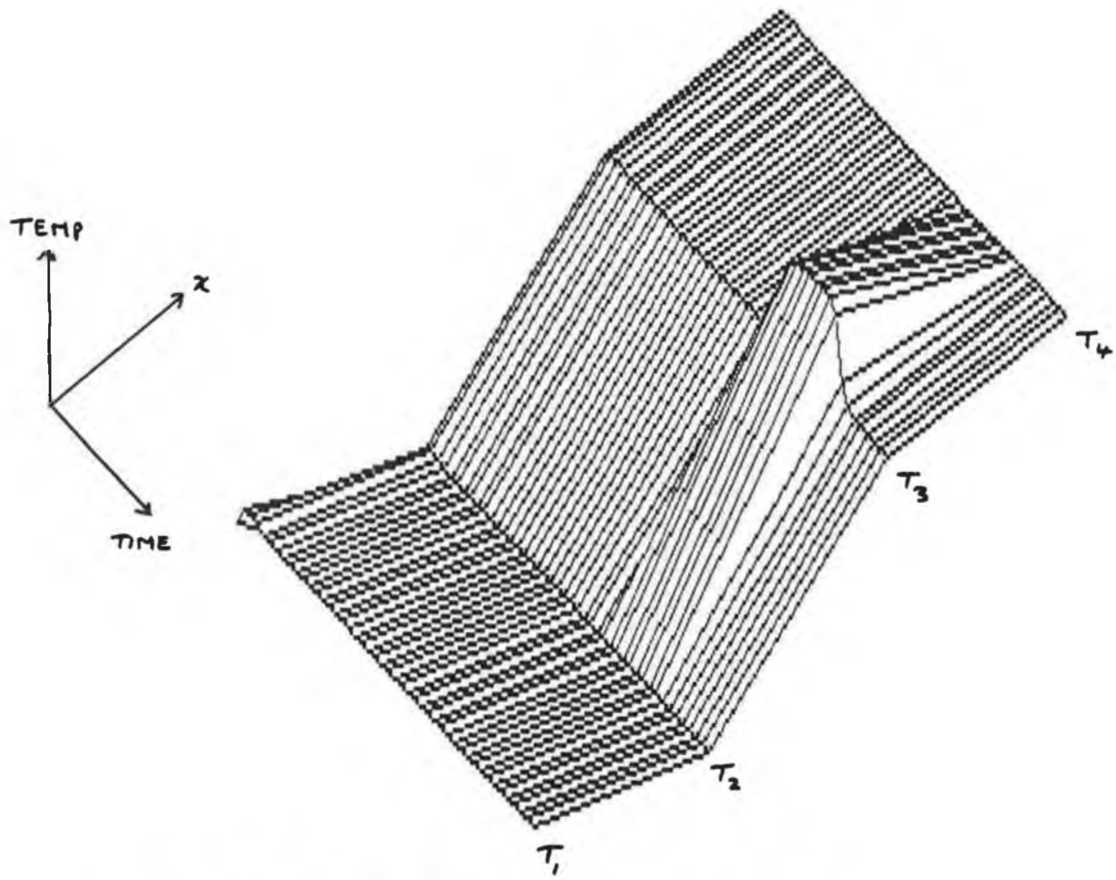

FIG. 6.12: Temperature Profile - Disturbance on T3

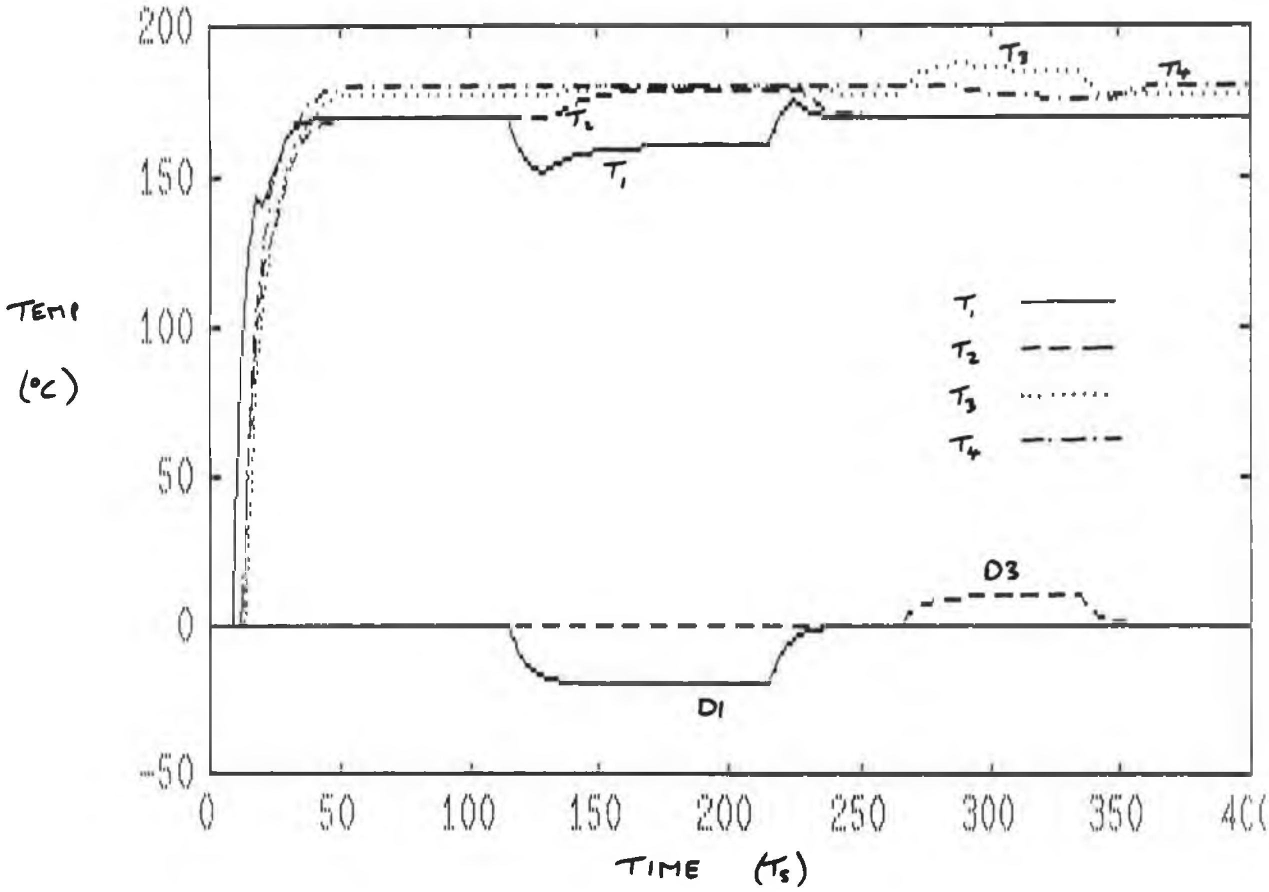

FIG. 6.13: Disturbances on T1 and T3

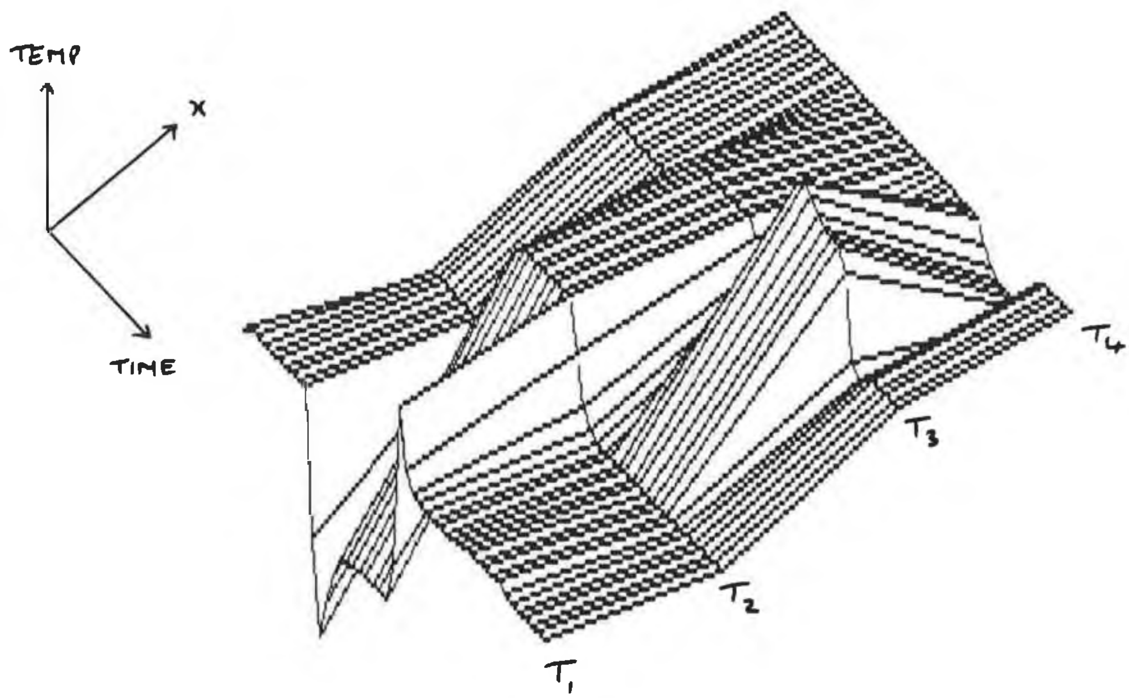

FIG. 6.14: Temperature Profile - Disturbances on T1 and T3

both T2 and T4 will be relaxed if necessary to ensure that T1 and T3 do not breach their safety constraints. Allowing offsets on T4 does not necessarily compromise the overall control objective, i.e. a constant quality product extruded at the die. The primary control loop involving screw speed and\or back valve position can compensate for the offset on T4 to achieve constant melt temperature at the die. LBS ensures the safety of the process although the quality of the extruded product may be degraded.

A full simulation of the use of LBS for barrel wall temperature control was presented. Modelling realistic disturbances, which are typically of a step form in nature, and measurement noise, the simulation results showed that LBS controlled the temperature profile using a reduced number of actuators very efficiently. Extremely large and unrealistic disturbances had to be employed to illustrate the performance characteristics in switching between different regulators.

CONCLUSIONS AND RECOMMENDATIONS

A detailed review of the relatively novel field of intelligent control was presented. This is a multi-disciplinary area with research progressing along many independent and diverse paths. Several different explicit definitions have been proposed with many implicit definitions generated by the nature of particular research paths. Intelligent control has become intimately associated with the development of autonomous robotics and systems. The use of particular tools, eg. expert systems or neural networks, is often considered to imply 'intelligent' control. Most work is concentrated on application specific solutions to problems, with little consideration of the underlying characteristics desirable. Most research effort is concentrated on higher-level intelligent functions, eg. planning, problem solving, and reasoning, to the detriment of the low-level controller employed. There is obvious need for a unifying framework. Such a theory should allow current research strands to be included and yet provide inspiration for new areas of investigation. A framework would strengthen the core principles and allow new ideas to flourish from a strong base.

A central theme evident from the survey conducted is an association, usually implicit, with emulation of one or more aspects of human intelligence or intelligent behaviour. A new definition of intelligent control was proposed explicitly defining the use of intelligent human behaviour as a reference model. A framework was developed based on this principle. The study of intelligent behaviour provides a source of constraints that may be used to analyse, or design, an intelligent control system. Consideration was given to psychological theories of intelligence. A set of necessary and sufficient conditions were derived from these. Learning ability was recognised as a sufficient condition for intelligent control. Although the exact nature of intelligence remains unclear, all theories agree that learning is a crucial element. Some additional characteristics of intelligent behaviour were identified as necessary conditions for learning. Prediction is a major element of learning theories. It guides learning by the accuracy of future predictions and in return improves its prediction mechanisms for better performance. The use of an online internal world model is intimately linked to this mechanism. Similarly, low-level parameter learning by model updating or adaptation is necessary. Learning is an active process which continuously occurs. Manipulation of the environment causes learning to happen and manipulation often has a dual role, both to manipulate and to probe. Four necessary conditions were thus derived.

The design of a learning based predictive controller (LPBC) was considered within the proposed framework. This can be viewed as an extension of classical adaptive control schemes. Adaptive control employs a simple form of parameter learning and is therefore a first step towards intelligent control. LBPC extends this theory to include the necessary conditions identified. Long range predictive control strategies form the basis of LBPC design. They employ future predictions of proposed actions and use on online internal model for this purpose. Parameter learning is accomplished by the addition of a recursive estimation scheme to update the model parameters with 'experience', i.e. input-output data. This feature thus employs sub-optimal probing. The applied control inputs are used for learning but their calculation does not account for uncertainties in the prediction model. This is the *certainty equivalence* principle. Despite this disadvantage, the method can be used as a first approach to intelligent control design within the framework. Future research should try to incorporate a dual-type control action for active probing.

Predictive Functional Control was chosen as the LRPC strategy. This was reformulated in terms of an ARMAX plant model to facilitate the addition of an adaptive layer. Recursive least squares was employed with a variable forgetting factor to counteract the causes of estimator windup and bursting. The stability and robustness of the ARMAX form of PFC was examined. Its performance was tested on several different plants under varying conditions and disturbances. It operated successfully in all cases and could control a nonminimum phase process without difficulty. The adapted version proved to be equally successful and was also able to control a nonminimum phase plant. This compares very favourably to other classical adaptive control laws which cannot control such difficult processes.

Complex, large-scale, spatially distributed systems were identified as a suitable application area for hierarchical intelligent control techniques. The distinguishing features of these problems include their compliexity, difficulties in modelling, and choice of several prossible control topologies among others. A plasticating extrusion process was selected as a particular example of this class of problem. A literature review revealed the difficulties encountered for modelling and control of these processes. Control schemes usually have two elements. The primary loop controls either temperature or pressure (or both) of the melt at the die inlet. These parameters are related to extrudate quality. A second consideration is control of the temperature profile of the extruder barrel wall. This profile is important for product quality and safety reasons.

The performance of the ARMAX PFC and adaptive PFC regulators was tested for control of the primary loop by simulation. Comparison of the results against other adaptive regulators applied to the same simulation models demonstrated their good operating characteristics. Although die temperature and pressure control are distributed parameter problems typical extruder construction reduces the control problem to a standard 'point' control objective.

Barrel wall temperature profile control is a Distributed Actuator Control (DAC) problem. This involves several choices and design decisions. Being spatially distributed and requiring profile control, one issue is the number and location of measurement points. A similar issue is the number and location of actuators to employ which has a considerable bearing on the achievable profile. Decisions as to which actuators and sensors must be matched and feedback topologies are also part of DAC. Constraints may exist due to typical construction practice. The system will invariably be multivariable with current control methods requiring 'square' systems (equal number of inputs and outputs). There is motivation to consider reducing the number of actuators or to use more sensors. Predictive control methods are desirable due to the large time constants and excessive time delays of the process dynamics.

A multivariable control technique for non-square systems with more outputs than inputs was proposed. This has the flavouring of some current appraoches to intelligent control. In global terms it is a two-level hierarchy that combines the benefits of single-loop or multivariable predictive controller blocks (LBPC) used at the lowest level with a decision logic block at the higher level to offset the degree of freedom deficiency. The decision logic switches smoothly between the low-level controllers based on the computed actuator signals. This Logic Based Strategy leads to smooth actuator control signals and pre-emptive action before a problem would normally cause an alarm. Zone constraints may thus be set at their maximum. The logic is a simple scalar comparison regardless of the order of the plant. Single-input geometric analysis extends elegantly to the general 'n'-input case.

LBS was demonstrated on academic multivariable examples. The correct operation and advantages of the strategy was observed. Use of LBS for DAC was considered applied to extruder barrel wall temperature profile control. Within construction and control objective constraints it was shown that successful profile control could be achieved with two actuators and four temperature measurement points. Two single-input LBS blocks were used with feedforward compensation employed to account for the effects of the other. LBS may be utilised with any control law design in conjunction with the decision logic. It is preferable though to use LBPC blocks in line with the intelligent control framework.

Several research possibilities exist as derivations of the work presented here. A more detailed study of intelligent behaviour is desirable to produse a rigorous set of constraints such that a detailed computational theory of intelligent control could be built. A more complete set of necessary and sufficient conditions could then be proposed. This should consider the hierarchical nature of intelligent control systems and refer to both high and low levels. The nature of learning mechanisms at these levels should be investigated especially with regard to the efficient integration of several levels.

As previously mentioned, LBPC formulation presented here is a first approach to intelligent control within the framework. A sub-optimal probing technique is used through recursive least squares and certainty equivalence. The use of dual control action, which applies inputs with objectives to both regulate the process and probe it to acquire more information, is preferable. Such action accounts for uncertainties in the online model used for prediction and is more in line with intelligent behaviour.

Relatively few detailed theoretical analyses of the properties of predictive control have been presented. Although good performance was demonstrated in simulation, rigorous theoretical analyses of adaptive predictive control is still required for stability and robustness proofs.

LPBC employs a self-compensator to account for disturbances of degree one or greater. Most LRPC strategies designed around ARMAX models utilise the disturbance model availalbe for this purpose. LBPC could be extended to operate in this fashion. Interesting comparisons could then be made between the two approaches to disturbance compensation.

Further work is required to assess possible application areas suitable for intelligent control and requiring the characteristics inherent in the use of such a strategy. Further extensions to the theory of DAC would be welcome. Possibilities include the development of a CAD package to automate or provide help in the design stages of DAC. General theories of how to combine outputs with actuators would also be beneficial, especially with regard to optimal positioning and numbers of actuators and sensors.

Behavioural learning theories used to develope the framework have also been applied in learning automata and in neural network research. The Associative Search Network is a good example. It would be interesting to investigate the relationship between these topics. Use of learning theories could also find successful application as learning rules in fuzzy logic controllers or in expert systems.

Intelligent control is a relatively new and immature field of research and will undoubtedly undergo many changes and modifications in the future. Future research and industrial needs will further refine its definition. The work described here has identified the core principles and presented a framework to allow future developments to proceed in a uniform and creative fashion.

REFERENCES

[1] Bode, H.W.: *NETWORK ANALYSIS AND FEEDBACK AMPLIFIER DESIGN*, D. VON NOSTRAND, NEW YORK, 1945

[2] Nyquist, H.: "*Regeneration Theory*", Bell System Tech. Jour., Vol.11, pp126-144, 1932

[3] Astrom, K.: *INTRODUCTION TO STOCHASTIC CONTROL THEORY*, Academic Press, 1970

[4] Astrom, K.J. "*Adaptive Feedback Control*", Proc. IEEE, Vol.75, No.2, pp185-217, 1987

[5] Astrom,K.J. "*Theory and Applications of Adaptive Control - A Survey*", Automatica, Vol.19, No.5, pp471-486, 1983

[6] Astrom, K.J.; Anton, J.J.; Arzen, K.E.: "*Expert Control*", Automatica, Vol.22, No.3, pp277-286, 1986

[7] Fu, K.S.: "*Learning Control Systems and Intelligent Control Systems: An Intersection of Artificial Intelligence and Automatic Control*", IEEE Trans. Automatic Control, Vol.16, No.1, pp70-72, 1971

[8] Saridis, G.N.: "*Toward the Realization of Intelligent Controls*", Proc. IEEE, Vol.67, No.8, 1979

[9] Astrom, K.J.: "*Toward Intelligent Control*", IEEE Control Systems Mag., pp60-64, April, 1989

[10] Bavarian, B.: "*Introduction to Neural Networks for Control*", IEEE Control Systems Mag., pp3-7, April, 1988

[11] Mesarovic, M.D.: *THEORY OF HIERARCHICAL MULTILEVEL SYSTEMS*, Academic Press, New York, 1970

[12] Saridis, G.N.; Valavanis, K.P.: "*On the Theory of Intelligent Controls*", SPIE Vol.848, Intelligent Robots and Computer Vision, pp488-495, 1987

[13] Saridis, G.N.: "*A Hierarchical Approach to the Control of a Prosthetic Arm*", IEEE Trans. Syst., Man., Cyb., Vol.7, No.6, pp407-420, 1977

[14] Saridis, G.N.; Valavanis, K.P.: "*Information Theoretic Approach for Knowledge Engineering and Intelligent Machines*", Proc. 1985 American Control Conference, pp1098-1103, 1985

[15] Saridis, G.N.: "*Intelligent Robotic Control*", IEEE Trans. Auto. Control, Vol.28, No.5, pp547-557, 1983

[16] Saridis, G.N.; Graham, G.H.: "*Linguistic Decision Schemata for Intelligent Robots*", Automatica, Vol.20, No.1, pp121-126, 1984

[17] Saridis, G.N.: "*Control Performance as an Entropy*", Control Theory and Advanced Technology, Vol.1, No.2, 1985

[18] Saridis, G.N.; Valavanis, K.P.: "*Analytical Design of Intelligent Machines*", Automatica, Vol.24, No.2, pp123-133, 1988

[19] Hayes-Roth, F.; Waterman, D.; Lenat, D.: *BUILDING EXPERT SYSTEMS*, Addison-Wesley, Massachusetts, 1983

[20] Antsaklis, P.J. et al.: "*Autonomous Control Systems: Architecture and Fundamental Issues*". Proc. 1988 American Control Conference, Atlanta, Georgia, 1988

[21] Stephanou, H.E.: "*An Evidential Framework for Intelligent Control*". IEEE Workshop on Intelligent Control, New York, 1985

[22] Pao, Y.H.: "*Some Views on Analytic and Artificial Intelligence Approaches*". IEEE Workshop on Intelligent Control, New York, 1985

[23] Tzafestas, S.G.: "*Integrated Sensor-Based Intelligent Robot System*". IEEE Control Systems Mag., April, 1988

[24] Hodgson, J.P.E.: "*Structures for Intelligent Control*". IEEE Symp. on Intelligent Control, Pennsylvania, 1987

[25] Erkmen, A.M.; Stephanou, H.E.: "*An Evidential Distance for Intelligent Control*". Proc. 25th Conf. Decision Control, Greece, 1986

[26] Meystel, A.: "*Cognitive Controllers for Autonomous Systems*". IEEE Workshop on Intelligent Control, New York, 1985

[27] Jackson, P.: "*Review of Knowledge Representation Tools and Techniques*". IEE Proc. Pt.D, Vol.134, No.4, 1987

[28] Nau, D.S.: "*Expert Computer Systems*". Computer, pp63-85, Feb. 1983

[29] Michalski, R.: *MACHINE LEARNING*, Ed: Michalski, Palo Alto, 1983

[30] Tzafestas, S.G.; Abu El Ata-Doss, S.; Papakonstantinou, G.: "*Expert System Methodology in Process Supervision and Control*". in KNOWLEDGE-BASED SYSTEM DIAGNOSIS, SUPERVISION, AND CONTROL, Ed: Tzafestas, Plenum, 1988

[31] Abu El Ata-Doss, S.; Brunet, J.: "*Online Expert Supervision for Process Control*". Proc. 25th Conf. Decision Control, Greece, 1986

[32] Gidwani, K.K.: "*The Role of Artificial Intelligence Systems in Process Control*". Proc. 1985 American Control Conference, 1985

[33] Freeman, D.D.: "*Artificial Intelligence Applications in Process Control*". Proc. 1985 American Control Conference, 1985

[34] Beaverstock, M.; Bristol, E.H.; Fortin, D.: "*Expert Systems as a Stimulus to Improved Process Control*". Proc. 1985 American Control Conference, 1985

[35] Kaemmerer, W.E.; Christopherson, P.D.: "*Using Process Models with Expert Systems to Aid Process Control Operators*". Proc. 1985 American Control Conference, 1985

[36] Karsai, G. et al.: "*Knowledge Based Approach to Real-Time Supervisory Control*". Proc. 1988 American Control Conference, Atlanta, Georgia, 1988

[37] Leech, W.J.: "*A Rule Based Process Control Method with Feedback*". ISA\86 Int. Conf. & Exhibit, Texas, April, 1986

[38] Erkmen, A.M.; Stephanou, H.E.: "*Sensor-Based Grasp Control: An Evidential Reasoning Approach*". Proc. IEEE Workshop on Languages for Automation, 1987

[39] Mina, I.: "*KMPR: An Experimental Knowledge-Based Modelling Prototype for Robots*", Proc. IEEE Conf. Robotics and Autom., Raleigh, NC, 1987

[40] Ong, K.K.; Seviora, R.E.; Dasiewicz, P.: "*Knowledge-Based Position Estimation for a Mutlisensor House Robot*" in APPLICATIONS OF AI IN ENGINEERING PROBLEMS, Springer-Verlag, 1986

[41] Arkin, R.C.: "*Motor Schema Based Navigation for a Mobile Robot*", Proc. IEEE Conf. Robotics and Autom., Raleigh, NC, 1987

[42] Dean, T.: "*High Level Planning and Low Level Control*", SPIE Vol.848, Intelligent Robots and Computer Vision, pp496-501, 1987

[43] Reynolds, D.E.; Boulton, C.B.; Martin, S.C.: "*AI Applied to Real-Time Control: A Case Study*", in APPLICATIONS OF AI IN ENGINEERING PROBLEMS, Springer-Verlag, 1986

[44] Bennett, M.E.: "*Real-Time Continuous AI Systems*", IEE Proc. Pt.D, Vol.134, No.4, pp272-277, July, 1987

[45] Porter, B.; Jones, A.H.; McKeown, C.B.: "*Real-Time Expert Controllers for Plants with Actuator Nonlinearities*", IEEE Symp. on Intelligent Control, Pennsylvania, 1987

[46] Astrom, K.J.: "*Adaptation, Auto-tuning, and Intelligent Control*", Workshop on Intelligent Control at 1988 ACC, Atlanta, Georgia, June, 1988

[47] Pang, G.K.H.: "*A Blackboard Control Architecture for Real-Time Control*", Proc. 1988 American Control Conference, Atlanta, Georgia, 1988

[48] Moore, R.L.; Hawkinson, L.B.; Levin, M.E.; Knickerbocker, C.G.: "*Expert Control*", Proc. 1985 American Control Conference, 1985

[49] Zadeh, L.A. "Fuzzy Sets", Information and Control, Vol.88, pp.338-353, 1965

[50] Zadeh, L.A.: "*Outline of a New Approach to the Analysis of Complex Systems and Decision Processes*", IEEE Trans. Sys., Man, Cyb., Vol.3, No.1, 1973

[51] McVicar-Whelan, P.J.: "*Fuzzy Sets for Man-Machine Interaction*", Int. J. Man-Mach. Studies, No.8, pp687-697, 1976

[52] Tang, K.L.; Mulholland, R.J.: "*Comparing Fuzzy Logic with Classical Controller Designs*", IEEE Trans. Sys., Man, & Cyb., Vol.17, No.6, 1987

[53] Mamdani, E.H.: "*Application of Fuzzy Logic to Approximate Reasoning using Linguistic Synthesis*", IEEE Trans. Computer, Vol.26, pp1182-1191, 1977

[54] Procyk, T.; Mamdani, E.H.: "*A Linguistic Self-Organizing Process Controller*", Automatica, Vol.15, pp15-30, 1979

[55] Braae, M.; Rutherford, D.: "*Fuzzy Relations in a Control Setting*", Kybernetes, Vol.7, pp185-188, 1978

[56] Kickert, W.; Van Nauta Lemke, H.: "*Application of a Fuzzy Controller in a Warm Water Plant*", Automatica, Vol.12, pp301-308, 1976

[57] Mamdani, E.H.; Assilian, S.: "*An Experiment in Linguistic Synthesis with a Fuzzy Logic Controller*", Int. J. Man-Mach. Studies, No.7, pp1-13, 1975

[58] Maiers, J.; Sherif, Y.S.: "*Applications of Fuzzy Set Theory*". IEEE Trans. Sys., Man, Cyb., Vol.15, No.1, 1985

[59] Umbers, I.; King, P. "*An Analysis of Human Decision-making in Cement Kiln Control and the Implications for Automation*". Int. J. Man-Mach. Studies, Vol.12, pp11-23, 1980

[60] Mandic, N.J.; Scharf, E.M.; Mamdani, E.H.: "*Practical Application of a Heuristic Fuzzy Rule-Based Controller to the Dynamic Control of a Robot Arm*". IEE Proc. Pt.D, Vol.132, pp190-203, 1985

[61] Rummelhart, D.E.; McClelland, J.L.: *PARALLEL DISTRIBUTED PROCESSING: EXPLORATIONS IN THE MICROSTRUCTURES OF COGNITION*. Vol.1&2, Cambridge: MIT Press, 1986

[62] Minsky, M; Papert, S.: *PERCEPTRONS*. Cambridge: MIT Press, 1969

[63] Jones, W.P.; Hoskins, J.: "*Back-Propagation*". BYTE, pp155-162, Oct., 1987

[64] Hill, W.F.: *LEARNING: A SURVEY OF PSYCHOLOGICAL INTERPRETATIONS*. 4th Ed., Harper&Row, New York, 1985

[65] Guez, A.; Eilbert, J.L.; Kam, M.: "*Neural Architecture for Control*". IEEE Control Systems Mag., pp22-25, April, 1988

[66] Hopfield, J.J.; Tank, D.W.: "*Neural Computation of Decisions in Optimization Problems*". Biol. Cyber., Vol.52, No.3, pp1-25, 1985

[67] Guez, A.; Protopescu, V.; Barhen, J.: "*On the Stability, Storage Capacity and Design of Nonlinear Neural Networks*". IEEE Trans. Sys., Man, Cyb., Vol.18, No.1, pp80-87, 1988

[68] Psaltis, D.; Sideris, A.; Yamamura, A.A.: "*A Multilayered Neural Network Controller*". IEEE Control Systems Mag., pp17-21, April, 1988

[69] Anderson, C.W.: "*Learning to Control an Inverted Pendulum with Connectionist Networks*". Proc. 1988 American Control Conference, Atlanta, Georgia, 1988

[70] Barto, A.G.: "*Adaptive Neural Networks for Learning Control: Some Computational Experiments*". IEEE Workshop on Intelligent Control, New York, 1985

[71] Barto, A.G.; Sutton, R.S.; Anderson, C.W.: "*Neuronlike Adaptive Elements that can Solve Difficult Learning Control Problems*". IEEE Trans. Sys., Man, Cyb., Vol.13, No.5, pp834-846, 1983

[72] Sutton, R.S.; Barto, A.G.: "*Toward a Modern Theory of Adaptive Networks: Expectation and Prediction*". Psychological Review, Vol.88, No.2, pp135-170, 1981

[73] McDonnell, M.; McCorkell, C.: "*Learning Based Predictive Control: An Approach to the Intelligent Control of Industrial Processes*". 1st Nat. Conf. AI & Cognitive Science, Dublin, 1988

[74] Workshop Report: "*Challenges to Control: A Collective View*". IEEE Trans. Automatic Control, Vol.32, No.4, pp275-285, April, 1987

[75] Wittenmark, B.; Astrom, K.J.: "*Practical Issues in the Implementation of Self-Tuning Control*". Automatica, Vol.20, No.5, pp595-605, 1984

[76] Kawamura, S.; Miyazaki, F.; Arimoto, S.: "*Realization of Robot Motion Based on a Learning Method*". IEEE Trans. Sys., Man, Cyb., Vol.18, No.1, pp126-136, 1988

[77] Arimoto, S.; Kawamura, S.; Miyazaki, F.: "*Bettering Operation of Robots by Learning*". J. Robotic Systems, Vol.1, No.2, pp123-140, 1984

[78] Oh, S.R.; Bien, Z.; Suh, I.H.: "*An Iterative Learning Control Method with Application for the Robot Manipulator*". J. Robotic Systems, Vol.4, No.5, pp508-514, 1988

[79] Bien, Z.; Huh, K.M.: "*Higher-order Iterative Learning Control Algorithm*", IEE Proc. Pt.D, Vol.136, No.3, pp105-112, May, 1989

[80] Craig, C.: *ADAPTIVE CONTROL OF MECHANICAL MANIPULATORS*. Addison-Wesley, 1988

[81] Bondi, B.; Casalino, G.; Gambardella, L.: "*On the Iterative Learning Control Theory for Robotic Manipulators*", IEEE J. Robotics & Automation, Vol.4, No.1, pp14-22, 1988

[82] Furuta,K.; Yamakita, M.: "*The Design of a Learning Control System for Multivariable Systems*". IEEE Symp. on Intelligent Control, pp371-376, Pennsylvania, 1987

[83] Kawamura, S.; Miyazaki, F.; Arimoto, S.: "*Intelligent Control of Robot Motion Based on Learning Method*". IEEE Symp. on Intelligent Control, pp365-370, Pennsylvania, 1987

[84] Narendra, K.S.; Thathachar, M.A.L.: "*Learning Automata - A Survey*". IEEE Trans. Sys., Man, Cyb., Vol.4, No.4, 1974

[85] Oomen, B.J.: "*Ergodic Learning Automata Capable of Incorporating A Priori Information*". IEEE Trans. Sys., Man, Cyb., Vol.17, No.4, pp717-723, 1987

[86] Thathachar, M.A.L.; Sastry, P.S.: "*A New Approach to the Design of Reinforcement Schemes for Learning Automata*". IEEE Trans. Sys., Man, Cyb., Vol.15, No.1, 1985

[87] Charniak, E.; McDermott, D.: *INTRODUCTION TO ARTIFICIAL INTELLIGENCE*. Addison-Wesley, 1985

[88] Rich, E.: *ARTIFICIAL INTELLIGENCE*. McGraw-Hill, 1983

[89] Meystel, A.: "*Intelligent Control: Issues and Perspectives*". IEEE Workshop on Intelligent Control, New York, 1985

[90] Saridis, G.N.: "*Intelligent Control: A New Engineering and Scientific Reality*". IEEE Workshop on Intelligent Control, New York, 1985

[91] Albus, J.S.: *BRAINS, BEHAVIOUR, AND ROBOTICS*. BYTE Publications, 1981

[92] Miller, W.T.: "*Sensor-Based Control of Robotic Manipulators Using a General Learning Algorithm*". IEEE J. Robotics Auto., Vol.3, No.2, 1987

[93] Stephanou, H.E.: "*Knowledge-Based Control Systems*". IEEE Workshop on Intelligent Control, New York, 1985

[94] Best, J.B.: *COGNITIVE PSYCHOLOGY*. West Publishing Co., 1986

[95] Ruokangas, C.C.; Black, M.S.; Martin, J.F.; Schoenwald, J.S.: "*Integration of Multiple Sensors to Provide Flexible Control Strategies*". Proc. IEEE Conf. Robotics Automation, San Francisco, 1986

[96] Chiu, S.L.; Morley, D.J.; Martin, J.F.: "*Sensor Data Fusion on a Parallel Processor*". Proc. IEEE Conf. Robotics Automation, San Francisco, 1986

[97] Flynn, A.: "*Combinig Sonar and Infrared Sensors for Mobile Robot Navigation*". Int. J. Robotics Research, Vol.7, No.6, Dec. 1988

[98] Allen, P.K.: "*Integrating Vision and Touch for Object Recognition Tasks*". Int. J. Robotics Research, Vol.7, No.6, Dec. 1988

[99] Stansfield, S.: "*A Robotic Perceptual System Utilizing Passive Vision and Active Touch*". Int. J. Robotics Research, Vol.7, No.6, Dec. 1988

[100] Int. J. Robotics Research, *Special Issue on Sensor Data Fusion*, Vol.7, No.6, Dec. 1988

[101] Harmon, S.Y.; Bianchini, G.L.; Pinz, B.E.: "*Sensor Data Fusion Through a Distributed Blackboard*". Proc. IEEE Conf. Robotics Automation, San Francisco, 1986

[102] Luo, R.C.; Lin, M.; Scherp, R.S.: "*The Issues and Approaches of a Robot Multi-Sensor Integration*". Proc. IEEE Conf. Robotics and Autom., Raleigh, NC, 1987

[103] Durrant-Whyte, H.F.: "*Sensor Models and Mulisensor Integration*". Int. J. Robotics Research, Vol.7, No.6, Dec. 1988

[104] Durrant-Whyte, H.F.: "*Consistent Integration and Propagation of Disparate Sensor Observations*". Int. J. Robotics Res., Vol.6, No.3, 1987

[105] Henkind, S.J.; Harrison, M.C.: "*An Analysis of Four Uncertainty Calculi*". IEEE Trans. Sys., Man, Cyb., Vol.18, No.5, pp700-714, 1988

[106] Pang, D.; Bigham, J.; Mamdani, E.H.: "*Reasoning with Uncertain Information*". IEE Proc. Pt.D, Vol.134, No.4, 1987

[107] Garvey, T.D.; Lesh, S.A.; Lowrance, J.D.; et. al.: "*The Theory and Practice of Evidential Reasoning*". Proc. IEEE Workshop on Languages for Automation, 1987

[108] Marr, D.: *VISION*. Freeman, 1982

[109] Fischler, ; Firschein, : *THE EYE. THE BRAIN. AND THE COMPUTER*. 1988

[110] Skinner, B.F.: *SCIENCE AND HUMAN BEHAVIOUR*. Macmillan, New York, 1953

[111] Skinner, B.F.: *BEYOND FREEDOM AND DIGNITY*. Knopf, New York, 1971

[112] Neisser, U.: *COGNITIVE PSYCHOLOGY*. Appleton-Century-Crofts, New York, 1967

[113] Rescorla, R.A.; Wagner, A.R.: "*A Theory of Pavlovian Conditioning*". in *CLASSICAL CONDITIONING II*, New York, 1972

[114] Flavell, J.H.: *THE DEVELOPMENTAL PSYCHOLOGY OF JEAN PIAGET*. Van Nostrand, New York, 1963

[115] Gagné, R.M.: "*The Acquisition of Knowledge*". Psychological Review, Vol. 69, pp355-365, 1962

[116] Richalet, J.; Papon, J.: "*Industrial Applications of Internal Model Control*". IFAC-IFIP-IMACS: 7th Conference on Digital Computer Applications to Process Control, Vienna, Austria, Sept., 1985

[117] DeKeyser, R.M.C.; Van de Velde, P.G.A.; Dumortier, F.A.G.: "*A Comparative Study of Self-adaptive Long-Range Predictive Control Methods*". Automatica, Vol. 24, No. 2, pp149-163, 1988

[118] Kramer, K.; Unbehauen, H.: "*Survey to Adaptive Long-Range Predictive Control*". 12th IMACS World Congress, Vol. 1, pp358-363, Paris, July, 1988

[119] Clarke, D.W.; Tuffs, P.S.; Mohtadi, C.: "*Self-tuning Control of a Difficult Process*". IFAC Identification and System Parameter Estimation, York, UK., 1985

[120] Anderson, B.D.O.: "*Adaptive Systems, Lack of Persistency of Excitation and Bursting Phenomena*". Automatica, Vol. 21, No. 3, pp247-258, 1985

[121] Abi Karam, M.; Abu El Ata, S.; Estival, J.L.; Richalet, J.: "*PFC: Cas Monovariable Linéaire*". ADERSA Report No. 1015, 1987

[122] Bruijn, P.M.; Verbruggen, H.B.; Appeldoorn, O.V.: "*Predictive Control: A Comparison and Simple Implementation*". IFAC Symposiun LCA: Components, Instruments, and Techniques for Low Cost Automation, Valencia, Spain, Nov., 1986

[123] Richalet, J.P.; Rault, A.; Testud, J.L.; Papon, J.: "*Model Predictive Heuristic Control: applications to industrial processes*". Automatica, Vol. 14, 1978

[124] Bruijn, P.M.; Verbruggen, H.B.: "*Model Algorithmic Control using impulse response models*". Journal A, Vol 25, No. 2, 1984

[125] Cutler, C.R.; Ramaker, B.L.: "*Dynamic Matrix Control: A computer control algorithm*". JACC, San Francisco, 1980

[126] Rouhani, R.; Mehra, R.K.: "*Model Algorithmic Control (MAC); Basic Theoretical Properties**". Automatica, Vol. 18, pp401-414, July 1982

[127] Bars, R.; Haber, R.: "*Robustness of Some Predicted Control Algorithms based on Nonparametric Models*". 12th IMACS World Congress, Paris, Vol. 2, July, 1988

[128] DeKeyser, R.; VanCauwenberghe, A.R.: "*Extended Prediction Self-Adaptive Control*". 7th IFAC Symp. on Ident. and Syst. Param. Estim., York, U.K., 1985

[129] Ydstie, B.E.: "*Extended Horizon Adaptive Control*". 9th IFAC World Congress, Budapest, Hungary, 1984

[130] Soderstrom, T.; Stoica, P.: *SYSTEM IDENTIFICATION*. Prentice Hall, New York, 1989

[131] DeKeyser, R.M.C.; van Cauwenberghe, A.R.: "*Simple Self-Tuning Multistep Predictors*". 6th IFAC Symp. on Ident. and Syst. Par. Est., Washington, June, 1982

[132] Rodellar, J.; Martin-Sanchez, J.: "*Predictive Structural Control*". Conf. on Structural Control, 1986

[133] Martin-Sanchez, J.M.; Shah, S.L.; Fisher, D.G.: "*A Stable Adaptive Predictive Control System*". Int. J. Control, Vol. 39, No. 1, pp215-234, 1984

[134] Mosca, E.; Zappa, G.: "*Convergence of Multipredictor Self-tuning Regulators under Mismatching Conditions*". Proc. 25th Conf. on Decision and Control, Athens, Greece, Dec., 1986

[135] Peterka, V.: "*Predictor-based Self-tuning Control*". Automatica, Vol. 20. No. 1, pp39-50, 1984

[136] Clarke, D.W.; Tuffs, P.S.; Mohtadi, C.: "*Generalised Predictive Control. Part 1: The basic algorithm. Part 2: Extensions and interpretations*". Automatica, Vol 23, No. 2, 1985

[137] Mohtadi, C.; Clarke, D.W.: "*Generalised Predictive Control, LQ, or Pole-placement: A unified approach.*". Proc. 25th Conf. on Decision and Control, Athens, Greece, Dec., 1986

[138] Clarke, D.W.; Mohtadi, C.: "*Properties of Generalized Predictive Control*". 10th IFAC World Congress, Vol. 10, pp63-74, Munich, West Germany, July, 1987

[139] Bitmead, R.R.; Gevers, M.; Wertz, V.: "*Optimal Control Redesign of Generalized Predictive Control*". IFAC Symposium Adaptive Systems in Control and Signal Processing, Glasglow, U.K., April, 1989

[140] Richalet, J.; Abu el Ata-Doss, S.; Delineau, L.; Estival, J.L.: "*Model Based Predictive Control of Exotic Systems*". World IFAC Congress, Talin, USSR, Aug., 1990

[141] Richalet, J.; Abu el Ata-Doss, S.; Arber, C.; Kuntze, H.B.; Jacubasch, A.; Schill, W.: "*Predictive Functional Control: Application to fast and accurate robots*". Proc. 10th IFAC World Congress, Munich, 1987

[142] Kuntze,H.-B.; Jacubasch, A.; Richalet, J.; Arber, Ch.: "*On the Predictive Functional Control of an Elastic Industrial Robot*". Proceedings of 25th Conference on Decision and Control Athens, Greece, Dec., 1986

[143] Tham, M.T.; Morris, A.J.: "*An Introduction to Self-Tuning Control*". 4th IEE Workshop on Self-Tuning and Adaptive Control, Oxford, March, 1987

[144] Tadmor, Z.; Klein, I.: "*Engineering Principles of PLasticaticating Extrusion*". Robert E. Kreiger Publishing Co., New York, 1978

[145] Lee, W.; Lee, L.J; "*Computer Control of Extrusion*". Comprehensive Polymer Science, Vol. 7 (Ed.: Aggarwal, S.L.), Pergamon, Great Britain, 1989

[146] Costin, M.H.; Taylor, P.A.; Wright, J.D.: "*A Critical Review of Dynamic Modelling and Control of Plasticating Extruders*". Polymer Eng. & Sci. Vol. 22, No. 7, pp393, Et Seq., 1982

[147] Hassan, G.A.; Parnaby, J.: "*Model Reference Optimal Steady-State Adaptive Computer Control of Plastics Extrusion Processes*". Polymer Eng. & Sci., Vol. 22, No. 5, pp276, Et Seq., 1981

[148] Dennis-Germuska, D.; Taylor, P.A.; Wright, J.D.: "*Adaptive and Multivariable Control of a Single Screw Extrusion System*". Canadian Journal Chem. Eng., Vol. 62 No. 6, pp790, 1984

[149] Stevenson, J.F.: "*Extrusion of Rubber and Plastics*". Comprehensive Polymer Science, Vol. 7 (Ed.: Aggarwal, S.L.), Pergamon, Great Britain, 1989

[150] Tadmor, Z.; Lipshitz, S.D.; Lavie, R.: "*Dynamic Model of a Plasticating Extruder*". Polymer Eng. & Sci. Vol. 14, No. 2, pp112-119, 1974

[151] Brauner, N.; Lavie, R.; Tadmor, Z.: "*Control of Plasticating Extruder*". Int. IFAC Conf. Instrum. & Autom. Paper Rubber & Plastics Ind Brussells, 1976

[152] Fenner, R.T.; Cox, A.P.D.; Isherwood, D.P.: SPE ANTEC Tech. Papers, Vol. 24, pp494,1978

[153] Reber, D.H.; Emerson Lynn, R.; Frech, E.J.: "*A Mathematical Model for Predicting Dynamic Behaviour of a Plasticating Extruder*". Polymer Engng. Sci., Vol. 13, No. 5,pp346-356, 1973

[154] White, D.H.; Schott, N.R.: "*Dynamic Testing of Plastics Extrusion Systems*". Proc. 30th Annual Technical Conference of Society of Plastics Engineers, Chicago, IL, pp797-801, 1972

[155] Fontaine, W,: Dissertation, Ohio State University, 1975

[156] Costin, M.H.; Taylor, P.A.; Wright, J.D.: "*On the Dynamics and Control of a Plasticating Extruder*", Polymer Eng. & Sci., Vol. 22, No. 17, pp1095, Et Seq., 1982

[157] Chan, D.; Nelson, W.; James Lee, L.: "*Dynamic Behaviour of a Single Screw Plasticating Extruder Part II: Dynamic Modelling*". Polymer Eng. Sci., Vol. 22 No. 2 pp152-161, 1986

[158] Nelson, R.W.; Chan, D.; Yang, B.; James Lee, L.: "*Dynamic Behaviour of a Single Screw Plasticating Extruder Part I: Experimental Study*". Polymer Eng. Sci., Vol. 26 No. 2 pp144-151, 1986

[159] Chan, D.; Lee, L.J.: "*Dynamic Modelling of a Single Screw Plasticating Extruder*", Spe. Antec Tech. Papers, Vol. 30, pp77-80, 1984

[160] Kochhar, A.K.; Parnaby, J.: "*Comparison of Stochastic Identification Techniques for Dynamic Modelling of Plastics Extrusion Processes*". Proc. Instn. Mech. Eng., Vol. 192, pp299, Et Seq., 1978

[161] Kochhar, A.K.; Parnaby, J.: "*Dynamical Modelling and Control of Plastics Extrusion Processes*". Automatica, Vol. 13, pp177, Et Seq., 1977

[162] Bezanson, L.W.; Harris, S.L: "*Identification and control of an extruder using multivariable algorithms*". IEE Proc., Vol. 133, Pt. D, No. 4, July, 1986

[163] Bezanson, L.W.; Harris, S.L.: "*Identification and Control of an Extruder Using Multivariable Algorithms*". Iasted Journal Control and Computing, Vol. 13, No. 1, pp145-152, 1985

[164] Wolovich, W.: "*Linear Multivariable Systems*". Springer Verlag, New York, 1974

[165] Goodwin, G.C.; McInnis, B.C., Wang, J.C.: "*Model reference adaptive control for systems having non-square transfer functions*". Proc. 21st CDC, Orlando, Florida, 1982

[166] Wang, F.; Lang, S.: "*Periodic tracking adaptive control for multivariable systems having more outputs than inputs*". IEEE Trans. Auto. Contr., Vol AC-32, No. 12, 1987

[167] McDonnell, M.; Abu el Ata-Doss, S.: "*Predictive Functional Control of Multivariable Systems with More Outputs than Inputs*". 28th Conf. on Decision and Control, Florida, December, 1989

[168] McDonnell, M.: "*Single-Input Multiple-Output control using MONOREG*". ADERSA Report No. 1155, 1988

[169] McDonnell, M.; McCorkell, C.; Abu el Ata-Doss, S.: "*Multivariable Control Strategy using Decision Logic*". 6th IMC Conf. on Adv. Manuf. Tech., Dublin, August, 1989

[170] Pohjolainen, S.: "*On the Optimal Tuning of a Robust Controller for Parabolic Distributed Parameter Systems*", Automatica, Vol. 23, No. 6, 1987

[171] Khargonekar, P.P.; Poolla, K.: "*Robust Stabilization of Distributed Systems*", Automatica, Vol. 22, No. 1, pp77-84, 1986

[172] Burke, S.E.; Hubbard, J.E.: "*Distributed Actuator Control Design for Flexible Beams*", Automatica, Vol. 24, No. 5, pp619-627, 1988

[173] Kobayashi, T.; Luo, Z.H.: "*Dynamic Shape Control for a Flexible Beam*", Int. J. Systems SCI., Vol. 19, No. 6, pp985-997, 1988

[174] Kelemen, M.; Kannai, Y.; Horowitz,I.: "*One-Point Feedback Approach to Distributed Linear Systems*", Int. J. Control, Vol. 49, No. 3, pp969-980, 1989

[175] Korbicz, J.; Zgurovsky, M.Z.; Novikov, A.N.: "*Suboptimal Sensors Location in the State Estimation Problem for Stochastic Non-Linear Distributed Parameter Systems*", Int. J. Systems Sci., Vol. 19 No. 9, pp1871-1882, 1988

[176] Issa, R.I.; Spaulding, D.B.: "*Unsteady One-Dimensional Compressible Frictional Flow with Heat Transfer*", J. Mech Eng. Sci. Vol. 14, pp365, Et Seq., 1972

[177] Leffew, K.W.; Stiso, M.J.; Langhorst, H.: "*Application of digital control techniques to a laboratory extrusion process*", Proc. 1987 ACC, 1987